Explanation-Guided Backdoor Attacks on Model-Agnostic RF Fingerprinting

Tianya Zhao*, Xuyu Wang*§, Junqing Zhang†, Shiwen Mao‡
*Knight Foundation School of Computing and Information Sciences, Florida International University, Miami, FL 33199, US
†Department of Electrical Engineering and Electronics, University of Liverpool, Liverpool, L69 3GJ, United Kingdom
‡Department of Electrical and Computer Engineering, Auburn University, Auburn, AL 36849, US
Emails: tzhao010@fiu.edu, xuywang@fiu.edu, junqing.zhang@liverpool.ac.uk, smao@ieee.org

IEEE INFOCOM 2024 - IEEE Conference on Computer Communications | 979-8-3503-8350-8/24/$31.00 ©2024 IEEE | DOI: 10.1109/INFOCOM52122.2024.10621289

***Abstract*—Despite the proven capabilities of deep neural networks (DNNs) for radio frequency (RF) fingerprinting, their security vulnerabilities have been largely overlooked. Unlike the extensively studied image domain, few works have explored the threat of backdoor attacks on RF signals. In this paper, we analyze the susceptibility of DNN-based RF fingerprinting to backdoor attacks, focusing on a more practical scenario where attackers lack access to control model gradients and training processes. We propose leveraging explainable machine learning techniques and autoencoders to guide the selection of positions and values, enabling the creation of effective backdoor triggers in a model-agnostic manner. To comprehensively evaluate our backdoor attack, we employ four diverse datasets with two protocols (Wi-Fi and LoRa) across various DNN architectures. Given that RF signals are often transformed into the frequency or time-frequency domains, this study also assesses attack efficacy in the time-frequency domain. Furthermore, we experiment with potential defenses, demonstrating the difficulty of fully safeguarding against our attacks.**

***Index Terms*—Backdoor Attack, Radio Frequency Fingerprinting, Explainable Machine Learning, Security.**

I. Introduction

The Internet of Things (IoT) has become increasingly popular in recent years, with wireless technology being integrated into more and more aspects of our daily lives. As the number of wireless devices grows, an effective and efficient device authentication method is essential [1]–[3]. Radio frequency (RF) fingerprinting has become a promising technique for authenticating RF devices because it is more difficult to tamper and spoof compared to traditional methods [4], [5].

RF fingerprints are unique properties generated by inherent physical imperfections in the analog circuitry of RF emitters during the manufacturing process [6], [7]. These imperfections affect the transmitted signals slightly, but they do not affect the overall performance of the devices. As a result, every RF emitter has a unique fingerprint, including ultra-low-power devices and legacy ones. With the widespread use of deep learning, fingerprints can be automatically extracted and classified by deep neural networks (DNNs). Specifically, due to the excellent performance on feature extraction, many RF fingerprinting systems deploy convolutional neural networks (CNNs) to extract RF fingerprint features and classify different devices [8]–[13].

§The corresponding author is Xuyu Wang (xuywang@fiu.edu).

While DNNs offer powerful capabilities for RF fingerprinting, they also bring inherent vulnerabilities, including susceptibility to evasion and backdoor attacks [14]–[18]. Recent studies have explored the adverse impacts of these attack techniques in relevant domains. For example, Moosavi-Dezfooli *et al.* propose a universal perturbation that can fool DNNs on any image in the computer vision domain [19]. Nevertheless, evasion attacks typically involve an iterative process of perturbing the input sample based on gradients derived from the target model [20]. This iterative nature results in a significant computational burden for attackers. Therefore, backdoor attacks become a hot topic since it is more robust and practical from the perspective of attackers. In today's deep learning landscape, the integration of diverse cloud platforms, pre-trained models, and public datasets has become essential. Nonetheless, ensuring the security of such resources presents significant challenges. Malicious attackers can introduce problematic datasets and pre-trained models, thereby compromising the performance of inference tasks. Furthermore, attackers can invade the cloud infrastructure and manipulate loss functions during the training process to disrupt model performance. Given these circumstances, backdoor attacks can be classified into three main types: poisoning-based backdoor attacks, weights-oriented backdoor attacks, and structure-modified backdoor attacks [21]. For instance, BadNets is one of the first backdoor attacks that employ a visible trigger to deceive DNNs. The presence of this trigger in an image causes it to be incorrectly classified into a target class predefined by the attacker [15].

While extensive research has focused on backdoor attacks across various domains, there is limited analysis of the security vulnerabilities of deep learning-based RF fingerprinting systems. Given that RF fingerprinting enables device identification and impacts the security of broader applications, it is crucial to investigate potential backdoor threats targeting DNN-based RF fingerprinting. Therefore, this paper examines backdoor attacks on DNN-based RF fingerprinting to address the significance of understanding the security risks posed to these safety-critical systems.

Challenges. Implementing backdoor attacks on RF fingerprinting systems poses several challenges. First, these systems are crucial for security purposes, prompting system providers to incorporate robust protections. Existing powerful backdoor

attacks typically involve creating a trigger generator that requires accessing gradient information or modifying loss functions. However, in the case of high-level security systems like RF fingerprinting, it would be impractical for them to expose gradient information. Therefore, designing a powerful trigger without knowing and manipulating the gradient in the training process is a challenging task. Second, while backdoor attacks have been thoroughly investigated in the image domain, it is important to note that triggers designed for images may not be applicable or effective for RF signals. DNN-based RF fingerprinting typically uses in-phase/quadrature (I/Q) samples in the time domain as input data, which are fundamentally different from images. Therefore, a different approach is needed when considering backdoor attacks in RF fingerprinting. Third, the added trigger should not significantly impact the system's performance and be resistant to a certain level of defense methods. This poses a unique challenge for RF fingerprinting systems since input I/Q data often undergoes signal processing, transforming it into the frequency or time-frequency domain. This requires the trigger to be effective in both the time domain and the frequency domain.

Solution. To address the above challenges, we propose a practical backdoor attack for DNN-based RF fingerprinting by only poisoning some training data, without controlling other training components (*e.g.,* loss function and model structure). Our method leverages the concept of black-box adversarial attacks, where we construct a surrogate model due to the inherent limitations in manipulating the training process. Then, we employ LIME (Local Interpretable Model-agnostic Explanations) [22] on the surrogate model to identify the important areas that will guide us to place the trigger in a strategic location. The encoder component of an autoencoder is used to generate feature values for the backdoor trigger. To evaluate the effectiveness of the backdoor attack, we conduct a thorough study of its performance under various trigger detection and defense strategies. This comprehensive assessment allows us to validate the robustness and effectiveness of our approach. The main contributions of this paper are as follows.

- To the best of our knowledge, this is the first work to investigate backdoor attacks on RF fingerprinting. We develop a practical model-agnostic trigger generation method without the need for access to gradients and additional training components.
- We deploy an eXplainable Artificial Intelligence (XAI) approach and an autoencoder to generate the backdoor trigger that satisfies the realistic adversarial constraints. We then explore the effectiveness of this trigger in both the time and frequency domains.
- We evaluate trigger detection and mitigation techniques, demonstrating that our backdoor trigger is stealthy and difficult to fully defend against.
- We conduct extensive experiments across four different datasets and three different model architectures to validate our proposed backdoor attack. The results show that our attack can achieve over 97% success rate for most cases.

The rest of the paper is organized as follows. Section II discusses the related work and Section III introduces background on LIME. Section IV provides the problem statement and threat model. LIME-guided backdoor attacks are elaborated in Section V. Section VI presents the experimental evaluations and analysis. Finally, Section VII concludes this paper.

II. Related Work

Previous works have studied backdoor attacks in a wide range of applications. In the image domain, BadNet [15] first demonstrates the vulnerability of DNNs by embedding a visible trigger onto the lower right corner of the image. Chen *et al.* [23] first discuss stealthy triggers, enabling poisoned inputs to evade human inspection. In addition to the above backdoor attacks that require adding triggers and modifying labels, clean-label attacks that can attack DNNs without modifying labels are also discussed [24]–[27]. Besides, [28] crafts triggers by training a separate generator model, which requires the ability to manipulate the whole training process. To restrict attacker capabilities, Li *et al.* [29] adopt techniques from image steganography to generate sample-specific triggers without knowing the gradient information. Zeng *et al.* [30] first examine image backdoor attacks from a frequency perspective, demonstrating the importance of the frequency domain in designing both attacks and defenses.

Except for the image domains, backdoor attacks also be demonstrated as a threat to DNNs in other domains. For instance, [31] proposes backdoor attacks on wireless traffic prediction in both centralized and distributed training scenarios. Jiang *et al.* [32] employ generative adversarial networks (GAN) to create backdoored time series data. TrojanFlow [33] implements attacks on network traffic classification by simultaneously optimizing a trigger generator and the target model. However, training this generator requires manipulation of gradients and the loss function during the training process, necessitating significant adversary capability. Severi *et al.* [34] employ an XAI tool to guide clean-label backdoor attacks against malware classifiers. In fact, their approach does not utilize powerful DNNs to generate trigger values, which may limit the performance of the attack.

There are several key distinctions between our work and related research. First, I/Q data for RF fingerprinting is a two-dimensional stream in the time domain. Traditional triggers designed for static images may not be feasible in this case. Second, it is natural to consider the effectiveness of backdoor attacks in the time-frequency domain, since time-domain I/Q data may be processed by short-time Fourier transform (STFT). Third, to ensure the practicality of our attack in real-world scenarios, we impose restrictions on the adversary's capabilities. Specifically, the attacker can only tamper with a small portion of training data while not having access to the gradient or the ability to modify the training method.

We note that Zhao *et al.* [35] recently proposed backdoor attacks on RF signal classification, which is concurrent with ours. However, they did not consider the XAI tool to guide attacks and frequency domain attacks.

III. Background: LIME

While DNNs achieve impressive performance across many domains, their black-box nature makes their internal decision processes opaque and difficult to interpret. XAI techniques have thus emerged to provide humans with insight into how these complex models arrive at predictions. LIME is one of the first model-agnostic XAI methods to locally interpret complex black-box DNNs with an interpretable model (e.g., linear regression). To achieve this, LIME approximates the DNN f_θ by generating a series of perturbations for a given sample x, represented as $x_1^{'}, x_2^{'}, ..., x_p^{'}$, where certain feature values in x are randomly set to 0. Then, these perturbed samples are fed to the DNN to obtain predicted labels. With the perturbed samples and corresponding labels, LIME trains a weighted linear regression to approximate the decision boundary as

$$\arg\min_{g \in G} \sum_{i=1}^{p} \pi_x(x_i^{'})(f_\theta(x_i^{'}) - g(x_i^{'}))^2, \tag{1}$$

where G denotes the set of interpretable models such as linear regression and decision trees. The proximity measure $\pi_x(x_i^{'}) = exp(-d(x, x_i^{'})^2/\sigma^2)$ calculates the similarity between the original input x and each perturbed input $x_i^{'}$ based on a distance function $d(\cdot,\cdot)$. In this paper, we employ cosine distance as the distance function.

IV. Problem Statement and Threat Model

In this section, we first describe the problem statement and then introduce the threat model.

A. Problem Definition

The typical backdoor attack pipeline for DNN-based RF fingerprinting is illustrated in Fig. 1. Users first upload and store I/Q samples in the cloud. At this stage, attackers may inject triggers into a small portion of the data. The user then specifies the model architecture, loss function, and decides whether signal processing is necessary. While the attacker cannot access model gradients or training, poisoning the sub-dataset D_s ensures a backdoor is embedded into the model after training completes. As a result, the model maintains normal accuracy for regular samples but produces predictions for the target class when dealing with samples containing the backdoor trigger.

B. Threat Model

1) Adversary's Goal: This paper focuses on targeted backdoor attacks, where the adversary aims to induce misclassifications to a specific target class. The attacker's goal is to obtain a backdoored classifier f_b by poisoning a fraction of the training data. An optimal backdoored classifier f_b would behave identically to a clean classifier f on unperturbed inputs $\mathbf{x}$, but generate adversary-chosen predictions y_t on backdoored inputs $\mathbf{x}_b$ as:

$$f_b(\mathbf{x}) = f(\mathbf{x});\ f_b(\mathbf{x}_b) = y_t \neq f(\mathbf{x}_b). \tag{2}$$

In addition to the attack *effectiveness* mentioned above, the attacker has two other goals: *stealthiness* and *robustness*.

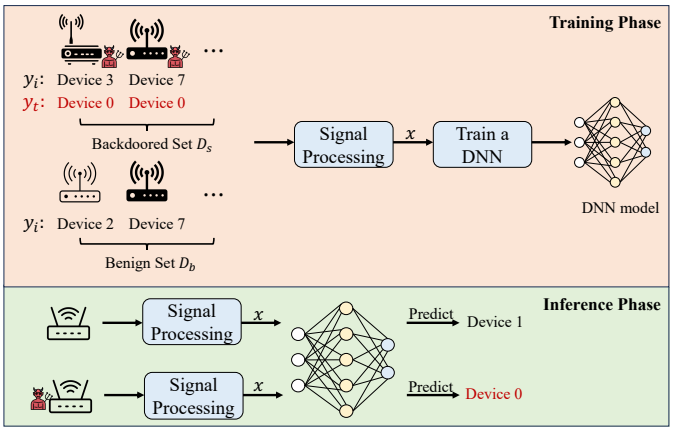

Fig. 1. Overview of the backdoor attack pipeline for DNN-based RF fingerprinting.

Stealthiness demands the trigger remains concealed and undetectable, ensuring that it does not raise suspicion or attract attention. Robustness ensures that the backdoor attack remains effective even when faced with certain defensive mechanisms.

2) Adversary's Capability: In recent years, the widespread use of cloud platforms, pre-trained models, and public datasets has become essential to various workflows. However, for security-critical RF fingerprinting systems, it is vital to limit adversary capabilities. In this paper, we assume that attackers are only permitted to poison some training data, but they have no access to or control over other training components such as the training loss and model architecture. During the inference stage, attackers can only query the trained model with poisoned data but cannot access the model's internal information or the inference process. Limiting adversary's knowledge and control in this manner reflects realistic threat models applicable to many real-world scenarios, given the prevalence of cloud resources.

V. LIME-Guided Backdoor Attacks

A. Overview

In this paper, we design backdoor attacks on general RF fingerprinting systems under restricted attack capabilities. Let $D_{train} = \{(\mathbf{x}_i, y_i)\}_i^N$ denote the training dataset containing N samples, where each $\mathbf{x}$ represents 2×256 raw I/Q data and y_i is the corresponding device category. To build a poisoned training set D_p, attackers need to inject triggers into a small subset D_s of the full training dataset D_{train}. This portion is defined as poisoning rate $\gamma \doteq \frac{|D_s|}{|D_{train}|}$. The poisoned training set D_p consists of this backdoored data D_s and the remaining benign data D_b, i.e., $D_p = D_s \cup D_b$.

After constructing the poisoned training set D_p, the backdoored DNN f_θ is trained on this dataset. We assume users follow a standard supervised learning approach, training the RF fingerprinting model by minimizing its classification error on D_p, formulated as:

$$\min_\theta \mathbb{E}_{(\mathbf{x},y)\in D_p} \mathcal{L}_{CE}(f_\theta(\mathbf{x}), y), \tag{3}$$

where $\mathcal{L}_{CE}$ represents the widely used cross-entropy loss and θ denotes the model parameters. Before deploying the model in the real world, the user typically evaluates performance on a separate benign set. As illustrated in Section IV-B1, the backdoored model must perform properly on clean samples in order to deceive the user.

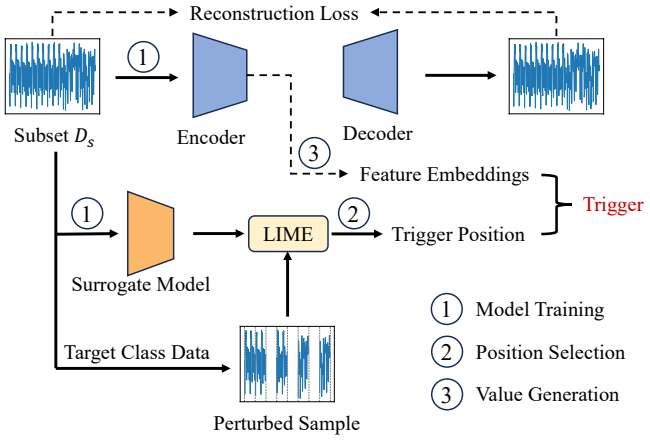

Fig. 2. Overview of our proposed trigger generation.

The most critical step in the above attack procedure is generating effective backdoor triggers under the limited adversary capabilities. Since we preclude the attacker from leveraging gradients, we employ both a surrogate model and autoencoder to craft triggers using only the small poisoned subset D_s, as illustrated in Fig. 2. Step 1 involves training the surrogate model and autoencoder for future trigger generation. Step 2 selects the trigger placement position, and step 3 determines the corresponding trigger values.

B. Trigger Position Selection

I/Q samples, encompassing both real and imaginary parts in the time domain, are collected from diverse protocols, modulation techniques, and can even be influenced by environmental factors. This characteristic makes I/Q data less straightforward and interpretable in comparison to images. Consequently, attempting to manually devise an effective backdoor trigger through direct observation of the raw I/Q samples may be challenging and inefficient.

Backdoor attacks can successfully deceive the model because optimizing the poisoned training data will shift the model's decision boundaries to accommodate the introduction of backdoor triggers. Hence, targeted backdoor attacks can be achieved by modifying input areas that are highly influential toward the targeted class prediction. The key challenge of this approach is to select these influential regions in a model-agnostic manner, without access to gradient information. XAI techniques emerge as a powerful tool to highlight influential features for a prediction. In this paper, we employ LIME to pinpoint input areas that strongly push the model toward the target class, allowing us to strategically insert triggers.

Because triggers must be inserted prior to training and we cannot use the target model itself, this impedes us from deploying LIME to determine feature importance and guide trigger selection. As RF fingerprints result from minute imperfections inherent to the device itself, we assume that the extraction of RF fingerprints should be carried out from similar regions across multiple samples from the same device. This extraction process should remain independent of the DNN's structure. Therefore, we can employ a surrogate model that enables LIME to perform significance analysis for identifying salient regions from this alternate model to guide trigger insertion for the target model. Specifically, we employ a simple CNN as the surrogate model, while avoiding excessive training time that could raise user suspicion. The key insight is that important regions for the surrogate are likely also to be impactful for the target model since RF fingerprints intrinsically depend on device characteristics rather than model structure.

After the surrogate CNN model finishes training, all examples belonging to the target class will be perturbed to generate inputs for LIME, as described in Section III. To generate input perturbations $x^{'}$ for LIME, superpixels of the original I/Q data are randomly selected using a Bernoulli distribution. We define superpixels as areas of 32 contiguous I/Q samples from the input. Then, perturbations are created by randomly deactivating chosen superpixels, setting the I/Q values within them to 0. These perturbed samples are then fed as inputs to an interpretable explanation model, Lasso regression in this case, to obtain importance weights $\mathbf{w}$ for each superpixel indicating its significance to the prediction. The Lasso model provides sparse feature weights highlighting the most influential superpixels for RF fingerprinting. To mitigate bias from individual samples, we calculate average importance weights across all target class examples as

$$\mathbf{w} = \frac{1}{M}\sum_{i}^{M}\mathbf{w}_i;\ \mathbf{w}_i = \underset{g=Lasso}{\arg\min}\mathcal{L}(f_s, g, \pi, \mathbf{x}_i), \forall y_i = y_t \quad (4)$$

where $\mathcal{L}$ is the loss of the Lasso regression and M is the number of examples $\mathbf{x}_i$ from the target class y_t in the subset D_s. The importance weight $\mathbf{w}_i$ for each data sample belonging to the target class is calculated by LIME. By taking the average of these weight vectors, $\mathbf{w}$ can effectively capture the regions most relevant to the target class for RF fingerprinting. Since the importance weights from LIME are concentrated in specific regions, we only select the superpixel with the highest weight as the location for inserting the trigger.

C. Trigger Value Generation

After identifying the trigger position using LIME, the next challenge is selecting appropriate trigger values. However, I/Q samples contain intrinsic preamble structures and dependencies between data points that originate from communication protocols, such as Wi-Fi or LoRa. This means we cannot arbitrarily select trigger values as it could introduce apparent anomalies that reveal the trigger's presence. A prior work [34] leverages XAI tools to select trigger values from the data

itself. However, in our case, the data exhibits varying formats and structures, making it impractical and unreasonable to directly employ existing I/Q data as the trigger data. Instead, inspired by steganography techniques [29], [36], [37], we aim to embed target device fingerprint features into the important region as the backdoor trigger. To achieve this, we employ an autoencoder to generate low-dimensional features specifically designed for the trigger. By doing this, when the victim model receives samples containing the embedded trigger, it recognizes the important region exhibiting target class features and consequently misclassifies to the target label naturally.

Similarly, we only use the subset D_s to train the autoencoder. The encoder part f_e learns to represent the key features and dependencies within each I/Q sample. The decoder f_d then attempts to reconstruct the original input from the encoded representation. The mean squared error (MSE) between the original and reconstructed inputs is used as the loss function to update the parameters of the autoencoder. To avoid biased features, we use the trained encoder to extract and average the representations of all I/Q data from the target class in D_s as

$$\mathbf{e} = \frac{1}{M}\sum_{i}^{M}\mathbf{e}_i;\ \mathbf{e}_i = f_e(\mathbf{x}_i),\ \forall y_i = y_t; \tag{5}$$

where $\mathbf{e}_i$ represents the embedding feature vector extracted by the encoder for each data $\mathbf{x}_i$, and $\mathbf{e}$ denotes the average of all such embeddings. This can eliminate individual sample biases and environmental interference, resulting in more stable feature embedding. This average feature embedding is then deployed as the trigger value. Compared to GAN-based approaches [33], using an autoencoder for trigger generation has two advantages. First, it does not require manipulating the training process, better matching practical attacker capabilities. Second, the GAN training process introduces additional overhead and makes the overall model training more challenging. This can potentially interfere with the effectiveness of the backdoor attack.

D. Summary

In this section, we will describe how to integrate the aforementioned methods for designing triggers. Algorithm 1 provides a comprehensive overview of how to combine LIME and autoencoder to generate the trigger.

Lines $1-12$ involve training a surrogate model f_s and an autoencoder on the subset D_s concurrently. The surrogate model is trained using a conventional supervised learning scheme, optimizing it with cross-entropy loss and the Adam optimizer. On the other hand, the autoencoder is trained in an unsupervised manner by minimizing the reconstruction loss, which is the mean squared error between the original input and reconstructed output. This allows the autoencoder to learn a latent representation that can faithfully reconstruct inputs. Lines $15-17$ generate importance weights by explaining the surrogate model's predictions using an interpretable model. Line 19 produces trigger values from the autoencoder's encoder f_e. Line 21 identifies the superpixel with maximum average importance weight and sets it to 1, with other regions set to 0. Line 22 averages the feature embeddings to obtain a representative trigger pattern. Line 23 performs a pointwise product between the averaged embedding and the selected importance region to insert the trigger. Overall, the trigger generation process is time-efficient because the surrogate model and autoencoder are trained concurrently, and the position and value selection occur in parallel.

Algorithm 1 Backdoor trigger generation

INPUT: Subset $D_s = \{(\mathbf{x}_i, y_i), i = 1, \ldots, N\}$, surrogate model f_s parameters θ, encoder f_e and decoder f_d parameters ψ, target class label y_t;

OUTPUT: Backdoor trigger t.

Step 1: Train the surrogate model and autoencoder

1: **while** θ have not converged **do**
2: **for** $(\mathbf{x}_i, y_i)^N \in D_s$ **do**
3: $\mathcal{L}_{CE} \leftarrow CrossEntropy(f_s(\mathbf{x}_i, y_i))$
4: **end for**
5: $\theta \leftarrow \theta - lr_\theta \cdot \nabla_\theta \mathcal{L}_{CE}$
6: **end while**
7: **while** ψ have not converged **do**
8: **for** $(\mathbf{x}_i, y_i)^N \in D_s$ **do**
9: $\mathcal{L}_{MSE} \leftarrow \frac{1}{N}\sum_i^N (f_d(f_e(\mathbf{x}_i)), \mathbf{x_i})^2$
10: **end for**
11: $\psi \leftarrow \psi - lr_\psi \cdot \nabla_\psi \mathcal{L}_{MSE}$
12: **end while**

Step 2: LIME-guided trigger generation

13: **for** $(\mathbf{x}_i, y_i)^M, \forall y_i = y_t$ **do**
14: `// Select trigger position`
15: $\xi_i^K \leftarrow Bern(p)$
16: $\mathbf{x}_i{}'^K \leftarrow \mathbf{x}_i \odot \xi_i^K$ `//Pointwise product`
17: $\mathbf{w}_i \leftarrow Lasso(\mathbf{x}_i{}'^K, f_s(\mathbf{x}_i{}'^K), \pi_{x_i}(\mathbf{x}_i{}'^K))$
18: `// Select trigger value`
19: $\mathbf{e}_i \leftarrow f_e(\mathbf{x}_i)$
20: **end for**
21: $\mathbf{p} \leftarrow PositionMax(\frac{1}{M}\sum_i^M \mathbf{w}_i)$
22: $\mathbf{e} \leftarrow \frac{1}{M}\sum_i^M \mathbf{e}_i$
23: $t \leftarrow \mathbf{e} \odot \mathbf{p}$
24: **return** t

VI. Experimental Evaluation and Analysis

A. Experiment Setup

The learning rate, max epochs, and poisoning rate λ are set to 0.0001, 100, and 0.1, respectively. The target class label is set to 0 (the first device) across all cases. All experiments are conducted on a server with an Intel Xeon E5-2650L v4 CPU and 8 NVIDIA GeForce GTX 1080Ti GPU.

1) Victim Models: We evaluate three different model architectures on I/Q time domain inputs: a three-layer MLP (multilayer perceptron), a CNN following the structure of [8], and a GRU (gated recurrent unit) model consisting of two GRU layers. The GRU model uses an embedding vector length of 256 and two dense layers to extract time-series features for RF fingerprinting. Additionally, we test a CNN model on

TABLE I
DATASET SUMMARY.

Dataset	# of samples	# of devices
ORACLE	128,000	16
CORES	135,776	58
WiSig	102,945	130
Ours	24,000	10

Fig. 3. LoRa transmitters and a USRP receiver.

spectrogram inputs in the time-frequency domain, denoted as S-CNN. This CNN has a similar structure to the previous one but with modifications to the input layer to fit spectrograms.

2) Attack Models: The surrogate model employs a simple 1D CNN architecture taking the 2×256 I/Q samples as a 1×256 input with two channels. It contains two 1D convolutional layers with 64 and 16 kernels of size 3, respectively. The autoencoder is comprised of three convolutional layers as the encoder and three corresponding deconvolutional layers as the decoder.

3) Datasets: This paper employs three public datasets and an original dataset collected by ourselves, incorporating both Wi-Fi and LoRa protocols. Table I summarizes key information about these datasets. The original ORACLE dataset [8] is captured with 16 USRP X310 transmitters and a USRP B210 receiver. [38] is collected by UCLA CORES lab, consisting of 163 consumer Wi-Fi cards arranged in a grid at the Orbit Testbed [39]. For our work, we use 58 devices of this dataset and denote it as CORES. Conducted by the same team as the CORES, the WiSig dataset [40] is collected by 41 unspecified USRP receivers to capture wireless signals from 174 COTS Wi-Fi cards. The wireless devices communicate with an 802.11a/g access point over channel 11. Besides, we also aggregate the Wi-Fi datasets into a combined dataset for additional analysis. To ensure that our analysis is manageable and focused, we only select portions of the three extensive datasets in this paper.

As shown in Fig. 3, our dataset is created using ten commercial off-the-shelf LoRa transmitters (Pycom LoPy4), and a USRP N210 software-defined radio platform as the receiver. Due to differing sampling rates and preamble structures, the original captured I/Q data for LoRa is 2×1024 in size. This is downsampled to 2×256 to conform to the input size requirements of the models.

Testing across these diverse datasets and standards provides comprehensive evaluation and insights into the attack's impact across various model types and input domains. The breadth of experiments enables robust and thorough analysis.

B. Evaluation Metrics

1) Effectiveness: To analyze the effectiveness of our attack, we employ the attack success rate (ASR) and benign accuracy (BA) as the metrics. ASR is the ratio of successfully attacked poison samples to the total number of poison samples, while BA denotes the accuracy achieved on benign samples.

2) Stealthiness: Visual inspection is insufficient for evaluating trigger stealthiness. Therefore, this study employs three approaches to quantify it, namely (i) trigger size, (ii) isolation forest [41], and (iii) STRIP [42]. (i), we use l-norms to directly measure the size of triggers and the difference between the l-norms of the input before and after adding triggers. (ii) Isolation forest is an unsupervised anomaly detection algorithm that identifies rare and dissimilar points instead of constructing a model based on normal samples. The underlying idea is that poisoned samples may be detected as outliers due to their similarity in comparison to the highly diverse background points. We keep default settings in this paper. (iii) STRIP detects poisoned samples by measuring the predicted entropy of samples generated by applying various inputs to suspicious images. The entropy quantifies the randomness of the predictions. Thus, attacks with higher entropy are more difficult for STRIP to detect. The rationale behind this lies in the fact that the presence of triggers leads to the incorporation of these triggers in each sample. As a result, the predicted value of the target class increases regardless of the input, leading to lower entropy compared to normal samples.

3) Robustness: The last goal of the attack is to ensure its robustness against defense methods. Pruning is one of the most direct approaches to sanitize the victim model. Prior work [43] reveals that the backdoor often relies on a specific group of neurons for trigger recognition, rendering them unresponsive to clean data. Fine-tuning the model on the clean dataset can further mitigate the attacker's capabilities. In this work, we use a pruning rate of 50% as the first defense method. Then, we apply Fine-Pruning [43] to retrain the victim model using the available clean dataset provided to the defenders. We consider two different learning rates: the first is the same as the learning rate used during training, and the second is five times larger to help the model escape local minima.

C. Backdoor Triggers

In addition to our proposed attack, we evaluate three other backdoor triggers for comparison. Following [23], we use Gaussian noise (GN) to mimic channel noise, which is a reasonable trigger for RF fingerprinting tasks. We also implement a BadNet trigger [15] by prominently perturbing the first 8 I/Q samples. To better analyze our method's efficacy, we conduct tests using a trigger denoted as LG, which inserts Gaussian noise into the region identified by LIME. Our attack differs from LG in using target class statistics to craft a representative trigger, rather than arbitrary noise. Comparisons among these triggers will demonstrate the effectiveness of our techniques for generating tailored and stealthy backdoor attacks on RF fingerprinting. Fig. 4 shows examples of the different trigger patterns in both the time and time-frequency domains. The time-domain plot of BadNet appears smaller than the rest of the data because the scale is dominated by the large trigger values. This makes the normal data look smaller in comparison.

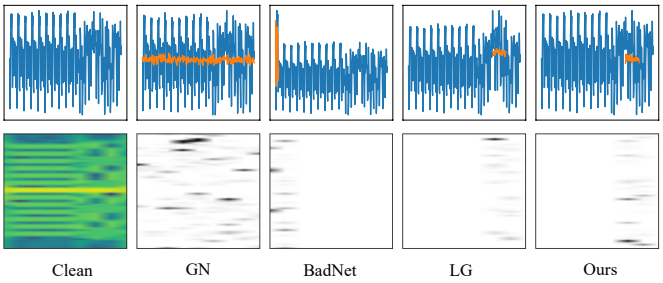

Fig. 4. Examples of triggers in the time domain and time-frequency domain. For better visualization, the time-frequency domain triggers are amplified by a factor of 10.

D. Effectiveness Evaluation

Table II presents the classification accuracy achieved on each dataset using standard training without attacks. MLP is excluded for LoRa time domain fingerprinting as it failed to reach eligible accuracy (only about 20%) on this data. This may be attributed to the downsampling process, which could have hindered the MLP model from learning the underlying features needed for accurate classification. As illustrated in Section IV-B1, the threat model should achieve comparable BA on clean samples as standard training while maximizing ASR on poisoned samples as much as possible. Maintaining accuracy on clean data is crucial for avoiding detection.

TABLE II
ACCURACY WITH STANDARD TRAINING.

		ORACLE	WiSig	CORES	Combined	LoRa
Raw I/Q	MLP	0.7179	0.9343	0.9897	0.8043	-
	CNN	0.9389	0.9669	0.9952	0.9049	0.8050
	GRU	0.8425	0.9140	0.9885	0.8770	0.7250
Spectrogram	CNN	0.8794	0.9733	0.9998	0.9386	0.8155

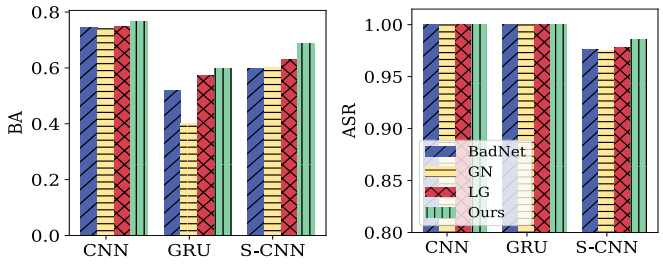

Fig. 5. BA and ASR on the LoRa dataset. S-CNN refers to the spectrogram input CNN model.

The effectiveness results of the various backdoor attacks on LoRa fingerprinting are summarized in Fig. 5. All four backdoor attacks lead to a degradation in BA, especially for GRU and S-CNN (spectrogram CNN). Among these attacks, the GN attack causes the most significant BA degradation, while our method results in the least degradation. Regarding the ASR, the attacks in the time domain prove to be effective for both GRU and CNN models. However, when transforming to the time-frequency domain, ASR decreases for all attacks. Our proposed trigger retains the highest ASR in the time-frequency domain while GN experiences the lowest ASR. Overall, our proposed attack consistently achieves high ASR with lower BA impact compared to other attacks, demonstrating its effectiveness across diverse models and input domains in LoRa fingerprinting.

Table III shows that our attack is also effective for Wi-Fi fingerprinting. On the ORACLE dataset, all three victim models in the time domain experience a substantial decrease in BA, while such decrease is not observed in the time-frequency domain. Although universal BA drops on the time domain ORACLE data, our approach substantially reduces the extent of this decrease. Specifically, BadNet and GN lower BA around 15% for MLP, while our attack only reduces BA by 6% without sacrificing ASR. Compared to BadNet and GN, our trigger improves BA by about 3% and 2% for CNN and GRU, respectively. On other datasets, our attack even improves BA when using the MLP model. This may be because our triggers contain target device features and emphasize important regions, which assists MLP in categorizing inputs more accurately. This additional information helps to compensate for the limitations of MLP, which has slightly weaker feature extraction than CNN and GRU. Notably, GRU exhibits strong resilience against the GN attack for I/Q data, with substantially lower ASR than other attacks across all datasets. This suggests that the GRU model is intrinsically robust due to its ability to learn long-term dependencies and resist noise. Even on this defensive model, our proposed attack still outperforms other attacks, attaining the highest BA on three datasets while receiving two highest and two second-highest ASR.

In the time-frequency domain, the differences between attacks are less significant compared to the time domain. However, GN still performs poorer than others, particularly in terms of BA. In contrast, our attack continues to exhibit a high ASR while keeping the BA unaffected. Despite the less pronounced discrepancies in the time-frequency domain, our method proves to be effective, maintaining a high ASR without compromising on the accuracy of benign samples even after employing STFT to transform I/Q data into spectrograms. Furthermore, the CNN model exhibits higher BA but lower ASR in the time-frequency domain versus the time domain. This may be because RF fingerprinting features become more pronounced after STFT, enhancing the discrimination of clean samples. Meanwhile, triggers could lose some abilities after transformation, reducing attack effectiveness in the time-frequency domain. This is the rationale behind the adoption of data preprocessing in many defense methods [44].

E. Stealthiness Evaluation

As shown in Fig. 4, most triggers are difficult to discern through human inspection. The BadNet trigger is more visible in the time domain. Meanwhile, the GN trigger is more apparent in the time-frequency domain. In general, the stealthiness of these triggers makes them challenging to detect without close inspection. To provide further analysis, we employ l-

TABLE III

BA, ASR, AND ASR AFTER PRUNING FOR BACKDOOR ATTACKS ON WI-FI FINGERPRINTING ACROSS VARIOUS DATASETS AND MODEL ARCHITECTURES. THE BEST RESULT IN EACH CASE IS DENOTED IN BOLD, WHILE THE SECOND BEST IS UNDERLINED.

Dataset		Time-domain									Time-frequency domain		
	Model	MLP			CNN			GRU			CNN		
	Metric	BA	ASR	Prune	BA	ASR	Prune	BA	ASR	Prune	BA	ASR	Prune
ORACLE	BadNet	0.5768	**0.9993**	**0.8975**	0.8434	0.9709	0.9661	0.7860	0.9692	0.6993	**0.8765**	**0.9612**	0.8916
	GN	0.5645	0.9624	0.8452	0.8485	0.9214	0.8776	0.7918	0.1574	0.1049	0.8726	0.9060	0.8696
	LG	0.6049	0.9937	0.8304	0.8734	0.9753	0.9707	0.8016	**0.9776**	0.5371	0.8751	0.9594	**0.8992**
	Ours	**0.6515**	0.9951	0.8338	**0.8762**	**0.9794**	**0.9761**	**0.8088**	0.9713	**0.9306**	0.8732	0.9381	0.8727
WiSig	BadNet	0.9339	**0.9973**	0.9566	**0.9484**	0.9781	0.9720	0.9127	0.9638	0.9248	0.9709	0.9908	0.8798
	GN	0.9343	0.9573	0.8293	0.9430	0.9313	0.8360	0.9151	0.7470	0.2013	0.9727	**0.9932**	**0.8823**
	LG	0.9394	0.9953	**0.9916**	0.9438	0.9739	**0.9751**	0.9131	0.9629	0.9280	0.9713	0.9855	0.8633
	Ours	**0.9424**	0.9942	0.9913	0.9473	**0.9784**	0.9619	**0.9176**	**0.9782**	**0.9350**	**0.9787**	0.9921	0.8808
CORES	BadNet	0.9905	0.9987	0.9989	0.9955	0.9992	0.9411	0.9615	0.9842	0.8945	0.9996	0.9987	**0.9000**
	GN	0.9896	0.9333	0.8579	0.9876	0.9701	0.8169	0.9840	0.2894	0.1205	0.9861	0.9112	0.8958
	LG	0.9919	0.9990	0.9994	0.9968	0.9971	0.9768	0.9853	0.9944	0.9885	0.9996	0.9948	0.8990
	Ours	**0.9926**	**1.000**	**1.000**	**0.9974**	**1.000**	**1.000**	**0.9865**	**0.9991**	**0.9988**	**0.9998**	**0.9997**	0.8999
Combined	BadNet	0.7808	0.9971	0.8951	0.8909	0.9731	0.9662	0.8592	0.9624	0.8833	0.9338	0.9195	0.6924
	GN	0.7841	0.9781	0.8846	0.8988	0.9539	0.8776	**0.8798**	0.6203	0.2213	**0.9345**	0.9163	0.6684
	LG	0.8010	0.9972	0.9895	0.9079	**0.9734**	0.9643	0.8741	**0.9799**	0.8828	0.9249	**0.9346**	0.6988
	Ours	**0.8120**	**0.9984**	**0.9898**	**0.9083**	0.9733	**0.9666**	0.8731	0.9777	**0.8918**	0.9342	0.9325	**0.7739**

norms and two anomaly detection methods for additional quantitative insights into stealthiness.

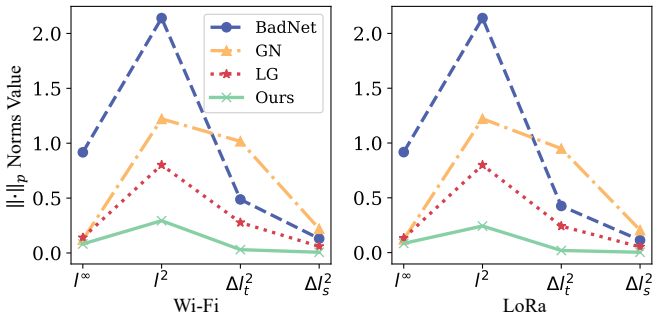

Fig. 6. $||\cdot||_p$ norms on different triggers. Δl_t^2 denotes the l^2 norm changes after adding triggers in the time domain, while Δl_s^2 denotes the changes after STFT.

Fig. 6 presents the trigger size in terms of norms. The trends are similar for both Wi-Fi and LoRa datasets. As expected, the visible BadNet trigger exhibits the largest l^2-norm and l^∞-norm. However, it is important to highlight that the GN trigger demonstrates the most noticeable changes in l-norms of the samples before and after trigger addition. In the time-frequency domain, BadNet also causes smaller perturbations than GN, consistent with previous visual analysis. In contrast, our proposed triggers have the lowest l-norm values and changes in both the time and time-frequency domains across all datasets.

Table IV presents the isolation forest results. The ORACLE and LoRa datasets have the highest anomaly rates. However, even for the clean samples, the time domain ORACLE data has a 16% anomaly rate, while the LoRa data has 44% and 28% anomaly rates in the time and time-frequency domains,

TABLE IV

THE RESULTS OF ISOLATION FOREST. HIGHER VALUES INDICATE MORE POISONED SAMPLES REMOVED.

	Time Domain				Time-frequency Domain			
	BN	GN	LG	Ours	BN	GN	LG	Ours
ORACLE	0.2969	0.3522	0.2254	0.1685	0.1424	0.1608	0.0607	0.0437
WiSig	0.0228	0.0279	0.0124	0.0102	0.1575	0.1758	0.0897	0.0839
CORES	0.0751	0.0615	0.0330	0.0205	0.1916	0.5252	0.0598	0.0939
Combined	0.0928	0.0921	0.0571	0.0383	0.2203	0.1367	0.0555	0.0567
LoRa	0.5789	0.7083	0.5733	0.4773	0.7250	0.9100	0.5343	0.4437

respectively. This explains the lower clean accuracy rates for these datasets compared to others, as shown in Table II. In general, our method consistently achieves the lowest anomaly rate, approaching that of clean samples. Despite its smaller l^2-norm, the GN attack has a higher anomaly rate than BadNet for time domain data. Interestingly, anomaly rates increase for almost all datasets in the time-frequency domain. For instance, on CORES, the anomaly rates for BadNet and GN increase from about 7% in the time domain to 19% and 52% in the time-frequency domain, respectively. This demonstrates data processing can provide some defense against backdoor attacks by making triggers more visible than in the original time domain. This observation aligns with frequency domain backdoor attacks on images [30]. However, even after transforming to the time-frequency domain, our trigger still results in samples with the lowest anomaly rates.

Table VI presents the STRIP results. Overall, image-based defenses seem less effective for RF fingerprint data. Although the entropy difference between clean and poisoned samples is not highly pronounced, STRIP successfully detected BadNet attacks on ORACLE and Combined datasets in the time domain, and GN attacks in the time-frequency domain. This may be because STRIP is designed to detect backdoors in

TABLE V
ASR OF DEFENDING BACKDOOR ATTACKS ON WI-FI FINGERPRINTING USING FINE-PRUNING WITH DIFFERENT LEARNING RATES.

Dataset	Model	MLP				CNN				GRU				S-CNN			
	lr↓	BadNet	GN	LG	Ours	BadNet	GN	LG	Ours	BadNet	GN	LG	Ours	BadNet	GN	LG	Ours
ORACLE	$1 \times lr$	0.8014	0.7581	0.9799	0.9594	0.9575	0.8116	0.9706	0.9682	0.6860	0.0778	0.7771	0.7994	0.8898	0.8838	0.8949	0.8956
	$5 \times lr$	0.0708	0.1296	0.2651	0.3846	0.5074	0.4208	0.5977	0.6382	0.1227	0.0506	0.1509	0.1807	0.2256	0.1270	0.2774	0.2973
WiSig	$1 \times lr$	0.8122	0.7295	0.9481	0.9483	0.9522	0.9262	0.9789	0.9631	0.5339	0.0471	0.7376	0.7592	0.9699	0.9658	0.9760	0.9709
	$5 \times lr$	0.0129	0.0609	0.1232	0.2059	0.0332	0.1764	0.3922	0.7068	0.0276	0.0107	0.1706	0.3183	0.0542	0.0875	0.1246	0.3989
CORES	$1 \times lr$	0.9170	0.8169	0.9996	1.0000	0.7340	0.7938	0.9977	1.0000	0.9784	0.0199	0.9820	0.9957	0.9012	0.8799	0.8854	0.9991
	$5 \times lr$	0.0169	0.0170	0.1768	0.3426	0.0356	0.0287	0.1907	0.4706	0.0304	0.0171	0.1294	0.2694	0.0172	0.0189	0.0180	0.0757
Combined	$1 \times lr$	0.8532	0.7044	0.9531	0.9274	0.9580	0.8151	0.9782	0.9660	0.8793	0.1149	0.8412	0.9046	0.6797	0.6674	0.7049	0.7102
	$5 \times lr$	0.0222	0.0719	0.2164	0.2709	0.1626	0.3624	0.4189	0.5126	0.0305	0.0221	0.2543	0.3902	0.1719	0.1144	0.2779	0.3597

TABLE VI
ENTROPY DIFFERENCE PRODUCED BY STRIP BETWEEN BENIGN AND POISONED INPUTS. NEGATIVE VALUES IN BOLD INDICATE POTENTIAL DETECTION.

	Time Domain				Time-frequency Domain			
	BN	GN	LG	Ours	BN	GN	LG	Ours
ORACLE	**-0.0053**	0.0027	0.0035	0.0045	0.0396	**-0.0030**	0.0039	0.0092
WiSig	0.0779	0.0200	0.1463	0.0942	0.1489	0.0500	0.1040	0.0817
CORES	0.1013	0.0070	0.1393	0.0882	0.2438	0.0542	0.0828	0.2985
Combined	**-0.0080**	0.0021	0.0029	0.0031	0.0379	**-0.0536**	0.0042	0.0154
LoRa	0.0253	0.0190	0.0197	0.0279	0.0880	0.0353	0.0077	0.0963

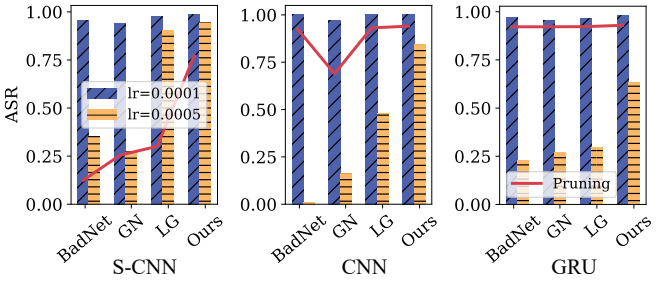

Fig. 7. Pruning (red line) and Fine-Pruning with different learning rates on LoRa fingerprinting.

image data. For 2D time-series data, STRIP fails to identify invariant triggers after stacking inputs. Only BadNet triggers with large l^2-norm are detected in some cases. Similarly, after transforming the time series data to spectrograms using STFT, only GN triggers can be detected. It is noted that our triggers still exhibit a notable increase in entropy, rendering them less susceptible to detection. Based on the three detection methods, our trigger demonstrates the best stealthiness, being not only the smallest in scale but also the least likely to be detected by the detection algorithms.

F. Robustness Evaluation

Attackers need to ensure the backdoored model remains robust when victims deploy defense methods. We assume that defenders have a small labeled clean dataset, 30% of the training set in this paper, and can modify the victim model. Table III shows the ASR after implementing pruning for Wi-Fi fingerprinting. Except for defending GRU against GN attacks, which show some effect, pruning does not significantly reduce ASR in other cases. Fig. 7 presents the case for LoRa fingerprinting. Pruning provides some defense for S-CNN, substantially reducing ASR of BadNet and GN attacks. However, it does not significantly help for other cases, which might be attributed to the limited capacity offered by pruning.

Table V and Fig. 7 summarize retraining results with the original and five times the learning rates. When using the original learning rate, only the GN attack on GRU can be effectively mitigated. However, other cases see limited effect. In contrast, retraining with a larger learning rate drastically reduces ASR for all attacks. In the case of the MLP model, BadNet and GN attacks are almost entirely disabled, while our attack maintains an ASR ranging from 20% to 38%. This trend is consistent across other models. Furthermore, for CORES data in the time-frequency domain, this defense strategy proves highly effective, with even our attack method retaining only 7% of the ASR. This is probably because that CORES has more distinctive features, allowing retraining to effectively forget trigger patterns. This also explains why CORES consistently achieves the highest BA. In general, our attack is challenging to be completely defended while BadNet and GN are defeated.

VII. CONCLUSION

In this paper, we proposed the first effective backdoor attack on RF fingerprinting systems. To address real-world scenarios, we limited the attacker's capabilities without manipulating gradients or additional training components. To achieve this, we employed an XAI method to guide trigger placement and an autoencoder to extract features as trigger values. Extensive experiments were conducted on three public WiFi datasets and one self-collected LoRa dataset. We compared our approach with three other attacks on various neural network architectures. Additionally, we investigated both time-domain I/Q samples and their time-frequency domain spectrograms. Our proposed attack outperforms the benchmark approaches in terms of effectiveness, stealthiness, and robustness of our attack under different detection and defense methods.

ACKNOWLEDGMENTS

This work is also supported in part by the NSF (CNS-2321763, CNS-2319343, CNS-2317190, IIS-2306791, IIS-2306789, CNS-2107190, and CNS-2319342).

REFERENCES

[1] Y. Zou, J. Zhu, X. Wang, and L. Hanzo, "A survey on wireless security: Technical challenges, recent advances, and future trends," *Proc. IEEE*, vol. 104, no. 9, pp. 1727–1765, 2016.

[2] E. Perenda, S. Rajendran, G. Bovet, M. Zheleva, and S. Pollin, "Contrastive learning with self-reconstruction for channel-resilient modulation classification," in *Proc. IEEE Conf. Computer Communications (INFOCOM)*. IEEE, 2023, pp. 1–10.

[3] Q. Xu, R. Zheng, W. Saad, and Z. Han, "Device fingerprinting in wireless networks: Challenges and opportunities," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 1, pp. 94–104, 2015.

[4] J. Zhang, G. Shen, W. Saad, and K. Chowdhury, "Radio frequency fingerprint identification for device authentication in the internet of things," *IEEE Commun. Mag.*, 2023.

[5] S. Riyaz, K. Sankhe, S. Ioannidis, and K. Chowdhury, "Deep learning convolutional neural networks for radio identification," *IEEE Commun. Mag.*, vol. 56, no. 9, pp. 146–152, 2018.

[6] 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, 2021.

[7] L. Peng, A. Hu, J. Zhang, Y. Jiang, J. Yu, and Y. Yan, "Design of a hybrid rf fingerprint extraction and device classification scheme," *IEEE Internet Things J.*, vol. 6, no. 1, pp. 349–360, 2018.

[8] K. Sankhe, M. Belgiovine, F. Zhou, S. Riyaz, S. Ioannidis, and K. Chowdhury, "ORACLE: Optimized radio classification through convolutional neural networks," in *Proc. IEEE Conf. Computer Communications (INFOCOM)*. IEEE, 2019, pp. 370–378.

[9] G. Shen, J. Zhang, A. Marshall, L. Peng, and X. Wang, "Radio frequency fingerprint identification for LoRa using spectrogram and cnn," in *Proc. IEEE Conf. Computer Communications (INFOCOM)*. IEEE, 2021, pp. 1–10.

[10] A. Al-Shawabka, P. Pietraski, S. B. Pattar, F. Restuccia, and T. Melodia, "DeepLoRa: Fingerprinting LoRa devices at scale through deep learning and data augmentation," in *Proc. 22nd Int. Symp. Theory, Algorithmic Foundations, and Protocol Design for Mobile Networks and Mobile Computing (MobiHoc)*, 2021, pp. 251–260.

[11] 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 Conf. Computer Communications (INFOCOM)*. IEEE, 2020, pp. 646–655.

[12] 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.

[13] 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.

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

[15] T. Gu, B. Dolan-Gavitt, and S. Garg, "BadNets: Identifying vulnerabilities in the machine learning model supply chain," *arXiv preprint arXiv:1708.06733*, 2017.

[16] Y. Gao, B. G. Doan, Z. Zhang, S. Ma, J. Zhang, A. Fu, S. Nepal, and H. Kim, "Backdoor attacks and countermeasures on deep learning: A comprehensive review," *arXiv preprint arXiv:2007.10760*, 2020.

[17] D. Adesina, C.-C. Hsieh, Y. E. Sagduyu, and L. Qian, "Adversarial machine learning in wireless communications using rf data: A review," *IEEE Commun. Surveys Tuts.*, vol. 25, no. 1, pp. 77–100, 2022.

[18] J. Liu, M. Nogueira, J. Fernandes, and B. Kantarci, "Adversarial machine learning: A multilayer review of the state-of-the-art and challenges for wireless and mobile systems," *IEEE Commun. Surveys Tuts.*, vol. 24, no. 1, pp. 123–159, 2021.

[19] S.-M. Moosavi-Dezfooli, A. Fawzi, O. Fawzi, and P. Frossard, "Universal adversarial perturbations," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 1765–1773.

[20] N. Akhtar and A. Mian, "Threat of adversarial attacks on deep learning in computer vision: A survey," *IEEE Access*, vol. 6, pp. 14 410–14 430, 2018.

[21] Y. Li, Y. Jiang, Z. Li, and S.-T. Xia, "Backdoor learning: A survey," *IEEE Trans. Neural Netw. Learn. Syst.*, 2022.

[22] M. T. Ribeiro, S. Singh, and C. Guestrin, ""why should I trust you?" explaining the predictions of any classifier," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining*, 2016, pp. 1135–1144.

[23] X. Chen, C. Liu, B. Li, K. Lu, and D. Song, "Targeted backdoor attacks on deep learning systems using data poisoning," *arXiv preprint arXiv:1712.05526*, 2017.

[24] A. Shafahi, W. R. Huang, M. Najibi, O. Suciu, C. Studer, T. Dumitras, and T. Goldstein, "Poison frogs! targeted clean-label poisoning attacks on neural networks," *Advances in Neural Information Processing Systems*, vol. 31, 2018.

[25] T. Zheng and B. Li, "First-order efficient general-purpose clean-label data poisoning," in *Proc. IEEE Conf. Computer Communications (INFOCOM)*. IEEE, 2021, pp. 1–10.

[26] R. Ning, J. Li, C. Xin, and H. Wu, "Invisible poison: A blackbox clean label backdoor attack to deep neural networks," in *Proc. IEEE Conf. Computer Communications (INFOCOM)*. IEEE, 2021, pp. 1–10.

[27] S. Zhao, X. Ma, X. Zheng, J. Bailey, J. Chen, and Y.-G. Jiang, "Clean-label backdoor attacks on video recognition models," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 14 443–14 452.

[28] T. A. Nguyen and A. Tran, "Input-aware dynamic backdoor attack," *Advances in Neural Information Processing Systems*, vol. 33, pp. 3454–3464, 2020.

[29] Y. Li, Y. Li, B. Wu, L. Li, R. He, and S. Lyu, "Invisible backdoor attack with sample-specific triggers," in *Proc. IEEE/CVF Int. Conf. Computer Vision*, 2021, pp. 16 463–16 472.

[30] Y. Zeng, W. Park, Z. M. Mao, and R. Jia, "Rethinking the backdoor attacks' triggers: A frequency perspective," in *Proc. IEEE/CVF Int. Conf. Computer Vision*, 2021, pp. 16 473–16 481.

[31] T. Zheng and B. Li, "Poisoning attacks on deep learning based wireless traffic prediction," in *Proc. IEEE Conf. Computer Communications (INFOCOM)*. IEEE, 2022, pp. 660–669.

[32] Y. Jiang, X. Ma, S. M. Erfani, and J. Bailey, "Backdoor attacks on time series: A generative approach," in *Proc. IEEE Conf. Secure and Trustworthy Machine Learning (SaTML)*. IEEE, 2023, pp. 392–403.

[33] R. Ning, C. Xin, and H. Wu, "Trojanflow: A neural backdoor attack to deep learning-based network traffic classifiers," in *Proc. IEEE Conf. Computer Communications (INFOCOM)*. IEEE, 2022, pp. 1429–1438.

[34] G. Severi, J. Meyer, S. Coull, and A. Oprea, "Explanation-Guided backdoor poisoning attacks against malware classifiers," in *Proc. 30th USENIX Security Symp.*, 2021, pp. 1487–1504.

[35] T. Zhao, Z. Tang, T. Zhang, H. Phan, Y. Wang, C. Shi, B. Yuan, and Y. Chen, "Stealthy backdoor attack on rf signal classification," in *Proc. IEEE Int. Conf. Computer Communications and Networks (ICCCN)*. IEEE, 2023, pp. 1–10.

[36] S. Baluja, "Hiding images in plain sight: Deep steganography," *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[37] S. Li, M. Xue, B. Z. H. Zhao, H. Zhu, and X. Zhang, "Invisible backdoor attacks on deep neural networks via steganography and regularization," *IEEE Trans. Depend. Sec. Comput.*, vol. 18, no. 5, pp. 2088–2105, 2020.

[38] S. Hanna, S. Karunaratne, and D. Cabric, "Open set wireless transmitter authorization: Deep learning approaches and dataset considerations," *IEEE Trans. on Cogn. Commun. Netw.*, vol. 7, no. 1, pp. 59–72, 2020.

[39] D. Raychaudhuri, I. Seskar, M. Ott, S. Ganu, K. Ramachandran, H. Kremo, R. Siracusa, H. Liu, and M. Singh, "Overview of the ORBIT radio grid testbed for evaluation of next-generation wireless network protocols," in *Proc. IEEE Wireless Communications and Networking Conference*, vol. 3. IEEE, 2005, pp. 1664–1669.

[40] S. Hanna, S. Karunaratne, and D. Cabric, "WiSig: A large-scale wifi signal dataset for receiver and channel agnostic RF fingerprinting," *IEEE Access*, vol. 10, pp. 22 808–22 818, 2022.

[41] F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation forest," in *Proc. 8th IEEE Int. Conf. Data Mining*. IEEE, 2008, pp. 413–422.

[42] Y. Gao, C. Xu, D. Wang, S. Chen, D. C. Ranasinghe, and S. Nepal, "Strip: A defence against trojan attacks on deep neural networks," in *Proc. 35th Annual Computer Security Applications Conf.*, 2019, pp. 113–125.

[43] K. Liu, B. Dolan-Gavitt, and S. Garg, "Fine-pruning: Defending against backdooring attacks on deep neural networks," in *Proc. Int. Symp. Research in Attacks, Intrusions, and Defenses*. Springer, 2018, pp. 273–294.

[44] C. Guo, M. Rana, M. Cisse, and L. Van Der Maaten, "Countering adversarial images using input transformations," *arXiv preprint arXiv:1711.00117*, 2017.