Overcoming Security Vulnerabilities in Deep Learning–based Indoor Localization Frameworks on Mobile Devices

SAIDEEP TIKU and SUDEEP PASRICHA, Department of Electrical and Computer Engineering, Colorado State University, Fort Collins, Colorado

Indoor localization is an emerging application domain for the navigation and tracking of people and assets. Ubiquitously available Wi-Fi signals have enabled low-cost fingerprinting-based localization solutions. Further, the rapid growth in mobile hardware capability now allows high-accuracy deep learning–based frameworks to be executed locally on mobile devices in an energy-efficient manner. However, existing deep learning–based indoor localization solutions are vulnerable to access point (AP) attacks. This article presents an analysis into the vulnerability of a convolutional neural network–based indoor localization solution to AP security compromises. Based on this analysis, we propose a novel methodology to maintain indoor localization accuracy, even in the presence of AP attacks. The proposed secured neural network framework (S-CNNLOC) is validated across a benchmark suite of paths and is found to deliver up to 10× more resiliency to malicious AP attacks compared to its unsecured counterpart.

CCS Concepts: • **Security and privacy** → **Domain-specific security and privacy architectures**; • **Computing methodologies** → *Neural networks*;

Additional Key Words and Phrases: Indoor localization, security, reliability, indoor navigation, spoofing, AP attacks, deep learning

ACM Reference format:
Saideep Tiku and Sudeep Pasricha. 2019. Overcoming Security Vulnerabilities in Deep Learning–based Indoor Localization Frameworks on Mobile Devices. *ACM Trans. Embed. Comput. Syst.* 18, 6, Article 114 (November 2019), 24 pages.
http://dx.doi.org/10.1145/3362036

1 INTRODUCTION

In the early 1980s, the unintended deviation of a commercial airliner from its designated path due to unreliable navigation equipment led to 269 casualties [1]. This prompted U.S. authorities to recognize the need for a reliable global localization solution. As a result, the Global Positioning System (GPS) 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 [2]. 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

This research was supported by the National Science Foundation (NSF) under grant number ECCS-1646562.
Author addresses: S. Tiku and S. Pasricha, Department of Electrical and Computer Engineering, Colorado State University, Fort Collins, Colorado, USA, 80523-1373; email: {saideep, sudeep}@colostate.edu.
Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
© 2019 Association for Computing Machinery.
1539-9087/2019/11-ART114 $15.00
http://dx.doi.org/10.1145/3362036

RIGHTSLINK

works have started to focus on the modeling and characterization of GPS spoofing [42] and time reliability–based attacks [43] and further propose the utilization of crowdsourcing methodologies to detect and localize spoofing attacks [44]. Regardless of such advances, the recent history of attacks on GPS for outdoor navigation [3, 4] motivates stronger security features. However, indoor localization is an emerging technology with a similar purpose and is poised to reinvent the way we navigate within buildings and subterranean locales [49]. However, on the academic front, limited attention is being payed toward 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 Wi-Fi 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 Wi-Fi router positions on that map with Google [5]. Nowadays, companies such as Amazon and Target are also starting to track customers at their stores [6]. 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 [45], 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., Wi-Fi Access Points or WAPs) to produce incorrect localization information [7, 8]. 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, [6, 9, 10] (trilateration) or the angles at which signals from two or more APs are received [8, 11] (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 [12]. 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 triangula-

RIGHTSLINK

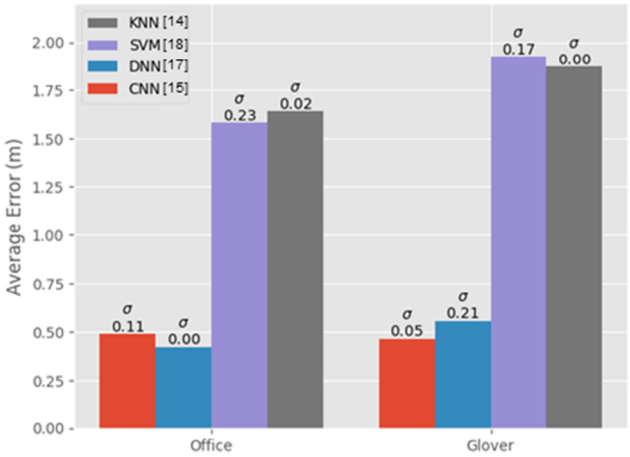


Fig. 1. Average indoor localization error (in meters) for fingerprinting techniques based on deep neural networks (DNNs), convolutional neural networks (CNNs), support vector machines (SVM), and *k*-nearest-neighbor (KNN). Results are shown for two different indoor paths.

tion [13], 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 Received Signal Strength Indicator (RSSI) signature obtained from APs accessible at that location [14, 15] (fingerprinting is discussed in more detail in Section 2). 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 of interest 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). Among 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.

Simple fingerprinting-based indoor localization frameworks that use techniques such as *k*-nearest-neighbor (KNN) can utilize outlier detection-based techniques to overcome some security issues [16]. However, recent work on improving Wi-Fi 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) [15] have been proposed and shown to improve fingerprint-based indoor localization accuracy on smartphones. Figure 1 shows the improvements when using CNN and deep neural network– (DNN) [17] based localization approaches as compared to more traditional techniques such as KNN [14] and support vector machines (SVM) [18]. Based on the improvements achieved through CNN- and DNN-based algorithms, indoor localization solutions in the future are expected to benefit from

RIGHTSLINK

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 article, 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 [15] as an example and propose security enhancements for it. The novel contributions of our work are as follows:

- 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 Reference [15] and Reference [17];
- For the first time, we conduct an in-depth experimental analysis on the impact of AP-based attacks on CNN- [15] and DNN- [15] 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 Reference [15] (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.

2 BACKGROUND AND RELATED WORK

2.1 Received Signal Strength Indicator

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 1 mW). The value of RSSI is reported in dBm and is given by

$$RSSI\ (dBm) = 10 \cdot log \frac{P_r}{P_{ref}}. \tag{1}$$

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

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

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 Wi-Fi capable smartphone from a Wi-Fi AP. However, the free space models based on Equations (1) and (2) do not extend well for practical applications. In reality, the propagation of radio signals is influenced by various effects. Figure 2 illustrates some of these effects as a radio signal travels from its source (WAP2) toward 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 (2), which was designed to function in free space.

2.2 Fingerprint-based Indoor Localization

Since the first efforts on fingerprinting-based indoor localization about two decades ago, such as with the work in RADAR [19], a significant level of advancement has been achieved in this

RIGHTSLINK

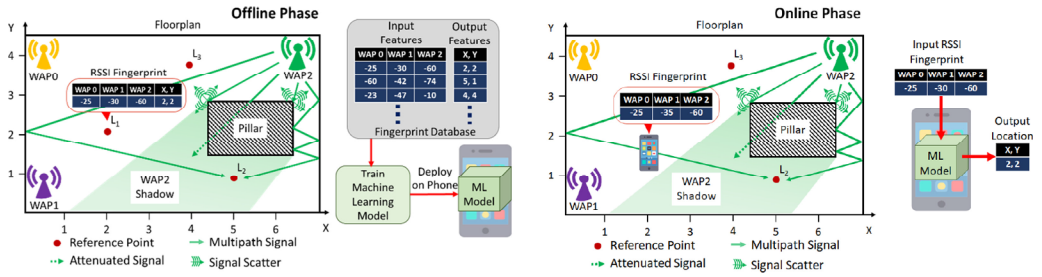


Fig. 2. A representation of the offline and online phases in the fingerprinting process for indoor localization for a given floorplan.

area. However, the general premise of fingerprinting-based indoor localization has remained unchanged. As shown in Figure 2, fingerprinting-based localization is carried out in two phases. In the first phase (called the offline or training phase), the RSSI values for visible WAPs are collected for a given floorplan at reference points L1, L2, L3, and so on, 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 2) 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 2. 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 are available in the training phase [41]. 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 2) 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) [20], Bluetooth [21], ZigBee [22], and Wi-Fi [14]. 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. However, Wi-Fi-based indoor localization frameworks have gained traction due to the ubiquitous availability of WAPs in indoor locales and the fact that most people nowadays carry smartphones that come equipped with Wi-Fi transceivers, making WAP-based indoor localization a cost-effective and popular choice [14, 15]. For this reason, in our work, we assume the use of WAPs as signal sources for fingerprinting-based indoor localization.

2.3 Challenges with Indoor Localization

As a result of the popularity of Wi-Fi fingerprinting, efforts in recent years have been made to overcome its limitations, such as energy efficiency [14], variations due to device heterogeneity [23, 50, 51], and temporal degradation effects on localization accuracy [24]. However, in recent years as indoor localization services are beginning to be prototyped and deployed, researchers have raised

RIGHTSLINK

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 [7, 8] and performing continuous sanity checks on the database using checksums [25]. 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 [26]. 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 [15].

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 [28, 29]. The jammer can achieve this goal by emitting Wi-Fi 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 Wi-Fi signals on a channel such that legitimate WAPs never sense the channel to be idle and therefore do not transmit any information [30]. 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 Wi-Fi 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 [47] and using machine learning [48]. However, both works solely focus on detecting a spoofing attack either in real time or offline. Techniques such as the one presented in Reference [32] 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.

RIGHTSLINK

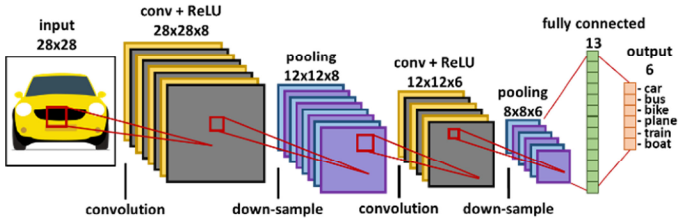


Fig. 3. A general representation of the various components of a CNN.

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., Reference [27], 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 [15, 17, 41] 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 [15] 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.

3 CNNLOC FRAMEWORK OVERVIEW

This section provides a brief overview of convolutional neural networks (CNNs) and the CNNLOC framework presented in Reference [15].

3.1 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 3, a CNN model has three main functional components or layers: convolution+ Regularized Linear Unit (ReLU), 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 Reference [15] and Reference [33].

RIGHTSLINK

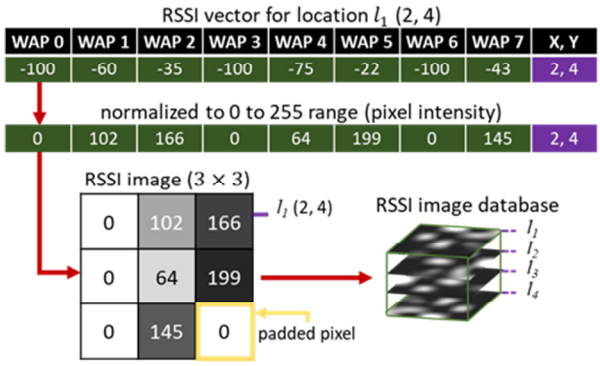


Fig. 4. A simplified overview of the conversion of an RSSI fingerprint to an image in the CNNLOC [15] indoor localization framework.

3.2 Indoor Localization with CNNLOC

The CNNLOC indoor localization framework [15] 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.

A simplified overview of the process of converting an RSSI fingerprint vector into an image is shown in Figure 4. 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 4, 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 4. 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.

4 LOCALIZATION SECURITY ANALYSIS

In this section, we perform a WAP RSSI vulnerability analysis on the deep learning–based indoor localization frameworks presented in Reference [15] (CNNLOC) and Reference [17] (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 5. The Office and Glover paths in the figure are 64 and 88m long and the reference locations used to capture Wi-Fi 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 7). We used an HTC U11 smartphone [39] to capture Wi-Fi fingerprints along the indoor paths and test for localization accuracy.

RIGHTSLINK

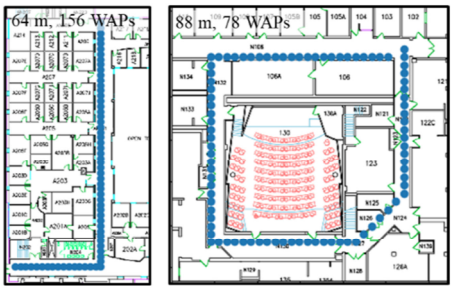


Fig. 5. Two indoor benchmark paths (Glover and Office) with reference points denoted by blue markers. The path lengths and Wi-Fi densities are denoted at the top of the maps.

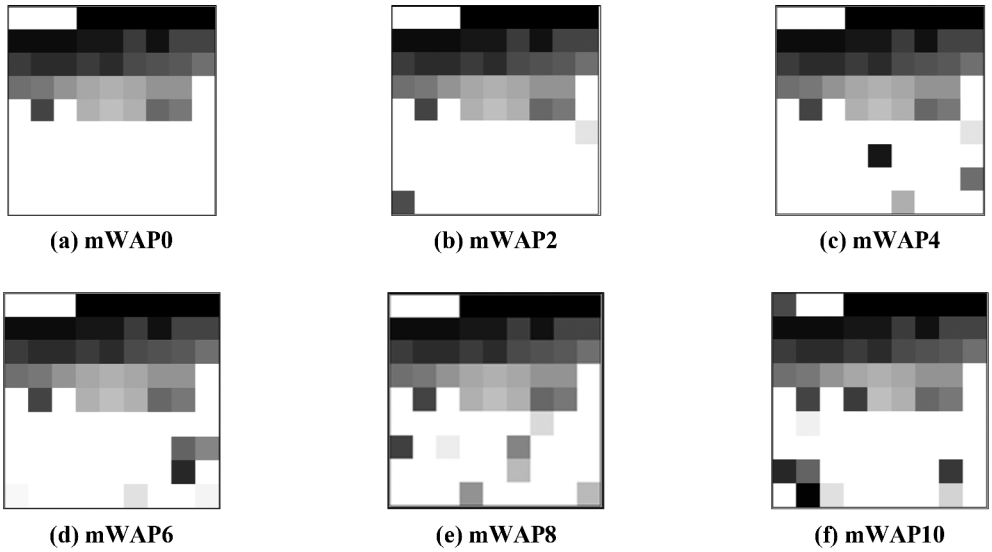


Fig. 6. Fingerprint images generated from RSSI vectors using the methodology described in CNNLOC [15]; (a) represents the "mWAP0" fingerprint image that should be ideally generated when the initial RSSI vector is not tainted by a malicious WAP (mWAP = 0); panels (b)-(f) show fingerprint images in the 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.

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 Wi-Fi RSSI fingerprints in the presence of one or more malicious WAP nodes (Wi-Fi 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.

RIGHTSLINK

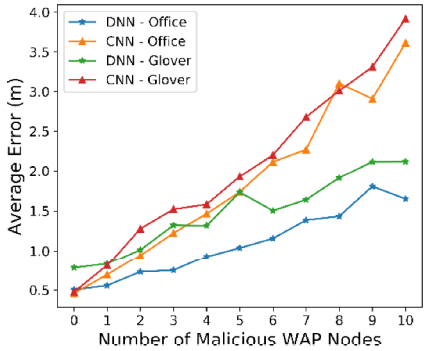


Fig. 7. Results for the impact of malicious WAPs on deep learning model accuracy on the Office and Glover paths. Average localization error for the CNN [15] and DNN [17] localization frameworks is shown for an increasing number of malicious WAPs.

Figure 6 shows the fingerprint images generated using an RSSI fingerprint based on the methodology described in CNNLOC [15]. 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 6(a). This image (Figure 6(a)) is not tainted by malicious WAPs (mWAPs) in the surrounding area and therefore is labeled as "mWAP0." The image labeled "mWAP2" (Figure 6(b)) is generated for the case when two WAPs of 78 are malicious WAPs that generate spurious signals between −100 to 0 dBm (their impact can be clearly seen with the two non-white pixels on the bottom half of the image). Similarly, Figure 6(c)–(f) shows 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 [34] may be applied to remove them. However, such filtering is not always possible. For instance, in Figure 6(d) with six malicious WAPs, we observe only five tainted pixels that are visually decipherable as compared to the untainted image (Figure 6(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 6(e)) and mWAP10 (Figure 6(f)).

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 [15] and a DNN-based [17] indoor localization framework. The results of this experiment are shown in Figure 7. We captured the average indoor localization error for the Office and the Glover paths (shown earlier in Figure 5) 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 a 100 trials and averaged the resulting localization error. From Figure 7, 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 toward 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.

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 8. We can observe that the worst-case localization errors for DNN and CNN models are significantly higher than the

RIGHTSLINK

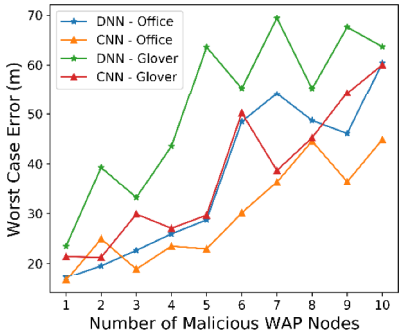


Fig. 8. Worst-case localization error for CNN and DNN, with respect to increasing number of malicious WAPs on the Office and Glover paths.

average errors shown in Figure 7 as the number of malicious WAPs are increased. With only one 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 50m with only six 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 article 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 [33]. 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 [36].

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.

5 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 as follows:

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

RIGHTSLINK

- 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 Wi-Fi scans; and the localization app translates the captured Wi-Fi 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 Wi-Fi 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 Wi-Fi or any other portable battery powered Wi-Fi transceiver to capture data about 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 Wi-Fi fingerprints while capturing steps taken and walking direction data, similar to the approach described in Reference [37]; this would allow anyone with a smartphone to create their own fingerprint database, which can be used to more strategically place Wi-Fi 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 runtime (i.e., in the online phase).

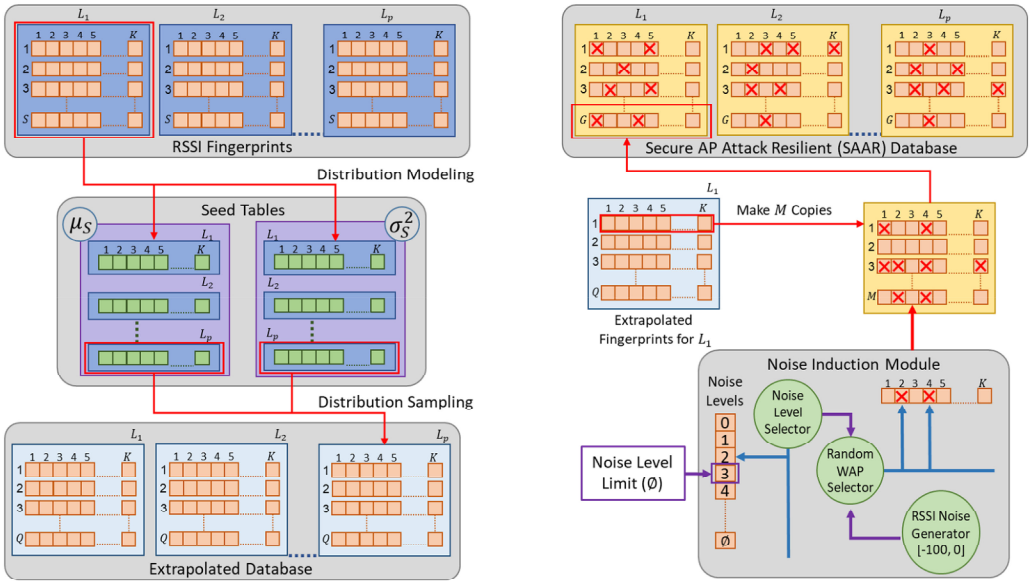
6 S-CNNLOC FRAMEWORK

In this section, we discuss the design of our S-CNNLOC framework to overcome the vulnerability of the CNNLOC [15] indoor localization framework against malicious WAP-based jamming and spoofing attacks in indoor environments.

6.1 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 Reference [15]). In general, deep learning models often require a large number of samples per class to produce good results. However, capturing Wi-Fi 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).

RIGHTSLINK



(a) Fingerprint database extrapolation **(b) Malicious behavior induction**

Fig. 9. 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 9(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 9(a)). The seed tables can be represented as:

$$\mu_{S(i,j)},\ \sigma^2_{S(i,j)},\quad i \in [1, K]\,,\ j \in [1,\ P], \tag{3}$$

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 9(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:

$$RSSI_{(i,j)} \sim N(\mu_{S(i,j)}, \sigma^2_{S(i,j)})\ \forall\ i \in [1, K], j \in [1, P], \tag{4}$$

where $RSSI(i, j)$ is the RSSI in dBm of the ith WAP at the jth 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

RIGHTSLINK

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 9(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.

6.2 Malicious Behavior Induction

From our analysis of CNN-based indoor localization in Section 4, we observed that fluctuations in one individual pixel value of the Wi-Fi 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 Wi-Fi 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 [38] 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 Reference [38] 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 4, 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. Toward this goal, as shown in Figure 9(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 9(b)) has three major components. For a given RSSI vector, the noise level selector submodule picks values from a discrete uniform distribution such that$\theta \sim U\{0, Ø\}$, 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:

$$\begin{aligned} &w_c \sim U\{1, K\}, \quad c \in [1, \theta] \\ &s.t., W_\theta = \{w_1, w_2, w_3 \ldots w_\theta\}. \end{aligned} \tag{5}$$

The newly generated RSSI vectors ($RSSI^{Noisy}_{(i,\ j)}$) are tainted by random noise at the *i*th WAP position, if the WAP was chosen by the random WAP selector submodule as shown by

RIGHTSLINK

Equation (6):

$$RSSI_{(i,j)}^{Noisy} = \begin{cases} I, & if\ i \in W_\theta \\ RSSI_{(i,j)}, & otherwise \end{cases} \quad j \in [1, P], I \sim U\{-100, 0\}, \tag{6}$$

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 (4) 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.

7 EXPERIMENTS

7.1 Experimental Setup

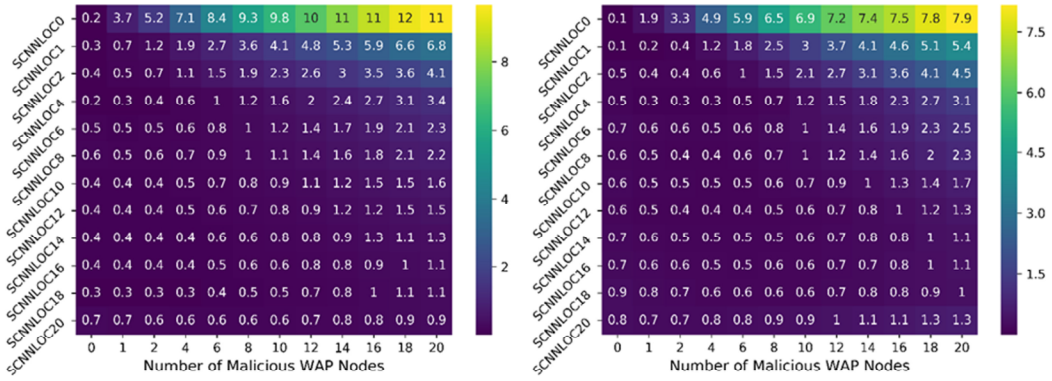
We initially compare the accuracy and stability of our proposed (S-CNNLOC) framework to its vulnerable counterpart (CNNLOC [15]) using two benchmark paths. These paths are shown in Figure 5 with each fingerprinted location (reference point) denoted by a blue marker. The paths were selected due to their salient features that may impact location accuracy in different ways. The 64-m 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-m 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-m intervals and captured 10 Wi-Fi 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 [15]. The trained S-CNNLOC model was deployed as an Android app on the HTC U11 smartphone. The values of Q and M (discussed in Section 6) are set to 100 and 10, respectively. Based on these values of Q and M, the Office path has 64,000 samples and the Glover path has 88,000 samples. To study the impact of malicious WAPs on indoor localization performance, we used a real Wi-Fi transceiver [40] 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.

7.2 Experimental Results

7.2.1 Analysis of Noise Induction Aggressiveness. We first performed a sensitivity analysis on the value of Ø (upper limit of noise induction; discussed in Section 6.2). Several CNN models were trained: S-CNNLOC1 (Ø = 0; no malicious WAPs), S-CNNLOC2 (Ø = 1), up to S-CNNLOC20

RIGHTSLINK



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

Fig. 10. 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).

(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.

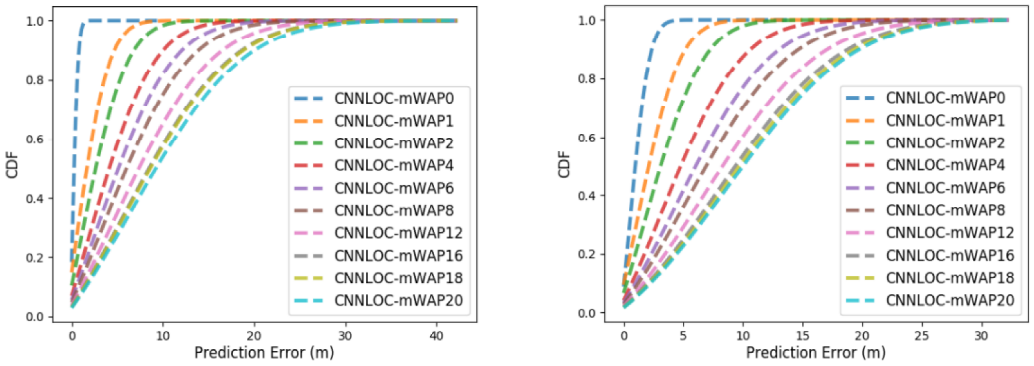
Figure 10 shows the heatmap for the mean localization errors (in meters) with annotated standard deviation of various scenarios on the Office path (Figure 10(a)) and the Glover path (Figure 10(b)). The *y*-axis shows various S-CNNLOC variants with different values of Ø varying from 1 to 20. The *x*-axis shows the number of mWAPs present in the online phase. In Figure 10, 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 10(a) and (b) represents the vulnerability of the specific S-CNNLOC model to an increasing number of mWAP nodes.

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 perform 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).

7.2.2 Comparison of Attack Vulnerability. In this section, we contrast the performance of our proposed S-CNNLOC framework with CNNLOC [15]. Figure 11(a) and (b) shows the cumulative

RIGHTSLINK



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

Fig. 11. Localization performance of CNNLOC [15] with a varying number of malicious WAPs (from 0 to 20) in the online phase.

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 m for a majority of scenarios) when there are no malicious WAPs (CNNLOC-mWAP0), However, in both the Office (Figure 11(a)) and the Glover paths (Figure 11(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 11(b), where the localization error does not scale by much when going from 12 malicious WAPs to 16 and 20.

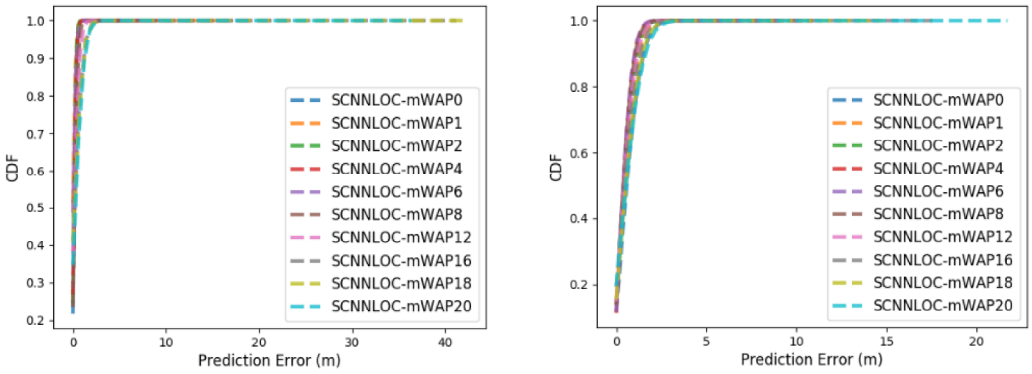
An important aspect to note from looking at Figure 11 is the significant drop in localization accuracy when going from a scenario with no malicious WAPs (CNNLOC-mWAP0) to a scenario with one malicious WAP (CNNLOC-mWAP1). This accuracy drop is apparent on both paths and clearly depicts the high vulnerability of unsecured CNN models to the presence of even a single malicious WAP node.

From Figure 11, we can conclude that a malicious third party can significantly degrade the localization accuracy of a CNN-based indoor localization model such as CNNLOC [15], with just a very small number of malicious WAP nodes.

Figure 12 highlights the resiliency of the S-CNNLOC model toward malicious WAP-based attacks for the same setup as for the experiment with CNNLOC in Figure 11, 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 m for the Office path (Figure 12(a)) and under 3.5 m for the Glover path (Figure 12(b)). The S-CNNLOC model for the Office path performs better than for the Glover path as the Wi-Fi density on the Office path is about 2× the Wi-Fi density of the Glover path, and thus malicious WAPs only impact a small fraction of the total WAPs along the Office path.

In summary, based on the results shown in Figures 11 and 12, 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 [15], for the Office and Glover paths.

RIGHTSLINK



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

Fig. 12. Localization performance of our S-CNNLOC with a varying number of malicious WAPs (from 0 to 20) in the online phase.

Table 1. 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

7.2.3 Extended Analysis on Additional Benchmark Paths. We conducted further experimental analysis on a more diverse set of benchmark indoor paths. Table 1 presents the salient features of the three new benchmark paths used in this analysis. The benchmark path suite shown in Table 1 consists of the EngrLabs, LibStudy, and the Sciences paths, with a description of environmental factors that may affect the localization performance of Wi-Fi-based indoor localization frameworks. Each path has a length ranging from 58 to 68 m and 10 Wi-Fi fingerprint samples were collected at 1-m intervals on each path, similarly 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 13 presents the means and standard deviations of the localization error with our proposed S-CNNLOC and the CNNLOC [15] 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 two and four 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 Wi-Fi network compared to the EngrLabs

RIGHTSLINK

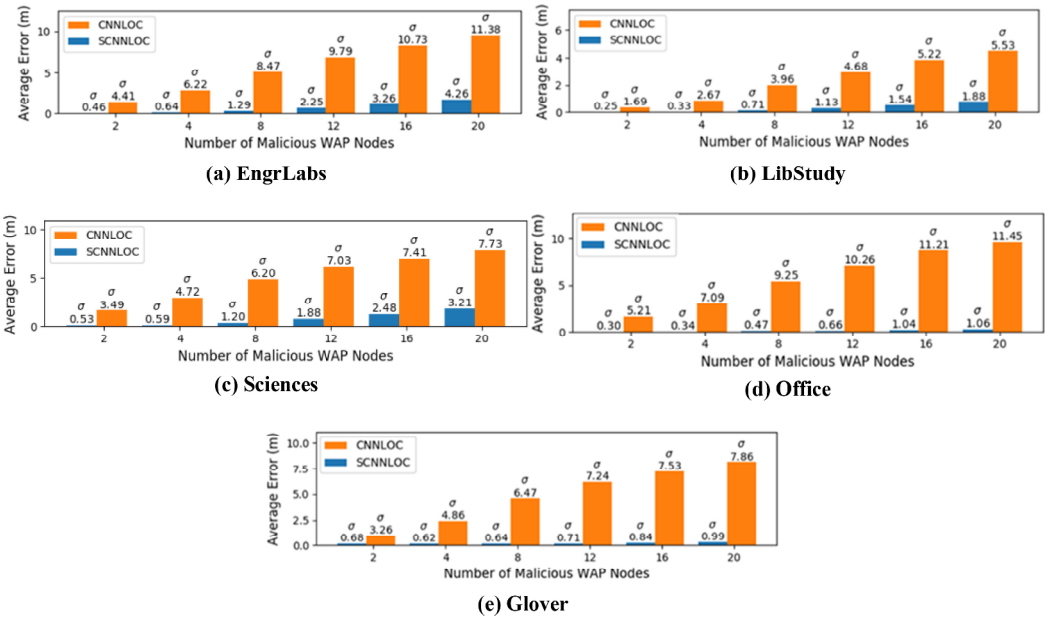


Fig. 13. The average localization error and its standard deviation of the proposed S-CNNLOC framework as compared to CNNLOC [15] for the benchmark path suite from Table 1.

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 [15] 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.

8 GENERALITY OF PROPOSED APPROACH

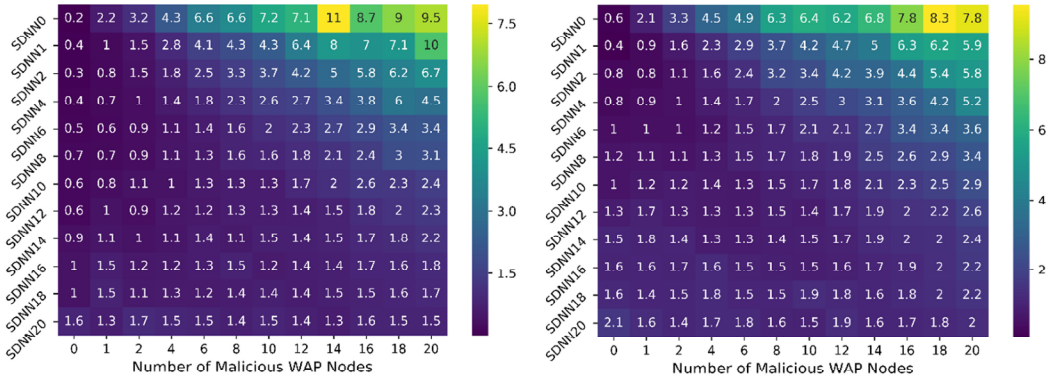
In this section, we highlight the generality and the versatile nature of our proposed security aware approach by applying it to another deep learning–based approach proposed in Reference [17]. We first present a discussion of the proposed work in Reference [17]. Later, we use Wi-Fi fingerprints generated in Section 7.1 to train the secure-DNN (SDNN) model and compare its prediction accuracy results to the conventional methodology described in Reference [17].

8.1 Denoising Autoencoder-based DNN Framework

The DNN-based approach in Reference [17] 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.

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

RIGHTSLINK



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

Fig. 14. 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 $\emptyset = 0, \emptyset = 2, \ldots, \emptyset = 20$ (y-axis).

output. Further, the HMM allows for greater resistance to minor variations in WAP RSSI over time.

8.2 Security-aware DNN Training in the Offline Phase

To train the SDNN model, we use the augmented security aware fingerprints used to train the S-CNNLOC 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 7.2.1. The results for this experiment are captured in Figure 14.

In Figure 14, we observe that the mean localization errors for the baseline SDNN0 models for the Office and Glover paths increase by 48× and 13× 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 [17] trained with $\emptyset = 18$. In the next subsection, we present an extended analysis on the performance of SDNN as compared to a conventional unsecured DNN model.

Figure 15 presents an analysis on the stability of the conventional unsecured DNN-based framework [17] 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 1. From Figure 15, we observe the prediction accuracy of the conventional DNN-based approach presented in Reference [17] systematically degrades (increased average error) as the number of stochastically placed malicious Wi-Fi access points on various paths are increased. The SDA stage in Reference [17] 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 Reference [17] 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

RIGHTSLINK

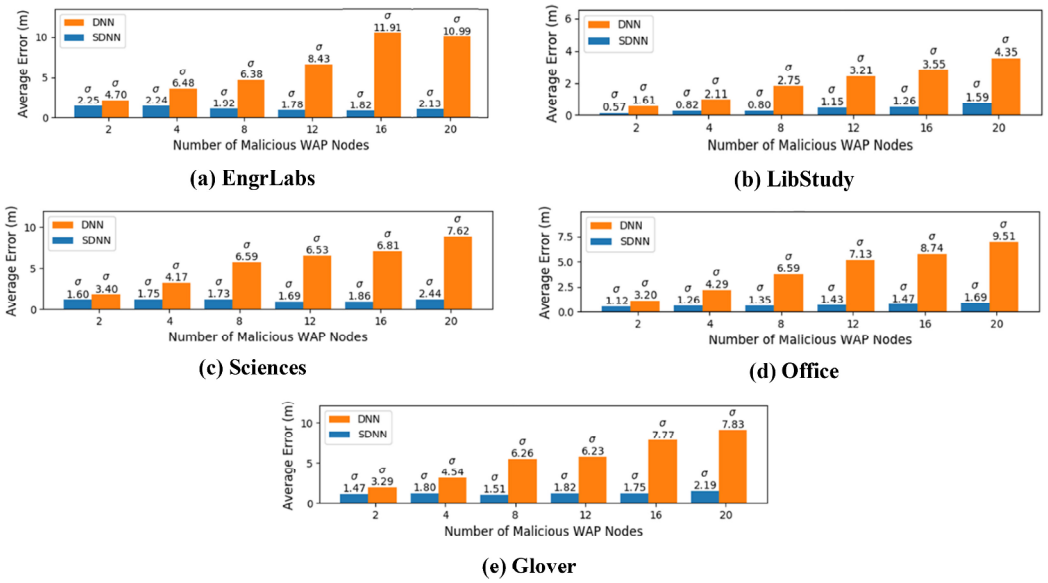


Fig. 15. The average localization error and its standard deviation of the proposed S-DNN framework as compared to DNN [17] for the benchmark path suite from Table 1.

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.

However, 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 15, this greatly improves SDNN's resilience to malicious WAPs in the online phase and delivers up to 10× 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.

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.

9 CONCLUSIONS AND FUTURE WORK

In this article, 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 WAP spoofing and jamming attacks. Our analysis highlighted the significant degradation in localization accuracy

RIGHTSLINK

that can be induced by an attacker with very minimal effort. For instance, our experimental studies suggest that an unsecured CNN-based indoor localization solution can place a user up to 50 m 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 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.

As a part of future work, we will be focusing on improving the quality of localization and navigation. Toward this goal, a possible extension of our work can be to predict the path taken by the user using multiple Wi-Fi fingerprints as an attack is taking place. In such situations, the machine learning model could correct a previous prediction (path taken) based on the upcoming predictions and vice versa. This methodology may improve the localization accuracy and stability in corner cases in the online phase where fingerprints at a location are similar in structure to others fingerprint that are spatially separated by large distances.

REFERENCES

[1] The Plane Crash That Gave Americans GPS. 2019. Retrieved from https://www.the atlantic.com/technology/archive/2014/11/the-plane-crash-that-gave-americans-gps/382204/.

[2] A brief history of GPS. 2019. Retrieved fromhttps://www.pcworld.com/article/2000276/a-brief-history-of-gps.

[3] This GPS Spoofing Hack Can Really Mess with Your Google Maps Trips. 2019. Retrieved from https://www.forbes.com/sites/thomasbrewster/2018/07/12/google-maps-gps-hack-takes-victims-to-ghost-locations/.

https://www.gpsworld.com/spoofing-in-the-black-sea-what-really-happened/.

[5] Wi-Fi RTT (IEEE 802.11mc). 2019. Retrieved from https://www.source.android.com/devices/tech/connect/wifi-rtt.

[6] Top 33 indoor localization services in the US. 2019. Retrieved from https://www.technavio.com/blog/top-33-indoor-location-based-services-lbs-companies-in-the-us.

[7] Y. Chen, W. Sun, and J. Juang. 2010. Outlier detection technique for RSS-based localization problems in wireless sensor networks. In *Proceedings of the International Conference on Structural Integrity and Exhibition (SICE'10).*

[8] A. Khalajmehrabadi, N. Gatsis, D. J. Pack, and D. Akopian. 2017. A joint indoor WLAN localization and outlier detection scheme using LAS-SO and elastic-net optimization techniques. *IEEE Trans. Mobile Comput.* 16, 8 (2017), 2079–2092.

[9] J. Schmitz, M. Hernández, and R. Mathar. 2016. Real-time in-door localization with TDOA and distributed software de-fined radio: Demonstration abstract. In *Proceedings of the Conference on Information Processing in Sensor Networks (IPSN'16).*

[10] D. Vasisht, S. Kumar, and D. Katabi. 2015. Sub-nanosecond time of flight on commercial Wi-Fi cards. In *Proceedings of the Special Interest Group on Data Communication Conference (SIGCOMM'15).*

[11] Z. Chen, Z. Li, X. Zhang, G. Zhu, Y. Xu, J. Xiong, and X. Wang. 2017. AWL: Turning spatial aliasing from foe to friend for accurate WiFi localization. In *Proceedings of the Conference on Emerging Networking Experiments and Technologies (CoNEXT'17).*

[12] Z. Lu, W. Wang, and C. Wang. 2014. Modeling, evaluation and detection of jamming attacks in time-critical wireless applications. *IEEE Trans. Mobile Comput.* 13, 8 (2014), 1746–1759.

[13] E. Soltanaghaei, A. Kalyanaraman, and K. Whitehouse. 2018. Multipath tri-angulation: Decimeter-level wi-fi localization and orientation with a single unaided receiver. In *Proceedings of the Conference on Mobile Systems, Applications, and Services (MobiSys'18).*

[14] S. Pasricha, V. Ugave, C. W. Anderson, and Q. Han. 2015. LearnLoc: A framework for smart indoor localization with embedded mobile devices. In *Proceedings of the Conference on Hardware/Software Codesign and System Synthesis (CODES+ISSS'15).*

[15] A. Mittal, S. Tiku, and S. Pasricha. 2018. Adapting convolutional neural networks for indoor localization with smart mobile devices. In *Proceedings of the Conference on the Great Lakes Symposium on VLSI (GLSVLSI'18).*

[16] W. Meng, W. Xiao, W. Ni, and L. Xie. 2011. Secure and robust Wi-Fi finger-printing indoor localization. In *Proceedings of the Conference on Indoor Positioning and Indoor Navigation (IPIN'11).*

RIGHTSLINK

[17] W. Zhang et al. 2016. Deep neural networks for wireless localization in indoor and outdoor environments. *Neurocomputing* 194 (2016), 279–287.
[18] Y. K. Cheng, H. J. Chou, and R. Y. Chang. 2016. Machine-learning indoor localization with access point selection and signal strength reconstruction. In *Proceedings of the Vehicular Technology Conference (VTC'16)*.
[19] P. Bahl and V. Padmanabhan. 2000. RADAR: An in-building RF-based user location and tracking system. In *Proceedings of the IEEE International Conference on Computer Communications (INFOCOM'00)*.
[20] Ubisense Research Network. 2017. Retrieved from http://www.ubisense.net/.
[21] P. Dickinson, G. Cielniak, O. Szymanezyk, and M. Mannion. 2016. Indoor positioning of shoppers using a network of Bluetooth Low Energy beacons. In *Proceedings of the Indoor Positioning and Indoor Navigation (IPIN'16)*.
[22] S. Lau, T. Lin, T. Huang, I. Ng, and P. Huang. 2009. A measurement study of zigbee-based indoor localization systems under RF interference. In *Proceedings of the Workshop on Experimental Evaluation and Characterization (WINTECH'09)*.
[23] H. Zou et al. 2016. A robust indoor positioning system based on the procrustes analysis and weighted extreme learning machine. *IEEE Trans. Wireless Comput.* 15, 2 (2016), 1252–1266.
[24] L. Chang, X. Chen, J. Wang, D. Fang, C. Liu, Z. Tang, and W. Nie. 2015. TaLc: Time adaptive indoor localization with little cost. In *Proceedings of the MobiCom Workshop on Challenged Networks (CHANTS'15)*.
[25] D. Barbará, R. Goel, and S. Jajodia. 2000. Using checksums to detect data corruption. In *Proceedings of the International Conference on Extending Database Technology*.
[26] L. Ou, Z. Qin, Y. Liu, H. Yin, Y. Hu, and H. Chen. 2016. Multi-user location correlation protection with differential privacy. In *Proceedings of the International Conference on Parallel and Distributed Systems (ICPADS'16)*.
[27] C. Wu, Z. Yang, and Y. Liu. 2015. Smartphones based crowdsourcing for indoor localization. In *IEEE Trans. Mobile Comput.* 14, 2 (2015), 444–457.
[28] L. Lazos and M. Krunz. 2011. Selective jamming/dropping insider attacks in wireless mesh networks. *IEEE Trans. Netw.* 25, 1 (2011), 30–34.
[29] Z. Lu, W. Wang, and C. Wang. 2014. Modeling, evaluation and detection of jamming attacks in time-critical wireless applications. *IEEE Trans. Mobile Comput.* 13, 8 (2014), 1746—1759.
[30] W. Xu, W. Trappe, Y. Zhang, and T. Wood. 2005. The feasibility of launch-ing and detecting jamming attacks in wireless networks. In *Proceedings of the Conference on Mobile ad hoc Networking and Computing (MobiHoc'05)*.
[31] C. Wang, L. Zhu, and L. Gong et al. 2018. Accurate sybil attack detection based on fine-grained physical channel information. *Sensors* 18, 3 (2018), 878.
[32] A. A. A. Silva et al. 2015. Predicting model for identifying the malicious activity of nodes in MANETs. In *Proceedings of the Symposium on Computers and Communication (ISCC'15)*.
[33] Y. LeCun et al. 1998. Gradient-based learning applied to document recognition. *Proc. IEEE* 86, 11 (1998), 2278–2324.
[34] J. S. Lee. 1983. Digital image smoothing and the sigma filter. *Comput. Vis. Graph. Image Process.* 24, 2 (1983), 255–269.
[35] X. Wang et al. 2015. DeepFi: Deep learning for indoor fingerprinting using channel state information. In *Proceedings of the Wireless Communications and Networking Conference (WCNC'15)*.
[36] J. Machaj, P. Brida, and R. Piché. 2011. Rank based fingerprinting algorithm for indoor positioning. In *Proceedings of the Indoor Positioning and Indoor Navigation (IPIN'11)*.
[37] Y. Shu et al. 2016. Gradient-based fingerprinting for indoor localization and tracking. *IEEE Trans. Indust. Electr.* 63, 4 (2016), 2424–2433.
[38] F. Zhang, N. Cai, J. Wu, G. Cen, H. Wang, and X. Chen. 2018. Image de-noising method based on a deep convolution neural network. *IET Image Process.* 12, 4 (2018), 485–493.
[39] HTC U11. Retrieved from https://www.htc.com/us/smartphones/htc-u11.
[40] MAC Address Clone on my TP-Link. Retrieved from https://www.tp-link.com/us/support/faq/68/.
[41] M. Mohammadi, A. Al-Fuqaha, M. Guizani, and J. Oh. 2018. Semisupervised deep reinforcement learning in support of iot and smart city ser-vices. *IoT J.* 5, 2 (2018), 624–635.
[42] J. A. Larcom and H. Liu. 2013. Modeling and characterization of GPS spoof-ing. In *Proceedings of the Conference on Technologies for Homeland Security (HST'13)*.
[43] C. Bonebrake and L. Ross O'Neil. 2014. Attacks on GPS time reliability. *IEEE Secur. Priv.* 12, 3 (2014), 82–84.
[44] K. Jansen, M. Schäfer, D. Moser, V. Lenders, C. Pöpper, and J. Schmitt. 2018. Crowd-GPS-Sec: Leveraging crowdsourcing to detect and localize GPS spoofing attacks. In *Proceedings of the Symposium on Security and Privacy (SP'18)*.
[45] K. Corina. and A. MacWilliams. 2011. Overview of indoor positioning technologies for context aware AAL applications. *Ambient Assisted Living*, 2011.
[46] V. Spasova and I. Iliev. 2014. A survey on automatic fall detection in the context of ambient assisted living systems. *Int. J. Adv. Comput. Res.* 4, 1 (2014), 94.
[47] Á. M. Guerrero-Higueras, N. DeCastro-García, F. J. Rodríguez-Lera, and V. Matellán. 2017. Empirical analysis of cyber-attacks to an indoor real time localization system for autonomous robots. *Comput. Secur.* 70 (2017), 422–435.

RIGHTSLINK

[48] Á. M. Guerrero-Higueras, N. Matellan. 2018. Detection of cyber-attacks to indoor real time localization systems for autonomous robots. *Robot. Auton. Syst.* 99 (2018), 75–83.
[49] C. Langlois, S. Tiku, and S. Pasricha. 2017. Indoor localization with smartphones: Harnessing the sensor suite in your pocket. *IEEE Cons. Electr. Mag.* 6, 4 (2017), 70–80.
[50] S. Tiku and S. Pasricha. 2019. PortLoc: A portable data-driven indoor localization framework for smartphones. *IEEE Des. Test* 36, 5 (2019), 18–26.
[51] S. Tiku, S. Pasricha, B. Notaros, and Q. Han. 2019. SHERPA: A lightweight smartphone heterogeneity resilient portable indoor localization framework. In *Proceedings of the IEEE International Conference on Embedded Software and Systems (ICESS'19)*.

Received August 2019; revised August 2019; accepted September 2019

RIGHTSLINK