# White-Box Adversarial Attacks on Deep Learning-Based Radio Frequency Fingerprint Identification

Jie Ma*, Junqing Zhang*, Guanxiong Shen*, Alan Marshall*, and Chip-Hong Chang†

* Department of Electrical Engineering and Electronics, University of Liverpool, Liverpool, L69 3GJ, United Kingdom
Email: {Jie.Ma, Junqing.Zhang, Guanxiong.Shen, Alan.Marshall}@liverpool.ac.uk
† School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore
Email: echchang@ntu.edu.sg

*Abstract*—Radio frequency fingerprint identification (RFFI) is an emerging technique for the lightweight authentication of wireless Internet of things (IoT) devices. RFFI exploits unique hardware impairments as device identifiers, and deep learning is widely deployed as the feature extractor and classifier for RFFI. However, deep learning is vulnerable to adversarial attacks, where adversarial examples are generated by adding perturbation to clean data for causing the classifier to make wrong predictions. Deep learning-based RFFI has been shown to be vulnerable to such attacks, however, there is currently no exploration of effective adversarial attacks against a diversity of RFFI classifiers. In this paper, we report on investigations into white-box attacks (non-targeted and targeted) using two approaches, namely the fast gradient sign method (FGSM) and projected gradient descent (PGD). A LoRa testbed was built and real datasets were collected. These adversarial examples have been experimentally demonstrated to be effective against convolutional neural networks (CNNs), long short-term memory (LSTM) networks, and gated recurrent units (GRU).

*Index Terms*—Adversarial Attack, Radio Frequency Fingerprint Identification, Deep Learning

## I. INTRODUCTION

Internet of things (IoT) devices are deployed widely in smart homes and cities, connected healthcare, and industry 4.0, etc. [1]. IoT networks may carry private, sensitive and/or confidential information, and device authentication is the first line of defence against spoofing attacks in an IoT network. Conventional authentication methods rely on cryptographic algorithms and software addresses such as media access control (MAC) addresses. However, cryptographic algorithms require a secure key distribution mechanism, which may not be affordable or practical for low-cost IoT devices [2]. Software addresses can be easily tampered with. Thus, there is a strong need for a lightweight IoT authentication scheme [3].

Radio frequency fingerprint identification (RFFI) is an emerging device authentication technique [4]. The transmitter chain of a wireless device consists of numerous analogue components such as mixers, oscillators, power amplifiers, etc. These components deviate slightly from their nominal specifications due to their inevitable manufacturing processes variations, such as IQ imbalance, carrier frequency offset (CFO), and PA non-linearity [4]. An RFFI system can ex-

tract and analyze the unique distortion induced by these imperfections onto the emitted waveforms to identify different devices [4]. The RFFI process is deployed at the receiver and requires no modification to the IoT end nodes. Machine learning based feature extractors have been used for RFFI as they are communication protocols agnostic and require no domain knowledge to train. Deep learning techniques [5]–[7], such as convolutional neural networks (CNNs) [6], [8], long-short term memory (LSTM) networks [8], gated recurrent units (GRU) [7] networks, and transformer [9], are particularly good at encoding discriminative data features in the latent space.

Despite their excellent classification performance [5]–[7], deep neural networks are prone to adversarial attacks, also known as evasion attacks [10]–[12]. It has been once and again demonstrated that a well-trained deep learning network can be deluded into incorrect prediction simply by adding maliciously constructed small perturbations to its input. The maliciously perturbed inputs, known as adversarial examples, can be made very subtle and imperceptible to the user or even automated inspection tools. A popularly cited example is the addition of visually undetectable noise in the picture of a panda to trick a deep neural network into misclassifying it as a gibbon [10]. Adversarial examples have been widely studied and have evolved rapidly in computer vision tasks but they have recently been extended to the wireless domain, e.g. the modulation classification attack reported in [13], [14].

To the best knowledge of the authors, there are only two investigative studies [12], [15] of adversarial examples on deep learning-based RFFI. In [12], the authors have examined the effects of adversarial examples on CNNs, but not other deep learning models used in RFFI. In [15], the evaluation of adversarial examples on a deep learning model is restricted to only I/Q samples. Deep learning models that are trained using other forms of signal representation have not been evaluated.

This paper aims to investigate the vulnerability of deep learning models in RFFI systems. Two attack methods were investigated, namely fast gradient sign method (FGSM) [10] and projected gradient descent (PGD) [11]. We built a long-range (LoRa) testbed and collected real datasets. Three deep learning models, namely CNN, LSTM, and GRU were built.
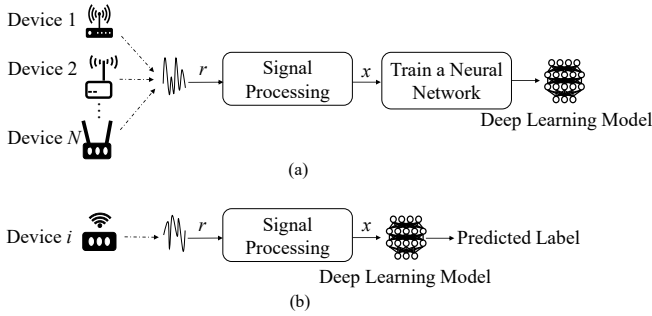
Fig. 1: RFFI overview (a) Training stage. (b) Inference stage.

Our main contributions are listed as follows.

- Adversarial examples produced by FGSM or PGD can severely degrade the identification performance. In a CNN-based RFFI system, 92.7% and 94.6% of devices were shown to be successfully misclassified by FGSM and PGD attacks, respectively.
- We experimentally corroborated that low quality signals are more vulnerable to evasion attacks. Under *SNR = 50* dB, 42.2% of the devices in RFFI using GRU were misclassified by PGD. When the SNR decreased to 20 dB, 91.3% of the devices were misclassified.
- We showed that a targeted attack by PGD is feasible. Superimposing elaborately generated perturbations onto the inputs can mislead the deep learning model to output a specific label set by the adversary. In the experiment of a targeted attack, up to 93.2% of packets are mistaken to be transmitted from the wrong target device.

The rest of the paper is organized as follows. Section II introduces the RFFI process. Adversarial examples for RFFI are presented in Section III. Section IV introduces the design details of the RFFI system and evaluates the effectiveness of adversarial examples on different deep learning models in the RFFI system. Section V concludes the paper.

## II. Radio Frequency Fingerprint Identification

A deep learning-based RFFI system is illustrated in Fig. 1. In the training stage, the deep learning model is trained with RF signals transmitted from $N$ known legitimate devices. In the inference stage, the trained model is used to predict which device has transmitted the signals it receives.

### A. Training

*1) Signal Processing:* The received signals, denoted as $r$, need to be processed according to RFFI requirements. The preprocessing usually includes packet detection, synchronization, normalization, CFO estimation and compensation. Detailed algorithms can be found in [6].

The captured IQ samples are complex numbers, which cannot be processed by most neural networks. They are transformed to a specific signal representation $x$ before being sent into the neural network for training. The signal representations used in previous work include FFT results [16], spectrogram [17], error signal [18], etc.
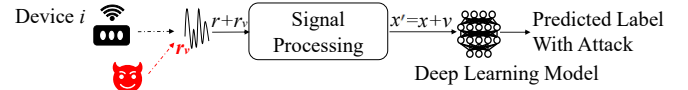


Fig. 2: Evasion attack on an RFFI system.

*2) Training:* A deep learning-based RFFI system needs to be adequately trained before deployment. In the training process, the parameters $\theta$ of the deep learning model $f(\cdot; \theta)$ are continuously updated with the following objective:

$$\theta = \operatorname*{argmin}_{\theta} \sum_{(x, l^{true}(x))} J(f(x; \theta), l^{true}(x)), \quad (1)$$

where $x$ is the input to the deep learning model and $f(x; \theta)$ is its corresponding output. $l^{true}(x)$ denotes the ground truth label of $x$. $J(\cdot)$ is the loss function.

### B. Inference

The inference stage is illustrated in Fig. 1(b). The received signal is processed by the same signal processing module as in the training stage. The transformed signal is fed into the trained deep learning model $f(\cdot; \theta)$. The output $f(x; \theta)$ is a probability vector over all the labels. The index $k$ of the element with the highest probability is selected as the predicted device label $l^x$ of the input $x$. The inference process can be mathematically expressed as

$$l^x = \operatorname*{argmax}_{k}(f(x; \theta)). \quad (2)$$

## III. Evasion Attacks on RFFI

### A. Threat Model

An evasion attack in the form of adversarial examples is mounted in the inference stage of RFFI. This paper considers white-box attacks, which assume an adversary has full knowledge of the victim, including training data, deep learning model, etc [19]. Therefore, the adversary employs the processed signal $x$ and model $f(x; \theta)$ to generate a perturbation $v$. The adversarial example, $x'$, can be written as

$$x' = x + v, \quad (3)$$

As shown in Fig. 2, adversarial examples are input to the pre-trained deep learning model to fool the model into making different predictions, given as

$$l^{x'} = \operatorname*{argmax}_{k}(f(x'; \theta)). \quad (4)$$

Evasion attacks can be categorized into non-targeted and targeted attacks based on the adversary's goal. In the non-targeted attack, the aim is to make the prediction of the adversarial example different from the original label, i.e.,

$$l^{x'} \neq l^x. \quad (5)$$

The goal of the targeted attack is to have all adversarial examples predicted to be the target label, i.e.,

$$l^t == l^{x'}, \quad (6)$$

where $l^t$ is the target label set by the adversary. '==' denotes verifying whether the equation on the left equals the right one.

## B. Metric

Perturbation to Signal Ratio (PSR) is the ratio of the power of the perturbation, $P_v$ to the power of the original signal, $P_x$. PSR can be expressed mathematically as:

$$PSR = 10 \log_{10}(\frac{P_v}{P_x}). \qquad (7)$$

Success Rate (SR) is defined as the probability of success for an adversarial attack. For the non-targeted attack, it can be evaluated by:

$$SR = \frac{\sum_{i=1}^{N} \zeta\left(l^{x'_i} \neq l^{x_i}\right)}{N}, \qquad (8)$$

where $N$ is the total number of test samples, $\zeta(\cdot)$ is an indicator function that returns one if its argument is true (attack successfully) and zero otherwise. For targeted attacks with the target label $l^t$, SR can be evaluated by:

$$SR = \frac{\sum_{i=1}^{N} \zeta\left(l^{x'_i} == l^t\right)}{N}, \qquad (9)$$

## C. Adversarial Examples Generation Methods

Among the white-box adversarial examples generation methods, Fast Gradient Sign Method (FGSM) and its iterative version, Projected Gradient Descend (PGD) method, are most popular and are bases for other improved evasion attacks.

*1) Fast Gradient Sign Method (FGSM):* FGSM is a non-iterative approach to produce perturbations that can reliably cause a wide variety of deep neural networks to malfunction with a single projected gradient step (one-step). It can be designed as targeted and non-targeted attacks [10]. FGSM generates perturbations by calculating the gradient of the loss function relative to the neural network input. Only the gradient sign is used by FGSM, not the whole gradient information.

Specifically, the adversarial examples for a non-targeted attack are constructed by

$$x' = x + \varepsilon \cdot \text{sign}\left(\nabla_x J(f(x;\theta), l^x)\right), \qquad (10)$$

and adversarial examples for targeted attacks are generated by

$$x' = x - \varepsilon \cdot \text{sign}\left(\nabla_x J(f(x;\theta), l^t)\right), \qquad (11)$$

where $\nabla_x$ indicates the gradient of the model for an original sample $x$ with the label $l$, $\varepsilon$ denotes a control parameter for the amplitude of perturbation, $\text{sign}(\cdot)$ indicates the gradient direction. $\text{sign}(\cdot)$ returns 1 (-1) if the value of the gradient direction is greater (smaller) than 0.

*2) Projected Gradient Descent (PGD):* PGD is an iterative method with random initialization to produce adversarial examples. It iteratively applies FGSM and projects the perturbed sample back to the norm multiple times with step size in the scale of the total perturbation bound. PGD is more powerful than FGSM because it mitigates the underfitting adversarial example due to the linear approximation of the decision boundary around the data point. The procedure of PGD is given in Algorithm 1, where $\|\cdot\|_p$ is the norm of the perturbation (such as $L_2$ and $L_\infty$). $\text{Clip}_{x,\alpha}(\cdot)$ denotes clipping the argument to the range $[x - \alpha, x + \alpha]$.

---

**Algorithm 1:** PGD Algorithm

**Input:** signal $x$, label $l$ or $l^t$, loss function $J(\cdot)$, size of the perturbation $\varepsilon$, the size on each iteration step $\alpha$, number of iteration $I$

**Output:** An adversarial example $x'_I$ with
$\|x'_I - x\|_p \leq \varepsilon$

1 Initialize $x'_0 = x$, $i = 1$
2 **for** *i in range I* **do**
3    **if** *Non-targeted attack* **then**
4       $x'_i =$
      $\text{Clip}_{x,\alpha}\left\{x'_{i-1} + \alpha \cdot \text{sign}\left(\nabla_x J(f(x;\theta), l^x)\right)\right\}$
5    **else if** *Targeted attack* **then**
6       $x'_i =$
      $\text{Clip}_{x,\alpha}\left\{x'_{i-1} - \alpha \cdot \text{sign}\left(\nabla_x J(f(x;\theta), l^t)\right)\right\}$
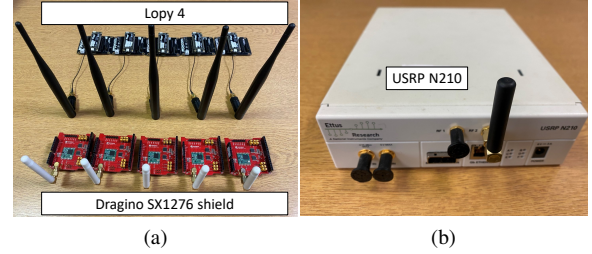
7 **return** $x'_I$

---



Fig. 3: Experimental devices. (a) Transmitter: LoRa. (b) Receiver: USRP.

## IV. EXPERIMENTAL EVALUATION

### A. Setup

*1) Datasets:* LoRa is used as a case study in this paper. Eight preambles at the start of each LoRa packet are captured for RFFI. The IQ samples are then converted into channel-independent spectrograms, which are in the time-frequency domain and robust to the channel variation. We refer interested readers to [17] for detailed derivation.

As shown in Fig. 3(a), ten LoRa devices, consisting of five LoPy4 and five Dragino SX1276 shields, are used. To distinguish these ten devices, we label them as Device 1, Device 2 ... Device 10, respectively. These devices are configured with bandwidth $B = 125$ kHz and carrier frequency $f_c = 868.1$ MHz. The receiver end is a USRP N210 software-defined radio (SDR) platform, as shown in Fig. 3(b), whose sampling rate $f_s = 1$ MHz. The Communications Toolbox Support Package for USRP Radio of MATLAB R2021b is used for accessing data from USRP. The packets are collected in an office environment and the SNR is about 70 dB. The datasets contain 1000 packets from each device. For each device, 90% and 10% of the packets are used for training and testing, respectively.

*2) Models:* We apply adversarial attacks to three popular deep learning models, namely CNN, LSTM, and GRU, whose
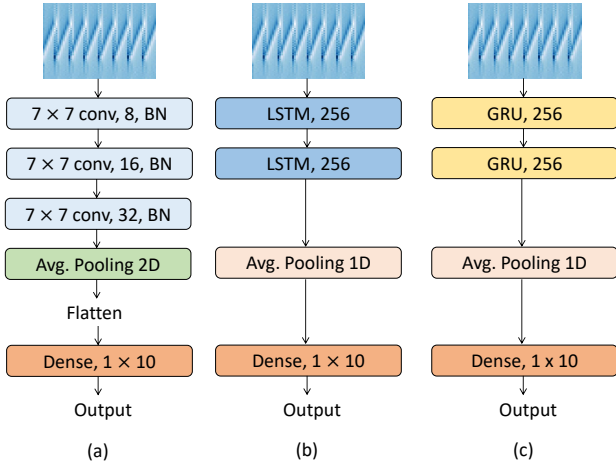
Fig. 4: Three deep learning network architectures: (a) CNN. (b) LSTM network. (c) GRU network.



Fig. 5: (a) Channel independent spectrogram of clean sample, $x$. (b) Perturbation calculated by PGD, $v$. (c) Adversarial example, $x'$.

architectures are shown in Fig. 4.

- CNN (Fig. 4(a)): it consists of one convolutional layer of 8 $7 \times 7$ kernels, followed by one convolutional layer of 16 $7 \times 7$ kernels and one convolutional layer of 32 $7 \times 7$ kernels, one 2D average pooling layer, and one dense layer activated by the softmax function.
- LSTM (Fig. 4(b)): it consists of two 256-unit LSTM layers, one global 1D average pooling layer, and one softmax activated dense layer.
- GRU (Fig. 4(c)): it is similar to the LSTM (Fig. 4(b)), but the LSTM layers are replaced by the GRU layers.

We use a prefix-suffix notation "*A-B*" to denote an adversarial example generation method *A* mounted on a deep learning model *B*. For example, "FGSM-CNN" refers to the attack on a CNN-based RFFI system using the adversarial examples generated by FGSM.

*3) Training Configuration:* The deep learning training was carried out on a PC equipped with a GPU of NVIDIA GeForce GTX 1660Ti. The neural networks are implemented with the TensorFlow library. FGSM and PGD are implemented based on CleverHans.v3.1.0 [20], which is a software library to benchmark machine learning systems' vulnerability to adversarial examples. The standardized reference implementations of attacks are provided in this library.

### B. Comparison Between Clean Sample and Adversarial Example

A channel-independent spectrogram of a clean sample, its corresponding perturbation and adversarial example are illustrated in Fig. 5.

- Fig. 5(a) is the channel-independent spectrogram of one sample in the original datasets. We can observe eight repeating upchirps from Fig. 5(a), and these upchirps contain unique features of the device.
- Fig. 5(b) is the perturbation calculated by PGD based on the spectrogram shown in Fig. 5(a).
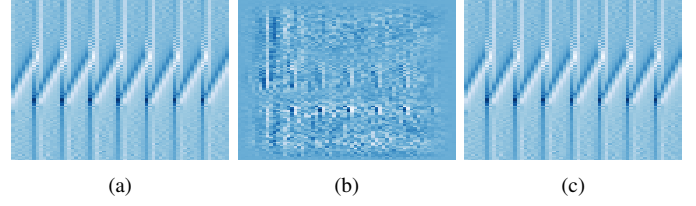- Fig. 5(c) is the corresponding adversarial example.

Visually we can barely distinguish the spectrogram of the clean sample and adversarial example in Fig. 5. Therefore, an objective metric is required to quantify the distortion of the sample and the effectiveness of the attack. In the adversarial example generation algorithms, the maximum perturbation of the sample is controlled by $\varepsilon$ ((10), (11) and Algorithm 1). In this paper, we use PSR to assess the power of perturbation signal, and then we measure the success rate SR of different algorithms. PSR and SR enable an objective quantitative evaluation of the effectiveness of using different adversarial attack methods to spoof a wireless communication receiver.

### C. Non-targeted Attacks With Different Perturbations

To demonstrate the effectiveness of different adversarial examples, we compare them with additive white Gaussian noise (AWGN) which is a common natural disturbance in RFFI systems. We show the confusion matrices for different perturbations and mounted FGSM and PGD on RFFI to compare their variations in SR against PSR.

Fig. 6(a) shows the confusion matrix of the CNN-based RFFI without attack and the classification accuracy reaches 98.4%. Fig. 6(b) shows the confusion matrix of non-targeted attack produced by PGD when $PSR = -35$ dB and the overall accuracy reduces to 1.4%. As it is a non-targeted attack, the misclassification results are dispersed across all labels.

Fig. 7 shows the SR of non-targeted attacks when PSR changes from $-45$ dB to $-20$ dB. Deep learning-based RFFI systems suffer more severe misclassifications when PSR is higher. Specifically, when the PSR is $-40$ dB, The SRs of CNN-PGD RFFI and CNN-FGSM RFFI reach 94.6% and 92.7%, respectively. The SRs of all the adversarial attacks continue to increase with increasing PSR. Also, the iterative PGD is found to be more effective than the one-step FGSM.

Fig. 7 shows that the same power of AWGN, when injected into the input signal, seldom has any impact on the performance of these RFFI classifiers. Apparently, while RFFI classifiers are robust against AWGN, they are vulnerable to adversarial examples as evinced by their significantly compromised prediction accuracy under these attacks.

### D. Non-targeted Attacks Under Different SNRs

The attacks in the previous sections assume good quality datasets are acquired under a high SNR. However, in a real
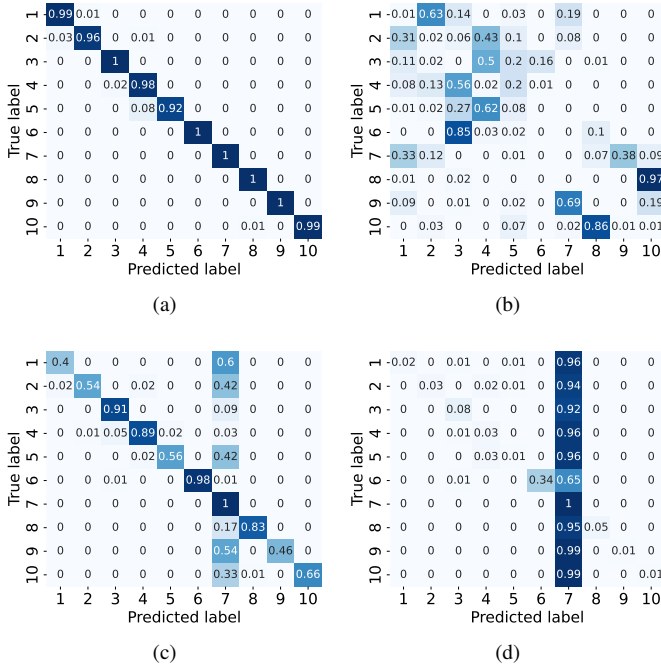
Fig. 6: Confusion matrix. (a) RFFI without adversarial attack. (b) CNN-PGD with non-targeted attacks. $PSR = -35$ dB. (c) CNN-PGD with targeted attacks. $PSR = -30$ dB. (d) CNN-PGD with targeted attacks. $PSR = -25$ dB.
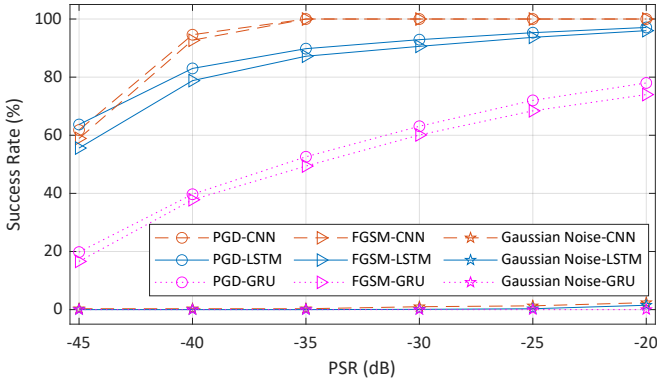
Fig. 7: SR of non-targeted attacks under different PSRs.

Fig. 8: SR of non-targeted attacks at different SNRs. $PSR = -40$ dB.

RFFI system, AWGN will reduce the SNR. Its effect on the quality of the datasets cannot be ignored. Taking AWGN into consideration, $x'$ is expressed in (12).

$$x' = x + n + v, \tag{12}$$

where $n$ is the AWGN.

The former results were obtained under $SNR = 70$ dB, which means the quality of the datasets is high. In this section, we emulate different SNRs by adding artificial AWGN into the original clean datasets.

Fig. 8 presents the SRs of different models and adversarial attack methods against SNRs. It can be observed that the effectiveness of all evasion attacks decreases at different ra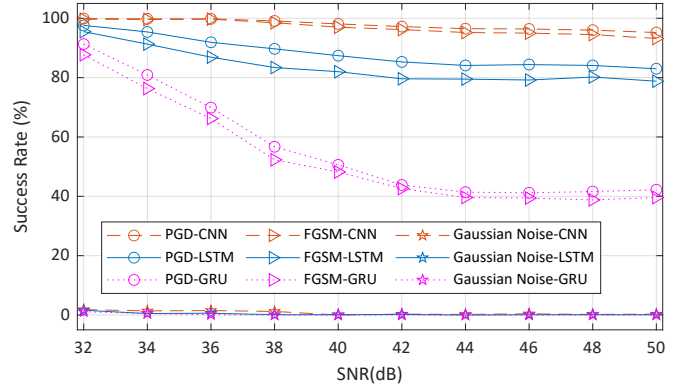tes with increasing SNR. Taking FGSM-GRU as an example, the SR is about $87.7\%$ when $SNR = 32$ dB but it reduces to about $39.6\%$ when $SNR = 50$ dB. The adversarial examples are much more likely to succeed when the SNR is low, regardless of the deep learning model. It implies that an active adversary can monitor the received signal strength and channel characteristics and seize a good opportunity to mount the attack to increase the attack success rate.

### E. Targeted Attacks

PGD is utilized to implement targeted attacks. We arbitrarily select Device 7 as the target. Fig. 6(c) shows the confusion matrix of PGD-CNN with the targeted attack at $PSR = -30$ dB, where $36.1\%$ of the devices are misclassified as Device 7. When the PSR increases to $-25$ dB, $93.2\%$ of the devices are misclassified as Device 7, as shown in Fig. 6(d). A successful targeted attack is hence demonstrated. The effectiveness of the targeted attack can be further enhanced by higher PSR to cause more devices to be misclassified as Device 7.

To investigate the performance of targeted attacks against different targets, we set each class as the target one by one. The results are shown in Fig. 9. Unlike a confusion matrix, each matrix element in Fig. 9 represents the percentage of devices identified as the selected target. For example, $80\%$ packets from Device 6 are identified as originating from Device 5 in Fig. 9(a) when the selected target is Device 5. Apparently, PGD can cause the victim model to misclassify most devices as the selected target indiscriminately, which indicates the effectiveness of PGD for targeted attacks on deep learning-based RFFI.

## V. CONCLUSION

In this paper, we presented white-box evasion attacks on deep learning-based RFFI. Specifically, we studied non-targeted and targeted attacks using two adversarial example generation methods, namely FGSM and PGD. A LoRa-RFFI testbed was built and real datasets were collected. The attacks were applied to CNN, LSTM and GRU-based RFFI systems. The results reveal that deep learning-based RFFI systems are vulnerable to adversarial attacks. Compared with Gaussian noise, the carefully crafted perturbation can completely cripple
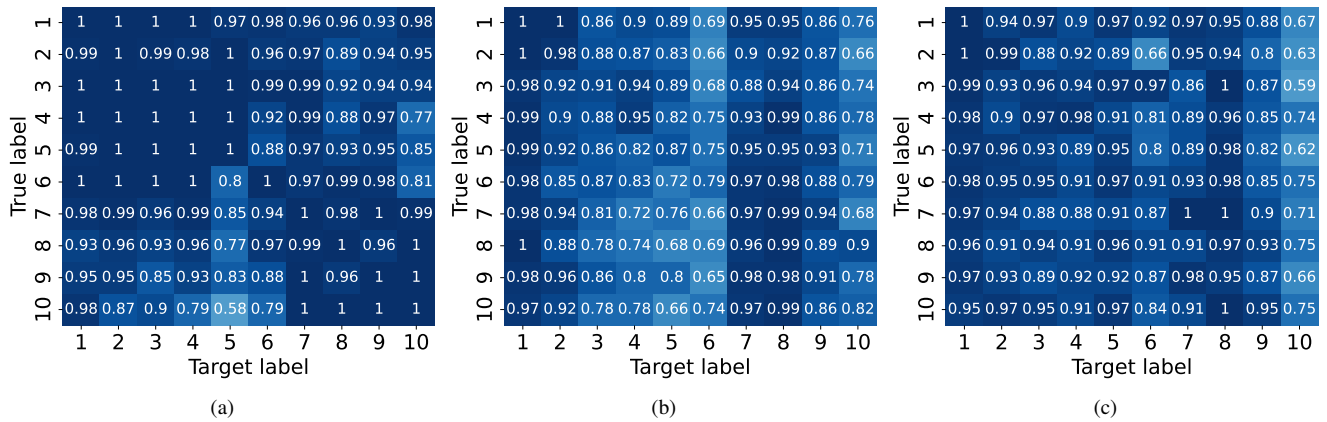
Fig. 9: Targeted attacks by PGD on (a) CNN, (b) LSTM, and (c) GRU-based RFFI, with $PSR = -5$ dB.

the RFFI. The iterative method PGD was demonstrated to be more effective than the one-step method FGSM, regardless of SNR. In short, the successes of both targeted and non-targeted attacks signify a real threat to the use of modern RFFI for physical layer security. In a white-box attack, the adversary has all the knowledge of the victim and is able to calculate the accurate gradient to result in a significant security problem. In our future work, we will investigate more practical black-box or grey-box attacks where the adversary has no knowledge about the victim's deep learning model. We will also develop lightweight countermeasures against these successful evasion attacks.

### REFERENCES

[1] S. B. Baker, W. Xiang, and I. Atkinson, "Internet of Things for smart healthcare: Technologies, challenges, and opportunities," *IEEE Access*, vol. 5, pp. 26 521–26 544, Nov. 2017.

[2] J. Zhang, G. Li, A. Marshall, A. Hu, and L. Hanzo, "A new frontier for IoT security emerging from three decades of key generation relying on wireless channels," *IEEE Access*, vol. 8, pp. 138 406–138 446, 2020.

[3] V. Hassija, V. Chamola, V. Saxena, D. Jain, P. Goyal, and B. Sikdar, "A survey on iot security: application areas, security threats, and solution architectures," *IEEE Access*, vol. 7, pp. 82 721–82 743, 2019.

[4] J. Zhang, R. Woods, M. Sandell, M. Valkama, A. Marshall, and J. Cavallaro, "Radio frequency fingerprint identification for narrowband systems, modelling and classification," *IEEE Trans. Inf. Forensics Security*, vol. 16, pp. 3974–3987, Jun. 2021.

[5] T. Jian, B. C. Rendon, E. Ojuba, N. Soltani, Z. Wang, K. Sankhe, A. Gritsenko, J. Dy, K. Chowdhury, and S. Ioannidis, "Deep learning for RF fingerprinting: A massive experimental study," *IEEE Internet Things Mag.*, vol. 3, no. 1, pp. 50–57, 2020.

[6] G. Shen, J. Zhang, A. Marshall, L. Peng, and X. Wang, "Radio frequency fingerprint identification for LoRa using spectrogram and CNN," in *Proc. IEEE Int. Conf. Comput. Commun. (INFOCOM)*, Vancouver, BC, Canada, Jul. 2021, pp. 1–10.

[7] D. Roy, T. Mukherjee, M. Chatterjee, E. Blasch, and E. Pasiliao, "RFAL: Adversarial learning for RF transmitter identification and classification," *IEEE Trans. on Cogn. Commun. Netw.*, vol. 6, no. 2, pp. 783–801, 2019.

[8] G. Shen, J. Zhang, A. Marshall, L. Peng, and X. Wang, "Radio frequency fingerprint identification for LoRa using deep learning," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 8, pp. 2604–2616, 2021.

[9] G. Shen, J. Zhang, A. Marshall, M. Valkama, and J. Cavallaro, "Radio frequency fingerprint identification for security in low-cost IoT devices," in *Proc. Asilomar Conference on Signals, Systems, and Computers*, 2021.

[10] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," in *Proc. Int. Conf. Lrn. Repr. (ICLR)*, San Diego, CA, USA, May 2015.

[11] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," in *Proc. Int. Conf. Lrn. Repr. (ICLR)*, Vancouver Convention Center, Vancouver, BC, Canada, Nov. 2018.

[12] 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.*, vol. 9, no. 11, pp. 9012–9024, 2021.

[13] B. Kim, Y. E. Sagduyu, K. Davaslioglu, T. Erpek, and S. Ulukus, "Channel-aware adversarial attacks against deep learning-based wireless signal classifiers," *IEEE Trans. Wireless Commun.*, 2021.

[14] 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, 2019.

[15] F. Restuccia, S. D'Oro, A. Al-Shawabka, B. C. Rendon, K. Chowdhury, S. Ioannidis, and T. Melodia, "Generalized wireless adversarial deep learning," in *Proceedings of the 2nd ACM Workshop on Wireless Security and Machine Learning*, 2020, pp. 49–54.

[16] A. Al-Shawabka, F. Restuccia, S. D'Oro, T. Jian, B. C. Rendon, N. Soltani, J. Dy, S. Ioannidis, K. Chowdhury, and T. Melodia, "Exposing the fingerprint: Dissecting the impact of the wireless channel on radio fingerprinting," in *Proc. IEEE Int. Conf. Comput. Commun. (INFOCOM)*, Toronto, ON, Canada, Jul. 2020, pp. 646–655.

[17] G. Shen, J. Zhang, A. Marshall, and J. R. Cavallaro, "Towards scalable and channel-robust radio frequency fingerprint identification for LoRa," *IEEE Trans. Inf. Forensics Security*, vol. 17, pp. 774–787, 2022.

[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, Jan. 2018.

[19] 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, Feb. 2019.

[20] N. Papernot, F. Faghri, N. Carlini, and I. Goodfellow., "Technical report on the cleverhans v2.1.0 adversarial examples library," *arXiv preprint arXiv:1610.00768*, 2018.