IEEE ComSoc IEEE Communications Society | IEEE COMPUTER SOCIETY | IEEE Signal Processing Society | VTS | IEEE Transactions on Machine Learning in Communications and Networking

Received March, 2023; revised August, 2023 and December 2023; accepted February, 2024; Date of publication XX Month, 2024; date of current version XX Month, 2024. The associate editor coordinating the review of this article and approving it for publication was Gunes Karabulut Kurt.

Digital Object Identifier 10.1109/TMLCN.2024.1234567

Stealthy Adversarial Attacks on Machine Learning-Based Classifiers of Wireless Signals

Wenhan Zhang[1], Student Member, IEEE, Marwan Krunz[1], Fellow, IEEE, and Gregory Ditzler[2]

[1]University of Arizona, Tucson, AZ 85712 USA
[2]EpiSys Science, Inc. (EpiSci), Philadelphia, PA 18128 USA

Corresponding author: W. Zhang (email: wenhanzhang@arizona.edu).

This research was supported by the Army Research Office under Contract No. W911NF-21-C-0016, the National Science Foundation (grants Nos. 1943552, 2229386, and 1822071), and by the Broadband Wireless Access & Applications Center (BWAC).

ABSTRACT Machine learning (ML) has been successfully applied to classification tasks in many domains, including computer vision, cybersecurity, and communications. Although highly accurate classifiers have been developed, research shows that these classifiers are, in general, vulnerable to adversarial machine learning (AML) attacks. In one type of AML attack, the adversary trains a surrogate classifier (called the *attacker's classifier*) to produce intelligently crafted low-power "perturbations" that degrade the accuracy of the targeted (*defender's*) classifier. In this paper, we focus on radio frequency (RF) signal classifiers, and study their vulnerabilities to AML attacks. Specifically, we consider several exemplary protocol and modulation classifiers, designed using convolutional neural networks (CNNs) and recurrent neural networks (RNNs). We first show the high accuracy of such classifiers under random noise (AWGN). We then study their performance under three types of low-power AML perturbations (FGSM, PGD, and DeepFool), considering different amounts of information at the attacker. On one extreme (so-called "white-box" attack), the attacker has complete knowledge of the defender's classifier and its training data. As expected, our results reveal that in this case, the AML attack significantly degrades the defender's classification accuracy. We gradually reduce the attacker's knowledge and study five attack scenarios that represent different amounts of information at the attacker. Surprisingly, even when the attacker has limited or no knowledge of the defender's classifier and its power is relatively low, the attack is still significant. We also study various practical issues related to the wireless environment, including channel impairments and misalignment between attacker and transmitter signals. Furthermore, we study the effectiveness of intermittent AML attacks. Even under such imperfections, a low-power AML attack can still significantly reduce the defender's classification accuracy for both protocol and modulation classifiers. Lastly, we propose a two-step adversarial training mechanism to defend against AML attacks and contrast its performance against other state-of-the-art defense strategies. The proposed defense approach increases the classification accuracy by up to 50%, even in scenarios where the attacker has perfect knowledge of the defender and exhibits a relatively large power budget.

INDEX TERMS Deep learning, signal classification, adversarial machine learning, shared spectrum, wireless security

I. Introduction

Machine learning (ML) based signal classification plays an important role in next-generation wireless systems. It can be used, for example, to identify the underlying protocol or modulation scheme of the received signal in a spectrum-sharing scenario, e.g., coexisting Wi-Fi and cellular transmissions over the unlicensed 5/6 GHz bands [1]–[3], and LTE/radar transmissions over the CBRS band [4], [5].

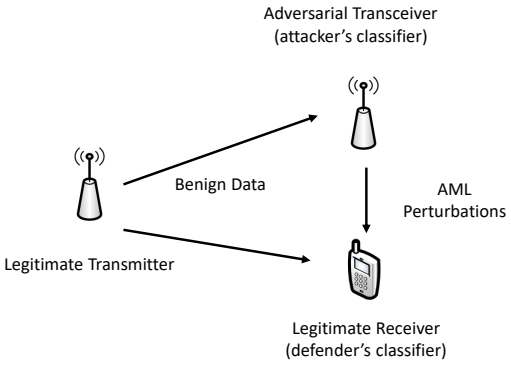


FIGURE 1: AML perturbations attack on a signal classifier in wireless systems.

It can also be used to identify anomalies, rogue signals, and selective-jamming attacks [6]–[8]. Signal classification may also be used for RF fingerprinting [9]–[11] to provide awareness of nearby emitters and avoid radio interference. Deep neural networks (DNNs) were used in [12], [13] to identify signal types not included in the training phase, i.e., unknown signals. In [14], [15], the authors proposed more advanced DNNs, including fusion convolutional neural networks (CNNs) and self-supervised DNNs to improve the accuracy of modulation classification. Recent DNN-based RF signal classifiers also use recurrent neural networks (RNNs) [16], [17] (see also [18], [19] and the references therein for related work on ML-based signal classification).

However, ML classifiers are vulnerable to adversarial machine learning (AML) attacks. These attacks can infer membership [20], leave a backdoor in the data [21], poison the data [22], or mislead the classifier into assigning wrong labels during normal operation [23]–[27]. In this paper, we focus on the last type of AML attacks. Specifically, we investigate the impact of AML perturbations on signal classifiers, considering realistic aspects of the wireless scenario. AML perturbations have mainly been studied in the context of object classification/recognition, but more recently in the context of RF signal classification (e.g., [28]–[34]). In such attacks, an adversary trains a *surrogate* DNN, henceforth called the *attacker's classifier*, to produce cleverly crafted perturbations that are difficult to detect. When combined with the original (a.k.a. "benign") samples, these perturbations can mislead the *defender's classifier* into wrongly classifying the signal type (see Figure 1).

Several factors contribute to the effectiveness of an AML attack, including how much knowledge the attacker has about the defender and what imperfections the AML perturbations may encounter before reaching the defender's classifier. In [28]–[31], the authors studied AML attacks in two extreme scenarios: the attacker has full knowledge of the defender's classifier (*white-box attack*) or it has zero knowledge (*black-box attack*). Specifically, in [28] the authors adapted the original Fast Gradient Sign Method (FGSM) for generating perturbations [23] to attack modulation classifiers, assuming the attacker has perfect knowledge of the defender's classier. The authors in [29]–[31] showed that DNN-based signal classifiers are vulnerable to both white-box and black-box attacks. These attacks only represent two extremes. In many practical scenarios, the attacker has partial knowledge of the defender's classifier. The authors in [32]–[34] analyzed AML attacks that require prior knowledge (exact or probabilistic) of the channel state between the attacker and defender, *assuming that the attacker knows the DNN architecture (including the trainable weights and loss function) used by the defender's classifier*. Note that due to differences in the dynamics of the transmitter-attacker and transmitter-defender channels, the benign signal seen by the attacker will be different from the one seen by the defender, which will result in different trained weights even for the same DNN (ultimately, impacting the effectiveness of the attack, as later shown in our simulations).

Prior works on RF signal classification have not extensively examined differences in the *hyperparameters* of the DNN structures, even when such structures are trained under the same data. Our study examines both aspects (differences in the input as well as differences in the hyperparameters). In particular, we observe that knowledge of the defender's classifier plays an important role in the strength of the attack. Intuitively, the attack is stronger when both the attacker and defender apply the same DNN than when they use different DNNs. Even under the same DNN architecture, differences in the hyperparameters can also affect the attacker's effectiveness (even when the attacker and defender use the same training and testing datasets). For example, if two CNNs differ in filter sizes at the Cov2D layer(s) or in the number of layers, the attack can be less effective. We further observe the attack's effectiveness is reduced even when the defender and attacker apply the same DNN but train it with different seeds. This implies that knowledge of defenders' DNN structure is critical for AML attacks. In our work, we first examine the impact of AML perturbations under a white-box model. We use the results as a reference point to evaluate other attack scenarios where the attacker has partial knowledge of the defender.

Previous works (e.g., [28]–[34]) primarily focused on *modulation classification* attacks. Such works used CNN-based classifiers as examples but did not consider sequence-to-sequence models such as RNNs [16], [17]. Our paper evaluates both *protocol* and *modulation* classifiers, considering CNN- and RNN-based designs. We start with FGSM, as a simple technique to generate AML perturbations [23]. We then extend the treatment to multi-step attacks by considering Projected Gradient Descent (PGD) [24] and DeepFool [25]. We evaluate these attacks under different knowledge levels for both modulation and protocol classifiers.

The authors in [32]–[34] considered the problem of synchronization between the attacker and defender. The syn-

chronizing problem in our paper differs from theirs in two main aspects. First, we focus on studying the synchronization issue for *input-dependent* AML attacks (e.g., FGSM, PGD, and DeepFool). In contrast, the shift-invariance property demonstrated in [32]–[34] pertains to *input-independent* attacks, e.g., Universal Adversarial Perturbation (UAP). In the UAP attack, the same matrix of perturbations is generated for all different benign inputs. This matrix effectively fools all inputs with high probability [35]. Consequently, the defender receives the same perturbation for all inputs, a notable contrast from the perturbations we examine in our paper. Second, UAP, being input-independent, allows the generated attack on one window to be effective on other windows, as demonstrated in [32]–[34]. In our paper, we refer to this coarse-scale misalignment as *inter-window shift*. However, the misalignment can also be a fraction of a window, a scenario we refer to as *intra-window shift*. In this case, the shift-invariance property of the UAP attack is no longer valid. For *input-dependent* AML attacks, we study the impact of both inter- and intra-window shifts.

Finally, we propose a two-step defense mechanism to improve the robustness of the defender's classifier to AML attacks. Our defense approach relies on training multiple classifiers with various adversarial examples [23], each at a given level of perturbations. During normal operation (testing phase), a separate DNN-based estimator is used to predict the level of perturbations of the AML attack (including the possibility of no attack). Subsequently, one of the retrained classifiers is selected for robust signal detection.

Our contributions are summarized as follows:

- In addition to modulation classifiers, we extend the study of AML attacks to protocol classifiers used in spectrum-sharing scenarios (prior work focused only on modulation classification). In contrast to [28]–[34] where only a CNN classifier was studied, in our work we consider two CNNs and three RNNs (e.g., LSTM and bidirectional LSTM).
- In contrast to previous work, which considered two extreme cases of the attacker's knowledge (i.e., white-box and black-box attacks), our paper studies a range of (partial) levels of knowledge.
- We study AML attacks under practical considerations of a typical wireless network setting, including unsynchronized transmitter/attacker operation, non-persistent AML perturbations, and channel degradations. We evaluate the attacks under various imperfections and show that these attacks can still significantly reduce the defender's accuracy.
- We propose a defense approach based on enhanced adversarial training. Traditional adversarial training relies on retraining a single classifier under a particular attack setting, and hence is not effective under other attack settings. Our proposed defense mechanism shows better robustness and improves the defender's accuracy by 30-50% compared to conventional adversarial training.

II. System Model

We consider a wireless communication system that consists of a legitimate transmitter-receiver pair and an adversarial device (see Figure 1). The transmitter generates RF signals according to one of several possible protocols (for protocol classification) or modulation schemes (for modulation classification) in an interleaved manner, i.e., one protocol or modulation scheme is active at a time. Without loss of generality, we assume that the defender's classifier resides within the legitimate receiver[1]. This classifier is trained to identify the protocol (or modulation scheme) based on the *received* baseband I/Q samples, which we refer to as *benign data* or *benign input*. The attacker generates its perturbations based on overheard benign data. These perturbations interfere with the defender's classifier, pushing it into wrongly classifying the received samples. We refer to the combined benign data plus perturbations as *adversarial data*.

The output of the defender's classifier is represented by the mapping $z = g(x;\theta)$, where x is a window of I/Q samples and θ is the set of learnable DNN parameters, i.e., weights and biases. The input x is in $\mathbb{R}^{2\times N}$, where N is the window size (in consecutive samples) and the first (second) row represents the sequence of I (Q) values, respectively. The input matrix x is passed through the DNN and is represented by a feature vector resulting from a projection and nonlinear (activation) function, $\sigma(\cdot)$. The classifier assigns a label $f(x;\theta) = \arg\max_k(\sigma(z)_k)$ to the received input, where $k \in K$ and σ is a softmax function. In this formulation, $\sigma(z)_k$ is the numerical output of classifier f corresponding to the kth protocol (or modulation) type.

At any given time, let $\mathbf{H}_{td}$ be the channel matrix from the legitimate transmitter to the defender, $\mathbf{H}_{ta}$ be the channel matrix from the legitimate transmitter to the attacker, and $\mathbf{H}_{ad}$ be the channel matrix from the attacker to the defender. We assume AWGN $\{n_d\}$ and $\{n_a\}$ at the receive chains of the legitimate receiver (defender) and attacker, respectively. In the absence of AML perturbations, the defender receives $x_d = \mathbf{H}_{td}x_{t'} + n_d$, where $x_{t'}$ is the transmitted waveform. The attacker receives $x_a = \mathbf{H}_{ta}x_{t'} + n_a$. The adversary uses its signal x_a to generate and transmit AML perturbations η. In the presence of AML perturbations, the defender receives $x_d^* = \mathbf{H}_{td}x_{t'} + \mathbf{H}_{ad}\eta + n_d$. We introduce a variable τ to indicate the time lag between the arrival of the benign signal at the defender and the arrival of the corresponding AML perturbations. Accordingly, the signal received by the defender becomes $x_d^*(\tau) = \mathbf{H}_{td}x_{t'} + \mathbf{H}_{ad}\eta(\tau) + n_d$.

Several approaches can be used to generate η. Such approaches were studied in the context of computer vision and natural language processing. In this paper, we apply these approaches in the context of RF signal classification. Specifically, the attacker seeks to determine AML perturbations that, when combined with the original signal, fall within an ℓ^∞ ball determined by ϵ and that maximize the classification

[1]We use the legitimate receiver and the defender interchangeably in our paper.

error. More formally, the adversary would ideally solve:

$$\max_{\eta} \ \mathbb{I}\{f(x_d;\theta) \neq f(x_d^*;\theta)\} \quad s.t. \ \|\eta\|_\infty \leq \epsilon \tag{1}$$

where $\mathbb{I}$ is an indicator function that reflects the number of misclassified labels in a given training set. We seek the smallest possible perturbations. To achieve this goal of finding the perturbation efficiently, we add a constraint on η. Note that $\epsilon > 0$ is a user-defined parameter that limits the power of the perturbation and ensures the attack is difficult to identify by the defender. Instead of constraining η, one can also attempt to find the minimal η that is sufficient to change the estimated label. This is done by solving the following minimization problem:

$$\eta^* = \arg\min_{\eta} \|\eta\|_\infty \quad s.t. \ \ f(x_d;\theta) \neq f(x_d^*;\theta). \tag{2}$$

This type of perturbation, proposed by Moosavi-Dezfooli *et al.* [25] is called the DeepFool attack.

III. DNN Structures

This section discusses the DNNs we consider for protocol and modulation classification, as well as the datasets used to train and test them.

A. DNNs for Protocol Classification

We consider four DNN structures for protocol classification, as shown in Figure 2. Three of these structures are stacked RNNs, each made of dense layers as well as Long Short-Term Memory (LSTM) and/or bidirectional LSTM (BiLSTM) layers. The last DNN is a CNN, modified from *LeNet* [36] by replacing the *Conv2D* of LeNet with *Conv1D* layers to efficiently transform and extract features from the time-domain sequence. In addition, we remove the padding layer from *LeNet* to improve the accuracy. The kernel size for the *Conv1D* layer is set to two, and its stride is set to one. The activation functions for the *Conv1D* and the fully connected layers are scaled exponential linear units. The output layer in each classifier is soft-max. To train and test the protocol classifiers S_1 to S_4, we generate a dataset of 15,000 inputs (see Section VI), each containing 512 pairs of I/Q samples. AWGN is added to the samples to achieve a given signal-to-noise ratio (SNR)[2]. Approximately 60% of the dataset is used for training, 20% for validation (i.e., early stopping, hyperparameter tuning, etc.), and 20% for testing. We monitor the cross-entropy and use early stopping with a patience of three.

B. DNNs for Modulation Classification

We also consider the modulation classifier proposed by O'Shea *et al.* and apply it to the RML 2016.10a dataset [37]. We abbreviate O'Shea *et al.*'s DNN as VT-CNN2. VT-CNN2 is a four-layer CNN that uses two convolutional layers and two fully connected layers. The hidden layer activations are Rectified Linear unit (ReLu). The output layer is a soft-max. The RML 2016.10a dataset comprises 220,000 data segments (i.e., windows), representing 11 modulation schemes. There are 20 SNR values that range from -18 dB to 20 dB in steps of 2 dB. This results in 1,000 windows of samples per modulation scheme per SNR. We use 50% of the data for training, 5% for validation and early stopping, and 45% for testing. The RML 2016.10a dataset is available in windows of 128 samples (I/Q pairs) each, with a stride of 64 samples, i.e., two successive windows overlap by 64 samples.

[2]Unless specificied otherwise, the SNR in this paper refers to the SNR for the Tx-attacker channel, i.e., SNR_{T-A}.

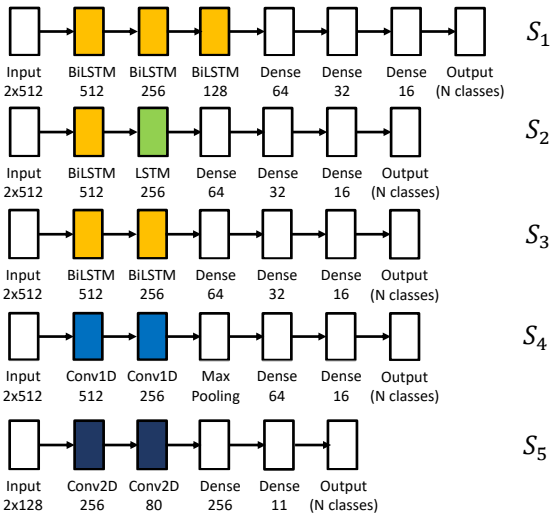


FIGURE 2: DNNs considered in this work for RF signal classification. Structures S_1-S_3 are RNN-based classifiers, structures S_4 is a LeNet-based CNN classifier, and structure S_5 is the VT-CNN2 classifier.

IV. Adversarial ML Attacks

We consider three different approaches for generating adversarial data: FGSM, PGD, and DeepFool. Although other approaches have been proposed in the literature, these three are often applied to wireless communication systems.

A. FGSM Attack

FGSM uses the gradients of a DNN to generate a perturbation η and, subsequently, the adversarial data $x_{adv} = x + \eta$ [23]. Ideally, the defender would predict the same class for x and x_{adv} if η is less than the given precision. However, the adversary can craft η and cause the defender's classifier to change its decision on the perturbed data. We denote the DNN's mapping function as $f : \mathbb{R}^{2\times N} \mapsto [0,1]^K$ with parameters θ. Even though the difference between x_{adv} and x is the small perturbation η, the difference $f(x+\eta;\theta) - f(x;\theta)$ is not linear in η. In fact, the impact of η can be learned and amplified by FGSM to change the label sign by calculating backpropagated gradients. The adversarial perturba-

tion is formally given by $\eta = \epsilon \operatorname{sign}(\nabla_x L(x, y; \theta))$, where $L(x, y; \theta)$ is the loss function of the classifier (typically, cross-entropy) with parameters θ [23]. The adversarial data are generated by maximizing the loss with respect to the classifier's input x and true label y based on the gradients $\nabla_x L(x, y; \theta)$. The authors in [29] proposed a new parameter ϵ_{acc} for adapting ϵ during the generation of the FGSM perturbations. In Section VII.F, we compare the results in [29] with the unmodified FGSM approach and show that both versions of FGSM lead to reducing the defender's accuracy.

B. PGD Attack

FGSM can be interpreted as a one-step approach to maximize the impact of the perturbations. PGD is a more powerful variant of FGSM that uses multiple steps to project the gradient on the negative loss function [24]. We consider a constraint set $\mathcal{Q}$ for perturbation power ϵ. Starting from the initial point x_0, PGD iterates over the equation $x_{t+1} = P_{\mathcal{Q}}(x_t + \alpha \operatorname{sign}(\nabla_x L(x, y; \theta))$ until a stopping condition is met, where $P_{\mathcal{Q}}$ is a projection operator that ensures that the output satisfies the constraint and t is the iteration number, $t = 0, 1, 2, \ldots, T$. In other words, PGD generates the perturbation in T iterations using a step size α. Clearly, the choice of α and T significantly impacts the performance of the PGD attack. Section VII studies the classification performance of PGD-based perturbations under different α and T.

C. DeepFool Attack

In DeepFool [25] ϵ is not set a priori; instead, the adversarial perturbation is determined by the smallest η needed to change the label $f(x; \theta)$. We can calculate the perturbation for x as in Equation (2). The same notation for $f(x; \theta) = \arg\max_k(\sigma(z)_k)$ is used as in Section II. To show the changes in $\sigma(z)$ with t, let $\sigma(g(x; \theta))$ be the output activation function that generates K outputs corresponding to the number of classes. DeepFool continues until the accumulative perturbation η changes the input's label. For multi-class problems, DeepFool updates the gradient changes between all other labels and the label that the target model predicts, and chooses the label with the smallest change as the direction to accumulate the perturbation. To find the closest possible perturbation that would mislead the classifier, we need to calculate the gradient of $\sigma(g(x; \theta))$. Therefore, this work considers the perturbation vector directed to the decision boundary between the originally predicted label and a fake label $\hat{y}$. The perturbation at each t can be written as: $\eta_t \leftarrow \frac{|\sigma(g(x;\theta))'_{\hat{y}}|}{\|\nabla\sigma(g(x;\theta))_{\hat{y}})\|_2^2} \nabla\sigma(g(x;\theta))_{\hat{y}}$. DeepFool returns η as the sum of perturbation at each step (η_t). The DeepFool algorithm is summarized in Algorithm 1.

D. Energy of Perturbations

In all the previously discussed perturbation methods, the parameter ϵ controls the power (or energy) of the perturbations. This ϵ is sometimes called *the adversarial budget*

Algorithm 1 DeepFool Attack (Multi-Class Classification)

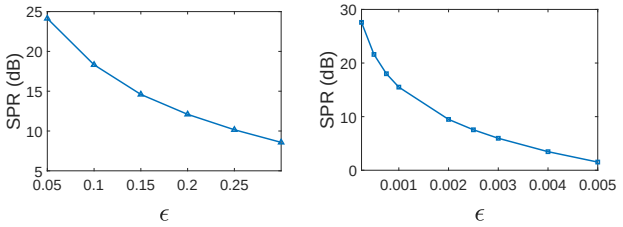
Input: Input x, classifier f
Output: Perturbation η

```
Initialize t ← 0, x_t ← x
while f(x_t; θ) = f(x; θ) do
    for k ≠ f(x; θ) do
        ∇σ(g(x_t; θ))_k ← ∇σ(g(x_t; θ))_k − ∇σ(g(x_t; θ))_{f(x;θ)}
        σ(g(x_t; θ))'_k ← σ(g(x_t; θ))_k − σ(g(x_t; θ))_{f(x;θ)}
    end for
    ŷ ← arg min_{k≠f(x;θ)} |σ(g(x_t;θ))'_k| / ||∇σ(g(x_t;θ))_k||_2^2
    η_t ← |σ(g(x;θ))'_ŷ| / ||∇σ(g(x;θ))_ŷ)||_2^2 ∇σ(g(x;θ))_ŷ
    x_{t+1} ← x_t + η_t
    t ← t + 1
end while
return η = Σ_t η_t
```

[38]. A larger ϵ implies that the perturbations can have a larger impact on the input, which results in a lower classification accuracy of the adversarial dataset. A larger ϵ means the adversary requires more energy. To reflect the energy level of the perturbation, we define the Signal-to-Perturbation Ratio (SPR) as the energy ratio between the received signal and the perturbation: $E(x)/E(\eta)$, where $E(x)$ is the average signal energy that received by the defender before the additive perturbation: $E(x) = \sum_{n=1}^{N} \frac{1}{N}|x[n]|^2 = \sum_{n=1}^{N} \frac{1}{N}(\mathrm{Re}\{x[n]\}^2 + \mathrm{Im}\{x[n]\}^2)$. $\mathrm{Re}\{x[n]\}$ and $\mathrm{Im}\{x[n]\}$ correspond to the I/Q values contained in the nth input sample. $E(\eta)$ is the energy of the perturbation generated by the attacker without including the channel impact between the attacker and defender. The relationship between SPR and ϵ is not in closed form because the energy of each window of samples varies from one window to another, and the perturbation vector differs for each class of data. As a result, to obtain the SPR as a function of ϵ, we must first generate the perturbations and then compare the average energy between the benign signals and perturbations. We show such relationships in Figure 3(a)-(b). It can be observed that the SPR drops quickly with ϵ at the beginning. This trend slows down when ϵ is large.

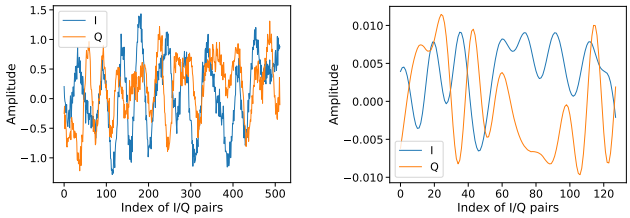
Recent research proposed ML approaches for detecting low-power interference [39], [40]. According to the method in [39], an adversarial signal can be detected when the interference power is 10 dB below the benign signal. Therefore, the adversarial perturbations will be hidden if the SPR exceeds 10 dB. According to Figures 3(a)-(b), an SPR > 10 dB corresponds to $\epsilon < 0.25$ and $\epsilon < 0.002$, for the protocol and modulation datasets, respectively.

Indeed, the range of values for ϵ depends on the specific dataset used. In our experiments, the samples in the two datasets exhibit significantly different amplitudes, as shown in the examples in Figure 4. Thus, for the same ϵ, the impact of the perturbations will be greater on the modulation classifier than on the protocol classifier. This is why we



(a) (b)

FIGURE 3: Relationship between the SPR and ϵ for : (a) protocol classification dataset, and (b) RML 2016.10a dataset.



(a) (b)

FIGURE 4: Amplitude for a segment of I/Q samples: (a) 5G waveform in the protocol dataset, (b) 64-QAM waveform in the RML 2016.10a dataset.

evaluated AML attacks on the modulation classifier using $\epsilon \in [0.0005, 0.005]$ and on the protocol classifier using $\epsilon \in [0.05, 0.3]$.

V. Adversarial Attacks with Limited Knowledge

In this section, we assess the impact of the attacker's knowledge of the defender's classifier on the effectiveness of an AML attack, considering the aforementioned three techniques for generating AML perturbations. A white-box attack (full knowledge) is expected to cause the most degradation in the defender's classification accuracy. What is less clear is how much reduction in the attack's effectiveness, if any, results from limiting the amount of information available to the attacker. Accordingly, we consider scenarios where the attacker possesses only partial information about the defender. We divide such knowledge into classifier and data domains, and consider different levels of knowledge for the attacker in both domains. Under partial knowledge, the attacker's DNN ends up being different in structure and/or trainable weights than the defender's DNN. Even with such differences, our results show that the attack is still effective, but such effectiveness depends on the similarity between the surrogate and defender classifiers. This observation confirms the concept of *attack transferability*, defined as the ability of an attack generated using one DNN classifier to impact the performance of another DNN classifer [41], [42]. However, the level of transferability is a function of the dissimilarity between the two DNNs. Recent studies [43], [44] corroborate our findings, prompting the authors of these works to suggest applying transformations and input diversity during the training of the attacker's DNN so as to improve the efficacy of attack transferability.

A. Limited Knowledge of Defender's Classifier

We consider realistic scenarios in which the attacker trains a classifier $f_a(x;\theta_a)$ that is not identical to the defender's classifier $f_d(x;\theta_d)$. In this case, the loss can be represented by $L(x_d^*, y_a; \theta_a)$ because the label for the corresponding input needs to be estimated by the attacker's classifier. The difference between $f_a(x;\theta_a)$ and $f_d(x;\theta_d)$ has a direct impact on the loss function. We study the following four levels of the attacker's knowledge and test their impacts on the perturbations.

Attack A_1: In this scenario, the attacker knows the hyperparameters of the defender (i.e., the network type, number of hidden neurons, activation functions, etc), but does not know the exact values of the defender's trained weights. This may result from using different random initializations or different learning rates during the training. As a result, the adversary's and defender's classifiers will have different weights and biases even if they have the same classification performance. For our simulations, we use two different sets of random seeds to initialize two classifiers before training them and keep all other settings the same.

Attack A_2: In this attack, the adversary knows the overall structure of the defender's DNN but does not know other hyperparameters. For example, the attacker may know that the defender's classifier uses a seven-layer CNN model with *Conv1D* as the first two layers, but the attacker does not know the filter numbers of these layers. For our simulations, we assume that the attacker knows the number of layers, their types, and their order but does not know these layers' filter numbers (or unit numbers for RNNs).

Attack A_3: In this case, the attacker knows the type of classifier that the defender uses (e.g., CNN or RNN), but not its structure. To study this attack, we use a differently structured classifier of the attacker to generate the adversarial perturbations. Sometimes, we consider the same type of DNN but with different layer numbers (e.g., we use a three-layer RNN structure S_1 for the defender but use a two-layer structure S_3 for the attacker).

Attack A_4: In this attack, the attacker knows nothing about the defender's classifier. The mapping function f_a can differ significantly with classifier types, especially if a CNN represents features differently than an RNN. In this scenario, we consider the situation when the attacker uses RNN structure S_1 as the classifier to generate the adversarial perturbations. Still, the defender uses the CNN structure S_4 as the detector and vice versa.

B. Limited Knowledge of Defender's Training Data

In a practical wireless setting, the benign samples received by the attacker, $x_a = \mathbf{H}_{ta} x_{t'} + n$, and those received by the

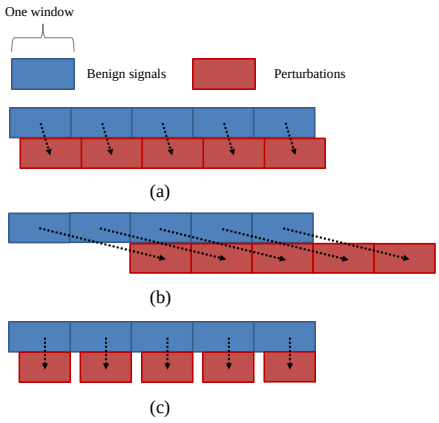


FIGURE 5: Examples of imperfect synchronization and incomplete sequences of perturbations. (a) Intra-window perturbation shift, (b) inter-window perturbation shift, (c) Incomplete-window perturbations.

defender, $x_d = \mathbf{H}_{td}x_{t'} + n$, are different due to channel impact. The attacker trains its classifier f_a based on the dataset x_a. Because the training data sets at the attacker and defender differ, the parameters θ_a and θ_d will differ. As a result, the adversarial perturbations must be generated with $f_a(x;\theta_a)$. The loss function $L(x_d^*, y_d; \theta_d)$ is approximated by $L(x_a, y_a; \theta_a) + \eta^T \nabla_{x_a} L(x_a, y_a; \theta_a)$. We denote this type of attack as A_{tr}.

C. Imperfect Synchronization between Perturbations and Benign Data at Defender

Due to differences in propagation delays, as well as processing delays of benign data at the attacker, the adversary cannot guarantee that its perturbations will be perfectly synchronized with the benign data received by the defender [37]. We study such imperfect synchronization and analyze its impact on the defender's classification performance. In our setup, the defender's waveform is sampled by a fixed-length moving window before being sent to the classifier. Therefore, we consider two situations: intra- and inter-window shifts, as shown in Figures 5(a) and 5(b). Note that our study of the impact of imperfect synchronization does not mean that the defender has any way to control or even estimate the degree of mis-synchronization.

D. Incomplete Sequences of Perturbations

The adversary may act intermittently to prevent being detected, generating its perturbations for only a fraction of the time, as shown in Figure 5(c). In this setting, the attacker listens to the channel at the beginning of the transmission and sends part of the perturbation to be superposed with the benign signal at the defender. For simplicity, we assume that during its active periods, the attacker's perturbations are synchronized with the benign signal.

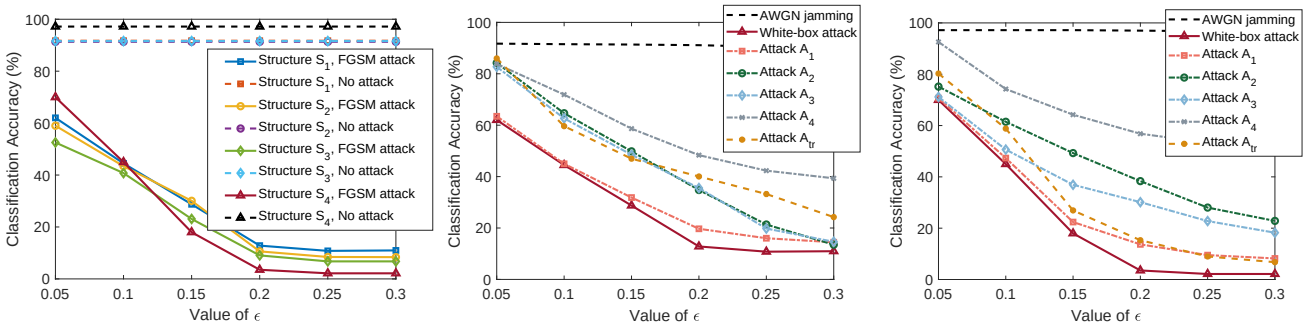
E. Limited Energy Ratio between Perturbations and Channel Impact

Previously, we depicted the relationship between the SPR and ϵ before accounting for channel effects (see Figures 3). We also study the channel impact between the attacker and the defender, assuming AWGN channels. In this case, the total interference received by the defender is the sum of the adversary's perturbations and the channel noise. The Perturbation-to-Noise Ratio (PNR) was introduced to measure the relationship between the transmitted power of the adversarial perturbations and the noise/fading of the channel between the attacker and defender. In Section II, we expressed the perturbations received by the defender as $\mathbf{H}_{ad}\eta + n_d$. The PNR, denoted as $E(\mathbf{H}_{ad}\eta)/E(n_d)$, is averaged over all received baseband I/Q pairs. To evaluate the channel impact between the attacker and defender, we treat the received signal at the defender without attack as benign. Note the benign signals already include the AWGN noise between the transmitter and defender. To further determine the channel noise between the attacker and defender, we use the energy of the benign signals as the reference and vary such channel noise in several levels. After determining the channel noise between the attacker and defender, we further vary perturbations to evaluate their impact under different PNRs. The SNR in the attacker-defender channel is related to SPR and PNR as SNR = SPR × PNR or, equivalently, SNR [dB] = SPR [dB] + PNR [dB]. Therefore, if the attacker wants to ensure the perturbations are undetectable, it should have a PNR value below SNR -10 dB.

VI. Datasets

For protocol datasets, the *Matlab Wi-Fi, LTE*, and *5G Toolboxes* were used to generate signals. Of the various possible features, we use the baseband I/Q samples at the defender (with AGWN) as input to the classifier. I/Q samples are obtained before decoding the signal, providing a rich representation of the actual waveform. The simulated waveforms are divided into multiple sequences by applying a sliding window with a step size of one, each consisting of 512 I/Q pairs. Simulated transmissions are sent at the same center frequency, over a 20 MHz channel. In addition, we consider the LTE, Wi-Fi, and 5G NR as the classes of signals transmitted under an AWGN channel with SNR = 15 dB. The Wi-Fi waveforms are transmitted by generating baseband samples of 802.11ac (VHT) with BPSK modulation and $1/2$ coding rate. The LTE waveforms are generated by downlink with reference channel R.9, which uses a 64 QAM modulation. We also generate 5G waveforms using 5G DL FRC with QPSK modulation and a coding rate of $1/3$ with a subcarrier spacing of 15 kHz. These sequences form the datasets to train and test the four protocol classifiers. We generate a dataset of 15,000 inputs, with approximately 5,000 samples for each label (Wi-Fi, LTE, and 5G).

In addition to the 15,000 windows of samples, we also consider a much larger set of 220,000 windows. Specifically,



(a) (b) (c)

FIGURE 6: Accuracy of proposed DNN classifiers under benign and FGSM-based perturbations: (a) All four DNNs under white-box attacks, (b) RNN under limited-knowledge attacks, (c) CNN under limited-knowledge attacks.

to better illustrate the impact of adversarial perturbations on classification accuracy, we consider the publicly available RML 2016.10a dataset for modulation classification [37]. This dataset comprises noisy I/Q samples for 11 modulation schemes: 8PSK, BPSK, QPSK, QAM16, QAM64, CPFSK, GFSK, PAM4, WBFM, AM-DSB, and AM-SSB. Each modulation scheme is represented in 1,000 windows of samples for each given SNR, with the SNR varying from -18 dB to 20 dB in steps of 2 dB. Thus, the RML 2016.10a dataset includes 220,000 windows of samples (20,000 windows per modulation scheme), each consisting of 128 I/Q pairs.

VII. Performance Evaluation

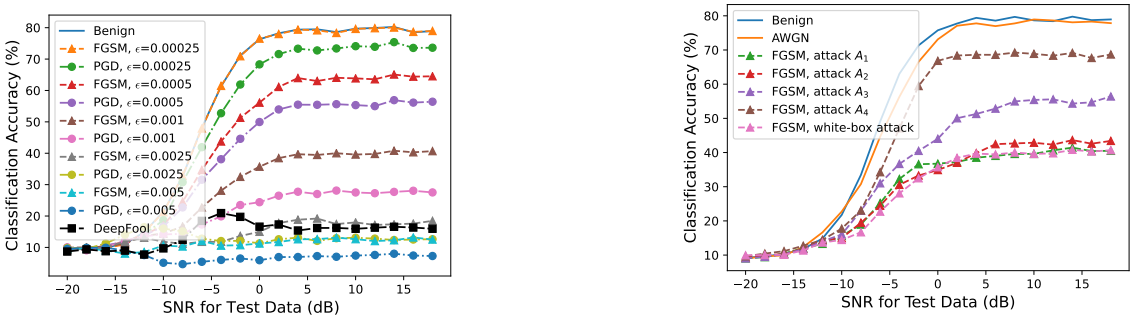
In this section, we evaluate the impact of FGSM, PGD, and DeepFool attacks when the attacker possesses different knowledge levels about the defender. We then test the impact of mis-synchronization attack, persistence, and channel noise, considering FGSM as a representative example. We apply our evaluation to both protocol and modulation datasets.

A. FGSM Attacks

Figure 6 depicts the classification performance at the defender vs. ϵ, considering FGSM attacks on the protocol dataset. As shown in Figure 6(a), the RNN structures S_1-S_3 achieve approximately 91% accuracy under benign AWGN perturbations. In contrast, the CNN structure S_4 achieves 97% accuracy (refer to the dashed lines for benign performance). The three RNN structures S_1-S_3 have comparable performance but have various bidirectional LSTM designs. The accuracy drops for all four classifiers as we increase the budget of the adversarial FGSM perturbations via ϵ. Note these are white-box attacks where the adversary is capable of the most damage. We also observe that structure S_1 has the highest average accuracy over all ϵ settings among the three proposed RNN models. Therefore, we use structure S_1 in later evaluation to represent the RNN classifier. Even though the CNN performs best under benign perturbations, it suffers more from AML attacks. When ϵ exceeds 0.1, the CNN model performs the least accurately among the different structures. All the models' accuracy saturates when ϵ is higher than 0.2, indicating that the white-box attack can mislead the defender's classifier with limited power control. These results demonstrate an accurate classifier is not necessarily a *robust* classifier.

After evaluating the white-box attacks, we consider attack scenarios where the attacker has incomplete knowledge (as described in Section V) of the defender's classifier and/or the training dataset used by the defender. The accuracy for RNN (i.e., structure S_1) is shown in Figure 6(b). The impact of attack A_1 is close to the white-box attack. This result is expected because the attacker has the same hyperparameters as the defender. Although the classifiers are trained with different seeds, one can still inherit most of the properties from the other. Attack A_2 exchanges the filter number of the first two layers, and attack A_3 uses one less layer (e.g., remove the third layer of structure S_1) for the attacker. Both show similar performance as the defender, which means these hyperparameters are relatively important for generating adversarial perturbations. Attack A_4 has the weakest attack effect. This is because the attacker applies the CNN structure S_4 to generate the adversarial signals for the RNN model (i.e., the adversary does not know the structure of the defender). Even though both classifier types can classify the waveforms accurately, the actual trained model differs significantly from the others. Therefore, a well-crafted perturbation for the CNN may not achieve the expected effect on RNNs. Attack A_{tr} uses the different training datasets to generate the perturbations. Thus, it shows more variance than other attacks. It has an equivalent trend with attacks A_2 and A_3, but slows when ϵ exceeds 0.15.

The accuracy of CNN (i.e., structure S_4) is shown in Figure 6(c). Similar to the RNN observations, the attack's impact depends heavily on the adversary's level of knowledge about the defender. In the simulation, attack A_2 exchanges the filter number of the two *Conv1D* layers, and attack A_3 removes the second *Conv1D* layer at the attacker side. Compared to the RNN, the layer and filter number



(a) (b)

FIGURE 7: Classification accuracy vs. SNRs for VT-CNN2 using RML 2016.10a dataset. (a) FSGM, PGD, and DeepFool under white-box attacks (several values for ϵ are considered), (b) FSGM and AWGN under limited-knowledge attacks (ϵ = 0.001).

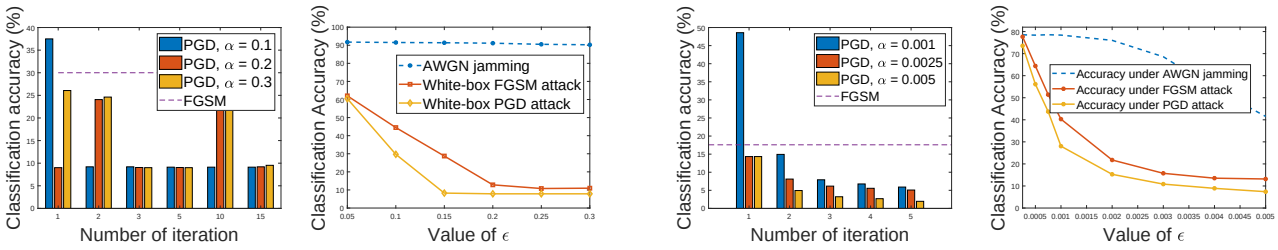
setting play a more critical role in CNNs. As a result, attacks A_3 and A_4 show different trends with varying ϵ. In contrast, attack A_{tr} shows a strong similarity with attack A_1, which implies the CNN model can suffer a more severe attack than the RNN, even when the attacker has limited knowledge of the data.

We then show the impact of FGSM under the white-box attacks using the RML 2016.10a dataset. We use VT-CNN2 as the benchmark classifier for the defender. The adversarial budget, ϵ, varies from 0.00025 to 0.005. As ϵ increases, the perturbations exhibit higher power (i.e., lower SPR), and reduce more accuracy of the defender. We evaluate the defender under different SNRs and summarize the results in Figure 7(a). In addition to the white-box attack, we study FGSM perturbations under four limited-knowledge attacks. To keep the energy of the perturbation low, we explore the FGSM attacks with $\epsilon = 0.001$ as an example. As shown in Figure 7(b), limited-knowledge attacks A_1 and A_2 show close accuracy with the white-box attack. This result suggests that a small structure change may not heavily impact the FGSM adversarial signals for VT-CNN2 on RML 2016.10a dataset. However, when the attacker's knowledge is further reduced, the impact of FGSM becomes weaker (shown as the attacks A_3 and A_4). This indicates that the attack can be significantly weakened if the defender's knowledge is less than a certain level. However, these imperfect knowledge attacks are still stronger than AWGN with equivalent power.

B. PGD Attacks

Due to the CNNs and RNNs having similar accuracy trends under FGSM attacks as shown in Figure 6(b) and 6(c), we use RNN (structure S_1) as the classifier for the protocol dataset to show the remainder of the attack schemes. We first study the impact of step sizes and maximum iteration numbers on PGD-based perturbations. Under the white-box attack, we test the classification accuracy of the defender's classifier while fixing $\epsilon = 0.15$. Recall that the PGD attack is computed over multiple steps (iterations) of gradient descent, and is parameterized by ϵ and α. The parameter ϵ regulates the power budget (same as FGSM), whereas α controls the step size. α can be chosen from a wide range because the projection in PGD always pushes the perturbed signal into the constraints of ϵ, as described in Section IV.B. While a larger ϵ can strengthen the attack, a larger α does not guarantee a stronger attack, which was also observed in [45], [46]. Using the CIFAR-10 dataset, Croce and Hein [45] showed that when α is twice the value of ϵ, the PGD attack becomes weaker than when using smaller values of α. Figure 8(a) depicts the classification accuracy under PGD perturbations versus the number of iterations for three values of α with $\epsilon = 0.15$ (protocol dataset). The defender's accuracy does not decrease when α goes from 0.2 to 0.3 because α is larger than ϵ. We observe that PGD with $\alpha = 0.1$ achieves the lowest accuracy after ten iterations. Accordingly, we chose $\alpha = 0.1$ for PGD and evaluated this attack for different values of ϵ. Figure 8(b) shows the defender's classification accuracy under FGSM and PGD attacks. PGD attacks are stronger than FGSM attacks when ϵ ranges from 0.05 to 0.3. These results suggest that PGD may be more effective at generating perturbations.

Comparable trends are observed in VT-CNN2 using RML 2016.10a dataset. To ensure that PGD attacks result in perturbations with limited energy, we fix $\epsilon = 0.0025$, and vary α from 0.001 to 0.005. Figure 8(c) shows the classification accuracy for different values of α. When α is close to ϵ, the value of T has a visible impact on the effectiveness of the PGD attack, particularly when T increases from 1 to 2. After a few iterations, the impact becomes less significant. Similar trends are observed under the other two small values of α. Moreover, we evaluate the accuracy of the defender's classifier under attacks as a function of ϵ when testing SNR is 16 dB, as shown in Figure 8(d). FGSM and PGD are quite effective in degrading the defender's classification accuracy. As expected, the accuracy goes down with a larger ϵ. Generally, PGD is an iterative attack and can impact the classification accuracy more than the one-round FGSM attack. We compared and summarized



(a) (b) (c) (d)

FIGURE 8: Impact of α, number of iterations, and ϵ in the PGD attack: (a) Classification accuracy vs. number of iterations with α, (b) classification accuracy vs. ϵ for different attacks (α = 0.1), using the DNN structure S_1 and the protocol dataset. (c) Classification accuracy vs. number of iterations under various α, averaged over all SNRs, (d) classification accuracy vs. ϵ for different attacks (α = 0.01, SNR = 16 dB), for VT-CNN2 using RML 2016.10a dataset.

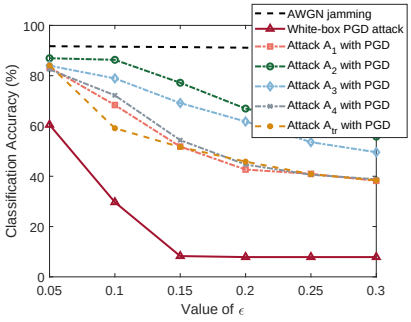


FIGURE 9: Accuracy of an RNN-based classifier (structure S_1) under different limited-knowledge PGD attacks.

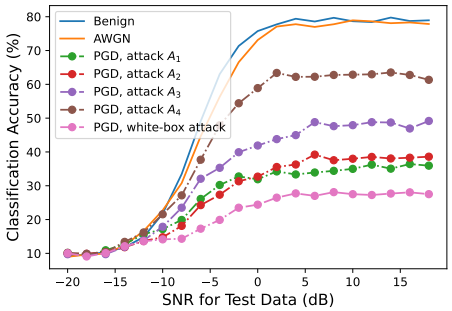


FIGURE 10: Classification accuracy vs. SNRs for VT-CNN2 using RML 2016.10a dataset under limited-knowledge PGD and AWGN attacks (ϵ = 0.001).

the impact of FGSM and PGD attacks in Figure 7(a), where we allow a sufficient iteration number for the PGD attacks for comparison. When ϵ is very small, PGD similarly impacts the classification performance as FGSM. As ϵ increases, the difference between PGD and FGSM is more pronounced. In our case, the accuracy gap between PGD and FGSM only grows when ϵ increases from 0.00025 to 0.001, but drops after that point (i.e., as ϵ further increases).

In addition to PGD under white-box attacks, we evaluate the limited knowledge adversary for the protocol dataset in Figure 9 and for RML 2016.10a dataset in Figure 10. Figure 9 compares the different knowledge levels of PGD attacks with $\alpha = 0.1$ and $T = 20$ to the AWGN attack. The PGD-based attacks significantly impact the defender's classifier when we allow a larger ϵ value. Similar to the FGSM trends, the limited-knowledge PGD attacks show a weaker impact. Attacks A_1 and A_{tr} are closer than other attacks. This performance is because they have the closest knowledge of the defender. Attacks A_2 and A_3 have similar performance as the defender, which is consistent with FGSM results in Figure 6(b).

In Figure 10, we explore the PGD attacks with $\epsilon = 0.001$, $\alpha = 0.01$, and $T = 20$, under different SNRs for RML 2016.10a dataset. We observe the attacks become weaker with less knowledge of the defender, similar to FGSM in Figure 7(b). The attacker in A_1 and A_2 loses a little information about the defender's classifier, and has the closest classification accuracy to the white-box attack. Attacks A_3 and A_4 become weaker due to the imperfect adversary's knowledge.

C. DeepFool Attacks

We first compare FGSM and DeepFool in terms of the defender's accuracy and SPR, assuming a white-box attack. A range of ϵ is considered for FGSM. DeepFool is not parameterized by ϵ, so it has only one entry in Table 1. From this table, we observe that FGSM with a larger ϵ reduces the defender's accuracy but requires more energy (lower SPR). This observation is in line with the observations in [30], [33]. FGSM with ϵ = 0.2 has the closest SPR to DeepFool's. Therefore, in Table 2, we fix ϵ to 0.2 and compare FGSM and DeepFool under different knowledge levels. The DeepFool attack results in a classification accuracy of 8.13%, compared to 12.82% for the FGSM attack. Even if we consider the FGSM attack with a higher ϵ value, for instance $\epsilon = 0.25$, which gives rise to a lower SPR, DeepFool is still a stronger attack.

Table 2 summarizes the SPR and accuracy under limited-knowledge attacks A_1-A_4 (previously defined in Section

Adversarial Scheme	SPR (dB)	Accuracy under Attack
FGSM with $\epsilon = 0.05$	24.14	62.03%
FGSM with $\epsilon = 0.10$	18.11	44.49%
FGSM with $\epsilon = 0.15$	14.60	28.74%
DeepFool	12.12	8.13%
FGSM with $\epsilon = 0.2$	12.10	12.82%
FGSM with $\epsilon = 0.25$	10.17	10.97%

TABLE 1: Comparison between DeepFool and FGSM with different ϵ values (white-box attack) using the protocol dataset.

Adversarial Scheme	SPR (dB)	Accuracy under Attack
DeepFool A_4	20.00	76.97%
DeepFool A_3	10.22	57.99%
DeepFool A_2	11.41	61.87%
DeepFool A_1	11.16	38.42%
FGSM A_4	12.10	48.35%
FGSM A_3	12.10	35.56%
FGSM A_2	12.10	34.78%
FGSM A_1	12.10	19.69%

TABLE 2: Comparison between DeepFool and FGSM with ϵ = 0.2 (limited-knowledge attacks) using the protocol dataset.

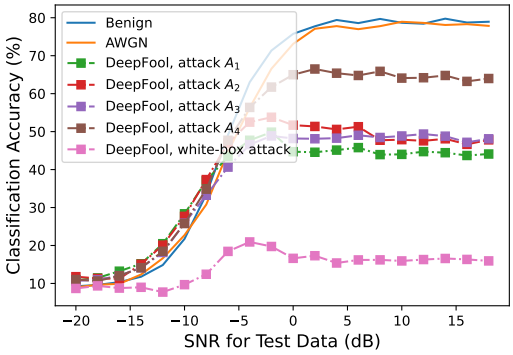


FIGURE 11: Classification accuracy vs. SNRs for VT-CNN2 using RML 2016.10a dataset under DeepFool and AWGN attacks (ϵ = 0.001).

V.A). We observe that the FGSM attack becomes more effective with more knowledge, as the defender's accuracy drops from 48.35% under attack A_4 to 19.69% under attack A_1. The SPR under limited-knowledge FGSM attacks remains the same because ϵ is fixed when generating the FGSM perturbations. In the case of DeepFool, although an attack with more knowledge is supposed to cause more harm, this is not always the case. For example, DeepFool attack A_3 is more impactful than DeepFool attack A_2, although it has less knowledge of the defender. Moreover, the SPR in DeepFool varies with knowledge levels since the attack does not have an ϵ parameter that can be directly controlled. While DeepFool's perturbations force classification errors at the defender, the attack is not guaranteed to be more effective than FGSM, especially in the limited-knowledge scenarios. Under limited knowledge, the difference between estimated and actual classifiers may be amplified during the iterations of the DeepFool algorithm. In attack A_1, even though we keep the same classifier structure for both attacker and defender, the different seeds for training initialization can still make the attacker's network slightly different in the final mapping function. As a result, the perturbation generated based on the attacker's classifier may not perform as expected on the defender's classifier. Attacks A_2, A_3, and A_4 are less effective than attack A_1. This is expected given that such attack scenarios consider less information about the defender.

From the point of view of interference power, DeepFool-based perturbations exhibit more fluctuations in their SPR. Specifically, in attacks A_2, A_3, and A_4, DeepFool perturbations exhibit lower SPR than their FGSM counterparts but are still less effective than FGSM in terms of degrading the accuracy. One justification for this observation is that DeepFool calculates the gradient changes for all the possible labels and chooses the shortest direction among these labels to update the perturbation at each step. However, the estimation of the boundary between different labels heavily relies on the anticipated outcome of the defender's classifier, which is only partially known by the attacker. As a result, the imperfect knowledge of the attacker can weaken DeepFool more than FGSM.

We further consider the DeepFool for VT-CNN2 on RML 2016.10a dataset and show the limited-knowledge attacks over all SNRs. As shown in Figure 11, DeepFool attack relies more on the information of the defender. Even DeepFool attack is stronger than FGSM and PGD with $\epsilon = 0.001$ under the white-box assumption, it becomes weaker with imperfect knowledge. The attacker has significant knowledge of the defender in attack A_1. Nevertheless, the reduction in performance is not as much as the white-box attack. The limitation of the knowledge weakens the impact of DeepFool. Even though, DeepFool can still outperform the AWGN attack on a similar power level by 15% in the worst situation (attack A_4). We then show that DeepFool attack has very low energy of the generated perturbation. The results of different attack schemes tested under 16 dB are summarized in Table 3. All these attacks can reduce the accuracy of the defender's classifier while maintaining the high SPR. Such reduction changes are based on the knowledge level of the attacker. When the attacker's classifier performs more similarly to the defender, the generated perturbations can be more effective. Under the least-knowledge attack A_4, the attacker uses the RNN classifier to generate the perturbation and applies it to the VT-CNN2 classifier on the defender side. It still decreases the classification by approximately 20% with minimal perturbation energy (i.e., the SPR is still high).

D. Impact of Synchronization

We evaluate the accuracy of the defender under intra- and inter-window shifted perturbations to simulate the imperfect

Adversarial Scheme	SPR (dB)	Accuracy under Attack
DeepFool A_4	14.59	63.23%
DeepFool A_3	8.34	47.17%
DeepFool A_2	13.18	46.64%
DeepFool A_1	15.65	43.70%
DeepFool white-box attack	15.74	17.32%

TABLE 3: Comparison between DeepFool attacks under different knowledge levels using the modulation dataset.

synchronization. The results on the protocol dataset are shown in Figure 12(a) and (b). Both intra- and inter-window shifts weaken the strength of the FGSM attack; however, the shifted FGSM attacks still degrade the performance further than AWGN. The equivalent AWGN means that the attacker transmits the AWGN noise instead of FGSM perturbation, where AWGN has the same energy as the FSGM attack under given ϵ. The intra-window shifted attack can be weakened a lot even only has one sample step shift, as shown in Figure 12(a). The shift between the signals and perturbations can further reduce the attack performance until the shift size reaches around 100 samples. Similarly, the first several steps for the inter-window shift have a more significant impact on the attack, as shown in Figure 12(b). When the shift step achieves around 50 windows, the effect of shifted attack starts to converge. In an actual attack, the attacker cannot control such synchronizations; however, our results can be used as the referring point to understand the impact of the asynchronization and estimate the defender accuracy for the attacker.

We further evaluate the impact of perturbation shifts on the RML 2016.10a dataset. We train classifiers with the data over whole SNRs and analyze the performance for testing data under different SNRs. We consider testing data with the highest SNR (18 dB) and use it as an example scenario to show the impact of synchronization, and later for completeness, and channel effect. Figure 12(d) and (e) show the impact of synchronization for RML dataset. We consider a smaller range of the sample shifts than the protocol classification dataset because the window length of the RML dataset is 128, other than 512. Similar to the protocol classification dataset results, the first several steps drop the attack strength a lot for the intra-window shift as shown in Figure 12(d). The shifted perturbations perform comparably when the shift step exceeds ten samples. The FGSM attack with low ϵ (e.g., ϵ = 0.001) has a similar effect as the equivalent AWGN attack when the intra-window shift is greater than ten steps. For the inter-window attack as shown in Figure 12(e), the effectiveness of the FGSM attack is reduced even with one window shift. However, the further shift does not degrade the attack more. This is because the testing data in the RML 2016.10a dataset is shuffled by default, and the impact of an inter-window shift larger than one step is the same as a random-step shift. The order of I/Q pairs is unknown, so the inter-window shifted perturbations are similar to the shuffle. Overall, the FGSM attack with larger ϵ suffers less for both the intra-window and inter-window shifts.

E. Impact of Completeness

In an ideal attack, the attacker can continue to send the streaming of perturbations that are superposed to the defender's signal. However, it can be stealthier if the attacker sends the perturbation discontinuously. The impact of the perturbation completeness for the protocol dataset is explored and summarized in Figure 12(c). The attack can still be effective even after losing some perturbation samples, especially when missing parts are less than 50. With more perturbations missing, the attack becomes weaker. Nevertheless, the incomplete attack with 300 samples losing is still more substantial than the equivalent AWGN attack (shown as dashed lines above). Note that our full sample length is 512 for the protocol classification problem, indicating that the AML attack with half perturbation interrupted is still effective. The impact of the completeness for RML 2016.10a dataset is summarized in Figure 12(f). Both the AWGN and FGSM attacks are impaired due to truncation. The impairment has a near-linear relationship with the number of missing samples when the missing amount exceeds ten. Even if the FGSM attack degrades with losing samples, it can still be more powerful than the AWGN attack with the equivalent energy.

F. Channel Impact

The efficacy of an AML attack depends on both the channel type (e.g., Raleigh fading vs. AWGN) as well as channel conditions. We assume that all three channels (Tx-attacker, Tx-defender, and attacker-defender) are AWGN, and we evaluate the effect of the channel conditions between the attacker and defender. To do that, we first obtain the power of the received (benign) signal at the input to the attacker based on the power of the transmitted benign signal and the given SNR value for the TX-attacker channel (SNR_{T-A}). For the protocol dataset, we set SNR_{T-A} to 15 dB during training and testing. For the RML 2016.10a dataset, AWGN is already embedded in the signal at different SNR_{T-A} values, so during training we use the average of all the samples in this dataset (over all SNR_{T-A} values) to determine the average power of the received benign signal. Testing of the modulation classifier is done at $\text{SNR}_{T-A} = 18$ dB (the highest SNR in the RML dataset). For both protocol and modulation classification, let β denote the ratio between the (average) power of the incoming signal at the attacker and the noise power of the attacker-defender channel. For a fixed β, (hence, fixed noise power, E_n, of the attacker-defender channel), we vary the power of the perturbations by varying the PNR. Recall that the 'N' in the PNR refers to the AWGN of the attacker-defender channel. Figure 13 below depicts the classification accuracy versus PNR for different values of β. It is clear from the figure that for a given β, the noise of the attacker-defender channel impacts the effectiveness

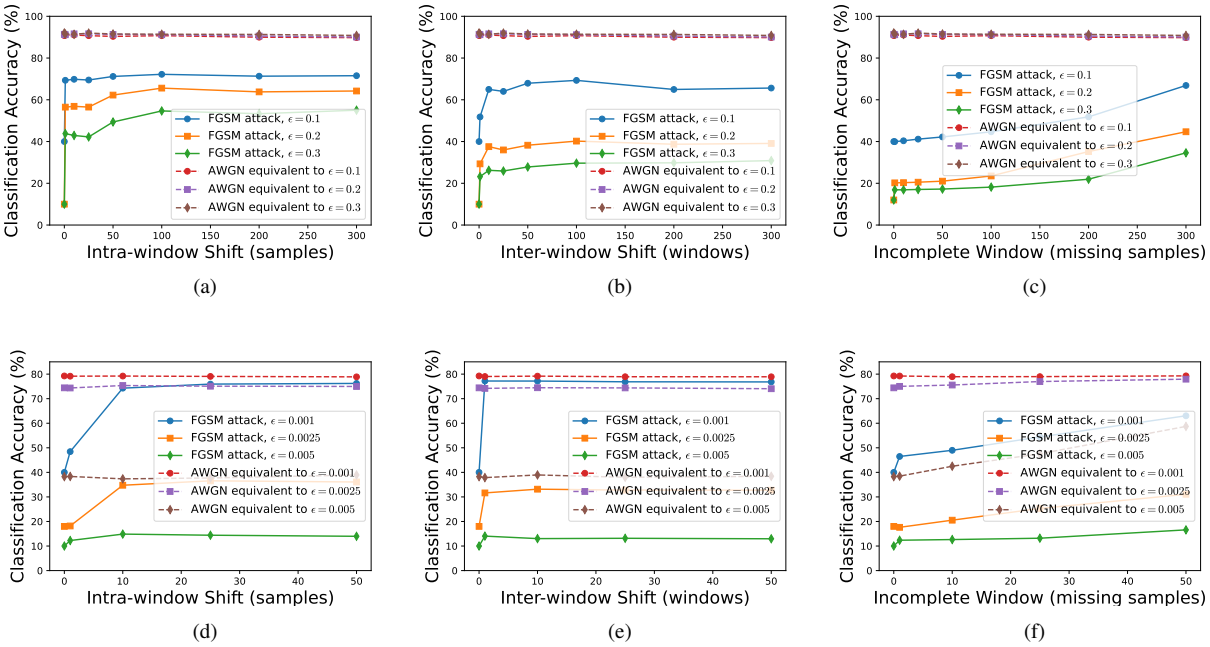


FIGURE 12: Impact of imperfect synchronization and incomplete sequences of perturbations: (a) Intra-window perturbation shifts (protocol dataset), (b) inter-window perturbation shifts (protocol dataset), (c) incomplete-window perturbations (protocol dataset), (d) intra-window perturbation shifts (RML 2016.10a dataset), (e) inter-window perturbation shifts (RML 2016.10a dataset), (f) incomplete-window perturbations (RML 2016.10a dataset).

of the attack. This can be observed for all values of β. Another key observation is that for small to medium values of β, the attack is still significant even at small PNR values. For example, when $\beta = 0$ dB (very noisy attacker-defender channel, relative to the power of the received benign signal) and a PNR of -5 dB (perturbations power is 5 dB less than attacker-defender noise power), the classification accuracy is about 20% for both protocol and modulation classifiers. Even with lower PNR values (e.g., -10 and -15 dB for the protocol classifier), the attack is still significant.

For both the protocol and RML 2016.10a datasets, we consider the β from 0 dB to 15 dB with a step size of 5 dB. The defender's accuracy reduces when PNR increases for all values of β. When β is low, the channel noise between the attacker and defender can degrade the classification accuracy even with slight perturbations. Channel noise here can be regarded as the traditional jamming attack. In Figure 13(a), such noise makes the accuracy of the defender drop to around 70%. As the β increases, the channel condition improves, and the defender's accuracy also rises. For example, when PNR is around -10 dB, the defender's accuracy performs better under larger β. As β increases, the channel noise decreases. As shown in Figure 13(b), when β = 15 dB, the defender has an accuracy of 80%, which aligns with the observation under the benign data.

As previously mentioned, the authors in [29] modified the FGSM attack and evaluated its performance under different SNR values. Their study was conducted using the RML

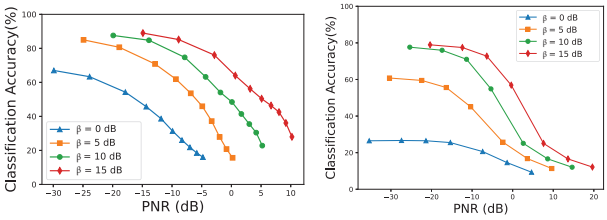


FIGURE 13: Accuracy of the defender classifier under different PNRs when the power of channel noise is fixed. (a) Protocol dataset with embedded AWGN noise (test $\text{SNR}_{T-A} = 15$ dB), (b) RML 2016.10a dataset with embedded AWGN noise (test $\text{SNR}_{T-A} = 18$ dB).

2016.10a dataset and the VT-CNN2 classifier, assuming an AWGN channel and a white-box attack. In their results (Figure 2 in [29]), the defender's accuracy dropped to 0% when SNR $= 10$ dB and PNR $= 0$ dB. Our results in Figure 13(b) show that the unmodified FGSM attack reduces the defender's accuracy to around 40% when $\beta = 10$ dB and PNR $= 0$ dB. This implies that even the (unmodified) FGSM algorithm can significantly reduce the defender's accuracy, although not to the level achieved by the ϵ adaption approach in [29]. Our findings on FGSM are aligned with other works, e.g., [31], which also showed the efficacy of the original FGSM attack. Note that channel information may be leveraged to design very effective (channel-dependent)

AML attacks, as done in [32], [33]. However, even when the technique used to generate the perturbations is channel-agnostic (e.g., the classical FGSM), our results above show that the attack is still impactful over a wide range of SNR and SPR values.

It is important to note that different studies in the literature were conducted under different simulation settings; some rooted in hardware experiments, while others consider specific channel models and types of attacks (e.g., UAP attacks). For instance, the study of channel effects in [32]–[34] is based on a Rayleigh fading model, whereas our study considers an AWGN channel. Intuitively, the success of an attack depends on both the channel model (e.g., AWGN vs. Rayleigh fading) as well as channel conditions. These disparities can lead to variations in the AML attack efficacy. For instance, unmodified AML attacks might become less effective in a fading channel; however, their potency increases if the attacker and defender are in close proximity. Consequently, a meaningful comparison necessitates applying a similar setup.

VIII. Defense Against Adversarial Attacks

In this section, we investigate several defense mechanisms against AML attacks. First, we provide a summary of related work on this topic.

A. Related Work on Defense Mechanisms

Recently, several defenses have been proposed against AML attacks on DNN models [47]–[53]. Olowononi *et al.* [47] presented an encryption mechanism to hide the DNN internal weights, parameters, and training data from an adversary. They also presented three techniques to improve the defender's robustness: input pre-processing, adversarial training, and post-processing. He *et al.* [48] evaluated adversarial training, randomization, defensive distillation, and gradient masking to defend against adversarial attacks. Adesina *et al.* [49] presented statistical approaches to monitor metrics such as the peak-to-average power ratio (PAPR), the distribution of softmax outputs of the DNN classifier, and median absolute deviation (MAD) of the data for adversarial signal detection. They also evaluated the efficacy of adversarial training and randomization to mitigate AML attacks. Of the various defense mechanisms proposed in the literature, adversarial training remains one of the most robust methods [54], [55]. Moreover, some methods in [47] and [48] may not be effective for broadcasted RF signals due to their vulnerability to eavesdropping. Accordingly, we present a novel adversarial training approach to improve the robustness of protocol and modulation classifiers.

Several new defense mechanisms have recently been proposed in the literature (e.g., [26], [45], [56]), but were often countered by more potent attacks that are capable of bypassing these defenses. In principle, certified defenses (CD) ensures that a given classifier is robust to adversarial perturbations as long as these perturbations are constrained by a given bound. The authors in [57]–[59] proposed CD mechanisms that offer robustness guarantees against norm-bounded attacks. Recent research employs techniques like convex outer approximation [57], semi-definite relaxation [58], and differential privacy [59] to efficiently determine upper bounds on the worst-case loss. Random smoothing, a prevalent CD method [60], [61], introduces noise to input data and employs statistical approaches to measure the model's resilience to perturbations and provide probabilistic guarantees on its resistance to bounded perturbations. The widespread adoption of CD stems from the simplicity and effectiveness of this approach across diverse models and input variations. Lipschitz-based methods [62], [63] are variants of CD that also gained attention. These methods center on regulating the network's Lipschitz constant — a metric of a function's sensitivity to input changes. By ensuring minimal output variations in response to slight input perturbations, these methods train networks to inherently maintain stability and robustness. Despite the versatility of CD techniques against various attacks, their practical use in the wireless domain is limited because of the difficulty of establishing a meaningful bound on the attacker's perturbations, which undermines the efficacy of these techniques.

B. Adversarial Training

Adversarial training [23], in which a network is trained on adversarial examples, is one of the few defenses against adversarial attacks that withstand strong attacks. As a result, instead of updating the loss function based on a benign input x, the new loss function at the trained defender classifier is calculated based on both benign and adversarial inputs, as follows:

$$\tilde{L}(x, y; \theta) = \gamma L(x, y; \theta) + (1 - \gamma) L(x_{adv}, y; \theta). \quad (3)$$

The key idea behind this strategy is to increase the model's robustness by ensuring that the model predicts the same class for legitimate and perturbed examples. Considering the same attack generation method previously described: the defender first trains a DNN, denoted as $\text{DNN}_{\text{naive}}$, using benign data then the attacker steals DNN's structure, including all the weights and biases. In our defense mechanism, the defender uses $\text{DNN}_{\text{naive}}$ to develop its AML perturbations and combines them with benign data to retrain its DNN. The retraining dataset consists of the original and the self-perturbed data, resulting in a data augmentation compared to the $\text{DNN}_{\text{naive}}$ training. The retrained DNN is denoted by $\text{DNN}_{\text{defense}}$. To balance the impact between benign and adversarial data (i.e., the losses for both types of data), we set the sample number of both parts the same. As a result, portion parameter γ is 0.5, and the retrained DNN can have relatively good accuracy on both the benign and the perturbed data.

One important aspect of adversarial training is setting the parameters of the AML generator. For FGSM, this would be the value of ϵ. First, we consider a scenario where the defender uses a fixed value of ϵ, irrespective of the ϵ

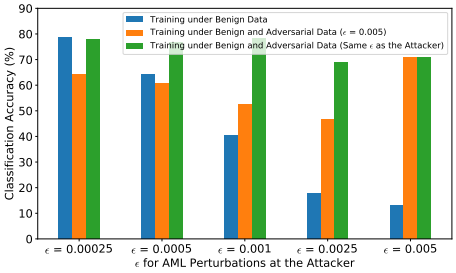


FIGURE 14: Classification accuracy when the defender's DNN is trained using benign or benign + AML data vs. ϵ of the attacker's FGSM perturbations. RML 2016.a dataset with SNR $= 16$ dB.

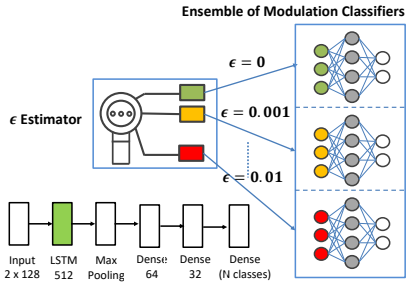


FIGURE 15: Two-step structure for robust classification of the received adversarial signals.

used during the attack (testing) phase. Considering the RML dataset and the VT-CNN2 network as a basis, we study the classification accuracy of the defender's DNN ($\text{DNN}_{\text{defense}}$) in three scenarios: (1) $\text{DNN}_{\text{defense}}$ is trained using benign data (i.e., $\text{DNN}_{\text{defense}}$ and $\text{DNN}_{\text{naive}}$ are identical), (2) $\text{DNN}_{\text{defense}}$ is retrained using a combination of benign and AML data, where the FGSM perturbations used for retraining are produced using $\epsilon = 0.005$, and (3) $\text{DNN}_{\text{defense}}$ is retrained using a combination of benign and AML data, where the FGSM perturbations used for retraining are produced using the same ϵ using by the adversary during the test phase. Note that in the second scenario, the choice of $\epsilon = 0.005$ is triggered by our interest in considering a reasonably small ϵ that leads to high SPR values, i.e., stealth attacks. The third scenario reflects the best-case performance of the defender, as it requires the defender to learn the specific ϵ used by the attacker, which is hard to obtain in a real attack.

Figure 14 shows the defender's classification performance for the three scenarios for different values ϵ used in the attacker's AML perturbations, i.e., the ϵ of the test dataset. In scenario one (blue bars), the higher ϵ of the attacker's AML data, the stronger the attack, and, hence, the worse the performance of the defender's classifier. Scenario two is presented in the orange bars. Interestingly, the inclusion of AML perturbations as part of the defender's training dataset improves the defender's classification performance only when the value of ϵ used by the attacker is close enough to the ϵ=0.005 used in the defender's AML training dataset (the accuracy increases from blue to orange bars). To improve the performance under benign-only training data, the defender need not exactly pinpoint the attacker's ϵ, i.e., a coarse estimate of ϵ is sufficient. For example, the performance under scenario two is better than that of scenario one when $\epsilon = 0.0025$ and $\epsilon = 0.001$. This is because the ϵ in the FGSM attack only impacts the energy of the perturbation. In other words, the perturbation vectors generated under $\epsilon = 0.0025$ and $\epsilon = 0.005$ points in the same direction but at different scales. The adversarial samples with high ϵ help the DNN know the direction of the perturbations. They can help improve the accuracy for lower ϵ by providing the same perturbation direction. When ϵ of the defender's training set is significantly different from ϵ of the attacker's testing set, the performance in scenario two can be worse than scenario one, i.e., the AML training data poisons the original (benign) dataset. In scenario three, shown in green bars, we assume that the defender and attacker use the same value of ϵ. This is a strong assumption since it is hard for the defender to know attacker's ϵ in advance. However, the results indicate that the classification performance can be improved if the defender can estimate ϵ.

We build a two-step structure for robust classification even under adversarial data. Figure 15 shows that the adversarial signal detector first approximates the ϵ value of the received signal and then assigns the signal to the corresponding modulation classifier. These classifiers are adversarially trained with a specific ϵ to perform well when receiving the same ϵ adversarial signals. We start with the design of the detector. Different neural networks, including CNN and RNN, are considered. We train the detector to predict the ϵ of the received signal from one of four possible values, where $\epsilon \in \{0, 0.001, 0.003, 0.005\}$. Figure 15 shows the LSTM network that achieves the best performance with an accuracy of 72%. The confusion matrix of the LSTM-based detector is shown in Figure 16. Although there are incorrect classifications, the misclassified are typically mostly drop in the adjacent values of ϵ. If we consider the accuracy as the sum of correct and adjacent labels, the average accuracy can achieve 96.75%. It indicates the detector can reasonably estimate the ϵ of the received signals.

Figure 17 compares the classification accuracy between VT-CNN2 and our approach, where the solid lines represent the testing accuracy for the VT-CNN2 and the dashed lines are for the proposed two-step defense mechanism. When the SNR is high, the accuracy of benign data is approximately 76%. This performance is lower than VT-CNN2, especially when the SNR is from -10 to 10 dB. However, the slight decrease in the performance from the adversarial information is negligible compared to the increase in the accuracy of the adversarial data. In other words, although adversarial training slightly sacrifices some accuracy on benign data to defend

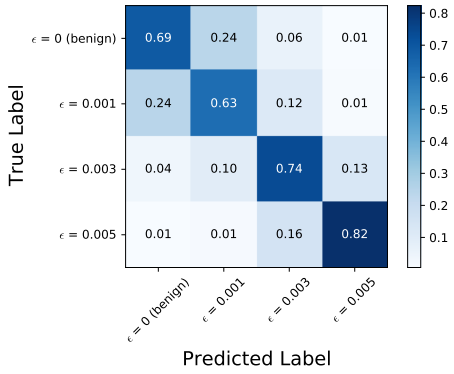


FIGURE 16: Confusion matrix of the LSTM-based detector.

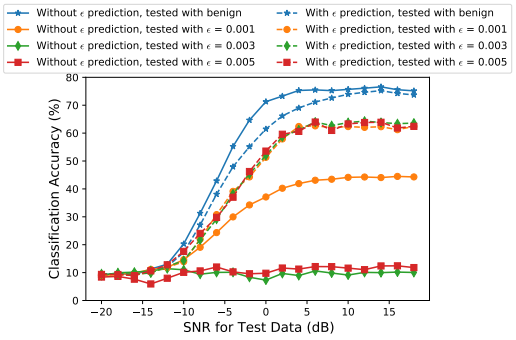


FIGURE 17: Comparison between VT-CNN2 and the proposed ϵ prediction mechanism on classification accuracy vs. testing SNRs with adversarial data using different ϵ. $\gamma = 0$ is used for adversarial training.

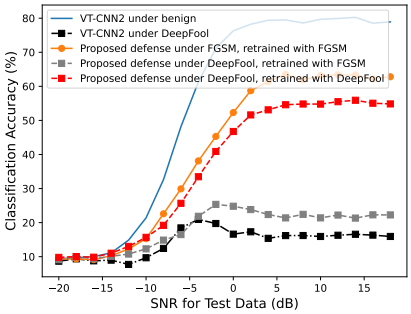


FIGURE 18: Evaluation of the proposed defense mechanism under FGSM and DeepFool attacks for different SNRs, when adversarial training is done using FGSM or DeepFool perturbations.

the attacks, the retrained model outperforms the original VT-CNN2 across all adversarial perturbations. For example, VT-CNN2 only achieves 10% accuracy when ϵ is high (e.g., $\epsilon = 0.005$). In contrast, the adversarially trained model achieves approximately 60% accuracy on adversarial data with different values of ϵ. Overall, our structure combines the benefit of all the classifiers in the second step and is robust to all four adversarial signals we considered.

We study a defense mechanism that is based on training the defender's classifier using either FGSM- or Deepfool-based perturbations, under DeepFool attacks. We summarize the results in Figure 18. The black and blue plots show the accuracy of the original VT-CNN2 modulation classifier, while the grey, red, and orange plots are for the retrained VT-CNN2 classifier and the proposed defense mechanism. It is anticipated that training with FGSM perturbations but testing it under DeepFool attacks yields a relatively lower accuracy improvement than testing it under FGSM attacks. This is attributed to the dissimilar nature of perturbations generated by these two attacks. For SNR greater than 0 dB, the proposed defense with FGSM-based adversarial training provides 8% improvement in accuracy relative to the original VT-CNN2 classifier when the attacker uses DeepFool perturbations. When we retrain the two-step defense mechanism with DeepFool perturbations and test under DeepFool attacks, we observe that the defender's accuracy significantly increases to 57% at high SNRs, as shown in the red plot. Similar to the orange line (trained and tested under FGSM), the defense mechanism's accuracy greatly improves when training and testing are done using the same attack type.

C. Autoencoder-based Defense

The authors in [50], [51] use an autoencoder before RF classifier to mitigate the impact of additive perturbations. We utilize the autoencoder-based defense mechanism as described in [50], [51]. Specifically, the denoising autoencoder (DA) architecture is chosen to be a fully connected DNN with 256-128-64-128-256 neurons at each layer. Note this is the same structure as Sahay *et al.* [50]. The DA was trained to minimize the mean squared error over 100 epochs. At evaluation time, the adversarial and benign signals are passed through the DA and then passed through the modulation classifier. Ideally, the DA would remove the adversarial perturbations without causing degradation to the classifier's performance under benign input.

Figure 19(a) shows the amplitudes of the original and DA-reconstructed waveform for the RML 2016.10a dataset, respectively. Visually, the denoised signal in blue is similar to the original signal in grey. This observation demonstrates that the DA successfully reconstructs the input. Then, the FGSM signals are passed to the DA using different values of ϵ. As shown in Figure 19(b)-(d), the denoised signal (in blue) is similar to the grey line when ϵ is small (0.001). As ϵ increases, the reconstructed signal deviates further from the original signal. The perturbation can have larger amplitudes with larger ϵ values, which results in the benign and adversarial signals deviating further. Unfortunately, the DA fails to denoise data perfectly. The DA's reconstruction error shows that the approach is ineffective under large perturbations.

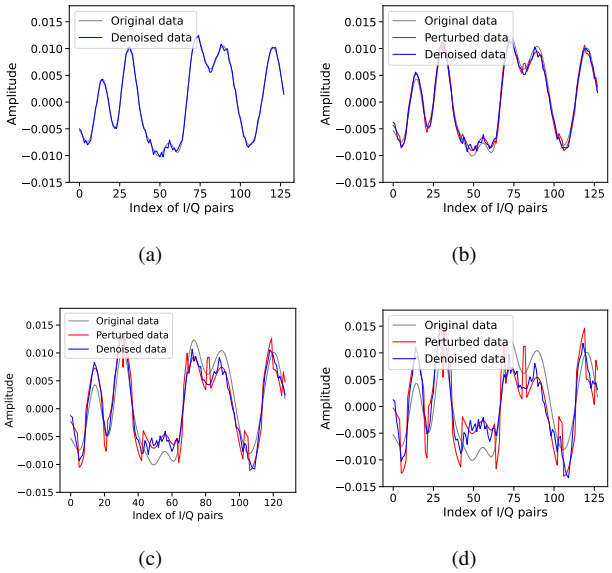


FIGURE 19: Examples of an FGSM perturbed and DA reconstructed signal in the RML 2016.10a dataset for: (a) $\epsilon = 0$ (benign), (b) $\epsilon = 0.001$, (c) $\epsilon = 0.003$, and (d) $\epsilon = 0.005$.

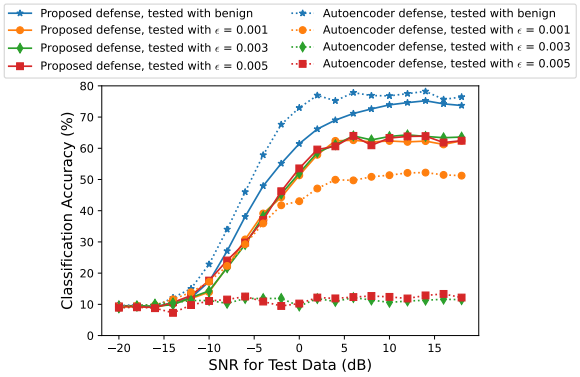


FIGURE 20: Comparison between the proposed and autoencoder-based defenses against FGSM attacks for various testing SNRs and ϵ.

Figure 20 compares the DA defense and our proposed defense under FGSM attacks. The DA defense improves the defender's accuracy when $\epsilon = 0.001$ to 50% at high SNRs; however, the defender's accuracy degrades to near 10% as ϵ increases. In contrast, our proposed approach outperforms the DA method, and can improve the accuracy to more than 65% under attacks (except $\epsilon = 0$, i.e., no attack). Although the results in [50], [51] show the DA's effectiveness, their perturbations use small values of ϵ. Our results show that the DA may not be a suitable defense mechanism for larger ϵ.

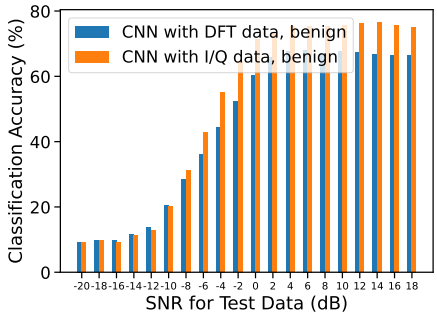


FIGURE 21: Comparison between the CNNs trained under benign raw I/Q data and DFT transformed data on classification accuracy vs. testing SNRs.

D. Ensemble-based Defense

We extend our evaluation to include an ensemble-based defense approach. Inspired by [52], we train three DNN models: a fully connected neural network (FCNN), a CNN, and an RNN. Additionally, we consider both the original time-domain I/Q data as well as a frequency-domain version obtained using the discrete Fourier transform (DFT). Thus, we end up with six trained classifiers: three DNNs trained in the time domain and three DNNs trained in the frequency domain. The outputs of six classifiers are averaged to form an ensemble prediction, following the strategy outlined in [52].

While the authors of [52] demonstrated impressive accuracy for their classifier leveraging both time and frequency representations, our observations show that the DNN classifiers trained on frequency-domain transformed data do not attain the same accuracy as the time-domain models. Figure 21 shows the CNN's accuracy that is trained with I/Q and the DFT data. The two classifiers have similar accuracy when the SNR is less than -8 dB; however, the CNN trained with I/Q data has better accuracy than their DFT counterpart as the SNR increases.

A potential explanation for the disparities in our findings and those of [52] could stem from differences in the datasets. Specifically, the datasets employed in [52] include only four modulation types. In contrast, our study of modulation classification is based on the full RML 2016.10a dataset, which consists of 11 modulation types, including two amplitude modulation schemes (AM-DSB and AM-SSB). These two modulation schemes were not a part of the dataset used in [52]. Applying DFT to amplitude modulation data can potentially lead to the loss of crucial temporal features, resulting in lower accuracy for a DFT-trained classifier compared to a classifier trained on raw (time-domain) I/Q data. Furthermore, irrespective of whether the data are processed in the time or frequency domain, including additional classes in the dataset adds complexity to the decision boundary, which can lead to class overlap and reduction in accuracy.

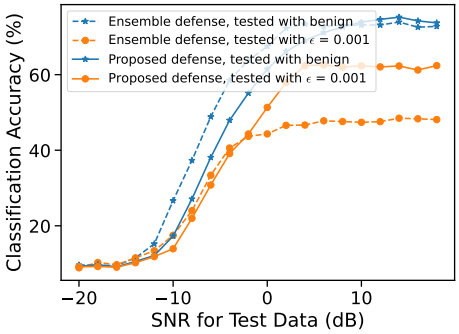


FIGURE 22: Comparison between the proposed defense mechanism and ensemble-based defense on classification accuracy vs. testing SNRs with adversarial data using ϵ = 0.001.

Figure 22 compares our defense to the ensemble-based approach. The ensemble strategy surpasses our defense (depicted by solid lines) when the SNR ranges from -10 to 5 dB. The trend is reversed for SNR > 5 dB. The ensemble defense's accuracy is nearly 50% in high SNR scenarios under FGSM attacks with $\epsilon = 0.001$. Our defense exhibits an accuracy exceeding 60% in such scenarios, establishing a more effective safeguard than the ensemble approach for these experiments.

IX. Conclusions

Machine learning, particularly deep learning, plays an increasingly important role in wireless communications and can achieve state-of-the-art performances without handcrafted features. While these DNNs achieve satisfactory performance, they are also vulnerable to adversarial perturbations, limiting the classifiers' robustness. Most of these perturbations are undetectable at the input to the deep learning classifier; however, the classifier's output has significant changes. Thus, the strength of the attack is strong if the performance goes down and the SPR keeps high, which also makes the perturbation hard to detect.

This work studied the vulnerability of DNN-based classifiers to AML-based jamming attacks for signal classification datasets. We considered two different signal classification types, namely, protocol and modulation classification. By adding different types of AML-based perturbations while maintaining a relatively high SPR level, all DNNs significantly reduce the classification accuracy. We considered various adversarial approaches, including the FGSM, PGD, and DeepFool attacks. The decrease in performance when the adversarial signals have a high SPR, further shows that highly successful attacks can be challenging to detect [64].

The results show that these attacks can negatively impact the defender's accuracy. We observed similar trends on the DNN-based classifier for the protocol and modulation datasets. The effectiveness of the AML perturbations depends on the amount of information the adversary has regarding the structure and training dataset of the defender's classifier. Accordingly, we studied different attack scenarios with varying levels of knowledge. In one extreme, an attacker with full knowledge of the defender (white-box attack) significantly degrades the defender's accuracy. Compared to traditional jamming, where the attacker transmits only AWGN noise, the proposed AML-based attack requires much less transmit power to mislead the classifiers.

We also observed that DNNs are vulnerable to these attacks even if the attacker has imperfect synchronization, incomplete sequence, or under the noisy channel, of both the protocol and modulation classification. We generate attacks under these more practical cases and evaluate the impact of attacks of different synchronization, sequence length, and channel noise levels. We show that these imperfect attacks can still effectively drop the defender's accuracy in a certain imperfection range.

Finally, we propose the counter measurements for AML attacks and address one limitation of adversarial training. The proposed mechanism splits the defense into two steps: ϵ estimation and classifier retraining. In the first step, the ϵ estimator accurately estimates ϵ, and the adversarial training in the second step can counter a more specific attack. The proposed structure combines the benefit of all the classifiers in the second step. As a result, the two-step defense shows better robustness and effectively improves the defender's accuracy under different budget settings of attacks compared to the single-classifier retraining.

X. Acknowledgements

This research was supported by the Army Research Office under Contract No. W911NF-21-C-0016, the National Science Foundation (grants Nos. 1943552, 2229386, and 1822071), and by the Broadband Wireless Access & Applications Center (BWAC). Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the sponsors. G. Ditzler was affiliated with the University of Arizona and Rowan University when this work was performed.

REFERENCES

[1] W. Zhang, M. Krunz, and G. Ditzler, "Intelligent jamming of deep neural network based signal classification for shared spectrum," in *Proc. of the IEEE Military Communications Conference (MILCOM)*, November 2021, pp. 987–992.

[2] W. Zhang, M. Feng, M. Krunz, and A. H. Y. Abyaneh, "Signal detection and classification in shared spectrum: A deep learning approach," in *Proc. of the IEEE Conference on Computer Communications (INFOCOM)*, May 2021, pp. 1–10.

[3] F. A. Bhatti, M. J. Khan, A. Selim, and F. Paisana, "Shared spectrum monitoring using deep learning," *IEEE Transactions on Cognitive Communications and Networking*, vol. 7, no. 4, pp. 1171–1185, 2021.

[4] S. Sarkar, M. Buddhikot, A. Baset, and S. K. Kasera, "DeepRadar: A deep-learning-based environmental sensing capability sensor design for CBRS," in *Proc. of the International Conference on Mobile Computing and Networking (MobiCom)*, March 2021, pp. 56–68.

[5] W. M. Lees, A. Wunderlich, P. J. Jeavons, P. D. Hale, and M. R. Souryal, "Deep learning classification of 3.5-GHz band spectrograms with applications to spectrum sensing," *IEEE Transactions on Cognitive Communications and Networking*, vol. 5, no. 2, pp. 224–236, 2019.

[6] L. J. Wong, W. H. Clark, B. Flowers, R. M. Buehrer, W. C. Headley, and A. J. Michaels, "An RFML ecosystem: Considerations for the application of deep learning to spectrum situational awareness," *IEEE Open Journal of the Communications Society*, vol. 2, pp. 2243–2264, 2021.

[7] Y. E. Sagduyu, T. Erpek, and Y. Shi, "Adversarial machine learning for 5G communications security," *arXiv preprint arXiv:2101.02656*, 2021.

[8] D. Adesina, C.-C. Hsieh, Y. E. Sagduyu, and L. Qian, "Adversarial machine learning in wireless communications using RF data: A review," *IEEE Communications Surveys & Tutorials*, pp. 1–25, 2022.

[9] 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 of Things Magazine*, vol. 3, no. 1, pp. 50–57, 2020.

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

[11] N. Soltani, G. Reus-Muns, B. Salehi, J. Dy, S. Ioannidis, and K. Chowdhury, "RF fingerprinting unmanned aerial vehicles with non-standard transmitter waveforms," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 12, pp. 15 518–15 531, 2020.

[12] Y. Shi, K. Davaslioglu, Y. Sagduyu, W. Headley, M. Fowler, and G. Green, "Deep learning for RF signal classification in unknown and dynamic spectrum environments," in *Proc. of the IEEE International Symposium on Dynamic Spectrum Access Networks (DySPAN)*, November 2019, pp. 1–10.

[13] Z. Qin, H. Ye, G. Y. Li, and B.-H. F. Juang, "Deep learning in physical layer communications," *IEEE Wireless Communications*, vol. 26, no. 2, pp. 93–99, 2019.

[14] S. Zheng, P. Qi, S. Chen, and X. Yang, "Fusion methods for CNN-based automatic modulation classification," *IEEE Access*, vol. 7, pp. 66 496–66 504, 2019.

[15] K. Davaslioglu, S. Boztaş, M. C. Ertem, Y. E. Sagduyu, and E. Ayanoglu, "Self-supervised RF signal representation learning for nextG signal classification with deep learning," *IEEE Wireless Communications Letters*, vol. 12, no. 1, pp. 65–69, 2023.

[16] S. Rajendran, W. Meert, D. Giustiniano, V. Lenders, and S. Pollin, "Deep learning models for wireless signal classification with distributed low-cost spectrum sensors," *IEEE Transactions on Cognitive Communications and Networking*, vol. 4, no. 3, pp. 433–445, 2018.

[17] W. Zhang and M. Krunz, "Machine learning based protocol classification in unlicensed 5 GHz bands," in *Proc. of the IEEE International Conference on Communications Workshops (ICC Workshops)*, May 2022, pp. 1–6.

[18] J. Wang, Q. Gao, X. Ma, Y. Zhao, and Y. Fang, "Learning to sense: Deep learning for wireless sensing with less training efforts," *IEEE Wireless Communications*, vol. 27, no. 3, pp. 156–162, 2020.

[19] C. Zhang, P. Patras, and H. Haddadi, "Deep learning in mobile and wireless networking: A survey," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 3, pp. 2224–2287, 2019.

[20] Y. Shi and Y. Sagduyu, "Membership inference attack and defense for wireless signal classifiers with deep learning," *IEEE Transactions on Mobile Computing*, vol. 22, no. 7, pp. 4032–4043, July 2023.

[21] R. Ning, C. Xin, and H. Wu, "TrojanFlow: A neural backdoor attack to deep learning-based network traffic classifiers," in *Proc. of the IEEE Conference on Computer Communications (INFOCOM)*, May 2022, pp. 1429–1438.

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

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

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

[25] S.-M. Moosavi-Dezfooli, A. Fawzi, and P. Frossard, "Deepfool: A simple and accurate method to fool deep neural networks," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, June 2016, pp. 2574–2582.

[26] N. Carlini and D. Wagner, "Towards evaluating the robustness of neural networks," in *Proc. of the IEEE Symposium on Security and Privacy (S&P)*, May 2017, pp. 39–57.

[27] 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.

[28] B. Flowers, R. M. Buehrer, and W. C. Headley, "Evaluating adversarial evasion attacks in the context of wireless communications," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 1102–1113, 2020.

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

[30] M. Sadeghi and E. Larsson, "Physical adversarial attacks against end-to-end autoencoder communication systems," *IEEE Communications Letters*, vol. 23, no. 5, pp. 847–850, 2019.

[31] Y. Lin, H. Zhao, Y. Tu, S. Mao, and Z. Dou, "Threats of adversarial attacks in DNN-based modulation recognition," in *Proc. of the IEEE Conference on Computer Communications (INFOCOM)*, July 2020, pp. 2469–2478.

[32] B. Kim, Y. Sagduyu, K. Davaslioglu, T. Erpek, and S. Ulukus, "Over-the-air adversarial attacks on deep learning based modulation classifier over wireless channels," in *Proc. of the Annual Conference Information Sciences and Systems*, March 2020, pp. 1–6.

[33] B. Kim, Y. E. Sagduyu, K. Davaslioglu, T. Erpek, and S. Ulukus, "Channel-aware adversarial attacks against deep learning-based wireless signal classifiers," *IEEE Transactions on Wireless Communications*, pp. 3868 – 3880, 2021.

[34] B. Kim, Y. E. Sagduyu, T. Erpek, K. Davaslioglu, and S. Ulukus, "Channel effects on surrogate models of adversarial attacks against wireless signal classifiers," in *Proc. of the IEEE International Conference on Communications*, 2021, pp. 1–6.

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

[36] Y. LeCun *et al.*, "LeNet-5, convolutional neural networks," *URL: http://yann. lecun. com/exdb/lenet*, vol. 20, no. 5, p. 14, 2015.

[37] T. J. O'Shea and J. Corgan, "Convolutional radio modulation recognition networks," *arXiv preprint arXiv:1602.04105*, 2016.

[38] D. Schwartz and G. Ditzler, "Bolstering adversarial robustness with latent disparity regularization," in *Proc. of the IEEE/INNS International Joint Conference on Neural Networks*, July 2021, pp. 1–8.

[39] N. V. Abhishek and M. Gurusamy, "JaDe: Low power jamming detection using machine learning in vehicular networks," *IEEE Wireless Communications Letters*, vol. 10, no. 10, pp. 2210–2214, 2021.

[40] S. Kokalj-Filipovic, R. Miller, and G. Vanhoy, "Adversarial examples in RF deep learning: Detection and physical robustness," in *Proc. of the IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, 2019, pp. 1–5.

[41] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," in *Proc. of the International Conference on Learning Representations (ICLR)*, July 2015, pp. 1–11.

[42] J. Bruna, C. Szegedy, I. Sutskever, I. Goodfellow, W. Zaremba, R. Fergus, and D. Erhan, "Intriguing properties of neural networks," in *Proc. of the International Conference on Learning Representations (ICLR)*, Dec. 2013.

[43] C. Xie, Z. Zhang, Y. Zhou, S. Bai, J. Wang, Z. Ren, and A. L. Yuille, "Improving transferability of adversarial examples with input diversity," in *Proc. of the IEEE/CVF conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 2730–2739.

[44] W. Wu, Y. Su, M. R. Lyu, and I. King, "Improving the transferability of adversarial samples with adversarial transformations," in *Proc. of the IEEE/CVF conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 9024–9033.

[45] F. Croce and M. Hein, "Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks," in *Proc. of the International Conference on Machine Learning (ICML)*, 2020, pp. 2206–2216.

[46] P.-Y. Chiang, J. Geiping, M. Goldblum, T. Goldstein, R. Ni, S. Reich, and A. Shafahi, "Witchcraft: Efficient PGD attacks with random step

size," in *Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 3747–3751.
[47] F. O. Olowononi, D. B. Rawat, and C. Liu, "Resilient machine learning for networked cyber physical systems: A survey for machine learning security to securing machine learning for CPS," *IEEE Communications Surveys & Tutorials*, vol. 23, no. 1, pp. 524–552, 2021.
[48] K. He, D. D. Kim, and M. R. Asghar, "Adversarial machine learning for network intrusion detection systems: A comprehensive survey," *IEEE Communications Surveys & Tutorials*, vol. 25, no. 1, pp. 538–566, 2023.
[49] D. Adesina, C.-C. Hsieh, Y. E. Sagduyu, and L. Qian, "Adversarial machine learning in wireless communications using RF data: A review," *IEEE Communications Surveys & Tutorials*, vol. 25, no. 1, pp. 77–100, 2023.
[50] R. Sahay, R. Mahfuz, and A. El Gamal, "Combatting adversarial attacks through denoising and dimensionality reduction: A cascaded autoencoder approach," in *Proc. of the IEEE Annual Conference on Information Sciences and Systems (CISS)*, 2019, pp. 1–6.
[51] 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. of the International Conference on Military Communications and Information Systems (ICMCIS)*, 2019, pp. 1–6.
[52] R. Sahay, C. G. Brinton, and D. J. Love, "A deep ensemble-based wireless receiver architecture for mitigating adversarial attacks in automatic modulation classification," *IEEE Transactions on Cognitive Communications and Networking*, vol. 8, no. 1, pp. 71–85, 2021.
[53] R. Sahay, D. J. Love, and C. G. Brinton, "Robust automatic modulation classification in the presence of adversarial attacks," in *Proc. of the Annual Conference on Information Sciences and Systems (CISS)*, 2021, pp. 1–6.
[54] A. Shafahi, M. Najibi, M. A. Ghiasi, Z. Xu, J. Dickerson, C. Studer, L. S. Davis, G. Taylor, and T. Goldstein, "Adversarial training for free!" *Advances in Neural Information Processing Systems*, vol. 32, pp. 1–12, 2019.
[55] E. Wong, L. Rice, and J. Z. Kolter, "Fast is better than free: Revisiting adversarial training," in *Proc. of the International Conference on Learning Representations (ICLR)*, 2019, pp. 1–17.
[56] T. Bai, J. Luo, J. Zhao, B. Wen, and Q. Wang, "Recent advances in adversarial training for adversarial robustness," in *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, Aug. 2021, pp. 4312–4321. [Online]. Available: https://doi.org/10.24963/ijcai.2021/591
[57] E. Wong and Z. Kolter, "Provable defenses against adversarial examples via the convex outer adversarial polytope," in *Proc. of the International Conference on Machine Learning (ICML)*, 2018, pp. 5286–5295.
[58] A. Raghunathan, J. Steinhardt, and P. Liang, "Certified defenses against adversarial examples," in *Proc. of the International Conference on Learning Representations (ICLR)*, 2018.
[59] M. Lecuyer, V. Atlidakis, R. Geambasu, D. Hsu, and S. Jana, "Certified robustness to adversarial examples with differential privacy," in *Proc. of the IEEE Symposium on Security and Privacy (S&P)*, 2019, pp. 656–672.
[60] H. Salman, J. Li, I. Razenshteyn, P. Zhang, H. Zhang, S. Bubeck, and G. Yang, "Provably robust deep learning via adversarially trained smoothed classifiers," *Advances in Neural Information Processing Systems*, vol. 32, 2019.
[61] J. Cohen, E. Rosenfeld, and Z. Kolter, "Certified adversarial robustness via randomized smoothing," in *Proc. of the International Conference on Machine Learning (ICML)*, 2019, pp. 1310–1320.
[62] S. Lee, J. Lee, and S. Park, "Lipschitz-certifiable training with a tight outer bound," *Advances in Neural Information Processing Systems*, vol. 33, pp. 16 891–16 902, 2020.
[63] K. Leino, Z. Wang, and M. Fredrikson, "Globally-robust neural networks," in *Proc. of the International Conference on Machine Learning (ICML)*, 2021, pp. 6212–6222.
[64] C. Frederickson, M. Moore, G. Dawson, and R. Polikar, "Attack strength vs. detectability dilemma in adversarial machine learning," in *Proc. of the IEEE/INNS International Joint Conference on Neural Networks*, July 2018, pp. 1–8.

Wenhan Zhang[S'19] received the B.S. degree in electrical engineering and automation from Hefei University of Technology, China, in 2016, and the M.S. degree in electrical engineering from Syracuse University in 2018. He is working toward his Ph.D. degree with the Department of Electrical and Computer Engineering at the University of Arizona. His research interests include mobile edge computing, wireless communications, and applications of machine learning in wireless networks.

Marwan Krunz[S'93-M'95-SM'04-F'10] is a Regents Professor of electrical and computer engineering at the University of Arizona. He also holds a joint appointment as a professor of computer science. From 2015 to 2023, he was the Kenneth VonBehren Endowed Professor in ECE. Currently, he directs the Broadband Wireless Access and Applications Center (BWAC), a multi-university NSF/industry center that focuses on next-generation wireless technologies. He is also an Affiliated Faculty of the UA Cancer Center. Previously, he served as the Site Director for the Connection One Center. He served as the chief scientist for two startup companies that focus on 5G and beyond systems and machine learning for wireless communications. He has published more than 330 journal articles and peer-reviewed conference papers and is a named inventor on ten patents. His latest H-index is 62. His research interests include wireless communications and protocols, network security, and machine learning. He was an Arizona Engineering Faculty Fellow and an IEEE Communications Society Distinguished Lecturer. He received the NSF CAREER Award. He was the TPC Chair for several conferences and symposia, including INFOCOM'04, SECON'05, WoWMoM'06, and Hot Interconnects 9. He was a general chair for WiOpt'23, vice-chair for WiOpt'16, and the general co-chair for WiSec'12. He served as the Editor-in-Chief for the IEEE Transactions on Mobile Computing. He served as an editor for numerous IEEE journals.

Gregory Ditzler[S'04–M'15-SM'21] received a B.Sc., M.Sc. and Ph.D. from the Pennsylvania College of Technology (2008), Rowan University (2011), and Drexel University (2015), respectively. He is a Technical Director at EpiSys Science (EpiSci). He was previously an Associate Professor with the Department of Electrical and Computer Engineering at Rowan University (2022/23) the University of Arizona (2015–22). His research interests include machine learning, adversarial learning, neural networks, concept drift, and applications life sciences, and cybersecurity. In 2016 and 2018, he was a Summer Faculty Fellow with the Air Force Research Laboratory. He received an Outstanding Article Award from IEEE Computational Intelligence Society Magazine, in 2018, the Best Paper at the IEEE International Conference on Cloud and Autonomic Computing, in 2017, and the Best Student Paper at the IEEE/INNS International Joint Conference on Neural Networks, in 2014. He was a recipient of the NSF CAREER Award. He was an Associate Editor for IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS and Cluster Computing.