Threat of Adversarial Attacks on DL-Based IoT Device Identification

Zhida Bao, *Graduate Student Member, IEEE*, Yun Lin, *Member, IEEE*, Sicheng Zhang, *Graduate Student Member, IEEE*, Zixin Li, *Graduate Student Member, IEEE*, and Shiwen Mao, *Fellow, IEEE*

***Abstract*—With the rapid development of the information technology, the number of devices in the Internet of Things (IoT) is increasing explosively, which makes device identification a great challenge. Deep neural networks (DNNs) have been used for device identification in IoT due to their superior learning ability. However, DNNs are susceptible to adversarial attacks, which can greatly degrade the accuracy of deep learning (DL) models for device identification. The adversarial attack is one of the fundamental security concerns for DNNs, and it is of great importance to study the generation of adversarial examples and to examine the attack effects for the design of robust DNN-based device identification schemes. In this article, we examine the effects of nontargeted and targeted adversarial attacks on convolutional neural network (CNN)-based device identification and propose combined evaluation indicators of logits to enrich the evaluation criteria. Our experimental results demonstrate that the identification accuracy degrades with the increase of the perturbation level and iteration step size, and the proposed combined evaluation indicators are effective to show the individual device signal differences. The insights from this study will be useful for the design of robust DL-based IoT systems.**

***Index Terms*—Adversarial attacks, convolutional neural network (CNN), deep learning (DL), device identification, Internet of Thing (IoT) security.**

I. Introduction

THE INFORMATION technology and communication networks are developing rapidly in the upcoming 6G era [1], [2], which provides a technical foundation for the interaction of a large number of devices and the generation of massive data in the Internet of Things (IoT). Modern information technology has greatly promoted the explosion of the IoT, enabling it to support numerous networked devices more efficiently and conduct fast transmission [3]. However, in the increasingly large-scale modern IoT, the numbers of devices and signals have exploded, and distinguishing each unique device and signal has become a challenge [4]. Therefore, the device identification technology and its security strategy are growingly becoming a hot topic in the field [5], which aims to provide a safer and smarter decision-making method for the IoT.

Generally, automatic modulation analysis (AMC) plays an important role in signal recognition, which can effectively ensure the security of the physical layer [2], [6]–[8]. The radio frequency (RF) fingerprint identification method based on machine learning has been shown to address the authentication issues effectively [9], and promote the development of wireless device security systems in the IoT. However, traditional identification technologies may have limited practical impacts in the continuously more complicated IoT [10]. Traditional methods are faced with challenges, such as massive training samples [11], noise influence, and high-dimensional feature learning [12]. In order to solve the above problems, researchers try to apply deep learning (DL) in the IoT [13] and some DL-related methods have been proposed for the works of signal recognition [14], [15]. DL has many advantages in device identification, such as excellent automatic analysis capabilities [16], individual feature extraction without complicated processes [17], a fully trained model with a small number of training samples, and the ability to characterize individual signals with fewer fingerprint feature dimensions [18].

However, some security issues arise while DL provides a powerful technical method. Recently, the IoT faces much more risks than before with DL models widely used, posing a great threat to the training process, testing process, and privacy security of the models. Data poisoning attacks [19] and backdoor attacks [20] will destroy the integrity of IoT model training. Adversarial attacks [21]–[25] threaten the complete model testing by means of algorithm flaws. Privacy inference attacks [26] causes data leakage based on gradient updates. The adversarial attack is a malicious behavior that uses adversarial examples to deceive the models, which seriously threatens the stability of the IoT. Szegedy *et al.* [27] first showed that introducing a carefully designed weak perturbation could trick the DL classifier to produce completely wrong prediction results in the image classification task. Such weak perturbation is called adversarial examples, which refer to the input example created by deliberately adding subtle perturbations to the data sets. The perturbed input data fool the machine learning

Manuscript received June 12, 2021; revised August 30, 2021; accepted October 2, 2021. Date of publication October 14, 2021; date of current version May 23, 2022. This work was supported in part by the National Natural Science Foundation of China under Grant 61771154; in part by the Fundamental Research Funds for the Central Universities under Grant 3072021CF0815; and in part by the Key Laboratory of Advanced Marine Communication and Information Technology, Ministry of Industry and Information Technology, Harbin Engineering University, Harbin, China. *(Corresponding author: Yun Lin.)*

Zhida Bao, Yun Lin, Sicheng Zhang, and Zixin Li are with the College of Information and Communication Engineering, Harbin Engineering University, Harbin 150001, China (e-mail: linyun@hrbeu.edu.cn).

Shiwen Mao is with the Department of Electrical and Computer Engineering, Auburn University, Auburn, AL 36849 USA (e-mail: smao@ieee.org).

Digital Object Identifier 10.1109/JIOT.2021.3120197

2327-4662 © 2021 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See https://www.ieee.org/publications/rights/index.html for more information.

Authorized licensed use limited to: Anglia Ruskin University (a.r.u.). Downloaded on February 19,2024 at 17:42:57 UTC from IEEE Xplore. Restrictions apply.

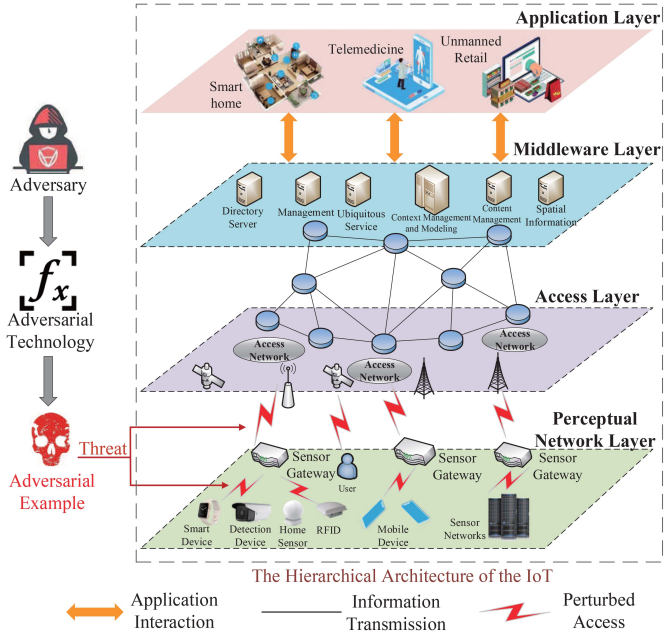


Fig. 1. Schematic diagram of the IoT under the adversarial attacks. Adversary will destroy the reliable transmission of IoT by carefully designed perturbations, causing serious identification security risks.

model to produce error output with high confidence. Presently, adversarial examples have been demonstrated to seriously degraded the reliability of DL models in computer vision [28], speech recognition [29], and text classification [30].

It is worth noting that Sadeghi and Larsson [31] applied adversarial examples to modulation recognition for the first time, and found that the DL model in signal recognition is also vulnerable to adversarial attacks. Since then, adversarial attacks in the signal domain have aroused widespread research interests. Exploratory attacks are applied to the IoT, causing serious losses in wireless communication throughput [32]. RF signal adversarial examples can cause misclassification with small waveform changes [33], [34]. In [35]–[37], the over-the-air (OTA) attacks have been considered with channel effect, exposing the vulnerability of the receiver against wireless adversarial attack. In addition, the targeted attacks in the RF signals have also been verified in [38], demonstrating the attacker can make the machine learning model output the expected category. According to [39], most researches focus on digital attacks, that is, adversarial attacks are lunched directly at the receiver, which can damage the performance of device sensors in the IoT. As shown in Fig. 1, the adversarial examples can cause serious disasters to the IoT, such as damaging the equipment authentication, device identification and reliable transmission, and harming the communication security and privacy protection in the IoT. Although methods for safe device authentication in the IoT has been proposed [40], adversarial attacks can often break the common security system.

In order to make a credible evaluation of the signal adversarial examples, some evaluation standards are proposed. In previous work, Lin *et al.* [41], [42] evaluated the imperceptibility of adversarial examples by recovering and comparing the waveforms before and after the signal was perturbed. The fitting difference of the waveforms was also used to measure the detectability of an adversarial attack [43]. The statistical test analysis before and after the perturbation was depended on the peak-to-average-power ratio of the data points in [44]. Bit error rate is introduced for the evaluation of adversarial examples in [45]. However, the above works utilized modulated signal data sets rather than actual individual device signals. Moreover, these evaluation indicators are only used for nontargeted attacks, and cannot measure the targeted attack performance against the models. The objective analysis of the output given by the key network layers can also provide an important reference for developing the security strategy of the IoT.

In this article, we verify the threat of adversarial attacks on individual device identification, and explore the factors that affect the effectiveness of nontargeted attacks and targeted attacks. Due to the diversity of the generation methods of adversarial examples, it is universally significant for nontargeted attacks and targeted attacks to explore the maximum perturbation limit range and iterative step length. In addition, we train the complex neural network [46] by generated data sets to recognize the individual devices. Since the complex neural network is a DL technology that has been proven effective in signal processing, which has achieved good results in tasks such as AMC [47].

Then adversarial examples of individual device signals are generated to evaluate this model performance.

In particular, we propose combined evaluation indicators of logits to evaluate the performance of targeted attacks. The logits layer retains the models initial judgment on the input samples, which is conducive to observing the inducing performance of the targeted attack to the model, and detecting high-quality robust signals. With the combined evaluation indicators of logits, we can evaluate the robustness of the original signals and the performance of the adversarial attack algorithms from the perspectives of the source and target logits difference.

The main contributions of this article are summarized as follows.

1) We evaluate nontargeted attacks with adversarial examples against complex neural networks under the white-box attack scenario on convolutional neural network (CNN)-based device identification, which provides a useful reference for the design of robust DL models.
2) We analyze targeted attacks generated by representative algorithms on CNN-based device identification, where the adversarial examples will fool the CNN-based classifier toward a specific output result. The targeted attack experiment has strong practical significance and provides useful insights on the design of robust CNN models.
3) We propose to use combined evaluation indicators of logits in this article, which helps to reveal the real differences of different examples in the model prediction output and to promote the comprehensiveness of classification model evaluation.

The remainder of this article is organized as follows. The preliminaries on complex neural networks and adversarial examples are summarized in Section II. The generation of

Authorized licensed use limited to: Anglia Ruskin University (a.r.u.). Downloaded on February 19,2024 at 17:42:57 UTC from IEEE Xplore. Restrictions apply.

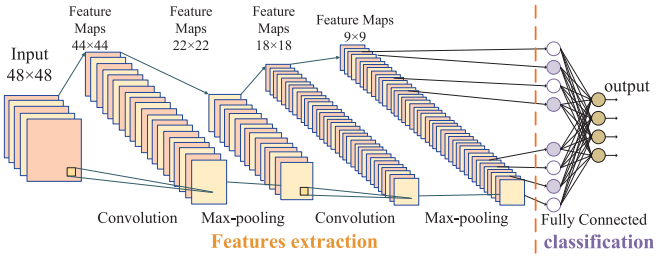


Fig. 2. Schematic diagram of the structure of a CNN classifier. The convolutional layers and the pooling layers extract features of input samples. The fully connected layers integrate the extracted features and output the classification results.

individual signal data sets, adversarial attack algorithms, and the proposed combined evaluation indicators of logits are provided in Section III. We present experimental results and discussions in Section IV. The conclusions and future work are presented in Section V.

II. Preliminaries

A. Convolutional Neural Network

CNN is a representative category of deep neural network (DNN) and have been widely used in many areas [48]. CNN usually consists of an input layer, multiple convolutional layers, pooling layers, multiple fully connected layers, and output layers [49]. The structure of a CNN classifier is shown in Fig. 2. An original example is fed into the network and enters the convolution layer. The convolution layer contains several convolution kernel functions, which acts on the input example to extract local features of the example. The convolution layer is followed by the sampling layer, whose main function is to subsample the feature map in the convolution layer to reduce the resolution and thereby reduce the complexity of the model. Generally, a convolutional layer combined with a subsampling layer consists of a feature extraction process. The last layer of CNN is composed of multiple fully connected layers. After feature mapping, the final classification result is obtained. Each neuron cell of the output feature surface in the convolutional layer is locally connected to its input, and the weighted summation is added to the local input through the corresponding connection weight, while the offset value is obtained to adjust the input value of the neuron [50].

In the convolutional layer, the feature map of the previous layer is convoluted with a learnable convolution kernel, and the obtained result is fed into an activation function. The output of the convolutional layer is composed of a new feature map with a different dimension. Each output feature map can be obtained by the convolution of a combination of multiple feature maps in the previous layer [51], and the operation of the convolutional layer is as follows:

$$X_j^l = f\left(\sum_{i\in M_{l-1}} X_i^{l-1} * K_{ij}^l + b_j^l\right) \tag{1}$$

where X_j^l represents the jth feature map of the lth layer, K_{ij}^l is the convolution kernel function, $f(\cdot)$ is the activation function, b_j^l is the bias of the jth feature map in the lth layer, and M_{l-1} represents the selected input feature map set.

B. Complex Neural Network

We will use the generated data sets to train a complex neural network model [46], and then the trained model for device identification. Since the individual device signals are usually in the complex form, it is natural and more effective to apply a complex neural network model to process these data sets. The design of the proposed model is discussed in the following.

1) Convolutional Layer Construction: The complex-numbered convolution operation can be realized by real-numbered convolutions [52]. As an example, let's consider the convolution of a complex vector $\vec{s} = \vec{u} + j\vec{v}$ and a complex matrix $\mathbf{L} = \mathbf{A} + j\mathbf{B}$, where $\mathbf{A}$ and $\mathbf{B}$ are real-numbered matrices, and $\vec{u}$ and $\vec{v}$ are real-numbered vectors. The convolution process of $\mathbf{L}$ and $\vec{s}$ is as follows:

$$\mathbf{L} \otimes \vec{s} = (\mathbf{A} * \vec{u} - \mathbf{B} * \vec{v}) + j(\mathbf{B} * \vec{u} + \mathbf{A} * \vec{v}). \tag{2}$$

The convolution of complex numbers can be divided into real convolution and imaginary convolution of the two vectors, and the convolution between the real part and the imaginary part is performed separately.

According to the theory of converting complex convolution into real convolution, the convolution between a complex feature map $\mathbf{M}$ and a complex convolution kernel $\mathbf{K}$ can be expressed as follows:

$$\mathbf{M} \otimes \mathbf{K} = (\mathbf{M}_R\mathbf{K}_R - \mathbf{M}_I\mathbf{K}_I) + j(\mathbf{M}_R\mathbf{K}_I + \mathbf{M}_I\mathbf{K}_R) \tag{3}$$

where R represents the real part and I represents the imaginary part. Fig. 3 shows the convolution process of the complex feature map $\mathbf{M}$ and the complex convolution kernel $\mathbf{K}$. In Fig. 3, the green (light blue) blocks represent the real (imaginary) parts of the convolution kernel, and the blue (brown) blocks represents the real (imaginary) parts of the complex feature map. After the convolution operation, the output feature map is still divided into real and imaginary parts (blue and brown, respectively).

2) Batch Normalization: Batch normalization helps to speed up the learning of the neural networks and normalize a set of complex numbers into a standard normal complex distribution. Since the standard method of batch normalization is only applicable to real-valued data, it is necessary to derive the following complex batch normalization method.

Define $\tilde{a}$ as the centered and scaled value of a complex data a, given by

$$\tilde{a} = \frac{a - \mathbb{E}[a]}{\sqrt{1/\mathbf{D}}} \tag{4}$$

where $\mathbf{D}$ is a 2×2 covariance matrix given by

$$\mathbf{D} = \begin{pmatrix} D_{rr} & D_{ri} \\ D_{ir} & D_{ii} \end{pmatrix}. \tag{5}$$

$(\mathbf{D})^{-1/2}$ is determined by the positive definite property of $\mathbf{D}$, which can be computed by adding an arbitrary matrix $\mathbf{X}$ to the Tikhonov regularization of $\mathbf{D}$. In (5), D_{rr}, D_{ri}, D_{ir}, and D_{ii} can be obtained by the covariance operation accordingly.

After (4), the resulting $\tilde{a}$ has a standard complex distribution with mean $t = 0$, covariance $\phi = 1$, and pseudo covariance $P_c = 0$. The normalization of plural batches, $BN(\cdot)$, is given by

$$BN(\tilde{a}) = \phi \cdot \tilde{a} + \eta \tag{6}$$

Authorized licensed use limited to: Anglia Ruskin University (a.r.u.). Downloaded on February 19,2024 at 17:42:57 UTC from IEEE Xplore. Restrictions apply.

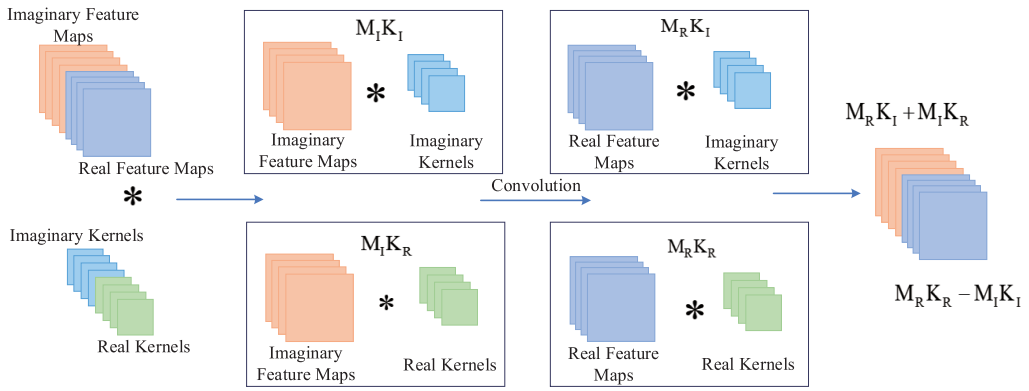


Fig. 3. Schematic diagram of the complex convolution process. Both the real and imaginary parts need to participate in the computation process.

where ϕ is the scaling parameter and η is the bias.

3) Fully Connected Layer: The fully connected layer acts as a classifier in the DNN [53]. To fully utilize the complex-valued network, a complex fully connected layer is used at the end of the complex convolutional network. Similar to the complex convolutional kernel, the complex fully connected layer is also performed through the alternating product of the real and imaginary parts of its weights and the real and imaginary parts of the input signal.

4) Complex Weight Initialization: In DNN, a proper weight initialization helps to prevent sharp changes in gradients. In this article, the convergence of training will be accelerated by a proper initialization of the complex weights, which follows the same principle as the initialization of real-valued weights.

A complex weight W_i can be expressed in polar coordinates or Cartesian coordinates as

$$W_i = |W_i|e^{j\alpha} = \Re\{W_i\} + j\Im\{W_i\} \tag{7}$$

where α and $|W_i|$ are the phase and modulus (or, amplitude) of W_i, respectively; and $\Re\{\cdot\}$ and $\Im\{\cdot\}$ represent the real and imaginary parts, respectively. When W_i is symmetrically distributed around 0, the variance of W_i can be estimated from a single parameter of the Rayleigh distribution. We use the expected value of 0 and the variance of $2\sigma^2$ to initialize the complex weight value from the Rayleigh distribution. The parameter σ will be set differently according to different neural network architectures.

5) Complex Activation Function: The complex activation function used in this article is denoted by $\mathbb{C}$ReLU$(\cdot)$, which is defined as

$$\mathbb{C}\text{ReLU}(z) = \text{ReLU}(\Re(z)) + j\text{ReLU}(\Im(z)) \tag{8}$$

where ReLu$(\cdot)$ is the traditional rectified linear activation function. The complex activation function calculates the activation unit of the current function in the real domain and the complex domain, and finally forms a unified $\mathbb{C}$ReLU(z).

C. Adversarial Example

As mentioned above, adversarial examples refer to a type of artificially constructed examples, which threaten the classifier models and produce wrong predictions by adding specific perturbations to the input data. DL can classify different examples

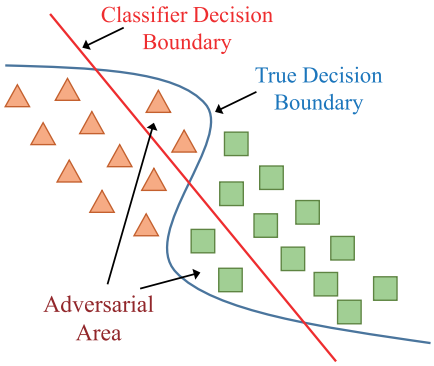


Fig. 4. Schematic diagram of generating adversarial examples. The classifier decision boundary is inconsistent with the true decision boundary, resulting in adversarial examples.

if the model is well trained, and the decision hyperplane, composed by specific detection algorithms, will help distinguish between normal examples and abnormal examples [54]. The optimal decision surface can be obtained if the model is trained with a large amount of labeled training data sets with the same perturbation of the test data sets [55]. However, it is usually difficult to train the model to cover all the example features, and the existing methods cannot guarantee that the model will successfully recognize all example categories. Insufficient training data sets will cause a considerable difference between the model decision surface and the real decision surface. The area corresponding to this difference is the space where the adversarial examples are located.

In the example shown in Fig. 4, triangles and squares represent two different types of examples, respectively. The red line represents the decision boundary of the classifier, and the black line represents the true decision boundary. The two boundaries intersect with each other. If the data exists in the intersecting area, it is considered as an adversarial example, which will cause the classifier to classify it incorrectly.

After the adversarial examples were demonstrated in [27] for the first time, a number of related studies have been reported in the literature, especially in the area of computer vision. Most of the proposed adversarial attacks can be expressed as expressions under the constraint of l_p-norm.

Authorized licensed use limited to: Anglia Ruskin University (a.r.u.). Downloaded on February 19,2024 at 17:42:57 UTC from IEEE Xplore. Restrictions apply.

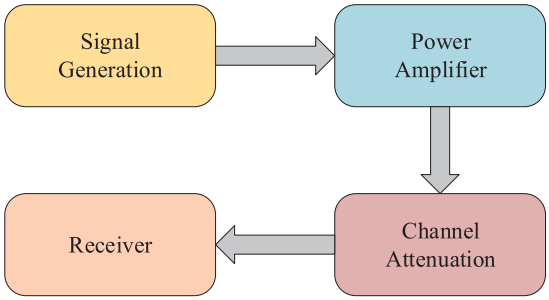


Fig. 5. Process to generate the individual device data sets. In order to identify a individual communication signal transmitter, the collected communication transmitter source signal will be used as the preliminary processing object. We extract power amplifiers from the entire transmitter system and use unified receiving equipment.

Goodfellow *et al.* [21] proposed a high-dimensional linear theory of adversarial examples and the fast gradient signs method (FGSM) that is constrained by the l_∞-norm, which shows that various adversarial example generation methods can be developed based on gradients. Kurakin *et al.* [22] proposed further enhancements that increase the gradient optimization from one-step to an iterative approach. Moosavi-Dezfooli *et al.* [24] proposed the DeepFool scheme to find the closest distance from the original input to the decision boundary of adversarial examples, which is based on the constraint of l_2-norm. Carlini and Wagner [25] proposed an optimization-based adversarial attack method that can effectively fool many defense models under the three measurement l_0, l_2, and l_∞ norm. The adversarial methods based on the optimization of the l_p-norm help to reduce the perceptibility of the perturbation, and the appropriate constraint method can fool the neural networks with the imperceptible perturbation.

III. Methodology

A. Creation of Individual Identification Data Sets

The process of individual communication signal transmission includes signal generation, signal power amplification, channel attenuation, and signal reception, as shown in Fig. 5. During the process, the signal power amplifier magnifies the subtle differences between signals to facilitate the identification of individual device signals. Considering that different receiving devices even with identical parameters could have different effects on signal processing (due to different design and other factors), a unified receiver is used in all the experiments in this article. We provide detailed descriptions of the signal generating device, the signal power amplifier device, and the signal receiving device in the following.

1) Signal Source: Signals are generated using a software tool (i.e., without using real hardware). These signals will be provided to the signal generators after being generated in their own format. After that, the signal generators set the carrier frequency and power of the input signals, and then send the up-converted signals. In the acquisition process, the signal generators are used to change the power, modulation mode, and center frequency of the input signals from the power amplifiers. The frequency of transmitting random data by the signal generators is 433 MHz. The maximum output from the signal generators to the power amplifier is set to 10 dBm.

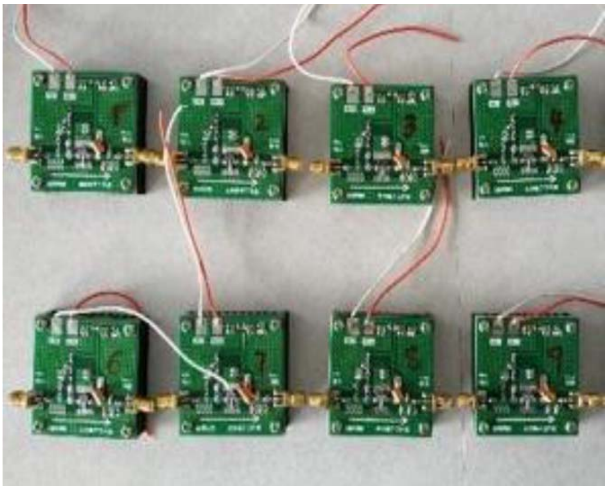


Fig. 6. Eight BLT53A power amplifier for generating individual device signals. Due to the difference of the device physical layers, the amplifiers can be distinguished from each other by the individual device signals.

2) Power Amplifier: The power amplifiers are the core component in the process of device identification. The modulated signals are transmitted to the power amplifiers through an RF cable, and the input signal power is set to 0 dBm. In order to ensure that all the extracted features have the same physical layer characteristics, we select eight power amplifiers of the same model Taidacent BLT53A as shown in Fig. 6 [56]. With the output of these amplifiers generating the required individual device signals, we make full use of the individual differences caused by the inherent dynamic nonlinearity of the amplifiers for individual identification.

3) Channel Attenuation: In order to extract the pure individual signals as much as possible and avoid the influence of other factors on the adversarial perturbation, the cable of the power amplifier is directly connected to the baseband signal receiver. At the meanwhile, for purpose of protecting the receiver, a 30 dB channel attenuator is used between the amplifiers and the receiver.

4) Receiver Device: The signal acquisition equipment is used to collect the in-phase and quadrature (IQ) data sets. We set the sampling rate of the IQ data to 5 MHz, the sampling point to 20000, and the sampling bandwidth to 500 kHz. We collect 1600 samples in total, of which each power amplifier generates 200 samples, and these samples are evenly distributed to avoid sample skew during the model training.

B. Generation of Adversarial Examples

An adversarial attack is to deliberately introduce perturbations to the data sets samples to fool a trained model (e.g., a classifier) to produce a wrong output (e.g., misclassification) with high confidence [57]. Generally speaking, considering a learning system $M(\cdot)$ and a clean input example a, and we assume that example a is correctly classified by the learning system, i.e., $M(a) = y_{\text{true}}$. If there is another example $\tilde{a}$ that is almost identical to a but has been misclassified by the system as $M(\tilde{a}) \neq y_{\text{true}}$, we call example $\tilde{a}$ an adversarial example.

Authorized licensed use limited to: Anglia Ruskin University (a.r.u.). Downloaded on February 19,2024 at 17:42:57 UTC from IEEE Xplore. Restrictions apply.

Kurakin *et al.* [22] first introduced the concept of linear interpretation. Intuitively, it would be unreasonable for the classifier to respond differently to input a and its perturbed version with perturbation ξ, i.e., $\tilde{a} = a + \xi$. In general, for effective classifier design, as long as $\|\xi\|_\infty < \delta$ (δ is sufficiently small), the classifier should classify a and $\tilde{a}$ into the same category. Assuming that the activation function is linear (modeled as multiplication with a vector $\vec{z}$), then the input to the classifier for $\tilde{a}$ is

$$\vec{z}^{\mathrm{T}}\tilde{a} = \vec{z}^{\mathrm{T}}a + \vec{z}^{\mathrm{T}}\xi, \|\xi\|_\infty < \delta \tag{9}$$

thus the classifier output becomes $M(\vec{z}^{\mathrm{T}}a) + M(\vec{z}^{\mathrm{T}}\xi)$. If the perturbation ξ satisfies the maximum norm constraint, i.e., $\xi = \delta \cdot \mathrm{sign}(\vec{z})$, the maximizer of the adversarial perturbation will be attained. For high-dimensional problems, a small change in the input can cause a large change in the output.

According to the previous expressing, adversarial examples are constraint under the l_p-norm. In the signal domain, in addition to measuring the defined range of perturbations, the l_p-norm can also measure the waveform difference between the original signal and the adversarial example. Among them, attacks based on l_0 and l_2 norms will produce more obvious abnormal changes in the waveform [43]. The attack methods in this article are all constraint by the l_∞-norm that are used as the perturbation threshold of all sampling points in the waveform, so that the shape of the perturbed waveform maintains the previous outline but still make the classifier misclassify.

Generally, adversarial attacks are divided into nontargeted attacks and targeted attacks. Nontargeted attacks do not fool the model toward a certain fixed category, as long as it produces perturbations that cause any misclassification [39]. Its loss function is as follows:

$$\underset{\tilde{a}}{\mathrm{argmax}} \; J(f(\theta, \tilde{a}), l) \tag{10}$$

where $f(\cdot)$ is the selected network model, l is the most likely output category, and J is the loss function used to evaluate the modeling effect of the algorithm on the data sets [58]. Adversarial attacks can maximize the loss function, making the model unable to obtain the optimal parameters θ to achieve the purpose of the attack. Therefore, the nontargeted attack only needs to ensure that the perturbation is added along the opposite direction of the gradient, and the distance between the perturbed sample and the original sample is increased.

In targeted attack, the attacker has a specific target and needs to ensure that the current attack can drive the network learn and classify toward a desired direction. Its loss function is as follows:

$$\underset{\tilde{a}}{\mathrm{argmin}} \; J\big(f(\theta, \tilde{a}), l^*\big) \tag{11}$$

where the original label l is replaced with the type l^* that the attacker expects the final output of the classifier model to be. That is, the targeted attack needs to subtract the calculated perturbation from the original sample to obtain the adversarial examples. In order to improve the credibility of the target type, the loss function of the target needs to be minimized to output the specified result. The typical methods to generate adversarial examples are as follows.

1) Fast Gradient Signs Method: FGSM is one of the simplest methods to generate adversarial examples [21]. The key is to make the input example move in the direction of decreasing category confidence. Let θ be the model parameter and ξ be the perturbation level. Let $J(\theta, a, l)$ be the loss function for training the neural network and $\nabla_a J(\theta, a, l)$ is the partial derivative of the loss function. We can linearize the loss function around the current value of ξ to obtain the maximum norm limit of perturbation. Specifically, we first obtain the gradient value G_a as

$$G_a = \nabla_a J(\theta, a, l). \tag{12}$$

Next, we calculate the perturbation level ξ as follows:

$$\xi = \delta \cdot \mathrm{sign}(G_a). \tag{13}$$

Under the targeted attack, the adversarial example generated by FGSM is as follows:

$$a' = a + \delta \cdot \mathrm{sign}\big(\nabla_a J_{F,l^*}(a)\big) \tag{14}$$

where F is the objective function and l^* is the target category. Each time FGSM is implemented, the gradient will be updated from the original signal a to generate adversarial example a'.

2) Basic Iterative Method: This method improves the effect of FGSM with an iterative optimizer [22]. The basic iterative method (BIM) executes FGSM with small step sizes and crops the updated adversarial examples to the effective range. A total of T iterations are performed this way. In BIM, the gradient operation should be performed first to obtain the gradient value of the tth operation

$$G_{a_t} = \nabla_{a_t} J\big(\theta, a'_t, l\big). \tag{15}$$

The gradient in the $(t+1)$th iteration is then update as

$$a'_{t+1} = \mathrm{Clip}_{a,\delta}\big\{a'_t + \delta \cdot \mathrm{sign}\big(G_{a_t}\big)\big\} \tag{16}$$

where $\mathrm{Clip}_{a,\delta}\{x\}$ means to cut x to the range $[a-\delta, a+\delta]$. Under the targeted attack, the adversarial example generated by BIM is as follows:

$$a'_{t+1} = a_t - \mathrm{Clip}\big\{\delta \cdot \mathrm{sign}\big(\nabla_a J_{F,l^*}(a_t)\big)\big\}. \tag{17}$$

3) Projected Gradient Descent: Projected gradient descent (PGD) can be regarded as a generalized form of BIM, with no constraints on the iteration step [23]. To constrain the adversarial perturbation, PGD projects the adversarial examples learned in each iteration into the δ neighborhood of the benign examples, so that the value of the adversarial perturbation will not exceed δ. The update method is as follows:

$$a'_{t+1} = \mathrm{Proj}\big\{a'_t + \delta \cdot \mathrm{sign}\big(G_{a_t}\big)\big\}. \tag{18}$$

The Proj$\{\cdot\}$ operation projects the updated adversarial examples into the δ neighborhood with an effective range. The adversarial example generated by PGD with targeted attack is as follows:

$$a'_t = a'_{t-1} - \mathrm{Proj}\big\{\delta \cdot \mathrm{sign}\big(\nabla_a J_{F,l^*}\big(a'_{t-1}\big)\big)\big\}. \tag{19}$$

Authorized licensed use limited to: Anglia Ruskin University (a.r.u.). Downloaded on February 19,2024 at 17:42:57 UTC from IEEE Xplore. Restrictions apply.

4) Momentum Iterative Method: Inspired by the momentum optimizer, Dong *et al.* [59] proposed to integrate momentum memory into the iterative process, and introduced a new iterative algorithm named momentum iterative method (MIM). Because the iterative FGSM-based method will move the adversarial examples to the direction of the gradient symbol in each iteration, which is prone to local optimal solutions and overfitting. MIM integrates momentum into the iterative FGSM, so that the direction of each model update could remain stable. The gradient of MIM is calculated as follows:

$$G_{t+1} = \mu G_t + \frac{\nabla_a J_\theta\left(a'_t, l\right)}{\left\|\nabla_a J_\theta\left(a'_t, l\right)\right\|_1} \tag{20}$$

where μ is a decay factor that affects the attack effect. When $\mu = 0$, MIM will become an ordinary iterative attack. After updating the parameter G_{t+1} by accumulating the velocity vector in the gradient direction, the updated adversarial example for a T-iteration procedure is as follows:

$$a'_{t+1} = a'_t + \frac{\delta}{T} \cdot \text{sign}(G_{t+1}). \tag{21}$$

When combine the targeted attack with MIM under the l_∞ constraint, the adversarial example can be got as following:

$$a'_{t+1} = a'_t - \frac{\delta}{T} \cdot \text{sign}\left(\mu G_t + \frac{\nabla_a J_\theta\left(a'_t, l^*\right)}{\left\|\nabla_a J_\theta\left(a'_t, l^*\right)\right\|_1}\right). \tag{22}$$

C. Combined Evaluation Indicators of Logits

The maximum predicted value output by the last layer (i.e., softmax) of DNNs serves as the confidence. As the exponential function is used in softmax, a large input leads to a much larger output [60], which leads to the difficulty of acquiring the true difference between samples in the prediction results. The evaluation of the adversarial attack performance from the logits layer of DNNs is more microscopic. Logits represents the function that maps the probability (in [0, 1]) to the entire real number domain with the following mapping:

$$L = \ln\left(\frac{p}{1-p}\right) \tag{23}$$

where L represents the logits value and p represents a certain probability value. Logits in DNNs represent the layer before softmax without normalization.

We propose a set of improved general indicators called combined evaluation indicators of logits. While capturing the confidence of the output, these indicators show the classification effect of the model on the current sample before and after the perturbation is introduced in a more intuitive manner. The proposed combined evaluation indicators of logits express the logits difference from two perspectives. The source logits difference represents the maximum output of subtracting all incorrect types from the real class output, which can be described as follows:

$$\Delta\text{logits} = l_s - l_T, \quad l_T = \max(l_k \ \forall k \neq s) \tag{24}$$

where l_s is the logits value of the original category and l_T is the logits value with the largest prediction among other categories except the original category. Furthermore, considering the goals of targeted attack, the target logits difference can be evaluated as follows:

$$\Delta\text{logits} = l_t - l_S, \quad l_S = \max(l_i \ \forall i \neq t) \tag{25}$$

where l_t is the logits value of the selected target category and l_S is the maximum logits value except for the target category.

From the perspective of the target category, the performance of the targeted attack and the robustness of different signals can be measured more intuitively. The logits output layer of the complex neural network is used to evaluate the effect on the output of the classifier model after the adversarial perturbation is introduced in this article, while the combined evaluation indicators of logits help to intuitively analyze the performance of targeted attacks.

IV. Experimental Study and Discussions

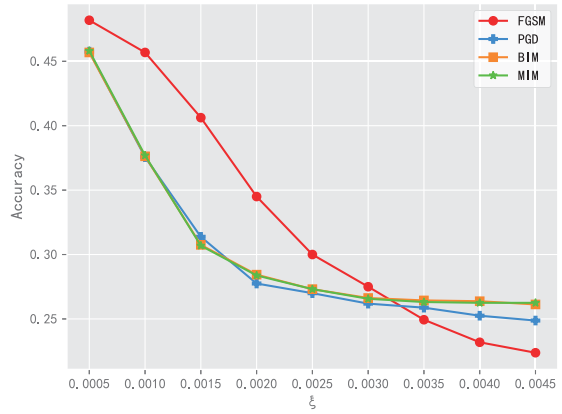
As shown in Section III, four different methods can be applied to generate adversarial examples. Two types of different attack methods are considered for testing in our experimental study: 1) nontargeted attack and 2) targeted attack. In the experiments of nontargeted attacks, we control the variables to evaluate the influence on the signal receiver. In the targeted attack experiments, 200 original examples for each signal are used to study which kind of the signals can be identified by the receiver after the adversarial attack.

In this study, all the experiments were carried out on an NVIDIA GeForce GTX 1080Ti, and only one GPU was used for one operation. Based on the CleverHans library [61], we implemented model selection of the generation methods of the adversarial examples. TensorFlow and Keras machine learning frameworks were used to train the DNN models. The open-source library provides a method for constructing adversarial examples and provides a reference for standardized implementation of adversarial examples.

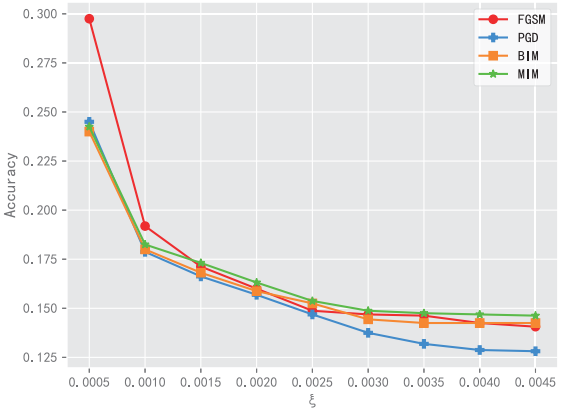
A. Evaluation of Different Perturbation Levels Under Nontargeted Attacks

First, to examine the effect of different attack methods on identification of individual device signals, Fig. 7 shows the trend in the identification accuracy under the four attack methods. The signal-to-noise ratio (SNR) is set to 0 and 15 dB, respectively, and the iteration step λ is 0.0004. In this experiment, we use the one-step attack method FGSM and three iterative attack methods, i.e., BIM, PGD, and MIM, and their effects are evaluated by the accuracy of individual device signal identification under different perturbation levels ξ. The experimental results show that as the perturbation level is increased, all the four attack methods cause degraded identification accuracy. As Fig. 7(a) shows, the effect of the iterative attacks is stronger than that of the one-step attack when the SNR is 0 dB. As the perturbation level continues to increase, the effect of FGSM exceeds that of the iterative attack methods eventually. Fig. 7(b) demonstrates the trend of the four attack methods along with increased perturbation level when the SNR is 15 dB. The identification accuracy tends to 12.5% as ξ

Authorized licensed use limited to: Anglia Ruskin University (a.r.u.). Downloaded on February 19,2024 at 17:42:57 UTC from IEEE Xplore. Restrictions apply.



(a)



(b)

Fig. 7. Effect of nontargeted attacks with increased perturbation level ξ. The iteration step λ is 0.0004. The one-step attack method FGSM and the three iterative attack methods BIM, PGD, and MIM are evaluated. The experiment shows that the identification accuracy decreases with increased perturbation level ξ. (a) SNR $=$ 0 dB. (b) SNR $=$ 15 dB.

is further increased, which also represents the characteristics under randomized distribution of the nontargeted attacks.

The four attack methods are all gradient-based attacks, and FGSM is the only one-step attack among them. Gradient-based attacks are essentially seeking to maximize the loss value. Since FGSM does not require an iterative operation, when the highest disturbed threshold is given, perturbation is generated in the direction of the initial gradient loss maximization, but the other three methods need to find a new gradient direction after each iteration. The received signal contains more noise in low-SNR, which affects the gradient and iterative operation of the iterative attacks and limits their attack effect, resulting in better performance of FGSM than iterative attacks after ξ is greater than 0.003. In addition, the received signal retains more original information containing adversarial examples in high-SNR, and high-quality signals will generate more powerful perturbation. Therefore, for the same perturbation level, the identification accuracy for SNR = 15 dB is lower than that for SNR = 0 dB. In real communication scenarios, we are always committed to finding a solution that can achieve a higher SNR at the receiver, which, however, makes the adversarial attacks more effective.

B. Evaluation of Different Iteration Steps Under Nontargeted Attacks

Some characteristics of the iterative attacks have been revealed in the previous experiments. In this experiment, we study the impact of iteration steps on the attack effect, by comparing the iterative methods BIM, PGD, and MIM with FGSM under the same perturbation level. The experiment results are presented in Fig. 8, which are obtained when the SNR is 0 and 15 dB, respectively. It can be seen from the figures that under the same perturbation level ξ, with a larger iteration step λ (i.e., "iter" in the legend), the identification accuracy exhibits a downward trend. When λ is 0.0001, the attack effect of PGD soon maintains a stable trend as ξ is increased when SNR is low, as shown in Fig. 8(a). However, when λ is larger than 0.0001, the accuracy of individual identification drops sharply as ξ is increased, and then it converges to a stable value as ξ is further increased. In Fig. 8(d), PGD's accuracy approaches a stable value more quickly than that in Fig. 8(a) as ξ is increased. After convergence, the accuracy gap between the case when λ is 0.0005 and the case when λ is 0.0001 is about 5%.

Among the three iterative attack methods, PGD achieves a more significant effect compared to BIM and MIM. We also find that FGSM demonstrates a superior performance in all the cases. Its attack effect eventually exceeds the attack effects of PGD, BIM, and MIM as ξ is increased. It also can be found from Fig. 8 that when the iteration step is low, accuracy tends to start to converge at a lower ξ, which can be attributed to the iterative attack is a neural network training process that optimizes the goal is an adversarial perturbation. The iterative attack is dedicated to increase the value of the cross-entropy loss function in the process of generating perturbation. Generally, when the distance between the original sample and the decision boundary is large, a larger iteration step helps to increase the loss function quickly, thereby generating more powerful adversarial examples. However, when the iteration step reaches a certain level, the strategy of increasing the iteration step to enhance the attack performance is of limited effect.

C. Evaluation of Signal Identification Under Targeted Attacks

The previous experiments are conducted for nontargeted attacks. There has also been considerable interest in targeted attacks in more advanced scenarios. Therefore, we study the attack effects of the adversarial examples in a targeted attack experiment. In this experiment, we choose eight types of signals, denoted by PA1, PA2, PA3, PA4, PA5, PA6, PA7, and PA8, to examine the number of successfully identified signals when the perturbation level is 0, 0.001, and 0.002, and the SNR is at 0 and 15 dB, respectively. We generate 200 adversarial examples for each type of signals. The confusion matrices of classification results are plotted in Fig. 9.

The experimental results show that a high SNR and a high perturbation level can greatly strengthen the attack effect of adversarial examples. As shown in Fig. 9(a) and (d), in the case of no attack, the number of correctly identified signals is

Authorized licensed use limited to: Anglia Ruskin University (a.r.u.). Downloaded on February 19,2024 at 17:42:57 UTC from IEEE Xplore. Restrictions apply.

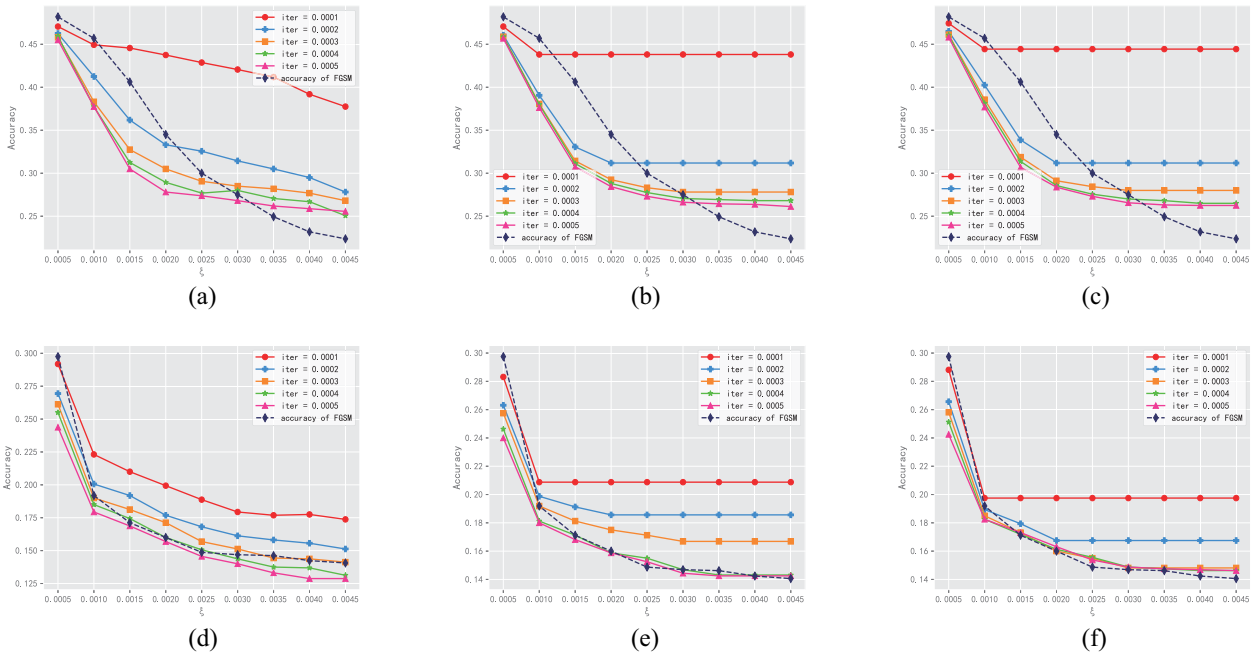


Fig. 8. Effect of nontargeted attacks with the increased perturbation level ξ when SNR is 0 dB for (a)–(c), and SNR is 15 dB for (d)–(f). The effect of FGSM is compared with that of the three iterative schemes. PGD has the strongest attack effect among the three iterative attacks, and FGSM also achieves a strong attack effect. (a) PGD, SNR = 0 dB. (b) BIM, SNR = 0 dB. (c) MIM, SNR = 0 dB. (d) PGD, SNR = 15 dB. (e) BIM, SNR = 15 dB. (f) MIM, SNR = 15 dB.

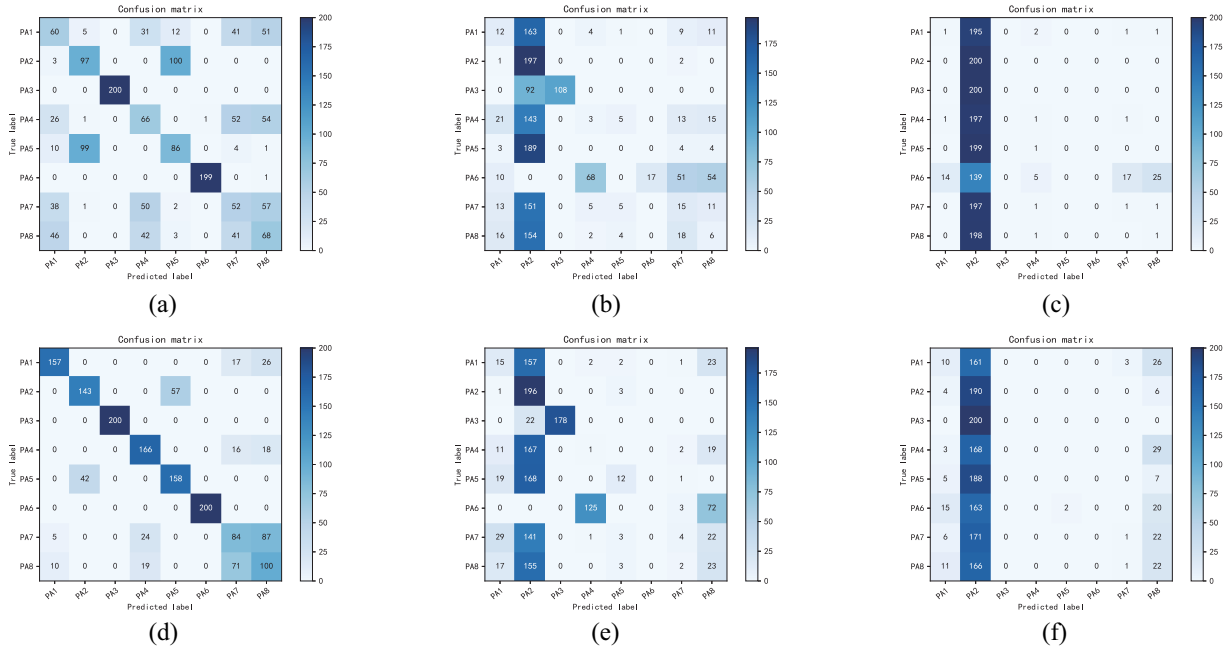


Fig. 9. Confusion matrices for evaluating the impact of adversarial examples on the classification performance of the identification model under the targeted attacks. (a) and (d) are the cases of not being attacked. We choose BIM as the attack method. The attack effects are more obvious for the cases with a high SNR and a high perturbation level. (a) No attack, SNR = 0 dB. (b) BIM, $\xi = 0.001$, SNR = 0 dB. (c) BIM, $\xi = 0.002$, SNR = 0 dB. (d) No attack, SNR = 15 dB. (e) BIM, $\xi = 0.001$, SNR = 15 dB. (f) BIM, $\xi = 0.002$, SNR = 15 dB.

the largest when the SNR is 15 dB, which is far more than that when the SNR is 0 dB. This result shows that under no attack, the receiver's misclassification of the signal is mainly caused by the channel noise. As SNR is increased, the noise perturbation decreases, and the accuracy of individual device signal identification is improved. Fig. 9(b) and (c) show that when the original signals are attacked, increasing the perturbation levels greatly strengthens the effectiveness of targeted attacks, making it much easier for the receiver to misidentify the signals as the expected target PA2.

The targeted attack experiment has strong practical significance in that an excellent attack effects provide us with useful insights. In summary, increasing the perturbation level is most effective for strengthening the targeted attack effect. When

Authorized licensed use limited to: Anglia Ruskin University (a.r.u.). Downloaded on February 19,2024 at 17:42:57 UTC from IEEE Xplore. Restrictions apply.

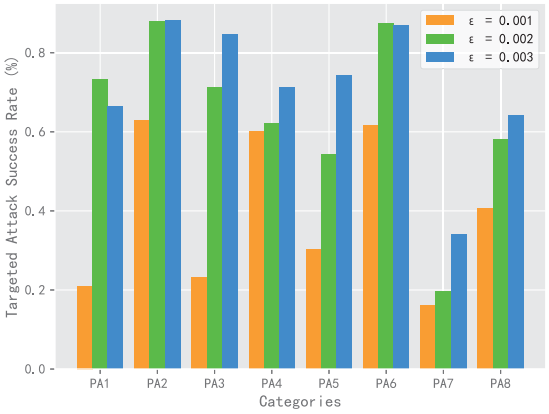


Fig. 10. Targeted attack success rates at different perturbation levels. When ξ rises to 0.002 from 0.001, the attack success rate on average is increased by 30%. Increasing ξ further, the attack success rate tends to increase as well.

the perturbation level is increased, the original signals contain more features of the directed target, consequently it is more likely to be misjudged by the receiver.

D. Evaluation of Attack Success Rate Under Targeted Attacks

Motivated by the previous experiment, we make an overall evaluation of the targeted attack effects under different perturbation levels. In this experiment, we examine the targeted attack success rate when ξ is 0.001, 0.002, and 0.003. The targeted attack success rate, denoted by ASR, is defined as

$$\mathrm{ASR} = \frac{N_{\mathrm{sum}}}{N_{\mathrm{total}}} \tag{26}$$

where N_{sum} and N_{total} represent the number of examples classified into the target class and the total number of examples, respectively.

The experimental results are presented in Fig. 10, which show that increasing the perturbation level can significantly increase the success rate of targeted attacks, especially for PA1 and other vulnerable signals. When ξ is 0.001, the attack success rate on PA1 is only 20.0%; when ξ is 0.002, the attack success rate on PA1 becomes 67.5%. However, we also find that in this experiment, there are some signals that are robust to targeted attacks. For example, increasing ξ has a limited effect on the attack success rate for PA7. This is because that the structure of these signals is relatively stable and the ability to resist noise is strong. As a result, they can defend against attacks from adversarial examples to a certain extent. However, for most signals, as the perturbation level is increased, the targeted attack success rate rises sharply.

E. Evaluation With Combined Evaluation Indicators of Logits

After evaluating the attack effect as indicated by the model prediction accuracy, we evaluate the logits layer of the data sets from a micro perspective to measure the inducibility of these signal samples. From the previous analysis, when the signal quality is higher, the adversarial perturbation is less likely to be affected by noise, and the response of

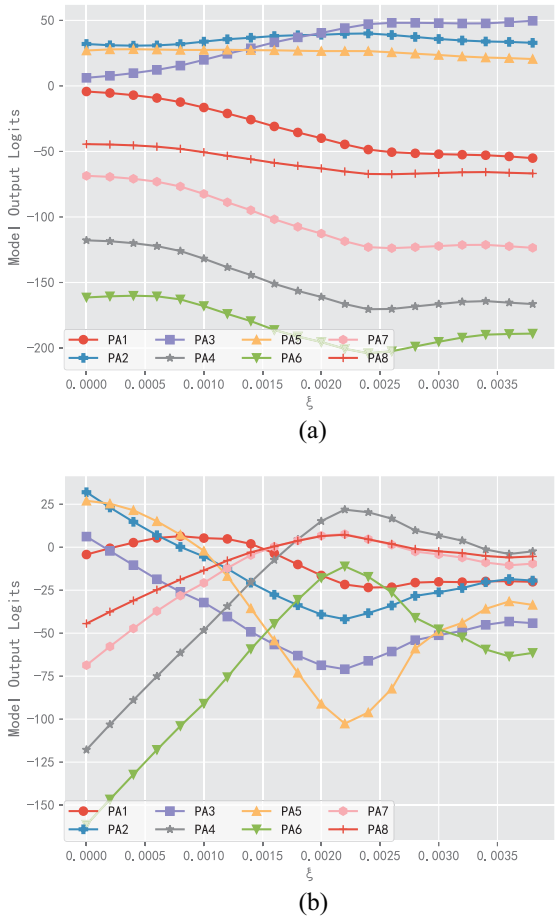


Fig. 11. BIM is used to evaluate the change of logits output value of various categories of individual device signals with ξ when SNR = 15 dB. Experiments show that with the increase of the disturbance level, the targeted attack effect is gradually effective with the increase of the perturbation level. (a) Original category PA2, target category PA3. (b) Original category PA2, target category PA4.

the model to the targeted attack is strongly related to the adversarial perturbation. With this consideration, the experiments in this part are all implemented under the condition of SNR = 15 dB.

Fig. 11 shows the change of the model logits prediction value of the PA2 signal with increased perturbation level in the case of different target-oriented types. In Fig. 11(a), when there is no attack ($\xi = 0$), the predicted value of the original category PA2 is the highest, indicating that the model can output the correct prediction results most of the time. As the perturbation level is increased, the model predicted value of the original category begins to decrease, and the confidence of the target PA3 shows a gradually increasing trend. As $\xi = 0.0018$, the curves of PA2 and PA3 cross each other, and the model begins to predict PA2 as PA3, indicating that the targeted attack has begun to succeed. It is worth noting that the adversarial attack is effective when $\xi = 0.0002$, and the original category is identified as the wrong category in Fig. 11(b). However, when $\xi = 0.0018$, PA2 is recognized as the target category PA4. Since the targeted attack requires the loss function to be maximized toward the desired category, which requires more interaction with the model. Therefore,

Authorized licensed use limited to: Anglia Ruskin University (a.r.u.). Downloaded on February 19,2024 at 17:42:57 UTC from IEEE Xplore. Restrictions apply.

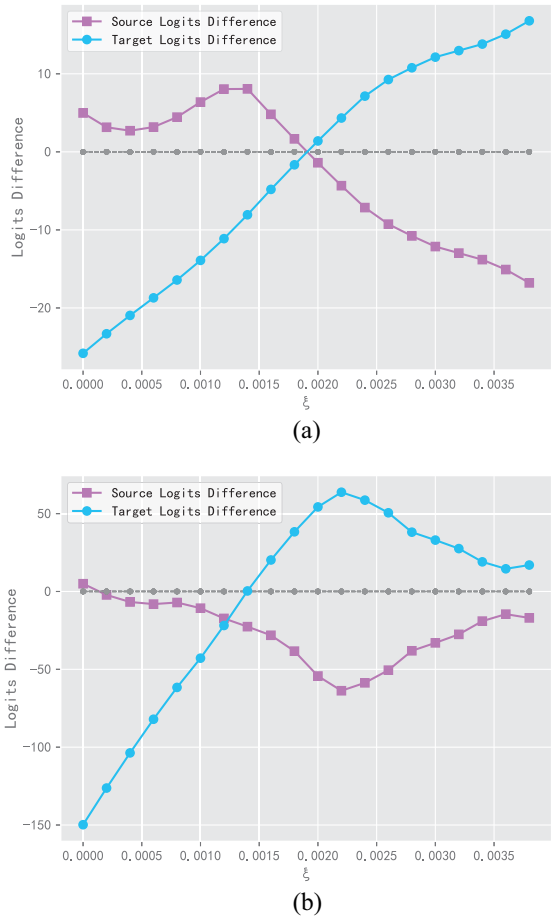


Fig. 12. From the source category and the target category, the logits difference of model prediction under the adversarial attacks is evaluated when SNR = 15 dB. Experiments show that with the increase of the perturbation level, the target category difference shows an upward trend, indicating that the targeted attack performance has been strengthened. (a) Original category PA2, target category PA3. (b) Original category PA2, target category PA4.

the targeted attack requires a stronger perturbation intensity to shorten the perception gap between the original category and the target category.

Finally, we evaluate the logits difference predicted by the model and present the results in Fig. 12. Before ξ reaches 0.0018, the source logits difference is always positive, indicating that the model has a high degree of confidence in the prediction of signal PA2 in Fig. 12(a). With the increase of the perturbation level ξ, the logits difference of the source class continues to decrease, and the confidence of the target class continues to grow, indicating that the effect of the targeted attack is becoming stronger. It can be seen from Fig. 12(b) that the intersection of the two curves is not at the zero level, indicating that the adversarial attack first forces the model to misclassify the original signal into other types, and then successfully induces it to the expected target type.

As shown above, targeted attacks are more complicated than nontargeted attacks, but they have a broader development space and practical uses. By means of the combined evaluation indicators of logits, the anti-perturbation ability of signal samples can be analyzed.

V. Conclusion and Future Work

In this article, we examined the security of DL-based device identification under adversarial examples. We found that DL models were vulnerable to nontargeted adversarial attacks, as the misidentification rate could rise sharply with a smaller perturbations. Our investigation showed that iterative attack methods were more effective to fool the DL models generally. Increasing the perturbation level and iterative steps can increase the success rate of adversarial attacks, but the recognition accuracy of DL models will converge to a stable value as the perturbation level is further increased. We also evaluated the effectiveness of targeted attacks, and the results showed that DL models were also sensitive to targeted attacks, resulting in outputting the categories as expected by the attacker. Finally, we use the proposed combined evaluation indicators of logits to quantify the fine-grained classification effect of different individual device signals, enriching the evaluation criteria for signal adversarial examples.

Our study indicates that adversarial attacks pose a great threat to the security of device identification in the IoT. For future work, we will explore the following strategies: 1) we will consider the black-box attack with alternative models to attack the various targeted models; 2) due to the high real-time requirements for adversarial attacks in actual scenarios, we will design simpler and more powerful attack algorithms; and 3) future researches will be oriented to a more realistic physical environment, and the channel effects will be fully considered to evaluate adversarial attacks initiated by transmitters.

References

[1] M. Wang, Y. Lin, Q. Tian, and G. Si, "Transfer learning promotes 6G wireless communications: Recent advances and future challenges," *IEEE Trans. Rel.*, vol. 70, no. 2, pp. 780–807, Jun. 2021.

[2] Y. Wang, G. Gui, H. Gacanin, T. Ohtsuki, H. Sari, and F. Adachi, "Transfer learning for semi-supervised automatic modulation classification in ZF-MIMO systems," *IEEE J. Emerg. Sel. Topics Circuits Syst.*, vol. 10, no. 2, pp. 231–239, Jun. 2020.

[3] J. Lin, W. Yu, N. Zhang, X. Yang, H. Zhang, and W. Zhao, "A survey on Internet of Things: Architecture, enabling technologies, security and privacy, and applications," *IEEE Internet Things J.*, vol. 4, no. 5, pp. 1125–1142, Oct. 2017.

[4] L. Cui, S. Yang, F. Chen, Z. Ming, N. Lu, and J. Qin, "A survey on application of machine learning for Internet of Things," *Int. J. Mach. Learn. Cybern.*, vol. 9, no. 8, pp. 1399–1417, Jun. 2018.

[5] F. Samie, L. Bauer, and J. Henkel, "From cloud down to things: An overview of machine learning in Internet of Things," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 4921–4934, Jun. 2019.

[6] S. Huang, C. Lin, W. Xu, Y. Gao, Z. Feng, and F. Zhu, "Identification of active attacks in Internet of Things: Joint model- and data-driven automatic modulation classification approach," *IEEE Internet Things J.*, vol. 8, no. 3, pp. 2051–2065, Feb. 2021.

[7] S. Huang, Y. Yao, Z. Wei, Z. Feng, and P. Zhang, "Automatic modulation classification of overlapped sources using multiple cumulants," *IEEE Trans. Veh. Technol.*, vol. 66, no. 7, pp. 6089–6101, Jul. 2017.

[8] Y. Lin, Y. Tu, and Z. Dou, "An improved neural network pruning technology for automatic modulation classification in edge devices," *IEEE Trans. Veh. Technol.*, vol. 69, no. 5, pp. 5703–5706, May 2020.

[9] Y. Lin, X. Zhu, Z. Zheng, Z. Dou, and R. Zhou, "The individual identification method of wireless device based on dimensionality reduction and machine learning," *J. Supercomput.*, vol. 75, no. 6, pp. 3010–3027, Dec. 2017.

[10] P. Khan, B. S. K. Reddy, A. Pandey, S. Kumar, and M. Youssef, "Differential channel-state-information-based human activity recognition in IoT networks," *IEEE Internet Things J.*, vol. 7, no. 11, pp. 11290–11302, Nov. 2020.

Authorized licensed use limited to: Anglia Ruskin University (a.r.u.). Downloaded on February 19,2024 at 17:42:57 UTC from IEEE Xplore. Restrictions apply.

[11] D. Roy, T. Mukherjee, M. Chatterjee, E. Blasch, and E. Pasiliao, "RFAL: Adversarial learning for RF transmitter identification and classification," *IEEE Trans. Cogn. Commun. Netw.*, vol. 6, no. 2, pp. 783–801, Jun. 2020.

[12] Y. Wang, M. Liu, J. Yang, and G. Gui, "Data-driven deep learning for automatic modulation recognition in cognitive radios," *IEEE Trans. Veh. Technol.*, vol. 68, no. 4, pp. 4074–4077, Apr. 2019.

[13] K. Anupama, Y. C. Rao, and V. K. Gurrala, "A machine learning approach to monitor water quality in aquaculture," *Int. J. Performability Eng.*, vol. 16, no. 12, pp. 1845–1852, Dec. 2020.

[14] Y. Lin, Y. Tu, Z. Dou, L. Chen, and S. Mao, "Contour stella image and deep learning for signal recognition in the physical layer," *IEEE Trans. Cogn. Commun. Netw.*, vol. 7, no. 1, pp. 34–46, Mar. 2021.

[15] Y. Dong, X. Jiang, H. Zhou, Y. Lin, and Q. Shi, "SR2CNN: Zero-shot learning for signal recognition," *IEEE Trans. Signal Process.*, vol. 69, pp. 2316–2329, Mar. 2021, doi: 10.1109/TSP.2021.3070186. [Online]. Available: https://ieeexplore.ieee.org/document/9392373

[16] T. Baltrušaitis, C. Ahuja, and L. Morency, "Multimodal machine learning: A survey and taxonomy," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 2, pp. 423–443, Feb. 2019.

[17] Y. Tu, Y. Lin, and J. Wang, "Semi-supervised learning with generative adversarial networks on digital signal modulation classification," *Comput. Mater. Continua*, vol. 55, no. 2, pp. 243–254, May 2018.

[18] K. Merchant, S. Revay, G. Stantchev, and B. Nousain, "Deep learning for RF device fingerprinting in cognitive communication networks," *IEEE J. Sel. Topics Signal Process.*, vol. 12, no. 1, pp. 160–167, Feb. 2018.

[19] C. Zhu, W. Huang, H. Li, G. Taylor, C. Studer, and T. Goldstein, "Transferable clean-label poisoning attacks on deep neural nets," in *Proc. 36th Int. Conf. Mach. Learn. (PMLR)*, May 2019, pp. 7614–7623.

[20] S. Zhao, X. Ma, X. Zheng, J. Bailey, J. Chen, and Y. Jiang, "Clean-label backdoor attacks on video recognition models," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 14443–14452.

[21] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," in *Proc. ICLR*, San Diego, CA, USA, May 2015, pp. 1–11.

[22] A. Kurakin, I. J. Goodfellow, and S. Bengio. *Adversarial Examples in the Physical World*. Accessed: Nov. 2016. [online] Available: https://arxiv.org/abs/1607.02533.

[23] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu. *Towards Deep Learning Models Resistant to Adversarial Attacks*. Accessed: Sep. 2019. [Online]. Available: https://arxiv.org/abs/1706.06083.

[24] S. 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.

[25] 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.

[26] Z. Wang, M. Song, Z. Zhang, Y. Song, Q. Wang, and H. Qi, "Beyond inferring class representatives: User-level privacy leakage from federated learning," in *Proc. IEEE INFOCOM Conf. Comput. Commun.*, Apr. 2019, pp. 2512–2520.

[27] C. Szegedy *et al.*, "Intriguing properties of neural networks," in *Proc. ICLR*, Apr. 2014, pp. 1–10.

[28] N. Akhtar and A. Mian, "Threat of adversarial attacks on deep learning in computer vision: A survey," *IEEE Access*, vol. 6, pp. 14410–14430, 2018.

[29] X. Wang *et al.*, "Adversarial examples for improving end-to-end attention-based small-footprint keyword spotting," in *Proc. IEEE ICASSP*, Brighton, U.K., May 2019, pp. 6366–6370.

[30] E. Wallace, P. Rodriguez, S. Feng, I. Yamada, and J. Boyd-Graber, "Trick me if you can: Human-in-the-loop generation of adversarial examples for question answering," *Trans. Assoc. Comput. Linguist.*, vol. 7, no. 1, pp. 387–401, Jul. 2019.

[31] M. Sadeghi and E. G. Larsson, "Physical adversarial attacks against end-to-end autoencoder communication systems," *IEEE Commun. Lett.*, vol. 23, no. 5, pp. 847–850, May 2019.

[32] Y. E. Sagduyu, Y. Shi, and T. Erpek, "IoT network security from the perspective of adversarial deep learning," in *Proc. IEEE Int. Conf. Sens. Commun. Netw. (SECON)*, Jun. 2019, pp. 1–9.

[33] S. Kokalj-Filipovic, R. Miller, N. Chang, and C. L. Lau, "Mitigation of adversarial examples in RF deep classifiers utilizing autoEncoder pre-training," in *Proc. Int. Conf. Mil. Commun. Inf. Syst. (ICMCIS)*, May 2019, pp. 1–6.

[34] S. Kokalj-Filipovic, R. Miller, and J. Morman, "Targeted adversarial examples against RF deep classifiers," in *Proc. ACM Workshop Wireless Security Mach. Learn.*, May 2019, pp. 6–11.

[35] Y. E. Sagduyu, Y. Shi, and T. Erpek, "Adversarial deep learning for over-the-air spectrum poisoning attacks," *IEEE Trans. Mobile Comput.*, vol. 20, no. 2, pp. 306–319, Feb. 2021.

[36] B. Kim, Y. E. Sagduyu, K. Davaslioglu, T. Erpek, and S. Ulukus, "Over-the-air adversarial attacks on deep learning based modulation classifier over wireless channels," in *Proc. 54th Annu. Conf. Inf. Sci. Syst. (CISS)*, Princeton, NJ, USA, Mar. 2020, pp. 1–6.

[37] Y. Shi, K. Davaslioglu, and Y. E. Sagduyu, "Over-the-air membership inference attacks as privacy threats for deep learning-based wireless signal classifiers," in *Proc. 2nd ACM Workshop Wireless Security Mach. Learn.*, Jul. 2020, pp. 61–66.

[38] S. Bair, M. Del Vecchio, B. Flowers, A. J. Michaels, and W. C. Headley, "On the limitations of targeted adversarial evasion attacks against deep learning enabled modulation recognition," in *Proc. ACM Workshop Wireless Security Mach. Learn.*, May 2019, pp. 25–30.

[39] B. Flowers, R. M. Buehrer, and W. C. Headley, "Evaluating adversarial evasion attacks in the context of wireless communications," *IEEE Trans. Inf. Forensics Security*, vol. 15, pp. 1102–1113, Aug. 2020, doi: 10.1109/TIFS.2019.2934069. [Online]. Available: https://ieeexplore.ieee.org/document/8792120

[40] Q. Tian *et al.*, "New security mechanisms of high-reliability IoT communication based on radio frequency fingerprint," *IEEE Internet Things J.*, vol. 6, no. 5, pp. 7980–7987, Oct. 2019.

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

[42] Y. Lin, H. Zhao, X. Ma, Y. Tu, and M. Wang, "Adversarial attacks in modulation recognition with convolutional neural networks," *IEEE Trans. Rel.*, vol. 70, no. 1, pp. 389–401, Mar. 2021.

[43] H. Zhao, Y. Lin, S. Gao, and S. Yu, "Evaluating and improving adversarial attacks on DNN-based modulation recognition," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2020, pp. 1–5.

[44] S. Kokalj-Filipovic, R. Miller, and G. Vanhoy, "Adversarial examples in RF deep learning: Detection and physical robustness," in *Proc. IEEE Global Conf. Signal Inf. Process. (GlobalSIP)*, Ottawa, ON, Canada, Nov. 2019, pp. 1–5.

[45] M. Z. Hameed, A. György, and D. Gündüz, "The best defense is a good offense: Adversarial attacks to avoid modulation detection," *IEEE Trans. Inf. Forensics Security*, vol. 16, pp. 1074–1087, Sep. 2020, doi: 10.1109/TIFS.2020.3025441. [Online]. Available: https://ieeexplore.ieee.org/document/9201397

[46] C. Trabelsi *et al.*, "Deep complex networks," in *Proc. Int. Conf. Learn. Represent.*, Jan. 2018, pp. 1–19.

[47] Y. Tu, Y. Lin, C. Hou, and S. Mao, "Complex-valued networks for automatic modulation classification," *IEEE Trans. Veh. Technol.*, vol. 69, no. 9, pp. 10085–10089, Sep. 2020.

[48] D. Bhavana, K. K. Kumar, M. B. Chandra, P. V. S. K. Bhargav, D. J. Sanjana, and G. M. Gopi, "Hand sign recognition using CNN," *Int. J. Performability Eng.*, vol. 17, no. 3, pp. 314–321, Mar. 2021.

[49] E. Shelhamer, J. Long, and T. Darrell, "Fully convolutional networks for semantic segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 4, pp. 640–651, Apr. 2017.

[50] V. Badrinarayanan, A. Kendall, and R. Cipolla, "SegNet: A deep convolutional encoder–decoder architecture for image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 12, pp. 2481–2495, Dec. 2017.

[51] J. Yu, A. Hu, G. Li, and L. Peng, "A robust RF fingerprinting approach using multisampling convolutional neural network," *IEEE Internet Things J.*, vol. 6, no. 4, pp. 6786–6799, Aug. 2019.

[52] A. Adhikari, A. Ram, R. Tang, and J. Lin, "Rethinking complex neural network architectures for document classification," in *Proc. Conf. North Amer. Assoc. Comput. Linguist. Human Lang. Technol.*, Minneapolis, MN, USA, Jun. 2019, pp. 4046–4051.

[53] Y. Zhang, W. Chan, and N. Jaitly, "Very deep convolutional networks for end-to-end speech recognition," in *Proc. IEEE ICASSP*, New Orleans, LA, USA, Mar. 2017, pp. 4845–4849.

[54] S. Pouyanfar *et al.*, "A survey on deep learning: Algorithms, techniques, and applications," *ACM Comput. Surveys*, vol. 51, no. 5, pp. 1–36, Sep. 2018.

[55] J. Choo and S. Liu, "Visual analytics for explainable deep learning," *IEEE Comput. Graph. Appl.*, vol. 38, no. 4, pp. 84–92, Jul. 2018.

[56] H. Zhao, Q. Tian, L. Pan, and Y. Lin, "The technology of adversarial attacks in signal recognition," *Phys. Commun.*, vol. 43, Dec. 2020, Art. no. 101199.

[57] A. Graese, A. Rozsa, and T. E. Boult, "Assessing threat of adversarial examples on deep neural network," in *Proc. 15th IEEE Int. Conf. Mach. Learn. Appl.*, Anaheim, CA, USA, Dec. 2016, pp. 1–6.

Authorized licensed use limited to: Anglia Ruskin University (a.r.u.). Downloaded on February 19,2024 at 17:42:57 UTC from IEEE Xplore. Restrictions apply.

[58] S. A. Fezza, Y. Bakhti, W. Hamidouche, and O. Déforges, "Perceptual evaluation of adversarial attacks for CNN-based image classification," in *Proc. 11th Int. Conf. Qual. Multimedia Exp. (QoMEX)*, Berlin, Germany, Jun. 2019, pp. 1–6.
[59] Y. Dong *et al.*, "Boosting adversarial attacks with momentum," in *Proc. IEEE/CVF CVPR*, Salt Lake City, UT, USA, Jun. 2018, pp. 9185–9193.
[60] S. Horiguchi, D. Ikami, and K. Aizawa, "Significance of softmax-based features in comparison to distance metric learning-based features," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 5, pp. 1279–1285, May 2020.
[61] N. Papernot *et al. Technical Report on the CleverHans V2.1.0 Adversarial Examples Library*. Accessed: Aug. 2016. [Online]. Available: https://arxiv.org/abs/1610.00768.

Zhida Bao (Graduate Student Member, IEEE) received the B.S. degree in information engineering from Nanjing University of Aeronautics and Astronautics, Nanjing, China, in 2020. He is currently pursuing the Ph.D. degree with the College of Information and Communication Engineering, Harbin Engineering University, Harbin, China.

His research interests include communication technology, machine learning, and security analysis.

Yun Lin (Member, IEEE) received the B.S. degree from Dalian Maritime University, Dalian, China, in 2003, the M.S. degree from Harbin Institute of Technology, Harbin, China, in 2005, and the Ph.D. degree from Harbin Engineering University, Harbin, in 2010.

From 2014 to 2015, he was a Research Scholar with Wright State University, Dayton, OH, USA. He is currently a Full Professor with the College of Information and Communication Engineering, Harbin Engineering University. He has authored or coauthored more than 150 international peer-reviewed journal or conference papers, including the IEEE INTERNET OF THINGS, IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS, IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, IEEE TRANSACTIONS ON COGNITIVE COMMUNICATIONS AND NETWORKING, IEEE TRANSACTIONS ON RELIABILITY, IEEE ACCESS, INFOCOM, GLOBECOM, ICC, VTC, and ICNC. He has four high-cited papers and several best conference papers. His current research interests include machine learning and data analytics over wireless networks, signal processing and analysis, cognitive radio and software defined radio, artificial intelligence, and pattern recognition.

Prof. Lin is an Editor of the IEEE TRANSACTIONS ON RELIABILITY, *KSII Transactions on Internet and Information Systems*, and *International Journal of Performability Engineering*. In addition, he was the General Chair of the ADHIP 2020, the TPC Chair of the MOBIMEDIA 2020, ICEICT 2019, and ADHIP 2017, and a TPC Member of GLOBECOM, ICC, ICNC, and VTC. He had successfully organized several international workshops and symposia with top-ranked IEEE conferences, including INFOCOM, GLOBECOM, DSP, and ICNC.

Sicheng Zhang (Graduate Student Member, IEEE) received the B.S. degree in communication engineering from Harbin Engineering University, Harbin, China, in 2019, where he is currently pursuing the Ph.D. degree in information and communication engineering.

His current research interests include signal processing, physical layer security, machine learning, and data analysis.

Zixin Li (Graduate Student Member, IEEE) received the B.S. degree from the College of Information and Communication Engineering, Harbin Engineering University, Harbin, China, in 2021, where he is currently pursuing the M.S. degree in communication engineering.

His research interests include wireless communication, deep learning, and physical-layer security.

Shiwen Mao (Fellow, IEEE) received the Ph.D. degree in electrical engineering from Polytechnic University, Brooklyn, NY, USA, in 2004.

He is currently a Professor and the Earle C. Williams Eminent Scholar Chair of Electrical and Computer Engineering with Auburn University, Auburn, AL, USA. His research interest include wireless networks, multimedia communications, and smart grid.

Prof. Mao was a co-recipient of the 2021 IEEE Communications Society Outstanding Paper Award, the IEEE Vehicular Technology Society 2020 Jack Neubauer Memorial Award, the IEEE ComSoc MMTC 2018 Best Journal Paper Award and the 2017 Best Conference Paper Award, the Best Demo Award from IEEE SECON 2017, the Best Paper Awards from IEEE GLOBECOM 2019, 2016, and 2015, IEEE WCNC 2015, and IEEE ICC 2013, and the 2004 IEEE Communications Society Leonard G. Abraham Prize in the Field of Communications Systems. He is on the Editorial Board of *IEEE/CIC China Communications*, IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS, IEEE INTERNET OF THINGS JOURNAL, IEEE OPEN JOURNAL OF THE COMMUNICATIONS SOCIETY, *ACM GetMobile*, IEEE TRANSACTIONS ON COGNITIVE COMMUNICATIONS AND NETWORKING, IEEE TRANSACTIONS ON NETWORK SCIENCE AND ENGINEERING, IEEE TRANSACTIONS ON MOBILE COMPUTING, IEEE MULTIMEDIA, IEEE NETWORK, and IEEE NETWORKING LETTERS. He is a member of the ACM.

Authorized licensed use limited to: Anglia Ruskin University (a.r.u.). Downloaded on February 19,2024 at 17:42:57 UTC from IEEE Xplore. Restrictions apply.