DISSERTATION

SECURE, ACCURATE, REAL-TIME, AND HETEROGENEITY-RESILIENT INDOOR LOCALIZATION WITH SMARTPHONES

Submitted by

Saideep Tiku

Department of Electrical and Computer Engineering

In partial fulfillment of the requirements

For the Degree of Doctor of Philosophy

Colorado State University

Fort Collins, Colorado

Spring 2022

Doctoral Committee:

Advisor: Sudeep Pasricha

Shrideep Pallickara
Anthony Maciejewski
H. J. Siegel

ABSTRACT

SECURE, ACCURATE, REAL-TIME, AND HETEROGENEITY-RESILIENT INDOOR LOCALIZATION WITH SMARTPHONES

The advent of the Global Positioning System (GPS) reformed the global transportation industry and allowed vehicles to not only localize themselves but also to navigate reliably and in a secure manner across the world at high speeds. Today, indoor localization is an emerging IoT domain that is poised to reinvent the way we navigate within buildings and subterranean locales, with many benefits, e.g., directing emergency response services after a 911 call to a precise location (with sub-meter accuracy) inside a building, accurate tracking of equipment and inventory in hospitals, factories, and warehouses, etc. While GPS is the de-facto solution for outdoor positioning with a clear sky view, there is no prevailing technology for GPS-deprived areas, including dense city centers, urban canyons, and inside buildings and other covered structures, where GPS signals are severely attenuated or totally blocked, and affected by multipath interference. Thus, very different solutions are needed to support localization in indoor locales.

Popular solutions for indoor positioning with high accuracy leverage wireless radio signals, such as WiFi, Bluetooth ultra-wideband (UWB), etc. Due to the existing widespread deployment of WiFi access points (WAPs) in most indoor locales, using WiFi for indoor localization can lead to low-cost solutions. Many localization algorithms that utilize these wireless signals have been proposed, e.g., based on the principles of proximity, trilateration, triangulation, and fingerprinting. Studies have shown that fingerprinting-based algorithms deliver higher accuracy, without stringent synchronization or line-of-sight requirements and enable greater error resilience in the presence of frequently encountered multipath signal interference effects, than other alternatives.

A fingerprinting-based approach for indoor localization has two phases. In an offline phase, location-tagged wireless signal signatures, i.e., fingerprints, at known indoor locations are captured along a path and stored in a database. Each fingerprint in the database consists of a location and wireless signal characteristics, e.g., received signal strength (RSSI; which varies as a function of distance from the WAP), from visible WAPs at that location. This phase requires great manual effort of collecting several fingerprints at each location and comes at considerable cost. In the online phase, the observed RSS on the user's mobile device is used to query the fingerprint database and determine location (potentially after some interpolation). Such WiFi-based fingerprinting is a promising building block for low-cost indoor localization with mobile devices.

Unfortunately, there are many unaddressed challenges before a viable WiFi fingerprinting based solution can be realized: (i) the algorithms used for the matching of fingerprints in the online phase have a major impact on accuracy, however the limited CPU/memory/battery resources in mobile devices requires careful algorithm design and deployment that can trade-off accuracy, energy-efficiency, and performance (localization decision latency); (ii) the diversity of mobile devices poses another challenge as smartphones from different vendors may have varying device characteristics leading to different fingerprints being captured at the same location; (iii) security vulnerabilities due to unintentional or intentional WiFi jamming and spoofing attacks can create significant errors which must be overcome; and (iv) short-term and long-term variations in WAP power levels and the indoor environments (e.g., adding/moving furniture, equipment, changes in density of people) can also introduce errors during location estimation, that often corrected by the expensive collecting new fingerprints.

In this dissertation, we propose a new real-time machine learning based framework called *SARTHI* that addresses all of the abovementioned key challenges towards realizing a viable indoor

localization solution with smart mobile devices. To enable energy-efficient enhancements in localization accuracy, *SARTHI* includes lightweight yet powerful machine learning algorithms with a focus on achieving a balance between battery life and response time. To enable device heterogeneity resilience, we analyzed and identified device diversity invariant pattern matching metrics that can be incorporated into a variety of machine learning based indoor localization frameworks. *SARTHI* also addresses the challenges associated with the security of fingerprinting-based indoor localization frameworks in the presence of spoofing and jamming attacks. This is achieved by devising a novel methodology for training and deploying deep-learning algorithms that are specifically designed to be resilient to the vulnerabilities associated with intentional power level variation-based attacks. Finally, *SARTHI* addresses the challenges associated with short-term and long-term variations in WiFi fingerprints using novel low-overhead relativistic learning-based deep-learning algorithms that can deliver high-accuracy while simultaneously minimizing the fingerprint collection effort in the offline phase.

ACKNOWLEDGEMENTS

I would like to thank all the individuals whose encouragement and support have made the completion of this thesis possible.

First and foremost, I would like to express my sincere gratitude to my advisor, Prof. Sudeep Pasricha, who has guided me through the process of doctoral study with his insightful and valuable advice. It was only with his encouragement and motivation I was able to explore some of the complex yet exciting problems in the domain of indoor localization with smartphones. He ensured that I improve my attention to detail while conducting research, which benefited me in both academic and non-academic activities. His devotion to both my research and my personal development were invaluable to me throughout my doctoral studies. He nurtured the analytical mindset and gave sufficient time and opportunities to realize my true potential in accomplishing my Ph.D. goals. I appreciate all the help, guidance, and inspiration I received from Prof. Pasricha, who made it possible for me to survive the trial of graduate school with unforgettable memories and broadened horizons.

I would like to take this opportunity to thank the respected members of my Ph.D. committee, Prof. Shrideep Pallickara, Prof. Anthony Maciejewski, and Prof. H. J. Siegel. Their feedback helped me to rediscover my research and refine my work from different perspectives. I also much appreciate all the help I received from my mentors at Cryptotronix, Fiat Chrysler Automobiles, Mentor Graphics (now Siemens), and Micron Technology in helping me shape my career.

Furthermore, my special thanks to my dear friends and colleagues at Prof. Pasricha's EPIC Lab: Yaswanth Raparti, Sai Vineel Reddy Chittamuru, Ishan Thakkar, Daniel Dauwe, Vipin Kukkala, Yi Xiang, Nishit Kapadia, Ninad Hogade, Febin Sunny, Asif Anwar Baig Mirza, Ayush

Mittal, Prathamesh Kale, Varun Bhatt, Liping Wang, Sai Kiran Koppu, Joydeep Dey, Chris Langlois, Danish Gufran, Abhishek Balasubramaniam, Zemin Tao and Kamil Khan.

I am blessed to have a wonderful family – my father Ashutosh Kumar Tiku, my mother Sarojini Tiku, my brother Samvit Tiku and my extended family – Kanchan Mehra, Cherry Mehra and Atul Mehra. Their generosity, humility and guidance have made me continually strive to be a better person.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF RESEARCH PUBLICATIONS

- **S. Tiku**, S. Pasricha, "Siamese Neural Encoders for Long-Term Indoor Localization with Mobile Devices," IEEE/ACM Design, Automation and Test in Europe (DATE) Conference and Exhibition, 2022.
- L. Wang, **S. Tiku**, S. Pasricha, "CHISEL: Compression-Aware High-Accuracy Embedded Indoor Localization with Deep Learning," IEEE Embedded System Letters, 2021.
- **S. Tiku**, P. Kale, S. Pasricha, "QuickLoc: Adaptive Deep-Learning for Fast Indoor Localization with Mobile Devices," ACM Transactions on Cyber-Physical Systems (TCPS), vol. 5, no. 4, pp. 1-30, 2021.
- **S. Tiku**, S. Pasricha, B. Notaros, Q. Han, "A Hidden Markov Model based Smartphone Heterogeneity Resilient Portable Indoor Localization Framework," Journal of Systems Architecture, vol. 108, 2020.
- **S. Tiku**, S. Pasricha, "PortLoc: A Portable Data-driven Indoor Localization Framework for Smartphones," IEEE Design and Test, vol. 36, no. 5, pp. 18-26, 2019.
- **S. Tiku**, S. Pasricha, "Overcoming Security Vulnerabilities in Deep Learning Based Indoor Localization on Mobile Devices," ACM Transactions on Embedded Computing Systems (TECS), vol. 18, no. 6, pp. 114, 2019.
- **S. Tiku**, S. Pasricha, B. Notaros, Q. Han, "SHERPA: A Lightweight Smartphone Heterogeneity Resilient Portable Indoor Localization Framework," IEEE International Conference on Embedded Software and Systems (ICESS), 2019.

- A. Mittal, **S. Tiku**, S. Pasricha, “Adapting Convolutional Neural Networks for Indoor Localization with Smart Mobile Devices,” ACM Great Lakes Symposium on VLSI (GLSVLSI), 2018. **(Best Paper Award)**
- C. Langlois, **S. Tiku**, S. Pasricha, “Indoor localization with smartphones,” IEEE Consumer Electronics, vol. 6, no. 4, 2017.
- S. Pasricha, J. Doppa, K. Chakrabarty, **S. Tiku**, D. Dauwe, S. Jin, P. Pande, “Data Analytics Enables Energy-Efficiency and Robustness: From Mobile to Manycores, Datacenters, and Networks,” ACM/IEEE International Conference on Hardware/Software Codesign and System Synthesis (CODES+ISSS), 2017.
- **S. Tiku**, S. Pasricha, “Energy-Efficient and Robust Middleware Prototyping for Smart Mobile Computing,” IEEE International Symposium on Rapid System Prototyping (RSP), 2017.

1. INTRODUCTION

This chapter outlines the challenges associated with the domain of fingerprinting-based indoor localization and the necessities of addressing these challenges at different levels of abstraction using novel and energy-efficient pattern-matching approaches that enable reliable performance across a diverse set of mobile devices under real-world conditions. This chapter also gives a general overview of the contributions of this dissertation.

1.1. INDOOR LOCALIZATION BACKGROUND

The advent of the Global Positioning System (GPS) reformed the global transportation industry and allowed vehicles to not only localize themselves but also to navigate reliably and in a secure manner across the world at high speeds. Today, indoor localization is an emerging IoT domain with a similar purpose and is poised to reinvent the way we navigate within buildings and subterranean locales, with many benefits, e.g., directing emergency response services after a 911 call to a precise location inside a building, accurate tracking of equipment and inventory in hospitals, factories, and warehouses, etc. While GPS is the de-facto solution for outdoor positioning with a clear sky view, there is no prevailing technology for GPS-deprived areas, including dense city centers, urban canyons, and inside buildings and other covered structures, where GPS signals are severely attenuated or totally blocked, and affected by multipath interference. Thus, very different solutions are needed to support localization in indoor locales.

Towards this goal, a few decades worth of academic research has been performed in the direction of enabling indoor localization and navigation services [1]. In recent times, indoor location-based services are also experiencing an upsurge in interest from the industry [2].

Technology giants such as Apple [3] and Google [4] are focusing on enabling standardization of indoor localization technology. Apple in 2019, included Ultra-Wide-Band (UWB) radio transceivers in an attempt to avail spatial-awareness services to their users [5]. The inclusion of UWB in Apple devices has enabled iPhone users to locate nearby items such as keychains. In 2021, Apple announced their indoor maps program that utilizes WiFi fingerprinting (see section 1.1.5) to locate and navigate users in an indoor environment [3]. Google has been working with authorities at airports, malls, stadiums, and other public indoor environments to extend Google Maps to the indoor environment. As a part of this program, the authorized owner of a building shares the floorplan blueprint with Google Maps. The blueprint is then integrated into the Maps application, such that zooming into a building of the Maps UI presents the user with detailed information of floorplan and associated indoor landmarks [6]. However, this feature only allows the user to view the indoor map and the actual localization of the user is still carried out by the GPS, if available, which delivers poor accuracy. In recent years, Google has introduced a new IEEE WiFi standard (IEEE 802.11 MC), that would enable smartphones to localize themselves using the Time of Flight (see section 1.1.2) based approach known as Round Trip Time (RTT) [4]. The widespread adoption of this standard would enable Google to seamlessly extend their Maps application on smartphones, used by millions of people every month, to the indoor environment.

Given the ubiquitous presence of smartphones within the populous, its growing computational capabilities (section 1.2.4), and its rich suite of sensors and radios, smartphones have become the platform of choice for the purpose of indoor localization. Both Apple and Google have focused on enabling indoor localization through their smartphone platforms to deliver indoor localization services.

On the other hand, networking hardware companies such as Aruba have heavily invested in iBeacon (Bluetooth) and other wireless networking hardware to deliver indoor localization services for enterprise networks [7]. Retail outlets such as Target [8] now allow shoppers to locate and navigate themselves within shopping isles. Further, given the relatively inexpensive nature of infrastructure options available for enabling indoor location-based services, several small-scale business ventures are attempting to monopolize on the opportunity by enabling indoor localization services in public settings such as schools [9], airports [10], shopping malls [8], and so on [11]. To further highlight the central role of smartphones for indoor localization within the community of academia and industry, we note that all localization platforms discussed in the subsection so-far utilize various localization methods on smartphones to either compute and/or deliver localization information to the users.

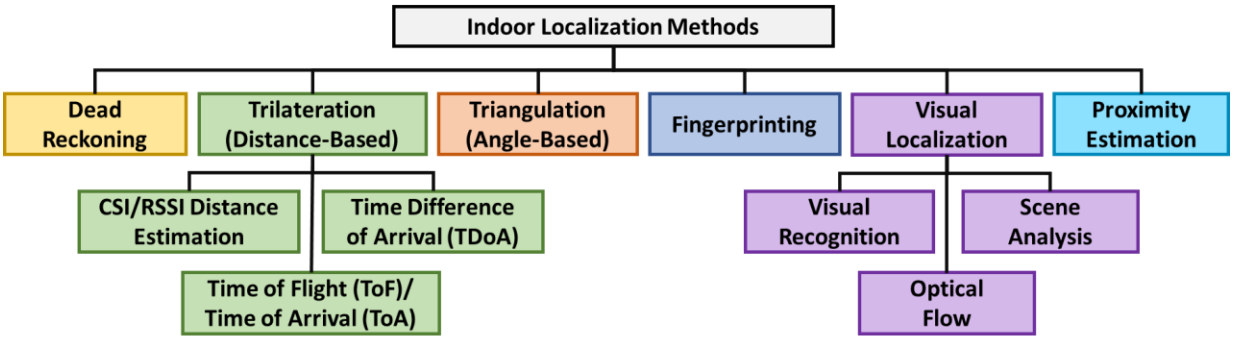


Figure 1. Taxonomy of indoor localization methods.

A hierarchical description of the taxonomy of all known indoor localization methods is captured in Figure 1. Each indoor localization technique described in Figure 1 requires methodology specific radios and sensing equipment. Employing a specific kind of sensing scheme such as radio signals for indoor localization introduces domain specific challenges. Some common challenges associated with utilizing radio signals arise from the interactions of these signals with

environmental objects and artifacts. The radio signal that is being observed at a specific location could have traveled directly from the source to the receiver, also called Line of Sight (LoS) communication. Alternatively, a signal may have reflected off walls and other environmental objects to reach the receiver following multiple paths (Multipath), also called Non-Line of Sight (NLOS) communication. Another commonly known aspect of radio signals traveling in the indoor environment is shadowing, where certain locations observe no or degraded reception as the signals get blocked by a nearby object (see section 1.1.5). These environmental interactions can severely impact on the quality of the indoor localization methodology being used.

To overcome methodology specific limitations and challenges, a realistic consumer-ready indoor localization framework can consist of a hybrid combination of these methodologies that utilize a diverse set of sensors and radios. In the following sub-sections, we discuss the various indoor localization methods, associated sensors and radios, and their known adoptions in the industry [9] [12].

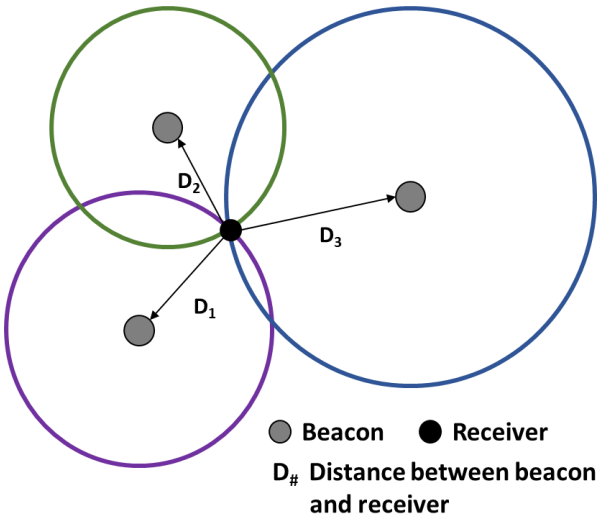


Figure 2. A graphical representation of the trilateration indoor localization process.

1.1.1. DISTANCE-BASED TRILATERATION

Indoor localization using trilateration is carried out using a series of distance estimations between the smartphone and external beacons. Figure 2 depicts the process of localizing a receiver in a two-dimensional space. The location of the receiver is in reference to the location of the beacons. With distance measurements at a minimum of three unique known locations, a two-dimensional position relative to the beacons can be established. The critical differentiating aspect between distance-based trilateration, Time Difference of Arrival (TDoA) and Time of Flight or Time of Arrival (ToF/ToA) techniques are the methodologies used to capture the distance between the beacon and the receiver. Given the ubiquity of smartphones, they are often chosen as the receivers. This is because contemporary smartphones consist of a wide variety of sensors and radios thereby enabling flexibility in the choice of the specific type of beacon that could be deployed such as WiFi, Bluetooth and/or UWB. Utilizing pre-deployed WiFi Access Points (APs) can drive down the deployment costs associated with the indoor localization infrastructure.

The simplest form of distance estimation employed for trilateration is using the signal power and distance relationship on a given floorplan. The measure of signal power can be indirectly established through Received Signal Strength Indicator (RSSI) or more recently Channel State Information (CSI). The measure of the RSSI or CSI values change proportionally to the distance from its origin and this can be used to estimate the distance to a beacon. While this methodology can be applied using any radio beacon, such as Bluetooth, Zigbee, Ultra-Wide-Band (UWB) etc. an early example of research in the RSSI distance estimation using WiFi radios is the EZ localization algorithm [13], whereas the work in [14] utilizes CSI to passively evaluate the distance between the CSI beacon and the receiver. But such estimations are often error prone because of interference and multipath effects (NLOS environments). It also requires the distance model to be

established and maintained for each radio beacon independently. Given the limitations and very high maintenance requirements associated with this approach, RSSI/CSI-distance estimation-based trilateration has not been observed to be widely adopted in the real-world.

In order to overcome the challenge of building independent RSSI-distance models for each radio source (such as a WiFi AP), researchers proposed the Time of Flight (ToF) or Time of Arrival (ToA) ranging approach. It works on the idea of measuring the time it takes for a signal to travel from a source to a receiver or vice versa and then using the time taken to estimate the distance between various (minimum three for localization on a 2D floorplan) pairings of radio sources to the target smartphone's user [15].

Multilateration is an extension of ToA/ToF based methodology. This approach involves a signal sent from a mobile point, which is received by two or more fixed points. The difference in time at which each of the fixed points receives the signal corresponds to the difference in distance between the mobile point and each of the fixed points [16] [17]. An alternate method is to have each of the fixed points send out a signal simultaneously and to calculate the position based on the difference in time at which these signals are received by the mobile point. These strategies can be used to find the location of the mobile point in relation to the fixed points. One constraint of this approach is that the fixed points require a method for precise time synchronization. Smartphones do not contain radios that are designed for multilateration by default, so radio-based methods would require external sensors and/or beacons limiting the ubiquity of this approach. ToA/TDoA-based approaches also require an accurate measurement of time between a signal being transmitted and received. Considering, that these timing measurements need to be carried out synchronously at the transmitter and receiver end, trilateration-based approaches require highly specialized

hardware that is not yet widely available in the off-the-shelf in the off-the-shelf devices available to consumers.

As an alternative to maintaining and synchronizing the clocks on several devices, the Round-Trip-Time (RTT) approach utilizes the time it takes for a signal to travel from the signal source to the receiver and then back to the transmitter. This approach only requires the one of the transmitter-receiver pairs to maintain timing information. The RTT based approach is now supported by newer smartphones and WiFi APs that support the RTT standard are sold by Google. Given that it has been accepted as an IEEE standard, we expect more WiFi AP manufacturers to adapt this RTT standard.

1.1.2. TRIANGULATION

In an effort to reduce the number of transceivers involved in the localization process, triangulation-based indoor localization captures the angles of the user carrying a transceiver at a minimum of two known locations. Figure 3 depicts the process of indoor localization using only two receivers. Given the angle at which the transmitter/beacon is in reference to the receiver enables the localization of the beacon with reference to the known locations of the receivers. This reduces the number of unique pieces of hardware required for triangulation-based methods by one. The only sensor in a typical smartphone that can estimate an angle to a known location is the magnetometer, which is prone to interference. Due to this fact, triangulation-based systems using a smartphone require the addition of other external sensors. Angle-of-arrival (AoA) techniques are often used to determine the angle between an array of receiving antennas and a transmitting source. One technique to measure the arrival angle is similar to the time difference of arrival (TDoA) and is accomplished by measuring the time difference at which the signal arrives at each antenna in

the array to calculate an incident angle to the array. Another method is based on spacing the antennas in the array a known wavelength apart and measuring the phase difference of the received signal between each of the antennas to calculate an incident angle [18]. An external antenna array would be required to measure AoA in smartphones as these devices do not contain such an array.

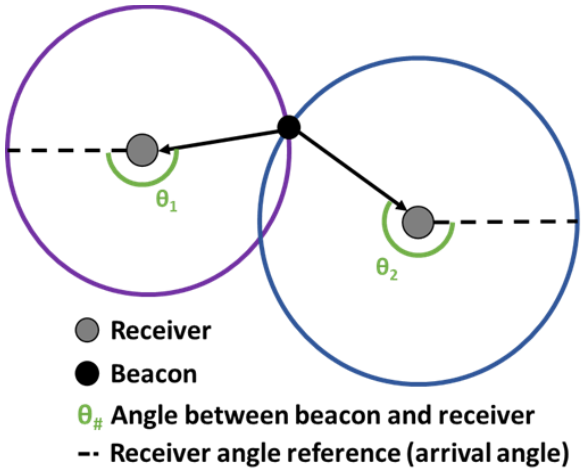


Figure 3: A graphical representation of triangulation-based indoor localization.

It is important to note that both Triangulation and Trilateration-based techniques are adversely affected by interactions with indoor objects and artifacts leading to multipath signals and shadowing.

1.1.3. DEAD RECKONING

Dead reckoning refers to the class of techniques in which sensor data is used along with the previously known position to determine the current position. The most commonly used strategy in this area is known as pedometer-based dead reckoning. This strategy works by first detecting and then counting steps and using this data with stride length information to estimate distance traveled. Figure 4 shows a simplistic strategy for step detection in FootPath (indoor navigation) [19]. Steps

can be detected if there is a difference in acceleration p on the low-pass filter in the vertical direction in a given time window.

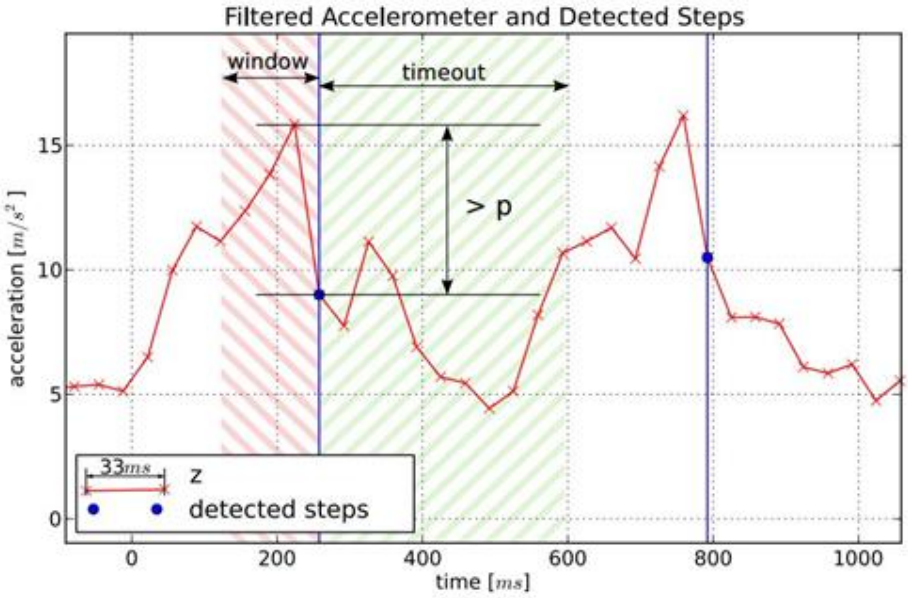


Figure 4. Detection of steps using accelerometer data on a smartphone [19].

In [20], stride length is modeled to have a linear relationship with step frequency, whereas [21] models the motion of the pelvis as an inverted pendulum to approximate stride length. A heading (direction) estimation is achieved with magnetometers and horizontal acceleration data. The step count, along with stride length and heading estimate combine to form a movement vector. This movement vector can be applied to a previous location to approximate the current location. The motion sensors found in smartphones (the accelerometer, gyroscope, and magnetometer) are capable of high sampling and update rates and allow such pedometer dead reckoning [22], [23] [24]. The pedometer-based approach has its challenges, e.g., distance calculations can accumulate errors because of an imperfect stride length estimation or irregular walking pattern. This approach is also ineffective for alternate means of transportation that do not require a step motion such as wheelchairs, moving walkways, and subway trains, among others.

Given that dead reckoning first assumes that the initial location is known, it is often used in conjunction with other indoor localization techniques formulate a larger framework that can be deployed in the real world. Additionally, all forms of dead reckoning approaches discussed in this sub-section accumulate error over time. Researchers have proposed utilizing map matching [25] and particle filters [26] [27] in attempts to limit the error accumulation.

1.1.4. VISUAL LOCALIZATION

One or more of the smartphone's cameras can be used as input data sources for localization through a variety of methods. A key requirement is that the camera must be exposed and unobstructed for these localization strategies to be effective.

The camera on a smartphone can be used for recognition of visual cues in the environment (visual recognition). A company called ByteLight uses different coded pulses in overhead LED lighting within a building that can be picked up by a smartphone camera to indicate that the device is located within a certain section of that building [28].

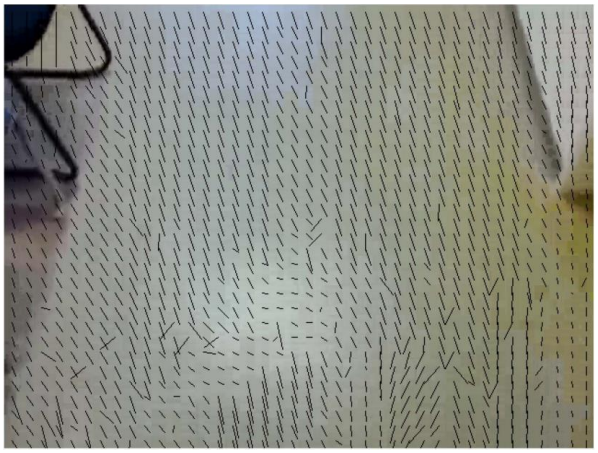


Figure 5. Estimation of motion vectors captured using a smartphone camera pointed towards the floor [29].

Camera information can also be important for detecting motion and rotation. A process known as optical flow measures the distance at which points of interest move. If the distance between the camera and various points of interest is known, the distance traveled can be extrapolated. Optical flow is commonly used for indoor flying drones by using a camera pointed at the ground to estimate change in location and speed [30]. As shown in Figure 5, smartphone cameras have also been used to capture the optical flow of the room floor for direction and velocity estimation [29]. But floors that lack visual features or are reflective lead to reduced accuracy.

As compared to other indoor localization methodologies discussed so far, visual localization presents itself as the most human like methodology. However, most works are highly sensitive indoor contextual changes, such as furniture movement, people blocking camera view, and camera perspective. Additionally, smartphone camera based indoor localization technologies targeted towards human are also restrictive in terms of the carry-mode of the smartphone. For example, a user might wish to place the smartphone in their pocket instead of pointing it to the localization environment and listen to step-by-step voice navigation, which is not possible using smartphone camera-based indoor localization. Certain privacy concerns might also arise, as the user of the camera-based indoor localization application may inadvertently capture and report the faces, presence, and actions of other humans in their environment.

In the following sections, we discuss fingerprinting based indoor localization, which attempts to alleviate the limitations and challenges associated with many of the indoor localization techniques discussed so far.

1.1.5. FINGERPRINTING-BASED INDOOR LOCALIZATION

Popular solutions for indoor positioning with high accuracy leverage wireless radio signals, such as WiFi, Bluetooth ultra-wideband (UWB), etc. Due to the existing widespread deployment of WiFi Access Points (APs) in most indoor locales, using WiFi for indoor localization can lead to low-cost solutions. Many localization algorithms that utilize these wireless signals have been proposed, e.g., based on proximity, trilateration, triangulation, and fingerprinting. Studies have shown that fingerprinting-based algorithms deliver higher accuracy, without stringent synchronization or line-of-sight requirements and enable greater error resilience in the presence of frequently encountered multipath signal interference effects. An additional advantage to using WiFi fingerprinting-based indoor localization frameworks comes from the ubiquitous nature of smartphones [21] that come packed with a suite of sensors and can execute high-complexity algorithms in real-time. This further trims down the deployment costs associated with fingerprinting-based indoor localization.

However, given the benefits of fingerprinting-based indoor localization over conventional approaches, it comes with its own set of challenges. The following section presents a brief overview of fingerprinting-based indoor localization on smartphones followed by an in-depth discussion of the challenges associated with it.

1.2. OVERVIEW OF FINGERPRINTING-BASED INDOOR LOCALIZATION

As presented in Figure 6, the fingerprinting-based approach for indoor localization conventionally consists of two phases. In an offline phase (Figure 6: left), location-tagged wireless signal signatures, i.e., fingerprints, at known indoor locations or Reference Points (RPs) are

captured and stored in a database. Each fingerprint in the database consists of an RP and wireless signal characteristics, e.g., RSSI, from visible APs at that location.

It is important to note that the observed RSSI for a particular AP, such as AP2 in Figure 6, is an artifact of several unique interactions between the signal and the environment. For example, the RSSI for AP 2 at reference point L2 is observed not only due to the fading of signal power over the distance, but also the signal-attenuation due to the pillar (shadowing), signal scattering over sharp objects and the reception of a multipath signals. As highlighted in the previous section, such factors induce an adverse effect on localization techniques that only rely on the distance-based relationships between the transmitter receiver pair. In contrast, fingerprinting takes into account the environmental signal interactions along with the transmitter receiver distance relationships when localizing a user. For example, the fingerprint captured at L2 has the unique attribute of a degraded RSSI for AP2 (from -50 dB at L1 to -74 dB at L2 in Figure 6) and is a part of the fingerprint database. The database of fingerprints is used to train a Machine Learning (ML) model, such that the RSSI fingerprints are the input features to the model, and the locations are the output features. The ML model is then deployed on the user's smartphone or a cloud like service.

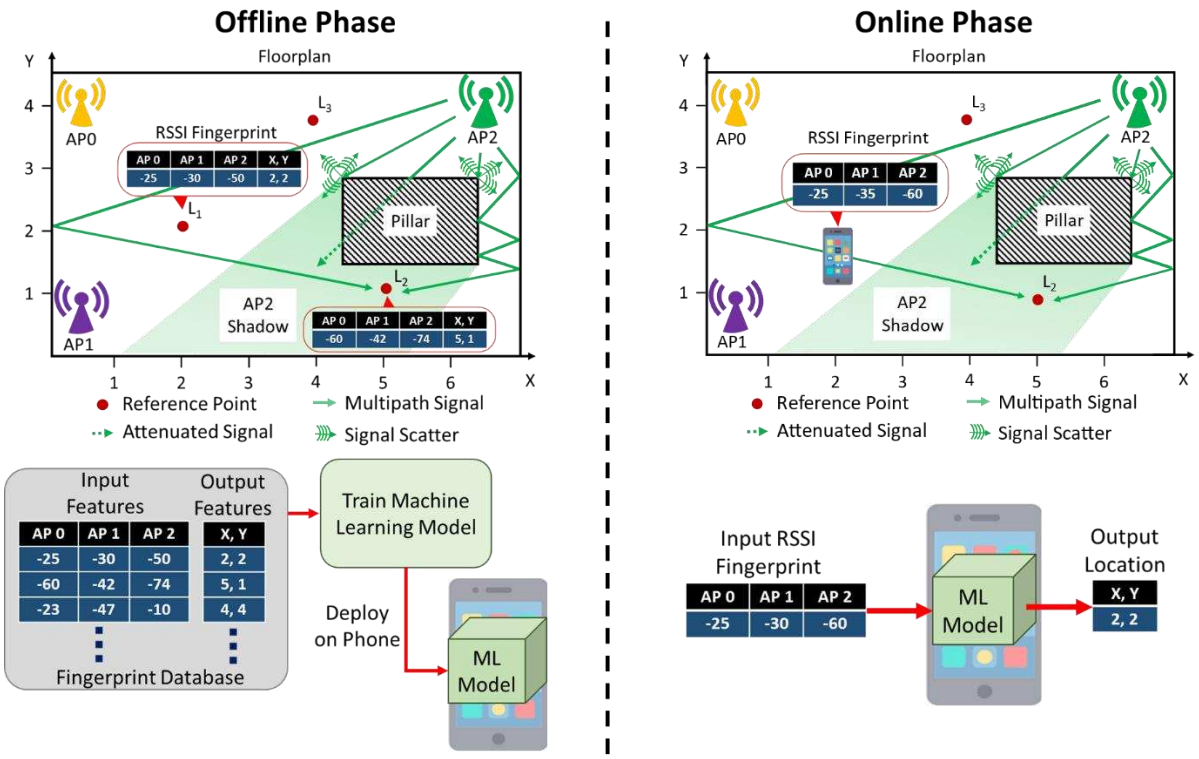


Figure 6. The offline (left) and online (right) phases of a fingerprinting-based indoor localization framework using three WiFi Access Points [31].

In the online phase (Figure 6: right), the observed RSSIs of the visible APs on the user's mobile device is used to query the localization ML model and determine location. It is important to note that the RSSI fingerprint captured in the online phase may not be perfectly identical to the fingerprints observed in the offline phase (RSSI of AP2 changes in Figure 6). The goal of the ML model is to identify the location in the offline phase whose fingerprint is the most similar to the one observed in the online phase.

Such WiFi-based fingerprinting is a promising building block for low-cost indoor localization with mobile devices. Unfortunately, there are many unaddressed challenges before a viable WiFi fingerprinting based solution can be realized: (i) the algorithms used for the matching of fingerprints in the online phase have a major impact on accuracy, however the limited CPU/memory/ battery resources in mobile devices requires careful algorithm design and

deployment that can trade-off accuracy, energy-efficiency and performance (decision latency); (ii) the diversity of mobile devices poses another challenge as smartphones from different vendors may have varying device characteristics leading to different fingerprints being captured at the same location; (iii) security vulnerabilities due to unintentional or intentional WiFi jamming and spoofing attacks can create significant errors which must be overcome; and (iv) short-term and long-term variations in AP power levels and the indoor environments (e.g., adding/moving furniture, equipment, changes in density of people) can also introduce errors during location estimation.

1.2.1. SMARTPHONE HETEROGENEITY

Perceived WiFi RSSI values for a given location captured by different smartphones can vary significantly. Figure 7 shows the impact of smartphone heterogeneity on the mean RSSI (vertical-axis) and its standard deviation (shaded region) for various WiFi APs (horizontal-axis) at a given location using smartphones noted as LG V20 (LG), BLU Vivo 8 (BLU) and OnePlus 3 (OP3). Figure 7 (top) shows how the captured RSSI values (y-axis) of APs visible at the same location (x-axis) are different on the LG smartphone and the OP3 smartphone. Figure 7 (bottom) shows even worse variations on the mean and standard deviation of the captured RSSI for the LG and BLU smartphones. These variations are a function of device specific characteristics such as WiFi chipset, antenna sensitivity etc. and create errors in fingerprinting-based localization.

For the realization of fingerprinting-based indoor localization across heterogenous platforms, there is a critical need for designing and developing frameworks that are resilient to such variations in RSSI.

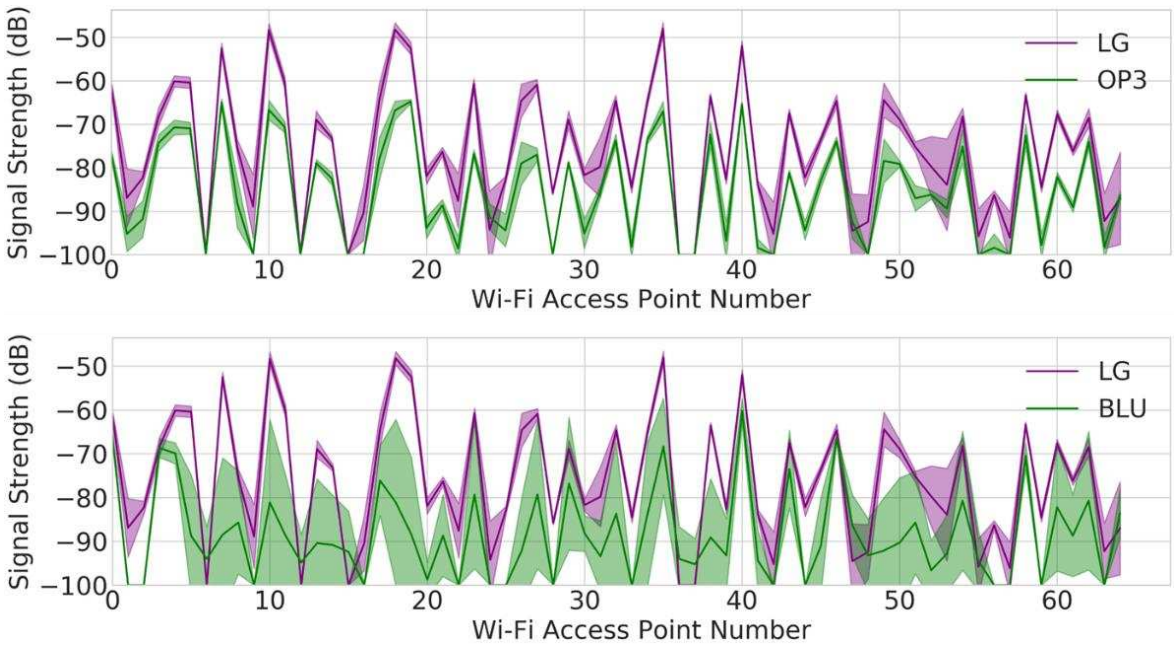


Figure 7. Impact of device heterogeneity on the mean and standard deviation of WiFi signal strength (RSSI) for WiFi access points at the same location in a building across the pairs of smartphones: LG V20 vs OnePlus3 (top) and LG V20 vs BLU Vivo 8 (bottom) [32].

1.2.2. TEMPORAL VARIATION IN FINGERPRINTING

An emerging challenge for fingerprinting-based indoor localization (especially WiFi-based) arises from the fluctuations that occur over time in the RSSI values of APs. Such temporal-variations in RSSI can arise from the combination of a myriad of environmental factors, such as human movement, radio interference, changes in furniture or equipment placement, etc. This issue is further intensified in cases where WiFi APs are removed or replaced by net-work administrators, causing the underlying fingerprints across the floorplan to change considerably. This leads to catastrophic loss in localization accuracy over time.

A naïve approach to overcome such a challenge would be to re-capture fingerprints across RPs once the framework tends to lose its localization accuracy and re-train the machine learning model. However, capturing fingerprints across the floorplan is an expensive and time-consuming endeavor. In an effort to reduce the costs associated with capturing fingerprints, researchers have

also proposed crowdsourcing-based approaches. Unfortunately, given the inconsistent temporal variations, device heterogeneity and human error from capturing crowdsourced fingerprints such approaches tend to deliver limited resilience.

1.2.3. SECURITY VULNERABILITIES IN FINGERPRINTING

Given the rising public adoption of indoor localization, researchers have raised concerns regarding the privacy and security of fingerprinting-based frameworks. Some of these security vulnerabilities are discussed here.

1.2.3.1. USER LOCATION PRIVACY

Recent works in the domain of fingerprinting-based indoor localization propose the use and deployment of resource intensive machine learning models that require large amounts of memory and computational capabilities [33] [34] [35]. Considering these frameworks need to be deployed on smartphone-like embedded platforms that may not meet such resource requirements, researchers propose deploying the models on cloud-based platforms or similar remote services. Such as approach compromises the privacy of the user as their location data may be intentionally or unintentionally shared with malicious third parties. To meet the security challenge, researchers now promote energy-efficient models that can be deployed on the smartphones themselves.

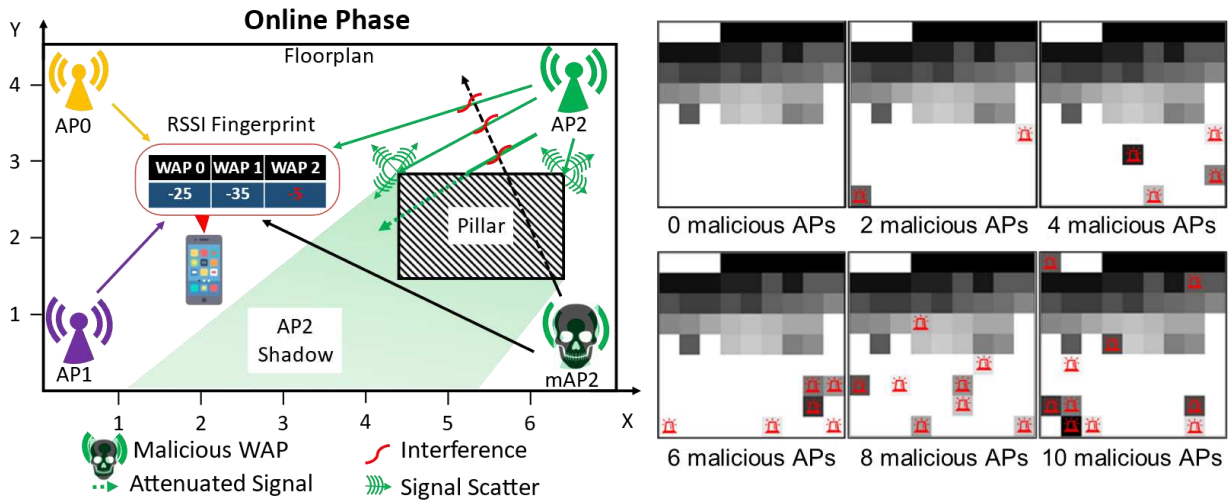


Figure 8. The representation of a WiFi spoofing and jamming attack on a floorplan (left) on only three APs and possible impact on WiFi fingerprints (right). Each WiFi fingerprint (right) is represented as a single channel black and white image, with pixel intensities indicating RSSI strength. Each fingerprint image on the right is designed to capture RSSI for 81 APs (9 × 9). A pixel with a red marker indicates a maliciously altered WiFi RSSI [31] using one or more malicious APs.

1.2.3.2. ACCESS POINT JAMMING OR INTERFERENCE

Fingerprinting-based indoor localization relies on the observed signals of APs. A malicious third party could place signal jammers (narrow band interference) in the vicinity. Such a jammer would both be able to create signal inference with trusted APs (non-malicious APs) thereby manipulating the observed signal strength of the user and be able to completely block the trusted AP from transmitting, leading to the user's smartphone to lose visibility of the AP. Such a scenario is depicted in Figure 8, where the malicious WiFi Access Point (mAP2) interferes with the signals from the trusted AP2 to manipulate the observed signal strength at the mobile device, leading to the alteration of inputs to the localization model.

1.2.3.3. MALICIOUS ACCESS POINTS OR SPOOFING

Spoofing is the mode of attack where the malicious third-party places transmitters in the vicinity of the indoor locale such that the transmitter broadcasts packets with the Media Access

Control (MAC) addresses of other legitimate and trusted APs. The MAC address could be obtained by a person walking within the target indoor locale or be dynamically captured by the malicious transceiver. A single transmitter may also be able to spoof packets of many APs with a variable output of power. This would enable the malicious AP to create a dynamic attack pattern that would be hard to detect and avoid. As presented in Figure 8, a combination of jamming and inference enables the malicious mAP2 to modify the RSSI for AP2 at the mobile device. A single malicious mAP could spoof packets of multiple APs (up to 10 shown in right of Figure 8), leading to degraded localization quality.

1.2.4. ENERGY LIMITATIONS OF SMARTPHONES

While the computational capabilities of smartphones have grown exponentially over the previous decade, such battery powered devices are heavily budgeted by their energy capacity. Figure 9 presents the growth in the technical specifications of the Apple iPhone since its inception in 2007. We observe that the specifications that are directly associated with computational capabilities i.e., CPU speed, RAM, and storage exhibit an exponential growth ranging between 30-60× over a period of 14 years. In comparison, battery capacity is observed to have grown by a meager 4×. Such a trend indicates that while we have gained the ability to execute high-complexity memory intensive workloads through energy-efficient SoCs (System-on-Chips) and heavily optimized software, the duration of time a smartphone is likely to last before it needs to be charged again (battery life) remains limited.

Such a challenge prompts researchers in the domain of indoor localization to design and deploy frameworks that take into consideration the energy requirements of several components

such as for motion (accelerometer, magnetometer etc.), wireless technologies (WiFi, Bluetooth etc.), cameras, and localization algorithms (neural networks, K-Nearest-Neighbor etc.).

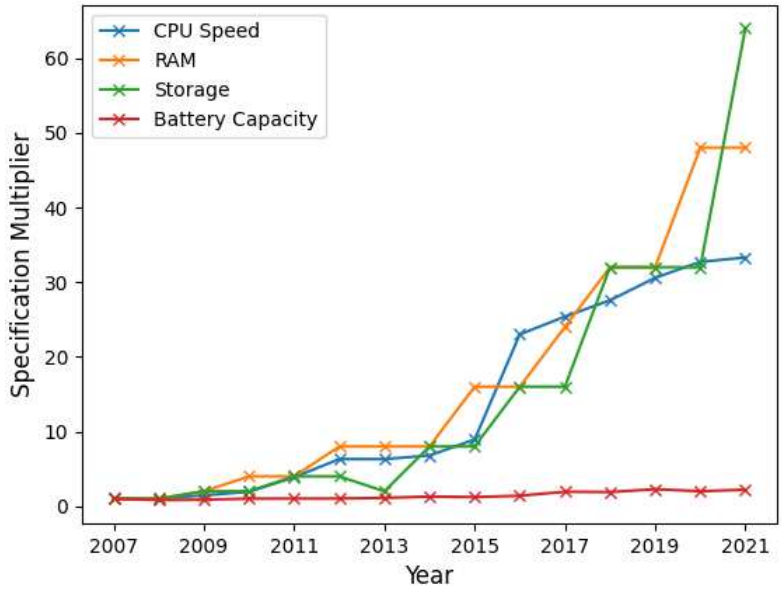


Figure 9. Trends in the technical specifications of the Apple iPhone from 2007 to 2021. The iPhone Pro, Max and SE categories are not considered. The CPU speed is computed as the summation of the number of cores times the maximum clock speed.

1.3. DISSERTATION OVERVIEW

In summary, there is a crucial need for a holistic fingerprinting-based indoor localization framework that can work on smartphones and overcome the aforementioned challenges in an energy-efficient and reliable manner. Such a framework is not easy to conceptualize because of the interdependencies that arise while attempting to address these challenges simultaneously. For example, enabling security against spoofing and jamming attacks, resilience to temporal variations and device heterogeneity may require increasing the complexity of machine learning models deployed on the smartphone, however, it may adversely affect the battery life of the device and also the responsiveness of the overall localization framework.

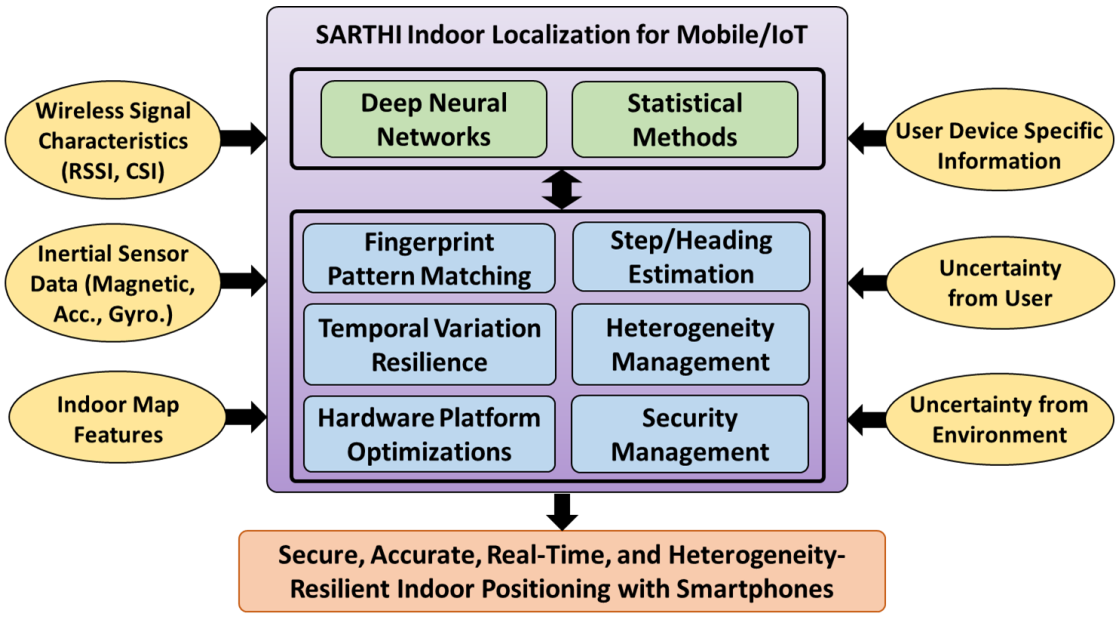


Figure 10. Overview of the proposed SARTHI indoor localization framework with specific working sub-components (blue) and input features and considerations (yellow).

To address these issues, we propose a real-time indoor localization framework (*SARTHI*) that utilizes deep learning and statistical methods to address the abovementioned challenges in a holistic manner. Figure 10 shows a high-level overview of the *SARTHI* framework with our published contributions describing various aspects of this framework. *SARTHI* is a multi-faceted fingerprinting-based indoor localization framework that combines distinct novel innovations targeted towards performance enhancement and domain-specific challenges namely fingerprint pattern matching using deep-leaning, temporal variation resistance, heterogeneity and security management approaches that can be fused with classical dead-reckoning based motion estimation techniques.

The design and deployment of our proposed framework *SARTHI* considers several real-world limitations and information resources such as user device information and characteristics, uncertainty in user movement and environment, indoor map features and intelligent use of inertial sensor data. Further details as captured in this dissertation are organized as follows:

In Chapter 2, we analyze the root cause of the device heterogeneity problem and its impact on fingerprinting-based indoor localization. We conduct an in-depth analysis of fingerprinted data to highlight the importance of using data driven pattern matching approaches for heterogeneous device-based indoor localization. Based on this analysis, we identified computationally inexpensive metrics. These metrics are then employed in the proposed light-weight indoor localization framework *PortLoc*, designed to deliver consistent localization accuracy when ported to heterogeneous smartphones. *PortLoc* was benchmarked against the state-of-the-art approaches using a suite of fingerprints collected using multiple heterogenous smartphones from various vendors across a diverse set of environmental conditions.

In Chapter 3, we extended and applied our observations from PortLoc to overcome its shortcomings and move towards the realization of a robust indoor localization framework SHERPA-HMM. In this work, we formulated the indoor localization problem as a Hidden Markov Model (HMM) that utilizes heterogeneity resilient metrics for reliable user path prediction. SHERPA-HMM is designed towards portability across heterogenous smartphones. It employs a lightweight software-based approach to combine unreliable noisy fingerprints over distinct smartphones and pattern matching/filtering to achieve superior localization accuracy.

In Chapter 4, we first formulate fingerprinting-based indoor localization as a classification problem such that each location or RP on the floorplan is represented as a class or label. We then adapted Convolutional Neural Networks (CNNs) to create a novel deep-learning-based indoor localization framework CNNLOC. To achieve this, we first proposed a new approach to transform RSSI fingerprints into images, which are then used to train a CNN model centered towards improving the localization robustness and accuracy. A hierarchical architecture is then implemented to scale the CNN across real-world buildings with many floors, rooms, and corridors.

Through extensive experimental evaluations we conclude that CNNLOC outperformed traditionally proposed machine learning approaches (such as K-Nearest-Neighbor and Support Vector Machines) and deep-learning-based models in terms of accuracy.

In Chapter 5, we identified and modeled various AP-based attacks that impact the localization accuracy of deep-learning-based indoor localization frameworks, such as the frameworks. For the first time, we conducted an in-depth experimental analysis on the impact of AP-based attacks on CNN and feed-forward DNN (Deep Neural Network) based indoor localization frameworks across indoor paths. Towards overcoming security risks associated with spoofing and jamming attacks on indoor localization-based localization platforms, we propose a novel methodology for constructing AP attack resilient deep learning models to create a secure version of the CNNLOC framework (which we call S-CNNLOC) for robust and secure indoor localization. S-CNNLOC is compared against the performance of CNNLOC for a varying number of malicious AP nodes, and across a diverse set of indoor paths. To further highlight the generalizability of our approach, we evaluated its effectiveness on a conventional feed forward DNN based indoor localization framework.

In Chapter 6, we analyze the impact of CNN model depth on an indoor localization framework in terms of the achievable prediction latency, localization accuracy, and smartphone battery life (location inference energy). For the first time, we adapt and explore the paradigm of conditional computing in the context of deep learning based indoor localization frameworks. The goal of the proposed framework to achieve a balance between several competing aspects of an indoor localization framework such as prediction latency, memory footprint, inference energy and the model complexity. We propose a novel localization framework that can dynamically adapt to the accuracy and latency needs of the target mobile platform at run-time. We compare the

performance of our proposed technique against state-of-the-art deep learning based indoor localization framework over a diverse set of target mobile devices and indoor environments.

In Chapter 7, we propose STONE, a framework that delivers stable and long-term indoor localization with mobile devices, without any re-training. We perform an in-depth analysis on how indoor localization accuracy can vary across different levels of temporal granularity (hours, days, months, year). We then adapt a Siamese triplet-loss centric neural encoder for fingerprinting-based indoor localization and propose temporal variation-aware fingerprint augmentation for robust fingerprinting-based indoor localization. The selection of training samples (triplets) is a critical aspect of training of Siamese neural networks and towards this we develop a floorplan-aware triplet selection algorithm that is crucial to the fast convergence and efficacy of our Siamese encoder-based approach. We also explore design tradeoffs and evaluate STONE against the state-of-the-art indoor localization frameworks.

Chapter 8 concludes this dissertation. We summarize our comprehensive body of research in this chapter and make recommendations for future work.

2. PORTLOC: A PORTABLE DATA-DRIVEN INDOOR LOCALIZATION FRAMEWORK FOR SMARTPHONES

The arrival of Global Positioning System (GPS) technology has revolutionized the way we navigate around the world. Today, every smartphone comes with a built-in GPS that is invaluable for outdoor navigation. Indoor localization technology holds a similar potential to disrupt the way we navigate within spaces that are unreachable by GPS, e.g., malls, buildings, and tunnels. Several startups such as IndoorAtlas, Target (Shopkick), and Zebra have already started to provide services that can help customers find products within a store [11] [36].

Unlike GPS for outdoor localization, no standardized solution exists for indoor localization. Therefore, a myriad of techniques have been developed that use various sensors and radio frequencies. Some commonly utilized radio signals are Bluetooth, RFID, UWB (Ultra-Wide Band), and WiFi [2]. Among these, WiFi based indoor localization has been the most widely researched, due to its low setup costs and easy availability. Indeed, WiFi access points are already deployed in most indoor locales and all smartphones support WiFi connectivity.

Despite the advantages of WiFi based indoor localization, there are also some drawbacks. WiFi signals suffer from weak wall penetration, multipath fading, and shadowing effects. These challenges make it difficult to establish a direct mathematical relationship between Received Signal Strength Indicator (RSSI) and distance from WiFi Access Points (WAPs). These issues have served as a motivation to use fingerprinting-based techniques. Fingerprinting is based on the idea that different locations indoors exhibit a unique signature of WAP RSSI values. Due to its independence from the RSSI-distance relationship, fingerprinting overcomes some of the aforementioned drawbacks associated with WiFi based indoor localization.

Fingerprinting is usually carried out in two phases. In the first phase (offline or training phase), the RSSI values for visible WAPs are collected along paths of interest. The resulting database of values may further be used to train models (e.g., machine learning-based) for location estimation. In the second phase (online or testing phase), the models are used to predict the location of a user based on visible WAP RSSIs.

A majority of the literature that utilizes fingerprinting employs the same smartphone for (offline) data collection and (online) location prediction [37] [38] [39]. This assumes that in a real-world setting, users would have access to the same smartphone as the one utilized in the offline phase. Today's diverse smartphone market, consisting of various brands and models, largely invalidates such an assumption. In reality, the smartphone user base is a distribution of heterogeneous mobile devices that vary in antenna gain, WiFi chipset, antenna shape, OS version, etc.

Recent works have shown that the perceived RSSI values for a given location captured by different smartphones can vary significantly [40]. This variation degrades the localization accuracy achieved through conventional fingerprinting. Therefore, there is a need for portable, device heterogeneity-aware fingerprinting techniques.

2.1. MOTIVATION AND CONTRIBUTION

In this chapter, we present a robust, lightweight, data-driven WiFi RSSI-based fingerprinting framework (PortLoc) that is portable across heterogeneous mobile devices with minimal accuracy loss. The main contributions of our work are:

- we conduct an in-depth analysis of fingerprinted data to highlight the importance of using data-driven pattern matching approaches for heterogeneous device-based indoor localization;

- we identify computationally inexpensive metrics that can be used to compare fingerprint features;
- we design the PortLoc framework for truly portable WiFi fingerprinting-based indoor localization;
- we create a set of benchmarks by collecting fingerprints with multiple heterogeneous devices across buildings, for testing the performance of PortLoc against state-of-the-art localization techniques.

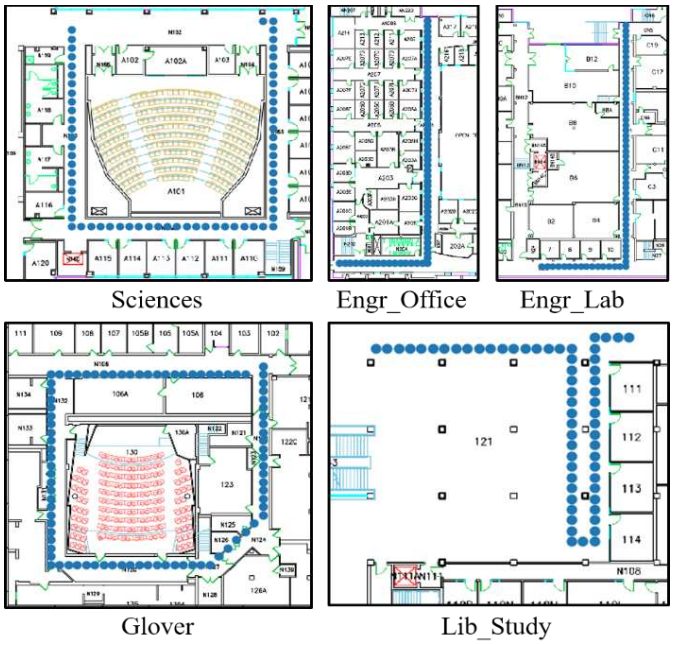
2.2. RELATED WORK

Addressing the challenges associated with WiFi fingerprinting-based indoor localization. Recent work on improving WiFi fingerprinting exploits the increasing computational capabilities of smartphones. For instance, more sophisticated Convolutional Neural Networks (CNN) and ensemble learning are being used in smartphones to improve indoor localization accuracy [38] [39] [41]. One of the concerns with utilizing such techniques are the severe energy limitations on mobile devices. Pasricha et al. [37] proposed an energy efficient fingerprinting-based technique. However, all prior work, including [37], is plagued by the same major drawback, i.e., the lack of device heterogeneity across the offline and online phases. This drawback leads to localization solutions that are untested for real-world scenarios.

In general, devices used by localization solution providers to collect WiFi fingerprints across locations in the offline phase are different from the devices owned by the users in the online phase. Some of the known factors that introduce device heterogeneity include different WiFi antennas, smartphone design materials, hardware drivers, and the OS. Techniques to overcome this issue fall into two major categories: calibration-based methods and calibration-free methods.

The simplest calibration-based approach for heterogeneous device calibration is to acquire RSSI values and location data manually for each new device [42], which is however not very practical. Once RSSI information is collected, manual calibration can be performed through transformations such as weighted-least square optimizations and time-space sampling [43]. These techniques can be aided by crowdsourcing schemes. However, such approaches suffer from accuracy degradation [44].

In calibration-free fingerprinting, the fingerprinting data is translated into a standardized form that is portable across devices. One such approach, known as Hyperbolic Location fingerprint (HLF) [45] uses the ratios of individual WAP RSSI values to form the fingerprint. Unfortunately, HLF significantly increases the dimensionality of the training data in the offline phase. The Signal Strength Difference (SSD) approach [46] reduces the dimensionality by taking only independent pairs of WAPs into consideration. But this approach causes accuracy deterioration. Improvement in accuracy through Procrustes-based shape analysis and uniform scaling of RSSI values was proposed in [40]. The RSSI values are standardized through a Signal Tendency Index (STI), while maintaining the dimensionality of the training data. The STI based technique was shown to perform better than SSD and HLF. Since STI is used in conjunction with Weighted Extreme Learning Machines (WELMs) for best performance, it is a computationally expensive technique. Also, the overall experiments are performed with a highly limited set heterogeneous smartphones, in a one-room-environment that is heavily controlled by the authors. In contrast, our PortLoc framework is a mobile friendly computationally inexpensive approach that is tested over a wide range of environments and heterogeneous mobile devices under realistic settings



(a)

Smartphone	Chipset	Android
OnePlus 3 (OP3)	Snapdragon 820	8.0
LG V20 (LG)	Snapdragon 820	7.0
Moto Z2 Force (MOTO)	Snapdragon 835	8.0
Samsung S7 (SS7)	Snapdragon 820	7.0
HTC U11 (HTC)	Snapdragon 635	8.0
BLU Vivo 8 (BLU)	MediaTech Helio P10	7.0

(b)

Figure 11. (a) Benchmark paths for indoor localization, (b) Smartphones used in experiments.

2.3. ANALYSES OF HETEROGENEOUS FINGERPRINTS

In this section, we first present an analysis of the impact of smartphone heterogeneity on a conventional indoor localization technique: Euclidean-based KNN.

To capture the impact of device heterogeneity we observe the performance of the KNN technique to localize six users with six distinct devices (Figure 11(b)) on five benchmark paths (Figure 11(a)). Figure 12 shows the localization accuracy across all smartphones and paths, for four scenarios where the KNN model was trained on four different smartphones. The most

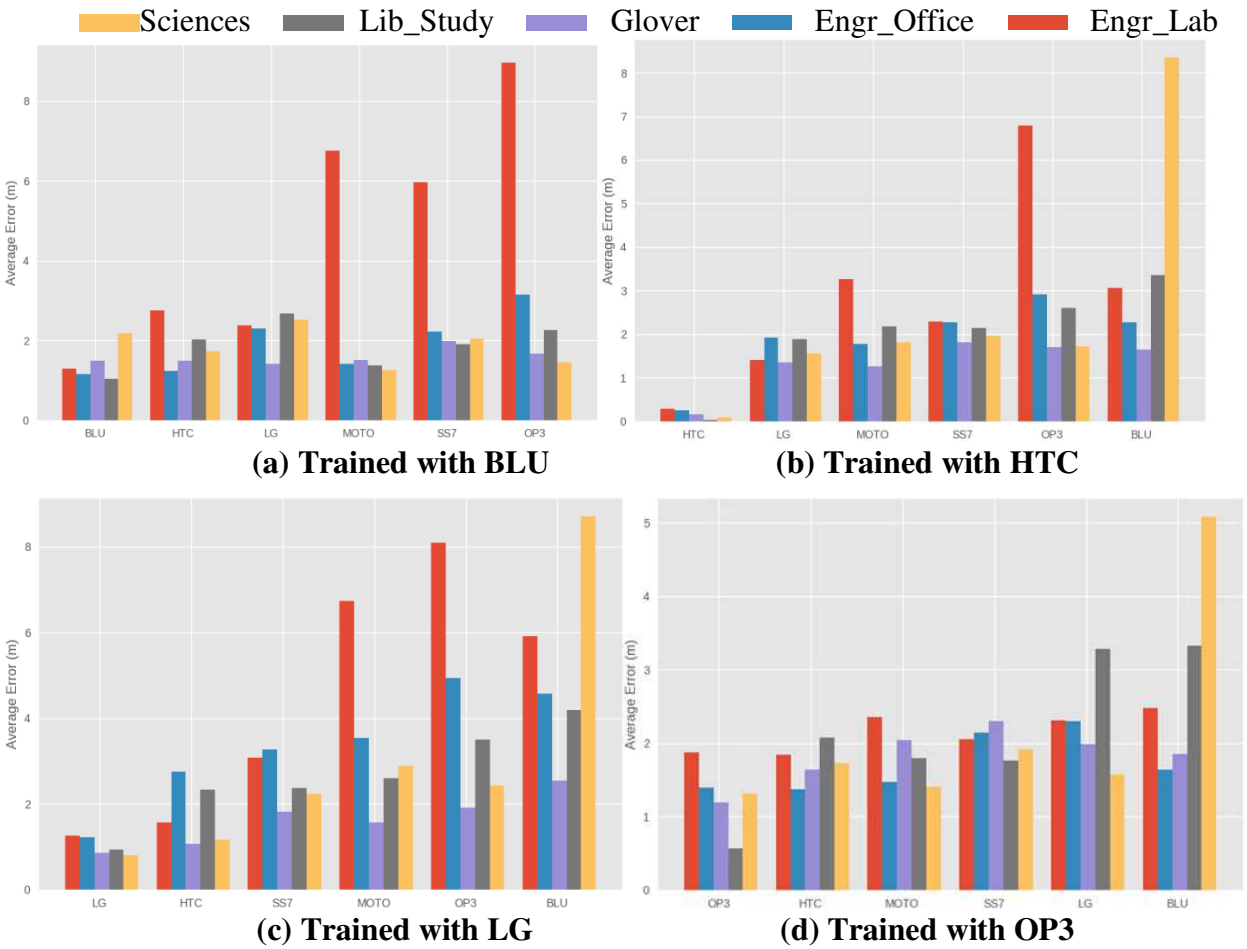
interesting observation is that the best results are achieved when the device under test is identical in the (offline) training and (online) testing phases. For example, the average localization accuracy of KNN remains stable (< 2meters) when trained with OP3 on all paths (Figure 12(d)). But this trend does not hold when the training device is not the same as the testing device. For example, training on the BLU smartphone leads to severe deterioration in accuracy in the Engr_Lab path when testing with the MOTO, SS7, and OP3 smartphones (Figure 12(a)). For the Engr_Lab path in Figure 12(c), we observe that the average error can be 8x between the best-case scenario (LG – LG), and worst-case scenario (LG – OP3). This suggests that a non-portable fingerprinting-based localization framework may be extremely unreliable and unpredictable. However, the degradation due to device heterogeneity is not always observable, and KNN may be able to deliver acceptable results in some cases. Examples of such instances are in Figure 12(b) for the Glover, Engr_Lab and Engr_Office paths. From the results in Figure 12, we set the acceptable limit on average error to two meters and focus only on cases where the average error from KNN is beyond the acceptable error limit.

To better comprehend the cause of degradation in performance due to heterogeneity, we conduct another experiment. As KNN only takes into consideration the raw RSSI strength values of APs, we compare the best performing heterogeneous training-testing pair (LG–HTC) to the worst performing pair (LG–OP3) in terms of observed RSSI as seen on the Engr_Lab path in Figure 12(c). For this experiment, we collected 100 RSSI fingerprints each using the LG, HTC, and OP3 smartphones at the same location on the Engr_Lab path.

The RSSI values for the best and the worst performing training-testing device pairs are presented in Figure 13(a) and Figure 13(b), respectively. The solid lines represent the mean values, whereas the shaded regions represent the standard deviations of RSSI values. From Figure 13(a),

there is a significant overlap in the RSSI values for the LG and HTC devices. This translates to a shorter Euclidian distance and therefore, produces good results using KNN. On the other hand, in Figure 13(b) we observe only slight overlap in the RSSI fingerprints. This gap in overlap leads to the deterioration of localization accuracy for the LG–OP3 device pair.

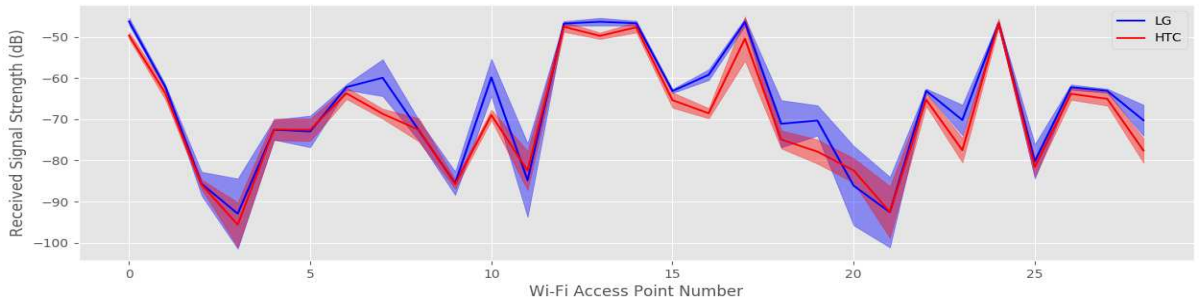
Another observation that can be made from Figure 13 is that the individual RSSI values of both fingerprints grow and drop at the same WAP. Therefore, a metric that captures this pattern of similarity for the two fingerprints should deliver better accuracy for our purposes. This serves as the core motivation for our proposed PortLoc framework, discussed next.



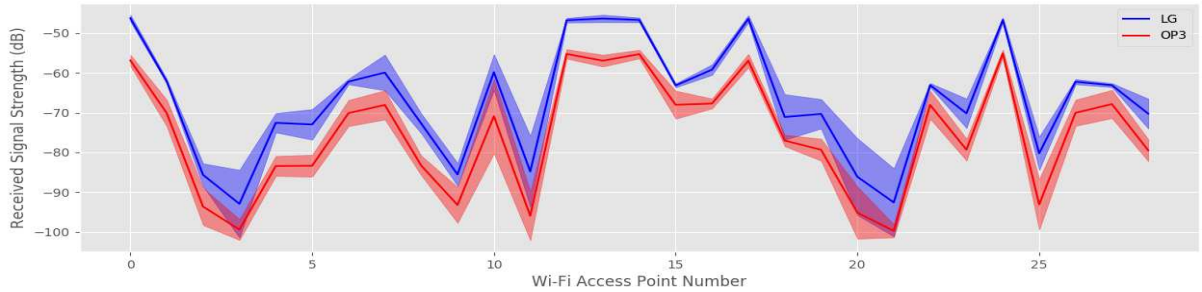
(a) Trained with BLU **(b) Trained with HTC**

(c) Trained with LG **(d) Trained with OP3**

Figure 12. Average Error for various benchmark paths using KNN algorithm.



(a) RSSI values of WAPs for LG–HTC device pair (best–case)



(b) RSSI values of WAPs for LG–OP3 device pair (worst–case)

Figure 13. Average RSSI values of each WAP for training and testing pairs. Shaded regions represent the standard deviation of RSSI.

2.4. PORTLOC FRAMEWORK

In this section, we first discuss the fingerprinting and fingerprint management process required by PortLoc. Then we present two variants of PortLoc based on two pattern matching metrics to enable heterogeneity-resilient indoor localization.

2.4.1. WIFI FINGERPRINTING

We utilize both the 2.4 GHz and 5 GHz WiFi frequencies to capture the RSSI of a WAP along with its Media Access Control (MAC) address and the location (x-y coordinate) at which the sample was taken. The MAC address allows us to uniquely identify a WAP. The RSSI values for WAPs visible at each location are stored in a tabular form with the MAC addresses and the location as table headers, such that each row vector of RSSI values represents a fingerprint for the location in that row. Fingerprints are collected along an indoor path on a smartphone, by the user.

This manual step is essential for any fingerprinting technique. It is important to note that the deliverable accuracy from any fingerprinting-based localization approach is directly correlated to the granularity of sampling along a path. We chose to sample at 1-meter intervals along paths, to achieve a sufficient accuracy of a few meters.

2.4.2. FINGERPRINTING DATABASE PRE-PROCESSING

The captured fingerprints can be easily polluted by the temporarily visible WiFi hotspots or third party owned WiFi APs. Utilizing such RSSI values in our fingerprints can significantly reduce the overall reliability and security of our localization framework. Therefore, we only capture and maintain RSSI values for trusted MAC addresses that are found to be reliable WAP sources. Further analysis of data revealed that WAPs with very low RSSI values (< -90dB) were highly unstable and made it difficult to maintain the shape of the RSSI fingerprint. This led us to filter out all RSSI values that are lower than -90dB. These pre-processing steps help to improve the overall stability of PortLoc

2.4.3. RSSI DATA-AWARE CORRELATION METRICS

To predict the users' location in the online phase of PortLoc, we compute the similarity metrics discussed below, for the fingerprint of the unknown location and the database of known locations. The weighted sum of the locations in the fingerprinting database that produce the greatest value is the new predicted location. The number of similar locations taken into consideration is set to be the square-root of the fingerprinted samples per location taken in the offline phase.

Spearman's Correlation Coefficient (SPRMN): In Figure 13, we observed that individual RSSI values for different smartphones may be further apart, but the RSSI values rise-and-fall together. When two or more variables increase (or decrease) in the same direction, but not always at the same rate, they are known as monotonically dependent variables. SPRMN is a non-parametric test of the monotonic relationship between two variables. SPRMN for a given sample is represented by r_s and by design is constrained as $-1 < r_s < 1$.

If the increase in one variable is followed by a decrease in the other variable, this is called an inverse monotonic relationship and is represented by a negative value. A positive value suggest that the variables increase and decrease together. The magnitude of r_s represents the strength of the positive or negative correlation between the two variables.

Zero Normalized Cross-Correlation (ZNCC): ZNCC is a popular metric in the field of signal processing, single particle analysis, and image matching. It is a measure of similarity between two time-series as a function of displacement. Unlike Spearman's correlation, ZNCC is not bounded within a range, instead it is purely based on the magnitude of the time-series. The higher the magnitude, the stronger the match between the two time-series, for the selected time displacement. For our purposes, we assume each fingerprint to be a time-series and calculate the value of ZNCC for zero displacement.

2.5. EXPERIMENTS

2.5.1. EXPERIMENTAL SETUP

2.5.1.1. HETEROGENEOUS DEVICES AND FINGERPRINTING

To investigate the impact of device heterogeneity, we employed 6 different smartphones (Figure 11(b)). Note that three of the devices have the same chipset. This allows us to explore the impact of device heterogeneity based on chipsets and vendors. We created an Android application

that recorded the x-y coordinate from the user and included a scan button. Once the scan button was pressed, 10 consecutive WiFi scans were conducted with an interval of 1 second. The RSSI value for each WAP and its MAC address was recorded in an SQLite database, and then processed as described in section 1.3.

2.5.1.2. INDOOR PATHS FOR LOCALIZATION BENCHMARKING

We compared the accuracy and stability of PortLoc and frameworks from prior work on five indoor paths in different buildings on our campus. (Figure 11(a); each fingerprinted location is denoted by a blue dot). The path lengths varied between 60 to 80 meters.

Each path was selected due to its salient features that may impact indoor localization. The Glover building is one of the oldest buildings on campus and constructed from wood and concrete. This path is surrounded by a combination of labs that hold heavy metallic equipment as well as large classrooms with open areas. A total of 81 unique WAPs are visible on this path. The Behavioral Sciences (Sciences) and Library (Lib_Study) are relatively new buildings on campus that have a mix of metal and wooden structures with open study areas and bookshelves. We observed 130 and 300 unique WAPs on the Sciences and Lib_Study paths, respectively. The Engr_Office path is on the second floor of the engineering building that is surrounded by small offices and covered by 180 WAPs overall. The Engr_Lab path is in the engineering basement and is surrounded by labs consisting of a sizable amount of electronic and mechanical equipment with about 120 visible WAPs. Both of these paths have large quantities of metal and electronics that lead to noisy WiFi fingerprints and can hinder indoor localization efforts.

2.5.1.3. COMPARISON WITH PREVIOUS WORK

We selected three prior works to compare against PortLoc. The first work (LearnLoc/KNN [47]) is a non-parametric approach based on the idea that similar data when observed as points in a multi-dimensional space would be clustered together. The second work (Rank Based Fingerprinting (RBF) [48]) claims that the rank of WAPs in a vector of ranked WAPs based on RSSI values remains stable across heterogeneous smartphones. Each vector of ranked WAPs represents a point in a Euclidian space, and these points for a given location on a floor map would be very close to each other. The third work combines Procrustes analysis and Weighted Extreme Learning Machines (WELM) [40] to predict the location of a user. Procrustes analysis allows the technique to scale and superimpose the RSSI fingerprints of heterogeneous devices and denote the strength of this superimposition as the Signal Tendency Index (STI). The STI metric is used to transform the original RSSI fingerprints, and then later used to train a WELM model in the online phase with the help of cloud servers.

2.5.2. EXPERIMENTAL RESULTS

2.5.2.1. ACCURACY COMPARISON FOR BENCHMARKING PATHS

Figure 14 shows the localization error across indoor benchmark paths for the two variants of PortLoc (PL_SPRMN, PL_ZNCC) and the prior works (KNN, RBF, STI-WELM).

The first notable observation from Figure 14, is that the RBF technique performs the worst on all paths. The baseline non-heterogeneity aware technique, KNN, significantly outperforms RBF on all benchmark paths. KNN also performs better than STI-WELM and PL_SPRMN in most cases. PL_ZNCC delivers superior accuracy as compared to prior works RBF and STI-WELM. On the Glover path, where we observed the least impact of smartphone heterogeneity, PL_ZNCC closely tracks KNN performance.

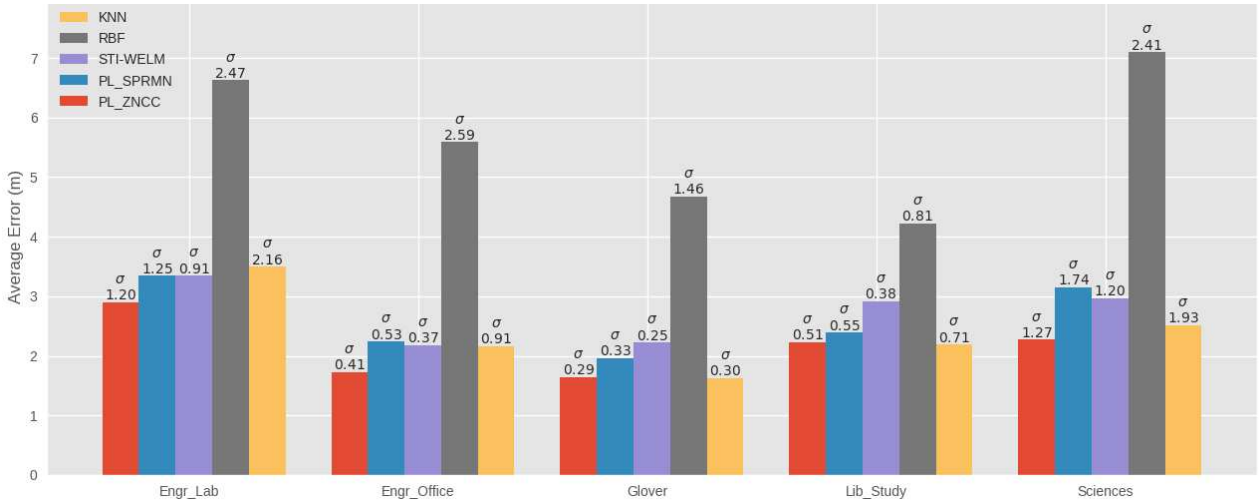


Figure 14. Average error and standard deviation (σ) for indoor benchmark paths and localization frameworks.

Unfortunately, Figure 14 does not compare the performance of localization frameworks on individual devices, and thus misrepresents the stability of KNN and other techniques across paths.

2.5.2.2. DETAILED PERFORMANCE OF LOCALIZATION TECHNIQUES

In the localization experiences of six users carrying smartphones from distinct vendors. The paths along with the training phase device combinations were chosen based on the analysis of the plots in Figure 12. We chose to focus on cases that demonstrated significant deterioration in localization error (above 2 meters) for the non-heterogeneity aware baseline KNN technique.

From Figure 15(a), HTC is the most stable device for KNN, i.e., is least affected by heterogeneity. In all other situations, localization is heavily impacted by heterogeneity. Figure 15(a) is also the only case where RBF performs better than KNN. This suggests that the observed order of strengths of RSSI values for WAPs remain relatively stable in the case of Figure 15(a) as compared to all other plots in Figure 15. Another notable aspect is that this improvement is not maintained when the training device is replaced by HTC in Figure 15(b) for the same benchmark

path. Overall, in Figure 15(a) and (b), PortLoc variants outperform RBF and STI-WELM whenever the localization error from the baseline KNN technique is greater than two meters.

We observe that the RBF technique performs the worst when there is a significant amount of metal in the surrounding environment. This is the case for the engineering building paths (Engr_Lab and Engr_Office) and the path in the Glover building. The perturbations in the WiFi AP RSSI values due to the metallic surroundings cause the ranks of the AP RSSI values to become highly unstable.

From Figure 15, we also observe that the PortLoc variants outperform STI-WELM in most training-testing device pairs. We believe PortLoc is able to deliver superior performance as it is a purely pattern matching based approach. On the other hand, the STI-WELM framework identifies the closest sampled locations from the offline phase using the shape matching based STI metric. The fingerprints of these closest locations are then used to train a WELM based neural network in the online phase itself. This neural network model is not specially designed for pattern matching, and sacrifices predictability of localization error for faster training time in the online phase.

It is interesting to note that under certain situations PL_SPRMN performs worse than STI-WELM, such as on the Glover (Figure 15(c)), Lib_Study (Figure 15(d)) and Sciences (Figure 15 (e)). But in all of these cases PL_ZNCC outperforms PL_SPRMN and STI-WELM. In contrast, the PL_SPRMN technique seems to perform slightly better than PL_ZNCC in some training-testing combinations for the engineering building paths (Figure 15(a), (b), (f)). These observations suggest that there is no clear and obvious winner among the two variants of PortLoc. We also note that for most paths in Figure 15, PortLoc variants, especially PL_ZNCC, perform closest to KNN in the case of non-significant heterogeneity-based accuracy loss. Our work thus strongly motivates the intelligent combination of computationally inexpensive pattern matching based techniques to

enhance the effectiveness of device heterogeneity aware localization frameworks that utilize fingerprinting.

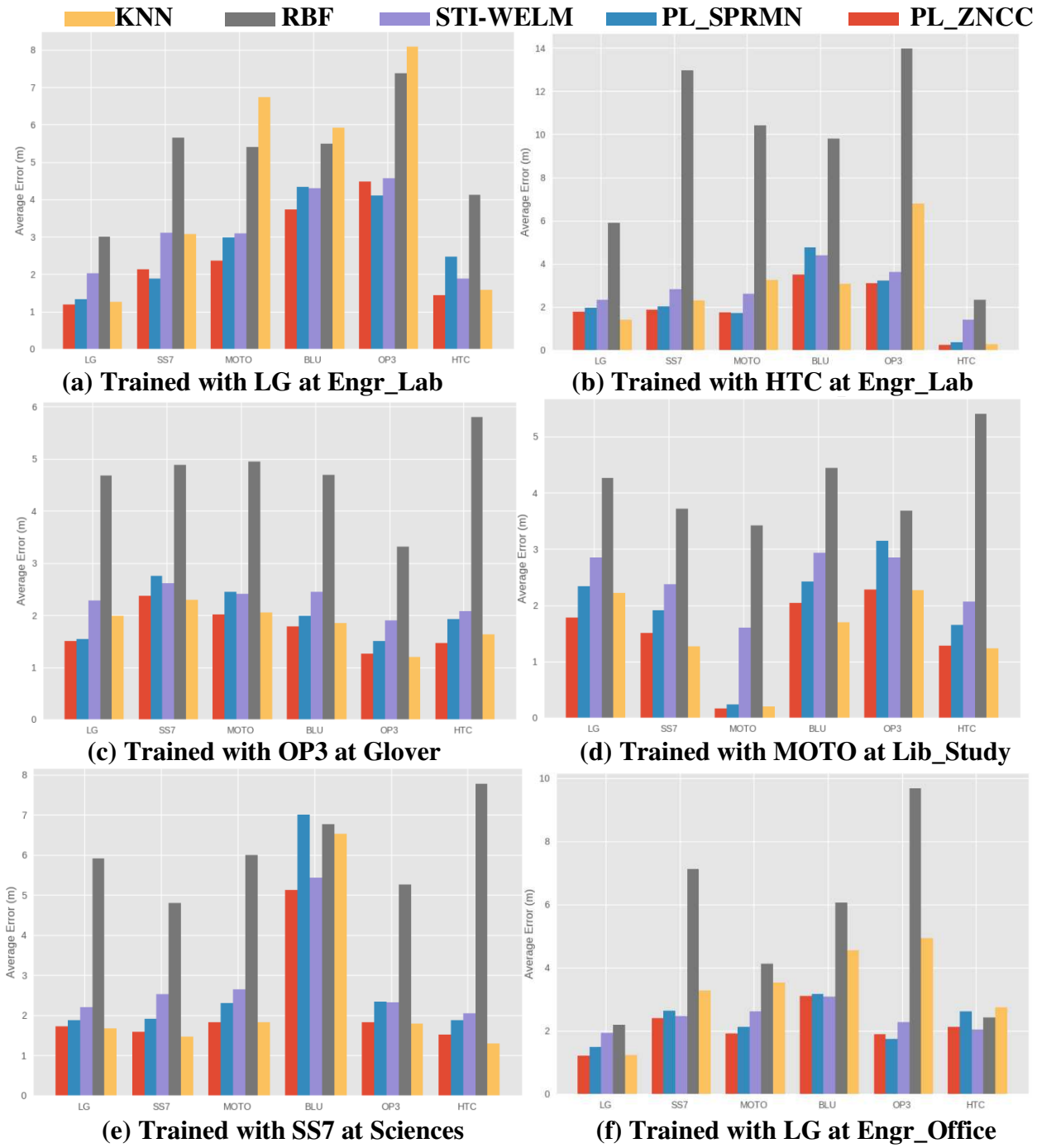


Figure 15. Average Error for various techniques for benchmark paths and training devices.

2.6. CONCLUSION

In this chapter, we have established that the proposed PortLoc framework is a computationally inexpensive solution to the device heterogeneity problem in the fingerprinting-based indoor localization domain. The advantage of establishing portable machine learning models that can be easily ported across devices with minimal loss in localization accuracy is a crucial step towards the actuation of fingerprinting-based localization frameworks in the real world.

3. A HIDDEN MARKOV MODEL BASED SMARTPHONE HETEROGENEITY RESILIENT PORTABLE INDOOR LOCALIZATION FRAMEWORK

The arrival of Global Positioning System (GPS) technology within smartphones has revolutionized the way we navigate in the outdoor world. Today, indoor localization technology holds a similar potential to disrupt the way we navigate within indoor spaces that are unreachable by GPS. An example scenario is localizing patients, staff, and equipment in large hospitals and assisted living facilities. Precise location information can allow first responders closest to a patient to be notified in emergencies. Some startups (e.g., Shopkick, Zebra) are also beginning to provide indoor localization services that can help customers locate products inside a store [11].

Unlike GPS for outdoor localization, no standardized solution exists for indoor localization. Therefore, a myriad of techniques have been developed that use various sensors and radio frequencies. Some commonly utilized radio signals are Bluetooth, ZigBee, and WiFi [2]. Among these, WiFi based indoor localization has been the most widely researched, due to its low setup cost and easy availability. Today, WiFi access points are deployed in most indoor locales around the world and all smartphones support WiFi connectivity.

Despite the advantages of WiFi based indoor localization, there are also some drawbacks. Many prior solutions perform indoor localization by measuring WiFi Received Signal Strength Indicator (RSSI) values and calculating distance from WiFi Access Points (WAPs). These works assume that wireless signal strength reduces in a deterministic manner as a function of distance from a signal source (i.e., WAP). But WiFi signals suffer from weak wall penetration, multipath fading, and shadowing effects in real-world environments, making it difficult to establish a direct mathematical relationship between RSSI and distance from WAPs. These issues have served as a

motivation for using fingerprinting-based techniques. Fingerprinting is based on the idea that each indoor location exhibits a unique signature of WAP RSSI values. Due to its independence from the RSSI-distance relationship, fingerprinting can overcome some of the aforementioned drawbacks with WiFi based indoor localization.

Fingerprinting is usually carried out in two phases. In the first phase (called offline or training phase), the RSSI values for visible WAPs are collected along indoor paths of interest. The resulting database of values may further be used to train models (e.g., machine learning-based) for location estimation. In the second phase (online or testing phase), the models are deployed on smartphones and used to predict the location of the user carrying the smartphone, based on real-time readings of WAP RSSI values on the smartphone.

A majority of the literature that utilizes fingerprinting employs the same smartphone for (offline) data collection and (online) location prediction [34] [38] [39] [47]. This assumes that in a real-world setting, users would have access to the same smartphone as the one used in the offline phase. But today's diverse smartphone market, with various brands and models, largely invalidates such an assumption. In reality, the smartphone user base is a distribution of heterogeneous devices that vary in antenna gain, WiFi chipset, OS version, etc. [49] [50] [51] [52] [53].

Recent work has shown that the perceived WiFi RSSI values for a given location captured by different smartphones can vary significantly [40]. This variation degrades the localization accuracy of conventional fingerprinting. Therefore, there is a need for portable and device heterogeneity-aware fingerprinting techniques. In this chapter, we present a lightweight WiFi RSSI fingerprinting framework for Smartphone Heterogeneity Resilient Portable localization with Hidden Markov Models (SHERPA-HMM) that is portable across smartphones with minimal accuracy loss. The novel contributions of our work are:

- We conduct an in-depth analysis of WiFi fingerprinting across smartphones to emphasize the importance of device heterogeneity-resilient indoor localization;
- We formulate the indoor localization problem as a Hidden Markov Model (HMM) that utilizes heterogeneity resilient metrics for user path prediction;
- We design the SHERPA-HMM framework for portable WiFi fingerprinting-based indoor localization; SHERPA-HMM employs a lightweight software-based approach to combine noisy fingerprints over distinct smartphones and pattern matching/filtering to improve location accuracy;
- We evaluate SHERPA-HMM against state-of-the-art localization techniques, across a variety of Android-based smartphones that are used for indoor localization along paths in real buildings.to the state-of-the-art.

3.1 RELATED WORK

Since the establishment of wireless RF signal based indoor localization a few decades ago, a significant level of advancement has been achieved in this area. In general, most indoor localization techniques fall under three major categories: 1) static propagation model-based, 2) triangulation/trilateration-based, and 3) fingerprinting-based. Early in-door localization solutions used static propagation model-based techniques that relied on the relationship between distance and WiFi RSSI gain [13]. These techniques only work well in open indoor areas as they do not take into consideration any form of multipath effects or shadowing due to walls and other indoor obstacles that invalidate the direct distance-RSSI relationship. This method also required the creation of a gain model for each individual Wireless Access Point (WAP) or WiFi router, which is a cumbersome undertaking. Triangulation/Trilateration-based methods use geometric properties

such as the distance between multiple APs (Trilateration) and the smartphone [54] or the angles at which signals from two or more WAPs are received [55]. Such methodologies may be more resilient to smartphone heterogeneity but are not resilient to multipath and shadowing effects. Some recent work has also investigated multipath effects for triangulation [56], but the proposed approach cannot be implemented on commodity smartphones, and hence has limited scalability.

WiFi fingerprinting-based approaches associate several sampled lo-cations (reference points) with the RSSI measured with respect to multiple WAPs [2] [38] [47]. These techniques are relatively resilient to multi-path reflections and shadowing as the reference point fingerprint captures the characteristics of these effects leading to improved indoor localization. Fingerprinting techniques use some form of machine learning techniques to associate WiFi RSSI captured in the online phase to the ones captured at the reference points in the offline phase. Recent work on improving WiFi fingerprinting exploits the increasing computational capabilities of smartphones. For instance, sophisticated Convolutional Neural Networks (CNNs) have been proposed to improve indoor localization accuracy on smartphones [38]. One of the concerns with utilizing such techniques is the vast amounts of training data required by these models to achieve high accuracy. This is a challenge as the collection of fingerprints for training is an expensive manual endeavor and often the lack of training data leads to poor accuracy.

To overcome this limitation, researchers often resort to building more complex frameworks that utilize hybrid techniques such as combining fingerprinting with dead reckoning [57] [58]. Dead reckoning refers to the use of inertial sensors and a previous known location to predict a future location. However, dead reckoning accumulates errors over time, and needs to be further augmented via map matching to be useful. Map matching utilizes compute intensive particle filtering based approaches along with the knowledge of known physical features on a map to

improve localization accuracy [25] [59]. These systems assume that the location of a user in real time is given by a distribution of particles. The location of every particle is then individually updated at every location prediction cycle and interaction of these particles with known physical features such as walls is also captured. Such methodologies often lead to highly compute intensive solutions. Utilizing such com-plex frameworks levy high energy and computational requirements on resource constrained smartphone platforms, despite their improving capabilities. In [37], an energy-efficient hybrid fingerprinting approach was proposed. However, most prior work, including [37], is plagued by the same drawback, i.e., lack of support for smartphone heterogeneity across both the offline and online phases. This leads to solutions that perform poorly in real-world scenarios.

The most intuitive approach for calibration to address device heterogeneity is to acquire RSSI values and location data manually for each new mobile device [42]. This is unfortunately not very practical. Once RSSI information is collected, manual calibration can be performed through transformations such as weighted-least squares optimizations and time-space sampling [42] [43] [60]. These techniques can be aided by crowdsourcing schemes. However, such approaches still suffer from accuracy degradation across devices [46].

In calibration-free fingerprinting, the fingerprinting data is translated into a standardized form that is portable across devices. One such approach, known as Hyperbolic Location Fingerprint (HLF) [45] uses the ratios of individual WAP RSSI values to form the fingerprint. But HLF significantly increases the dimensionality of the training data in the offline phase. The Signal Strength Difference (SSD) approach [46] reduces dimensionality by taking only independent pairs of WAPs into consideration. Improvement in accuracy over this approach through Procrustes-based shape analysis and uniform scaling of RSSI values was proposed in [40]. The RSSI values

are standardized via a Signal Tendency Index (STI), while maintaining the dimensionality of the training data. The STI-based technique was shown to perform better than SSD and HLF. However, as STI is used in conjunction with Weighted Extreme Learning Machines (WELMs) for best performance, it is very computationally expensive. Also, the experiments in [40] are performed with a limited set of smartphones, in a one-room-environment that is heavily controlled by the authors.

In contrast, our *SHERPA-HMM* framework provides a novel and computationally inexpensive approach that is tested for a wider set of environments and multiple mobile devices in realistic indoor settings.

3.2 HETEROGENEOUS FINGERPRINT ANALYSIS

We begin with an analysis of the impact of smartphone heterogeneity on a state-of-the-art indoor localization technique: Euclidean-based KNN [37]. To capture the impact of device heterogeneity we observe the performance of the KNN technique to localize six users on five benchmark paths (Figure 16) using six distinct devices (Table 1).

Figure 17 shows the boxplots (distribution) for localization error (in the online/testing phase) across all smartphones and indoor paths, for four scenarios where the KNN model was trained on four different smartphones. The most interesting observation is that, in general, the least error is achieved when the device under test is identical in the (offline) training and (online) testing phases. For example, the average localization error of KNN remains stable (< 2 meters) when trained and tested with the OP3 mobile device on all paths (Figure 17(d)). But this trend does not hold when the training device is not the same as the testing device. For example, training on the LG device leads to severe deterioration in accuracy in the Engr_Labs path when testing with the OP3, BLU,

and MOTO smartphones (Figure 17(c)). For the Engr_Labs path in Figure 17(a), the average error can be 6× between the best-case training-testing scenario (BLU–BLU), and worst-case scenario (BLU–OP3). This suggests that a fingerprinting-based indoor localization framework can be extremely unreliable and unpredictable, due to device heterogeneity.

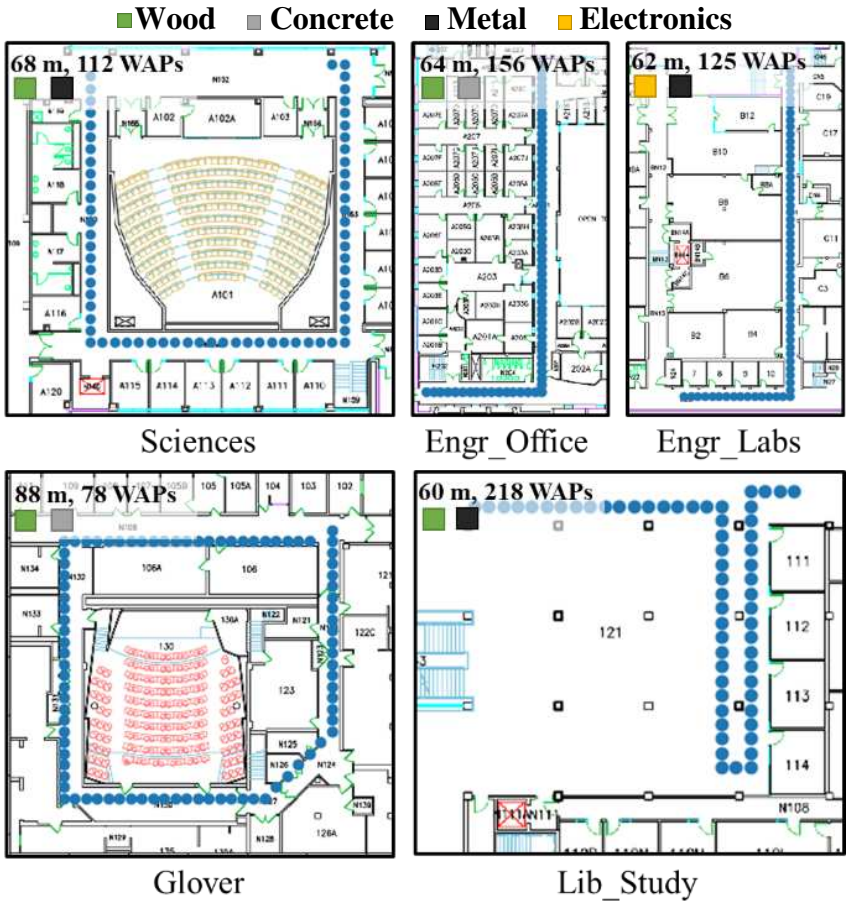


Figure 16. Benchmark paths for indoor localization (with path lengths and WAP density, and salient path features).

Table 1: Details of smartphones used in experiments.

Smartphone	Chipset	Android Version
OnePlus 3 (OP3)	Snapdragon 820	8.0
LG V20 (LG)	Snapdragon 820	7.0
Moto Z2 (MOTO)	Snapdragon 835	8.0
Samsung S7 (SS7)	Snapdragon 820	7.0
HTC U11 (HTC)	Snapdragon 635	8.0
BLU Vivo 8 (BLU)	MediaTech Helio P10	7.0

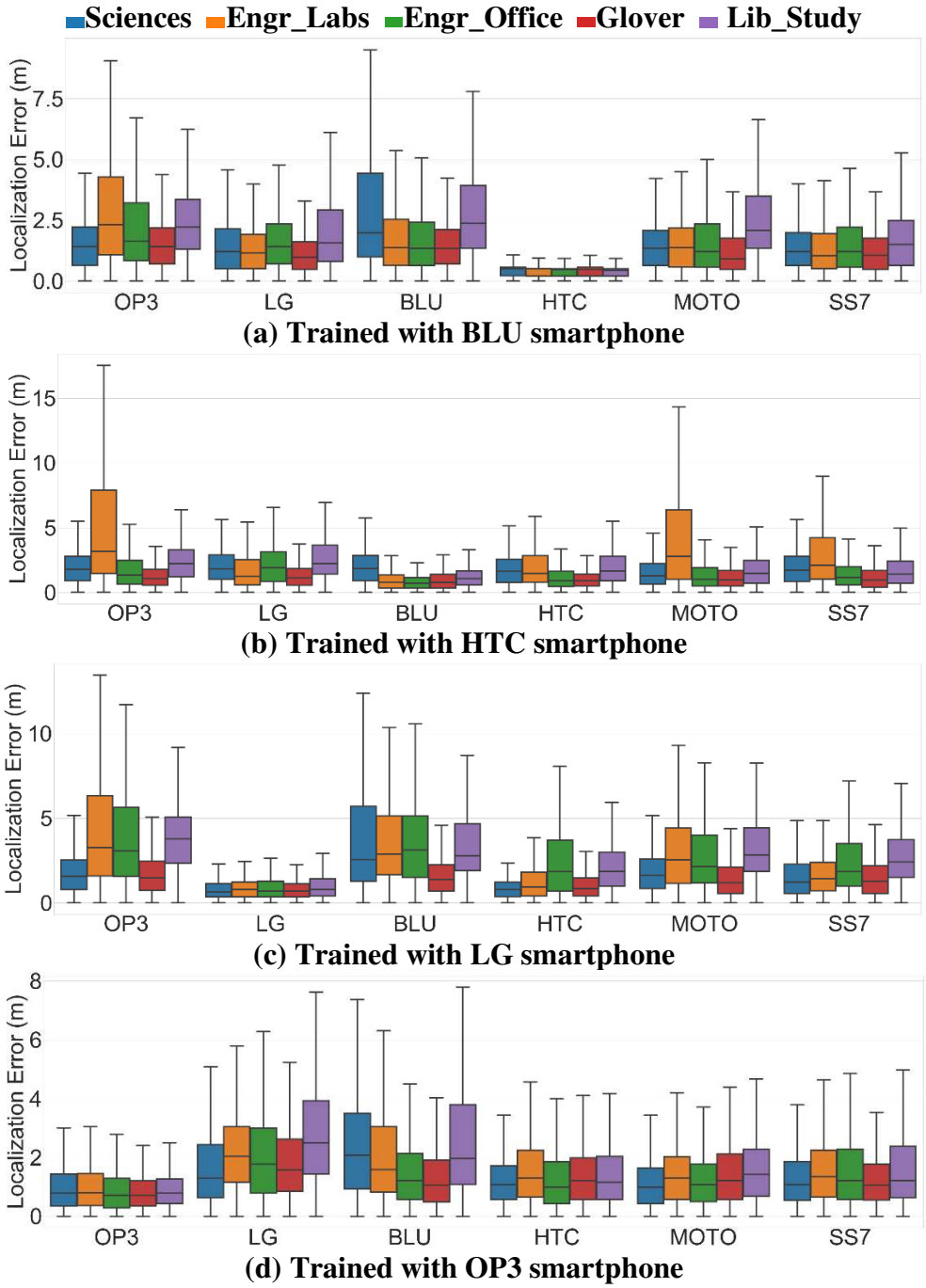
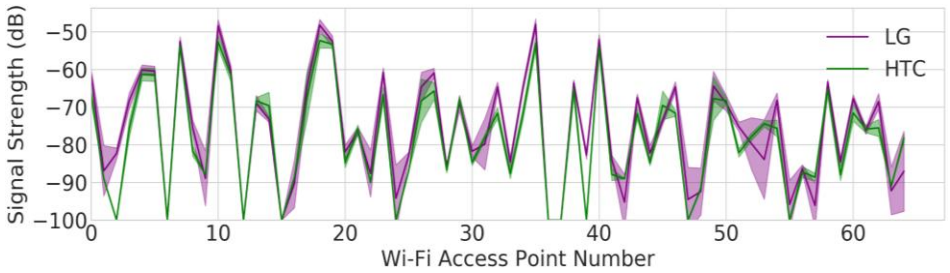


Figure 17. Error distribution for benchmark paths using KNN.

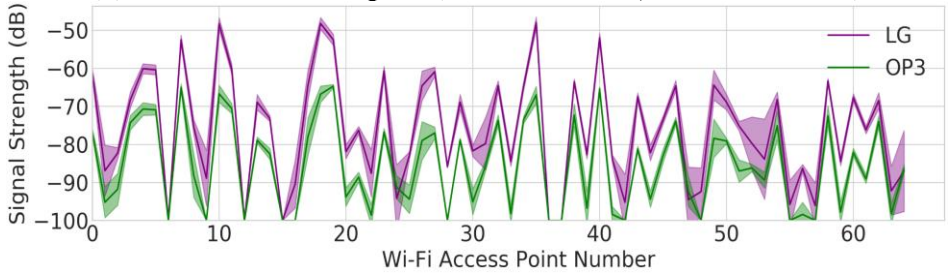
The RSSI values for the best and the two poorly performing training-testing device pairs are shown in Figure 18. The solid lines represent the mean values, whereas the shaded regions represent the standard deviations of RSSI values. From Figure 18(a), it can be observed that there

is a significant overlap in the RSSI values for the LG and HTC devices. This translates into a shorter Euclidian distance and therefore, produces good results using KNN. On the other hand, in Figure 18(b) we observe almost no overlap in the RSSI fingerprints. Instead, an in-consistent gain difference can be observed across the two devices. Further, in Figure 18(c), it can be seen that the BLU device exhibits a significant amount of noise due to variation in the WAP RSSI values for consecutive scans, which can be attributed to its less stable WiFi chipset, compared to the other mobile devices. This leads to severe misprediction when using Euclidian-based KNN. An interesting observation that can be made from looking at Figure 18 is that the overall shape of the fingerprints is similar, including in Figure 18(c), where the shape is similar to the mean fingerprint for the BLU device.

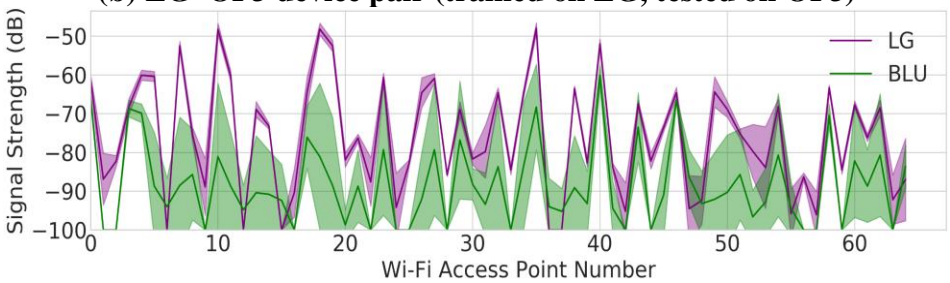
From Figure 18(c), the greater amount of noise from the BLU device is apparent as compared to the other devices, such as the HTC. Identifying and quantifying such noise when using a device for localization (i.e., in the online phase, which is distinct from the offline phase where the localization technique is trained) would allow us to take additional steps to improve localization accuracy. However, it is difficult to identify if a device is capturing noisy fingerprints in the online phase, given a limited set of fingerprints along a path. One approach to quantifying noisy readings could be to check for the Euclidian distance across consecutive scans in the online phase. Since consecutive online scans are conducted using the same device, they should not change significantly over short distances and be similar in terms of Euclidian distance.



(a) LG–HTC device pair (trained on LG, tested on HTC)



(b) LG–OP3 device pair (trained on LG, tested on OP3)



(c) LG–BLU device pair (trained on LG, tested on BLU)

Figure 18. RSSI values of each WAP for training and testing pairs. Shaded regions depict the standard deviation.

To test this hypothesis, we walked over the Engr_Labs indoor path with the BLU (most noisy fingerprints) and HTC (most stable fingerprints) smartphones while capturing WiFi fingerprints with consecutive scans during the walk. Figure 19 depicts the distribution of the Euclidian distance between consecutively captured WiFi fingerprints for the BLU and HTC devices over the Engr_labs path. From Figure 19, we observe that the consecutive scan distances for the HTC device are dis-tributed over a very short range, denoting a stable collection of WiFi fingerprints. However, the distances for the BLU device are distributed over a much wider range due to the variation/noise over consecutive WiFi scans. This approach can be used to identify mobile devices that capture unstable fingerprints during the online phase.

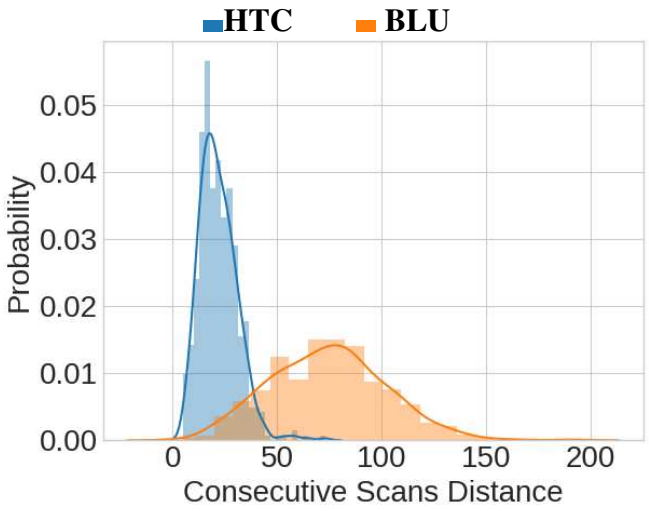


Figure 19. Probability distribution of the Euclidian distance across consecutive pairs of scans using the HTC and BLU smartphones on the Engr_Labs indoor path.

The discussion in this section suggests that a portable methodology that captures the pattern of similarity across fingerprints from heterogeneous smartphones and is able to overcome the noisy behavior of the testing devices, in an energy efficient manner, should deliver better accuracy for indoor localization. These observations serve as the motivation for our proposed SHERPA-HMM framework for lightweight and portable localization, as discussed in Section 3.4. The next section provides a background on HMMs that are used by SHERPA-HMM.

3.3 HIDDEN MARKOV MODEL (HMM) FORMULATION

In this section, we discuss the formulation of the indoor localization process as a Hidden Markov Model (HMM). An HMM statistical prediction model is one that estimates the next hidden state given the transition probability of moving from the current hidden state to the next hidden state and probabilities of observable states [61]. HMMs are particularly renowned for identifying patterns that change with time and have applications in the area of handwriting recognition [62], activity recognition [63], speech synthesis [64], etc. In this chapter, we utilize WiFi RSSI pattern

similarity as observable (non-hidden) states and predict the user's location or path taken by user which are not directly observable (hidden states).

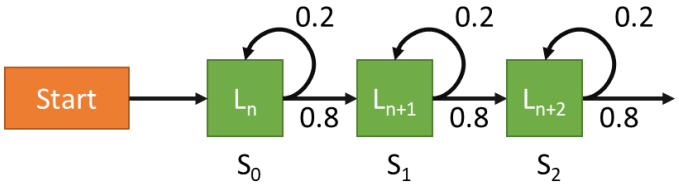


Figure 20. Reference points represented as states in a Hidden Markov Model with given transition probabilities from one state to another.

As shown in Figure 20, we can translate the indoor localization process into a Markov process by first assuming that discrete localizable locations (denoted by Ln, Ln+1, Ln+2…) on the indoor floor plan are the states. As there is no direct way of checking if the predicted position or state in the online phase is correct, these states are referred to as hidden states. Further, for a given path taken by a user in the online phase, there may be certain known probabilities of going from one hidden state to another. From Figure 20, we observe that a user is 80% likely to go to the next state and 20% likely to stay on the same states at any given time-step (Sn). In our case, we assume that a user moving on a path is equally likely to move in all directions by a finite amount.

The probabilities of transitioning from one state to another are also referred as the transition probabilities and are mathematically represented as a matrix. The transition matrix Tr is of size [L × L], where L is number of discrete hidden states (locations in our case). The probability value at Tr [i,j] is the transition probability of going from state i to state j in the next state transition. Additionally, the observable state information is mathematically expressed through the emission matrix E [K × S], where K is the number of observable states and S is the number of subsequent measurements of the observable states. In the context of our work, the observable states are the "WiFi pattern similarity" of a scanned unknown WiFi fingerprint (online RSSI vector) with respect

to the WiFi fingerprints associated with known locations (offline RSSI vectors). The setup of the emission matrix is discussed in the next section.

An HMM utilizes information from the observable states (emission matrix) and known transition probabilities (transition matrix) to identify a the most likely path or series of hidden states. This is achieved through the Viterbi algorithm [61]. The Viterbi algorithm identifies the most likely sequence of hidden states (Viterbi path) given the observed probabilities of observed states. More details on the Viterbi algorithm can be found in [61].

3.4 SHERPA-HMM FRAMEWORK

In this section, we first discuss the WiFi fingerprinting phase and fingerprint pre-processing required by SHERPA-HMM. Section 3.4.1 describes the offline training phase database created in SHERPA-HMM. Section 3.4.4 describes the software-based SHERPA-HMM framework and its main components that are used in the online testing phase: a noise resilient fingerprint sampling, a pattern matching metric, HMM-based location predictor, and additional optimizations.

3.4.1. WIFI FINGERPRINTING

We utilize both the 2.4 GHz and 5 GHz WiFi bands to capture the RSSI of a WAP along with its Media Access Control (MAC) address and the location (x-y coordinate) at which the sample (fingerprint) was taken. The MAC address allows us to uniquely identify a WAP. The average RSSI values for WAPs obtained through multiple scans at each location are stored in a tabular form, such that each row of RSSI values (fingerprint vector) characterizes a unique location. Fingerprints are collected along indoor paths with a smartphone. This step is essential for any fingerprinting technique. Note also that the deliverable accuracy from any fingerprinting-

based approach is correlated to the granularity of sampling along a path. We chose to fingerprint at 1-meter intervals along indoor paths, with the eventual goal of achieving a localization accuracy of within 2 meters.

3.4.2. FINGERPRINT DATABASE PRE-PROCESSING

The captured fingerprints can be easily polluted by temporarily visible untrusted WiFi hotspots. Utilizing such RSSI values in our fingerprints can significantly reduce the overall reliability and security of our localization framework. Therefore, we only capture and maintain RSSI values for trusted MAC addresses that are found to be reliable WAP sources (e.g., by checking for visible WAPs across several days and times-of-day). This pre-processing step helps to improve the overall stability of the SHERPA-HMM framework.

3.4.3. SHERPA-HMM OFFLINE/TRAINING PHASE

In the training phase, a dataset containing the means of all fingerprints taken at each sampled reference point (x-y coordinates shown as blue dots in Figure 16) is established and is stored in a tabular form identical to the fingerprinting dataset. Instead of storing multiple RSSI vector fingerprints for each reference point location, the mean RSSI dataset represents a collection of RSSI vectors where the noise in individual samples has been averaged out. The noise in the training phase dataset is heavily dependent on the smartphone used (as was observed in Figure 18). Therefore, storing the mean of RSSI vectors per reference point is an essential step to ensure the portability of the training database across heterogeneous mobile devices.

3.4.4. SHERPA-HMM ONLINE/TESTING PHASE

3.4.4.1. MOTION-AWARE PREDICTION DEFERRAL

Scanning for WiFi fingerprints is one of the most energy intensive aspects of fingerprinting-based indoor localization frameworks. In the real-world, the user may choose to stop and look at the surroundings while on a path. Any WiFi scans or location prediction cycles that may take place while the user has stopped would be wasted. To avoid such a scenario, SHERPA-HMM tracks the number of steps taken by the user as he or she walks along a path. SHERPA-HMM defers scanning for WiFi fingerprints until it detects that a significant number of steps have been taken since the last location of the user was predicted. Based on the experiments performed in section 3.6, we know that the average localization error over all paths for our framework is close to 2 meters and also the average step length of 0.5 meters can be assumed based on [65]. Therefore, SHERPA-HMM only scans for WiFi fingerprints once the user has taken at least four steps since the last location prediction started.

3.4.4.2. NOISE RESILIENT FINGERPRINT SAMPLING

Noise in the testing phase presents a problem as it leads to degraded localization accuracy. As observed in Figure 18(c), scanned WiFi fingerprints in the testing phase can be significantly impacted by noise. Also, the extent of noise observed varies from device to device. Therefore, the shape of a single offline (training) fingerprint, based on only one WiFi scan, may not match that of the online (testing) fingerprint from a noisy device. To overcome this challenge, we propose a methodology to reduce the impact of observed noise across heterogeneous smartphones and establish a prominent pattern match across the training dataset and the online phase samples.

As previously addressed, the mean RSSI vectors shown in Figure 18 are more reliable for establishing a pattern match across heterogeneous devices instead of individually scanned RSSI

fingerprints. Furthermore, recent advances in smartphone technology have led to the development of robust WiFi support in smartphones. From our preliminary experiments, we found that some smartphones (Table 1) can deliver up to 1 scan in a second. These observations support the idea of executing multiple WiFi scans in the online phase and using their mean for each location prediction.

Our framework opportunistically increases the number of scans required per prediction from 1 to 3 using the approach described in the next section (section 3.4.3). Once multiple consecutive WiFi scans are completed, their mean fingerprint is calculated and used to predict a user's location. The online phase mean fingerprint is compared with the mean fingerprint vectors from the offline database in the next step which uses Pearson's Cross-Correlation (PCC; discussed in section 3.4.4). The location prediction is then made using a lightweight HMM model with PCC-based values embedded in the emission matrix (discussed in section 3.4.4.5).

3.4.4.3. SMART NOISE REDUCTION WITH BOOSTED SCANS PER PREDICTION

The key motivation behind considering multiple WiFi scans per location prediction is to overcome any unpredictable noise across fingerprints from heterogeneous devices. However, too many WiFi scans can undesirably reduce the battery life of a smartphone. To strike a balance between battery life and indoor localization accuracy, SHERPA-HMM identifies situations in the localization process where consecutive fingerprints are noisy and lead to degraded localization performance. In such situations, SHERPA-HMM boosts the number of WiFi scans per prediction from one to up to three scans. To achieve this, SHERPA-HMM keeps a track of two quantities: maximum movable distance (D_{max}) and consecutive scan distance threshold (CSDT).

The maximum distance a user can move within two consecutive predictions is limited. From preliminary analysis and our previous work [66], we found that in the situations where noisy

fingerprints lead to highly erroneous localization predictions, the distance between consecutive predictions is over a threshold of distance a human can move in the allotted time. If the distance between consecutive location predictions is larger than D_{max}, its respective flag is set and SHERPA-HMM resorts to conducting a second scan. The maximum movable distance (D_{max}) threshold is governed by the following equation:

$$D_{max} = (T_{scan} + T_{predict}) \times S_{gait} \quad (1)$$

where T_{scan} and $T_{predict}$ are the times to complete the consecutive WiFi scans and to predict the user's location respectively, and S_{gait} is the average gait speed of the user. In our case, $T_{predict}$ was not significantly variable across smartphones and therefore, an upper bound value for $T_{predict}$ was empirically set to be 0.5 second for the devices shown in Table 1. Also, an upper bound gait speed of 2 m/s was used for S_{gait} based on a large-scale study performed on human gait speeds. A preliminary analysis found that the time taken for 1 WiFi scan (number of default scans) was heavily dependent on the smartphone being employed and even varied for each smartphone itself. Therefore, *SHERPA-HMM* utilizes a timer on the smartphone to record the time taken for consecutive WiFi scans at run-time and uses that value as T_{scan} in equation (1).

The consecutive scan distance threshold (CSDT) is the maximum allowable noise across consecutive scanned fingerprints above which we label the fingerprints as noisy. The value of CSDT is estimated based on the Euclidian distance between the fingerprints collected by the training device at each reference point. The assumption is that if the noise over consecutive scans is low, consecutive WiFi fingerprints captured by the same device should be very close in terms of Euclidian distance. Based on a preliminary analysis performed on the HTC and BLU devices (Figure 19) the value of CSDT was set to 25dB. For our setup with the SHERPA-HMM framework, if the Euclidian distance between the first two consecutive scans is above CSDT, the

noise threshold flag is set, and a third WiFi scan is conducted. The mean of all three WiFi scans is then used to predict the user's location. However, it is important to note that some of the noise resilience comes from the use of HMMs, therefore noise threshold alone may not guaranty degraded localization performance.

If both the noise threshold flag and the distance threshold flags are set, then SHERPA-HMM resorts to conducting three scans per location prediction until at least one of the flags are reset. It is important to note that in contrast to our previous work SHERPA [66] that utilizes three scans per prediction by default, the revised SHERPA-HMM framework only utilizes one scan per prediction by default, and only occasionally boosts up to three scans per prediction. In this manner, our revised framework delivers low-latency predictions in real-time.

3.4.4.4. HETEROGENEITY RESILIENT PATTERN MATCHING: PCC

Pearson's Cross-Correlation (PCC) [67] is measure of linear correlation between two vectors. It is a popular metric in the field of signal processing and pattern matching for voice. A 2D version of PCC is also used in image processing for template matching, a method used for identifying any incidences of a pattern or an object within a template image. PCC between a template vector (T) and a sample vector (X) can be expressed as:

$$PCC = \frac{cov(T, X)}{\sigma_T \, \sigma_X} \tag{2}$$

where, *cov(T, X)* represents the covariance and σ_T and σ_X are their respective standard deviations. *PCC* is limited to a range of -1 to 1, where the sign represents negative or positive linear relationship, respectively, and the magnitude represents the strength of a linear relationship. For our purposes, a positive high value of *PCC* would suggest a strong similarity between the

template (offline database in our case) and the sample (online mean fingerprint in our case). From (2), we observe that *PCC* is directly proportional to covariance (dot product of fingerprints) and inversely proportion to the standard deviation of sample *X* and *T*. Therefore, a sample exhibiting a high level of covariance with the template and a low standard deviation is likely to produce a stronger *PCC*.

3.4.4.5. SHAPE SIMILARITY FOCUSED HIDDEN MARKOV MODEL

As discussed in section 3.3, there are two inputs to a Hidden Markov Model: the transition matrix and the emission matrix. The transition matrix remains the same for a given path, whereas the emission matrix is updated and fed to the Viterbi algorithm in each prediction cycle.

The transition matrix describes the probability of moving from one location (hidden state) to the next. We set up the transition matrix such that a user at a location can move in any direction by two steps in each prediction cycle. For example, on a linear path a user at the location with label l has equal probability to go to the locations with label: $l - 2$, $l - 1$, $l + 1$, $l + 2$ (0.25 each) in the next prediction cycle.

The formulation of the emission matrix is the most critical component of the proposed framework. The emission matrix at any stage of the prediction cycle is given by E [L × S], where L is the number of locations and S is the number of WiFi scans conducted so far. At each location prediction cycle once one or more WiFi scans have been completed (as discussed in section 3.4.3), the PCC for each of the RSSI vectors of training data and the online mean RSSI vector is calculated. These PCC values now form a column vector of length L. The PCC column vector is normalized such that the sum of its values is 1. The normalized PCC column vector is now appended at the end of the emission matrix and fed to the Viterbi algorithm along with the transition matrix. The Viterbi algorithm in turn produces a series of the most likely reference points

or locations (Viterbi path) that the user has visited in the last S prediction cycles. The last location of the series of reference points is the predicted location of the user.

3.4.4.6. OPTIMIZING EMISSION MATRIX FOR PREDICTION TIME

In the real-world, a user may walk a very long path before reaching their final destination. This would result in a very large emission matrix, as each location prediction event will add one new column to the emission matrix. This will improve the overall localization accuracy of the user at each prediction cycle, however, it will also slow down the time it takes to produce a location prediction.

Even though we expect the location prediction of the user to improve as the emission matrix size increases, it may take its toll on battery life and prediction time. Therefore, to maintain the QoS for the SHERPA-HMM framework, we limit the maximum number of columns for the emission matrix to a limit called Scan Memory (Sm). Based on our analysis in section 3.6, we set the Sm to a value of 3. In this manner, the Viterbi algorithm at max predicts the last 3 locations the user has been to, based on the last 3 WiFi scan events. This optimization limits the location inference time in a predictable manner and in-effect optimizes our framework for energy consumption. Thus, the hardware overheads of implementing WBR are low and reasonable.

3.5. EXPERIMENTAL STUDIES

3.5.1. HETEROGENEOUS DEVICES AND FINGERPRINTING

To investigate the impact of smartphone heterogeneity, we employed six different smartphones (shown in Table 1). This allows us to explore the impact of device heterogeneity based on varying chipsets and vendors. We created an Android application that recorded the x-y

coordinate from the user and included a scan button. Once the scan button was pressed, multiple WiFi scans were performed. The RSSI value and MAC address for each WAP were recorded in an SQLite database, and then pre-processed (section 3.4.2).

3.5.2. INDOOR PATHS FOR LOCALIZATION BENCHMARKING

We compared the accuracy and stability of SHERPA-HMM and frameworks from prior work on five indoor paths in different buildings at a University campus. These paths are shown in Figure 16; with each fingerprinted location or reference point denoted by a blue dot. The path lengths varied between 60 to 80 meters, and the number of visible WAPs along these paths varied from 78 to 218. Each path was selected due to its salient features that may impact indoor localization. The Glover building is one of the oldest buildings on campus and constructed from wood and concrete. This path is surrounded by a combination of labs that hold heavy metallic equipment as well as large classrooms with open areas. The Behavioral Sciences (Sciences) and Library (Lib_Study) are relatively new buildings on campus that have a mix of metal and wooden structures with open study areas and bookshelves. The Engr_Office path is on the second floor of the engineering building that is surrounded by small offices. The Engr_Labs path is in the engineering basement and is surrounded by labs consisting a sizable amount of electronic and mechanical equipment. Both engineering paths are in the vicinity of large quantities of metal and electronics that lead to noisy WiFi fingerprints and can hinder indoor localization. A total of 6 users, each carrying a smartphone from a different vendor, walked on each indoor path and collected samples (fingerprints) for each location on that path. This set of data was utilized in the training phase. For the testing/online phase, each of these 6 users walked on each of these paths in a random manner, generating 10 walks each varying from 20 to 50 meters in length.

3.5.3. COMPARISON WITH PRIOR WORK

We selected four prior works to compare against SHERPA-HMM. The first work (LearnLoc/KNN [37]) is a lightweight non-parametric approach based on the idea that similar data when observed as points in a multi-dimensional space would be clustered together. Thus, given a vector of WiFi fingerprints in the testing phase, KNN identifies the K closest fingerprints based on Euclidean distance within its training model and produces the weighted sum of the coordinates of those K fingerprints. The second work (Rank Based Fingerprinting (RBF) [48]) claims that the rank of WAPs in a vector of ranked WAPs based on RSSI values remains stable across heterogeneous devices. It is functionally similar to KNN with the only difference being that each RSSI fingerprint vector in the training and testing phases is sorted and re-populated to store the rank of WAPs instead of raw RSSI values. The third work combines Procrustes analysis and Weighted Extreme Learning Machines (WELM) [40] to predict the location of a user. Procrustes analysis allows the technique to scale and superimpose the RSSI fingerprints of heterogeneous devices and denote the strength of this superimposition as the Signal Tendency Index (STI). The STI metric is used to transform the original RSSI fingerprints, and then used to train a WELM model in the online phase (STI-WELM) with the help of cloud servers. Lastly, we also compare SHERPA-HMM, to our previous work SHERPA [66], that utilizes a Pearson Correlation-based pattern matching metric to identify locations that are associated with offline WiFi fingerprints, and employs lightweight optimizations to deliver high accuracy indoor localization predictions in real-time.

3.6. EXPERIMENTAL RESULTS

3.6.1. SENSITIVITY ANALYSIS ON SCANS PER PREDICTION

To quantify the potential improvement of using mean RSSI vectors in our framework, we conducted a sensitivity analysis to compare the accuracy results for *SHERPA-HMM* using a single RSSI vector and the vectors formed by considering the mean of 1 to 5 scanned fingerprints. Figure 21 depicts the overall localization error for various values of scans per prediction over individual benchmark paths. Even though the overall errors for the *Engr_Office* and *Glover* paths are significantly lower than the other paths (discussed further in section 3.6.3), there is a similar trend in reduction of localization error for all paths as the number of scans per prediction increases. The most significant reduction is observed when moving from 1 to 2 scans per prediction, whereas there is almost no reduction as we move from 4 to 5 scans. This observation solidifies our claim of improvement in accuracy by using more than one scans per prediction, as was discussed in detail in section 3.4.4.2.

It is important to note that scans per prediction not only impacts the localization accuracy but also the energy consumed per prediction. A single WiFi scan can consume a notable amount of energy (~2400mJ when using LG). This motivated us to explore the most suitable value of maximum scans per prediction for *SHERPA-HMM*'s online phase. If the value is too small, such as the case for the *Lib_Study* path in Figure 21 there might not be a significant improvement in localization accuracy. However, if the value is too large, the smartphone may end up consuming a significant amount of energy for an insignificant improvement. From Figure 21, we observe that for most benchmark paths, a majority of the improvement is achieved by conducting only 3 consecutive scans. Therefore, the upper limit on scans per prediction is set to 3 for our framework.

We increase the number of scans per prediction from 1 to 3 in an intelligent manner, as discussed in section 3.4.4.3.

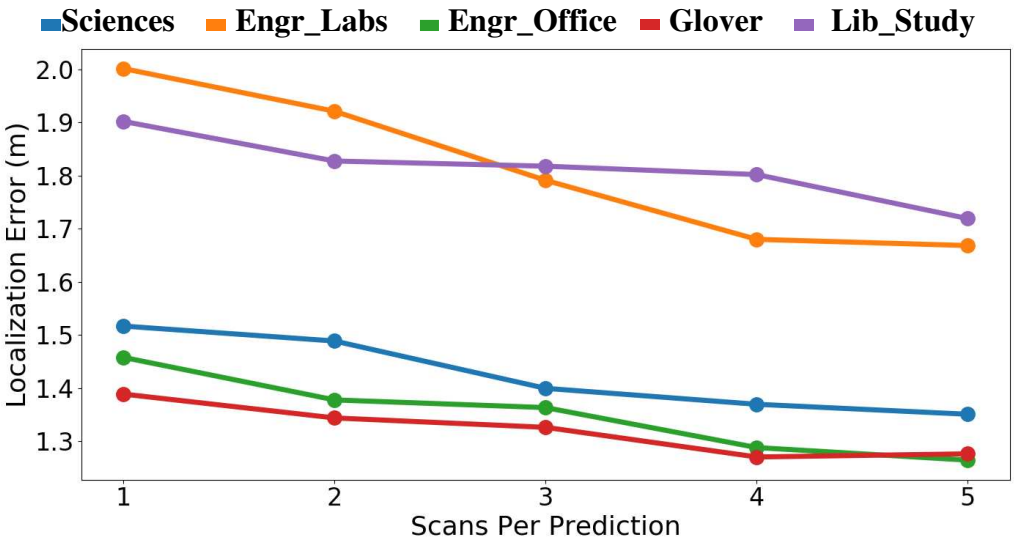


Figure 21. Variation in localization error for different values of scans per prediction (x axis) across various path benchmarks.

3.6.2. SENSITIVITY ANALYSIS ON SCAN MEMORY

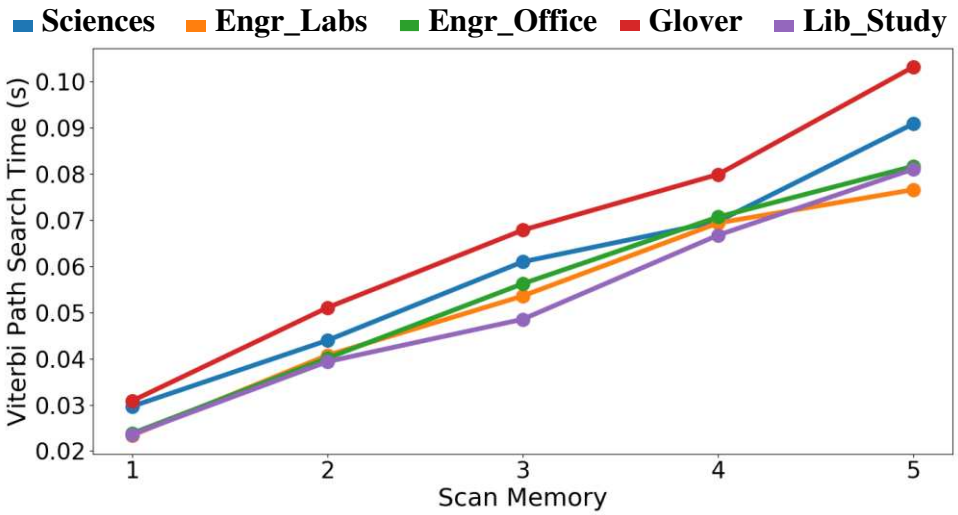
The scan memory variable discussed in section 3.4.4.6 can significantly impact the performance characteristics of the proposed SHER-PA-HMM framework. To quantify this, we perform a sensitivity analysis on the scan memory variable in an effort strike a balance between prediction latency and localization accuracy. Figure 22(a) and (b) present the trends on Viterbi path search times and average localization error across all devices on various paths in our benchmark suite. For this experiment, we analyze the change in Viterbi path search time and localization error when the scan memory (emission matrix width) ranges from 1 to 5. Setting the value of 1 for scan memory translates into only using the latest WiFi scan for location prediction without any historical knowledge, whereas a value of 5 suggests that the latest WiFi scan along

with previous four WiFi scan events were utilized to identify the current location. The results for this experiment were averaged out over all the devices.

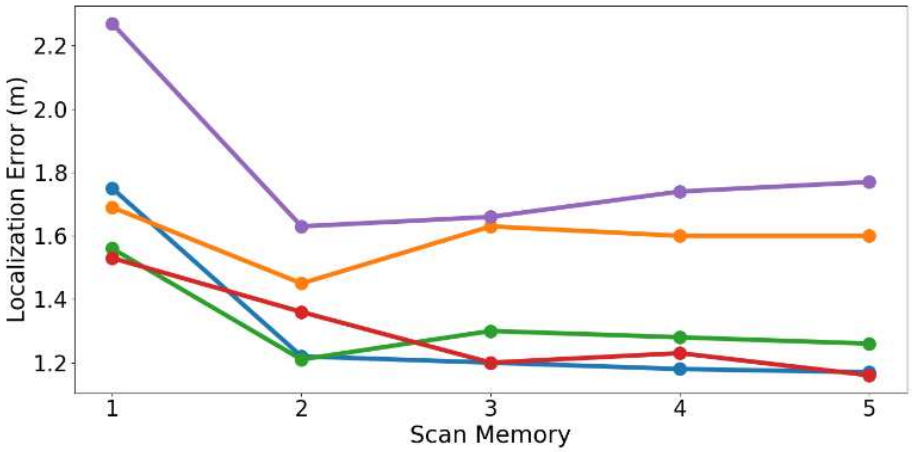
From Figure 22(a), we observe that the time taken by the Viterbi algorithm to deduce the most likely path taken increases linearly as scan memory is increased in the range from 1 to 5. This trend is consistent across the paths. We observe that the overall search time is generally the highest for the Glover path. This is mainly due to the fact that the Glover path is the longest benchmark path with 88 reference locations. Each reference location translates into a unique state in the Hidden Markov model. This increases the number of rows in the emission matrix. In Figure 22(a), we also observe that the search time grows by 5× as scan memory is increased from 1 to 5.

From Figure 22(b), we observe that as we increase scan memory the drop in localization error is most significant up to the point where scan memory is 3, beyond which we observe diminishing returns. Another notable aspect is that the most improvement is observed in the Lib_Study path. This can be attributed to the fact that the Lib_Study has a more complex zig-zag like path. This observation also highlights the prospective improvements that can be gained by using HMM models in more complex paths and dynamically increasing scan memory at run-time in an intelligent manner.

From our observations in Figure 22(a) and Figure 22(b), we set the value of scan memory for our HMM formulation to 3. This allows us to minimize the localization error without significantly impacting the overall prediction time of our proposed indoor localization framework. It is also important to note that the value of scan memory that delivers the best accuracy highly depends on the state space of the path. The user is responsible for identifying a good value of state space for each path individually.



(a) Viterbi path search time w.r.t scan memory



(b) Localization error in meters w.r.t scan memory

Figure 22. Variation in localization error and Viterbi path search time over scan memory for various benchmark paths.

3.6.3. PERFORMANCE OF LOCALIZATION TECHNIQUES

Figure 23 shows the individual plots that represent the contrast in the localization experiences of six users carrying smartphones from dis-tinct vendors. The paths along with the training phase device combinations were chosen based on the analysis of the plots in Figure 17. We focus on a subset of cases that demonstrate significant deterioration in error (> 2 meters) for the KNN technique.

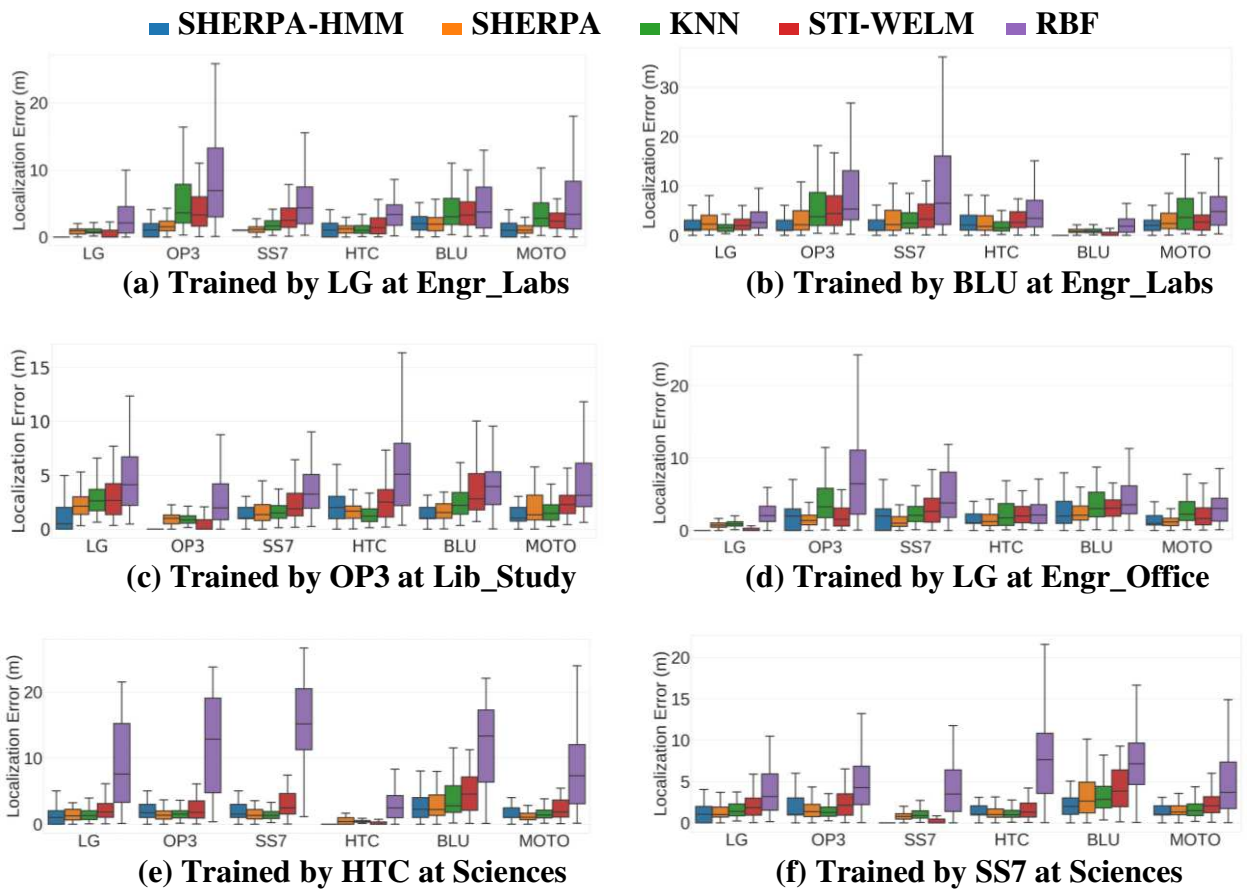


Figure 23. Localization error for various techniques on benchmark paths across training devices.

From Figure 23(a), it can be observed that HTC is the most stable de-vice for KNN, i.e., is least affected by heterogeneity. In all other situations, localization error is heavily impacted by heterogeneity. Overall, in Figure 23(a) and (b), SHERPA-HMM can be seen to outperform RBF and STI-WELM whenever the localization error from KNN is > 2 meters. SHERPA-HMM is also better than our SHERPA in most cases. We observe that RBF performs the worst when there is a significant amount of metal structures in the environment. This is the case for the engineering building paths (Engr_Labs, Engr_Office) and the path in the Sciences building. The perturbations in the WiFi WAP RSSI values due to the metallic surroundings cause the ranks of the WAP RSSI values to become highly unstable. We noted that RBF performed better than KNN for a few walks, but this was averaged out by poor results from other iterations of the same walk.

From Figure 23, we also observe that SHERPA-HMM outperforms STI-WELM in most training-testing device pairs, other than the non-heterogeneous cases (e.g., LG boxplot in 8(a), BLU boxplot in 8(b), etc.). SHERPA-HMM is able to deliver better performance in most cases as it is a purely pattern matching approach along a path. STI-WELM identifies the closest sampled locations from the offline phase using the scaling and shape matching based STI metric. The fingerprints of these closest locations are then used to train a WELM based neural network in the online phase. The work in [40] (STI-WELM) assumes a constant gain across heterogeneous devices which is not the case (from Figure 17) and does not compensate for noise across smartphones. The neural network model itself is not especially designed for pattern matching, and sacrifices predictability of localization error for faster training time in the online phase. Further, a neural network-based localization framework such as STI-WELM requires extremely large sets of training data which may not a be a realistic and scalable approach for indoor environments. In the few cases that SHERPA-HMM is outperformed by STI-WELM, SHERPA-HMM still performs within the acceptable range of accuracy and is very close to STI-WELM in terms of median error. We also note that for most paths considered in Figure 23, SHERPA-HMM outperforms KNN. In the few cases where it is outperformed by KNN, its accuracy loss is very low.

In some of the cases such as in Figure 23(d), we observe that SHER-PA-HMM delivers relatively higher localization error as compared to SHERPA. We found that the major cause of this was that the HMM model falsely predicts that a user has turned back when the user is actually moving forward along a path. This is caused by noisy fingerprints and the fact that we are using a simple transition matrix where the probability of the user moving in any direction is the same. Also, we do not utilize other motion sensors such as magnetic and gyroscope to identify situations

where the user is changing directions. However, even with this drawback SHERPA-HMM is able to meet our tar-get accuracy of 2 meters across the board.

The experiments performed in this work revealed that certain devices such as the low-cost BLU smartphone produce particularly noisy and inconsistent WiFi RSSI measurements. Even though SHERPA-HMM attempts to minimize the impact of noise by taking into account multiple WiFi scans for each location prediction, users should be wary of the quality limitations of such low-cost devices, especially when using them for indoor localization and navigation.

3.6.4. COMPARISON OF EXECUTION TIMES

To highlight the lightweight design of our approach, we show the mean execution time of location predictions for SHERPA-HMM and prior work frameworks executing on the OP3 device. For brevity, results for only one path (Lib_Study) are shown. The specific path was chosen for this experiment as it was the largest one with 13,080 data points (60 meters × 218 WAPs) available. The OP3 device was randomly chosen as we expect the overall trends of this experiment to re-main the same across smartphones.

The results of this experiment are shown in Figure 24. The RBF technique is found to take over 2 seconds to execute. This behavior can be attributed to the fact that RBF requires sorting of WiFi RSSI values for every scanned fingerprint in the testing phase, unlike any of the other techniques. STI-WELM takes the least time to predict locations. However, the highly degraded accuracy with STI-WELM, especially in the presence of device heterogeneity (as seen in Figure 23) is a major limitation for STI-WELM. After STI-WELM (Figure 24), SHERPA is one of the quickest localization frameworks with an average prediction time of 0.43 seconds that is slightly lower than the lightweight Euclidean-based KNN approach that takes 0.47 seconds for a

prediction. Finally, SHERPA-HMM delivers its prediction results in 0.48 seconds which is only slightly higher than KNN. As compared to SHERPA, SHERPA-HMM takes ~0.05 seconds longer but has proven to deliver significantly better results as shown in section 3.6.3.

In summary, from the results presented in this section, it is evident that our proposed SHERPA-HMM framework for is a promising approach that provides highly accurate, lightweight, smartphone heterogeneity-resilient indoor localization. A major strength of this framework is that it can be easily ported across smartphones without the need of any calibration effort or cloud-based service to execute.

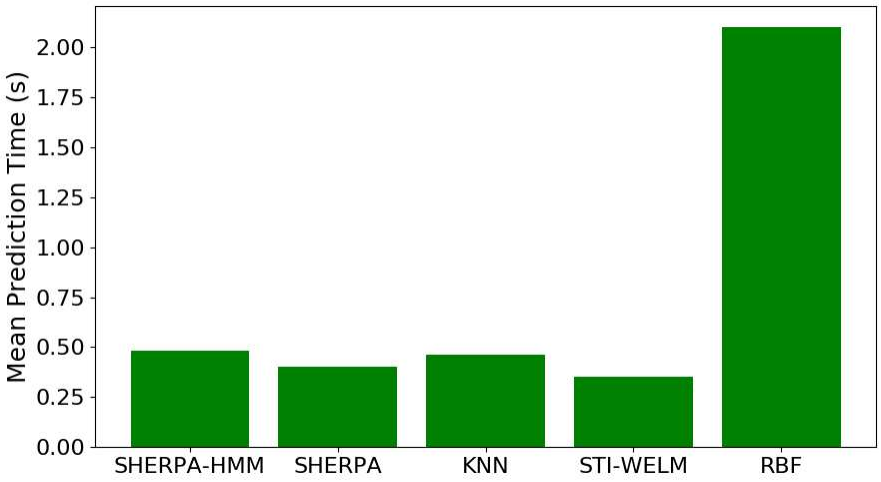


Figure 24. Mean indoor location prediction time for SHERPA-HMM and frameworks from prior work for the Lib_Study path using the OnePlus3 device.

3.7. CONCLUSION AND FUTURE WORK

In this chapter, we proposed the SHERPA-HMM framework that is a computationally lightweight solution to the mobile device heterogeneity problem for fingerprinting-based indoor localization. Our analysis in this work provides important insights into the role of mobile device heterogeneity on localization accuracy. SHERPA-HMM was able to deliver superior levels of

accuracy as compared to state-of-the-art in-door localization techniques using only a limited number of samples for each fingerprinting location. We also established that developing algorithms that can be easily ported across devices with minimal loss in localization accuracy is a crucial step towards the actuation of finger-printing-based localization frameworks in the real world.

As part of our future work, we would like to focus on improving the reliability of the proposed framework through incorporating inertial and magnetic information in the HMM formulation. This would greatly reduce the chances of the Viterbi algorithm from predicting false user movement direction changes. Another improvement could be to dynamically increase the scan memory variable such that user predictions are made with higher confidence in situations where the online fingerprint is noisy.

4. ADAPTING CONVOLUTIONAL NEURAL NETWORKS FOR INDOOR LOCALIZATION WITH SMART MOBILE DEVICES[1]

Existing outdoor location-based services have transformed how people navigate, travel, and interact with the world around them. Now, indoor localization techniques are emerging that have the potential to extend this outdoor experience across indoor locales. Industry is beginning to provide indoor location-based services to improve customer experience. For instance, Google can suggest products to its users through targeted indoor location-based advertisements [68]. Stores such as Target in the USA are beginning to provide indoor localization solutions to help customers locate products in a store and find their way to these products [8]. Services provided by these companies combine GPS, cell towers, and WiFi data to estimate the user's location. However, in the indoor environment where GPS signals cannot penetrate building walls, the accuracy of these geo-location services can be in the range of tens of meters, which is insufficient in many cases [69].

Many of the latest indoor localization techniques exploit radio signals, such as Bluetooth, UWB (Ultra-Wide Band) [70], RFID (Radio Frequency Identification) [33], or other customized radios. The key idea is to use characteristics of radio signals (e.g., signal strength or triangulation) to estimate user location relative to a radio beacon (wireless access point). But these techniques suffer from multipath effects, signal attenuation, and noise-induced interference [47]. Also, as these techniques require specialized wireless radio beacons to be installed in indoor locales, they are costly and thus lack scalability for wide-scale deployment.

[1] The work presented in this chapter was conducted in collaboration with Ayush Mittal

WiFi based fingerprinting is perhaps the most popular radio-signal based indoor localization technique being explored today. WiFi is an ideal radio signal source for indoor localization as most public or private buildings are pre-equipped with WiFi access points (APs). Lightweight middleware-based fingerprinting frameworks have been shown to run in the background to deliver location-based updates on smartphones [71]. Fingerprinting with WiFi works by first recording the strength of WiFi radio signals in an indoor environment at different locations. Then, a user with a smartphone can capture WiFi received signal strength indication (RSSI) data in real-time and compare it to previously recorded (stored) values to estimate their location in that environment. Fingerprinting techniques can deliver an accuracy of 6 to 8 meters [2], with accuracy improving as the density of APs increases. However, in many indoor environments, noise and interference in the wireless spectrum (e.g., due to other electronic equipment, movement of people, operating machinery, etc.) can reduce this accuracy. Combining fingerprinting-based frameworks with dead reckoning can improve this accuracy somewhat [37]. Dead reckoning refers to a class of techniques where inertial sensor data (e.g., from accelerometer, gyroscope) is used along with the previously known position data to determine the current location. But dead reckoning is known to suffer from error accumulation (in inertial sensors) over time. Also, these techniques are not effective for people using wheelchairs or moving walkways.

The intelligent use of machine learning (ML) techniques can help to overcome noise and uncertainty during fingerprinting-based localization [37]. While traditional ML techniques work well at approximating simpler input-output functions, computationally intensive deep learning models are capable of dealing with more complex input-output mappings and can deliver better accuracy. Middleware-based offloading [72] [73] and energy enhancement frameworks [37] [47] [50] [74] may be a route to explore for computation and energy-intensive indoor localization

services on smartphones. Furthermore, with the increase in the available computational power on mobile devices, it is now possible to deploy deep learning techniques such as Convolutional Neural Networks (CNNs) on smartphones. A CNN is a special type of Deep Neural Network (DNN) that is geared towards image matching and recognition. The most popular aspect of CNN is that it can automatically identify essential input features that make the most impact towards the correctness of the final output. This process is known as feature learning. Prior to deep learning, feature learning was an expensive and time intensive process that had to be conducted manually. CNN has been extremely successful in complex image classification problems and is finding applications in many emerging domains, e.g., self-driving cars [75].

In this chapter, we propose a new and efficient framework that uses CNN-based WiFi fingerprinting to deliver a superior level of indoor localization accuracy to a user with a smartphone. Our approach utilizes widely available WiFi APs without requiring any customized/expensive infrastructure deployments. The framework works on a user's smartphone, within the computational capabilities of the device, and utilizes the radio interfaces for efficient fingerprinting-based localization. The main novel contributions of this chapter can be summarized as follows:

- We developed a new technique to extract images out of location fingerprints, which are then used to train a CNN that is designed to improve indoor localization robustness and accuracy;
- We implemented a hierarchical architecture to scale the CNN, so that our framework can be used in the real world where buildings can have large numbers of floors and corridors;
- We performed extensive testing of our algorithms with the state-of-the-art across different buildings and indoor paths, to demonstrate the effectiveness of our proposed framework.

4.1 RELATED WORK

Several efforts aim to address the challenges in the domain of indoor localization. Here we summarize some of the key efforts.

Several RFID [33], [76] based indoor localization solutions that use proximity-based estimation techniques have been proposed. But the hardware expenses of these efforts increase dramatically with increasing accuracy requirements. Also, these approaches cannot be used with smartphones and require the use of specialized hardware. Indoor localization systems that use UWB [70] and ultrasound [77] [78] have similar requirements for additional (costly) infrastructure, and a lack of compatibility for use with commodity smartphones.

Triangulation based methods, such as [79], use multiple antennas to locate a person or object. But these techniques require several antennas and regular upkeep of the associated hardware. Most techniques therefore favor using the more lightweight fingerprinting approach, often with WiFi signals. UJIIndoorLoc [80] describes a technique to create a WiFi fingerprint database and employs a KNN (K-Nearest Neighbor) based model to predict location. Their average accuracy using KNN is 7.9 meters. Dead reckoning techniques use the accelerometer to estimate the number of steps, a gyroscope for orientation, and a magnetometer to determine the heading direction. Such techniques have been employed in [20] and [23], but have shown to deliver poor localization accuracy results when used alone.

Radar [81] and Indoor Atlas [10] proposed using hybrid indoor localization techniques. Radar [81] combines inertial sensors (dead reckoning) with WiFi signal propagation models, whereas Indoor Atlas [10] combines information from several sensors such as magnetic, inertial, and camera sensors, for localization. LearnLoc [37] combines non-deep ML models, dead

reckoning techniques, and WiFi fingerprinting to trade-off indoor localization accuracy and energy efficiency during localization on smartphones.

A few efforts have begun to consider deep learning to assist with indoor localization. The work in [82] presents an approach that uses DNNs with WiFi fingerprinting. The accuracy of the DNN is improved by using a Hidden Markov Model (HMM). The HMM takes temporal coherence into account and maintains a smooth transition between adjacent locations. But our analysis shows that the fine location prediction with the HMM fails in cases such as when moving back on the same path or taking a sharp turn. HMM predictions are also based on the previous position acquired through the DNN and hence, can be prone to error accumulation. DeepFi [35] and ConFi [83] propose approaches that use the Channel State Information (CSI) of WiFi signals to create fingerprints. But the CSI information in these approaches was obtained through the use of specialized hardware attached to a laptop. None of the mobile devices available today have the ability to capture CSI data. Due to this limitation, it is not feasible to implement these techniques on smartphones. Deep Belief Networks (DBN) [84] have also been used for indoor localization, but the technology is based on custom UWB beacons that lead to very high implementation cost.

In summary, most of the above-mentioned frameworks either require additional costly infrastructure or cannot be deployed on smart mobile devices. Our implementation-based analysis shows that these frameworks can become slow and resource intensive if used for large buildings with multiple floors and corridors.

Our proposed framework in this chapter, CNN-LOC, overcomes the shortcomings of these state-of-the-art indoor localization approaches. CNN-LOC creates input images by using RSSI of WiFi signals that are then used to train a CNN model, without requiring any specialized hardware/infrastructure. CNN-LOC is easily deployable on current smartphones. The proposed

framework also integrates a hierarchical scheme to enable scalability for large buildings with multiple floors and corridors/aisles.

4.2 CONVOLUTIONAL NEURAL NETWORKS

Convolutional Neural Networks (CNNs) are specialized DNNs with a focus on image classification. They are highly resilient to noise in the input data and have shown to deliver excellent results for complex image classification tasks. The smallest unit of any neural network (NN) is a perceptron and is inspired by the biological neuron present in the human brain.

Here y is the output, which is a weighted sum of the inputs x_i, with a weighted bias (w_0). NNs have inter-connected layers, and in each layer, there are several perceptrons, each with its own tunable weights and biases. Each layer receives some input, executes a dot product, and passes it to the output layer or the hidden layer in front of it [85]. This output is often applied to an activation function that gives an input-output mapping defined by logistic regression. The most common activation functions used are *sigmoid* and *tanh* functions. The goal of an NN is to approximate a functional relationship between a set of inputs and outputs (training phase). The resulting NN then represents the approximated function that is used to make predictions for any given input (testing phase).

While an NN often contains a small number of hidden layers sandwiched between the input and output layer, a Deep Neural Network (DNN) has a very large number of hidden layers. DNNs have a much higher computational complexity but in turn are also able to deliver very high accuracy. CNNs are a type of DNN that include several specialized NN layers, where each layer may serve a unique function. CNN classifiers are used to map input data to a finite set of output classes. For instance, given different animal pictures, a CNN model can be trained to categorize

them into different classes such as cats, dogs, etc. CNNs also make use of Rectified Linear Units (ReLu) as their activation function, which allows them to handle non-linearity in the data.

In the training phase, our CNN model uses a feed forward deep learning algorithm. To update the weights during the training phase, a Stochastic Gradient Descent (SGD) algorithm is used. Adam [86], an optimized version of SGD, is used to optimize the learning process. The algorithm is designed to take advantage of two well-known techniques: RMSprop [87] and AdaGrad [88]. SGD maintains a constant learning rate for every weight update in the network. In contrast, Adam employs an adaptive learning rate for each network weight; with the learning rate being adapted as the training progresses. RMSprop uses the mean (first-order moment) of past squared gradients and adjusts the weights based on how fast the gradient changes. Adam, to optimize the process, uses the variance (second-order moment) of past gradients and adjusts the weights accordingly.

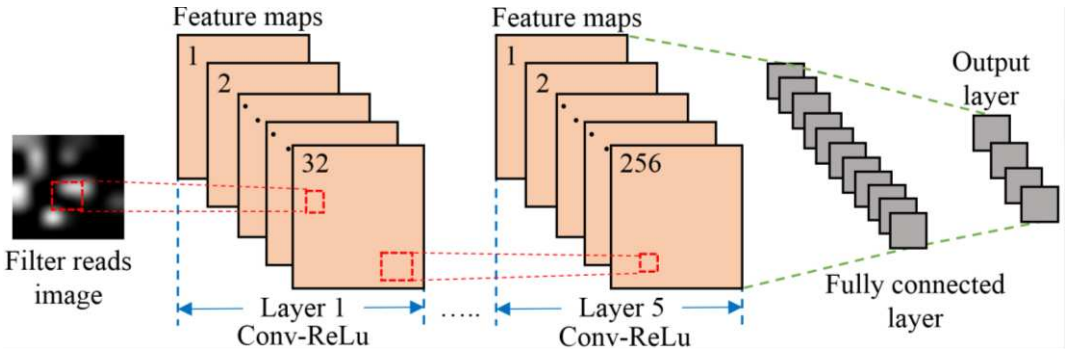


Figure 25. The architecture of a sample Convolutional Neural Network (CNN).

The structure of the CNN in *CNN-LOC* is inspired from the well-known CNN architectures, LeNet [89] and AlexNet [85]. Our CNN architecture is shown in Figure 25. The first hidden layer is partially connected to the input layer. This hidden layer only looks at a specific region of the input image at a time, and this region is known as a filter. The filter is shown by a rectangle (red-dotted lines). Each layer performs a convolution of a small region of the input image with the filter

and feeds the result to the ReLu activation function. Therefore, we refer to each layer as [Conv-ReLu]. To capture more details from the input image we can use a larger number of filters. For each filter, we get a feature map. For the first layer of [Conv-ReLU], we used 32 filters to create a set of 32 feature maps. We used five hidden layers of [Conv-ReLU], but only two are shown for brevity. The number of filters and layers are derived through empirical analysis as discussed in section 4.4. A 'stride' parameter determines the quantity of pixels that a filter will shift, to arrive at a new region of the input image to process. The stride and other 'hyperparameters' of our CNN are further discussed in section 4.4. In the end, a fully connected layer helps in identifying the individual class scores (in our case each class is a unique location). The class with the highest score is selected as the output. In this layer, all the neurons are connected to the neurons in the previous layer (green-dotted-lines).

In a conventional CNN, a pooling layer is used to down-sample the image when the size of the input image is too big. In our case, the input image is small and therefore we do not need this step. We want our CNN to learn all the features from the entire image.

4.3 CNNLOC FRAMEWORK

4.3.1. OVERVIEW

An overview of our CNN-LOC indoor localization framework is shown in Figure 26. In the framework, we utilize the available WiFi access points (APs) in an indoor environment to create an RSSI fingerprint database. Our framework is divided into two phases. The first phase involves RSSI data collection, cleaning, and pre-processing. This pre-processed data is used to create a database of images. Each image represents a WiFi RSSI based signature that is unique to a location (i.e., x-y co-ordinate). This database of images is used to train a CNN model. The trained model

is deployed on a smartphone. In the second phase, real time AP data is converted into an image and then fed to the trained CNN model to predict the location of the user. The CNN model predicts the closest block that was sampled as the users' location. A detailed description of the pre-processing is described in the next section.

4.3.2. PRE-PROCESSING OF RSSI DATA

The process of image database creation begins with the collection of RSSI fingerprints as shown in the top half of Figure 26. The RSSI for various APs are captured along with the corresponding x and y coordinates at the training locations. We only maintain information for known WiFi APs and hence clean the captured data. This ensures that our trained model is not polluted by unstable WiFi APs. On the RSSI scale, values typically range between -95 dB (lowest) to -0 dB (highest). We normalize the RSSI values on a scale from 0 and 100, where 0 represents the weak or null signal, and 100 represents the strongest signal.

Assume that while fingerprinting an indoor location, a total of K APs are discovered at N unique locations. These combine to form a two-dimensional matrix of size $N \times K$. Then the normalized RSSI fingerprint at the N^{th} location, denoted as l_N, is given by a row vector $[r_1, r_2, \ldots, r_K]$, denoted by R_N. Therefore, each column vector, $[w_1, w_2, \ldots, w_N]$ would represent the normalized RSSI values of the K^{th} AP at all N locations, denoted by W_K. We calculate the Pearson Correlation Coefficient (PCC) [90] between each column vector W_K and the location vector $[l_1, l_2, \ldots, l_N]$. The result is a vector of correlation values denoted as C. PCC is useful in identifying the most significant APs in the database that impact localization accuracy. The coefficient values range across a scale of -1 to +1. If the relationship is -1, it represents a strong negative relationship,

whereas +1 represents a strong positive relationship, and 0 implies that the input and output have no relationship.

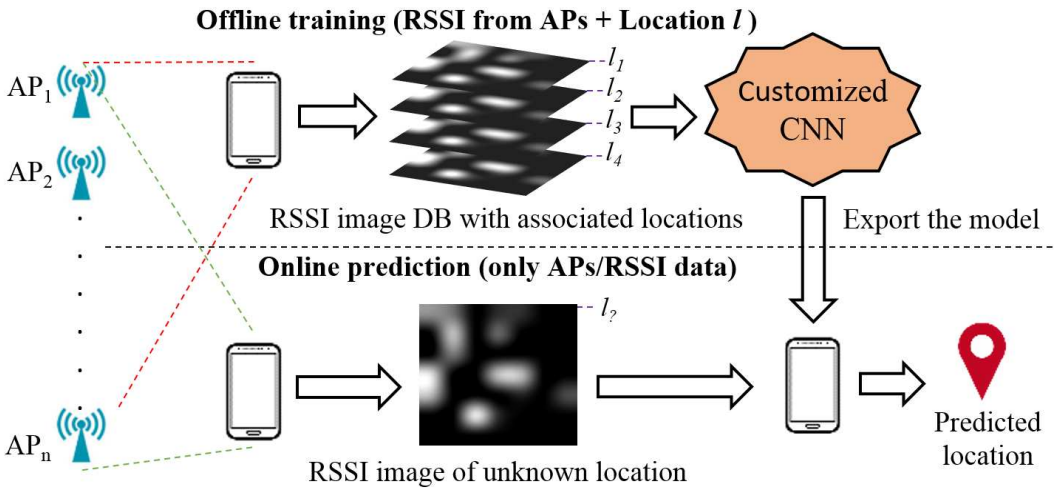


Figure 26. An overview of the CNN-LOC framework.

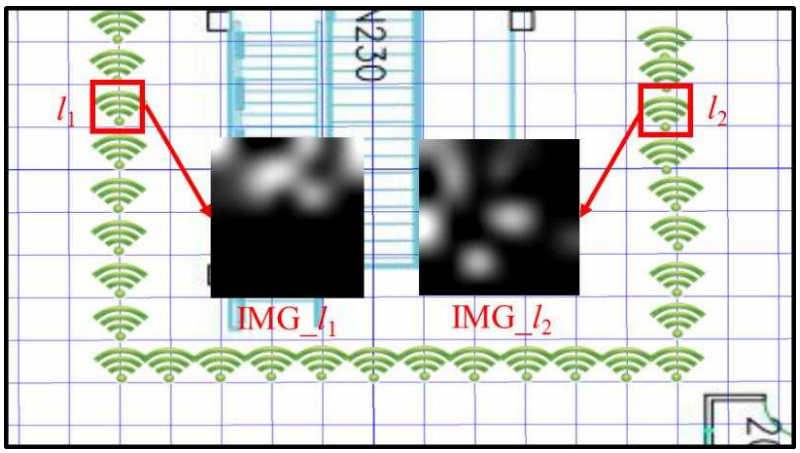


Figure 27. Unique images created for locations l1 and l2. The green icons represent locations that are fingerprinted along an indoor path. The two locations shown are 10 meters apart.

We only consider the magnitude of the correlation as we are only concerned with the strength of the relationship. APs with very low correlation with the output coordinates are not useful for the purpose of indoor localization. Therefore, we can remove APs whose correlation to the output coordinates is below a certain threshold ($|PCC| < 0.3$). This removes inconsequential APs from the

collected WiFi data and helps reduce the computational workload of the framework. The normalized RSSI data from the remaining high-correlation APs is used to create an RSSI image database, as explained in the next section.

4.3.3. RSSI IMAGE DATABASE

In this section, we present our approach to convert RSSI data for a given location into a greyscale image. A collection of these images for all fingerprinted locations forms the RSSI Image Database. To form greyscale images, a Hadamard Product (*HP*) [91] is calculated for each *R* and *C*. HP is defined as an element wise multiplication of two arrays or vectors:

$$HP = \sum_{i=1}^{N} R_i \circ C \tag{3}$$

The dimension of each *HP* is $1 \times K$. Then, the *HP* matrix is reshaped into a $p \times p$ matrix, which represents a 2D image as shown in Figure 27. The HP is padded with zeros in the case that *K* is less than p^2. Therefore, we now have a set of *N* images of size $p \times p$ in our database. These images are used to train the CNNs.

Figure 27 shows two images (*IMG_l_1 and IMG_l_2*) of size 7×7 created for two unique fingerprints (signatures) associated with two different locations. Each pixel value is scaled on a scale of 0 to 255. The patterns in each of these images will be unique to a location and change slightly as we move along an indoor path.

In equation (3), the product of PCC and normalized RSSI value for each AP is used to form a matrix. Its purpose is to promote the impact of the APs that are highly correlated to fingerprinted locations. Even though there may be attenuation of WiFi signals due to multipath fading effects, the image may fade but will likely still have the pattern information retained. These patterns that

are unique to every location can be easily learned by a CNN. The hyperparameters and their use in *CNN-LOC* is discussed next.

4.3.4. HYPERPARAMETERS

The accuracy of the CNN model depends on the optimization of the hyperparameters that control its architecture which is the most important factor in the performance of CNN. A smaller network may not perform well, and a larger network may be slow and prone to overfitting. There are no defined rules in deep learning that help in estimating the appropriate hyperparameters. Identifying the optimal values for the CNN hyperparameters is an empirical process and requires several iterations of experimentation and analysis. The estimated hyperparameters are also highly dependent on the input dataset. Below, we discuss results of our analysis of CNN hyperparameters for our indoor localization problem domain.

- Number of hidden layers: A large number of hidden layers lead to longer execution times and conversely, fewer hidden layers may produce inaccurate results. We found that 5 layers of [Conv-ReLU] works best for our domain.
- Size of filter: This defines the image area that the filter considers at a time, before moving to the next region of the image. A large filter size might aggregate a large chunk of information in one pass. The optimum filter size in our case was found to be 2×2.
- Stride size: The amount of pixels a filter moves by is dictated by the stride size. We set it to 1 because the size of our image is very small and we do not wish to lose any information.
- Number of filters: Each filter extracts a distinct set of features from the input to construct different feature maps. Each feature map holds unique information about the input image. The best results were obtained if we started with a lower number of filters and increased them in

the successive layers to capture greater uniqueness in the patterns. There were 32 filters in the first layer and were doubled for each subsequent layer up to 256 filters such that both the fourth and fifth layer had 256 filters.

4.3.5. INTEGRATING HIERARCHY FOR SCALABILITY

Our *CNN-LOC* framework is designed to scale up to larger problem sizes than that handled by most prior efforts. For this purpose, we enhanced our framework by integrating a hierarchical classifier. The resulting hierarchical classifier employs a combination of smaller CNN modules, which work together to deliver a location prediction. Figure 28 shows the hierarchical decision structure of the framework. Each CNN module has a label that starts with C. The CNN in the first layer (C1) classifies the floor numbers, and then in the next layer, C20 or C21 identify the corridor on that floor. Once the corridor is located, one of the CNNs from the third layer (C30 – C35) will predict the fine-grain location of the user. It is important to note that the CNN models in the third layer actually represent two models each, i.e., C30 includes both CNN models for the x and y axis. In this manner, we avoid using the hierarchal classifier twice for each axis.

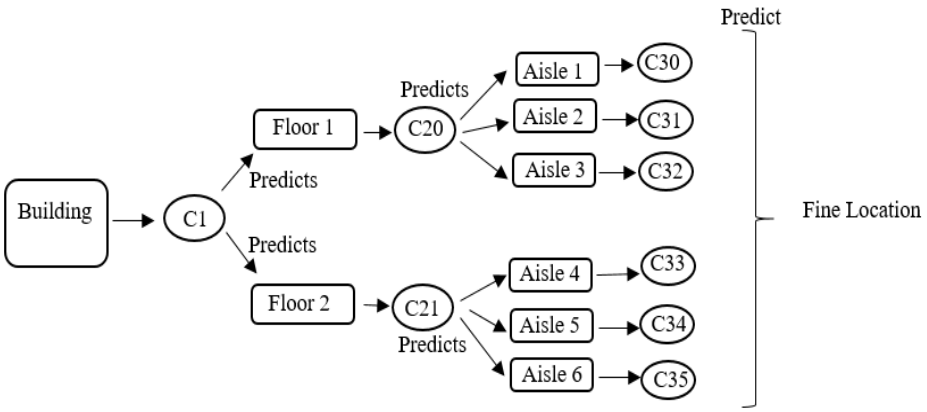


Figure 28. A general architecture for the hierarchical classifier.

4.4 EXPERIMENTS

4.4.1. EXPERIMENTAL SETUP

This section describes the *CNN-LOC* implementation and experimental results that were conducted on three independent indoor paths as described in Table 2. The corridors on the path are divided into a grid and labelled sequentially from 1 to *N*. Each square in the grid has an area of 1 m^2 and represents a "class". This allows us to treat indoor localization as a classification problem for CNN. Figure 29 shows an example of a path covered in the library building with labeled squares. Each label further translates into an x-y coordinate. Five WiFi scans were conducted at each square during the fingerprinting (training) phase.

Table 2. Indoor paths used in experiments.

Building	Path Length (m)	Shape
Library	30	U shape
Clark A	35	Semi-octagonal
Physics	28	Square shape

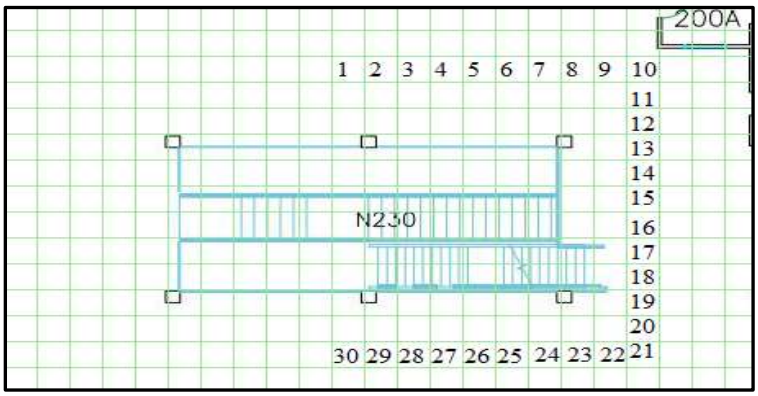


Figure 29. Library building path divided into a grid, with squares along the path labeled sequentially from 1 to 30.

An Android application was built to collect WiFi fingerprints (i.e., RSSI samples from multiple APs at each location) and for testing. The application is compatible with Android 6.0 and

was tested on a Samsung Galaxy S6. After fingerprint data collection, the data was pre-processed as described in the previous section for the CNN model. The entire data set is split into training and testing samples, so we can check how well our models perform. We used $1/5^{th}$ of the total samples for testing and $4/5^{th}$ of the samples were used for training. Also, we implemented different CNN models for location estimation along different axes. Thus, we use a dedicated CNN model to predict the x coordinate of a location. Similarly, a separate CNN model predicts the y coordinates. The output from these models is combined to get the final results.

4.4.2. EXPERIMENTAL RESULTS

We compared our *CNN-LOC* indoor localization framework with three other indoor localization frameworks from prior work. The first work we implemented is based on the approach in [92] and employs Support Vector Regression (SVR). This approach forms one or more hyperplanes in a multidimensional space such that it segregates similar data points, which are then used for regression. The second work is based on the KNN technique from [37], which is a non-parametric approach that is based on the idea that similar input will have similar outputs. Lastly, we compare our work against a DNN based approach [82] that improves upon conventional NNs by incorporating a very large number of hidden layers. All of these techniques supplement the WiFi fingerprinting approach with a machine learning model to provide robustness against noise and interference effects. Our experiments in the rest of this section first discusses the localization accuracy results for the techniques. Subsequently, we also discuss results for the scalability of our framework using a hierarchical classification enhancement approach. Lastly, we contrast the accuracy of our framework with that reported by other indoor localization techniques.

4.4.2.1. INDOOR LOCALIZATION ACCURACY COMPARISON

Figure 30 shows the paths predicted by the four techniques, for the indoor path in the Clark building. The green dots along the path represent the points where WiFi RSSI fingerprint samples were taken to create the training dataset. The distance between each of the green dots is 1 meter. In the training dataset, each green dot is converted into an image. The testing phase consists of the user walking along this path, and the red lines in Figure 30 show the paths predicted by the four techniques. It is observed that KNN [37] and SVR [92] stray off the actual path the most, whereas DNN and *CNN-LOC* perform much better. This is likely because KNN and SVR are both regression-based techniques where the prediction is impacted by neighboring data points. In cases where the sampled points are very close to each other, there may not be enough variation across neighboring samples for the regression-based techniques to work properly. The transition from one location to another is smoother for CNN as it is able to distinguish between closely spaced sampling locations due to our RSSI-to-image conversion technique. From Figure 30, it is evident that our *CNN-LOC* framework produces stable predictions for the Clark path.

Figure 31 shows a bar graph that summarizes the average location estimation error for the various techniques on the three different indoor paths considered. We found that the KNN approach is the least reliable among all techniques with a mean error of 5.5 meters and large variations across the paths. The SVR-based approach has a similar mean error as the KNN approach. The DNN based approach shows lower error across all of the paths. But it does not perform consistently across all of the paths and the mean error is always higher than that for *CNN-LOC*. This may be due to the fact that the filters in CNN are set up to focus on the image with a much finer granularity than the DNN approach is capable of. We also observe that all techniques perform the worst in the Physics department. This is due to the fact that the path in the Physics department is near the entrance of the building and has a lower density of WiFi APs as compared

to the other paths. The Library and Clark paths have a higher density of WiFi APs present; hence, better accuracy can be achieved. Our proposed *CNN-LOC* framework is the most reliable framework with the lowest mean error of less than 2 meters.

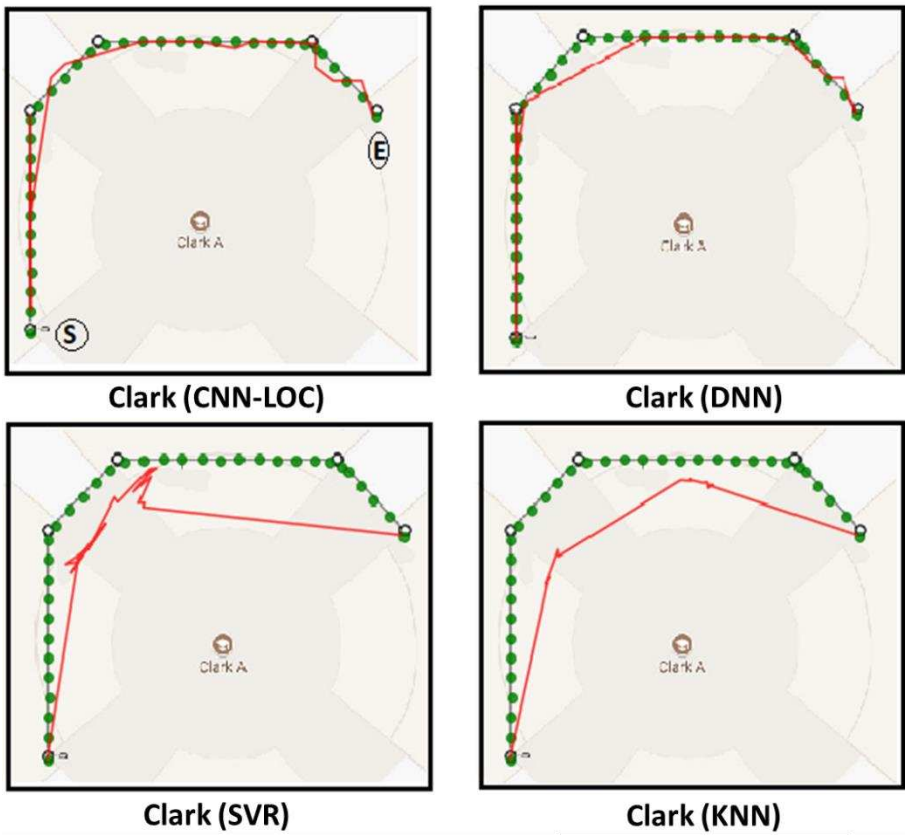


Figure 30. Path traced using different techniques.

4.4.2.2. CNN-LOC SCALABILITY ANALYSIS

We discuss results for the hierarchal *CNN-LOC* (Section 4.5) here. We consider a scenario when *CNN-LOC* is required to predict a location inside a building with two floors and with three corridors on each floor. The length of each corridor is approximately 30 meters. We combined several small CNNs (in our case 9 small CNNs), such that a smaller number of weights are associated with each layer in the network than if a single larger CNN was used.

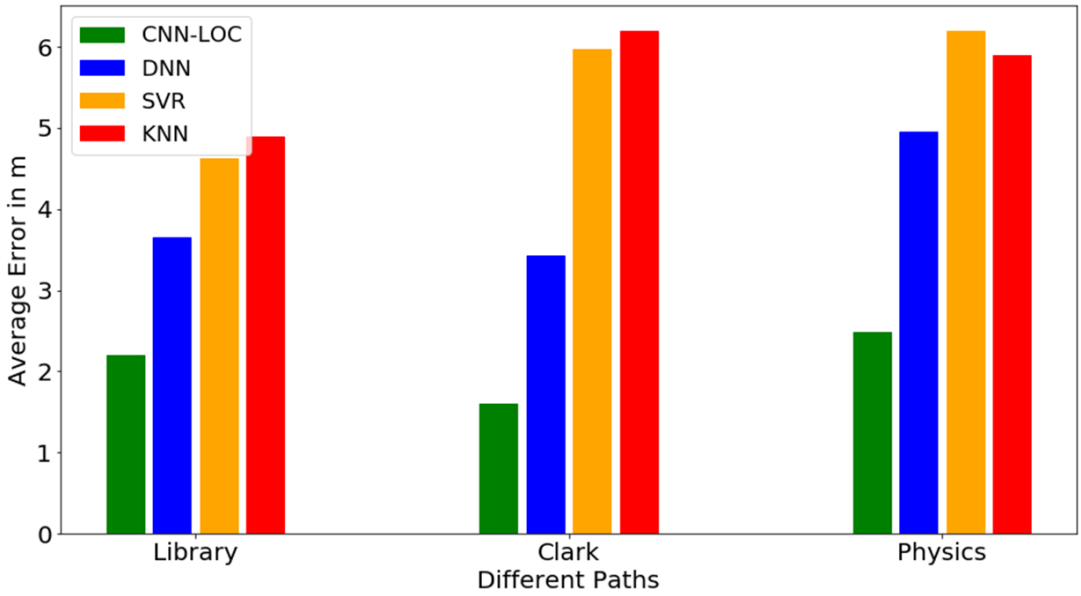


Figure 31. Comparison of indoor localization techniques.

We first analyzed the accuracy of predictions, for *CNN-LOC* with and without the hierarchical classifier. For the first and second layer of the hierarchical classifier (shown in Figure 28), the accuracy is determined by the number of times the system predicts the correct floor and corridor. We found that floors and corridors were accurately predicted 99.67% and 98.36% of times, respectively. For the final layer, we found that there was no difference in accuracy between the hierarchal and the non-hierarchal approach. This is because in the last level both the approaches use the same model.

Figure 32 shows the benefits in terms of time taken to generate a prediction with the hierarchical versus the non-hierarchical *CNN-LOC* framework. We performed our experiment for four walking scenarios ("runs") in the indoor environment (building with two floors and with three corridors on each floor). We found that the hierarchical *CNN-LOC* model only takes 2.42ms to make a prediction on average, whereas the non-hierarchical *CNN-LOC* takes longer (3.4ms). Thus, the hierarchical classifier represents a promising approach to reduce prediction time due to the

fewer number of weights in the CNN layers in the hierarchical approach, which leads to fewer computations in real-time.

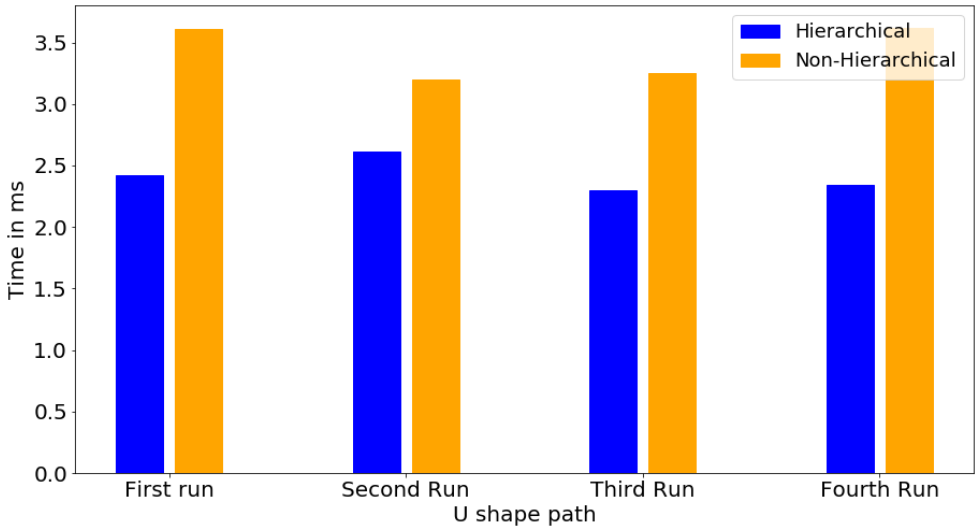


Figure 32. Execution time for Hierarchical CNN.

4.4.2.3. ACCURACY ANALYSIS WITH OTHER APPROACHES

Our experimental results in the previous sections have shown that *CNN-LOC* delivers better localization accuracy over the KNN [37], DNN [82] and SVR [92] frameworks. The UJIIndoorLoc [80] framework is reported to have an accuracy of 4 to 7 meters. Our average accuracy is also almost twice that of RADAR [81]. If we consider frameworks that used CSI (DeepFi [35] and ConFi [83]), our accuracy is very close to both at just under 2 meters. However, [35] and [83] use special equipment to capture CSI and cannot be used with mobile devices. In contrast, our proposed *CNN-LOC* framework is easy to deploy on today's smartphones, does not require any specialized infrastructure (e.g., custom beacons), and can be used in buildings wherever WiFi infrastructure pre-exists.

4.5. CONCLUSIONS

In this chapter, we presented the *CNN-LOC* framework that uses WiFi fingerprints and convolutional neural networks (CNNs) for accurate and robust indoor localization. We compared our work against three different state-of-the-art indoor localization frameworks from prior work. Our framework outperforms these approaches and delivers localization accuracy under 2 meters. *CNN-LOC* has the advantage of being easily implemented without the overhead of expensive infrastructure and is smartphone compatible. We also demonstrated how a hierarchical classifier can improve the scalability of this framework. *CNN-LOC* represents a promising framework that can deliver reliable and accurate indoor localization for smartphone users.

5. OVERCOMING SECURITY VULNERABILITIES IN DEEP LEARNING BASED INDOOR LOCALIZATION FRAMEWORKS ON MOBILE DEVICES

In the early 1980's, the unintended deviation of a commercial airliner from its designated path due to unreliable navigation equipment led to 269 casualties [93]. This prompted U.S. authorities to recognize the need for a reliable global localization solution. As a result, the Global Positioning System (GPS) being built for the U.S military, when completed, was promised to be available for public use. In the subsequent decade, GPS technology was completely commercialized [94]. These historic events reformed the global transportation industry and allowed vehicles to not only localize themselves but also to navigate reliably. To further enhance security of GPS based services, recent works have started to focus on the modeling and characterization of GPS spoofing [95] and time reliability-based attacks [96] and further propose the utilization of crowdsourcing methodologies to detect and localize spoofing attacks [97]. Regardless of such advances the recent history of attacks on GPS for outdoor navigation [98], [99] motivates stronger security features. On the other hand, indoor localization is an emerging technology with a similar purpose and is poised to reinvent the way we navigate within buildings and subterranean locales [2]. However, on the academic front, limited attention is being paid towards securing indoor localization and navigation frameworks against malicious attacks and ensuring that the future indoor localization frameworks are reliable.

Almost two decades of research has contributed to the evolution of the indoor localization and navigation domain. Several commercial solutions and standards are being established today to enable indoor localization in the public sector. For example, recently a new standard for WiFi was established in collaboration with Google that would allow anyone to set up their own localization

system by sharing their indoor floor map and the WiFi router positions on that map with Google [4]. Nowadays, companies such as Amazon and Target are also starting to track customers at their stores [11]. With an increasing number of startups in the area of indoor localization services security concerns pertaining to the commercialization of such technology are almost never discussed.

The explosion in the commercialization of indoor localization technology can be attributed to its usefulness for a wide variety of non-critical and critical applications. For example, depending on the context of the situation [100], navigating students to the correct classroom may represent non-critical applications, where some factor of unreliability would not lead to any serious repercussions. However, there are some applications in a time-critical response context and need an enhanced level of reliability and security. Such scenarios include navigating medical staff and equipment closest to a patient in the correct room at a hospital in real-time or notifying emergency responders to the location of a person in case of a serious health hazard such as a heart attack, collapse, or fire.

Unfortunately, malicious third parties can exploit the vulnerabilities of unsecured indoor localization components (e.g., WiFi Access Points or WAPs) to produce incorrect localization information [101], [102]. This may lead to some inconvenience in non-critical contexts (e.g., a student arrives at the wrong classroom), but can lead to dire consequences in more critical contexts (e.g., medical staff are unable to locate vital equipment or medicine needed for a patient in an emergency; or emergency response personnel are misdirected, causing a loss of lives). Tainted information from intentional or unintentional sources can lead to even more egregious real-time delays and errors. Therefore, similar to outdoor navigation systems, establishing secure and

reliable indoor localization and navigation systems holds an uncontested importance in this domain.

Despite much research on indoor localization solutions, the security and reliability concerns of the proposed indoor localization frameworks are often overlooked. The vulnerabilities and associated security methodologies that can be applied to an indoor localization framework are often tailored to the localization method used and a generalized security and reliability framework is not available.

For the purpose of indoor localization, at one end of the spectrum are triangulation/trilateration-based methods that either use geometric properties such as the distance between multiple APs and the receiver/smartphone, [54], [103] (trilateration) or the angles at which signals from two or more APs are received [102], [104] (triangulation). Such techniques are often prone to Radio Frequency (RF) interference and malicious node-based attacks. Some work has been done to overcome these vulnerabilities through online evaluation of signals and packets [105]. However, these indoor localization frameworks are inherently not resilient to multipath effects, where the RF signal reaches a destination after being reflected across different surfaces, and shadowing effects, where the RF signal fades due to obstacles. Some recent work has investigated multipath effects for triangulation [56], but these works do not apply to commodity smartphones (expected to be the de-facto portable device for indoor localization) and hence, have limited applicability.

On the other end of the spectrum are fingerprinting based methods that associate selected indoor locations (reference points) with a unique RSSI (Received Signal Strength Indicator) signature obtained from APs accessible at that location [37], [38] (fingerprinting is discussed in more detail in section 5.1). These techniques have proven to be relatively resilient to multi-path

reflections and shadowing, as the reference point fingerprint captures the characteristics of these effects, leading to improved indoor localization. However, fingerprinting requires a more elaborate offline-phase (i.e., setup) than triangulation/trilateration methods, where RSSI fingerprints need to be captured across indoor locations and stored in a fingerprint database, before being able to support localization or navigation (by referring to the database) in the online-phase, in real-time.

Fingerprinting-based techniques are not only vulnerable to interference and malicious node-based attacks but are also prone to database corruption and privacy/trust issues (discussed in the next section). Amongst the mentioned vulnerabilities, RSSI interference and malicious node or AP attacks are significantly easier to perform as they only require the attacker to gain physical access into the indoor location where the attack needs to take place. Once the attacker is at the site, they could, for instance, deploy battery powered AP units that would either interfere with the localization AP signals or spoof valid AP nodes. Moreover, a single malicious AP unit is capable of spoofing multiple packets for multiple valid APs in the area.

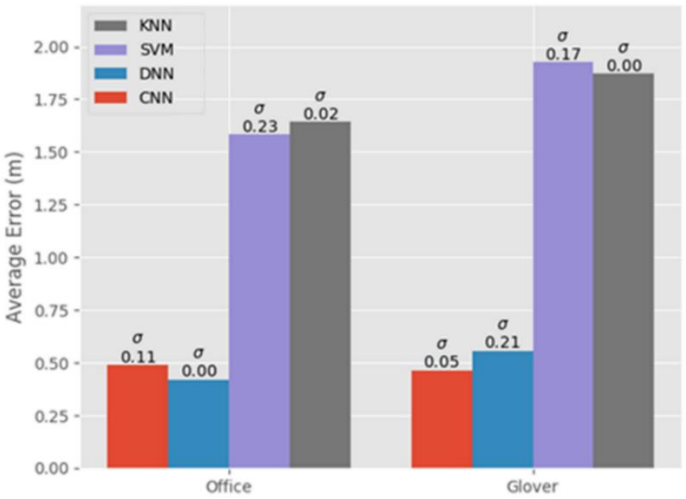


Figure 33. Average indoor localization error (in meters). Fingerprinting techniques based on deep neural networks (DNNs), convolutional neural networks (CNNs), support vector machines (SVM), and k-nearest-neighbor (KNN). Results are shown for two different indoor paths.

Simple fingerprinting-based indoor localization frameworks that use techniques such as KNN (k-nearest-neighbor) can utilize outlier detection-based techniques to overcome some security issues [106]. However, recent work on improving WiFi fingerprinting accuracy has tended to exploit the increasing computational capabilities of smartphones and utilize more powerful machine learning techniques. For instance, sophisticated convolutional neural networks (CNNs) [38] have been proposed and shown to improve fingerprint-based indoor localization accuracy on smartphones. Figure 33 shows the improvements when using CNN and deep neural network (DNN) [82] [107] [108] based localization approaches as compared to more traditional techniques such as KNN [37] and support vector machines (SVM) [92]. Based on the improvements achieved through CNN- and DNN-based algorithms, indoor localization solutions in the future are expected to benefit from the use of deep learning methodologies that have the potential to significantly reduce localization errors. However, to date, no studies have been performed to assess the impact on accuracy for malicious AP attacks on deep learning based indoor localization.

In this chapter, we present a novel method to overcome the security vulnerabilities of deep learning based indoor localization frameworks. We use the recent deep learning-based localization framework from [38] as an example and propose security enhancements for it. The novel contributions of our work are:

- We identify and model various AP-based attacks that impact the localization accuracy of deep learning-based indoor localization frameworks, such as the frameworks from [38] and [82];

- For the first time, we conduct an in-depth experimental analysis on the impact of AP-based attacks on CNN [38] and DNN [82] based indoor localization frameworks across indoor paths;
- We present a novel methodology for constructing AP attack resilient deep learning models to create a secure version of the CNNLOC framework from [38] (which we call S-CNNLOC) for robust and secure indoor localization;
- We compare the performance of S-CNNLOC against CNN-LOC for a varying number of malicious AP nodes, and across a diverse set of indoor paths.

5.1. BACKGROUND AND RELATED WORK

5.1.1. RECEIVED SIGNAL STRENGTH INDICATOR (RSSI)

RSSI is a measurement of the power of a received radio signal transmitted by a radio source. The RSSI is captured as the ratio of the received power (P_r) to a reference power (P_{ref}, usually set to 1mW). The value of RSSI is reported in dBm and is given by:

$$RSSI\ (dBm) = 10 \cdot log \frac{P_r}{P_{ref}} \tag{4}$$

The received power (P_r) is inversely proportional to the square of the distance (d) between the transmitter and receiver in free space and is given by:

$$P_r = P_t \cdot G_t \cdot G_r \left(\frac{\lambda}{4\pi d}\right)^2 \tag{5}$$

where P_t is the transmission power, G_t is the gain of transmitter, $G_r$is the gain of receiver, and λ is the wavelength. This inverse relationship between the received power and distance has

often been used by researchers to localize wireless receivers with respect to transmitters at known locations, e.g., estimating the location of a user with a WiFi capable smartphone from a WiFi AP. However, the free space models based on equations (4) and (5) do not extend well for practical applications. In reality, the propagation of radio signals is influenced by various effects. Figure 34 illustrates some of these effects as a radio signal travels from its source (WAP2) towards location (L2). The signals transmitted from WAP2 get scattered at the edges of the pillar, reflect off walls, and get attenuated as they pass through the pillar to reach the reference point L2. Also, the signals from WAP2 follow different paths (called multipath traversal) to reach location L2. These effects lead to an RSSI reading at L2 that does not correspond to equation (5) which was designed to function in free space.

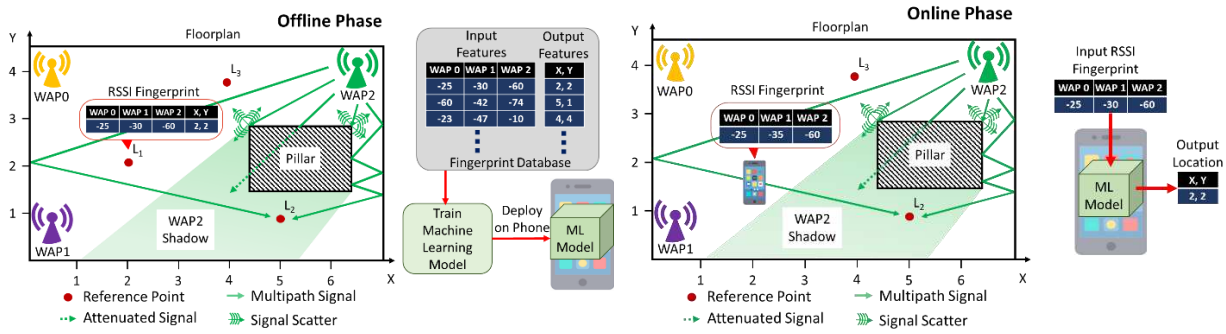


Figure 34. A representation of the offline and online phases in the fingerprinting process for indoor localization, for a given floorplan.

5.1.2. FINGERPRINT-BASED INDOOR LOCALIZATION

Since the first efforts on fingerprinting-based indoor localization about two decades ago, such as with the work in RADAR [81], a significant level of advancement has been achieved in this area. However, the general premise of fingerprinting based indoor localization has remained unchanged. As shown in Figure 34, fingerprinting-based localization is carried out in two phases. In the first phase (called the offline or training phase), the RSSI values for visible WiFi APs

(WAPs) are collected for a given floorplan at reference points L1, L2, L3 etc. identified by some coordinate system. The RSSI fingerprint captured at a given reference point consists of RSSI values (in dBm) for the WAPs in the vicinity and the X-Y coordinate of the reference point. The resulting database of location-tagged RSSI fingerprints (Figure 34) is then used to train models (e.g., machine learning-based) for location estimation such that the RSSI values are the input features, and the reference point location coordinates are the target (output) features. The trained machine learning model is then deployed to a mobile device as shown in the offline phase of Figure 34. In the second phase (called online or testing phase), the devices are used to predict the (X-Y coordinate) location of the user carrying the device, based on real-time readings of WAP RSSI values on the device. Contrary to the supervised learning approach discussed so far, some recent work also explores adapting semi-supervised deep reinforcement learning to deliver improved accuracy when very limited fingerprinting data is available in the training phase [109]. One of the major advantages of using fingerprinting-based techniques over other methods (e.g., trilateration/trilateration) is that knowledge of environmental factors such as multipath signal effects and RF shadowing are captured within the fingerprint database (such as for the reference point L2 in Figure 34) in the offline phase and thus leads to improved localization accuracy in the online phase, compared to other methods.

An important aspect of fingerprinting-based indoor localization is the choice of the signal-source utilized. Some commonly used signal-source options include Ultra-Wide-Band (UWB) [70], Bluetooth [110], ZigBee [111], and WiFi [37]. The choice of signal directly impacts the achievable localization accuracy as well as the associated setup and maintenance costs. For example, UWB APs need to be specially purchased and deployed at the target site, however, they have been shown to deliver a higher level of accuracy than many other signal types. On the other

hand, WiFi based indoor localization frameworks have gained traction due to the ubiquitous availability of WiFi access point (WAPs) in indoor locales and the fact that most people nowadays carry smartphones that come equipped with WiFi transceivers, making WAP-based indoor localization a cost-effective and popular choice [37], [38]. For this reason, in our work, we assume the use of WAPs as signal sources for fingerprinting-based indoor localization.

5.1.3. CHALLENGES WITH INDOOR LOCALIZATION

As a result of the popularity of WiFi fingerprinting, efforts in recent years have been made to overcome its limitations, such as energy-efficiency [37], variations due to device heterogeneity [32] [40] [66], and temporal degradation effects on localization accuracy [112]. However, in recent years as indoor localization services are beginning to be prototyped and deployed, researchers have raised concerns about the privacy, security, and other vulnerabilities associated with fingerprinting-based localization. Some commonly identified vulnerabilities and their mitigation strategies are discussed in the rest of this section.

Offline-Phase Database Security: The indoor localization fingerprint database consists of three pieces of information in each entry of the database: WAP Media Access Control (MAC) addresses, RSSI values of these WAPs, and the associated reference point location tag (e.g., XY co-ordinate of a location). A malicious third-party, may corrupt the database by changing the RSSI values associated with the MAC addresses or change the location where the samples were taken. This kind of an attack can completely jeopardize the functionality of an indoor localization framework, as the offline database holds the most crucial information required for any fingerprinting-based indoor localization framework to function. To mitigate such issues, researchers have proposed techniques such as outlier detection-based identification of corrupted

information [101], [102] and performing continuous sanity checks on the database using checksums [113]. Alternatively, even if the attackers are able to read the database, they can use the information such as reference point locations and WAP MAC addresses to launch other forms of attacks, as discussed next.

User Location Privacy: Some recently proposed indoor localization techniques exploit resource intensive machine learning models that need to be executed on the cloud or some other form of remote service, instead of the user's mobile device. These frameworks may compromise the user's privacy by either intentionally or unintentionally sharing the user's location with a third party. The leaked location and background information from one user can then be correlated to other users for their information [114]. However, recent advances have been able to optimize the execution of complex machine learning models on resource constrained mobile devices such that the location prediction computation does not need to be offloaded to the cloud or other types of remote services [38].

AP Jamming or Interference: An attacker may deteriorate the quality of localization accuracy in a specific region indoors by placing signal jammers (narrow band interference) in the vicinity [115], [105]. The jammer can achieve this goal by emitting WiFi signals to fill a wireless channel, thereby producing signal interference with any non-malicious WAPs on that channel. Alternatively, the jammer can also continuously emit WiFi signals on a channel such that legitimate WAPs never sense the channel to be idle and therefore do not transmit any information [116]. Such an attack may cause a mobile device to lose visibility of WAPs, reducing localization accuracy or preventing localization from taking place altogether.

Malicious AP Nodes or Spoofing: In this mode of attack, a malicious third-party places one or more transmitters at the target location to spoof the MAC address of valid WAPs used by the

fingerprinting-based localization framework. The MAC address could have been obtained by a person capturing WiFi information while moving in the target area. Alternatively, this information could have been leaked through a compromised fingerprint database. Also, the behavior of the malicious nodes in each case may change over time. The detection of spoofing-based attacks is also an active area of research in the robot localization domain. Approaches proposed include the empirical analysis of data collected at a post-localization phase [117] and using machine learning [118]. However, both works solely focus on detecting a spoofing attack either in real-time or offline. Techniques such as the one presented in [119] allow for the identification of malicious nodes using linear regression on data collected over a certain period of observation time. However, any delay in the mitigation of WAP-based attacks in real-time would leave the indoor localization framework vulnerable and may lead to tainted predictions, thereby disrupting the localization services or giving the attacker a window of opportunity.

Environmental Alterations: Changes or alterations in the indoor environment can induce unpredictable changes to the WAP-based fingerprints in the online phase. Such alterations could include moving furniture or machinery, or renovations in the building. Crowdsourcing-based techniques, e.g., [120], that update fingerprints on-the-fly may be more resilient to such effects, given that ample number of (crowd-sourced) fingerprint samples are collected in the area where the changes took place. However, deep learning based techniques may need to be retrained to accommodate for the changes, which may take several hours and thus be impractical for real-time adaptation.

From the discussion in this section, one observation is that launching attacks, such as jamming and spoofing, is relatively easy if the attacker is able to access the indoor location. Given the recent interest in deep learning-based fingerprinting to improve indoor localization accuracy

[38], [82], [109] there is a critical need to analyze and address security vulnerabilities for such solutions. However, to date, no prior work has explored the impact of malicious AP-based attacks on the accuracy and reliability of deep learning based indoor localization frameworks. Our goal in this work is to show, for the first time, how deep learning-based indoor localization frameworks such as CNNLOC [38] can be vulnerable to malicious AP-based attacks and further propose a methodology to address such vulnerabilities without loss in localization accuracy, on commodity mobile devices.

5.2. CNNLOC FRAMEWORK OVERVIEW

5.2.1. CONVOLUTIONAL NEURAL NETWORKS

Convolutional neural networks (CNNs) are a form of deep neural networks that are specially designed for image classification. They have been shown to deliver significantly higher classification accuracy as compared to conventional DNNs due to their enhanced pattern recognition capabilities. Note that from this point onward we use the term DNN to identify deep learning models that do not consist of convolutional layers. As shown in Figure 35, a CNN model has three main functional components or layers: convolution+ReLU (Regularized Linear Unit), pooling, and fully connected layers. The CNN model learns patterns in images by focusing on small sections of the image, known as a frame, from the input layer. The frame moves over a given image in small strides. Each convolutional layer consists of filter matrices that hold weight values. In the first layer, convolutional operations (dot products) are performed between the current input frame and filter weights followed by the ReLU activation function. The pooling layer is responsible for down sampling the output from a convolution+ReLU unit, thereby reducing the computational requirements by the next set of convolution layers. The final classification is

performed using a set of fully connected layers that often utilize a SoftMax activation function to calculate the probability distributions for various classes. In the testing phase of a CNN model, the class with the highest probability is the output prediction. Further details on the design of CNNs can be found in [38] and [89].

5.2.2. INDOOR LOCALIZATION WITH CNNLOC

The CNNLOC indoor localization framework [38] consists of two major components in the offline phase. The first component involves capturing the RSSI fingerprints for different locations, and then converting each RSSI fingerprint vector that is tied to a location (reference point) into an image tied to the same location. The second component of the offline phase is the training of a CNN model using the images created previously. In the online phase, the same process is used to create an image (based on observed RSSI values), which is fed into the trained CNN model for location prediction.

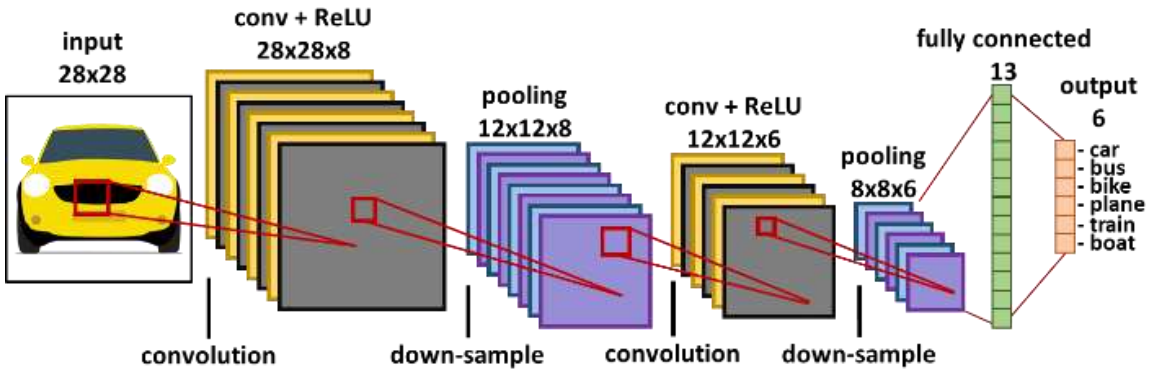


Figure 35. A general representation of the various components of a convolutional neural network (CNN).

A simplified overview of the process of converting an RSSI fingerprint vector into an image is shown in Figure 36. The RSSI vector consists of RSSI values in the range of -100 to 0 dBm (low

signal strength to high signal strength). These values are normalized to a range of 0 to 255, which corresponds to the pixel intensity on the image. The dimensions of the RSSI image are set to be the closest square to the number of visible WAPs on the path. For example, in Figure 36, the RSSI vector has a size of 8, and the closest square would have 9 pixels in it, therefore, the dimensions of the image are set to 3×3. A pixel with zero intensity is padded at the end to increase the size of the vector as shown in Figure 36. The generated image then becomes a part of the offline database of images used to train a CNN. In the online phase, this same process of image creation is used with the RSSI vector observed by the user at any location, and the resulting image is fed to the trained CNN model to get a location prediction. It is important to note that in the online phase of CNNLOC, the input image will always remain the same size as in the offline phase, such that each pixel in the image corresponds to specific MAC IDs. In case a specific MAC ID observed in the offline phase is no longer visible in the online phase, the pixel value corresponding to that MAC ID is set to zero.

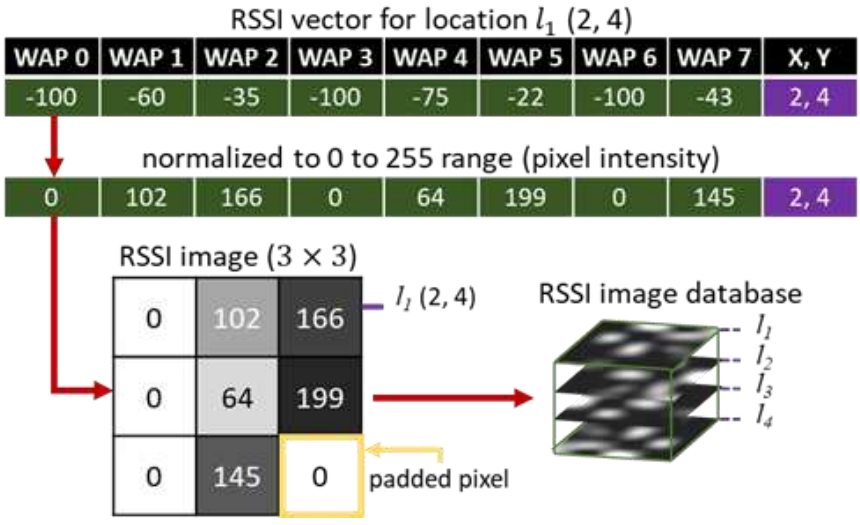


Figure 36. A simplified overview of the conversion of an RSSI fingerprint to an image in the CNNLOC indoor localization framework.

5.3. LOCALIZATION SECURITY ANALYSIS

In this section, we perform a WAP RSSI vulnerability analysis on the deep learning-based indoor localization frameworks presented in [38] (CNNLOC) and [82] (which uses DNNs). For this study, we modeled the two deep learning frameworks and contrasted their performance for the two indoor paths shown in Figure 37. The Office and Glover paths in the figure are 64 and 88 meters long and the reference locations used to capture WiFi RSSI are marked by blue dots. A detailed discussion on the salient features of these and other indoor benchmark paths we consider can be found in the experiments section (Section 5.6). We used an HTC U11 smartphone [121] to capture WiFi fingerprints along the indoor paths and test for localization accuracy.

A WAP-based security attack may include either WAP spoofing or WAP jamming. To establish the impact of such WAP-based attacks on localization accuracy, we must identify the behavior of the WiFi RSSI fingerprints in the presence of one or more malicious WAP nodes (WiFi spoofers/jammers). In our experience, the tainted fingerprint in the online phase will exhibit one of three behaviors: 1) the RSSI values from one or more visible WAPs exhibits a significant increase or decrease as compared to its offline counterpart, 2) a WAP whose RSSI value is usually not visible at the current reference point becomes visible, and 3) a WAP that is usually visible at the current reference point is no longer visible. As the range of received RSSI values from WAPs is between -100 to 0 dBm, the impact of the malicious WAP behavior on the fingerprints is to induce fluctuations in WAP RSSI values within this range, for the impacted fingerprints.

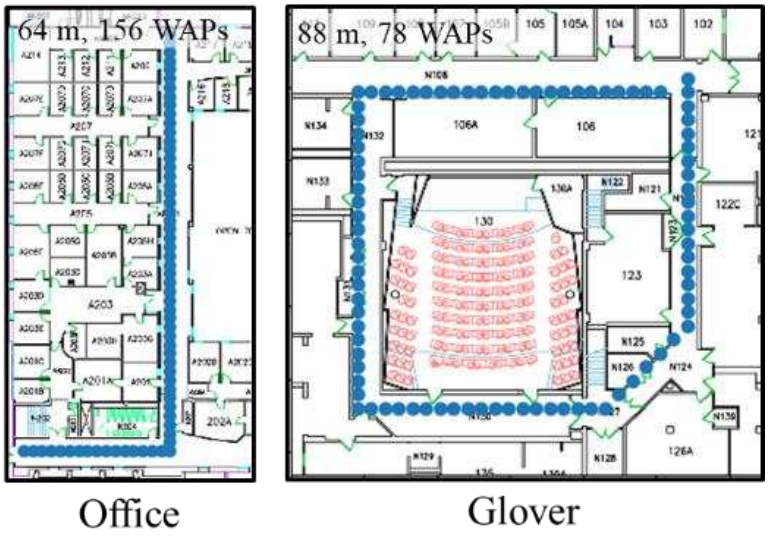


Figure 37. Two indoor benchmark paths (Glover and Office) with reference points denoted by blue markers. The path lengths and WiFi densities are denoted at the top of the maps.

Figure 38 shows the fingerprint images generated using an RSSI fingerprint based on the methodology described in CNNLOC [38]. Each image has a resolution of 9×9. The original RSSI vector (fingerprint) consists of 78 WAP values and is presented in its image form in Figure 38(a). This image (Figure 38(a)) is not tainted by malicious WAPs (mWAPs) in the surrounding area, and therefore is labeled as "mWAP0". The image labeled "mWAP2" (Figure 38(b)) is generated for the case when two WAPs out of 78 are malicious WAPs that generate spurious signals between -100 dBm to 0 dBm (their impact can be clearly seen with the two non-white pixels on the bottom half of the image). Similarly, Figure 38(c)-(f) show the generated images when the number of malicious WAPs is increased to 4, 6, 8, and 10, respectively. For most of these images, the tainted pixel values can be visually identified, and simple image local smoothing filters [122] may be applied to remove them. However, such filtering is not always possible. For instance, in Figure 38(d) with 6 malicious WAPs, we observe only 5 tainted pixels that are visually decipherable as compared to the untainted image (Figure 38(a)). This is because the sixth noisy pixel is a very minor disturbance that is hard to detect visually. Unfortunately, the datapoint represented by this

sixth pixel can have a significant impact on localization accuracy. Such scenarios also exist for the case of mWAP8 (Figure 38(e)) and mWAP10 (Figure 38(f)).

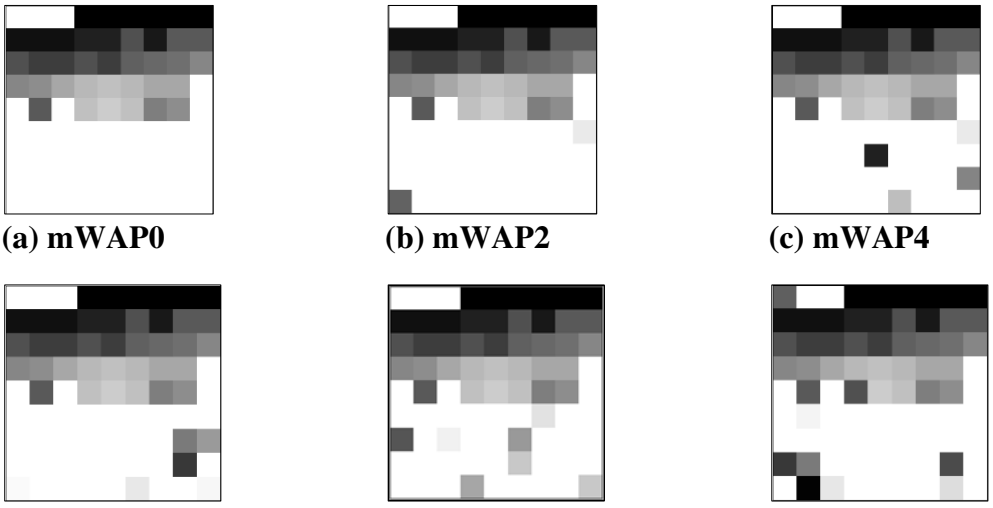


Figure 38. Fingerprint images generated from RSSI vectors using the methodology described in CNNLOC; (a) represents the "mWAP0" fingerprint image that should be ideally generated when the initial RSSI vector is not tainted by a malicious WAP (mWAP=0); (b)-(f) show fingerprint images in the presence of different number of malicious WAPs. The label "mWAPX" indicates X malicious WAPs, which introduce fluctuations in the RSSI values of the pixels corresponding to these WAPs.

To test the vulnerability of deep learning-based indoor localization frameworks in the presence of malicious WAPs, we analyzed the impact of a varying number of malicious WAPs on the localization accuracy of a CNN-based [38] and a DNN-based [82] indoor localization framework. The results of this experiment are shown in Figure 39. We captured the average indoor localization error for the Office and the Glover paths (shown earlier in Figure 37) for an increasing number of malicious WAP nodes (along the x-axis). For a scenario with malicious WAPs (e.g., mWAP = 1), we randomly selected the location of the malicious WAP over 100 trials and averaged the resulting localization error. From Figure 39, we observe that the average localization error of both CNN and DNN learning models increases monotonically in a majority of cases. The results

highlight the vulnerability of deep neural network based indoor localization models towards WAP-based attacks. Also, the CNN model for both paths is somewhat more vulnerable to malicious WAP-based attacks as compared to the DNN model. One possible explanation for this may be that CNN models are more sensitive to changes in patterns in the image as compared to variations across RSSI value inputs for the DNN model.

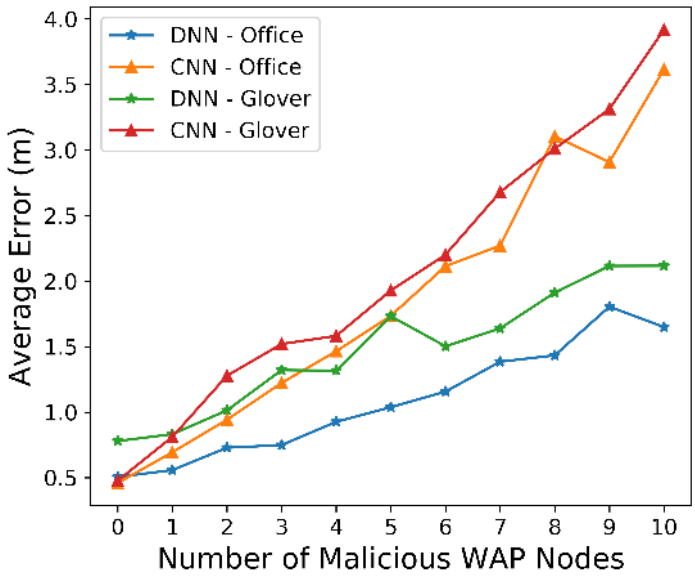


Figure 39. Results for the impact of malicious WAPs on deep learning model accuracy on the Office and Glover paths. Average localization error for the CNN [38] and DNN [82] localization frameworks is shown for an increasing number of malicious WAPs.

To further analyze the accuracy degradation of these deep learning models, we present the worst-case localization error for the two deep learning models in Figure 40. We can observe that the worst-case localization errors for DNN and CNN models are significantly higher than the average errors shown in Figure 39 as the number of malicious WAPs are increased. With only 1 malicious WAP, the localization error in the worst case can be higher by up to 20× for both paths and deep learning models. The worst-case localization error for the CNN model goes above 50

meters with only 6 malicious WAPs for the Glover path, which would put a user's predicted location at a completely different area on an indoor floorplan! The DNN model appears to be much more significantly impacted than the CNN model when it comes to worst case localization error.

From these experiments, it can be concluded that deep learning based indoor localization frameworks are highly vulnerable to WAP-based attacks. There is thus a strong motivation to improve attack resilience for these frameworks, to achieve both robust and high accuracy indoor localization. Even though DNN and CNN based models used for our experiments in this section produce a relatively similar level of degradation in localization accuracy, in the rest of the chapter we focus on addressing vulnerabilities for indoor localization systems that utilize CNN models. This is because CNNs have several advantages over DNNs when used for localization. A drawback of DNN models is that their computational complexity increases significantly with increase in hidden layers, which is not the case for CNN models [35]. The pooling layers in CNN models reduce the overall footprint after each convolutional layer, thereby reducing the computation required by the successive set of layers. Therefore, localization solutions that utilize CNN models instead of DNN models are inherently more scalable and energy-efficient [89]. Also, CNN models are better at identifying patterns in image data than DNNs, which make CNNs a more viable solution to overcome device heterogeneity issues (that are more readily apparent in image form) with indoor localization when using mobile devices [48].

The new observations and related discussions in this section highlight the importance of securing deep learning models against WAP-based attacks and serve as the motivation for our proposed security enhancements in this work, that aim to secure deep learning models used for indoor localization. We discuss the specific attack models and associated assumptions made in our work in the next section.

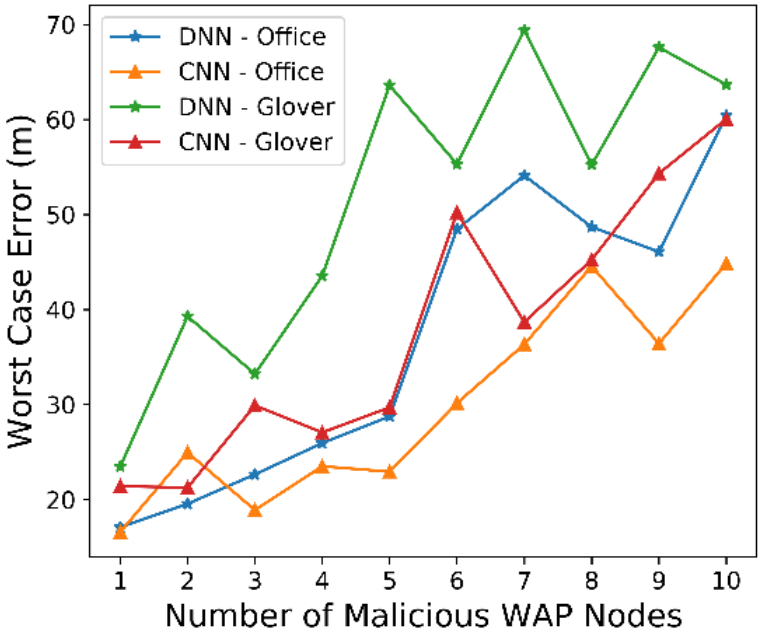


Figure 40. Worst-case localization error for CNN and DNN, with respect to increasing number of malicious WAPs on the Office and Glover paths.

5.4. PROBLEM FORMULATION

We now describe our problem objective and the assumptions associated with establishing a secure (WAP RSSI attack resilient) CNN-based indoor localization framework called Secure-CNNLOC (S-CNNLOC). The assumptions for our framework are:

- The offline fingerprint sampling process is carried out in a secure manner such that the collected fingerprints only consist of trusted non-malicious WAPs.

- The offline generated fingerprint database is comprised of images, each with a tagged reference point location; this database is stored at a secure, undisclosed location.

- A CNN model is trained using the offline fingerprint database and is encrypted and packaged as a part of an indoor localization app that is deployed on mobile devices.

- Once the localization app is installed by a user, the CNN model can only be accessed by that app.

- As the user moves about an indoor path, their mobile device conducts periodic WiFi scans; and the localization app translates the captured WiFi RSSI information into an image.

- The generated image is fed to the CNN model within the localization app on the mobile device, and the user's location is updated in real-time on a map displayed on the device.

- The process of WiFi scanning, fingerprint to image conversion, and location prediction continues until the user quits the localization app on their mobile device.

We make the following assumptions about the indoor environment:

- An attacker can physically access one or more of the indoor locales and paths in the online phase for which the indoor localization framework has been trained and set-up.

- The attacker can carry a smartphone equipped with WiFi or any other portable battery powered WiFi transceiver to capture data about WiFi access points (WAPs).

- The offline generated fingerprint database is secured and cannot be accessed by any malicious third party.

- It is generally known (to the attacker) that the indoor localization framework utilizes a deep learning-based approach, such as CNNs, to predict a user's location.

- The attacker is capable of conducting the analysis described in the previous section and place malicious WAP nodes at any randomly chosen locations along the indoor paths or locales that are being targeted for a service disruption attack.

- The attacker can walk about an indoor path and collect WiFi fingerprints while capturing steps taken and walking direction data, similar to the approach described in [123]; this would allow anyone with a smartphone to create their own fingerprint database which can be used to more strategically place WiFi jammers or spoofed WAPs as discussed in earlier sections.

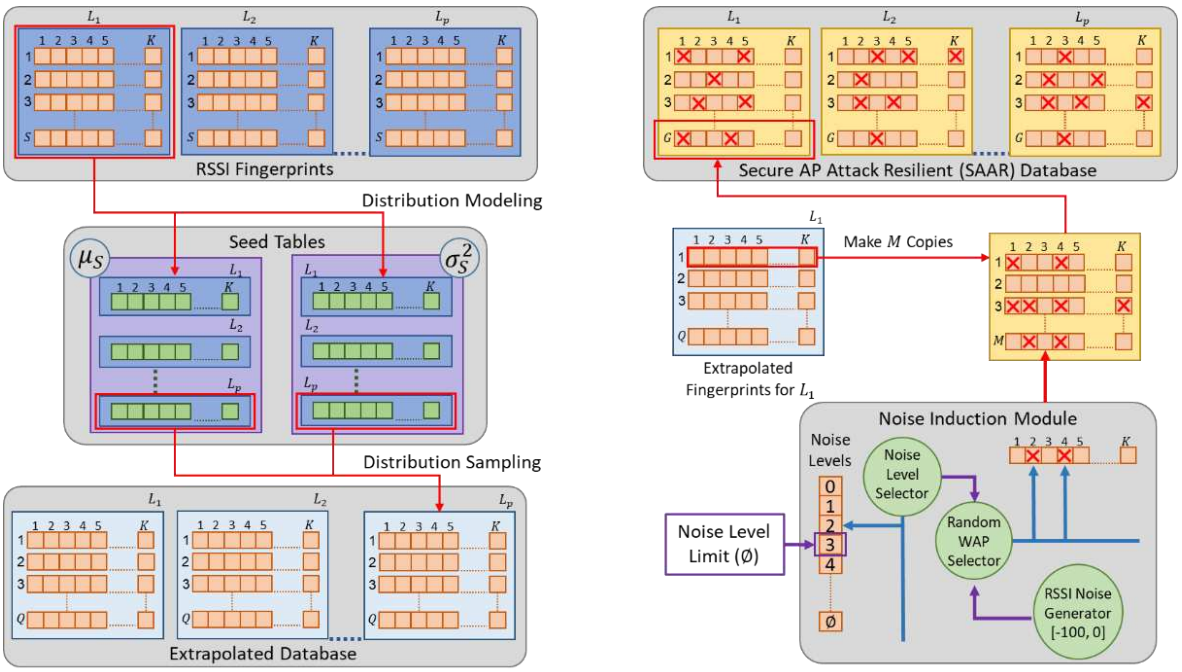
Problem Objective: Given the above assumptions, our objective is to create a secure CNN-based indoor localization framework (called S-CNNLOC) that is deployed on mobile devices and is resilient to malicious WAP RSSI attacks, by minimizing their impact on the localization accuracy at run-time (i.e., in the online phase).

5.5. S-CNNLOC FRAMEWORK

In this section, we discuss the design of our S-CNNLOC framework to overcome the vulnerability of the CNNLOC [38] indoor localization framework against malicious WAP-based jamming and spoofing attacks in indoor environments.

5.5.1. OFFLINE FINGERPRINT DATABASE EXTRAPOLATION

One of the major limitations of the CNNLOC framework comes from the small number of offline fingerprints considered per reference point (10 fingerprints in [38]). In general, deep learning models often require a large number of samples per class to produce good results. However, capturing WiFi fingerprints in any indoor localization framework is a time-consuming manual endeavor that is quite expensive to scale in volume (in terms of samples per reference point).



(a) **(b)**

Figure 41. An overview of the offline extrapolation of RSSI fingerprints and noise induction in the extrapolated fingerprints. The noisy and extrapolated set of RSSI fingerprints are converted into images and used to train the CNN model in our proposed S-CNNLOC framework.

To overcome this limitation, in our S-CNNLOC framework, we extrapolate the offline fingerprint database such that we obtain a larger number of samples per reference point. An overview of this process is presented in Figure 41(a). We sample a total of S RSSI fingerprints at each location (reference point) from L1 to LP, such that the RSSI vector has K WAPs (i.e., vector size is K). The complete set of fingerprints that are manually collected at P locations become the offline fingerprint database. The distribution of each WAP RSSI at a given location is modeled by their means and variances. This step is repeated for each reference point in the offline fingerprint database. The mean and standard deviations along with the reference location information are temporarily stored in tabular forms and are referred to as the seed tables (Figure 41(a)). The seed tables can be represented as:

$$\mu_{S(i,j)}, \sigma^2_{S(i,j)}, \qquad i \in [1, K], j \in [1, P] \tag{6}$$

where $\mu_{S(i,j)}$ and the $\sigma^2_{S(i,j)}$ are the tables that contain the means and variances of S WAP RSSIs for each location. These mean and variance seed tables (also shown in Figure 41(a)) can now be used to extrapolate a larger fingerprint database.

To generate a new offline fingerprint for a given reference point, the normal distribution based on the mean and variance (from the seed tables) for each WAP RSSI in each reference point fingerprint is randomly sampled Q times:

$$RSSI_{(i,j)} \sim N\left(\mu_{S(i,j)}, \sigma^2_{S(i,j)}\right) \forall\, i \in [1, K], j \in [1, P] \tag{7}$$

where *RSSI(i, j)* is the RSSI in dBm of the i^{th} WAP at the j^{th} reference point; and N represents the normal distribution. By randomly sampling each WAP from the reference point in seed tables, we generate Q new RSSI fingerprint vectors for the given reference point. Through this random sampling-based data extrapolation approach, we capture different combinations of RSSI values in a fingerprint and also scale the size of our offline dataset beyond the few samples that were collected in the offline phase. The complete set of Q RSSI vector fingerprints per reference point is the extrapolated fingerprint database, as shown in Figure 41(a). Subsequently, the extrapolated fingerprint database is fed to the next stage where we deliberately induce noise in the fingerprints in the database, as discussed next.

5.5.2. MALICIOUS BEHAVIOR INDUCTION

From our analysis of CNN-based indoor localization in section 5.3, we observed that fluctuations in one individual pixel value of the WiFi fingerprint image can lead to significant deterioration in the localization accuracy. This behavior can be attributed to the fact that the trained

CNN model is only good at making predictions for images (or RSSI information) that it has previously seen. Therefore, the CNNLOC framework becomes vulnerable to minor deviations or noise in the images that can be induced by WAP-based attacks or WiFi jammer attacks in the online phase, when the trained CNN model is used for location inference.

CNN models are designed to recognize one or more patterns within images that may be very different from each other, or may only have slight differences from each other. In our approach, we conjecture that relatively small-scale variations within and between images constructed from WAP RSSI values (for the purpose of pattern recognition for indoor localization) can be learned to be ignored by a CNN model. One way to accomplish this is by integrating an image filter with the CNN prediction model. A recent work [124] has shown how a salt and pepper noise filtering technique can provide some noise resilience for general image processing with CNNs. A separate set of convolutional layers are used in [124] whose sole purpose is to denoise an image. However, such an approach would be extremely inefficient for our problem as it would require using two different CNNs: one for denoising and another for classification, which would increase prediction time. Moreover, using an additional CNN would increase the memory footprint of our framework, which is a big concern for resource-constrained mobile devices.

We propose to use a single CNN-model for both image denoising and classification. Based on our analysis presented in section 5.3, we decide to conceptually model malicious behaviors such as WAP spoofing, WAP jamming, and even environmental changes as random fluctuations in the fingerprint data and expect the CNN model to be resilient to such fluctuations. Thus, by a calculated introduction of noise in the input dataset that is used in the training phase of the CNN model, we hope to teach the model to learn to ignore noise (due to malicious WAPs) in the inference phase. Towards this goal, as shown in Figure 41(b), for each fingerprint in the “clean”

(mWAP0) extrapolated database generated as discussed in the previous sub-section, M copies are constructed in a separate table. Then each of the M fingerprint vectors are fed to the proposed noise induction module that introduces random fluctuations in the WAP RSSI values, based on an upper limit (Ø) that is set by the user. The noise induction module (Figure 41(b)) has three major components. For a given RSSI vector, the noise level selector submodule picks values from a discrete uniform distribution such that$\theta \sim U\{0, \emptyset\}$, where "θ" is the number of WAPs in the RSSI vector whose RSSI value would be altered by the noise induction module. The random WAP selector arbitrarily identifies the set of WAP candidates "W_θ", where each WAP candidate "w_c" is picked to be between 1 to K as described by the expression:

$$\begin{gathered} w_c \sim U\{1, K\}, \quad c \in [1, \theta] \\ s.t., W_\theta = \{w_1, w_2, w_3 \ldots w_\theta\} \end{gathered} \tag{8}$$

The newly generated RSSI vectors ($RSSI_{(i,j)}^{Noisy}$) are tainted by random noise at the ith WAP position, if the WAP was chosen by the random WAP selector submodule as shown by equation (9):

$$\begin{gathered} RSSI_{(i,j)}^{Noisy} = \begin{cases} I, & if\ i \in W_\theta \\ RSSI_{(i,j)}, & otherwise \end{cases} \\ j \in [1, P], I \sim U\{-100, 0\} \end{gathered} \tag{9}$$

where *I* represents noise sampled from a discrete uniform distribution between -100 dBm to 0 dBm, *RSSI(i, j)* is the clean (untainted) RSSI from equation (9) and P is the number of reference points on a benchmark path for which fingerprint data has been collected. Thus, our proposed approach generates RSSI vectors that may have up to Ø noise-induced RSSI WAP values. Having a uniform distribution of 0 to Ø malicious WAPs ensures that the CNN model trained using the

generated data is resilient to a range of malicious WAP numbers and locations in the localization environment in the testing phase.

Following this process for all fingerprints in the clean training database, we generate G=Q×M fingerprints per reference point. The final number of RSSI fingerprints in the secure AP attack resilient (SAAR) database constructed by following the processes described in this section is G×P, where P is the number of reference points on a benchmark path. The SAAR training database is then used to train the CNN model which is subsequently deployed as an app on a mobile device and used to make online (real-time) location predictions for the user carrying the mobile device.

5.6. EXPERIMENTS

5.6.1. EXPERIMENTAL SETUP

We initially compare the accuracy and stability of our proposed (S-CNNLOC) framework to its vulnerable counterpart (CNNLOC [38]) using two benchmark paths. These paths are shown in Figure 37 with each fingerprinted location (reference point) denoted by a blue marker. The paths were selected due to their salient features that may impact location accuracy in different ways. The 64-meter Office path is on the second floor of a relatively recently designed building with a heavy use of wood, plastics, and sheet metal as construction materials. The area is surrounded by small offices and has a total of 156 WAPs visible along the path. The Glover path is from a very old building with materials such as wood and concrete used for its construction. This 88-meter path has a total of 78 visible WAPs and is surrounded by a combination of labs (heavy metallic equipment) and classrooms with open areas (large concentration of users).

In the offline phase for S-CNNLOC, a user carried the HTC U11 smartphone and traversed the path with reference points at 1-meter intervals and captured 10 WiFi scans at each reference point, storing the scanned values tagged with the corresponding reference point location data. The fingerprint sampling and storage methodology within the smartphone is similar to that described in CNNLOC [38]. The trained S-CNNLOC model was deployed as an Android app on the HTC U11 smartphone. The values of Q and M are set to 100 and 10 respectively. Based on these values of Q and M, the Office path has 64000 samples and the Glover path has 88000 samples. To study the impact of malicious WAPs on indoor localization performance, we used a real WiFi transceiver [125] to induce interference (from spoofing/jamming) and obtain "tainted" RSSI values in the vicinity of the indoor paths. These values were observed in the online phase. For some of our scalability studies where we consider the impact of multiple malicious WAPs, multiple such transceivers were considered, to generate multiple "tainted" RSSI values.

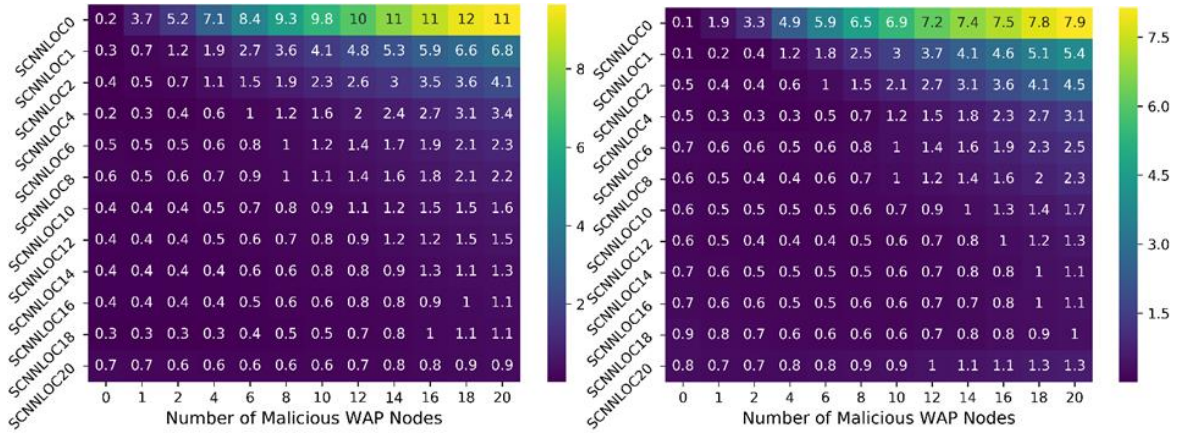
5.6.2. EXPERIMENTAL RESULTS

5.6.2.1. ANALYSIS OF NOISE INDUCTION AGGRESSIVENESS

We first performed a sensitivity analysis on the value of Ø (upper limit of noise induction; discussed in Section 5.2). Several CNN models were trained: S-CNNLOC1 (Ø = 0; no malicious WAPs), S-CNNLOC2 (Ø = 1), up to S-CNNLOC20 (Ø = 20), using the fingerprint data collected during the offline phase. Then the devised models were tested with fingerprints observed along the indoor paths in the online phase, in the presence of different numbers of malicious WAPs.

Figure 42 shows the heatmap for the mean localization errors (in meters) with annotated standard deviation of various scenarios on the Office path (Figure 42(a)) and the Glover path (Figure 42(b)). The y-axis shows various S-CNNLOC variants with different values of Ø varying

from 1 to 20. The x-axis shows the number of malicious nodes (mWAPs) present in the online phase. In Figure 42, the bright yellow cells of the heatmap, with higher annotated values, represent an unstable and degraded localization accuracy whereas the darker purple cells, with lower annotated values, represent stable and higher levels of localization accuracy. Each row of pixels in the heatmaps of Figure 42(a) and (b) represents the vulnerability of the specific S-CNNLOC model to an increasing number of mWAP nodes.



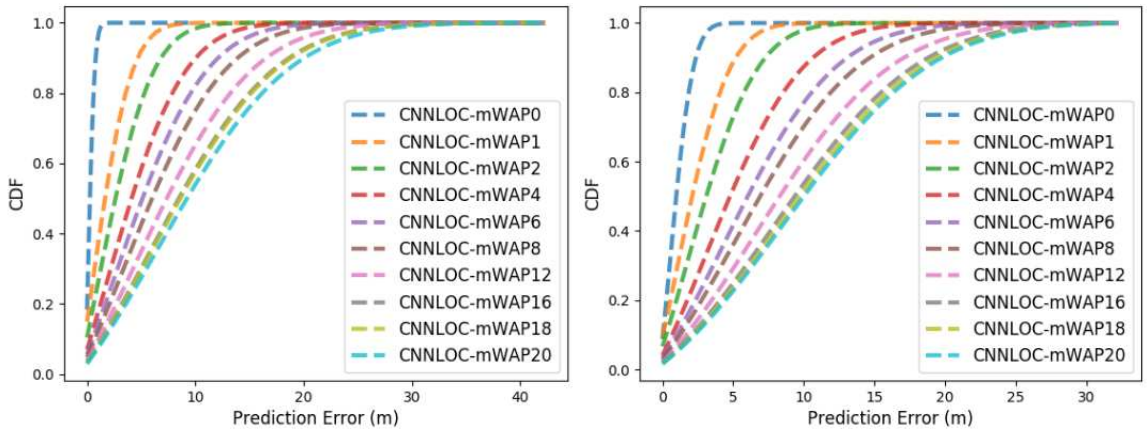
(a) Office **(b) Glover**

Figure 42. Heatmaps for the mean localization prediction errors with their annotated standard deviation for the Office (top) and Glover (bottom) benchmark paths. Results are shown for our proposed S-CNNLOC framework with Ø = 0, Ø = 1, … Ø = 20 (y-axis).

It can be observed that the S-CNNLOC0 model is least resilient to an increasing number of mWAPs on both paths. However, as the value of Ø is increased for the S-CNNLOC models, they per-form significantly better than S-CNNLOC0 (as illustrated by the darker rows for these models). This is because the S-CNNLOC0 model is not trained to mitigate variations for WAP RSSI values. Another observation is that beyond Ø = 18, the standard deviation and mean error for low values of malicious WAPs (mWAPs < 4) starts increasing for both paths. This is because highly noisy images in the SAAR database are unable to retain the original pattern required to

localize in safer environments (no malicious WAPs) or the opted CNNLOC model is unable to recognize underlying patterns in the input fingerprint images.

Overall, we observe that training the S-CNNLOC models with fingerprint extrapolation and noise induction (via the generated SAAR database) leads to better localization accuracy. Based on the results of these experiments we found that S-CNNLOC18 delivers good results across both paths. Therefore, we use the value of Ø = 18 in SAAR to train S-CNNLOC and use it for the rest of our experiments. Henceforth, whenever we refer to S-CNNLOC, we are referring to S-CNNLOC18 (S-CNNLOC with Ø=18).



(a) Office **(b) Glover**

Figure 43. Localization performance of CNNLOC with a varying number of malicious WAPs (from 0 to 20) in the online phase.

5.6.2.2. COMPARISON OF ATTACK VULNERABILITY

In this section, we contrast the performance of our proposed S-CNNLOC framework with CNNLOC [38]. Figure 43(a)-(b) show the cumulative distribution function (CDF) of the localization error for the CNNLOC models in the presence of different numbers of malicious WAPs (from 0 to 20 malicious WAPs per observed fingerprint), for the Office and Glover paths. The most immediate observation from the results is that the localization errors are significantly

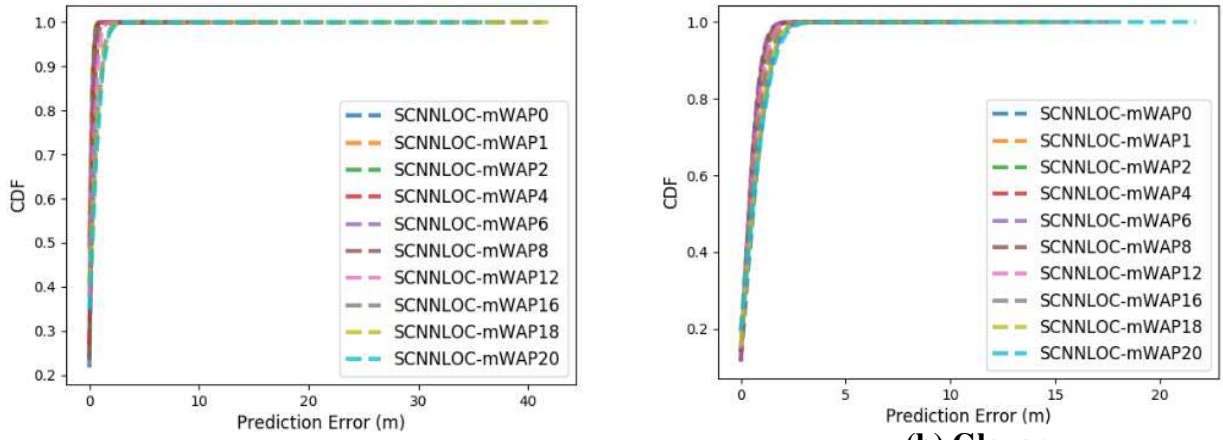
low (less than 1 meter for a majority of scenarios) when there are no malicious WAPs (CNNLOC-mWAP0), However, in both the Office (Figure 43(a)) and the Glover paths (Figure 43(b)), localization accuracy degrades as the number of malicious WAPs are increased. This degradation in accuracy does not scale linearly with increasing malicious nodes. For example, in the Office path, increasing the malicious AP nodes from 16 to 20 does not significantly increase the localization errors. A similar observation can be made from the Glover path in Figure 43(b), where the localization error does not scale by much when going from 12 malicious WAPs to 16 and 20.

An important aspect to note from looking at Figure 43 is the significant drop in localization accuracy when going from a scenario with no malicious WAPs (CNNLOC-mWAP0) to a scenario with one malicious WAP (CNNLOC-mWAP1). This accuracy drop is apparent on both paths, and clearly depicts the high vulnerability of unsecured CNN models to the presence of even a single malicious WAP node.

From Figure 43, we can conclude that a malicious third party can significantly degrade the localization accuracy of a CNN-based indoor localization model such as CNNLOC [38], with just a very small number of malicious WAP nodes.

Figure 44 highlights the resiliency of the S-CNNLOC model towards malicious WAP based attacks, for the same setup as for the experiment with CNNLOC in Figure 43, where the number of malicious WAPs in the online phase is varied from 0 to 20. We observe that 95-percentile of the localization error for the S-CNNLOC model, when under attack by up to 20 malicious WAP nodes (S-CNNLOC-mWAP20), remains under 2.5 meters for the Office path (Figure 44(a)) and under 3.5 meters for the Glover path (Figure 44(b)). The S-CNNLOC model for the Office path performs better than for the Glover path as the WiFi density on the Office path is about 2× the

WiFi density of the Glover path, and thus malicious WAPs only impact a small fraction of the total WAPs along the Office path.



(a) Office **(b) Glover**

Figure 44. Localization performance of our S-CNNLOC with a varying number of malicious WAPs (from 0 to 20) in the online phase.

In summary, based on the results shown in Figure 43 and Figure 44, we observe that our S-CNNLOC framework is about 10× more resilient to accuracy degradation in the average case, as compared to its unsecure counterpart CNNLOC [38], for the Office and Glover paths.

5.6.2.3. EXTENDED ANALYSIS ON ADDITIONAL BENCHMARK PATHS

We conducted further experimental analysis on a more diverse set of benchmark indoor paths. Table 3 presents the salient features of the three new benchmark paths used in this analysis. The benchmark path suite shown in Table 3 consists of the EngrLabs, LibStudy and the Sciences paths, with a description of environmental factors that may affect the localization performance of WiFi based indoor localization frameworks. Each path has a length ranging from 58 to 68 meters and 10 WiFi fingerprint samples were collected at 1-meter intervals on each path, similar to what we did with the Office and Glover paths described earlier. The EngrLabs path is in an old building

mostly made of concrete and is surrounded by labs consisting of heavy metallic instruments. The LibStudy and Sciences paths are situated in relatively newer buildings consisting of large amounts of metallic structures. The LibStudy path is in the library and is in a relatively open area and is usually heavily populated at most times. The Sciences path is surrounded by large classrooms.

Figure 45 presents the means and standard deviations of the localization error with our proposed S-CNNLOC and the CNNLOC [38] framework on each of the three paths while it is under the influence of 2 to 20 malicious WAPs in the online phase. We observe an increasing trend in mean and standard deviations of localization errors on all three paths for both S-CNNLOC and CNNLOC. However, we observed that the mean localization error of CNNLOC on all three paths is always more than 4× the average error for S-CNNLOC. For some situations, such as for 2 and 4 malicious WAPs on the EngrLabs and Sciences paths, the localization error for CNNLOC is about 25× higher (worse) on average as compared to its S-CNNLOC counterpart. The accuracy along the Libstudy path is relatively less affected than for the other paths. This can again be attributed to the fact that the LibStudy path has an unusually dense WiFi network compared to the EngrLabs and Sciences paths, and thus a relatively fewer number of malicious WAPs do not have as much of an impact on accuracy. These experiments with additional benchmark paths indicate that our proposed S-CNNLOC framework scales well over a wide variety of indoor paths with different environmental features whereas the unsecured CNNLOC [38] framework experiences a significant degradation in its localization error. The S-CNNLOC model consistently reduces the vulnerability of the proposed localization framework and thus represents a promising solution to secure deep learning-based indoor localization frameworks.

Table 3: Additional benchmark paths and their features.

Path Name	Length (m)	Number of WAPs	Environmental Features
EngrLabs	62	120	electronics, concrete, labs
LibStudy	68	300	wood, metal, open area
Sciences	58	130	metal, classrooms
Office	64	156	wood, concrete
Glover	88	78	wood, metal, concrete

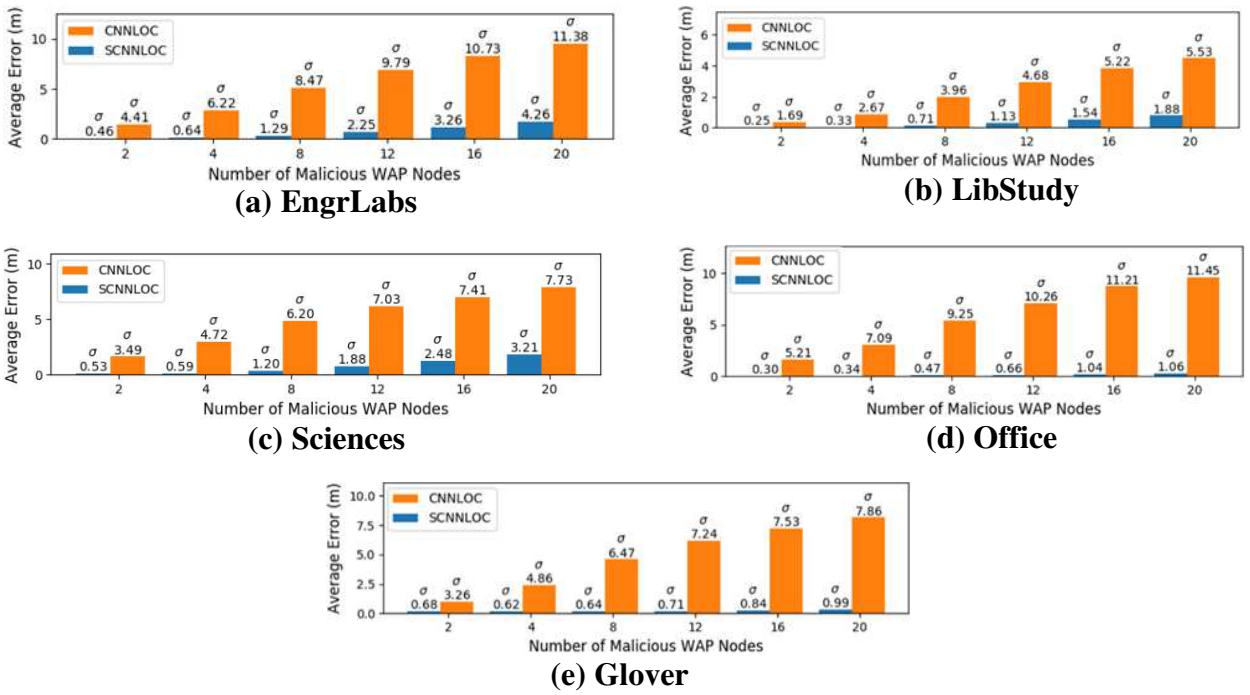


Figure 45. The average localization error and its standard deviation of the proposed S-CNNLOC framework as compared to CNNLOC for the benchmark path suite from Table 3.

5.7. GENERALITY OF PROPOSED APPROACH

In this section, we highlight the generality and the versatile nature of our proposed security aware approach by applying it to another deep learning-based approach proposed in [82]. We first present a discussion of the proposed work in [82]. Later, we use WiFi fingerprints generated in section 5.5.1 to train the secure-DNN (SDNN) model and compare its prediction accuracy results to the conventional methodology described in [82].

5.7.1. DENOISING AUTOENCODER BASED DNN FRAMEWORK

The DNN-based approach in [82] consists of three stages in the online phase. In the first stage, features are extracted from the RSSI fingerprints using a Stacked Denoising Autoencoder (SDA). The SDA's output is fed to a four-layer DNN model in the second stage that delivers a coarse location prediction. In the final stage, additional Hidden Markov Model (HMM) is used to finetune the coarse localization perdition received from the DNN model.

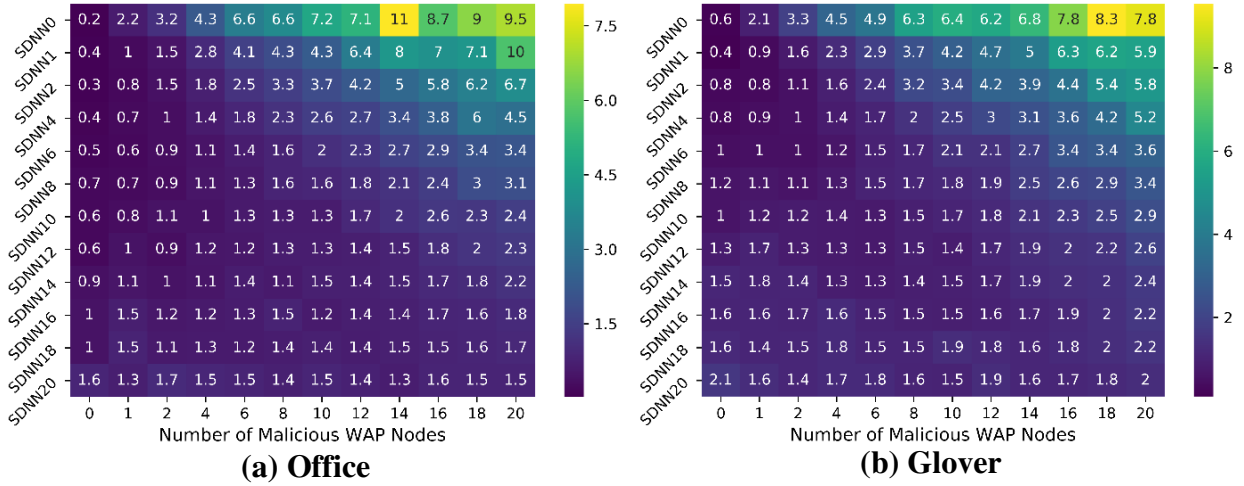


Figure 46. Heatmaps for the mean localization prediction errors with their annotated standard deviation for the Office (top) and Glover (bottom) benchmark paths. Results are shown for our proposed S-DNN framework with Ø = 0, Ø = 2, … Ø = 20 (y-axis).

The SDA enables the DNN model to identify and learn stable and reliable features from the input fingerprint information. Intuitively, SDA achieves this by zeroing-out input features based on a predefined probability and identifying input features that have a significant impact on the output. Further, the HMM allows for greater resistance to minor variations in WAP RSSI over time.

5.7.2. SECURITY AWARE DNN TRAINING IN THE OFFLINE PHASE

To train the SDNN model we use the augmented security aware fingerprints used to train the SCNNLOC model in the previous section. The only difference being that the fingerprints are not converted into images. To identify the stable value of Ø for noise induction module, we perform a sensitivity analysis using DNN models as done in section 5.7.1. The results for this experiment are captured in Figure 46.

In Figure 46, we observe that the mean localization errors for the baseline SDNN0 models for the Office and Glover paths increase by 48x and 13x in the presence of 20 malicious nodes respectively. For SDNN models trained with a larger value of Ø (14, 16, 18), the localization error remains lower as the number of malicious nodes in the online phase increase. For simplicity, we set the value of Ø to 18 for all paths. Beyond this point any reference to an SDNN model refers to DNN model [82] trained with Ø = 18. In the next subsection, we present an extended analysis on the performance of SDNN as compared to a conventional unsecured DNN model.

Figure 47 presents an analysis on the stability of the conventional unsecured DNN-based framework [82] as compared to secure-DNN (SDNN) model in the presence of an increasing number of malicious WAPs on a set of versatile paths with varying environmental characteristics as discussed in Table 3. From Figure 47, we observe the prediction accuracy of the conventional DNN-based approach presented in [82] systematically degrades (increased average error) as the number of stochastically placed malicious WiFi access points on various paths are increased. The SDA stage in [82] is supposed to learn prominent features by learning to encode prominent input features (ignoring noise) in the training phase. However, the noise in the training features over a short period of time is significantly lower and different from the addition of malicious WAPs in the online prediction phase. Due to the fact that the SDA does not learn to denoise malicious

fingerprints in the training phase the prediction accuracy of the method proposed in [82] degrades with the introduction of malicious WAPs in the testing or online phase. Further, the HMM model is unable to stabilize the final location prediction because it is designed to improve the fine-grain location based on the assumption that the consecutive coarse-grain predictions from the DNN are sufficiently close together. However, in the presence of malicious WAPs this assumption does not hold for the coarse-grain predictions causes the HMM to deliver unstable results.

On the other hand, the SDA component of the SDNN-based model learns to denoise and ignore malicious WAPs. This is achieved through stochastically zeroing out RSSI values, identifying stable trusted WAPs and denoising malicious WAPs over various fingerprints. As observed for various paths in Figure 47, this greatly improves SDNN's resilience to malicious WAPs in the online phase and delivers up to 10x better mean prediction accuracy such as in the case of 16 malicious WAPs on the EngrLabs path.

A notable aspect of our proposed approach is that it allows for the deep learning model to ignore malicious WAPs in the testing phase, however, the extent of resilience to the malicious WAP-based attacks is dependent on the deep learning model's ability to identify underlying pattern in the training fingerprints. Deep learning models such as CNNs and SDA-based approaches are more likely to deliver promising results as they are both designed to identify underlying stable patterns in the training phase. However, designing a deep learning model that delivers the best results in all situations is beyond the scope of this work.

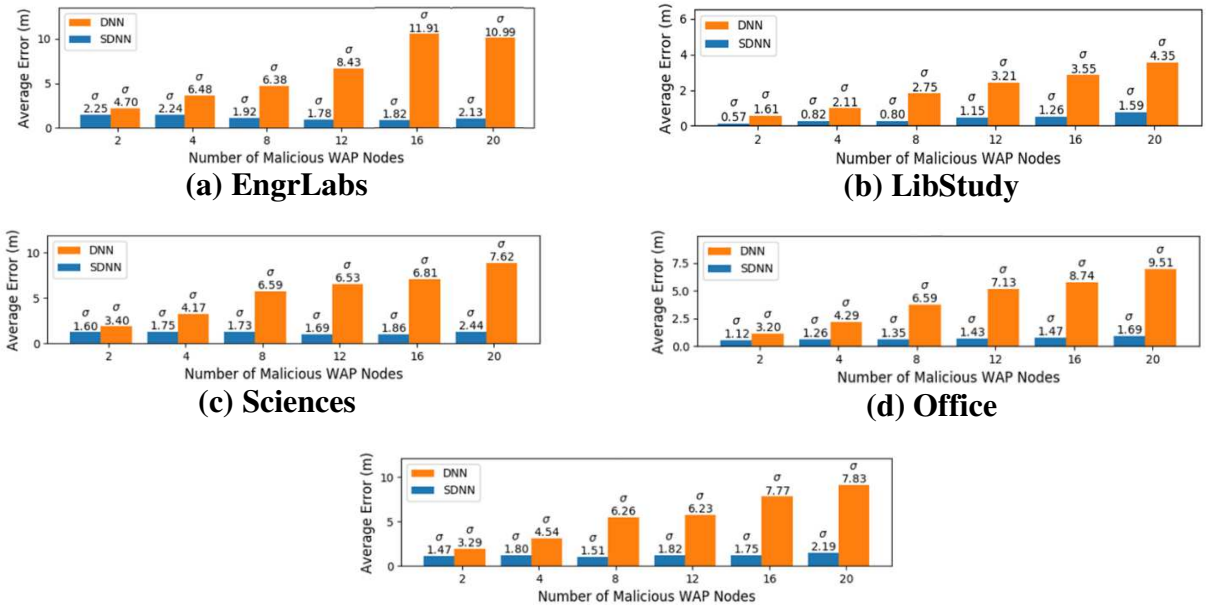


Figure 47. The average localization error and its standard deviation of the proposed S-DNN framework as compared to DNN for the benchmark path suite from Table 3.

Through experiments performed and the discussion of presented results, we can conclude that our proposed approach delivers superior stability of prediction accuracy of deep-learning-based models over a versatile set of benchmark paths. Furthermore, since our proposed approach of securing deep-learning-based models focuses on the training dataset instead of the model design, it can be generalized to a wide variety deep learning based indoor localization frameworks.

5.8. CONCLUSIONS

In this chapter, for the first time, we presented a vulnerability analysis of deep learning based indoor localization frameworks that are deployed on mobile devices, in the presence of wireless access point (WAP) spoofing and jamming attacks. Our analysis highlighted the significant degradation in localization accuracy that can be induced by an attacker with very minimal effort. For instance, our experimental studies suggest that an unsecured convolutional neural network

(CNN) based indoor localization solution can place a user up to 50 meters away from their actual location, with attacks on only a few WAPs. Based on our new observations, we devised a novel solution to provide resilience against such attacks and demonstrated it on a CNN-based localization framework to address its vulnerability to intentional RSSI variation-based attacks. To further highlight the generality of our proposed security aware approach we implemented it on a Deep Neural Network (DNN) based indoor localization solution. Our proposed vulnerability resilient framework was shown to deliver up to 10× superior localization accuracy on average, in the presence of threats from several malicious attackers, compared to the unsecured CNN and DNN-based localization framework.

6. QUICKLOC: OPTIMIZING LATENCY FOR DEEP LEARNING BASED INDOOR LOCALIZATION WITH MOBILE DEVICES

The commercialization of GPS technology in the 1980's completely reformed the transportation industry, simplifying the process of navigation for large ships and airplanes which were dependent on less reliable maps and compasses at the time. A major turning point was the development of the digital GPS ASIC created by Rockwell International, in 1988, using gallium arsenide (GaAs) semiconductor technology [126]. This enabled the first ever handheld GPS receiver to be produced for military applications. Further improvements over the next two decades led to the ubiquitous integration of GPS technology into mobile phones [94]. This empowered individuals to the point where bulky printed maps were no longer needed by automobile drivers and revolutionized outdoor terrestrial navigation around the globe.

Today, increasing capabilities of smart mobile devices are at a tipping point where they can now support localization and navigation technology within indoor environments, which promises to further remold the way humans interact within indoor spaces. As GPS signals cannot penetrate through into indoor locales, highly computationally expensive methods are required that can continuously capture and process wireless signals to support localization. Fortunately, inexpensive and ubiquitously owned smartphones today are computationally capable enough to support high-complexity machine learning models that can be fed by a dense suite of high-fidelity wireless interfaces and sensors on the device. Many researchers are pushing the boundaries on state-of-the-art design optimizations to achieve high-accuracy and real-time indoor localization capabilities on smartphones.

The advances in the indoor localization and navigation domain over the past decade have enabled new commercial and medical applications. Several solutions and standards are being recognized today to enable indoor localization in the public sector. A recent example is the new standard for WiFi that was established in collaboration with Google [4]. The new standard would allow anyone to set up their own localization system by sharing their indoor floor map and the WiFi router positions on that map with Google. Nowadays, companies such as Amazon and Target are also starting to track customers at their stores [11]. Indoor localization has found its applications in the medical industry by enabling the tracking of parkinsonian patients as they suffer from attacks associated with freezing of their walk gait [127]. However, such commercial or medical applications require high-quality (high accuracy and low response time) localization solutions and custom hardware components to be deployed at the target location, which drives up the setup and deployment costs.

One way to limit the setup and deployment costs associated with indoor localization services is to use relatively reliable wireless signal sources that are available freely. Due to the boom in the internet and network connectivity across the world in the previous decade, WiFi routers (access points) have become essential and a commonplace feature within indoor locales such as malls, warehouses, hospitals, and schools. Consequently, several recent efforts have focused on delivering high-accuracy localization and navigation solutions for the indoors through a technique called WiFi fingerprinting. Note that fingerprinting is an approach that is applicable for localization in both indoor and outdoor environments, although it is more widely used for indoor environments, whereas trilateration-based approaches (e.g., GPS) are more common for outdoor environments.

Indoor WiFi fingerprinting is based on the idea that each indoor location exhibits a unique signature that is comprised of WiFi signal strengths from visible WiFi routers at that location [128]. Such WiFi Access Point (WAP) Received Signal Strength Indicator (RSSI) values, along with the MAC IDs of these WAPs are captured across various indoor locations during a preprocessing phase, and used to train a model (e.g., machine learning based) that can be deployed on mobile devices. Post-deployment, this model can be used to predict a precise indoor location, given the WiFi RSSI and MAC ID values observed at the location. Alternatively, localization techniques have been proposed that are based on some form of distance relationship between the signal source and destination, such as triangulation [129] and trilateration [55]. However, these approaches suffer from weak wall penetration, multipath fading, and shadowing effects in real-world environments, making it difficult to establish a direct mathematical relationship between RSSI and distance from WAPs. By eliminating this distance relationship between the computed user location and WiFi signal source, WiFi fingerprinting with machine learning models is able to overcome the aforementioned challenges. Fingerprinting also has the advantage of not requiring knowledge of the precise locations of WAPs in an indoor locale, enabling non-intrusive localization.

The overall performance of a machine learning-based indoor localization and navigation framework can be evaluated through metrics such as accuracy, response-time, and scalability of the covered area. Further, the indoor localization framework may be subject to design constraints such as energy consumption, sensor type, and sensor resolution (fidelity). These constraints can be highly stringent when the indoor localization frameworks are deployed on off-the-shell commodity smartphones which have a limited energy budget and utilize severely power-limited processors. While simpler machine learning models such as K-Nearest-Neighbors (KNNs) and

Support Vector Machines (SVMs) are scalable, and incur lower energy costs and response times, these models have been shown to be outperformed by more complex and computationally expensive models such as feed-forward deep neural networks (DNNs) and convolutional neural networks (CNNs) that have higher response (inference) times. Moreover, it has been shown that increasing the depth of these neural networks leads to significantly improved localization accuracy but this comes at a cost of higher response times and energy.

Therefore, there is a compelling motivation for designing deep-learning based indoor localization frameworks with a focus on the optimization of their respective deep-learning models such that we can strike a balance between their response-time, energy and achievable localization accuracy. In this chapter, we present a novel approach for optimizing Convolutional Neural Networks (CNNs) for indoor localization that can be deployed on mobile devices towards the goal of meeting accuracy requirements (best achievable accuracy through state-of-the-art techniques), while minimizing response times. Our novel contributions in this work are as follows:

- We conduct an in-depth experimental analysis on the impact of CNN model depth on an indoor localization framework in terms of the achievable prediction latency and localization accuracy;
- For the first time, we adapt and explore the paradigm of conditional computing in the context of deep learning based indoor localization frameworks;
- We propose a novel localization framework that can dynamically adapt to the accuracy and latency needs of the target mobile platform at run-time;

- We compare the performance of our proposed technique against state-of-the-art deep learning based indoor localization framework over a diverse set of target mobile devices and indoor environments.

6.1. BACKGROUND AND RELATED WORK

6.1.1. RECEIVED SIGNAL STRENGTH INDICATOR (RSSI)

RSSI is a measurement of the power of a received radio signal transmitted by a radio source. The RSSI is captured as the ratio of the received power (P_r) to a reference power (P_{ref}, usually set to 1mW). The value of RSSI is reported in dBm and is given by:

$$RSSI\ (dBm) = 10 \cdot log \frac{P_r}{P_{ref}} \tag{10}$$

The received power (P_r) is inversely proportional to the square of the distance (d) between the signal transmitter and signal receiver in free space and is given by:

$$P_r = P_t \cdot G_t \cdot G_r \left(\frac{\lambda}{4\pi d}\right)^2 \tag{11}$$

where (P_r) is the transmission power, G_t is the gain of transmitter, G_r is the gain of receiver, and λ is the wavelength. This inverse relationship between the received power and distance has often been used by researchers to localize wireless receivers with respect to transmitters at known locations, e.g., estimating the location of a user with a WiFi capable smartphone from a WiFi WAP. However, the free space models based on equations (10) and (11) do not extend well for practical applications. In real environments, the propagation of radio signals suffers from attenuations and interference due to multipath propagation from signal scattering, reflection, and

diffraction on obstacles (such as walls, furniture, equipment, people, etc.). Such multipath and shadowing effects cause unpredictable variations in RSSI values at the receiver, thereby severely degrading the performance of free space model based indoor localization approaches, to the point of rendering them impractical for direct use [13].

6.1.2. INDOOR LOCALIZATION METHODOLOGIES

Since the inception of wireless radio frequency (RF) based localization a couple of decades ago, a considerable amount of progress has been made in this domain. Here we summarize some of the most noteworthy advancements in this area.

An RF based indoor localization framework, such as one relying on WiFi RF signals, can be classified into three broad sub-domains, i) static propagation based, ii) trilateration or triangulation based, and iii) fingerprinting based.

Static propagation model-based techniques are established on the idea that there is a direct correlation between the source signal strength and the distance at which this signal strength is measured. This concept is implemented at design-time by first making signal strength measurements at constant distance intervals from a source in a straight line. The drop in the signal strength in relation to the distance is captured as a static propagation model [55] [60] [130]. Finally, at run-time the location of the user is predicted based on the received signal strength that will correspond to a specific distance from source value in the model. Such static propagation models are only known to work under extremely controlled conditions with open areas, and no activity. They are often used in conjunction with an error correction system such as Bayesian filters [60] or additional receivers [129] that recalibrate the model over time. However, the RF signal propagation path between every user and source may be unique due to interactions with objects around them.

Further, RF transceiver characteristics can vary across devices and manufacturers, which adds to the scalability issues and unpredictability of such models in real-world environments.

Triangulation and trilateration-based techniques utilize multiple signal transceivers (e.g., WAPs) to locate people or assets in an indoor environment. They use distance measurements at run-time (by measuring time of flight) such as the distance between multiple WAPs and a mobile device (trilateration) [4] [55], or the angles at which the signals from two or more WAPs are received (triangulation) [56]. These techniques have shown to deliver higher accuracy and stability than static propagation models. The techniques can also tolerate device heterogeneity induced uncertainty, to a limited extent (albeit at a high maintenance cost for hardware and software support) [44]. However, these techniques (including Google's RTT [3]) have several limitations, e.g., they need physical locations of all WAPs which is information that may be difficult (or impossible) to obtain in many indoor locales; they require strict clock synchronization among WAPs and the receiver which is not easy to consistently achieve over time; and they may need sophisticated transceivers that are not available in most commodity mobile devices and WiFi Access Points. These techniques also do not work well due to signal interactions with objects in the environment that induce signal multipath, shadowing, and variation in propagation speed through materials other than air [131].

Fingerprinting techniques and their viable implementations can utilize machine learning domain to overcome the aforementioned challenges associated with signal interactions and maintenance costs. Our work therefore utilizes this approach. We discuss prior work in this area in the next sub-section.

6.1.3. FINGERPRINTING-BASED INDOOR LOCALIZATION

Due to the limitations of the static propagation model-based and triangulation or trilateration-based techniques, researchers are now increasingly focusing on fingerprinting-based indoor localization techniques. Fingerprinting can be implemented in two ways: 1) custom infrastructure based: where custom AP beacons are installed in indoor environments based on Ultra-Wide Band (UWB) [131], Bluetooth [110] or Zigbee [111], and 2) infrastructure-free: where freely available signal sources such as earth's magnetic field [57] [132] [133] and WiFi [80] are utilized. The former approach lacks scalability and suffers from high costs. Moreover, smartphones do not have transceivers for protocols such as UWB and Zigbee. The latter approach, because of its low cost and ease of setup, is therefore more preferable.

A generalized view of an infrastructure-free WiFi RSSI fingerprinting-based indoor localization framework is presented in Figure 48. Such frameworks usually consist of two phases: the offline (or training) phase and the online (or testing) phase. From Figure 48, we note that the offline phase consists of the user collecting WiFi fingerprints to create an RSSI fingerprint database. Each row of this database consists of RSSIs for various WAPs observed at a given location (reference point). The row of RSSI information is also known as an RSSI vector. One may collect RSSI vectors at each reference location multiple times (e.g., at different times of the day or week) to capture a broader range of RF signal behavior at that location. The same database is then used to train a Machine Learning (ML) model, where the RSSI vector is the input and the reference location is the output of the model. This model is finally deployed on to the mobile device that will be used by the end user for indoor localization. As discussed before, deploying indoor localization ML models on smartphones is becoming a common practice due to various infrastructure costs and computational capability benefits.

In the online or testing phase, as shown in Figure 48, the target mobile device captures an RSSI vector as a user moves across an indoor space. The RSSI vector is then fed to the ML model on the mobile device, that in turn predicts the location of the user. This process of capturing RSSI vectors and then predicting the user's location continues to occur in a cyclic fashion to create an ongoing stream of location prediction cycles. The time taken to complete a prediction cycle (i.e., prediction latency) and the accuracy of the predicted location are two key metrics that describe the responsiveness and effectiveness of an indoor localization framework. A truly real-time localization framework is expected to be responsive to the user's movement, providing continuous predictions (approximately taking no more than a few tens of milliseconds for each prediction), while maintaining an acceptable level of location prediction accuracy.

Over the previous decade, the area of fingerprinting-based indoor localization has been heavily explored. UjindoorLoc [80] describes a technique to create a WiFi fingerprint database and employs a KNN (K-Nearest Neighbor) based model to predict location. Their average accuracy using KNN is 7.9 meters. Radar [81] and Indoor Atlas [10] are early works that proposed using hybrid indoor localization techniques. Radar [81] combined inertial sensors (dead reckoning) with WiFi signal propagation models, whereas Indoor Atlas [10] combined information from several sensors such as magnetic, inertial, and camera sensors, for indoor localization. LearnLoc [37] combined non-deep ML models with inertial sensor data and WiFi fingerprinting to propose a framework that trades-off indoor localization accuracy and energy efficiency on smartphones.

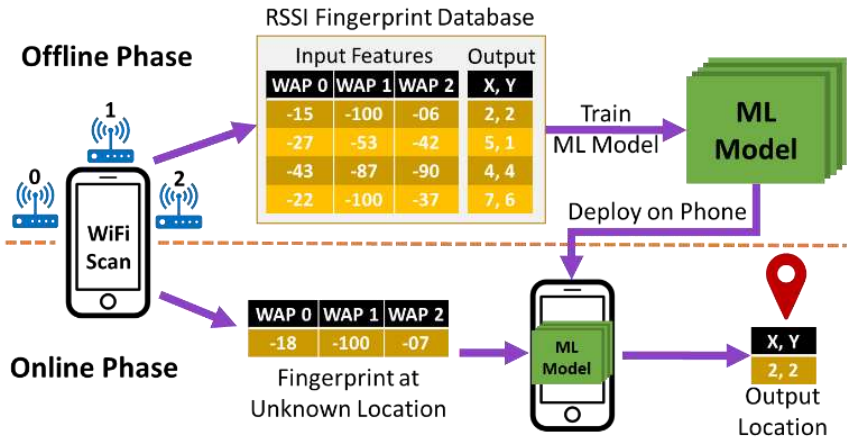


Figure 48. A generalized overview of the online and offline phases of fingerprinting-based localization frameworks.

As the computational capabilities of smartphones have increased in recent years, researchers have begun to explore the possibilities of deploying more complex algorithms such as DNNs on mobile devices towards the goal of attaining higher localization accuracies. Publicly available neural network frameworks such as TensorFlow and PyTorch have enabled rapid prototyping of complicated deep learning models and can be deployed on mobile devices with ease. The work in [82] presents an approach that uses DNNs and Hidden Markov Models (HMMs) for WiFi RSSI fingerprinting. DeepFi [35] and ConFi [83] propose approaches that use the Channel State Information (CSI) of WiFi signals to create fingerprints. But the CSI information in these approaches was obtained through the use of specialized hardware attached to a laptop. None of the smartphones available today have the ability to capture CSI data. Due to this limitation, it is not feasible to implement these techniques on smartphones. Deep Belief Networks (DBN) [33] have also been used for indoor localization, but the proposed technology is heavily reliant on custom UWB beacons that lead to a very high implementation cost. The work in [134] presents a deep-learning-based indoor localization framework that fuses fingerprints from two sources: WiFi, and magnetic signals, to produce the user's location estimate. A limitation of all of these prior works

on deep learning and fingerprinting based indoor localization is that they focus solely on indoor localization accuracy, without considering the responsiveness (i.e., prediction latency performance) of the proposed frameworks. The responsiveness of a fingerprinting-based indoor localization framework is heavily dependent on the prediction latencies of their respective machine learning models, as well as the specific mobile device platform that the model is deployed on. True real-time indoor localization can only be achieved if the time to sample signal fingerprints and producing a location prediction is small enough that there is no lag between user movement and location prediction displayed on the user's mobile device.

In summary, existing indoor localization frameworks focus extensively on prediction accuracy, however, very limited attention is placed on architectural optimization of existing deep learning models for lower prediction time and energy. To the best of our knowledge, there are no works in the area of fingerprinting-based indoor localization that explicitly focus on the optimization of deep learning models with an emphasis on reducing response-time with no loss (or gain) in accuracy. These factors are critical to achieve consistent performance and scaling of deep learning based indoor localization frameworks across a variety of mobile devices. Towards the end goal of creating responsive real-time indoor localization frameworks we propose the QuickLoc framework that adapts the early exit deep learning based architectural design philosophies presented in [135] and [136] to the domain of indoor localization, for the first time. QuickLoc has the capability to strike a balance between response time while maintaining high indoor localization accuracy.

6.2. CNNLOC FRAMEWORK OVERVIEW

In this section, we discuss the concepts associated with the WiFi fingerprinting-based indoor localization framework proposed in [38], called CNNLOC. We utilize CNNLOC as the baseline work due to the benefits of using WiFi RSSI only indoor localization and deep learning as highlighted in the previous section, as well as due to the fact that this recent work has been deployed on mobile devices and shown to outperform other solutions in the indoor localization problem domain. Our goal in this work is to improve upon the performance achievable by CNNLOC. However, please note that the design methodology proposed in this chapter can be applied to other deep learning frameworks as well, such as [33] [35] [38] [83] [134].

6.2.1 CONVOLUTIONAL NEURAL NETWORKS

Convolutional neural networks (CNNs) are a form of deep neural networks that are specially designed and optimized for image classification. They have been shown to deliver significantly higher classification accuracy as compared to conventional fast-forward only DNNs due to their enhanced pattern recognition and feature extraction capabilities.

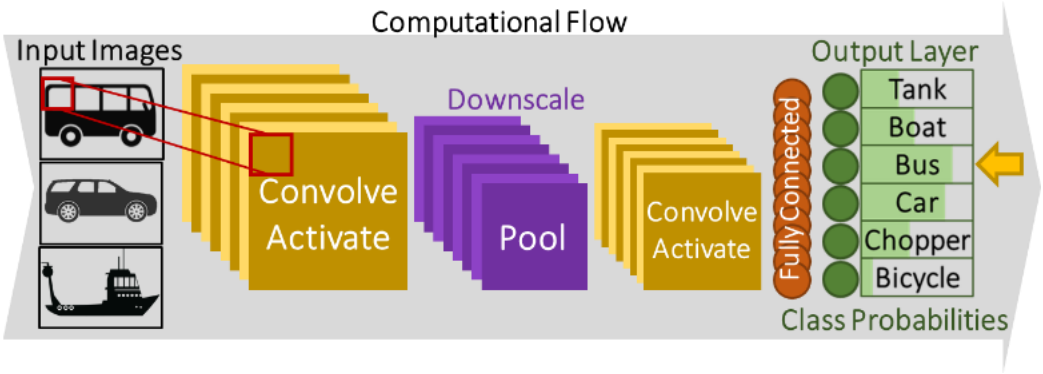


Figure 49. An example of a Convolutional Neural Network (CNN) design.

As shown in Figure 49, a typical CNN model has three main functional components (or types of layers): convolutional layers (that perform "convolve" operations), pooling layers (that "pool" or downsample the activations), and fully connected layers (that feed flattened data to the output for predictions). Convolutional and fully connected layers also have associated activation functions ("activate" operations) whose role is to introduce non-linearity into the neural network model, allowing the model to learn complex, non-linear patterns. ReLu (Rectified Linear Units) and its variants (such as leaky ReLu) are the most popular activation functions in the pattern recognition domain.

In general, CNN models learn patterns in images by focusing on small sections of the image, known as a frame, as shown in Figure 49. The frame moves over a given image in small strides. Each convolutional layer consists of filters (matrices) that hold weight values. The layer involves convolutional operations (dot products) performed between the current input frame and filter weights followed by passing the result through the activation function. The pooling layer is responsible for down sampling the output from a convolutional layer, thereby reducing the computational requirements by the next set of convolution layers. A set of fully connected layers is typically utilized after potentially multiple convolutional and pooling layers, to reduce the depth of activations (i.e., data propagating through the CNN) before a final classification can be performed. Typically, a SoftMax activation function is applied to the output of the last fully connected layer, to generate the probability distribution for the classes being predicted by the model. In the testing (or inference) phase of a CNN model, an image is fed to the model which in turn produces class probabilities. The class with the highest probability is identified as the output prediction. Further details on the design of CNNs can be found in [89] and [85].

6.2.2 INDOOR LOCALIZATION WITH CNNLOC

The CNNLOC indoor localization framework [38] consists of two major components in the offline phase. The first component involves capturing the RSSI fingerprints for different indoor locations, and then converting each RSSI fingerprint vector that is associated with a location (reference point) into an image associated with the same location. The second component of the offline phase is the training of a CNN model using the images created from RSSI vectors. In the online phase, the same process is used to create an image (based on observed RSSI values), which is fed into the trained CNN model for location prediction.

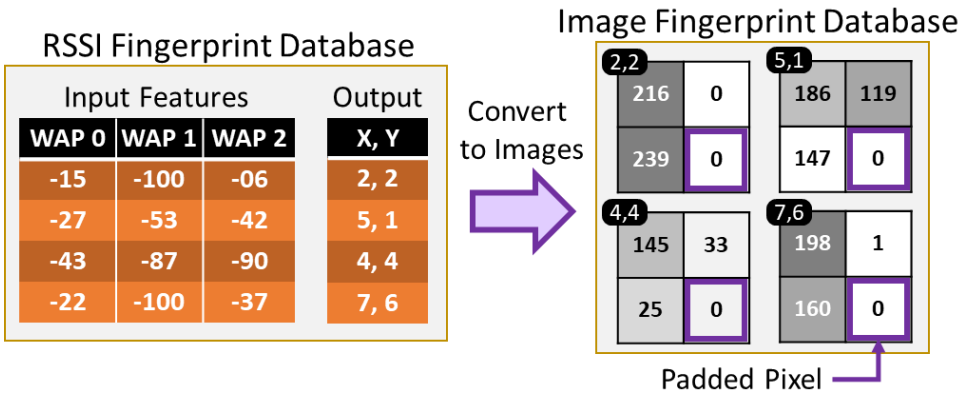


Figure 50. Converting RSSI fingerprint vectors to RSSI images.

A simplified overview of the process of converting an RSSI fingerprint vector into an image is shown in Figure 50. The RSSI vector consists of RSSI values in the range of -100 to 0 dBm (low signal strength to high signal strength). These values are normalized to a range of 0 to 255, which corresponds to the pixel intensity on the image. The dimensions of the RSSI image are set to be the closest square to the number of visible WAPs on the path. For example, in Figure 50, the RSSI vector has a size of 3, and the closest square would have 4 pixels in it, therefore, the dimensions of the image are set to 2×2. A pixel with zero intensity is padded at the end to increase the size of

the vector as shown in Figure 50. The generated image then becomes a part of the offline database of images used to train the CNN model.

In the online phase, this same process of image creation is used with the RSSI vector observed by the user at any location, and the resulting image is fed to the trained CNN model to get a location prediction. It is important to note that in the online phase of CNNLOC, the input image will always remain the same size as in the offline phase, such that each pixel in the image corresponds to the RSSI value from a WAP with a specific MAC IDs. In case a specific MAC ID observed in the offline phase is no longer visible in the online phase, we set the RSSI value for it to -100 dBm. This results in the pixel value corresponding to that MAC ID being set to zero.

6.3. LOCALIZATION INFERENCE ANALYSIS

We begin with an analysis of the impact of model depth on the state-of-the-art indoor localization framework CNNLOC [38] described in the previous section. To capture the impact, we train three unique CNN models for the paths shown in Figure 51 and deploy them on the four mobile devices summarized in Table 4. The first model has only one layer of convolution, the second model has two layers, and the third model has three layers of convolution. Due to small input image sizes, our models do not have pooling layers [38]. It is important to note that each of these models are trained to cover all of the paths shown in Figure 51. More details about the indoor paths and devices are covered in Section 6.6 Further, to curtail the complexity of this experiment, we utilize the same hyperparameters for the convolutional layers as in the model described in section 6.5.1.

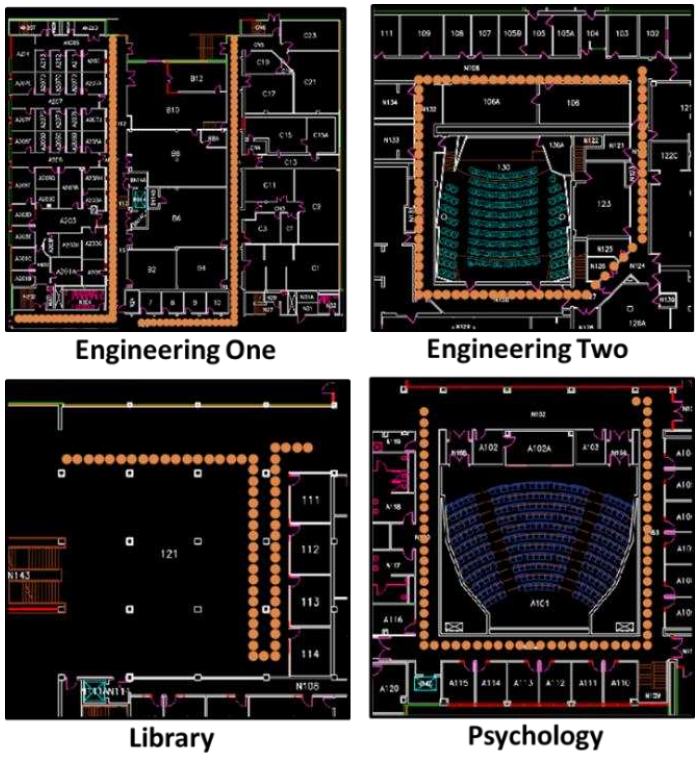


Figure 51. Indoor paths in different buildings for indoor localization analysis. Reference locations (where RSSI values were recorded to train the CNN models) along the indoor paths are indicated by orange dots.

Table 4: Details of smartphones used in experiments.

Smartphone	Chipset	CPU Freq.	RAM
OnePlus 3 (OP3)	Snapdragon 820	2350 MHz	6 GB
Moto Z2 (MZ2)	Snapdragon 835	2350 MHz	4 GB
Samsung S6 (GS6)	Exynos 7420	2100 MHz	3 GB
Samsung S7 (GS7)	Snapdragon 820	2300 MHz	4 GB

Figure 52 depicts the variation of model prediction accuracy and average latency for CNN models of varying depths deployed on the four different mobile devices. Considering the fact that smartphone chipsets are usually heterogenous in nature and consists of complex cores (clocked at higher frequencies) and simpler cores (clocked at lower frequencies), we report the latency values for situations where the model is specifically executed on the core clocked at the highest frequency

available. For each CNN model depth increment, we added an additional convolutional layer to the model. The most obvious observation is that in general the deepest model incurs significantly higher prediction (inference) latency. The model with three convolutional layers is up to 8x slower (OP3 device) than its shallow single convolutional layer counterpart. By increasing the model depth, we are able to boost the localization accuracy from 85% to 95%. However, this boost in localization accuracy comes at a hefty price of higher localization time. On the other hand, this observation also indicates that the patterns associated with 85% of the fingerprints are easily identifiable and utilizing deeper models is actually inefficient for most of the path covered by the user.

Prediction (inference) latency is a critical factor for the fulfillment of real-time indoor localization and navigation through fingerprinting. This is especially true for hybrid indoor localization frameworks that combine various techniques such as fingerprinting, dead reckoning, and particle filters at the same time to produce consistent high-fidelity results. For example, an indoor localization framework that depends on a machine learning model to inform other subsystems and aims to update the smartphone display every time the user moves by a 10^{th} of a meter, requires the predicted location to update every 30 milliseconds (assuming an average movement speed of 3m/s [137]). This is only achievable if the indoor localization framework is able to pre-process the fingerprint and produce an inference from the deep learning model at a latency that does not exceed approximately 30 milliseconds, based on our empirical experience with deploying and running such models for indoor localization on smartphones. In Figure 52, we observe that the 3-layer model is unable to deliver such latency on most mobile devices. We also tested CNN models with more than 3 layers (results omitted for brevity) and found a much higher

inference latency with those deeper models, which made them not very well suited to our real-time indoor localization inference time goals.

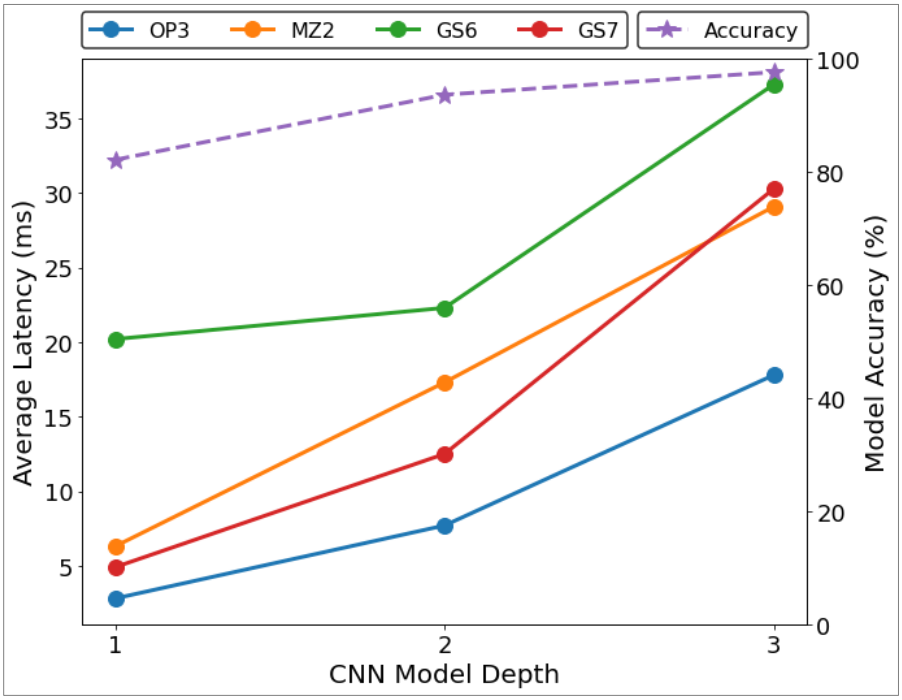


Figure 52. Relationship between CNN model depth, average prediction latency, and accuracy for the four smartphones.

Another observation from Figure 52 is the variation in prediction latency across the various mobile devices. While the latencies of OP3, MZ2 and GS7 devices are similar for a model depth with depth 1, the latencies for the models with a depth of 2 and 3 vary greatly. These localization latencies of the same CNN model are significantly dependent on the specifications and optimizations for the target mobile device. By comparing the device configurations in Table 4 and Figure 52, we conjecture that the DRAM specifications may play a crucial role in determining the prediction latency of the CNNLOC model. Further, we noted from our analysis that depending on the type of core the model workload is allocated to, the localization latency could be up to 3× worse than the ones reported in Figure 52.

As smartphones are powered by batteries and thus, limited by an energy budget, utilizing deep models for a task that can be accomplished using a shallower model wastes computational resources that could have been allotted to other tasks to further improve localization accuracy, such as direction estimation, via sensor fusion with inertial sensors, and using particle (or Kalman) filters [24] [138] [139]. Further, as we scale up the number of reference points on which RSSI readings are measured during the training phase, and the number of WAPs, the model depth and complexity needed to achieve accurate indoor localization will also increase. This will in turn result in higher prediction latencies, which will create a challenge for the deployment of such models on mobile devices.

The observations from the analysis in this section suggest a critical need for indoor localization frameworks that can deliver high localization accuracy without trading off localization latency and that can also perform consistently across a wide variety of heterogeneous mobile devices.

6.4. CONDITIONAL EARLY EXIT MODELS

From our analysis in the previous section, we observe that a shallower model is able to predict 85% of the locations accurately. This observation suggests that we do not need to use a deeper model to predict the user's location in every prediction cycle. Even though the deeper model can predict the user's location more accurately on average, it comes at the considerable cost of higher inference latency. Further, as the technique in CNNLOC [38] and other similar techniques are scaled up, the high complexity of the deployed model may become a barrier from its ubiquitous use in resource constrained devices such as smartphones and smartwatches.

Towards the goal of optimizing the inference latency of the indoor localization model, we exploit the observation that a large portion the WiFi fingerprints in the training dataset can be learned easily and effectively by simpler models. However, we also want to ensure that locations that can benefit from a deeper model can actually leverage the benefits of additional layers for improved prediction accuracy. To realize such an implementation, we build on the idea of early exit in deep learning models, as proposed in [135] and [136]. We explore the possibility of branching the computation after each convolutional operation to achieve an acceptable response based on uncertainty sampling methods such as confidence difference, confidence ratio, or entropy. These are discussed later in this section.

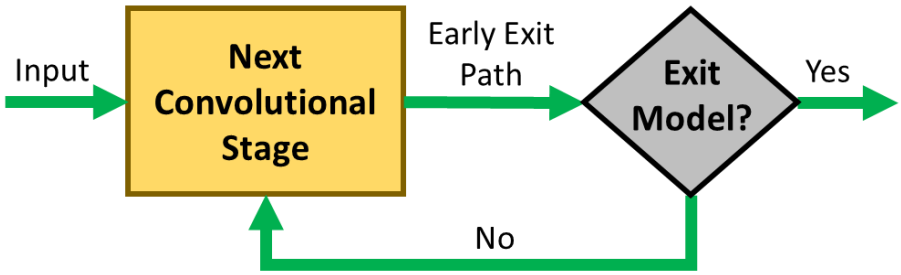


Figure 53. Early exit strategy depicted as a state machine.

The adapted conditional exit strategy can be captured as a state machine and is depicted in Figure 53. The input to the state machine is an image that is fed to the convolutional neural network. After each convolutional layer or stage, the output is fed to an exit path. The output class probabilities produced at the end of the exit path are then fed to an uncertainty sampling method. The satisfactory result of the uncertainty sampling method is used to recognize the validity of the predicted class at the current exit stage. In case we are confident of the model output at the current exit stage, the current prediction is accepted. In the case that we are not confident of our early prediction, we continue on to the next convolutional stage and evaluate the output of that stage for

conditional exit. In this manner, we expect a reduction in the inference time for a majority of the location prediction cycles through uncertainty sampling based early exits.

Consider the example that was shown in Figure 48 earlier. The model is fed an input image of a car and produces the probabilities for various vehicle classes, such as a tank, boat, bus etc. While the class probability of a bus is the highest, the probability of the input image being a car is only slightly lower. Such behavior is expected as the images of buses and cars may have similar features such as wheels and large windows. However, a CNN model is likely to easily differentiate between a boat and a car due to dissimilar features or patterns in the images. In this manner, if input images of a CNN model have significantly varying features, they can be easier to identify using shallow models. Further, the distribution of probabilities across the various output classes (as seen in Figure 48) can be utilized to capture the model's confidence in its prediction. The class of techniques used for this purpose are known as uncertainty sampling methods. A subset of these methods is explored in this work for our problem of fingerprinting-based indoor localization. The explored methods are described below:

- **Least Confidence:** This is the difference between the most confident prediction and 100% confidence;
- **Margin of Confidence:** This is the difference between the most confident and the second most confident prediction;
- **Ratio of Confidence:** This is the ratio between the top two highest class probabilities (most confident);
- **Entropy:** This is a concept derived from information theory that describes the level of uncertainty associated with one possible outcome, compared to all other outcomes [140].

The early exit strategy reduces inference latency by limiting the overall computation required for each prediction (inference). Other well-known techniques such as model compression [141] [142] and quantization [143] are orthogonal to this method and can be applied in conjunction with this approach. Based on the proposed early exit strategy, several predictions follow a shorter path to completion thereby establishing shallower computational paths, with lower latencies. This is also a highly beneficial behavior as shallower models are less likely to be a victim of the vanishing gradient problem [144]. Shallower models in our problem domain are sometimes able to identify some locations accurately that might be harder for deeper models to predict accurately, due to this issue. In our experiments, we found evidence of this phenomenon, where utilizing early exit models allowed for improving localization accuracy under some conditions.

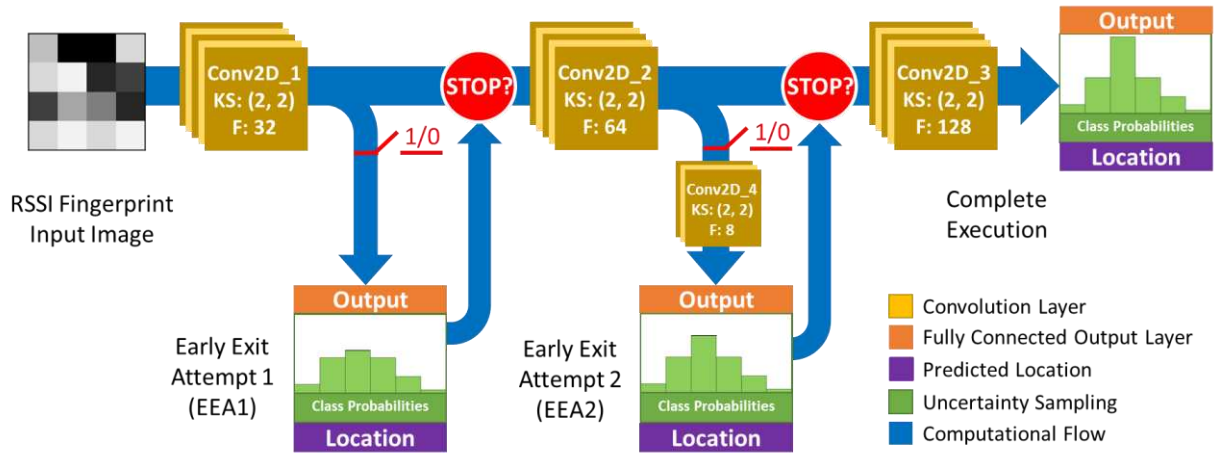


Figure 54. Overall flow of computation with conditional early exits for the proposed QuickLoc indoor localization framework.

It is important to noted in Figure 53 depicts the early exit model behavior as a state machine, it does not capture the specific conditional early exit model design presented in this work. The early exit path depicted in Figure 53may contain one or more neural network layers that are not a part of the original CNN model. We present the detailed process for creating the early exit model

that we used in the proposed QuickLoc fingerprinting-based indoor localization framework in the next section.

6.5 QUICKLOC FRAMEWORK

In this section, we discuss the design of our QuickLoc framework for the purpose of reducing inference latency.

6.5.1. QUICKLOC CNN MODEL DESIGN

The proposed model design for this work is depicted in Figure 54 and the number of parameters in each layer is presented in Table 5. The baseline model consists of three convolutional layers each with a small kernel size of 2×2 and a stride size of 1. A small kernel size is chosen as the RSSI fingerprint images have a small resolution as discussed in CNNLOC [38]. We further utilize the same filter size in each layer to maintain simplicity in this exploration. A real-world deployment could have different filter sizes at each convolutional layer. The baseline model is designed such that the number of filters is increased as the depth of the model increases. This forces the CNN model to learn increasing number of complex features as the model depth increases. The first convolutional layer "Conv2D_1" consists of 32 filters producing only 160 parameters, followed by the second layer "Conv2D_2" with 64 filters (8.2K parameters) and finally, the third convolutional layer "Conv2D_3" consists of 128 filters (32.8K parameters). Each convolutional layer is followed by a ReLu activation function, as in [38].

Based on our discussion in the previous section, we attempt to perform an early exit after each convolutional stage. The first early exit attempt (EEA1) comes after Conv2D_1 and only consists of a single output layer. Each fully connected output layer consists of 342 neurons (same as the total number of reference points) followed by the Softmax activation function. In EEA2, an

additional convolutional layer "Conv2D_4" with only a few filters (8 filters producing 2K parameters) is attached before the output layer. From Table 5, we observe that all of the output layers consist of a large number of parameters. As the QuickLoc model adds multiple output layers to the baseline model, the resulting model is expected to have a larger memory footprint. An analysis into memory footprint at run-time is presented in section 6.7.5. Further, the hyperparameter selection of the two early exit branches is discussed in the next subsection.

Table 5: Number of parameters in the QuickLoc model.

QuickLoc Layer	Number of Parameters
Conv2d_1	160
EEA1 Output	9,204,246
Conv2d_2	8,256
Conv2d_4	2,056
EEA2 Output	1,994,886
Conv2d_3	32,896
Output	31,913,046

6.5.2. QUICKLOC MODEL TRAINING

The training process for the model presented in Figure 54 begins with the baseline CNN models design and training as discussed in [38]. To highlight the full potential of our proposed technique we chose to train a single model for all of the buildings in our dataset instead of a model for each building.

Once the baseline model is established, the first early exit stage (Conv2D_1+ EEA1) is created by training the layers on the EEA1 path such that the weights associated with the convolutional layers of the baseline model (Conv2D_1) are frozen and remain unchanged in the training process. Once the layers associated with the exit path have been trained, they are manually attached to the full baseline model. This process is repeated for each convolutional layer in the baseline CNN model.

While designing the layers on each early exit path, two design philosophies are followed. The first is that the depth of the early exit itself is generally directly proportional to the depth of convolutional stage whose output is fed to the early exit. In this manner, we note that EEA2 is computationally more expensive than EEA1. The second is that the computational expense of an early exit should be significantly lower than the remaining computation in the baseline model. It is important to note that the expense of a computational path is dependent on several factors such as number of layers, number of parameters in each layer, and the types of layers. The proposed design in this work considers all of these factors.

6.5.3. UNCERTAINTY SAMPLING THRESHOLD

At each early exit attempt, the confidence associated with the predicted output is calculated through class probabilities using one of the various uncertainty sampling techniques presented in the previous section. If the uncertainty of the predicted class is within an acceptable threshold, the location prediction at the current early exit is accepted. The value of these thresholds and the acceptable range is dependent on the type of uncertainty sampling method used. A sensitivity analysis on the choice of the uncertainty sampling technique is presented in the experimental section (section 6.7.1).

6.5.4. POST-DEPLOYMENT CONFIGURATION ADAPTIVITY

From our analysis shown in Figure 52, we observe that the performance of a CNN model can vary significantly across different devices. Subtle variations such as SoC type and DRAM size can lead to significant performance variations for the same CNN model. Due to this behavior, a

one-for-all CNN model solution is inefficient and is likely to deliver inconsistent inference time on new devices not evaluated in the training phase.

Another notable challenge of the proposed early exit strategy is the computational or latency penalty due to inferences that are unable to confidently exit on any of the early exit attempts. The latencies associated with inferences or location predictions that completely fail to exit early would be generally greater than the baseline CNN model without early exit attempts.

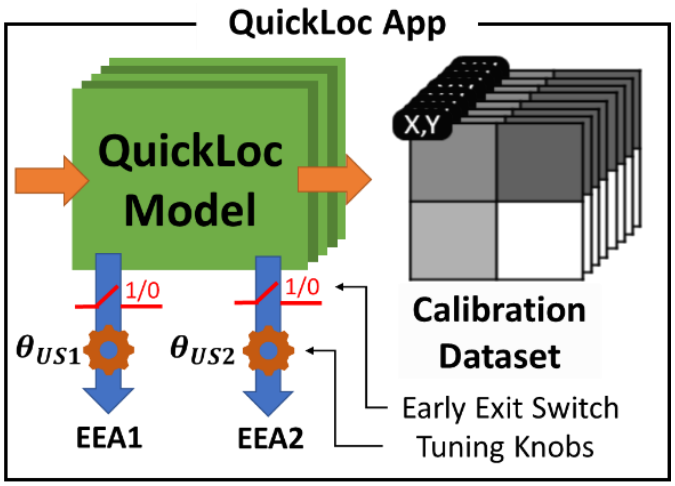


Figure 55. Contents of QuickLoc app package depicting tunable uncertainty sampling threshold (θ_{US}) and early exit switches as configurable parameters.

To overcome these challenges, we implemented the capability of enabling or disabling early exit paths once the model has been deployed on a smartphone and is in the testing phase. This is due to the fact that there may be multiple combinations of the ways the proposed early exit CNN model in the QuickLoc framework can be configured. For example, a model that only has EEA1 enabled, may deliver higher accuracy and lower inference time than a model with both EEA1 and EEA2 enabled. Once the model has been deployed on a smartphone, it undergoes a self-configuration process using a limited set of RSSI fingerprints and associated reference points to

identify an early exit configuration and uncertainty sampling threshold that delivers the best results.

Figure 55 depicts the various components of the QuickLoc indoor localization application (testing phase) and associated tunable parameters. The application also consists of a small set of labeled training data used for calibrating the various control knobs of the QuickLoc model. For each EEA, there are two control parameters: self-enable/disable switch and uncertainty sampling threshold value. Once the application is installed on a smartphone in the testing phase, the labeled training data is utilized to identify a localization error and inference latency for each possible configuration of the QuickLoc model (EEA and θ_{US}). This would allow us to identify multiple configurations that deliver higher accuracies than the baseline (no early exit) at lower inference latencies. We present an analysis later in section 6.7.3 that uses this approach to explore multiple configurations of the model across different smartphone devices.

For the purpose of this work, the default configuration is the one that produces a reduction in prediction latency with no loss in localization accuracy. However, in practice, the QuickLoc configuration can also be adjusted on-demand to meet specific latency goals at run-time. The benefit of such an approach is the ability to trade off latency with accuracy in the testing phase. For example, when using QuickLoc in combination with dead reckoning, one may choose to change the QuickLoc configuration with higher inference latency and lower localization error at run-time if the user is detected to be moving slower. To understand the impact of the various early exit configurations we present a sensitivity analysis on various devices later in section 6.7.2.

6.6. EXPERIMENTAL SETUP

6.6.1 HETEROGENEOUS DEVICE SPECIFICATIONS

To capture the variation in performance across heterogeneous devices, we first design and train the QuickLoc model based only on the OP3 device characteristics and then deploy our indoor localization model onto three other smartphones with unique hardware specifications in the testing phase. The specifications for each of these devices is captured in Table 4. This allows us to explore the impact of device specification heterogeneity such as DRAM and SoC type that can impact localization latency. Such a model design and training process is adopted to simulate a real-world scenario where the specifications of the target mobile platform may be unknown when deploying QuickLoc on new device.

6.6.2. INDOOR PATHS FOR LOCALIZATION BENCHMARKING

We compare the localization accuracy and latency for the proposed QuickLoc framework using a benchmark dataset with 342 reference locations. The benchmark spans over a large university campus with varying environmental conditions and WiFi WAP densities. The dataset covers four buildings. The paths within these buildings are shown in Figure 51; with each orange dot indicating a reference point on a path within the building that is one meter apart. The paths vary from 70 to 90 meters in length and the number of visible WAPs along these paths varies between 78 to 218. We collected data on these reference points at different times, and performed post-processing on the collected data to eliminate temporary WAPs, e.g., mobile hotspots created by individuals in the buildings.

The path sections in Engineering Building One consist of labs, mechanical equipment, and office spaces. This path was specifically chosen as it has the largest amount of electrical and

magnetic devices in its vicinity, that interacts with WiFi signals to produce noisy fingerprints. The psychology and library buildings were recently renovated with a mix of wooden and metallic structures in its surrounding environment. The path sections are mostly surrounded by large halls and classrooms such that the impact of multi-path effects and shadowing is relatively lower as compared to other buildings. Finally, the last building covered is an engineering building (Engineering Two). This is the most versatile path section covered. It is one of the oldest buildings on campus and mostly constructed of wood and concrete. The building consists of labs with metallic equipment, office spaces, and large classroom halls. The reference points for fingerprints over all buildings are 1-meter apart. Ten fingerprint samples per reference location were collected. The WiFi fingerprints in this benchmark were captured in the offline phase while holding the smartphones at an average height of 1.5 meters above ground such that the device screen is zenith facing. Testing (online phase) was performed by 5 users with heights varying between 175-192 cm. The users held the device close to their chest height while facing the smartphone display.

6.6.3. COMPARISON WITH PREVIOUS WORK

The performance of QuickLoc is compared to its non-early exit capable counterpart CNNLOC [38], which is the baseline model in our analysis. Additionally, we compare QuickLoc with conventional machine learning indoor localization frameworks that utilize K-Nearest Neighbor (KNN) [37] and Support Vector Regression (SVR) [92]. The KNN-based indoor localization framework [37] algorithm is based on the idea that the RSSI fingerprints at a given reference point would be close to each other in the Euclidian space. The SVR-based framework [92] attempts to create a set of hyperplanes, based on groups of RSSI fingerprints, each associated with a specific reference point. These frameworks utilize relatively light-weight machine learning

algorithms that lead to lower inference latency. The purpose of comparing QuickLoc against the works in KNN [37] and SVR [92] is to contrast the inference latency and accuracy of QuickLoc against known light-weight indoor localization platforms.

6.6.4. DEPLOYMENT AND EVALUATION

The early exit model is trained as described in section 6.5.2. The uncertainty sampling methods and threshold values for each early exit was empirically evaluated and set based on the OP3 device in the offline phase. The trained early exit model and the baseline models are deployed on smartphones using an Android app with timers for capturing latency. Once deployed, QuickLoc automatically reconfigures itself for the target smartphone. This is a one-time process that occurs at the first launch of the QuickLoc app.

We deployed the QuickLoc on smartphones using Tensorflow Lite and used the official C-based benchmarking application [145] over the Android Debug Bridge (ADB) to capture latency and memory requirements. This allows us to minimize the impact variations produced by the Android OS application manager layer. The energy analysis presented in section 6.7, is conducted by capturing battery drain characteristics attained using the BatteryManager API for Android [146]. We do not perform any form of post-training quantization on our Tensorflow Lite models. However, doing so would only further improve the inference latency of the QuickLoc model at the cost of localization accuracy. Lastly, WiFi RSSI fingerprint scans took anywhere from 1.5 to 4 seconds, depending on the smartphone being tested. As we move towards the eventual goal of real-time localization, higher sampling (scan) rates are needed. Recent efforts to enable monitor mode for WiFi chipsets for smartphones are a step in that direction, by enabling more frequent packet-by-packet updates to WAP RSSIs [147] [148].

6.7. EXPERIMENTAL RESULTS

6.7.1. SENSITIVITY ANALYSIS FOR UNCERTAINTY SAMPLING

In this subsection, we present results for a sensitivity analysis on the type of uncertainty sampling technique and its associated threshold value for our proposed QuickLoc model. The sensitivity analysis is conducted on the OP3 mobile device as it shows the least variation in prediction latency (Figure 52). Through the selection of this device we intend to describe the performance of QuickLoc on a smartphone whose prediction latency is the least flexible and hence, is expected to produce the least improvement. For simplicity, we utilize the same threshold values for EEA1 and EEA2 (both enabled).

Figure 56 presents the average localization errors and the prediction latencies on the left and right vertical axes, respectively; and the uncertainty threshold values on the horizontal axes, for the four uncertainty sampling techniques described in section 6.5.3 (margin of confidence, least confidence, ratio of confidence, and entropy). The dashed horizontal red lines and green lines represent the localization error and latencies (respectively) for the baseline CNNLOC framework. We observe that the performance of QuickLoc is greatly impacted by the choice of uncertainty sampling method. In Figure 56, we observe that the Least Confidence method performs the worst as there are no configurations of the uncertainty threshold value for which QuickLoc delivers higher accuracy at a lower latency than the baseline CNNLOC model. In contrast, Margin of Confidence and Entropy produce the most configurations with both improved latency and localization accuracy. Due to the logarithmic nature of localization error for the entropy method, it may not be the best choice for a framework variant that throttles the uncertainty threshold for a tradeoff between localization accuracy and latency. More analysis on this subject is presented in a

later subsection. From the analysis presented in this section, the margin of confidence is the best choice for the OP3 device. However, the appropriate adaptive configuration for each device may be unique for QuickLoc in the online phase on the target smartphone. The next sub-section highlights QuickLoc's configuration flexibility for various devices.

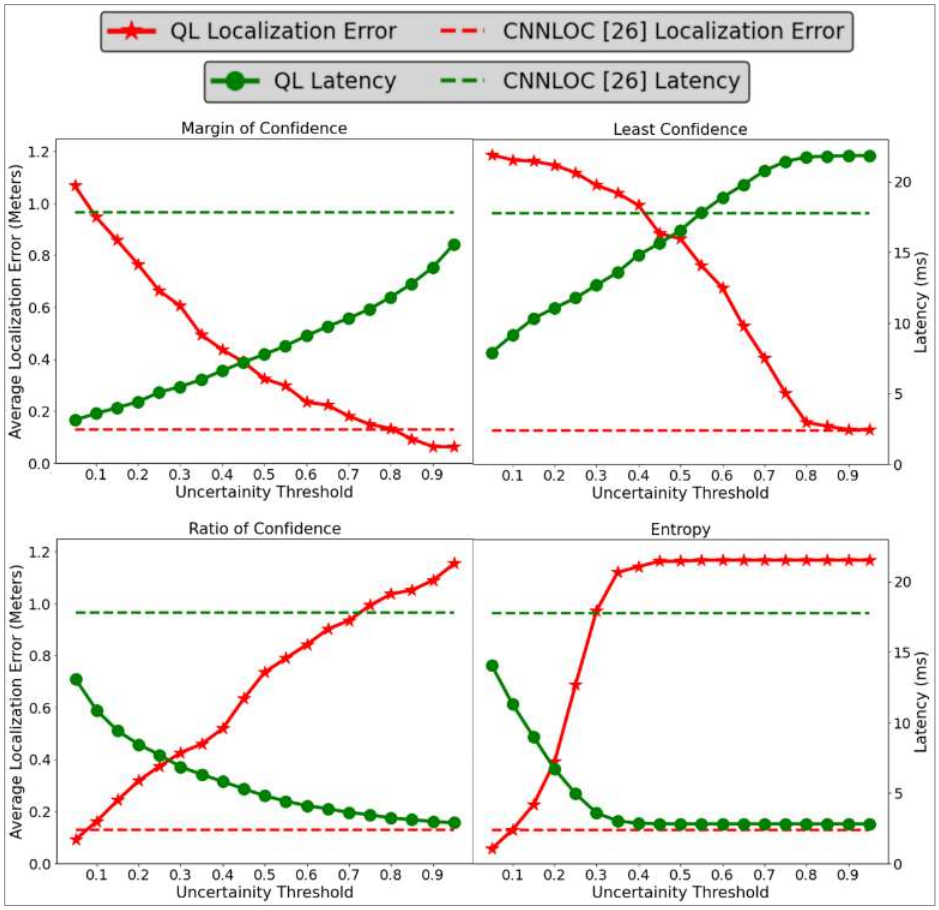


Figure 56. A comparison of average localization errors, in meters, and prediction latency across four uncertainty sampling techniques for QuickLoc (QL) as compared to the baseline CNNLOC framework.

6.7.2. SENSITIVITY ANALYSIS ON DEVICE HETEROGENEITY

Next, we explored the impact of device heterogeneity on achievable latency and localization error for QuickLoc as compared to the non-early exit model in CNNLOC [38].

Each curve in Figure 57 depicts the variation in achievable localization error with its associated latency. The curves are captured by varying the threshold parameter of the margin of confidence uncertainty sampling method across both EEA1 and EEA2. The star markings denote the baseline prediction latencies across various devices. We can make two observations from Figure 57. First, we note that for the devices excluding GS6, there exist several threshold values in QuickLoc that will produce significant reductions in latency and improved localization error. The reduction in localization error can be attributed to the enhanced learning capabilities introduced by the shorter exit paths that lead to fewer mispredictions due to the vanishing gradient problem. The second observation is that the user can achieve an exponential reduction in localization error by trading off some latency at run-time.

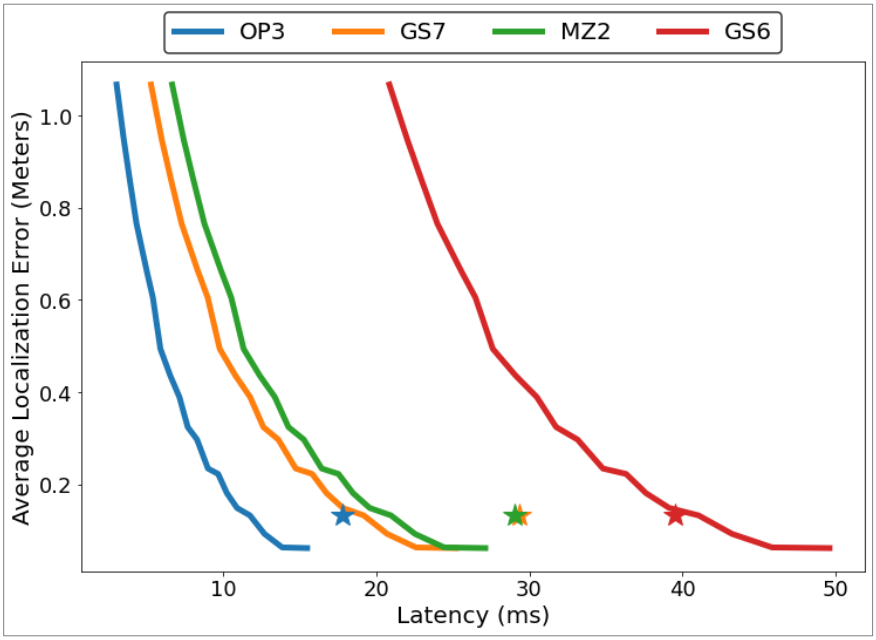


Figure 57. Achievable localization error with respect to prediction latency for four mobile devices. Baseline localization error and latency for each device is marked by the star symbol (the green and orange stars overlap).

Unfortunately, in this analysis QuickLoc is unable to achieve any improvement in latency for the GS6 device. However, it is important to note that Figure 57 only presents results for QuickLoc configurations where both EEA1 and EEA2 are enabled. As we observe from the results in the next subsection, there may be other configurations that deliver better results.

6.7.3 ANALYSIS OF EARLY EXIT PATH CONFIGURATION

Figure 58 presents the best achievable latency for each mobile device under various early exit configurations while meeting the baseline accuracy target requirements. We found that the best results (least latency) for each device are achieved when EEA1 is disabled (i.e., only EEA2 is enabled) as denoted by the green bars.

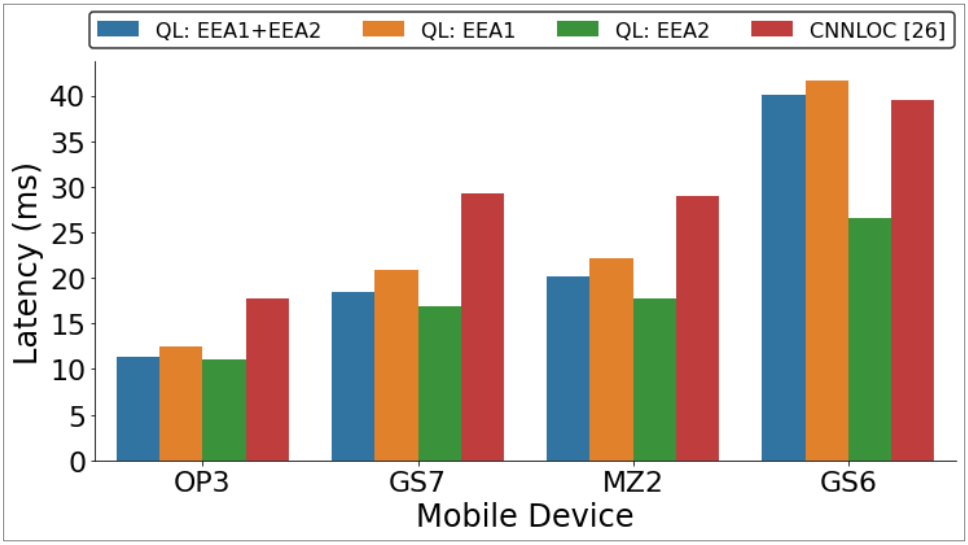


Figure 58. QuickLoc (QL) device performance under various early exit branch configurations.

In case of the GS6 device, we observe that there are no latency improvements when both the early exit paths are enabled (EEA1+EEA2) compared to CNNLOC. The localization latency further degrades when only EEA1 is enabled for GS6. It is important to note that while different

EEA configurations had no impact on the OP3 device, it had a significant impact on the GS6 device. This observation highlights the significance of having multiple EEA paths that can be configured for an unknown device in the online phase.

6.7.4 ANALYSIS OF INFERENCE ENERGY

The variation in smartphone specifications can greatly impact the energy required to perform a given task. Further, as smartphones are energy constrained devices that run on batteries, prediction latency alone does not dictate framework efficiency. To better highlight the energy savings (energy efficiency) of QuickLoc, we profiled the energy required per location prediction (inference energy) across various smartphones and QuickLoc configurations. The results of this analysis are shown in Figure 59.

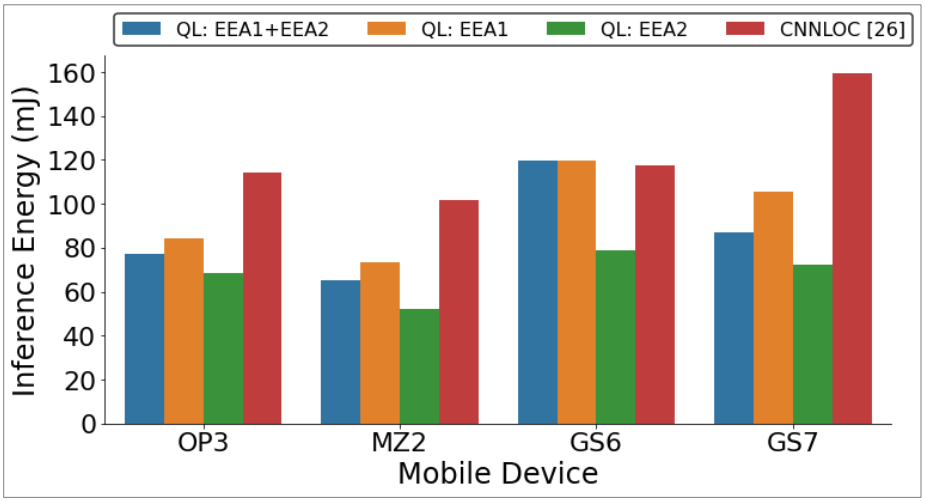


Figure 59. QuickLoc (QL) inference energy under various early exit branch configurations.

Figure 59, we observe QL: EEA2 consumes the least energy across all devices, as in Figure 58. This is because of the large number parameters in the EEA1 output layer that need to be processed every time and are held in memory as the model attempts to exit at EEA1. However, the

per-device inference energy trends do not follow the inference latency trends from Figure 58. Through Figure 59, we observe that the MZ2 device consumes the least energy per prediction as opposed to the OP3 device which has the fastest inference time. In general, we observe up to 45% reduction of inference energy (GS7) with QuickLoc as compared to baseline model.

6.7.5. ANALYSIS ON MEMORY FOOTPRINT

In this subsection, we present an analysis of the memory overhead of QuickLoc under various configurations. As the model utilized by QuickLoc has additional layers compared to the baseline work in CNNLOC [38], there is an increase in the memory required to deploy the model on a smartphone.

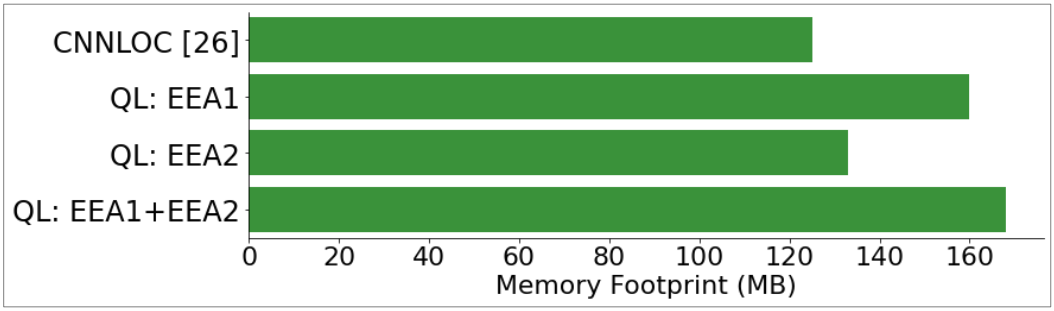


Figure 60. QuickLoc memory footprint with respect to CNNLOC.

Figure 60 describes the memory footprint of QuickLoc under various early exit configurations as compared to CNNLOC [38]. The most notable observation from Figure 60is the 25% increment in memory footprint when both the early exit branches are enabled (QL: EEA1+EEA2). This is followed by QL: EEA1 which has a 22% increment in memory footprint as compared to CNNLOC [38]. This behavior is mainly attributed to the very large number of parameters in the output later of EEA1 (9.2M parameters) as compared to EEA2 (1.9M parameters). QL: EEA2 only incurs a 3% increase in memory footprint and is therefore the most

favorable configuration in general, based on experiments performed in the previous sub-section. From this point onward, we use QL: EEA2 as the default configuration for QuickLoc when comparing it against prior works (Section 6.8.6).

While the memory footprint for QuickLoc is always expected to be higher than the baseline model, the specific increase we observe in our experiments is highly dependent on various factors such as original model complexity, number of early exit paths (or branches) enabled at the time of deployment, and the layer hyperparameters on each early exit path. Due to these factors, we advise caution when adapting ideas from QuickLoc into other model designs.

6.8.6. OVERALL QUICKLOC PERFORMANCE

Figure 61 and Figure 62 describe the accuracy and latency of QuickLoc (QL: EEA2 variant) as compared to CNNLOC [38], and non-deep machine learning frameworks that employ support vector regression (SVR) [92], and K-nearest Neighbor (KNN) [37]. As we do not cover the impact of device heterogeneity on model accuracy in this work, the results are only presented for the OP3 smartphone. We also utilize the same configuration of QuickLoc (QL: EEA2 with $\theta_{\mathrm{US2}} = 0.82$) for both the accuracy and latency results.

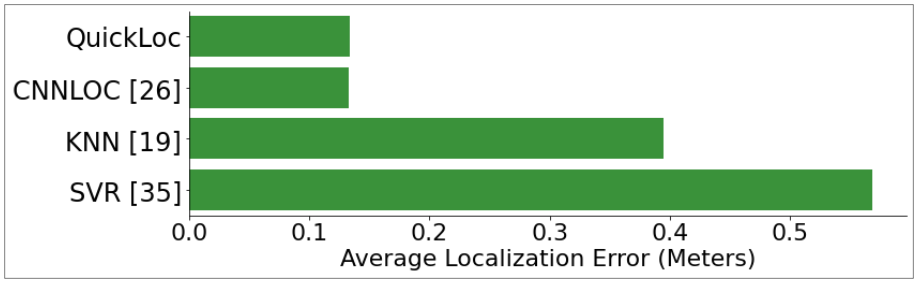


Figure 61. The average localization error in meters for various indoor localization frameworks.

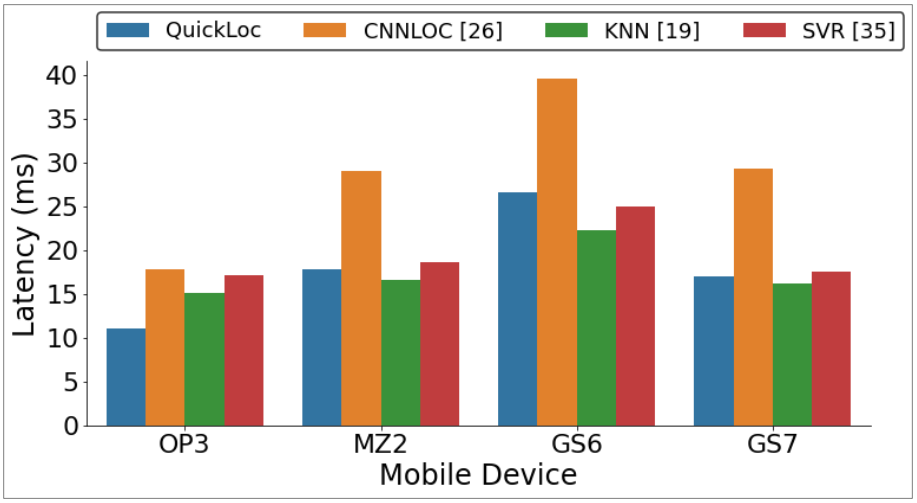


Figure 62. The prediction latency of various indoor localization frameworks with respect to QuickLoc.

From Figure 61, we observe that both CNNLOC [38] and QuickLoc deliver a considerable localization accuracy improvement over KNN [37] and SVR [92]. From the analysis presented in Figure 62, we observe that through QuickLoc we are able to achieve up to 42% reduction in prediction latency (GS7) while maintaining our target baseline localization accuracy (0.13 meters) achieved through CNNLOC [38]. Further, QuickLoc enables us to achieve inference latencies comparable to relatively light-weight non-deep learning indoor localization frameworks in most cases, while outperforming them on the OP3 device. The reason for QuickLoc having lower latency than KNN and SVR on the OP3 device is not entirely clear. We believe that the hardware on the OP3, specifically the DRAM, is geared towards faster and better locality-exploiting burst I/O modes at a cost of higher current draw (1750 mA; in contrast the GS7 only required an average current draw of 1300 mA), which may explain the lower latency for QuickLoc's access patterns on the OP3 device.

In summary, the QuickLoc indoor localization framework presented in this work significantly improves prediction latency without any loss in localization accuracy across

smartphones and indoor locales. Further, it enables a new form of run-time adaptiveness for deep-learning-based indoor localization frameworks that trades-off localization accuracy, inference latency, and energy against run-time memory footprint.

6.8. CONCLUSIONS

In this chapter, we presented an in-depth analysis of a deep learning based indoor localization framework that is expected to deliver accurate results on various mobile devices in real-time. Our analysis highlighted the significant lack of consistent performance across varying deep learning model depths and across diverse mobile devices. To overcome this challenge, we proposed the novel QuickLoc framework, that is able to adapt the localization latency for the target device through early exit strategies and reduce average localization error at the same time.

7. SIAMESE NEURAL ENCODERS FOR LONG-TERM INDOOR LOCALIZATION WITH MOBILE DEVICES

Contemporary geo-location services have eliminated the need for burdensome paper-based navigational maps that were dominant in the past. Owing to the localization technologies of today, our physical outdoor reality is now augmented by an additional layer of virtual map-based reality. Such a revolutionary shift has dramatically changed many aspects of human experience: geo-location data is now used for urban planning and development (roads, location of hospitals, telecom network design, etc.), augmented reality video games (Pokémon Go, Ingress Prime) and has even helped realize entirely new socio-cultural collaborations (Facebook marketplace, Meetup, etc.) [149].

Unfortunately, due to the limited permeability of GPS signals within indoor environments, such services cannot be easily extended into buildings such as malls, hospitals, schools, airports, etc. Indoor localization services can pro-vide immense value, e.g., during emergency evacuations or when locating people indoors in need of critical medical attention. In the future, such services could inform the architects of building design and make augmented indoor living a reality. Towards this goal, indoor localization is experiencing a recent upsurge in interest [2], including from industry (e.g., Google [4], Apple [3]).

Although substantial progress has been made in this area (see section 7.2), recent works suggest fingerprinting-based indoor localization as the most favorable solution [2] [31] [38] [123] [150] [151] [152]. While any form of radio fingerprinting works, the ubiquitous deployment of WiFi Access Points (APs), and the superior localization accuracies achieved through it make WiFi the clear choice of radio infrastructure to support in-door fingerprinting.

Conventionally, fingerprinting-based indoor localization consists of two phases. The first phase, known as the offline phase, comprises of capturing WiFi signal characteristics, such as RSSI (Received Signal Strength Indicator) at various indoor locations or Reference Points (RPs) in a building. The RSSI values from all APs observable at an indoor RP can be captured as a vector and represents a fingerprint associated with that RP. Such fingerprints collected across all RPs form a dataset, where each row in the dataset consists of an RSSI fingerprint along with its associated RP location. The collection of fingerprints to form the dataset is known to be a very time-consuming endeavor [37]. Consequently, publicly available datasets only contain a few fingerprints per RP (FPR). Using such datasets, a machine learning (ML) model can be trained and deployed on mobile devices (e.g., smartphones) equipped with WiFi transceivers. In the second phase, called the online phase, WiFi RSSI captured by a user is sent to the ML model running on the user-carried device, and used to compute and then update the user's location on a map of the indoor locale on the user's device display, in real time. Deploying such models on the user device instead of the cloud enables better data privacy, security, and faster response times [2].

Recent works report improved indoor localization accuracy through the use of deep learning-based classifiers [31] [38]. This is attributed to their superior ability at discerning underlying patterns within fingerprints. Despite these improvements, factors such as human activity, signal interferences, changes to furniture and materials in the environment, and also removal or replacement of WiFi APs (in the online phase) introduce changes in the observed RSSI fingerprints over time that can degrade accuracy [123] [151] [152]. For instance, our experiments suggest that in frameworks designed to deliver mean indoor localization error of 0.25 meters, these factors degrade error to as much as 6 meters (section 7.4.) over a short period of 8 months. Most prior

efforts in the indoor localization domain often overlook the impact of such temporal variations during the design and deployment stages, leading to significant degradation of accuracy over time.

In this chapter, we introduce STONE, a framework that delivers stable and long-term indoor localization with mobile devices, without any re-training. The main contributions of this work are:

- Performing an in-depth analysis on how indoor localization accuracy can vary across different levels of temporal granularity (hours, days, months, year);
- Adapting the Siamese triplet-loss centric neural encoders and proposing variation-aware fingerprint augmentation for robust fingerprinting-based indoor localization;
- Developing a floorplan-aware triplet selection algorithm that is crucial to the fast convergence and efficacy of our Siamese encoder-based approach;
- Exploring design tradeoffs and comparing STONE with state-of-the-art indoor localization frameworks.

7.1. BACKGROUND AND RELATED WORK

Broadly approached, indoor localization methodologies can be classified into three categories: (i) static propagation model-based, (ii) triangulation/trilateration-based, and (iii) fingerprinting-based [2]. Static propagation modeling approaches depend on the correlation between distance and WiFi RSSI gain, e.g., [13]. These techniques are functionally limited to open indoor areas given that multipath or shad-owing effects of signals attributed to walls and other indoor obstacles are not considered. These methods also required the cumbersome creation of a gain model for each individual AP. Triangulation/Trilateration-based methods use geometric properties such as the distance between multiple APs and the mobile device [54] (trilateration) or the angles at which signals from two or more APs are received [56] (triangulation). While such

methodologies may be resistant to mobile device specific variability (device heterogeneity), they are not resilient to multipath and shadowing effects [31]. As discussed in Section 7.1, WiFi fingerprinting-based approaches associate sampled locations (RPs) with the RSSI captured across several APs [31] [32] [38] [66] [123] [150] [151] [152]. These techniques are known to be resilient to multi-path reflections and shadowing as the RP fingerprint captures the characteristics of these effects leading to more accurate localization than with the other two approaches.

Fingerprinting generally employs ML to associate WiFi RSSI captured in the online phase to the ones captured at the RPs in the offline phase [81] [153]. Recent work on im-proving WiFi fingerprinting exploits the increasing computational capabilities of smartphones. For instance, Convolutional Neural Networks (CNNs) have been proposed to im-prove indoor localization accuracy on smartphones [31] [38] [154]. One major concern with fingerprinting is the enormous effort required to manually collect fingerprints for training. Openly available fingerprint datasets often only consist of a few fingerprints per RP [152]. This motivates the critical need for indoor localization frameworks that are competitive with contemporary deep-learning-based frameworks but require fewer fingerprints to be deployed.

An emerging challenge for fingerprinting-based indoor localization (especially WiFi-based) arises from the fluctuations that occur over time in the RSSI values of APs [32] [123] [151] [155], [156]. Such temporal-variations in RSSI arise from the combination of many environmental factors, such as human movement, radio interference, changes in furniture or equipment placement, etc. This issue is further intensified when WiFi APs are removed or replaced by network administrators, changing the underlying fingerprint considerably [37]. This leads to a catastrophic loss in localization accuracy over time (discussed in section 7.4).

The most straightforward approach to overcome temporal variation is to capture a large number of fingerprints over a long period of time (in offline phase). A deep-learning model trained using such a dataset would demonstrate resilience to degradation in localization accuracy as it witnesses (learns) the temporal fluctuations of RSSI values at various RPs. The work in [151] proposes such an approach by training an ensemble of models with fingerprints collect-ed over a period of several hours. The authors then take a semi-supervised approach, where the models are refit over weeks using a mix of originally collected labeled fingerprints and pseudo-labeled fingerprints generated by the models. This process is repeated over several months to demonstrate the strength of this approach. However, the collection of fingerprints at a high granularity of RPs (small distance be-tween RPs) over a long period of time in the offline phase is not scalable in practice.

To overcome the challenge of lack of available tempo-rally diverse fingerprints per RP, the authors in [155] propose a few-shot learning approach that delivers reliable accuracy using a few fingerprints per RP. The contrastive loss-based approach prevents the model from overfitting to the training fingerprints used in the offline phase. Unfortunately, their approach is highly susceptible to long-term temporal variations and removal of APs in the online phase. This forces the authors to recalibrate or re-train their model using new fingerprints every month.

Attempting to achieve calibration-free indoor localization, some researchers propose the standardization of fingerprints into a temporal-variation resilient format [123] [157]. One such approach, known as GIFT [123], utilizes the difference between individual AP RSSI values to form a new finger-print vector. However, instead of being associated with a specific RP, each GIFT fingerprint is associated with a specific user movement vector from one RP to another. How-

ever, GIFT degrades in accuracy over the long-term and is also highly susceptible to removal of APs (section 7.4).

Considering the general stability of simple non-parametric approaches over the long term, such as K-Nearest-Neighbor (KNN), the authors in [158] propose Long-Term KNN (LT-KNN), which improves the performance of KNN in situations where several APs are removed. However, LT-KNN fails to deliver the superior accuracies promised by deep-learning approaches and needs to be re-trained on a regular basis.

In summary, most indoor localization solutions are simply unable to deliver stable localization accuracies over time. The few prior efforts that aim to achieve stable long-term localization either require large amounts of fingerprints per RP captured over time, or frequent re-training (refitting) of the model using newly collected fingerprints. Our proposed STONE framework provides a long-term fingerprinting-based indoor localization solution with lower overhead and superior accuracy than achieved by prior efforts in the do-main, without requiring any re-training.

7.2. SIAMESE NETWORK AND TRIPLET LOSS: OVERVIEW

A Siamese network is a few-shot learning (requiring few labeled samples to train) neural architecture containing one or more identical networks [159] [160]. Instead of the model learning to associate an input image with a fixed label (classification) through an entropy-based loss function, the model learns the similarity between two or more inputs. This prevents the model from overfitting to the relationship be-tween a sample and its label. The loss function for a Sia-mese network is often a Euclidean-based loss that is either contrastive [159] or triplet [160].

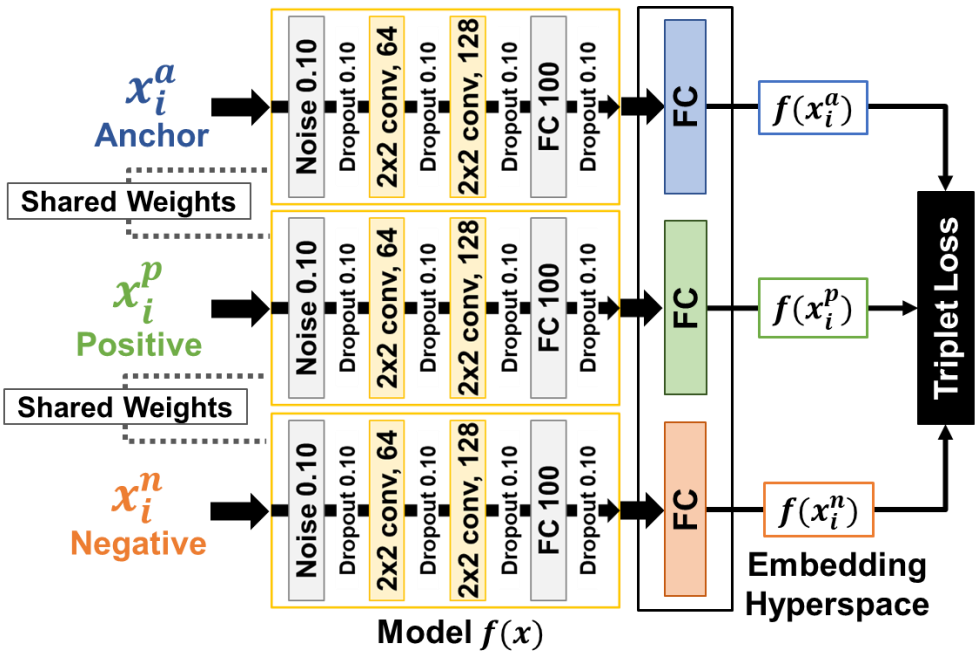


Figure 63. An example architecture of a Siamese encoder with triplet loss. A single CNN network is used, i.e., all the models share the same weights.

A Siamese network encoder using contrastive loss was proposed in DeepFace [159] for facial recognition. DeepFace focuses on encoding the input faces such that they are either pushed together or pulled apart in the embedded space based on whether they belong to the same person or not. The work in FaceNet [160] further improved on this idea using triplet loss that simultaneously pushes together and pulls apart faces of the same person and different persons, respectively.

An architectural representation of the Siamese model used in *STONE* (inspired by FaceNet) is presented in Figure 63. The Siamese network consists of a single deep neural architecture. Note that given the specific model details (covered in section 7.3.4), the model itself can be treated as a black-box system.

The model in Figure 63 can be represented as $f(x) \in R^d$ that embeds an image x into a d-dimensional Euclidean embedding space. Therefore, the images x_i^a(anchor), x_i^p(positive) and

x_i^n(negative) are embedded to form encodings $f(x_i^a), f(x_i^p)$ and $f(x_i^n)$ respectively, such that they belong in the same d-dimensional embedded hyperspace, i.e., $||f(x)||_2 = 1$. The anchor in a triplet is the reference label's sample with respect to which other label's samples are selected for the triplet. The triplet-based approach enables few-shot learning, as a single input to the training process is a combination of three different samples. Given a training set of *k*-classes and *n*-samples, the conventional classification approach [31] [38] [54] has a total of $k \times n$ samples to learn from. In contrast, the triplet loss-based approach has 3 samples per input, where each sample can be selected in $k \times n$ ways, i.e., a total of $(k \times n)^3$ inputs generated from the same dataset.

The goal of the overall Siamese encoder is to ensure that the anchor image is closer to all other images of the same label (positives), than it is to any image of other labels (negatives). Based on this discussion, the embeddings should satisfy equation (12)

$$\left\|f(x_i^a) - f\left(x_i^p\right)\right\|_2^2 \leq \|f(x_i^a) - f(x_i^n)\|_2^2 \tag{12}$$

However, it is important to note that equation (12) can be trivially solved if $f(x) = 0$. Therefore, the margin α is introduced to enforce the stability of equation (12). Finally, the triplet loss function $L(x_i^a, x_i^p, x_i^n)$ that is to be minimized is given as:

$$L = \left\|f(x_i^a) - f\left(x_i^p\right)\right\|_2^2 - \|f(x_i^a) - f(x_i^n)\|_2^2 + \alpha \leq 0 \tag{13}$$

The authors of FaceNet [160] remark that to achieve rapid convergence it is important to select triplets that violate the constraint in equation (13). Thus, for each triplet, we need to select a hard-positive x_i^p that poses great dissimilarity with the anchor, and a hard-negative x_i^n that poses great similarity with the anchor x_i^a. This may require the selection of triplets that satisfy both:

$$argmax_{x_i^p} \left\| f(x_i^a) - f\left(x_i^p\right) \right\|_2^2,$$
$$argmin_{x_i^n} \| f(x_i^a) - f(x_i^n) \|_2^2 \quad (14)$$

Evaluating $argmin$ and $argmax$ across the whole dataset is practically infeasible. To overcome this challenge, we present a novel and low-complexity indoor localization domain-specific approach for triplet selection in Section 7.3.5.

Once the embeddings for the training dataset have been produced, the embeddings and associated labels can be used to formulate a non-parametric model such as KNN. Later, this KNN model combined with the encoder can be used to classify an unlabeled sample as a known label.

Based on our discussion above, there are three salient features of Siamese networks that fit well to the challenges of long-term fingerprinting-based indoor localization: *(i)* Instead of associating a sample to its label, it learns the relationship between the samples of labels, *(ii)* Learning relationships between samples promotes generalization and suppresses the model's tendency to overfit the label-sample relationship, and *(iii)* It requires fewer samples per class/label to achieve good performance (few-shot leaning). Siamese networks will tend to avoid overfitting the training fingerprints and can minimize the offline fingerprint collection effort. The next section describes our framework that takes this approach for learning and classifying fingerprints.

7.3. STONE FRAMEWORK

7.3.1. OVERVIEW

A high-level overview of the proposed framework is presented in Figure 64. We begin in the offline phase (annotated by red arrows), where we capture RSSI fingerprints for various RPs across the floorplan. Each row in the fingerprint dataset consists of the RSSI values for each AP visible

across the floorplan and its associated RP. These fingerprints are used to train the Siamese encoder depicted in Figure 63. Once the Siamese encoder is trained, the encoder network itself is then used to embed the RSSI fingerprints in a *d*-dimensional hyperspace. The encoding of each RSSI vector and its associated RP, from the offline phase, form a new dataset. This new dataset is then used to train a non-parametric model. For our work, we chose the KNN classifier. At the end of the offline phase, the Siamese encoder and the KNN model are deployed on a mobile device.

In the online phase (green arrows), the user captures an RSSI fingerprint vector at an RP that is unknown. For any WiFi AP that is not observed in this phase, its RSSI value is assumed to be -100, ensuring consistent RSSI vector lengths across the phases. This fingerprint is pre-processed (see Section IV.B) and sent to the Siamese model. The encoding produced is then passed on to the KNN model, which finally predicts the user's location.

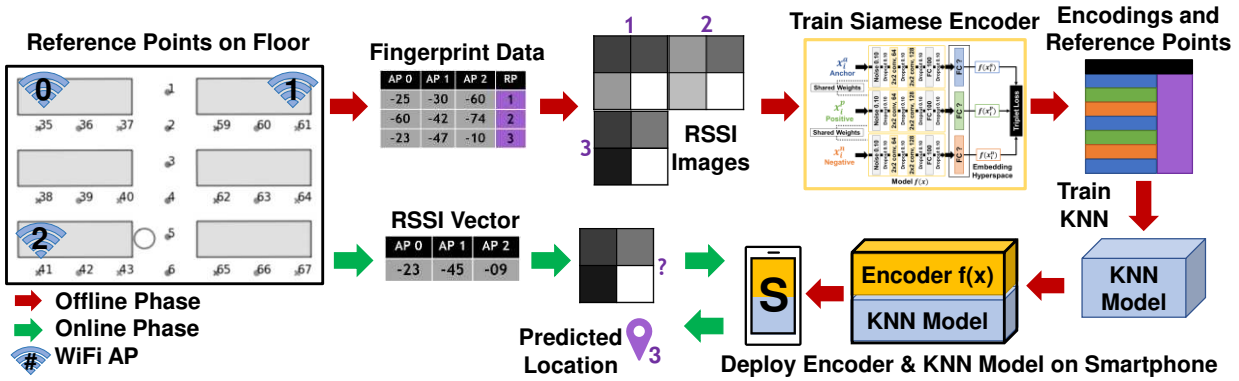


Figure 64. An overview of the STONE indoor localization framework depicting the offline (red arrows) and online (green arrows) phases.

In the following subsections, we elaborate on the main components of the *STONE* framework shown in Figure 64.

7.3.2. RSSI FINGERPRINTING PREPROCESSING

The RSSI for various WiFi APs along with their corresponding RPs are captured within a database as shown in Figure 64. The RSSI values vary in the range of -100 to 0 dB, where -100 indicates no signal and 0 indicates a full (strongest) signal. The RSSI values captured are then normalized to a range of 0 (weakest) to 1 (strongest) signal. Finally, each RSSI vector is padded with zeros such that the length of the vector reaches its closest square. Each vector is then reshaped as a square image. This process is similar to the one covered by the authors in [31]. At this stage, in the offline phase, we have a database of fingerprint images and their associated RPs, as shown in Figure 64.

7.3.3. LONG-TERM FINGERPRINT AUGMENTATION

A major challenge to maintain long-term stability for fingerprinting-based indoor localization is the removal of WiFi APs post-deployment (i.e., in the online phase) [152]. In the offline phase, it would be impossible to foretell which specific APs may be removed or replaced in the future. In the *STONE* framework, once an AP is removed or replaced, its RSSI value is set to -100. This translates into a pixel turning off in the input fingerprint image. *STONE* enables long-term support for such situations by emulating the removal of APs (turning off pixels of input images). When generating batches to train the Siamese encoder, we randomly set the value of a percentage of observable APs (*p_turn_off*) to 0. The value of *p_turn_off* is picked from a uniform distribution as described by:

$$p\_turn\_off = U(0.0, p\_upper) \tag{15}$$

where, *p_upper* is the highest percentage of visible APs that can be removed from a given fingerprint image. For the experiments in section VI, we chose an aggressive value of *p_upper=0.90*.

7.3.4. CONVOLUTIONAL NEURAL ENCODER

Given the superior pattern learning abilities of CNNs, we employ stacked convolutional layers to form the Siamese encoder. An architectural overview of the encoder is shown in Figure 63. We use 2 convolutional layers (conv) with filter size of 2×2 with the stride set to 1 and consisting of 64 and 128 filters, respectively. They are followed by a fully connected (FC) layer of 100 units. The length of the embedding (encoder output or last layer) was empirically evaluated for each floorplan independently. Based on our analysis, we chose a value for this hyperparameter in the range of 3 to 10. To enhance the resilience of STONE to short-term RSSI fluctuations, Gaussian noise ($\sigma = 0.10$) is added to the model input (as shown in Figure 63). Dropout layers are also interleaved between convolution layers to improve generalizability of the encoder. It is important to note that while the presented convolutional architecture functions well for our experiments and selected datasets, it may need slight modifications when porting to other datasets with a different feature space.

7.3.5. FLOORPLAN-AWARE TRIPLET SELECTION ALGORITHM

The choice of samples selected to form the triplets have a critical impact on the efficacy of the training and accuracy of the Siamese encoder. For a limited set of available fingerprints per RP (6-9 in our experiments), there are very few options in selecting a hard-positive. However, given an anchor fingerprint, selecting a hard-negative is a greater challenge due to the large number

of candidate RPs across the floorplan. The motivation for our proposed triplet selection strategy is that RPs that are physically close to each other on the floorplan would have RSSI fingerprints that are the hardest to discern. This strategy is specific to the domain of fingerprinting-based indoor localization as the additional information of the relationship between different labels (location of labels with respect to each other) may not be available in other domains (such as when comparing faces).

To implement our hard-negative selection strategy, we first pick an RSSI fingerprint from an anchor RP, chosen at random. For the given anchor RP_a, we then select the negative RP_n using a probability density function. Given the set of all K RPs, $\{RP_1, RP_2, \dots RP_k\}$, the probability of selecting the i^{th} RP as the hard-negative candidate is given by a bivariate Gaussian distribution around the anchor RP as described by the expression:

$$P(RP_i) \sim N_2(\mu_a, \sigma), \quad s.t. P(RP_a) = 0 \tag{16}$$

where $P(RP_i)$ is the probability of selecting it as the hard-negative and N_2 represents a bivariate Gaussian probability distribution that is centered around the mean at the anchor (μ_a). However, another anchor fingerprint should never be chosen as the hard-negative, and therefore we set the probability of selecting an anchor to zero. The expression in (16) ensures that the RPs closest to the anchor RP have the highest probability of being sampled. This probability then drops out as we move away from the anchor. The bivariate distribution is chosen based on the assumption that the indoor environment under test is two-dimensional (a single floor). Once the anchor and the negative RPs are identified for a given triplet, the specific RSSI fingerprint for each is randomly chosen. This is because we have only a few fingerprints per RP, and so it is easy to cover every combination.

The proposed triplet selection strategy is subsequently used to train the Siamese model as discussed in Section IV.A, whose output is then used to train the KNN model in the offline phase.

In the online phase, the encoder and the KNN model are deployed on the mobile device and used to locate the user on the floorplan, as illustrated in Figure 64(lower half).

7.4. EXPERIMENTS

7.4.1. EXPERIMENTAL SETUP

We evaluated the effectiveness of *STONE* across three large indoor paths derived from a publicly available dataset as well as based on our own measurement across multiple buildings. The next two subsections describe these paths, while the last subsection summarizes prior work that we compare against.

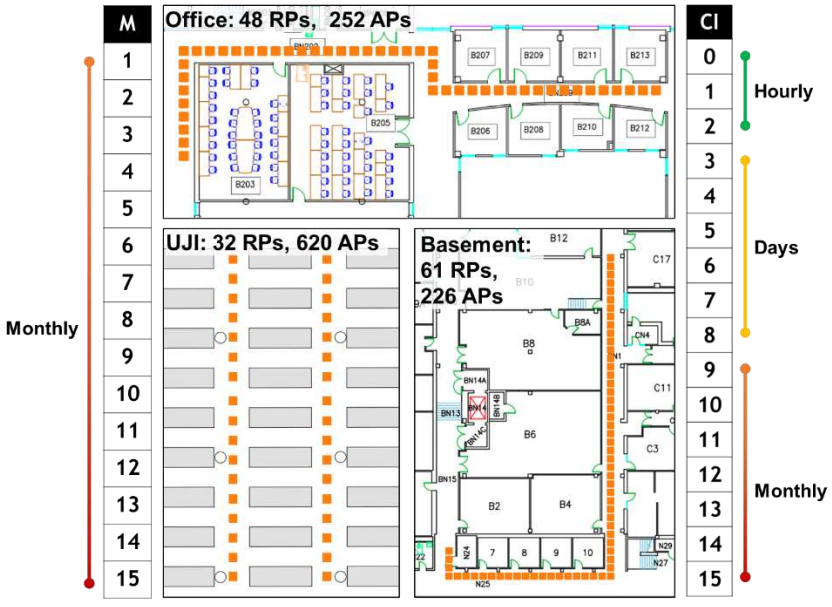


Figure 65. Indoor floorplans for long-term indoor localization evaluation, annotated with number of visible WiFi APs along the paths and RPs along the paths. Vertical scales show temporal granularities across months (left-UJI) and collection instances (right-Basement and Office).

7.4.1.1. FINGERPRINTING TEST SUIT: UJI

STONE was evaluated on the public dataset UJI [152]. This dataset covers two floors within a library. However, due to high floorplan similarity across the two floors, we present the results for floor 3, for brevity. The dataset consists of fingerprints that are collected for the RPs along paths, with multiple fingerprints per RP that are collected at different instances of time. We utilize RPs from the dataset for which the fingerprints (up to 9) were collected on the same day for training the models we compared. The data from the following 15 months is used for testing. The UJI floorplan we considered is presented in Figure 65 (bottom left of the figure). The RPs on the floorplan form a grid like structure over a wide-open area, which is different from the corridors evaluated for the Basement and Office indoor paths, discussed next.

7.4.1.2. FINGERPRINTING TEST SUITE: OFFICE AND BASEMENT

We also evaluated *STONE* at finer and broader granularity levels of hours, days, and months. The floorplan and associated details for these paths, captured from real buildings accessible to us, are presented in Figure 65. The fingerprints were captured from two separate indoor spaces: Basement (61 meters in length) and Office (48 meters in length). An LG V20 mobile device was used to capture fingerprints along paths. While the Office path fingerprints are captured in a section of a building with newly constructed faculty offices, the basement path is surrounded by large labs that contain heavy metallic equipment. The Office and Basement paths are thus unique with respect to each other (and also the UJI path) in terms of environmental noise and multipath conditions associated with the paths. Each measured fingerprint location is annotated by an orange dot (Figure 65) and measurements are made 1 meter apart. A total of 6 fingerprints were captured per RP at each collection instance (CI), under a span 30 seconds. The first three CIs (0–2), for both paths were on the same day, with each CI being 6 hours apart. The intention was to capture the effect of

varying human activity across different times in the day; thus, the first CI is early in the morning (8 A.M), the second at mid-day (3 P.M), and the third is late at night (9 P.M). The following 6 CIs (3–8) were performed across 6 consecutive days. The remaining CIs (9–15) were performed on the following months, i.e., each was ≈30 days apart.

Figure 66 depicts the ephemerality of WiFi APs on the Basement and Office paths across the 16 CIs (CIs:0–15 over a total span of 8 months). A black mark indicates that the specific WiFi AP (x-axis) was not observed on the indicated CI (y-axis). While capturing fingerprints across a duration of months, we did not observe a notable change in AP visibility up to CI:11. Beyond that, ≈20% of WiFi APs become unavailable. Note that the UJI dataset shows an even more significant change in visible WiFi APs of ≈50% around month 11; however, this change occurs much sooner in our paths, at C1:11, which corresponds to month 4 after the first fingerprint collection in CI:0. For the Office and Basement paths, we utilized a subset of CI:0 (fingerprints captured early in the morning) for the offline phase, i.e., training occurs only on this subset of data from CI:0. The rest of the data from CI:0 and CIs:1–15 was used for testing.

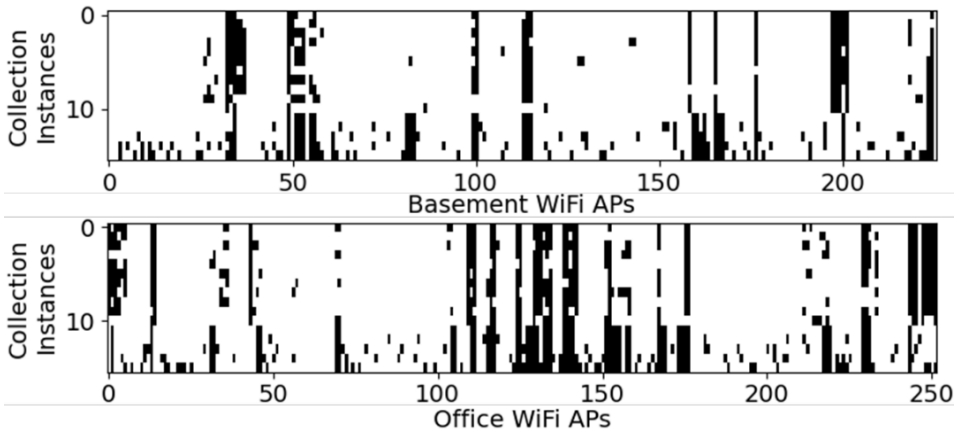


Figure 66. Ephemerality of WiFi APs across various collection instances for the Basement and the Office indoor paths.

7.4.1.3. COMPARISON WITH PRIOR WORK

We identified four state-of-the-art prior works to compare against our proposed *STONE* framework. The first work, LearnLoc or KNN [37] is a lightweight non-parametric approach that employs a Euclidean distance-based metric to match fingerprints. The technique in the work is incognizant of temporal-variation and serves as a one of the motivations for our proposed work. The second work, LT-KNN [158], is similar to [37] but has enhancements to maintain localization performance as APs are removed or replaced over time. LT-KNN achieves this by imputing the RSSI values of APs that have been removed (are no longer observable on the floorplan) using regression. The KNN model is re-trained using the imputed data to maintain localization accuracy over time. The third work, GIFT [123], achieves temporal-variation resilience by matching the change in the gradient of WiFi RSSI values as the user moves along a path on the floorplan. Fingerprint vectors are used to represent the difference (gradient) between two consecutive WiFi scans and are associated with a movement vector in the floorplan. Lastly, the fourth work, SCNN [31], is a deep learning-based approach that has been designed to sustain stable localization accuracy in the presence of malicious AP spoofing. While SCNN is not designed to be temporally resilient, it is intended to maintain accuracy under the conditions of high RSSI variability. This makes SCNN an excellent candidate for our work to be compared against.

7.4.2. EXPERIMENTAL RESULTS: UJI

Figure 67 presents the mean localization error in meters (lower is better) for the proposed *STONE* framework and the four other prior fingerprinting-based indoor localization techniques across 15 months of the UJI dataset. Between months 1-2, we observe that most previous works (KNN, SCNN, LT-KNN) experience a sharp increase in localization error. Given that there is no temporal-variation in the training and testing fingerprints for month 1, previous works tend to

overfit the training fingerprints, leading to poor generalization over time. In contrast, *STONE* remains stable and delivers ≈1 meter accuracy by not overfitting to the training fingerprints in month 1. We can also observe that GIFT provides the least temporal-resilience and has the highest localization error over time. The localization errors of *STONE*, SCNN, KNN and LT-KNN are around 2 meters (or less) up to month 10, followed by a severe degradation for KNN and SCNN. The significant change in APs at month 11, as discussed earlier, negatively impacts frameworks that are not designed to withstand the AP removal-based temporal-variation. In general, *STONE* outperforms all frameworks from months 2–11 with up to 30% improvement over the best performing prior work, LT-KNN, in month 9. Owing to the long-term fingerprint augmentation used in *STONE*, it remains stable and performs very similar to LT-KNN beyond month 11. Over the entire 15-month span, *STONE* achieves ≈0.3-meter better accuracy on average than LT-KNN. *Most importantly, LT-KNN requires re-training every month with newly collected (anonymous) fingerprint samples, whereas no re-training is required with STONE over the 15-month span.*

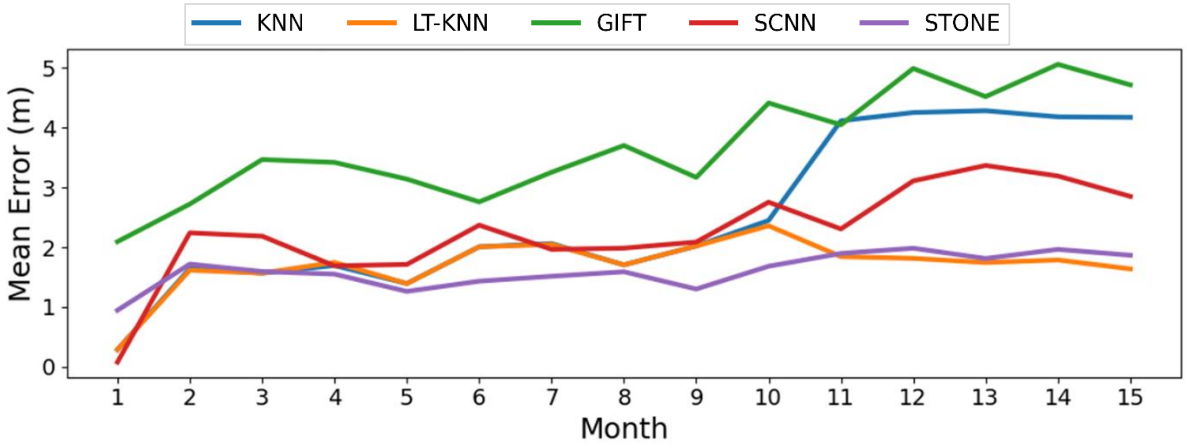


Figure 67. Comparison of localization error of various fingerprinting-based indoor localization frameworks over 15 months for the UJI indoor path.

7.4.3. EXPERIMENTAL RESULTS: OFFICE AND BASEMENT

Figure 68 depicts the contrast in mean indoor localization errors across localization frameworks for the Office and Basement indoor paths. Similar to the previous results, most frameworks (especially SCNN and GIFT) tend to overfit the training fingerprints in CI:0 followed by a sharp increase in localization error for CI:1. It is worth noting that there is merely a difference of 6 hours between CI:0 and CI:1. In contrast to previous works, *STONE* undergoes the least increase in localization error initially (CI:0–1), followed by a fairly slow increase in localization error. We observe that across both indoor paths, GIFT and SCNN tend to perform the worst overall. While both these techniques show some resilience to temporal variation at the hourly (CIs:0–2) and the daily scale (CIs:3–8), they both tend to greatly lose their efficacy at the scale of months (CIs:9–15). GIFT's resilience to very short-term temporal variation is in consensus with the analysis conducted by its authors, as it is only evaluated over a period of few hours [123]. We also note that SCNN performs worse on the Office and Basement paths, as compared to with the UJI path (previous subsection). This may be due to the larger number of classes (RPs) in the Office and Basement paths. Both KNN and LT-KNN perform well (1–2 meters of localization error) on the Basement path. However, the localization error of KNN tends to increase in later CIs, particularly on the Office path. *STONE* outperforms LT-KNN across most collection instances, including up to and beyond CI:11. *STONE* delivers sub-meter of accuracies over a period of weeks and months and performs up to 40% better than the best-known prior work (LT-KNN) over a span of 24 hours (CI:1–3 in Figure 68(b)), with superior localization performance even after 8 months. On average, over the 16 CI span, *STONE* achieves better accuracy than LT-KNN by ≈0.15 meter (Basement) and ≈0.25 meter (Office). As discussed earlier, *STONE* achieves this superior performance without requiring re-training, unlike LT-KNN which must be re-trained at every CI.

Overall, we attribute the superior temporal-variation resilience of *STONE*, to our floorplan-aware triplet selection, long-term AP augmentation, and also the nature of Siamese encoders that learn to differentiate between inputs instead of learning to classify a specific pattern as a label is also credited.

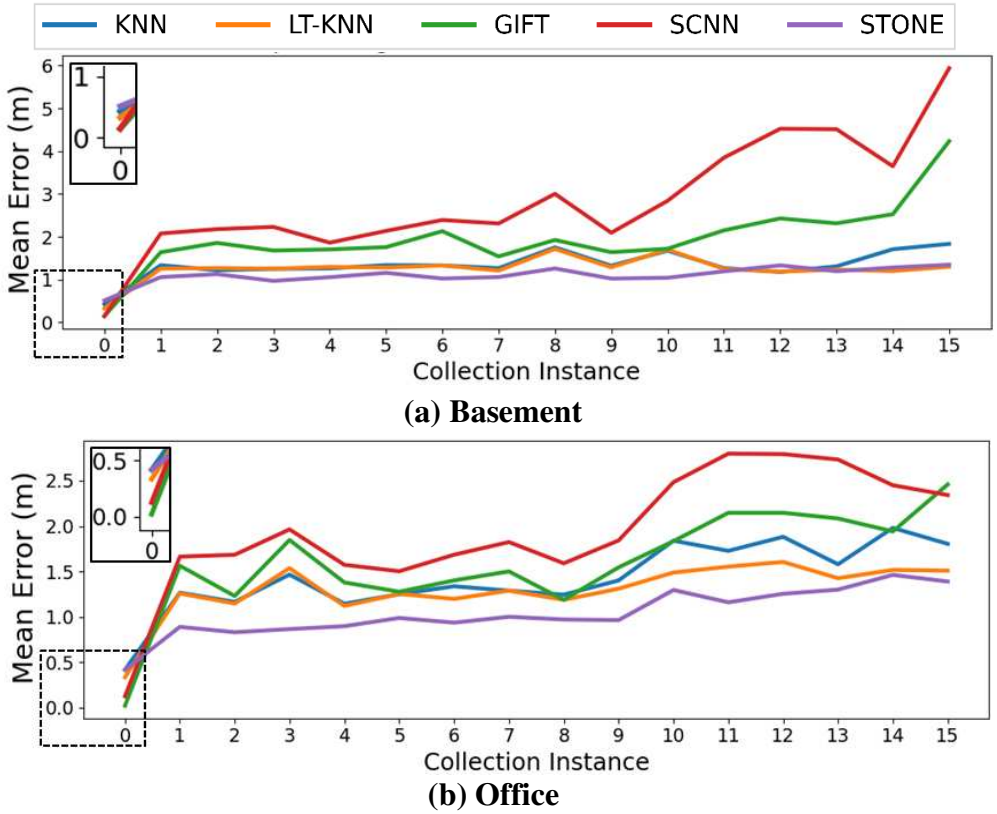


Figure 68. Localization errors of various frameworks over CIs for the Basement and Office indoor paths. Results for CI:0 are enlarged in the inset.

7.4.4. RESULTS: SENSITIVITY TO FINGERPRINTS PER RP

Considering that *STONE* is explicitly designed to deliver the best temporal-resilience using minimal fingerprints, we performed a sensitivity analysis by varying the number of fingerprints per RP (FPR) across all indoor paths considered, to study its impact on localization error. Figure 69 depicts the mean localization error as a heatmap (x-axis: timescale, y-axis: FPR) for different

variants of *STONE*, each trained using a different number of FPRs. The final column in Figure 69represents the mean localization error across the timeline. The experiment is repeated 10 times with shuffled fingerprints to avoid any form of fingerprint selection bias. From the figure, we observe that for all three indoor paths, the *STONE* framework when trained using 1 FPR performs the worst; conversely increasing FPR beyond 4 does not produce notable improvements. Overall, these results show that *STONE* is able to produce competitive indoor localization accuracy in the presence of temporal-variations using as few as 4 FPR. To contrast this with a conventional classification-based approach, SCNN [31] is deployed using as many as 8 FPR (2×) and is unable to deliver competitive localization errors over time. Moreover, mobile devices can take several seconds to capture a single fingerprint (WiFi scan), thus reducing the number of FPRs in the training phase can save several hours of manual effort.

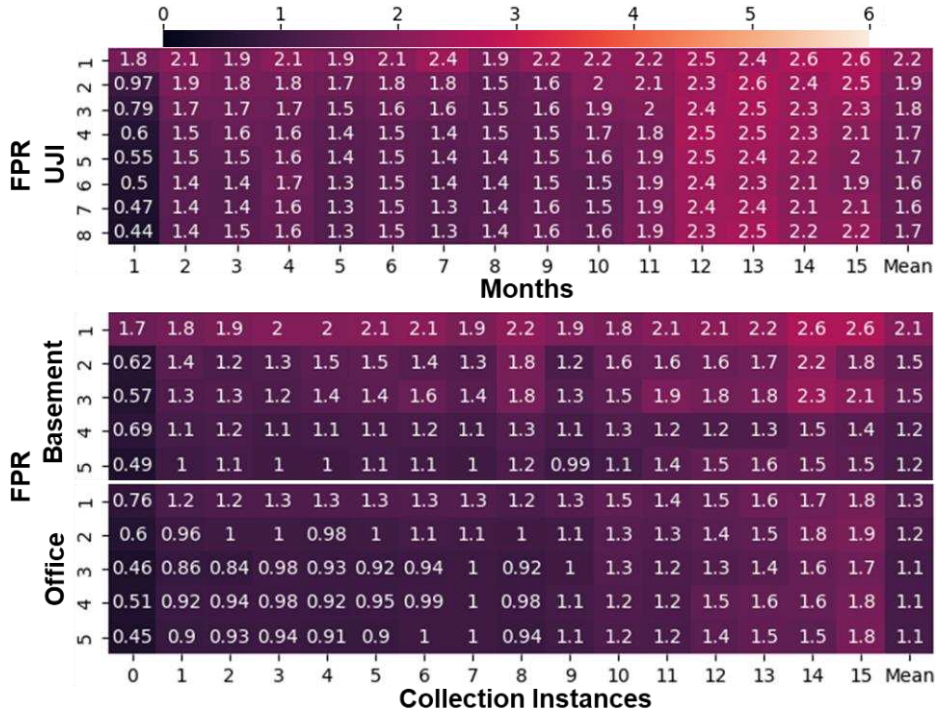


Figure 69. Sensitivity analysis on STONEs performance across varying number of fingerprints per RP (FPR) on UJI, Basement, and Office paths. Numbers in the heatmap cells show the obtained mean localization error.

7.5. CONCLUSION

In this chapter, we presented an effective temporal-variation resilient fingerprinting-based indoor localization framework called *STONE*. Our approach was evaluated against four state-of-the-art indoor localization frameworks across three distinct indoor paths. The experimental results indicate that *STONE* often delivers sub-meter localization accuracy and when compared to the best performing prior work, delivers up to 40% better accuracy over time, without requiring any re-training or model updating after the initial deployment. The ideas highlighted in this work, culminating in the *STONE* framework, represent promising directions for achieving low-overhead stable and long-term indoor localization with high-accuracy, while requiring the use of only a handful of fingerprints per reference point.

8. CONCLUSION AND FUTURE WORK SUGGESTIONS

8.1. RESEARCH CONCLUSION

In this dissertation, we present a framework for accurately and efficiently localizing people and assets within GPS-deprived indoor environments through state-of-the-art machine learning and statistical models that are suitable to be deployed on a smartphone and other similar energy-bound embedded platforms. Through critical experimental evaluations, we first identified core challenges in the domain such as device heterogeneity, temporal variations, and security that directly impact the performance of a fingerprinting-based indoor localization as described by energy-efficiency, prediction latency, accuracy and reliability of the framework. Contemporary fingerprinting-based indoor localization frameworks lack a holistic approach that can jointly tackle the aforementioned challenges. In this dissertation, we demonstrate that by using careful analysis of fingerprints and then using intelligent fingerprint augmentation methods, energy cognizant deep-learning models and a holistic approach towards framework design, we can accomplish the goals of practical indoor localization.

Towards this, we propose a real-time deep learning-based indoor localization framework (*SARTHI*) that is able to address the abovementioned challenges in a holistic manner. *SARTHI* uses a combination of (i) light-weight parametric and non-parametric pattern-matching models for to achieve exceptional localization accuracies, (ii) device heterogeneity resilient metrics, (iii) novel methodologies to overcome temporal variation through deep-learning, and (iv) generalized approaches to sensor fusion combining the aforementioned techniques with step and heading estimation. *SARTHI* employs information from several sources such as wireless signal

characteristics, inertial sensors, indoor map features, user device specific information and uncertainties from the user and their environment to deliver practical real-time indoor positioning.

Apart from these state-of-the-art advancements, *SARTHI* also recognizes and resolves previously unknown security challenges in the domain of fingerprinting-based indoor localization using deep-learning. Finally, *SARTHI* is designed bearing in mind the energy and computational limitations of embedded platforms such as smartphones. The superior advantages of *SARTHI* are validated through rigorous experimental evaluations performed against previously best-known works in the domain.

PortLoc is the first contribution of *SARTHI*. *PortLoc* presents an in-depth analysis of WiFi RSSI fingerprints. This analysis highlights the importance of using data-driven pattern matching approaches for heterogenous device-based indoor localization. Based on this analysis, computationally inexpensive metrics that can be used to compare and match fingerprints are identified. *PortLoc* is designed to be a truly portable (device heterogeneity resilient) WiFi fingerprinting-based indoor localization solution. The efficacy of *PortLoc* is evaluated on a benchmark suit containing fingerprints collected across multiple buildings using several smartphones from various vendors.

Our next contribution *SHERPA-HMM* is a hidden Markov model-based portable indoor localization framework that employs heterogeneity resilient distance metrics. The identification of such metrics generalizes the problem such that the localization accuracy for a variety non-parametric (KNN, SVM etc.) models can be improved. Further, *SHERPA-HMM* uses a lightweight software-based approach to combine noisy fingerprints over distinct smartphones and pattern matching/filtering to improve location accuracy. The proposed approach was evaluated against state-of-the-art localization techniques, across a variety of Android-based smartphones that are

used for indoor localization along paths in real buildings. The evaluations were performed for both localization accuracy and energy requirements when deployed on smartphones.

Next, we propose *CNNLOC*, a novel technique to extract images out of location fingerprints. The work adapts convolutional neural networks to the domain of indoor localization towards improving robustness and accuracy. A hierarchical architecture to scale is proposed such that the framework can be used in the real world where buildings can have large numbers of floors and corridors. *CNNLOC* is evaluated against three different state-of-the-art indoor localization frameworks from prior work. The proposed framework outperformed these approaches and delivered localization accuracy of under 2 meters.

Given the rapid adaption of deep-learning in the domain of fingerprinting-based indoor localization, for the first time, a vulnerability analysis of deep learning based indoor localization frameworks that are deployed on mobile devices, in the presence of wireless access points jamming and jamming attacks is presented. The analysis in chapter 5 revealed significant degradation of localization accuracy that can be induced by an attacker with very minimal effort. Based on the evaluations performed, a novel solution is devised to provide resilience against jamming and spoofing attacks. The secure variant of the proposed approach, *Secure-CNNLOC*, was found to deliver up to 10× superior localization accuracy on average, in the presence of threats from several malicious attackers, compared to the unsecured CNN and DNN-based localization framework.

Towards the goal of optimizing the cumbersome deep-learning approaches, *QuickLoc* was proposed as an integral component of *SARTHI*. The contribution first presented an analysis of the impact of CNN model depth on an indoor localization framework in terms of the achievable prediction latency and localization accuracy. From the analysis it was observed that the superior localization accuracies achieved through the use of deep-learning approaches such as CNNs came

at an ever-expanding cost of localization latency and memory requirement (model footprint). To overcome this challenge, for the first time, *QuickLoc* adapted and explored the paradigm of conditional computing (and early exit) in the context of deep learning based indoor localization. In this contribution, a novel localization framework was proposed that can dynamically adapt to the accuracy and latency needs of the target mobile platform at run-time. A comprehensive analysis of several real work factors that affect the performance and deployability of indoor localization were tested, such as, inference energy, memory footprint and device heterogeneity. Sensitivity analysis for specific to the domain of conditional computing such as, uncertainty sampling and early exit path configurations were also presented.

Lastly, we propose *STONE,* a Siamese neural encoder for long-term indoor localization with mobile devices. In *STONE*, we performed a comprehensive analysis on the impact different levels of temporal granularity (hours, days, months, year) on the achievable indoor localization accuracy using fingerprinting. *STONE* contributes to the domain by adapting a Siamese triplet-loss centric neural encoder for the purpose of indoor localization. Given that the selection of triplets is critical for the efficient convergence of the model (as discussed in chapter 7), we propose a floorplan-aware triplet selection algorithm that plays a crucial role in the training efficacy and localization performance of the overall framework. Based on our analysis of the long-term fingerprinting data, we additionally proposed temporal variation centric fingerprint augmentation methodologies for resilience against the removal of access points in the online phase over a period of a year. From the experimental evaluations, we found that *STONE* is able to deliver superior performance utilizing a small number of fingerprints per reference point (FPR). This is because *STONE* adapts the domain of few-shot learning to fingerprinting-based indoor localization. The ability to deliver competitive accuracies using a low FPR is critical, as it elevates the human effort of collecting

fingerprints in the offline phase, which is an expensive endeavor. Finally, we rigorously evaluated the *STONE* framework with state-of-art works in the domain. *STONE* establishes its superiority by delivering up to 40% improvement in localization accuracy over time using half the FPR as compared to previous works.

8.2. SUGGESTIONS FOR FUTURE WORK

With rapid growth in the computational capabilities of embedded platforms such as smartphones and improvements in the domain of pattern matching achieved through higher complexity deep-learning techniques, the performance of fingerprinting-based indoor localization frameworks would continue to improve. Through this dissertation we expect to lay the groundwork on top of which future advancements can be made. Given the nature of the core challenges in the domain of fingerprinting-based indoor localization (as discussed in chapter 1), and the current state-of-the-art deep learning technologies and its associated trends, we envision the following as the likely directions of our future work:

- *Offline phase fingerprint collection effort reduction*: Collecting high-quality fingerprints in the offline phase for the purpose of training machine learning models is an expensive manual endeavor [161] [162] [163] [164]. This burden is further intensified when fingerprints are required to be collected again while attempting to maintain the localization accuracy over time [165] [166]. The work in chapter 7 proposes a few-shot based approach that can alleviate the burden of fingerprint collection by requiring fewer sampler per reference point. However, owing to removal and replacement of APs and other environmental changes, even few-shot based approaches are bound to degrade in localization accuracy over long durations of time.

As an alternative to the collection of fingerprints across the whole floorplan, methodologies can be developed that help identify a small number of strategically selected reference points, on a given floorplan, such that they minimize the impact of temporal variation on localization accuracy.

- *Fingerprint augmentation methodologies for training deep learning models*: In several chapters across this dissertation, we have used various fingerprint augmentation strategies that help with the convergence of the deep-learning models (chapter 4) and alleviate the impact of temporal variation (chapter 7) or enable security (chapter 5). Various previous works also employ fingerprint augmentation methodologies for improved generalization and fast convergence of deep-learning models [167] [168] [169]. However, all such previous works, including ours, limit the evaluation of the augmentation method only to the specific sub-problem (such as device heterogeneity or temporal variation) only. At this time, there are no known experimental studies that evaluate the generalizability of fingerprint augmentation techniques across multiple domain challenges such as temporal variation, device heterogeneity and security in a cohesive manner. Further, even though many recent works adapt techniques from the domain of computer vision that has well established methodologies and APIs [170] [171] for image augmentation, there are no well-established standards or APIs for fingerprint augmentation available to the academics or engineers. Our future work will focus on filling this gap in this knowledge.
- *Fingerprint-centric noise reduction and AP inpainting:* The work proposed in chapter 7 enables resilience to short term noise, and AP removal through fingerprint augmentation. However, the specific deep-learning architecture employed is not designed towards this goal. It would be possible to improve upon resilience to short-term variational resilience and AP

removal by instead focusing on developing deep learning architectures and loss functions that are specifically designed towards reverting the fingerprints visible in the online phase to fingerprints as they were visible in the offline phase (back to the fingerprint). Considering the applications of vision-based systems in the domain of indoor localization, pixel inpainting could be adapted to "in-paint" the missing APs over time [172] [173] [174]. Such methodologies would enable us to segregate, the overall challenge into subcomponents that are handled by specialized models particularly designed to deal with the subcomponent.

- *Attention-based approaches for device heterogeneity resilience:* Invariability to device heterogeneity is crucial to the realization of fingerprinting-based indoor. The work in chapter 2 proposes and evaluates methodologies for improving device heterogeneity resilience through intelligent metrics. Recent works in the domain of natural language translation and computer vision have proposed attention and memory-based mechanisms that augment deep-learning models. Attention-based approaches [175] [176] [177] [178] have demonstrated improvements in these areas through improved pattern matching and also lower the computational requirements of the model deployed on the platform. Our future works will focus on adapting attention-based mechanisms to further improve resilience to device heterogeneity that is evaluated over long periods of time.
- *Lifelong learning and controlled forgetting for reduced maintenance fingerprinting-based indoor localization:* The work in chapter 7 proposes a methodology for indoor localization that is resilient to real-world temporal variation scenarios. While the Stone framework delivers sustained indoor localization accuracy over temporal variations including removal of WiFi APs, there is a general overall trend of degrading accuracy. Our evaluations of Stone and other state-of-the-art works in the domain suggest the need of retraining the deep-learning model

associated with the indoor localization frameworks. Unfortunately, collecting new fingerprints across the floorplan come at considerably high costs. Alternatives, such as crowdsourcing of fingerprints generally yields poor quality of samples limiting the achievable localization accuracy through re-training [179]. It is important to note, that acquiring unlabeled fingerprints or those fingerprints whole associated reference points or location is unknown are considerably easy to acquire. In such a scenario, fingerprints as observed on the user's device could be anonymously shared with the cloud. Our future works will attempt to monopolize on the trove of unlabeled fingerprints captured in the online phase (semi-supervised lifelong learning [180] [181]), to further improve the localization accuracy of the indoor localization frameworks in a semi-supervised continuous learning manner.

- *Advanced adversarial approaches for enhanced security of fingerprinting-based indoor localization frameworks:* Securing the indoor localization framework from third party attacks is of exceptional importance [117] [118] [119]. This is especially important under situations where indoor localization platforms are being used under critical conditions of life and safety such as fire evacuation, or by human-based mining operations. In chapter 5, we briefly evaluated an attack-methodology for convolutional model based indoor localization frameworks and present a training methodology that can overcome spoofing and Jamming attacks. Such attacks in the domain of deep learning are known as data poisoning and evasion attacks. There are however attack methodologies such as adversarial attacks have not been modeled and evaluated in the domain of fingerprinting-based indoor localization. Some such popular attack methodologies include limited-memory Broyden-Fletcher-Goldfarb-Shanno (BFGS), Jacobian-based Saliency Map Attack (JSMA), Deepfool, Carlini & Wagner Attack (C&W), Generative Adversarial Networks (GAN), and so on [182]. For the safety-critical

realization of fingerprinting-based indoor localization our future works will focus on the evaluation, resilience, and detection of such attack methodologies.

- *Temporal variation aware anomaly detection:* Several recent works in the domain of fingerprinting-based indoor localization focus on detecting spoofing or jamming attacks over long-periods of time [117] [118] [119]. These spoofing and jamming attack are detected as anomalies in the expected distribution of fingerprints visible on the floorplan. Most previous works are incognizant of temporal variations in the is signal characteristics that may be observed in the online phase. This may lead the deployed anomaly detection mechanism to falsely trigger when sufficient temporal variations have occurred. On the other hand, anomaly detection mechanisms [183] could also be used as a trigger to notify the indoor localization maintenance team of a possible requirement of fingerprint re-collection. Again, is such a scenario, an attacker might forcefully trigger such a system, such that fingerprints are re-collected on the floorplan. As covered across this dissertation, the collection of fingerprints in the offline phase can come at a significant financial cost, and so this mechanism can be used to attack an indoor localization company's financial stability. To safeguard against such scenarios, our future works will focus on the challenge of security and temporal variations jointly. The goal of such work would be to create spoofing and jamming detection mechanisms that are differentiable from temporal variations.
- *Handling unpredictability from embedded OS:* Device heterogeneity can manifest itself in various ways. For example, in chapters 2 and 3, we focus on aspects of device heterogeneity such as antenna gain that may impact the perceived RSSI signal characteristics. Later, in chapter 6, we focus on aspects of device heterogeneity that impact the latency of indoor localization deep-learning model deployed due to the variations in the memory and

computational capabilities across various devices. Another source of device heterogeneity is expressed through the use of traditional operating systems (non-real time operating systems) [184]. When a WiFi fingerprint scan is initiated on the user's smartphone in the online phase, a new background process or thread may be created that is supposed to return with the scan results. However, there are no guarantees associated with when the process is actually executed, the time it takes to execute the process and the time it takes to return the results to main foreground indoor localization application. All of these timings can vary considerably across devices, as hinted in chapter 6 by the unique WiFi scan retrieval times across various smartphones. Such unpredictability could severely affect the real-timeliness of the indoor localization framework, especially when fusing the fingerprinting-based localization results with other methodologies such as dead reckoning. Our future works will focus on the design and development of fusion techniques that take into consideration timing-based unpredictability associated with the lack real-time task scheduling and execution of the various components of an indoor localization framework.

- *Deep learning-based movement vector prediction:* In the domain of fingerprinting-based indoor localization, machine learning classification and regression models are used to predict the user's location. Most regression-based models have the ability to predict the user's location as an x-y coordinate on the floorplan. Where the floorplan is considered to be a continuous surface. One the other hand, classification-based models are deployed such that the user's location is predicted to be a tile on the floorplan grid. The classification-based gained more popularity in the domain of WiFi RSSI fingerprinting, given the upper limits of achievable localization accuracy. However, both of these approaches overlook the fact that in the online phase, the user may be walking while capturing an RSSI fingerprint. This implies that the RSSI

values captured are associated with the movement vector of the user and not a particular location on the map. At this time, there are no known works that utilize only one fingerprint to produce the user's movement vector on a floorplan. Future works will need to address this challenge.

- *Optimizing high-complexity deep-learning models towards improved energy and latency for embedded platforms:* As we improve the localization accuracy of deep-learning based indoor localization frameworks, we continue to train and deploy deep-learning models that take up more memory and require higher computational capabilities on the device they are deployed on. Given the secure nature of deploying the deep-learning model on smartphones (information is not shared across a network), indoor localization frameworks specifically targeted to be deployed on embedded platforms remain popular. However, this trend hints on the need for memory and latency optimizations that would be needed to realize high-complexity deep-learning based indoor localization models on smartphones. While compression and quantization-based model optimizations have been evaluated in the past, it is impertinent that we focus on deep-learning model architecture and domain specific optimizations [185]. One such example is that attention layers designed to handle RSSI information may not require 64-bit data types. This is because normalized WiFi RSSI values only range between 0 to 1, with a requirement of only two decimal points (7 bits). In this case, the output of an attention layer (unweighted Luong style [175]) only needs to hold up to 4 decimal points (14 bits).
- *Combining Channel State Information for superior resilience to device heterogeneity and temporal variation:* The Channel State Information (CSI), at the physical layer of the Open Systems Interconnect (OSI) model, contains the amplitude and phase information of each sub-carrier that is used to represent the attenuation and frequency deviation characteristics of the

signal propagating from the transmitter to the receiver. The amplitude attenuation occurs during the propagation of a transmitted signal and its interactions with environmental artifacts such as walls, pillars etc. leading to multipath and shadowing effects. Recent works such as in [185], have demonstrated that CSI fingerprints contain more discernable information than RSSI fingerprints leading to superior localization accuracy using fingerprinting-based localization in the indoor environment. Unfortunately, ubiquitously available off-the-shelf smartphones lack the relatively expensive networking hardware required to capture real-time CSI fingerprints in the online phase. However, it may be practical to capture CSI and RSSI fingerprints for each reference point on a given floorplan in the offline phase using specialized hardware. By identifying and learning the relationship between the CSI and RSSI fingerprints for various reference points, new synthetic RSSI fingerprints could be extrapolated. Given that CSI fingerprints contain more information about the environment than conventional RSSI fingerprints [185], the synthetic RSSI fingerprints may be used to train machine learning models that are relatively resilient to minor environmental fluctuations that lead to degradation in localization quality over time and across heterogenous devices. In the online phase, conventional RSSI fingerprints would be captured by the smartphone and fed to the machine learning model trained using synthetic RSSI fingerprints to achieve superior localization quality. Future work could evaluate the feasibility of such an approach.

BIBLIOGRAPHY

[1] F. Zafari, A. Gkelias and K. K. Leung, "A survey of indoor localization systems and technologies," Communications Surveys Tutorials, vol. 21, no. 3, pp. 2568-2599, 2019.

[2] C. Langlois, S. Tiku and S. Pasricha, "Indoor Localization with Smartphones: Harnessing the Sensor Suite in Your Pocket," IEEE Consumer Electronics Magazine, vol. 6, p. 70–80, 2017.

[3] Apple, "Apple Indoor Maps," [Online]. Available: https://register.apple.com/indoor. [Accessed 5 January 2022].

[4] Android, "Wi-Fi location: ranging with RTT," [Online]. Available: https://developer.android.com/guide/topics/connectivity/wifi-rtt. [Accessed January 2022].

[5] Apple, "Ultra Wideband Availability," [Online]. Available: https://support.apple.com/en-us/HT212274. [Accessed 28 January 2022].

[6] Google, "GO INSIDE WITH INDOOR MAPS," [Online]. Available: https://www.google.com/maps/about/partners/indoormaps/. [Accessed 22 02 2022].

[7] Aruba, "Aruba Location Services Data Sheet," [Online]. Available: https://www.arubanetworks.com/resource/aruba-location-services-data-sheet/. [Accessed 28 January 2022].

[8] IndoorLBS, "Target and Retailers Using Hybrid Indoor Location Tech to Enable," IndoorLBS, [Online]. Available: http://www.indoorlbs.com/new-

http://www.indoorlbs.com/new-blog-1/2015/11/30/target-and-other-retailers-using-indoor-location-tech. [Accessed 3 January 2022].

[9] Bluepath, "Enhance the Educational Experience with Wayfinding and Other Location-Based Services," [Online]. Available: http://www.bluepath.me/use-cases-indoor-navigation/educational.php. [Accessed 21 January 2022].

[10] Indoor Atlas, "Indoor Atlas," [Online]. Available: http://www.indooratlas.com/. [Accessed 21 January 2022].

[11] Technavio, "Top 33 Indoor Location-Based Services (LBS) Companies in the US," [Online]. Available: https://blog.technavio.org/blog/top-33-indoor-location-based-services-lbs-companies-in-the-us. [Accessed 6 January 2022].

[12] Situm, "You Safe: The App for Emergency Evacuations developed by DB System with Situm's indoor location technology," [Online]. Available: https://situm.com/en/blog-eng/geosurveillance-the-disruption-in-security/you-safe-the-app-for-emergency-evacuations-developed-by-db-system-with-situms-indoor-location-technology/. [Accessed 28 January 2022].

[13] K. Chintalapudi, A. P. Iyer and V. N. Padmanabhan, "Indoor localization without the pain," in International conference on Mobile computing and networking (MobiCom), 2010.

[14] Z. Li, T. Braun and D. Dimitrova, "A passive WiFi source localization system based on fine-grained power-based trilateration," in International Symposium on A World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2015.

[15] T. Karalar and J. Rabaey, "An RF ToF Based Ranging Implementation for Sensor Networks," in International Conference on Communications, 2006.

[16] F. Gustafsson and F. Gunnarsson, "Positioning using time-difference of arrival measurements," in International Conference on Acoustics, Speech, and Signal Processing (ICASSP), 2003.

[17] F. Hoflinger, J. Hoppe, R. Zhang, A. Ens, L. Reindl, J. Wendeberg and C. Schindelhauer, "Acoustic indoor-localization system for smart phones," in International Multi-Conference on Systems, Signals & Devices (SSD14), 2014.

[18] J. Xiong and K. Jamieson, "Arraytrack: A fine-grained indoor location system," in Symposium on Networked Systems Design and Implementation (NSDI), 2013.

[19] J. A. B. Link, P. Smith, N. Viol and K. Wehrle, "FootPath: Accurate map-based indoor navigation using smartphones," in International Conference on Indoor Positioning and Indoor Navigation, 2011.

[20] R. W. Levi and T. Judd, Dead reckoning navigational system using accelerometer to measure foot impacts, Google Patents, 1996.

[21] J. W. Kim, H. J. Jang, D.-H. Hwang and C. Park, "A Step, Stride and Heading Determination for the Pedestrian Navigation System," Journal of Global Positioning Systems, vol. 3, p. 273–279, 2004.

[22] R. Harle, "A Survey of Indoor Inertial Positioning Systems for Pedestrians," Communications Surveys & Tutorials, vol. 15, p. 1281–1293, 2013.

[23] U. Steinhoff and B. Schiele, "Dead reckoning from the pocket - An experimental study," in International Conference on Pervasive Computing and Communications (PerCom), 2010.

[24] H. Hellmers, A. Norrdine, J. Blankenbach and A. Eichhorn, "An IMU/magnetometer-based Indoor positioning system using Kalman filtering," in International Conference on Indoor Positioning and Indoor Navigation, 2013.

[25] S. Lamy-Perbal, N. Guenard, M. Boukallel and A. Landragin-Frassati, "A HMM map-matching approach enhancing indoor positioning performances of an inertial measurement system," in International Conference on Indoor Positioning and Indoor Navigation (IPIN), 2015.

[26] F. T. Alaoui, D. Betaille and V. Renaudin, "A multi-hypothesis particle filtering approach for pedestrian dead reckoning," in International Conference on Indoor Positioning and Indoor Navigation (IPIN), 2016.

[27] S. Beauregard, Widyawan and M. Klepal, "Indoor PDR performance enhancement using minimal map information and particle filters," in Position, Location and Navigation Symposium, 2008.

[28] S. Cangeloso, "Forget WiFiSlam - ByteLight uses LEDs for indoor positioning," [Online]. Available: https://www.extremetech.com/extreme/151068-forget-wifislam-bytelight-uses-leds-for-indoor-positioning. [Accessed 25 January 2022].

[29] J. A. B. Link, F. Gerdsmeier, P. Smith and K. Wehrle, "Indoor navigation on wheels (and on foot) using smartphones," in International Conference on Indoor Positioning and Indoor Navigation (IPIN), 2012.

[30] N. Gageik, M. Strohmeier and S. Montenegro, "An Autonomous UAV with an Optical Flow Sensor for Positioning and Navigation," International Journal of Advanced Robotic Systems, vol. 10, p. 341, 2013.

[31] S. Tiku and S. Pasricha, "Overcoming Security Vulnerabilities in Deep Learning–based Indoor Localization Frameworks on Mobile Devices," ACM Transactions on Embedded Computing Systems, vol. 18, p. 1–24, 2020.

[32] S. Tiku and S. Pasricha, "PortLoc: A Portable Data-Driven Indoor Localization Framework for Smartphones," Design & Test, vol. 36, no. 5, p. 18–26, 2019.

[33] H. Jiang, C. Peng and J. Sun, "Deep Belief Network for Fingerprinting-Based RFID Indoor Localization," in International Conference on Communications (ICC), 2019.

[34] X. Wang, X. Wang and S. Mao, "CiFi: Deep convolutional neural networks for indoor localization with 5 GHz Wi-Fi," in International Conference on Communications (ICC), 2017.

[35] X. Wang, L. Gao, S. Mao and S. Pandey, "DeepFi: Deep learning for indoor fingerprinting using channel state information," in Wireless Communications and Networking Conference (WCNC), 2015.

[36] S. P. Mohanty, U. Choppali and E. Kougianos, "Everything you wanted to know about smart cities: The Internet of things is the backbone," IEEE Consumer Electronics Magazine, vol. 5, p. 60–70, 2016.

[37] S. Pasricha, V. Ugave, Q. Han and C. Anderson, "LearnLoc: A framework for smart indoor localization with embedded mobile devices," in International Conference on Hardware/Software Codesign and System Synthesis (CODES+ISSS), 2015.

[38] A. Mittal, S. Tiku and S. Pasricha, "Adapting Convolutional Neural Networks for Indoor Localization with Smart Mobile Devices," in Great Lakes Symposium on VLSI (GLSVLSI), 2018.

[39] V. Singh, G. Aggarwal and B. V. S. Ujwal, "Ensemble based real-time indoor localization using stray WiFi signal," in International Conference on Consumer Electronics (ICCE), 2018.

[40] H. Zou, B. Huang, X. Lu, H. Jiang and L. Xie, "A Robust Indoor Positioning System Based on the Procrustes Analysis and Weighted Extreme Learning Machine," Transactions on Wireless Communications (TWC), vol. 15, p. 1252–1266, 2016.

[41] B. Soro and C. Lee, "Joint Time-Frequency RSSI Features for Convolutional Neural Network-Based Indoor Fingerprinting Localization," IEEE Access, vol. 7, p. 104892–104899, 2019.

[42] M. B. Kjærgaard, "Indoor location fingerprinting with heterogeneous clients," Pervasive and Mobile Computing, vol. 7, p. 31–43, 2011.

[43] C. Figuera, J. L. Rojo-Álvarez, I. Mora-Jiménez, A. Guerrero-Curieses, M. Wilby and J. Ramos-López, "Time-Space Sampling and Mobile Device Calibration for WiFi Indoor Location Systems," Transactions on Mobile Computing, vol. 10, p. 913–926, 2011.

[44] S. Yang, P. Dessai, M. Verma and M. Gerla, "FreeLoc: Calibration-free crowdsourced indoor localization," in International Conference on Computer Communications (INFOCOM), 2013.

[45] M. B. Kjærgaard and C. V. Munk, "Hyperbolic location fingerprinting: A calibration-free solution for handling differences in signal strength (concise contribution)," in International Conference on Pervasive Computing and Communications (PerCom), 2008.

[46] A. K. M. M. Hossain, Y. Jin, W.-S. Soh and H. N. Van, "SSD: A Robust RF Location Fingerprint Addressing Mobile Device Heterogeneity," Transactions on Mobile Computing, vol. 12, no. 1, p. 65–77, 2013.

[47] S. Pasricha, B. K. Donohoo and C. Ohlsen, "A middleware framework for application-aware and user-specific energy optimization in smart mobile devices," Pervasive and Mobile Computing, vol. 20, no. C, p. 47–63, 2015.

[48] J. Machaj, P. Brida and R. Piché, "Rank based fingerprinting algorithm for indoor positioning," in International Conference on Indoor Positioning and Indoor Navigation, 2011.

[49] J.-g. Park, D. Curtis, S. Teller and J. Ledlie, "Implications of device diversity for organic localization," in International Conference on Computer Communications (INFOCOM), 2011.

[50] B. K. Donohoo, C. Ohlsen, S. Pasricha, Y. Xiang and C. Anderson, "Context-Aware Energy Enhancements for Smart Mobile Devices," Transactions on Mobile Computing, vol. 13, no. 8, p. 1720–1732, 2014.

[51] Q. Wang, H. Luo, F. Zhao and W. Shao, "An indoor self-localization algorithm using the calibration of the online magnetic fingerprints and indoor landmarks," in International Conference on Indoor Positioning and Indoor Navigation (IPIN), 2016.

[52] S. Pasricha, J. R. Doppa, K. Chakrabarty, S. Tiku, D. Dauwe, S. Jin and P. P. Pande, "Data analytics enables energy-efficiency and robustness," in International Conference on Hardware/Software Codesign and System Synthesis Companion (CODES+ISSS), 2017.

[53] Y. Li, S. Williams, B. Moran and A. Kealy, "A Probabilistic Indoor Localization System for Heterogeneous Devices," Sensors Journal, vol. 19, p. 6822–6832, 2019.

[54] J. Schmitz, M. Hernandez and R. Mathar, "Demonstration Abstract: Real-Time Indoor Localization with TDOA and Distributed Software Defined Radio," in International Conference on Information Processing in Sensor Networks (IPSN), 2016.

[55] X. J. and J. K., "Towards fine-grained radio-based indoor location," in Mobile Computing Systems and Applications (HotMobile), 2012.

[56] E. Soltanaghaei, A. Kalyanaraman and K. Whitehouse, "Multipath Triangulation: Decimeter-level wi-fi localization and orientation with a single unaided receiver," in International Conference on Mobile Systems, Applications, and Services, 2018.

[57] U. Bolat and M. Akcakoca, "A hybrid indoor positioning solution based on Wi-Fi, magnetic field, and inertial navigation," in Workshop on Positioning, Navigation and Communications (WPNC), 2017.

[58] Y. Hu, X. Liao, Q. Lu, S. Xu and W. Zhu, "A segment-based fusion algorithm of WiFi fingerprinting and pedestrian dead reckoning," in International Conference on Communications in China (ICCC), 2016.

[59] C. Ascher, C. Kessler, R. Weis and G. F. Trommer, "Multi-floor map matching in indoor environments for mobile platforms," in International Conference on Indoor Positioning and Indoor Navigation (IPIN), 2012.

[60] A. Haeberlen, E. Flannery, A. M. Ladd, A. Rudys, D. S. Wallach and L. E. Kavraki, "Practical robust localization over large-scale 802.11 wireless networks," in International Conference on Mobile Computing and Networking (MobiCom), 2004.

[61] D. Han, H. Rho and S. Lim, "HMM-Based Indoor Localization Using Smart Watche BLE Signals," in International Conference on Future Internet of Things and Cloud (FiCloud), 2018.

[62] B.-J. Yoon and P. P. Vaidyanathan, "Context-Sensitive Hidden Markov Models for Modeling Long-Range Dependencies in Symbol Sequences," Transactions on Signal Processing, vol. 54, p. 4169–4184, 2006.

[63] S.-C. Poh, Y.-F. Tan, X. Guo, S.-N. Cheong, C.-P. Ooi and W.-H. Tan, "LSTM and HMM Comparison for Home Activity Anomaly Detection," in Information Technology, Networking, Electronic and Automation Control Conference (ITNEC), 2019.

[64] K. Oura, K. Tokuda, J. Yamagishi, S. King and M. Wester, "Unsupervised cross-lingual speaker adaptation for HMM-based speech synthesis," in International Conference on Acoustics, Speech and Signal Processing, 2010.

[65] F. Li, C. Zhao, G. Ding, J. Gong, C. Liu and F. Zhao, "A reliable and accurate indoor localization method using phone inertial sensors," in Conference on Ubiquitous Computing (UbiComp), 2012.

[66] S. Tiku, P. Sudeep, B. Notaros and Q. Han, "SHERPA: A Lightweight Smartphone Heterogeneity Resilient Portable Indoor Localization Framework," in International Conference on Embedded Software and Systems (ICESS), 2019.

[67] F. Coolidge, "An introduction to correlation and regression," in Statistics-A Gentle Introduction, SAGE Publications, Inc, 2006, p. 153–196.

[68] Investopedia, "How Google Maps Makes Money," [Online]. Available: https://www.investopedia.com/articles/investing/061115/how-does-google-maps-makes-money.asp. [Accessed 3 January 2022].

[69] Radial Analytics, "Case Study: Accuracy & Precision of Google Analytics Geolocation," Radial Analytics, [Online]. Available: https://radical-analytics.com/case-study-accuracy-precision-of-google-analytics-geolocation-4264510612c0. [Accessed 28 January 2022].

[70] Ubisense, "Ubisense Research Network," [Online]. Available: http://www.ubisense.net. [Accessed 28 January 2022].

[71] S. Tiku and S. Pasricha, "Energy-efficient and robust middleware prototyping for smart mobile computing," in International Symposium on Rapid System Prototyping Shortening the Path from Specification to Prototype (RSP), 2017.

[72] A. Khune and S. Pasricha, "Mobile Network-Aware Middleware Framework for Energy-Efficient Cloud Offloading of Smartphone," IEE Consumer Electronics Magazine, vol. 8, no. 1, pp. 42-48, 2019.

[73] B. K. Donohoo, C. Ohlsen and S. Pasricha, "AURA: An application and user interaction aware middleware framework for energy optimization in mobile devices," in International Conference on Computer Design (ICCD), 2011.

[74] B. Donohoo, C. Ohlsen, S. Pasricha and C. Anderson, "Exploiting spatiotemporal and device contexts for energy-efficient mobile embedded systems," in Design Automation Conference (DAC), 2012.

[75] V. Rausch, A. Hansen, E. Solowjow, C. Liu, E. Kreuzer and J. K. Hedrick, "Learning a deep neural net policy for end-to-end control of autonomous vehicles," in American Control Conference (ACC), 2017.

[76] Z. Chen and C. Wang, "Modeling RFID signal distribution based on neural network combined with continuous ant colony optimization," Neurocomputing, vol. 123, pp. 354-361, 2014.

[77] G. Borriello, A. Liu, T. Offer, C. Palistrant and R. Sharp, "WALRUS: Wireless Acoustic Location with Room-level Resolution using Ultrasound," in International Conference on Mobile Systems, Applications, and Services (MobiSys), 2005.

[78] M. Azizyan, I. Constandache and R. R. Choudhury, "SurroundSense: Mobile Phone Localization via Ambience Fingerprinting," in Mobile computing and networking (MobiCom), 2009.

[79] C. Yang and H.-r. Shao, "WiFi-based indoor positioning," Communications Magazine, vol. 53, p. 150–157, 2015.

[80] J. Torres-Sospedra, R. Montoliu, A. Martinez-Uso, J. P. Avariento, T. J. Arnau, M. Benedito-Bordonau and J. Huerta, "UJIIndoorLoc: A new multi-building and multi-

floor database for WLAN fingerprint-based indoor localization problems," in International Conference on Indoor Positioning and Indoor Navigation (IPIN), 2014.

[81] P. Bahl and V. N. Padmanabhan, "RADAR: an in-building RF-based user location and tracking system," in International Conference on Computer Communications (INFOCOM), 2000.

[82] W. Zhang, K. Liu, W. Zhang, Y. Zhang and J. Gu, "Deep Neural Networks for wireless localization in indoor and outdoor environments," Elsevier Neurocomputing, vol. 194, pp. 279-287, 2016.

[83] H. Chen, Y. Zhang, W. Li and X. Tao, "ConFi: Convolutional Neural Networks Based Indoor Wi-Fi Localization Using Channel State Information," IEEE Access, vol. 5, p. 18066–18074, 2017.

[84] Y. Hua, J. Guo and H. Zhao, "Deep Belief Networks and deep learning," in Intelligent Computing and Internet of Things, 2015.

[85] A. Krizhevsky, I. Sutskever and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in Advances in neural information processing systems, 2017.

[86] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," arXiv preprint arXiv:1412.6980, 2014.

[87] "RMSProp," [Online]. Available: http://www.cs.toronto.edu/~tijmen/csc321/slides/lecture_slides_lec6.pdf. [Accessed 21 January 2022].

[88] J. Duchi, E. Hazan and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization," Journal of Machine Learning Research, vol. 12, p. 2121–2159, 2011.

[89] Y. Lecun, L. Bottou, Y. Bengio and P. Haffner, "Gradient-based learning applied to document recognition," Proceedings of the IEEE, vol. 86, pp. 2278-2324, 1998.

[90] J. Benesty, J. Chen, Y. Huang and I. Cohen, "Pearson Correlation Coefficient," in Noise Reduction in Speech Processing, Springer Berlin Heidelberg, 2009, p. 1–4.

[91] G. P. H. Styan, "Hadamard products and multivariate statistical analysis," Linear Algebra and its Applications, vol. 6, p. 217–240, 1973.

[92] Y.-K. Cheng, H.-J. Chou and R. Y. Chang, "Machine-Learning Indoor Localization with Access Point Selection and Signal Strength Reconstruction," in Vehicular Technology Conference (VTC Spring), 2016.

[93] The Atlantic, "The Plane Crash That Gave Americans GPS," [Online]. Available: https://www.theatlantic.com/technology/archive/2014/11/the-plane-crash-that-gave-americans-gps/382204/. [Accessed 6 January 2022].

[94] pc world, " A brief history of GPS," [Online]. Available: https://www.pcworld.com/article/461346/a-brief-history-of-gps.html. [Accessed 6 January 2022].

[95] J. A. Larcom and H. Liu, "Modeling and characterization of GPS spoofing," in International Conference on Technologies for Homeland Security (HST), 2013.

[96] C. Bonebrake and L. R. O'Neil, "Attacks on GPS Time Reliability," Security & Privacy, vol. 12, p. 82–84, 2014.

[97] K. Jansen, M. Schäfer, D. Moser, V. Lenders, C. Pöpper and J. Schmitt, "Crowd-GPS-Sec: Leveraging Crowdsourcing to Detect and Localize GPS Spoofing Attacks," in Symposium on Security and Privacy (SP), 2018.

[98] Forbes, "This GPS Spoofing Hack Can Really Mess with Your Google Maps Trips," [Online]. Available: https://www.forbes.com/sites/thomasbrewster/2018/07/12/google-maps-gps-hack-takes-victims-to-ghost-locations/?sh=12ec02d06335. [Accessed 6 January 2022].

[99] GPS world, "Spoofing in the Black Sea: What really happened?," [Online]. Available: https://www.gpsworld.com/spoofing-in-the-black-sea-what-really-happened/. [Accessed 6 January 2022].

[100] C. K. Schindhelm and A. MacWilliams, "Overview of Indoor Positioning Technologies for Context Aware AAL Applications," in Ambient Assisted Living, Springer Berlin Heidelberg, 2011, p. 273–291.

[101] Y. C. Chen, W. C. Sun and J. C. Juang, "Outlier detection technique for RSS-based localization problems in wireless sensor networks," in SICE Annual Conference, 2010.

[102] A. Khalajmehrabadi, N. Gatsis, D. J. Pack and D. Akopian, "A Joint Indoor WLAN Localization and Outlier Detection Scheme Using LASSO and Elastic-Net Optimization Techniques," Transactions on Mobile Computing, vol. 16, p. 2079–2092, 2017.

[103] D. Vasisht, S. Kumar and D. Katabi, "Sub-Nanosecond Time of Flight on Commercial Wi-Fi Cards," SIGCOMM Computer Communication Review, vol. 45, p. 121–122, 2015.

[104] Z. Chen, Z. Li, X. Zhang, G. Zhu, Y. Xu, J. Xiong and X. Wang, "AWL: Turning spatial aliasing from foe to friend for accurate WiFi localization," in International Conference on Emerging Networking Experiments and Technologies, 2017.

[105] Z. Lu, W. Wang and C. Wang, "Modeling, Evaluation and Detection of Jamming Attacks in Time-Critical Wireless Applications," Transactions on Mobile Computing, vol. 13, p. 1746–1759, 2014.

[106] W. Meng, W. Xiao, W. Ni and L. Xie, "Secure and robust Wi-Fi fingerprinting indoor localization," in International Conference on Indoor Positioning and Indoor Navigation, 2011.

[107] L. Deng, "A tutorial survey of architectures, algorithms, and applications for deep learning," APSIPA Transactions on Signal and Information Processing, vol. 3, 2014.

[108] J. Jang and S. Hong, "Indoor Localization with WiFi Fingerprinting Using Convolutional Neural Network," in International Conference on Ubiquitous and Future Networks (ICUFN), 2018.

[109] M. Mohammadi, A. Al-Fuqaha, M. Guizani and J.-S. Oh, "Semisupervised Deep Reinforcement Learning in Support of IoT and Smart City Services," Internet of Things Journal, vol. 5, no. 2, pp. 624-635, 2018.

[110] P. Dickinson, G. Cielniak, O. Szymanezyk and M. Mannion, "Indoor positioning of shoppers using a network of Bluetooth Low Energy beacons," in International Conference on Indoor Positioning and Indoor Navigation (IPIN), 2016.

[111] S.-Y. Lau, T.-H. Lin, T.-Y. Huang, I.-H. Ng and P. Huang, "A measurement study of zigbee-based indoor localization systems under rf interference," in International workshop on Experimental evaluation and characterization, 2009.

[112] L. Chang, X. Chen, J. Wang, D. Fang, C. Liu, Z. Tang and W. Nie, "TaLc," in MobiCom Workshop on Challenged Networks (CHANTS), 2015.

[113] D. Barbará, R. Goel and S. Jajodia, "Using Checksums to Detect Data Corruption," in Advances in Database Technology (EDBT), 2000.

[114] L. Ou, Z. Qin, Y. Liu, H. Yin, Y. Hu and H. Chen, "Multi-User Location Correlation Protection with Differential Privacy," in International Conference on Parallel and Distributed Systems (ICPADS), 2016.

[115] L. Lazos and M. Krunz, "Selective jamming/dropping insider attacks in wireless mesh networks," IEEE Network, vol. 25, p. 30–34, 2011.

[116] W. Xu, W. Trappe, Y. Zhang and T. Wood, "The feasibility of launching and detecting jamming attacks in wireless networks," in International Symposium on Mobile Ad-Hoc Networking and Computing, 2005.

[117] Á. M. Guerrero-Higueras, N. DeCastro-García, F. J. Rodríguez-Lera and V. Matellán, "Empirical analysis of cyber-attacks to an indoor real time localization system for autonomous robots," Computers & Security, vol. 70, p. 422–435, 2017.

[118] Á. M. Guerrero-Higueras, N. DeCastro-García and V. Matellán, "Detection of Cyber-attacks to indoor real time localization systems for autonomous robots," Robotics and Autonomous Systems, vol. 99, p. 75–83, 2018.

[119] A. A. A. Silva, E. Pontes, A. E. Guelfi, I. Caproni, R. Aguiar, F. Zhou and S. T. Kofuji, "Predicting model for identifying the malicious activity of nodes in MANETs," in Symposium on Computers and Communication (ISCC), 2015.

[120] C. Wu, Z. Yang and Y. Liu, "Smartphones Based Crowdsourcing for Indoor Localization," Transactions on Mobile Computing, vol. 14, p. 444–457, 2015.

[121] HTC, "HTC U11," [Online]. Available: https://www.htc.com/us/smartphones/htc-u11. [Accessed 5 January 2022].

[122] J.-S. Lee, "Digital image smoothing and the sigma filter," Computer Vision, Graphics, and Image Processing, vol. 24, p. 255–269, 1983.

[123] Y. Shu, Y. Huang, J. Zhang, P. Coue, P. Cheng, J. Chen and K. G. Shin, "Gradient-Based Fingerprinting for Indoor Localization and Tracking," Transactions on Industrial Electronics, vol. 63, p. 2424–2433, 2016.

[124] F. Zhang, N. Cai, J. Wu, G. Cen, H. Wang and X. Chen, "Image denoising method based on a deep convolution neural network," IET Image Processing, vol. 12, p. 485–493, 2018.

[125] tp-link, "MAC Address Clone on my TP-Link," [Online]. Available: https://www.tp-link.com/us/support/faq/68/. [Accessed 5 March 2020].

[126] R. B. Langley, "The evolution of the GPS receiver," GPS World, vol. 11, no. 4, pp. 54-58, 2000.

[127] D. Murph, "Fraunhofer IIS uses Awiloc indoor positioning magic to guide museum patrons," [Online]. Available: https://www.engadget.com/2010-12-13-fraunhofer-iis-

uses-awiloc-indoor-positioning-magic-to-guide-mus.html. [Accessed 21 January 2022].

[128] B. Raffaele and F. Delmastro, "Design and analysis of a bluetooth-based indoor localization," in International Conference on Personal Wireless Communications, 2003.

[129] P. Krishnan, A. Krishnakumar, W.-H. Ju, C. Mallows and S. Gamt, "A system for LEASE: Location estimation assisted by stationary emitters for indoor RF wireless networks," in International Conference on Computer Communications (INFOCOM), 2004.

[130] A. Mackey, P. Spachos, L. Song and K. N. Plataniotis, "Improving BLE Beacon Proximity Estimation Accuracy Through Bayesian Filtering," Internet of Things Journal, vol. 7, no. 4, pp. 3160-3169, 2020.

[131] A. Alarifi, A. Al-Salman, M. Alsaleh, A. Alnafessah, S. Al-Hadhrami, M. A. Al-Ammar and H. S. Al-Khalifa, "Ultra-wideband indoor positioning," Sensors, vol. 16, no. 5, p. 707, 2016.

[132] K. P. Subbu, B. Gozick and R. Dantu, "LocateMe: Magnetic-fields-based indoor localization using smartphones," Transactions on Intelligent Systems and Technology, vol. 4, p. 1–27, 2013.

[133] H. Xie, T. Gu, X. Tao, H. Ye and J. Lv, "MaLoc: A practical magnetic fingerprinting approach to indoor localization using smartphones," in International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp), 2014.

[134] W. Shao, H. Luo, F. Zhao, Y. Ma, Z. Zhao and A. Crivello, "Indoor Positioning Based on Fingerprint-Image and Deep Learning," IEEE Access, vol. 6, pp. 74699-74712, 2018.

[135] P. Panda, A. Sengupta and K. Roy, "Conditional Deep Learning for Energy-Efficient and Enhanced Pattern Recognition," in Design, Automation and Test in Europe Conference and Exhibition (DATE), 2016.

[136] S. Teerapittayanon, B. McDanel and H.-T. Kung, "Branchynet: Fast inference via early exiting from deep neural networks," in International Conference on Pattern Recognition (ICPR), 2016.

[137] R. Paróczai and L. Kocsis, "Analysis of human walking and running parameters as a function of speed," Technology and Health Care, vol. 14, no. 4, pp. 251-260, 2006.

[138] R. E. Kalman, "A New Approach to Linear Filtering and Prediction Problems," Journal of Basic Engineering, vol. 82, p. 35–45, 1960.

[139] Q. Lu, X. Liao, S. Xu and W. Zhu, "A hybrid indoor positioning algorithm based on WiFi fingerprinting and pedestrian dead reckoning," in International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC), 2016.

[140] S. Vajda, C. E. Shannon and W. Weaver, "The Mathematical Theory of Communication," The Mathematical Gazette, vol. 34, p. 312, 1950.

[141] T.-J. Yang, Y.-H. Chen and V. Sze, "Designing Energy-Efficient Convolutional Neural Networks Using Energy-Aware Pruning," in Conference on Computer Vision and Pattern Recognition (CVPR), 2017.

[142] Y. Wang, Y. Bu, Q. Jin and V. A. Vasilakos, "Energy-Efficient Localization and Tracking on Smartphones," in International Conference on Future Internet Technologies, 2016.

[143] I. Hubara, M. Courbariaux, D. Soudry, R. El-Yaniv and Y. Bengio, "Quantized neural networks: training neural networks with low precision weights and activations," The Journal of Machine Learning Research, vol. 18, no. 1, p. 6869–6898, 2017.

[144] G. Huang, Z. Liu, L. V. D. Maaten and K. Q. Weinberger, "Densely Connected Convolutional Networks," in Conference on Computer Vision and Pattern Recognition (CVPR), 2017.

[145] Tensorflow, "Performance measurement," [Online]. Available: https://www.tensorflow.org/lite/performance/measurement. [Accessed 1 August 2021].

[146] Android Developers, "Battery Manager API," [Online]. Available: https://developer.android.com/reference/android/os/BatteryManager. [Accessed 26 January 2022].

[147] Android Play Store, "tPacketCapture," [Online]. Available: https://play.google.com/store/apps/details?id=jp.co.taosoftware.android.packetcapture. [Accessed 21 January 2022].

[148] Seemoo-Lab, "Nexmon," [Online]. Available: https://github.com/seemoo-lab/nexmon. [Accessed 28 January 2022].

[149] Nobel Systems, "20 Ways GIS Data is Used in Business and Everyday Life," [Online]. Available: https://nobelsystemsblog.com/gis-data-business/. [Accessed 5 January 2022].

[150] L. Wang, S. Tiku and S. Pasricha, "CHISEL: Compression-Aware High-Accuracy Embedded Indoor Localization with Deep Learning," in Embedded Systems Letters, 2021.

[151] D. Li, J. Xu, Z. Yang, Y. Lu, Q. Zhang and X. Zhang, "Train Once, Locate Anytime for Anyone: Adversarial Learning based Wireless Localization," in International Conference on Computer Communications (INFOCOM), 2021.

[152] G. Mendoza-Silva, P. Richter, J. Torres-Sospedra, E. Lohan and J. Huerta, "Long-Term WiFi Fingerprinting Dataset for Research on Robust Indoor Positioning," MDPI Data, vol. 3, p. 3, 2018.

[153] Y. a. B. C. Shu, G. Shen, C. Zhao, L. Li and F. Zhao, "Magicol: Indoor Localization Using Pervasive Magnetic Field and Opportunistic WiFi Sensing," Journal on Selected Areas in Communications, vol. 33, no. 7, pp. 1443-1457, 2015.

[154] M. Abbas, M. Elhamshary, H. Rizk, M. Torki and M. Youssef, "WiDeep: WiFi-based Accurate and Robust Indoor Localization System using Deep Learning," in International Conference on Pervasive Computing and Communications (PerCom), 2019.

[155] A. Pandey, R. Sequeira and S. Kumar, "SELE: RSS Based Siamese Embedding Location Estimator for a Dynamic IoT Environment," in Internet of Things Journal, 2021.

[156] S. Tiku, S. Pasricha, B. Notaros and Q. Han, "A Hidden Markov Model based Smartphone Heterogeneity Resilient Portable Indoor Localization Framework," Journal of Systems Architecture, vol. 108, p. 101806, 2020.

[157] C. Wu, J. Xu, Z. Yang, N. D. Lane and Z. Yin, "Gain Without Pain: Accurate WiFi-based Localization using Fingerprint Spatial Gradient," Interactive, Mobile, Wearable and Ubiquitous Technologies, vol. 1, p. 1–19, 2017.

[158] R. Montoliu, E. Sansano, O. Belmonte and J. Torres-Sospedra, "A New Methodology for Long-Term Maintenance of WiFi Fingerprinting Radio Maps," in International Conference on Indoor Positioning and Indoor Navigation (IPIN), 2018.

[159] Y. Taigman, M. Yang, M. Ranzato and L. Wolf, "DeepFace: Closing the Gap to Human-Level Performance in Face Verification," in Conference on Computer Vision and Pattern Recognition, 2014.

[160] F. Schroff, D. Kalenichenko and J. Philbin, "FaceNet: A unified embedding for face recognition and clustering," in Conference on Computer Vision and Pattern Recognition (CVPR), 2015.

[161] D. V. Le, N. Meratnia and P. J. Havinga, "Unsupervised Deep Feature Learning to Reduce the Collection of Fingerprints for Indoor Localization Using Deep Belief Networks," in International Conference on Indoor Positioning and Indoor Navigation (IPIN), 2018.

[162] Q. a. Q. H. Li, Z. Liu, N. Zhou, W. Sun, S. Sigg and J. Li, "AF-DCGAN: Amplitude Feature Deep Convolutional GAN for Fingerprint Construction in Indoor Localization Systems," Transactions on Emerging Topics in Computational Intelligence, vol. 5, no. 3, pp. 468-480, 2021.

[163] Z. Gu, Z. Chen, Y. Zhang, Y. Zhu, M. Lu and A. Chen, "Reducing fingerprint collection for indoor localization," Computer Communications, vol. 83, pp. 56-63, 2016.

[164] L. Li, X. Guo, N. Ansari and H. Li, "A Hybrid Fingerprint Quality Evaluation Model for WiFi Localization," Internet of Things Journal, vol. 6, p. 9829–9840, 2019.

[165] X. Zhang, Y. Jin, H.-X. Tan and W.-S. Soh, "CIMLoc: A crowdsourcing indoor digital map construction system for localization," in Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP), 2014.

[166] V. Moghtadaiee, S. A. Ghorashi and M. Ghavami, "New Reconstructed Database for Cost Reduction in Indoor Fingerprinting Localization," IEEE Access, vol. 7, pp. 104462-104477, 2019.

[167] W. Sun, M. Xue, H. Yu, H. Tang and A. Lin, "Augmentation of Fingerprints for Indoor WiFi Localization Based on Gaussian Process Regression," Transactions on Vehicular Technology, vol. 67, no. 11, pp. 10896-10905, 2018.

[168] W. Njima, M. Chafii, A. Chorti, R. M. Shubair and H. V. Poor, "Indoor Localization Using Data Augmentation via Selective Generative Adversarial Networks," IEEE Access, vol. 9, pp. 98337-98347, 2021.

[169] A. Mathur, T. Zhang, S. Bhattacharya, P. Velickovic, L. Joffe, N. D. Lane, F. Kawsar and P. Lio, "Using Deep Data Augmentation Training to Address Software and Hardware Heterogeneities in Wearable and Smartphone Sensing Devices," 2018.

[170] "Tensorflow Core: Data Augmentation," [Online]. Available: https://www.tensorflow.org/tutorials/images/data_augmentation. [Accessed 15 January 2022].

[171] "Pytorch: Transforming and Augmenting Images," [Online]. Available: https://pytorch.org/vision/master/transforms.html. [Accessed 15 January 2022].

[172] Z. Yan, X. Li, M. Li, W. Zuo and S. Shan, "Shift-Net: Image Inpainting via Deep Feature Rearrangement," in The European Conference on Computer Vision (ECCV), 2018.

[173] J. Yu, Z. Lin, J. Yang, X. Shen, X. Lu and T. S. Huang, "Generative Image Inpainting with Contextual Attention," in Computer Vision and Pattern Recognition (CVPR), 2018.

[174] K. Nazeri, E. Ng, T. Joseph, F. Z. Qureshi and M. Ebrahimi, "EdgeConnect: Generative Image Inpainting with Adversarial Edge Learning," in International Conference on Computer Vision (ICCV), 2019.

[175] M.-T. Luong, H. Pham and C. D. Manning, Effective Approaches to Attention-based Neural Machine Translation, arXiv:1508.04025, 2015.

[176] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser and I. Polosukhin, "Attention is All you Need," in Advances in Neural Information Processing Systems (NIPS), 2017.

[177] K. P. Nkabiti and Y. Chen, "Application of solely self-attention mechanism in CSI-fingerprinting-based indoor localization," Springer Neural Computing and Applications, pp. 1-14, 2021.

[178] A. M. Hafiz, S. A. Parah and R. U. A. Bhat, Attention mechanisms and deep learning for machine vision: A survey of the state of the art, arXiv:2106.07550, 2021.

[179] C. Wu, Z. Yang and C. Xiao, "Automatic Radio Map Adaptation for Indoor Localization Using Smartphones," Transactions on Mobile Computing, vol. 17, p. 517–528, 2018.

[180] J. Chen, M. Yang and J. Ling, "Attention-based label consistency for semi-supervised deep learning based image classification," Neurocomputing, vol. 453, pp. 731-741, 2021.

[181] Y. Kamiya, T. Ishii and O. Hasegawa, "Life-long Semi-supervised Learning: Continuation of Both Learning and Recognition," in Symposium on Computational Intelligence in Image and Signal Processing, 2007.

[182] A. Chakraborty, M. Alam, V. Dey, A. Chattopadhyay and D. Mukhopadhyay, Adversarial Attacks and Defences: A Survey, arXiv: 1810.00069, 2018.

[183] P. Chandana, C. Aishwarya and S. S. Muskan, "Anomaly detection in indoor localization using Machine Learning," in International Conference on Inventive Research in Computing Applications (ICIRCA), 2021.

[184] Y. J. Woo, J. Cho, D. Lim and E. Seo, "Cross-layer real-time support for JVM-based smartphone systems," in International Conference on Consumer Electronics (ICCE), 2012.

[185] W. Xun, L. Sun, C. Han, Z. Lin and J. Guo, "Depthwise Separable Convolution based Passive Indoor Localization using CSI Fingerprint," in Wireless Communications and Networking Conference (WCNC), 2020.