Adversarial Deep Learning for Indoor Localization With Channel State Information Tensors

Xiangyu Wang, *Student Member, IEEE*, Xuyu Wang, *Member, IEEE*, Shiwen Mao, *Fellow, IEEE*, Jian Zhang, *Member, IEEE*, Senthilkumar C. G. Periaswamy, and Justin Patton

***Abstract*—Fingerprinting-based indoor localization has been a research focus for GPS denied areas. The development of neural networks has greatly promoted its application in indoor localization systems. However, recent studies showed that the machine learning models, including state-of-the-art neural networks, are vulnerable to adversarial examples, and thus, neural network-based indoor localization systems are also under the threat of adversarial attacks. To investigate the effect of adversarial attacks on indoor localization systems and to make such systems resilient to adversarial attacks, we propose AdvLoc, an adversarial deep learning for indoor localization system. With the proposed AdvLoc system, the effect of adversarial attacks on indoor localization is studied under six types of adversarial attack methods in both black-box attack and white-box attack scenarios. Furthermore, adversarial training is utilized in offline training of the proposed AdvLoc system, which is effective against first-order adversarial attacks. The proposed AdvLoc system is implemented with commodity WiFi devices and evaluated with extensive experiments in two representative indoor environments. The experimental results verify the robustness of the proposed system against first-order adversarial attacks in representative indoor environments.**

***Index Terms*—Adversarial defense, adversarial examples, black-box attack, deep learning, indoor localization, white-box attack.**

I. Introduction

LOCATION-BASED services have drawn significant attention driven by the increasing popularity of Internet of Things (IoT) devices and applications for global positioning system (GPS) denied indoor environments. Emerging indoor localization systems adopt various radio-frequency (RF) signals, such as WiFi, RFID, Bluetooth, etc., [1]–[6]. Among these, the WiFi signal has been dominant in such systems that provide location estimation for the indoor environment in people's daily life, because of its omnipresence and lower cost.

Traditionally, indoor localization systems rely on signal processing techniques to estimate the distance between a transmitter and receiver, the Angle of Arrival (AoA), or the time of flight (TOF), for inferring the target location. For example, SpotFi [7] utilized a modified multiple signal classification (MUSIC) algorithm to achieve decimeter-level location accuracy by using AoA and ToF. Chronos [8] was able to compute the subnanosecond ToF and estimate the target location with decimeter-level accuracy as well. However, these techniques are limited by the quality of the signal. In the indoor environment, WiFi signals are scattered and reflected by walls and furniture, which result in the inevitable noisy WiFi measurements, especially the phase readings. To alleviate the negative effect contributed by the offsets, indoor localization systems usually employ powerful but time-consuming algorithms, such as the super-resolution algorithm used in SpotFi, which limits their performance for real-time applications.

Deep learning has been a hot topic since it has achieved great success in solving tasks, such as data compression, speech recognition, and image classification. Recently, indoor localization systems also benefit from the development of deep learning. Compared with traditional systems, deep learning makes such systems more efficient in location estimation, even though it would take more time for training the model. The first work applying deep learning to indoor localization is DeepFi [2], which leverages a stack of restricted boltzmann machines (RBMs) to build an autoencoder for extracting location features from WiFi channel state information (CSI). PhaseFi [1] and BiLoc [9] further improve the the location accuracy by leveraging different CSI data. Due to the fingerprinting method, the localization problem is transferred to a matching problem. In the training stage, autoencoders have to be trained at each training location for extracting fingerprints. The training process could be time consuming and the size of fingerprint data may restrict the deployment of the localization system in mobile devices, which usually have limited storage. To overcome the drawbacks of the autoencoder-based localization systems, CiFi [3] is the first work to utilize deep convolutional neural networks (DCNN) for indoor localization. With DCNN, location estimation is treated as a multiclass classification problem. Thus, the localization system only needs to

Manuscript received 29 November 2021; revised 12 February 2022; accepted 26 February 2022. Date of publication 1 March 2022; date of current version 23 September 2022. This work was supported in part by the NSF under Grant ECCS-1923163, Grant CNS-2105416, and Grant CNS-2107190; in part by the RFID Laboratory; and in part by the Wireless Engineering Research and Education Center at Auburn University, Auburn, AL USA. *(Xiangyu Wang and Xuyu Wang are co-first authors.) (Corresponding author: Shiwen Mao.)*

Xiangyu Wang and Shiwen Mao are with the Department of Electrical and Computer Engineering, Auburn University, Auburn, AL 36849 USA (e-mail: xzw0042@auburn.edu; smao@ieee.org).

Xuyu Wang is with the Department of Computer Science, California State University, Sacramento, CA 95819 USA (e-mail: xuyu.wang@csus.edu).

Jian Zhang is with the Department of Electrical and Computer Engineering, Kennesaw State University, Marietta, GA 30144 USA (e-mail: jianzhang@ieee.org).

Senthilkumar C. G. Periaswamy and Justin Patton are with the RFID Laboratory, Auburn University, Auburn, AL 36849 USA (e-mail: szc0089@auburn.edu; jbp0033@auburn.edu).

Digital Object Identifier 10.1109/JIOT.2022.3155562

train one DCNN model in the training process, and the fingerprints collected in the training stage are not essential for location estimation once the DCNN is trained successfully. Like CiFi, received signal strength (RSS) and CSI amplitude have also been utilized to train the DCNN model [10]–[13]. ResLoc [14], [15] proposed a sharing learning approach based on deep residual learning, which uses the bimodal CSI tensor data.

Even though deep neural networks (DNN) have achieved excellent performance on classification problems, some counter-intuitive properties of DNNs have also been exposed along with its popularity. Szegedy *et al.* [16] found that several machine learning models, including state-of-the-art neural networks, are vulnerable to adversarial examples. Goodfellow [17] verified the discovery by misleading the GoogLeNet [18] with adversarial examples. Deep learning-based indoor localization systems also face the threat of adversarial attacks. To evaluate and counteract the threat of adversarial attacks to DNN-based indoor localization systems, we propose AdvLoc, an adversarial deep learning for indoor localization system. Like traditional DCNN-based systems, AdvLoc operates in two stages: 1) an offline training stage and 2) an online location estimation stage. We apply adversarial attacks in the online stage, where the perturbations generated by adversarial attacks are be introduced to the existing clean inputs of the DCNN. In the offline stage, the DCNN-based localization model will be trained adversarially to enhance its robustness against the adversarial examples. Unlike the image classification models, the DCNN model in the indoor localization system processes the online inputs that do not belong to any existing class in the training data set (i.e., the mobile device may be placed at an arbitrary location, rather than a known training locations). Using the AdvLoc system, we evaluate the effects of six types of mainstream adversarial attacks on DCNN-based indoor localization with respect to accuracy and location error. To defend against such attacks, adversarial training is implemented in the offline training of the models. The experimental results validate that adversarial training utilized in the proposed AdvLoc system is an effective means to counteract the location errors cause by the first-order adversarial attacks.

The main contributions made in this article can be summarized in the following.

1) We demonstrate the threat of adversarial attacks to deep learning-based indoor localization systems by visualizing the adversarial examples and evaluating the impact of the various magnitude of perturbation of adversarial examples to location estimation. The effect of six types of representative adversarial attacks, including gradient-based, optimization-based, and spatial transformation-based attacks, on the indoor localization system, is investigated in both white-box and black-box attack scenarios.
2) To the best of our knowledge, this is the first work to employ adversarial training to enhance the robustness of WiFi CSI-based indoor localization systems. We introduce adversarial training into the traditional DCNN-based indoor localization. In the white-box attack scenario, the modified loss function successfully alleviates the negative effect resulted from the first-order adversarial attacks, especially the fast gradient sign attack (FGSM) [17].
3) The proposed AdvLoc system is implemented with commodity 5-GHz WiFi. We verified its performance in two representative indoor environments with extensive experiments. The experimental results exhibit the threat of adversarial attacks and show that adversarial training effectively improves the robustness of the localization system when the input examples are manipulated by first-order adversarial attacks.

The remainder of this article is organized as follows. Section II reviews related work. We present the AdvLoc design in Section III and our experimental study in Section IV. Finally, Section V concludes this article.

II. Related Work

With the advances in computing power, the availability of data, and the development of open-source platforms, deep learning has been recognized as a powerful tool for many real-world problems that cannot be solved by conventional machine learning techniques. However, as Szegedy *et al.* first unveiled in [16], using image classification as an example, the resilience of deep learning has been exposed to the threat of adversarial attacks. Nowadays, most AI-based services, such as Apple Face ID and Amazon Alexa, are highly dependent on the progress of deep learning in image classification and natural language processing (NLP). The vulnerability of deep learning networks places user privacy and public safety at risk.

Following the discovery in [16], Finlayson *et al.* [19] investigated the vulnerabilities of the medical AI systems under adversarial attacks and pointed out that the adversarial attacks may already be in place and contribute to medical fraud. The diagnostic performance could be affected easily by adding a small perturbation generated by the common adversarial attacks, while the manipulated diagnostic probability could deceive the automated fraud detector evaluating the medical claims. Furthermore, Finlayson *et al.* also indicated that the adversarial attacks are effective for extremely accurate medical classifiers even if the prospective attackers do not have access to the deep learning model. In [20], both white-box and black-box projected gradient descent (PGD) attacks were used to generate adversarial examples. The result showed that state-of-the-art medical models were misled in both scenarios. Furthermore, researchers have applied adversarial attacks in other real-world scenarios. For example, Thys *et al.* [21] proposed an approach to generate adversarial patches to hide a person from a DCNN-based human detector. Sharif *et al.* [22] presented an approach for generating eyeglass frames to fool state-of-the-art face recognition systems (FRSs). The experimental results showed that their techniques were effective for black-box FRSs, as well as state-of-the-art face detection systems (FDSs).

Not only traditional DCNNs but also the spatiotemporal graph convolutional network (ST-GCN) is facing the threat of adversarial attacks. Unlike the medical AI systems relying on

DCNN for image classification, action recognition applications utilizing ST-GCN for processing the skeleton data obtained from RGB-D sensors [23], [24]. Liu *et al.* [25] proposed constrained iterative attacks for skeleton actions (CIASA), which was based on FGSM and was able to disturb the joint locations in an action sequence. Even though the features of graph nodes and graph structure were discrete with certain predefined structures, the basic FGSM attack was able to fool the ST-GCN in the form of nontargeted attacks.

Even though the textual data are different from image data composed of continuous pixel values, adversarial examples affect DNN for text-based tasks as well. Three types of perturbation strategies, namely, insertion, modification, and removal, were introduced in TextFool [26] based on the concept of FGSM. Papernot *et al.* [27] showed that the recurrent neural network (RNN) is not immune to the adversarial attacks. The attack methods used in crafting adversarial image examples could be adapted to generate sequential adversarial text by leveraging computational graph unfolding. In a recent work [28], we investigated the problem of adversarial attacks on solar power generation forecasting, which is a regression problem, and showed that both DNN and a LASSO-based statistical model were vulnerable.

Recently, there has been considerable interest of applying deep learning to wireless communications and networking problems [29]. Because adversarial attack has been a common threat to deep learning systems, researchers have also investigated the impact of adversarial attacks in wireless systems. For example, modulation recognition is a key technology of cognitive radio (CR), for which deep learning techniques have been developed. Sadeghi and Larsson [30] demonstrated how adversarial examples degrade the model performance of radio signal (modulation) classification. Compared with traditional attacks such as jamming, the adversarial attack required much less power since only small perturbations were generated. Lin *et al.* [31] evaluated four representative adversarial attacks on modulation recognition. The results showed that regardless of white box or black box, adversarial attacks could reduce the accuracy of the target model, while the performance of iterative attacks was superior to that of single step attacks. A thorough study of adversarial attacks on IoT device identification (or device fingerprinting) was reported in [32], which was to identify specific wireless transmitters based on received signals. Although following the same specifications and using the same protocols, the devices can still be distinguished by the small defects incurred during the manufacturing process or the aging process.

III. AdvLoc System

Due to the popularity of mobile devices, location information has been an essential part of IoT. Recently, an increasing number of researchers have focused on WiFi-based indoor localization because of the ubiquitous availability and low cost of WiFi devices. Many indoor localization systems [1], [2], [9] rely on the fingerprinting method, which means the fingerprints of known locations need to be measured and stored in a database for online localization. To reduce the storage requirement, some systems [3], [10] treat indoor WiFi fingerprinting as a classification problem, where DCNN becomes the best choice owing to its great success in image classification. As Szegedy *et al.* [16] revealed the vulnerability of DCNN models to adversarial examples, consequently, the DCNN-based localization systems would also be susceptible to adversarial attacks. To combat such threats, we propose the AdvLoc system in this article, which utilizes adversarial training in the offline stage to enhance the robustness of the network, making it immune to adversarial examples.

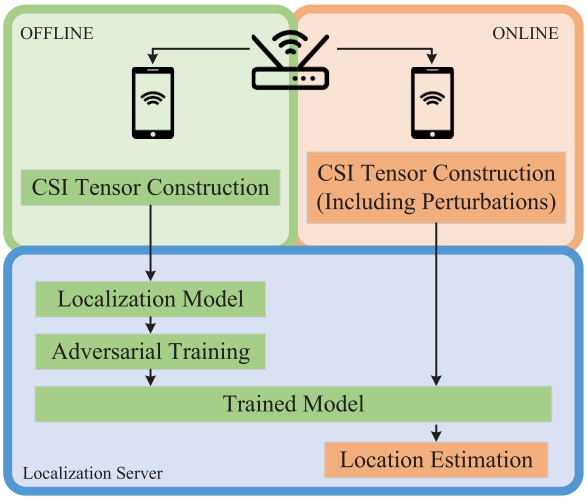

Fig. 1. Indoor localization architecture.

A. Architecture of the AdvLoc System

Fig. 1 depicts the architecture of the proposed AdvLoc system. Like traditional DCNN-based indoor localization systems, AdvLoc comprises of an offline stage and an online stage. In the offline stage, CSI tensors are constructed using the CSI data collected at the receiver for the mobile device placed at various known training locations. The core of AdvLoc is a deep residual learning model, ResNet [33], that learns location features from WiFi CSI data. The ResNet will be trained adversarially using CSI tensors to generate the model for online localization. New CSI data are collected from mobile devices placed at an unknown location in the online stage. The adversarial perturbations are generated and injected into CSI tensors in the online stage, while the new CSI tensors are constructed in the same way as in the offline stage. As a result, the wireless channel has no effect on the perturbations. Launching adversarial attacks during the process of CSI tensor generation is more feasible in both white-box and black-box scenarios (than from the channel or transmitter sides).

Specifically, in the offline stage, the training data set and verification data set are collected from identical positions. The collected observations, such as phase readings, are labeled by the coordinates of corresponding positions. The location with the highest similarity in the output of the DCNN model is selected as the output of the system. Therefore, we could assess how well our model fits the training data using the verification data set by examining the verification accuracy in

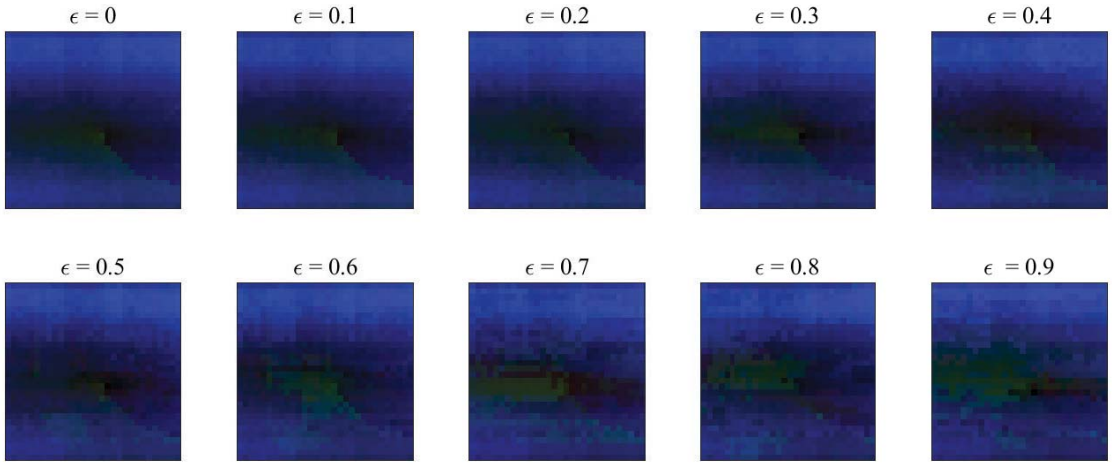

Fig. 2. Examples of CSI Tensors when different levels of pertubations are introduced, as indicated by the hyperparameter ϵ.

	Input Block	Conv_Block1	Conv_Block2	Conv_Block3	Conv_Block4	Output Block
ResNet-18	$\begin{bmatrix} 7\times 7, 64 \\ max\ pool \end{bmatrix}$	$\begin{bmatrix} 3\times 3, 64 \\ 3\times 3, 64 \end{bmatrix} \times 2$	$\begin{bmatrix} 3\times 3, 128 \\ 3\times 3, 128 \end{bmatrix} \times 2$	$\begin{bmatrix} 3\times 3, 256 \\ 3\times 3, 256 \end{bmatrix} \times 2$	$\begin{bmatrix} 3\times 3, 512 \\ 3\times 3, 512 \end{bmatrix} \times 2$	$\begin{bmatrix} average\ pool \\ fully\ connected\ layer \\ softmax \end{bmatrix}$
ResNet-50	$\begin{bmatrix} 7\times 7, 64 \\ max\ pool \end{bmatrix}$	$\begin{bmatrix} 1\times 1, 64 \\ 3\times 3, 64 \\ 1\times 1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1\times 1, 128 \\ 3\times 3, 128 \\ 1\times 1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1\times 1, 256 \\ 3\times 3, 256 \\ 1\times 1, 1024 \end{bmatrix} \times 6$	$\begin{bmatrix} 1\times 1, 512 \\ 3\times 3, 512 \\ 1\times 1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} average\ pooling \\ fully\ connected\ layer \\ softmax \end{bmatrix}$

Fig. 3. Architecture of the two ResNet models used in AdvLoc: ResNet-18 and ResNet-50 [33].

this stage. In the online stage, the testing data set is collected from the positions not used in the offline stage. Obviously, the classification accuracy would not be persuasive to demonstrate the localization performance of the system. In fact, the output of ResNet is used as the similarity to calculate the estimated location. The estimated location $\widehat{T}$ is computed by

$$\widehat{T} = \sum_{i=1}^{N} t_i \times p_i \tag{1}$$

where p_i is the output of the ResNet that depicts the similarity between the testing location and the training location i, and t_i is the known training location i.

B. CSI Tensor Construction

The CSI tensor used in AdvLoc consists of three slices. Two of the slices are generated with the estimated angle-of-arrival (AoA) values using the phase difference data from the three receiving antennas, while the third slice contains the measured CSI amplitude values. Considering that the Intel WiFi Link 5300 network interface card (NIC) only supports three antennas and 30 subcarriers for each antenna, the size of the CSI tensor is set to $30 \times 30 \times 3$. Fig. 2 depicts CSI tensors used in our AdvLoc system when different levels of perturbations are introduced (as indicated by the parameter ϵ). As we can see, when $\epsilon = 0$, no perturbation is added and the tensor is a clean input to the ResNet model. Whereas, the rest of the tensors are adversarial examples generated using the FGSM method, where ϵ is a hyperparameter that controls the magnitude of the perturbation. When ϵ is less than 0.4, the perturbation added in the tensors is negligible (i.e., visually invisible). However, the tensor will be distorted obviously, once ϵ is larger than 0.5. We shall study the relationship between ϵ and the location estimation error in the following sections.

C. Architecture of the ResNet Models

To investigate the effect of adversarial attacks on DCNN-based indoor localization systems, two popular ResNet models are adopted in the AdvLoc system, including ResNet-18 and ResNet-50 [33]. The ResNet-18 model will be leveraged as the localization model in our study of both white-box attacks and black attacks. In the study of black-box attacks, the ResNet-50 model will be trained as a substitute model for mimicking the localization model, i.e., the ResNet-18 model.

Fig. 3 presents the detailed structure of the localization models. The building blocks shown in the brackets depict the component of each block. For example, the input block of the ResNet-18 model includes 7×7 filters for generating 64 feature maps, then max pooling is leveraged to shrink the size of the feature maps. The Conv_Block4 of the ResNet-18 model is composed of two building units, each containing two 3×3 convolution layers. The shortcut connection exists in each building unit of the Conv_Blocks. Since localization is treated as a classification problem in the fingerprinting-based system, the cross-entropy loss is adopted in the training process.

D. Adversarial Attacks

Szegedy *et al.* [16] showed that adversarial examples hardly distinguishable from the originals can fool DCNN-based

image classifiers such as AlexNet [34]. Since then, security has become an important problem in AI/ML research, especially for privacy-sensitive applications such as localization. To better evaluate the resilience of DCNN-based localization systems against adversarial attacks and the effectiveness of defense strategies, we implement the following six types of adversarial attacks in this study.

1) Fast Gradient Sign Method: FGSM was proposed by Goodfellow *et al.* in 2015 [17]. The method obtains a perturbation, denoted by η, by calculating the gradient of the loss function $L(\cdot)$ with a given input, as

$$\boldsymbol{\eta} = \epsilon \cdot \text{sign}(\nabla_{\boldsymbol{x}} L(\theta, \boldsymbol{x}, y)) \tag{2}$$

where θ represents the parameters of a well-trained model; $\boldsymbol{x}$ and y are the input and its corresponding label, respectively; and ϵ is a hyperparameter, which controls the magnitude of the perturbation. Since $L(\cdot)$ is the loss function of the model, the perturbation η can be calculated by using the first derivative of $L(\theta, \boldsymbol{x}, y)$ through the backpropagation algorithm.

In 2017, Miyato *et al.* [35] modified FGSM by canceling the sign$(\cdot)$ function in (2). The new method, fast gradient method (FGM), is a generalization of FGSM, where the perturbation is give by

$$\boldsymbol{\eta} = \epsilon \cdot \frac{\nabla_{\boldsymbol{x}} L(\theta, \boldsymbol{x}, y)}{\|\nabla_{\boldsymbol{x}} L(\theta, \boldsymbol{x}, y)\|_2}. \tag{3}$$

With (3), the perturbation can be easily created. However, it is not safe to say that the perturbation will contribute to misclassification, even though the loss value for the target label to be misclassified is increased by introducing the perturbation.

2) Projected Gradient Descent: Based on the one-step FGM, an iterative version of FGM termed PGD was proposed in 2017 [36]. Madry *et al.* created the PGD adversary to enhance the robustness of the classifier against the first-order attacks. With the iterative method, the adversarial examples $\{\boldsymbol{x}_0^{\text{adv}}, \boldsymbol{x}_1^{\text{adv}}, \ldots, \boldsymbol{x}_{N+1}^{\text{adv}}\}$ are generated as follows:

$$\begin{aligned} \boldsymbol{x}_0^{\text{adv}} &= \boldsymbol{x} \\ \boldsymbol{x}_{N+1}^{\text{adv}} &= \text{Clip}_{x,\epsilon}\left\{\boldsymbol{x}_N^{\text{adv}} + \alpha \cdot \frac{\nabla_{\boldsymbol{x}} L(\theta, \boldsymbol{x})}{\|\nabla_{\boldsymbol{x}} L(\theta, \boldsymbol{x}, y)\|_2}\right\} \end{aligned} \tag{4}$$

where α is a hyperparameter for each iteration, which is usually set as ϵ/N for a given ϵ. With this approach, the perturbation is always small and around the original input $\boldsymbol{x}$ in the L^p ball. Also, $\text{Clip}_{x,\epsilon}$ is used to project the perturbation back into the L^p ball if necessary. PGD has been verified to be a stronger adversarial attack method than the one-step FGM/FGSM at the cost of transferability.

3) Momentum Iterative Method: Since PGD generates adversarial examples with a greedy approach along the direction of the gradient in each iteration, the local maxima could be reached easily, resulting in poor transferability. To solve this problem, the momentum-based method is integrated into FGSM. Instead of using the gradient in one iteration to update the perturbation, the momentum iterative method (MIM) leverages the gradient of the previous iterations to guide the update of the perturbation [37]. The memory of previous gradients can help to avoid the local maxima, which occur in PGD. Thus, it breaks the dilemma of choosing between the "underfitted" FGSM and the "overfitted" PGD.

To generate adversarial examples with MIM, we have

$$\begin{cases} \boldsymbol{g}_0 = 0 \\ \boldsymbol{x}_0^{\text{adv}} = \boldsymbol{x} \end{cases} \\ \begin{cases} \boldsymbol{g}_{N+1} = \mu \cdot \boldsymbol{g}_N + \frac{\nabla_{\boldsymbol{x}} L(\theta, \boldsymbol{x}_N^{\text{adv}}, y)}{\|\nabla_{\boldsymbol{x}} L(\theta, \boldsymbol{x}_N^{\text{adv}}, y)\|_1} \\ \boldsymbol{x}_{N+1}^{\text{adv}} = \boldsymbol{x}_N^{\text{adv}} + \alpha \cdot \text{sign}\left(\boldsymbol{g}_{N+1}\right). \end{cases} \tag{5}$$

Note that $\boldsymbol{g}_N$ includes the gradients from previous $(N-1)$ iterations with a decay factor of μ. Here α can also be set to ϵ/N when ϵ is given. Thus, MIM retains the transferability of adversarial examples under increased iterations.

4) DeepFool Attack: In FGSM/FGM, the choice of the hyperparameter ϵ significantly affects the performance of adversarial attacks, since ϵ decides the magnitude of perturbation. In DeepFool [38], perturbations are computed by solving optimization problems. For a binary affine classifier, $f(\boldsymbol{x}) = \boldsymbol{w}^T \boldsymbol{x} + b$, the optimal perturbation is given by

$$\begin{aligned} \boldsymbol{\eta}^*(\boldsymbol{x}) \coloneqq \text{argmin} \quad & \|\boldsymbol{\eta}\|_2 \\ \text{s.t.} \quad & \text{sign}(f(\boldsymbol{x_0} + \boldsymbol{\eta})) \neq \text{sign}(f(\boldsymbol{x}_0)) \end{aligned} \tag{6}$$

which has the following closed-form solution:

$$\boldsymbol{\eta}^*(\boldsymbol{x}) = -\frac{f(\boldsymbol{x_0})}{\|\boldsymbol{w}\|_2^2}\boldsymbol{w}. \tag{7}$$

The iterative method is adopted in DeepFool for general binary classifiers. In each iteration, Deepfool assumes f is linear in the neighborhood of the current $\boldsymbol{x}$. Hence the optimal perturbation is calculated as

$$\begin{aligned} \boldsymbol{\eta}^*(\boldsymbol{x}) = \underset{\boldsymbol{\eta}_N}{\text{argmin}} \quad & \left\|\boldsymbol{\eta}_N\right\|_2 \\ \text{s.t.} \quad & f(\boldsymbol{x}_N) + \nabla f(\boldsymbol{x}_N)^T \boldsymbol{\eta}_N = 0. \end{aligned} \tag{8}$$

Considering that multiclass classification can be split into multiple binary classification, Deepfool could also find the optimized perturbation effectively for a nonlinear multiclass neural network. Furthermore, it has been demonstrated that the adversarial examples generated by Deepfool have five times smaller perturbations comparing with those from FGSM on MNIST and CIFAR10 models.

5) Carlini Wagner Attack: Defensive distillation [39] is a popular defensive method, which robustifies neural networks to counteract adversarial examples. However, Carlini and Wagner proposed a type of attacks to make defensive distillation ineffective [40]. Among the various distance metrics used for evaluating similarities, the Carlini Wagner attacks (CW) are designed with the L_2, L_∞, and L_0 distance metrics. For the L_2 attack, adversarial examples are generated with $\boldsymbol{w}$, obtained by solving

$$\min \quad \left\{\left\|\frac{1}{2}(\tanh(\boldsymbol{w}) + 1) - x\right\|_2^2 + c \cdot f\left(\frac{1}{2}(\tanh(\boldsymbol{w}) + 1)\right)\right\} \tag{9}$$

where the loss function $f(\cdot)$ is defined as

$$f\left(\boldsymbol{x}^{\text{adv}}\right) = \max\left\{\max\left\{\zeta\left(\boldsymbol{x}'\right)_i : i \neq t\right\} - \zeta\left(\boldsymbol{x}^{\text{adv}}\right)_t, -\psi\right\} \tag{10}$$

where $\zeta(\cdot)_i$ is a logistic for class i, ψ controls the confidence with which the misclassification occurs, and c is a hyperparameter that tradeoffs between the magnitude of perturbation and success rate of attack. For the L_0 attack, considering that the L_0 metric is nondifferentiable, the pixels in $\boldsymbol{x}$ that affect the classifier significantly are selected and attacked with the Carlini and Wagner L_2 (CWL2) attack in an iterative manner.

To create adversarial examples with the L_∞ metric, the L_2 term in (9) is replaced by a penalty for any terms that exceed $\boldsymbol{\tau}$, i.e.,

$$\min \left\{ c \cdot f(\boldsymbol{x}+\boldsymbol{\eta}) + \sum_i \left[\left(\eta_i - \tau\right)^+ \right] \right\} \tag{11}$$

where $\boldsymbol{\tau}$ is decreased iteratively with an initial value of $\mathbf{1}$. Even though the CW attack has been demonstrated to have defeated the defensive distillation method, the time cost in generating adversarial examples using this method is much larger than that of all the previous attack methods.

6) Spatial Transformation Method: Unlike DeepFool and CW that construct adversarial examples by solving an optimization problem, the spatial transformation method (STM) constructs adversarial examples with a natural transformation of the original inputs [41]. The transformation parameters, i.e., $(\delta_u, \delta_v, \theta)$, could be optimized by the grid search or the PGD method. The position of a pixel (u, v) is updated as follows:

$$\begin{bmatrix} u' \\ v' \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \cdot \begin{bmatrix} u \\ v \end{bmatrix} + \begin{bmatrix} \delta_u \\ \delta_v \end{bmatrix}. \tag{12}$$

According to [41], STM can successfully defeat the CNN that was trained against an L_∞-bounded adversary.

E. White-Box and Black-Box Attacks

All the above attack methods are white-box attacks, which means that the adversary is capable of acquiring the knowledge of the target model, or even the training data set. This possibility is usually slim in practice, especially for accessing the model and data set related to personal privacy or homeland security. To make adversarial attacks more feasible, the more challenging black-box attacks have been investigated, where the attacker has no or limited knowledge of the model. We will also leverage black-box attack methods to evaluate the threat of adversarial attacks to the AdvLoc system.

A comparison of white-box and black-box attacks is shown in Fig. 4, where a substitute model is utilized to mimic the black-box model with infinite queries. Since information of the substitute model is open to the attacker, all of the attack methods designed for the white-box scenarios can be leveraged to fabricate adversarial examples in the black-box scenario. Due to the transferability of the adversarial examples, the black-box model would also be misled by the adversarial examples. However, this strategy is easy to be detected. Moreover, Papernot *et al.* [42] noticed that it will be intractable for attackers to build a substitute model with a limited number of queries. Thus, a jacobian-based data set augmentation technique (JAD) will be used in our AdvLoc system, which ensures that the substitute model is able to approximate the

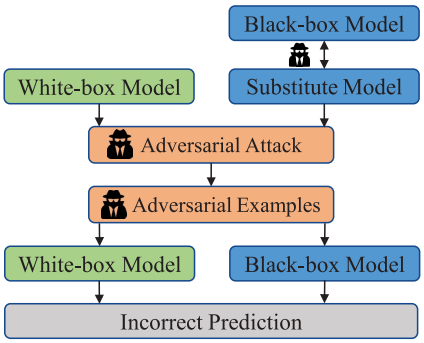

Fig. 4. Comparison of white-box and black-box attack approaches.

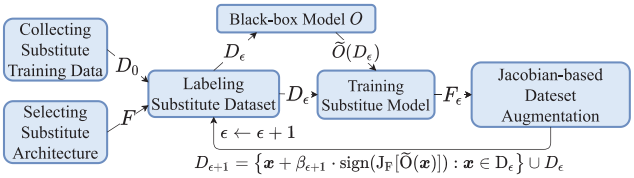

Fig. 5. Training the substitute model.

decision boundary of the black-box attack with a limited number of queries. Fig. 5 depicts the procedure of JAD. First, a small data set D_0 is collected and labeled by the black-box model O. The substitute model will be trained with the data set $(D_0, \widetilde{O}(D_0))$. Next, D_0 is augmented to generate a larger date set D_1 given by

$$D_1 = \left\{ \boldsymbol{x} + \beta \cdot \mathrm{sign}\left(J_F\left[\widetilde{O}(\boldsymbol{x})\right]\right) : \boldsymbol{x} \in D_0 \right\} \cup D_0 \tag{13}$$

where β is a parameter of augmentation, and J_F is the Jacobian matrix of the substitute model F. Thus, a growing augmented dataset will be generated iteratively and be leveraged to force the substitute model to approximate the black-box model. In this article, we would utilize JAD for all the previous attack methods to investigate the black-box attacks and defense for the indoor localization systems.

F. Where to Launch Adversarial Attacks

Due to the nature of wireless communication systems, adversarial attacks can be launched from three places, i.e., the transmitting side, the channel side, and the receiving side. For indoor localization systems, e.g., WiFi-based systems, the attacking transmitters (APs) play a role of transmitter. APs are an essential part of the communication infrastructure, which are usually better secured with various cybersecurity technologies. It is usually more challenging to inject perturbations through the transmitter (i.e., AP) side. On the other hand, adversarial attacks from the channel side would be more feasible because of the open wireless channels. However, the channel effect should be considered when generating adversarial perturbations. For advLov, we assume that the adversarial perturbations are injected when the CSI tensors are generated, which usually happens at the user side. Compared to APs and from the channel, receive-side (user side) attacks are more feasible because it is easier to hack into a personal user device,

e.g., using phishing and a malware, to inject adversarial perturbations. Moreover, the channel effect is also eliminated when the perturbations are introduced from the user side.

G. Adversarial Training

To make the AdvLoc system resilient to adversarial attacks, its localization model implements adversarial training, which enhances the robustness of the neural network by training it with a mixture of adversarial and clean examples. The basic idea of adversarial training is to augment the original loss function with an adversarial term, so that it will be resistant to adversarial examples. Goodfellow *et al.* [17] demonstrated that the adversarial loss function as follows:

$$\widetilde{L}(\boldsymbol{\theta}, \boldsymbol{x}, y) = \gamma \cdot L(\boldsymbol{\theta}, \boldsymbol{x}, y) + (1-\gamma) \cdot L(\boldsymbol{\theta}, \boldsymbol{x} + \boldsymbol{\eta}, y) \quad (14)$$

was effective to make the neural network immune to FGSM attacks, where $\boldsymbol{\eta} = \epsilon \cdot \text{sign}(\nabla_{\boldsymbol{x}} L(\theta, \boldsymbol{x}, y))$. In (14), γ is a hyperparameter to adjust the relative importance of the loss terms of the original and adversarial examples, which is set to 0.5 in our implementation of AdvLoc.

In the next section, we will leverage adversarial training to study the effect of defense for indoor localization systems against adversarial attacks. The resulting localization model that is adversarially trained will be called by the corresponding attack method used in adversarial training. For example, if the localization model is trained with loss function (14) and the disturbance $\boldsymbol{\eta}$ in (14) is generated using FGSM (or MIM and PGD), the resulting adversarially trained model will be called FGSM-AT (or MIM-AT and PGD-AT, respectively).

IV. Experimental Study

A. Experiment Configuration

To evaluate the performance of AdvLoc under adversarial attacks in the online stage, we deploy the six types of adversarial attacks in both white-box and black-box scenarios. The AdvLoc system is implemented with Intel 5300 NIC in the 5.58-GHz band. Two laptops are configured as an access point and a mobile device, respectively. The distance between adjacent antennas is adjusted to 2.68 cm, which is a half of the wavelength. To inject adversarial attacks in the online stage, CleverHans [43] is leveraged to generate adversarial perturbations for each new CSI tensor. Furthermore, both the localization model trained in the offline stage and the adversarial example generation model used in the online stage are implemented with the TensorFlow framework on a NVIDIA RTX 2080 GPU.

For the sake of diversity, we examine the AdvLoc system in two representative indoor environments, i.e., a straight corridor and a computer laboratory.

1) *Straight Corridor:* First, the AdvLoc system is deployed in a straight corridor in Broun Hall in the Auburn University campus. This indoor testbed covers an area of 8×24 m^2, which includes the rooms on both sides of the corridor. As a typical indoor structure, the straight corridor is simple. Since there is no obstacles that result in complex scattering and reflection of WiFi signals, the line-of-sight (LOS) path is the dominant component in this environment. As is shown in Fig. 6, the red squares represent the training locations in the offline stage, while the green dots denote the testing location in the online stage. The single access point is placed at the right end of the corridor in Fig. 6. The distance between consecutive training locations is 1.8 m.

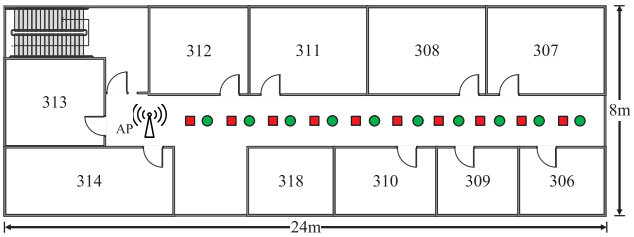

Fig. 6. Layout of the corridor scenario.

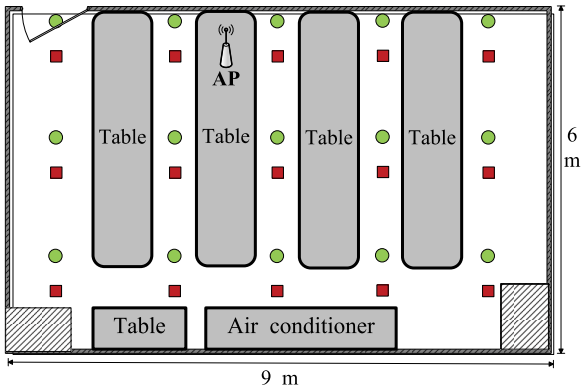

Fig. 7. Layout of the lab scenario.

2) *Computer Laboratory:* Next, we assess the AdvLoc performance in a computer laboratory, which is also located in Broun Hall. Compared with the corridor, the computer laboratory is a cluttered environment. Most of the LOS paths of WiFi signals are blocked by tables, chairs, and computer chassis. In this case, the access point is placed close to the north center of the laboratory so that it could cover the entire area. Fig. 7 depicts the selection of training positions (marked as red squares) and testing locations (marked as green dots). The distance between adjacent training locations is also 1.8 m.

To evaluate the system performance, we investigate the verification accuracy in the offline stage (see Section III-A). Because the training dataset and testing dataset are collected from identical locations, verification accuracy is defined as

$$\pi = \frac{\text{Number of correct predictions}}{\text{Total number of predictions}} \quad (15)$$

which indicates the capability of the DCNN model in solving the multiclass classification problem. In addition, we also evaluate the performance of the localization system by calculating the location estimation error $\mathcal{E}$, given by

$$\mathcal{E} = \left\| \widehat{T} - T \right\|_2 \quad (16)$$

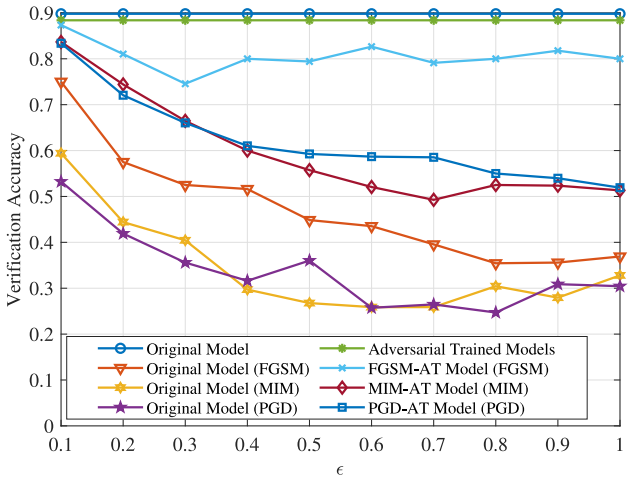

Fig. 8. Verification accuracy of the localization models in the lab environment.

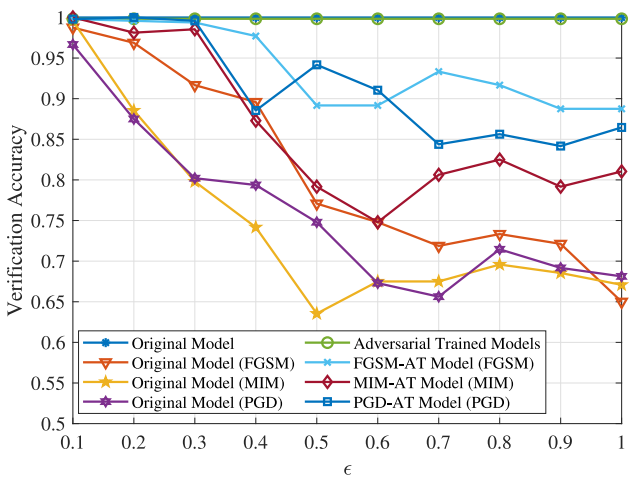

Fig. 9. Verification accuracy of the localization models in the corridor environment.

where $\widehat{T}$ is the estimated location given in (1) and T is the ground truth.

B. Verification Accuracy Under White-Box Attacks

We first confirm the verification accuracy of AdvLoc under white-box attacks in both indoor environments. For indoor localization systems, the training dataset and verification dataset are collected from identical positions. The verification accuracy gives us an unbiased assessment of how well our model fits the training data. Fig. 8 depicts the verification accuracy of the original localization model when not being attacked (called "Original Model"), and the verification accuracy of the original model when attacked by adversarial examples generated using FGSM, MIM, and PGD [called "Original Model (FGSM)," "Original Model (MIM)," and "Original Model (PGD)," respectively] in the lab setting. It shows that all the three attack methods successfully degrade the verification accuracy as ϵ is increased from 0.1 to 1. It is intuitive that a larger magnitude of perturbation causes a larger decrease in verification accuracy. Fig. 8 shows that the effects of PGD and MIM on the original model are comparable to each other, while FGSM is less effective than the two iterative methods.

Furthermore, adversarial training has been adopted in AdvLoc to combat adversarial attacks. Since the verification accuracy of adversarially trained localization models (i.e., FGSM-AT, MIM-AT, and PGD-AT) is very close when not being attacked, their average verification accuracy (called "Adversarial Trained Models" in Fig. 8) is very close to that of the original model. Thus, it is safe to say that adversarial training does not degrade the performance of the localization model when it is not attacked. With adversarial training, the verification accuracy of each model is enhanced remarkably when under adversarial attacks. For FGSM-AT, the attacked verification accuracy (the light blue line) remains above 0.74. When $\epsilon = 1$, the attacked verification accuracy of FGSM-AT reaches 0.8. Compared with the original model, FGSM-AT achieves an improvement of 0.12 in verification accuracy when $\epsilon = 0.1$, and an improvement of 0.44 when $\epsilon = 1$. In addition, the FGSM-AT curve is more stable for the whole range of ϵ, indicating that adversarial training is an effective defense against FGSM attacks. Similar to FGSM-AT, adversarial training also strengthens the robustness of the localization model against MIM and PGD attacks, even though the extent of the enhancements is not as notable as that of FGSM-AT. Nevertheless, the average improvements in verification accuracy achieved by MIM-AT and PGD-AT over the original model are still both greater than 0.25.

Fig. 9 presents the verification accuracy of the localization model in the corridor environment. As in Fig. 8, the localization model is attacked by three methods: 1) FGSM; 2) MIM; and 3) PGD. Since the corridor is a LOS dominant environment, the WiFi signals do not suffer from severe multipath effects. Therefore, the overall localization accuracy in the corridor is higher than 0.6, which is better than the lab case. With the increment of ϵ, all three attack methods contribute to degraded verification accuracy gradually, which is in accordance with the results shown in Fig. 8. In general, PGD and MIM are more effective than FGSM, even though FGSM decreases the verification accuracy to 0.65 when $\epsilon = 1$. Moreover, adversarial training is again an effective defense strategy for FGSM. For the adversarial examples generated by FGSM, the verification accuracy of FGSM-AT reaches 0.88 when ϵ is increased to 1. Both MIM-AT and PGD-AT also provide effective defense against the corresponding attacks, even though the extents of gains are not comparable to that of FGSM-AT.

To better evaluate the threat of adversarial attacks to indoor localization systems, three additional attack methods, STM, DeepFool, and CWL2, are also leveraged in the experiments. As shown in Fig. 10, the verification accuracy drops severely under these attacks. In the corridor case, all the three attacks reduce the verification accuracy to 0.13 or even worse. Similarly, the verification accuracy decreases from 0.9 to lower than 0.065 by all the attacks in the lab case. Thus, the optimization-based and the spatial transformation-based attack methods are also harmful to indoor localization system.

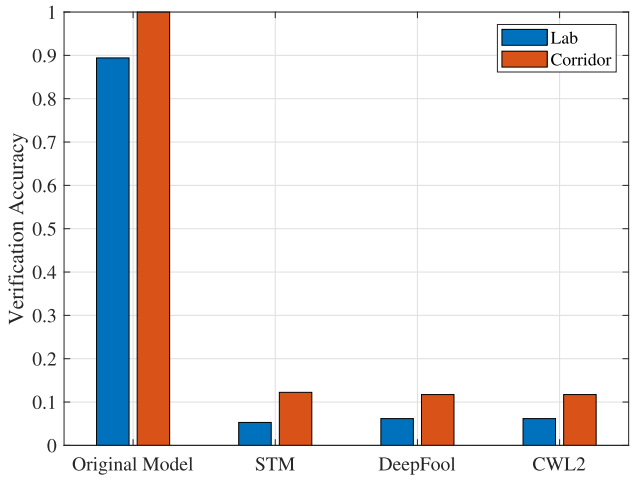

Fig. 10. Verification accuracy of the localization models attacked by CWL2, Deepfool, and STM.

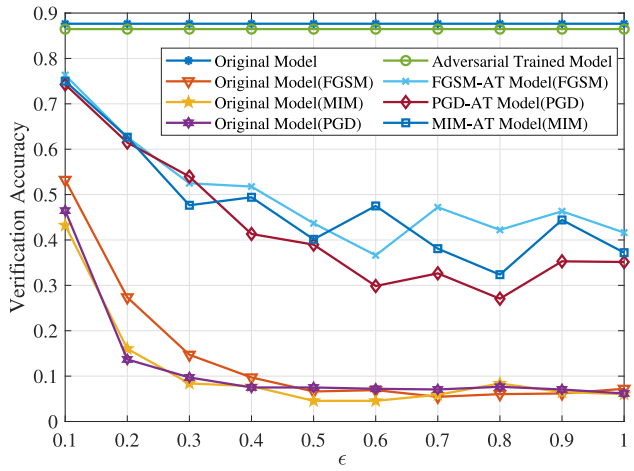

Fig. 11. Verification accuracy of the DCNN localization models in the lab environment.

In addition, according to [44], the localization model does not acquire transferability from the adversarial training, which means the model is still vulnerable to other types of adversarial attacks even if it is trained adversarially. Thus, further investigation is needed on adversarial training to take various types of attacks into account rather than a specific attack method.

In addition to the ResNet model, we also examine the effect of adversarial attacks and adversarial training on DCNN-based systems. The network used for comparison is composed of three convolutional layers. The kernel size for each layer is 8×8, 6×6, and 5×5, respectively, while 16 feature maps are generated in each convolutional layer. ReLu is used as the activation function following the convolutional layers. As in the ResNet model, cross-entropy loss is calculated for weight updates. Figs. 11 and 12 present the verification accuracy of the DCNN-based model under white-box attacks in the lab and corridor environments. As shown in Fig. 11, all the attack methods successfully degrade the verification accuracy in the lab environment. When ϵ reaches 0.4, all the verification

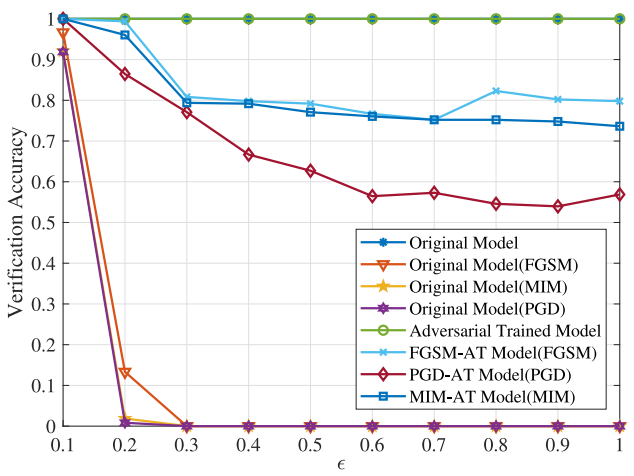

Fig. 12. Verification accuracy of the DCNN localization models in the corridor environment.

accuracies are reduced to lower than 0.1. Because of the simpler structure of DCNN, it is quite sensitive to adversarial attacks. With adversarial training, the performance of all the models is recovered to some extent. However, there is no clear performance difference among the models. The verification accuracy in the corridor case is presented in Fig. 12. Unlike the lab case, the corridor case is LOS-dominant. Thus, the verification accuracy of the original model remains at 1. However, the performance breaks down as ϵ goes up to 0.2. All the three attack methods reduce the verification accuracy to 0 when ϵ is 0.3. Adversarial training also achieves similar effectiveness in dealing with adversarial perturbations, even though the verification accuracy is not recovered to over 0.85. By examining the vanilla DCNN-based localization system, we notice that the robustness of such systems is determined by the complexity and depth of the network models. Shallow networks, such as the vanilla DCNN, are highly susceptible to the adversarial perturbations even with a low ϵ, which hampers us to examine the effect of perturbation magnitude to the system performance. Furthermore, [15] and [3] showed that a deeper DCNN usually achieves a better performance in fingerprinting-based indoor localization. Thus, we will investigate the effect of adversarial attacks to localization system using the ResNet model in the remainder of this section.

C. Location Error Under White-Box Attacks

Even though location estimation is treated as a multiclass classification problem in DCNN-based localization systems, a unique challenge in such localization systems is that an online input to the trained model usually does not belong to an existing class in the offline training dataset. For example, we label the CSI data collected from a position between point-A and point-B with label A in the testing dataset. The location prediction would be correct only if the localization system produces the same label. However, the testing position is usually between point-A and point-B. Obviously, it would be unfair to say that the location prediction is wrong when the prediction from the system is B. To address this issue, the output of the

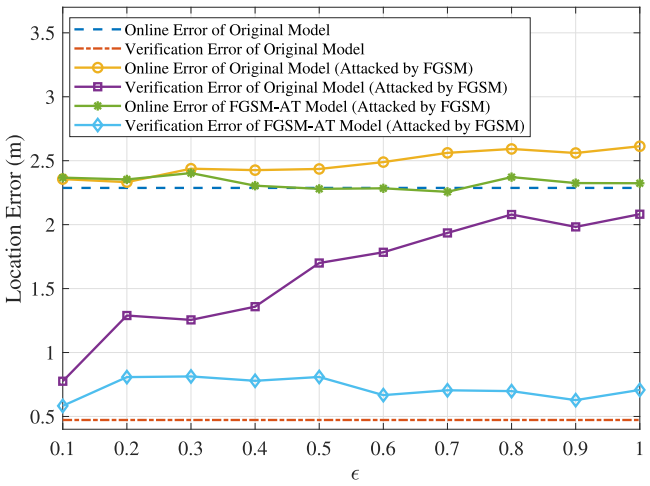

Fig. 13. Location error of the localization models when attacked by FGSM in the lab environment.

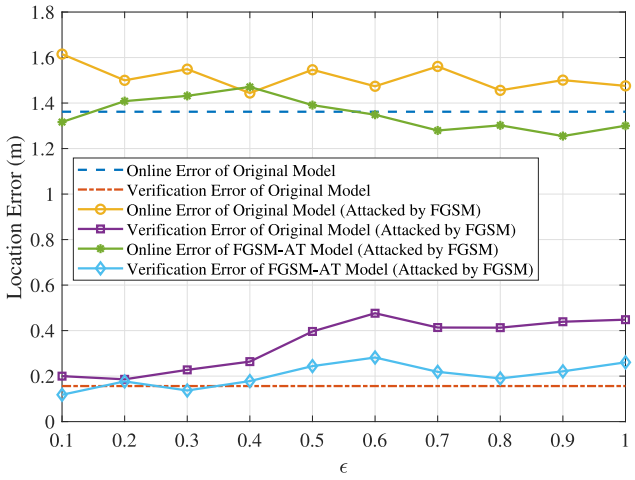

Fig. 14. Location error of the localization model attacked by FGSM in the corridor environment.

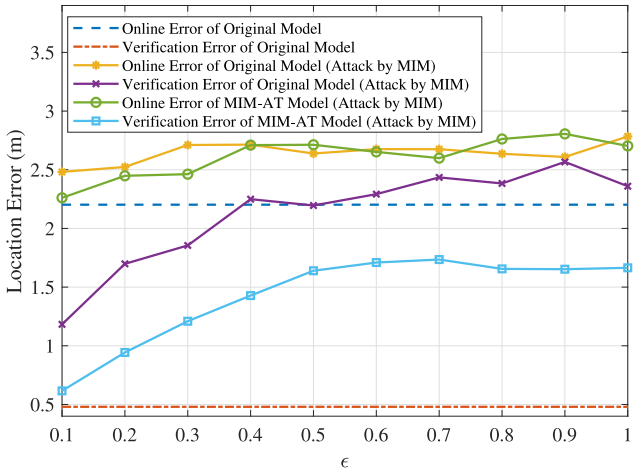

Fig. 15. Location error of the localization models attacked by MIM in the lab environment.

DCNN is usually used as similarity to calculate the estimated location using a Bayesian method (see Section III-A). Thus, test accuracy in the online stage may not precisely evaluate the performance of the system. In this article, location error is also utilized to measure the effect of adversarial attacks and adversarial training on the localization system.

First, we examine the performance of AdvLoc in the lab setting. Fig. 13 presents the location errors of FGSM-AT when attacked by FGSM, and of the original model when attacked by FGSM in verification and online testing. The blue dashed line is the online location error of the original model using clean inputs in online testing, while the verification location error for the same setup is denoted by a red dashed line. The errors are 2.28 and 0.47 m, respectively. It is obvious that the verification error rises with the increment of ϵ when the localization model is under attack, which is consistent with the verification accuracy shown in Fig. 8. For the online testing error, it also keeps going up along with the rise of ϵ. When $\epsilon = 0.1$, the adversarial examples increase the online testing error to 2.368 m. The highest online testing error, 2.613 m, occurs when $\epsilon = 1$. Furthermore, the performance of adversarial training is verified in Fig. 13 as well. Based on the FGSM-AT model, the upward trend of location errors in verification and online testing disappears. The online testing error of FGSM-AT stays around the error of the original model that leverages clean inputs. Even if $\epsilon = 1$, the increment of location error is only about 0.04 m, which is negligible in a lab environment. For the verification error, FGSM-AT guarantees that no verification error is higher than 0.81 m when the model is under attack. It is noteworthy that the verification error declines from 2.08 to 0.70 m, when ϵ is fixed at 1, once adversarial training is leveraged in the localization model.

For the corridor case, the location errors of FGSM-AT attacked by FGSM and the original localization model attacked by FGSM are shown in Fig. 14. Compared with Fig. 13, the upward trend of errors in the corridor case is not as obvious as that of in the lab case. For the online testing error when the original localization model is attacked by FGSM, the error does not increase with ϵ, even though FGSM deteriorates the localization error from 1.36 to 1.52 m on average. The verification location errors reveal a similar behavior. The maximum of the verification error increment is only 0.32 m when the original localization model is attacked by FGSM with $\epsilon = 0.6$. Adversarial training is still an effective defense strategy against FGSM in the corridor case. The green line in Fig. 14 represents the online testing errors when FGSM-AT is attacked by FGSM. As we can see, the errors of FGSM-AT are obviously lower than that of the original model attacked by FGSM. The average error of FGSM-AT is 1.36 m, which is closed to the average error of the original model with clean inputs, i.e., 1.3504 m.

The effect of MIM and the corresponding adversarial training on location error is depicted in Figs. 15 and 16, respectively. The verification error of the original model grows significantly when attacked by MIM, which is consistent with the results presented in Fig. 8. Furthermore, MIM causes much larger errors than FGSM. In Fig. 15, the verification location error reaches 2.36 m when attacked by adversarial examples generated by MIM with $\epsilon = 1$, which is much higher than

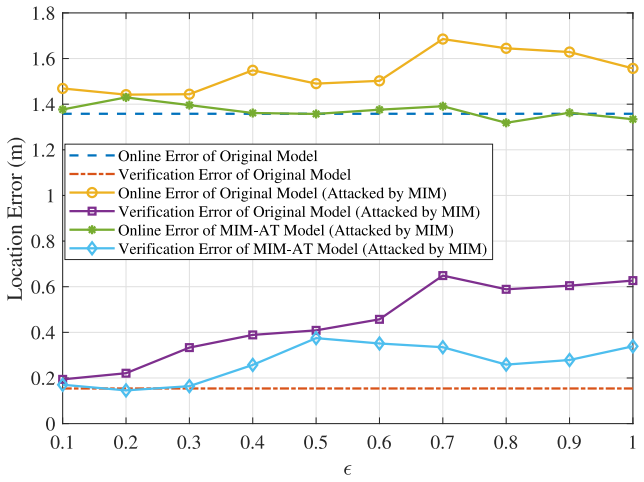

Fig. 16. Location error of the localization models attacked by MIM in the corridor environment.

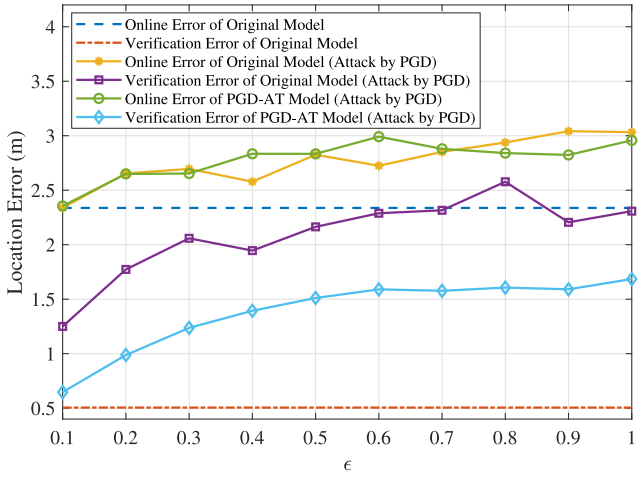

Fig. 17. Location error of the localization models attacked by PGD in the lab environment.

that of FGSM. A similar phenomenon is observed in the corridor case. The verification error reaches 0.62 m when $\epsilon = 1$, whereas the verification error is only 0.44 m when $\epsilon = 1$ with FGSM. MIM is thus a stronger attack method than FGSM. Additionally, Fig. 15 shows that MIM-AT does not effectively eliminate the effect of MIM. However, adversarial training successfully removes the rising trend of the online testing error in the corridor case with MIM-AT. According to Fig. 16, the MIM-AT model has a commensurable performance as the unattacked original model.

Figs. 17 and 18 present the location errors of PGD related experiments. First, the location errors in the lab case are given in Fig. 17. Similar to MIM, PGD, as an iterative attack method, degrades the verification precision remarkably. The location errors climb up with the increase of ϵ when the localization model is attacked by PGD. Nevertheless, the online testing error is not improved by adversarial training in the lab case, which is similar to the MIM related experiments. In the corridor case, adversarial training effectively enhances the online testing precision and verification precision.

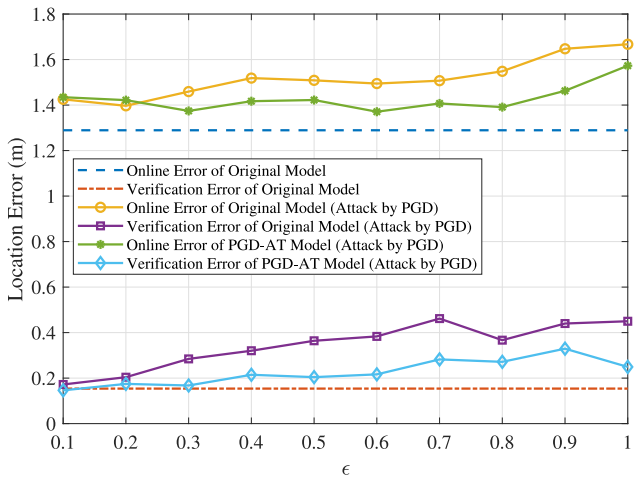

Fig. 18. Location error of the localization models attacked by PGD in the corridor environment.

It can be seen from Figs. 14, 16, and 18 that adversarial training could always reduce both online testing errors and verification errors in the corridor case. Moreover, the adversarial attacks, such as FGSM, MIM, and PGD, could not degrade much the performance of the localization model in the corridor environment. This is because the multipath effect is not as strong in the corridor case, and it is relatively easier for the DCNN model to distinguish the WiFi signals from different locations. Such "easy-to-distinguish" signals contribute to the robustness of the model, especially when the size of the training data set is not large. As a result, the effectiveness of adversarial attacks is constrained in the corridor case, and adversarial training is also more effective. In the lab case, the received WiFi signal is a superposition of the signals from multiple paths. The localization model becomes more gullible in facing with such noisy signals. Moreover, considering the fact that the class of the new CSI tensors in the online stage usually does not belong to any class used in offline training, such noisy signals make adversarial training struggle in the online testing. Hence, even though adversarial training achieves an acceptable performance in defending FGSM attacks, it is not as effective for stronger attacks, such as MIM and PGD, in the online stage.

We also examine the effect of optimization-based and spatial transformation-based attack methods, including CWL2, DeepFool, and STM, and their location errors in the lab and corridor environments are presented in Fig. 19. We find the optimization-based attacks, i.e., CWL2 and DeepFool, cause higher location errors in verification and online testing. Compared with FGSM, MIM, and PGD, DeepFool poses the strongest threat to localization systems in the lab case. Moreover, both CWL2 and DeepFool increase the testing errors in the corridor case to over 2 m, which is much higher than that caused by the traditional one-step or iterative attacks.

D. Location Error Under Black-Box Attacks

The white-box attacks rely on knowledge of the target DCNN model, which may not be available to adversaries in

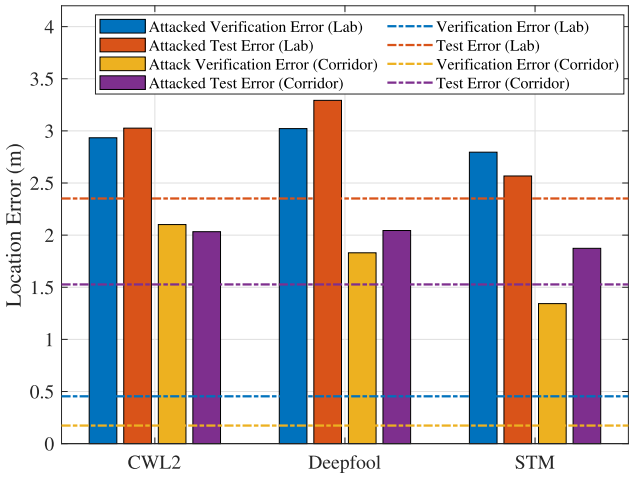

Fig. 19. Location error of the localization models attacked by CWL2, Deepfool, and STM in the white-box scenario.

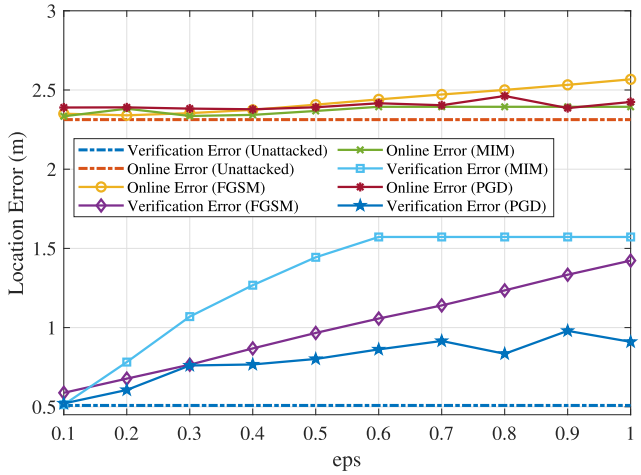

Fig. 20. Effect of black-box attacks on the location error of the localization models in the lab environment.

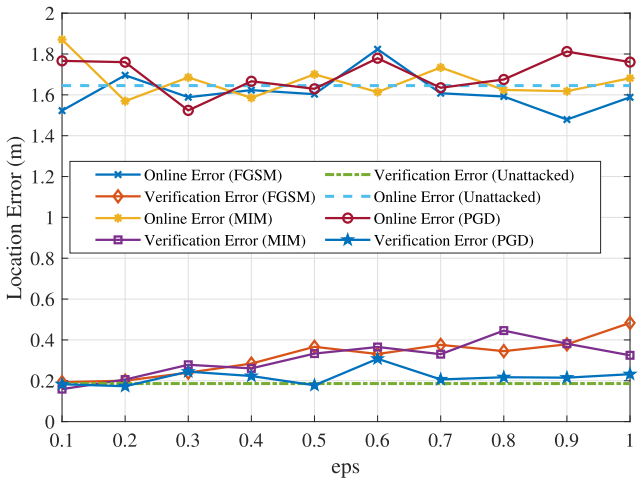

Fig. 21. Effect of black-box attacks on the location error of the localization models in the corridor environment.

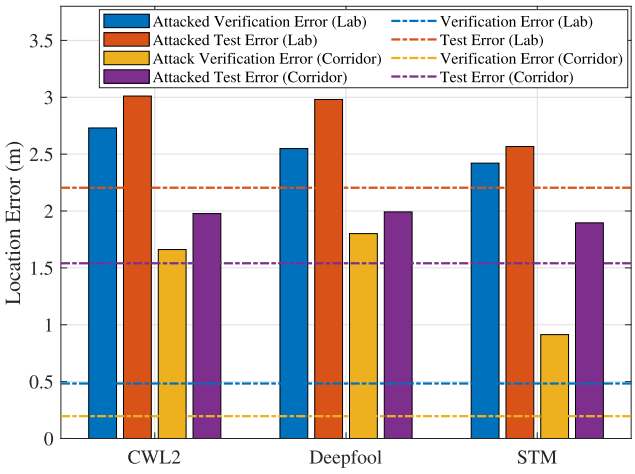

Fig. 22. Location error of the localization models attacked by CWL2, DeepFool, and STM in the black-box scenario.

many cases. Therefore, black-box attacks would be more practical in the real world. To investigate the threat of black-box attacks and evaluate the corresponding defense strategies, we implement all the previously mentioned attack methods based on the black-box attack approach.

First, FGSM, MIM, and PGD are deployed with the black-box approach to examine the their impacts in the lab case. As shown in Fig. 20, all the three attack methods exhibit outstanding performance in increasing the verification location error. However, the online testing errors are not affected by the attacks severely. The maximum increase in location error is only about 0.25 m under FGSM generated perturbation with $\epsilon = 1$. Compared with the white-box attacks, the degradation of online testing error is negligible in Fig. 20.

Fig. 21 describes the performance of the black-box attacks in the corridor case. Because of the robustness of the localization model, the online testing errors are not influenced much by the black-box attacks. For the verification error, the maximum increment is only about 0.3 m, even though a slightly upward trend is observed in Fig. 21. Thus, it is safe to say that our localization model for the corridor case is robust enough against black-box attacks. In other words, the adversarial examples generated by the substitute model (i.e., ResNet-50) for black-box attack fail to mislead the original DCNN model.

We also leverage the optimization-based and spacial transformation-based attack methods to evaluate the system under black-box attacks. Comparing Fig. 19 with Fig. 22, we notice that each result in Fig. 22 is lower than the corresponding result in Fig. 19. CWL2, DeepFool, and STM could not achieve similar performance when used for the black-box attack. The difference in the knowledge between the black-box model (i.e., ResNet-18) and the substitute model (i.e., ResNet-50) limits the performance of the attacks.

V. Conclusion

In this article, we presented AdvLoc, an adversarial deep learning for indoor localization system using CSI tensors, which is resilient against the typical first-order adversarial attacks. With the proposed AdvLoc system, we analyzed

the effect of six types common adversarial attacks in both white-box attack and black-box attack scenarios. The extensive experimental study exposed the threat of the adversarial attacks to indoor localization systems and validated the superior performance of the proposed AdvLoc system in defending against first-order adversarial attacks.

REFERENCES

[1] X. Wang, L. Gao, and S. Mao, "CSI phase fingerprinting for indoor localization with a deep learning approach," *IEEE Internet Things J.*, vol. 3, no. 6, pp. 1113–1123, Dec. 2016.

[2] X. Wang, L. Gao, S. Mao, and S. Pandey, "CSI-based fingerprinting for indoor localization: A deep learning approach," *IEEE Trans. Veh. Technol.*, vol. 66, no. 1, pp. 763–776, Jan. 2017.

[3] W. Wang, X. Wang, and S. Mao, "Deep convolutional neural networks for indoor localization with CSI images," *IEEE Trans. Netw. Sci. Eng.*, vol. 7, no. 1, pp. 316–327, Jan.–Mar. 2020.

[4] C. Yang, X. Wang, and S. Mao, "SparseTag: High-precision backscatter indoor localization with sparse RFID tag arrays," in *Proc. IEEE SECON*, Boston, MA, USA, Jun. 2019, pp. 1–9.

[5] J. Zhang *et al.*, "RFHUI: An RFID based human-unmanned aerial vehicle interaction system in an indoor environment," *Elsevier/KeAi Digit. Commun. Netw. J.*, vol. 6, no. 1, pp. 14–22, Feb. 2020.

[6] J. Zhang, Y. Lyu, J. Patton, S. C. G. Periaswamy, and T. Roppel, "BFVP: A probabilistic UHF RFID tag localization algorithm using Bayesian filter and a variable power RFID model," *IEEE Trans. Ind. Electron.*, vol. 65, no. 10, pp. 8250–8259, Oct. 2018.

[7] M. Kotaru, K. Joshi, D. Bharadia, and S. Katti, "SpotFi: Decimeter level localization using WiFi," in *Proc. ACM SIGCOMM*, London, U.K., Aug. 2015, pp. 269–282.

[8] D. Vasisht, S. Kumar, and D. Katabi, "Decimeter-level localization with a single WiFi access point," in *Proc. ACM NSDI*, Boston, MA, USA, Mar. 2016, pp. 165–178.

[9] X. Wang, L. Gao, and S. Mao, "BiLoc: Bi-modal deep learning for indoor localization with commodity 5GHz WiFi," *IEEE Access*, vol. 5, pp. 4209–4220, 2017.

[10] H. Chen, Y. Zhang, W. Li, X. Tao, and P. Zhang, "ConFi: Convolutional neural networks based indoor Wi-Fi localization using channel state information," *IEEE Access*, vol. 5, pp. 18066–18074, 2017.

[11] A. Mittal, S. Tiku, and S. Pasricha, "Adapting convolutional neural networks for indoor localization with smart mobile devices," in *Proc. ACM Great Lakes Symp. VLSI*, Chicago, IL, USA, May 2018, pp. 117–122.

[12] T. Zhang and Y. Man, "The enhancement of WiFi fingerprint positioning using convolutional neural network," in *Proc. Int. Conf. Comput. Commun. Netw. Technol.*, Wuzhen, China, Jun. 2018, pp. 479–483.

[13] M. Ibrahim, M. Torki, and M. ElNainay, "CNN based indoor localization using RSS time-series," in *Proc. IEEE Symp. Comput. Commun.*, Natal, Brazil, Jun. 2018, pp. 1044–1049.

[14] X. Wang, X. Wang, and S. Mao, "ResLoc: Deep residual sharing learning for indoor localization with CSI tensors," in *Proc. IEEE PIMRC*, Montreal, QC, Canada, Oct. 2017, pp. 1–6.

[15] X. Wang, X. Wang, and S. Mao, "Indoor fingerprinting with bimodal CSI tensors: A deep residual sharing learning approach," *IEEE Internet Things J.*, vol. 8, no. 6, pp. 4498–4513, Mar. 2021.

[16] C. Szegedy *et al.*, "Intriguing properties of neural networks," Dec. 2013, *arXiv:1312.6199*.

[17] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," Dec. 2014, *arXiv:1412.6572*.

[18] C. Szegedy *et al.*, "Going deeper with convolutions," in *Proc. IEEE CVPR*, Boston, MA, USA, Jun. 2015, pp. 1–9.

[19] S. G. Finlayson, J. D. Bowers, J. Ito, J. L. Zittrain, A. L. Beam, and I. S. Kohane, "Adversarial attacks on medical machine learning," *Science*, vol. 363, no. 6433, pp. 1287–1289, Mar. 2019.

[20] S. G. Finlayson, H. W. Chung, I. S. Kohane, and A. L. Beam, "Adversarial attacks against medical deep learning systems," Apr. 2018, *arXiv:1804.05296*.

[21] S. Thys, W. Van Ranst, and T. Goedemé, "Fooling automated surveillance cameras: Adversarial patches to attack person detection," in *Proc. IEEE CVPR Workshops*, Long Beach, CA, USA, Jun. 2019, pp. 49–55.

[22] M. Sharif, S. Bhagavatula, L. Bauer, and M. K. Reiter, "Accessorize to a crime: Real and stealthy attacks on state-of-the-art face recognition," in *Proc. ACM SIGSAC Conf. Comput. Commun. Security*, Vienna, Austria, Oct. 2016, pp. 1528–1540.

[23] S. Yan, Y. Xiong, and D. Lin, "Spatial temporal graph convolutional networks for skeleton-based action recognition," Jan. 2018, *arXiv:1801.07455*.

[24] X. Gao, W. Hu, J. Tang, P. Pan, J. Liu, and Z. Guo, "Generalized graph convolutional networks for skeleton-based action recognition," in *Proc. 9th Int. Conf. Comput. Pattern Recognit.*, Xiamen, China, Oct. 2020, pp. 43–49.

[25] J. Liu, N. Akhtar, and A. Mian, "Adversarial attack on skeleton-based human action recognition," Sep. 2019, *arXiv:1909.06500*.

[26] B. Liang, H. Li, M. Su, P. Bian, X. Li, and W. Shi, "Deep text classification can be fooled," Apr. 2017, *arXiv:1704.08006*.

[27] N. Papernot, P. McDaniel, A. Swami, and R. Harang, "Crafting adversarial input sequences for recurrent neural networks," in *Proc. IEEE MILCOM*, Baltimore, MD, USA, Nov. 2016, pp. 49–54.

[28] N. Tang, S. Mao, and R. M. Nelms, "Adversarial attacks to solar power forecast," in *Proc. IEEE GLOBECOM*, Madrid, Spain, Dec. 2021, pp. 1–6.

[29] Y. Sun, M. Peng, Y. Zhou, Y. Huang, and S. Mao, "Application of machine learning in wireless networks: Key techniques and open issues," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 4, pp. 3072–3108, 4th Quart., 2019.

[30] M. Sadeghi and E. G. Larsson, "Adversarial attacks on deep-learning based radio signal classification," *IEEE Wireless Commun. Lett.*, vol. 8, no. 1, pp. 213–216, Feb. 2019.

[31] Y. Lin, H. Zhao, Y. Tu, S. Mao, and Z. Dou, "Threats of adversarial attacks in DNN-based modulation recognition," in *Proc. IEEE INFOCOM*, Toronto, ON, Canada, Jul. 2020, pp. 2469–2478.

[32] Z. Bao, Y. Lin, S. Zhang, Z. Li, and S. Mao, "Threat of adversarial attacks on DL-based IoT device identification," *IEEE Internet Things J.*, early access, Oct. 14, 2021, doi: 10.1109/JIOT.2021.3120197.

[33] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE CVPR*, Las Vegas, NV, USA, Jun. 2016, pp. 770–778.

[34] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Commun. ACM*, vol. 60, no. 6, pp. 84–90, May 2017.

[35] T. Miyato, A. M. Dai, and I. Goodfellow, "Adversarial training methods for semi-supervised text classification," May 2016, *arXiv:1605.07725*.

[36] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," Jun. 2017, *arXiv:1706.06083*.

[37] Y. Dong *et al.*, "Boosting adversarial attacks with momentum," in *Proc. IEEE/CVF CVPR*, Salt Lake City, UT, USA, Jun. 2018, pp. 9185–9193.

[38] S.-M. Moosavi-Dezfooli, A. Fawzi, and P. Frossard, "DeepFool: A simple and accurate method to fool deep neural networks," in *Proc. IEEE CVPR*, Las Vegas, NV, USA, Jun./Jul. 2016, pp. 2574–2582.

[39] N. Papernot, P. McDaniel, X. Wu, S. Jha, and A. Swami, "Distillation as a defense to adversarial perturbations against deep neural networks," in *Proc. IEEE Symp. Security Privacy*, San Jose, CA, USA, May 2016, pp. 582–597.

[40] N. Carlini and D. Wagner, "Towards evaluating the robustness of neural networks," in *Proc. IEEE Symp. Security Privacy*, San Jose, CA, USA, May 2017, pp. 39–57.

[41] L. Engstrom, B. Tran, D. Tsipras, L. Schmidt, and A. Madry, "Exploring the landscape of spatial robustness," Dec. 2017, *arXiv:1712.02779*.

[42] N. Papernot, P. McDaniel, I. Goodfellow, S. Jha, Z. B. Celik, and A. Swami, "Practical black-box attacks against machine learning," in *Proc. ACM Asia Conf. Comput. Commun. Security*, Abu Dhabi, UAE, Apr. 2017, pp. 506–519.

[43] N. Papernot *et al.*, "Technical report on the CleverHans v2.1.0 adversarial examples library," Jun. 2018, *arXiv:1610.00768*.

[44] A. Kurakin, I. Goodfellow, and S. Bengio, "Adversarial machine learning at scale," Nov. 2016, *arXiv:1611.01236*.