Intelligent Jamming of Deep Neural Network Based Signal Classification for Shared Spectrum

Wenhan Zhang, Marwan Krunz, and Gregory Ditzler

Dept. Electrical & Computer Engineering, University of Arizona, Tucson, AZ
{wenhanzhang, krunz, ditzler}@email.arizona.edu

***Abstract*—Deep neural networks (DNNs) have recently been applied in the classification of radio frequency (RF) signals. One use case of interest relates to the discernment between different wireless technologies that share the spectrum. Although highly accurate DNN classifiers have been proposed, preliminary research points to the vulnerability of these classifiers to adversarial machine learning (AML) attacks. In one such attack, a surrogate DNN model is trained by the attacker to produce intelligently crafted low-power "perturbations" that degrade the classification accuracy of the legitimate classifier. In this paper, we design four DNN-based classifiers for the identification of Wi-Fi, 5G NR-Unlicensed (NR-U), and LTE LAA transmissions over the 5 GHz UNII bands. Our DNN models include both convolutional neural networks (CNNs) as well as several recurrent neural networks (RNNs) models, particularly LSTM and Bidirectional LSTM (BiLSTM) networks. We demonstrate the high classification accuracy of these models under "benign" (non-adversarial) noise. We then study the efficacy of these classifiers under AML-based perturbations. Specifically, we use the fast gradient sign method (FGSM) to generate adversarial perturbations. Different attack scenarios are studied, depending on how much information the attacker has about the defender's classifier. In one extreme scenario, called "white-box" attack, the attacker has full knowledge of the defender's DNN, including its hyperparameters, its training dataset, and even the seeds used to train the network. This attack is shown to significantly degrade the classification accuracy even when the FGSM-based perturbations are low power, i.e., the received SNR is relatively high. We then consider more realistic attack scenarios, where the attacker has partial or no knowledge of the defender's classifier. Even under limited knowledge, adversarial perturbations can still lead to significant reduction in the classification accuracy, relative to classification under AWGN with the same SNR level.**

***Index Terms*—Deep learning, signal classification, adversarial machine learning, shared spectrum, wireless security**

I. Introduction

Waveform discernment plays an important role in next-generation wireless systems. It is used to identify the underlying technologies in a spectrum-sharing scenario, e.g., coexisting Wi-Fi and cellular transmissions over the unlicensed 5/6 GHz bands [1] or LTE/radar transmissions over the CBRS band [2]. It can also be used to identify (without signal decoding) the nature of observed interference. Certain exogenous interference is caused by spurious emissions of benign effect; others may produce strong intentional (adversarial) or unintentional interference. In particular, malicious parties can generate many types of emissions, some aimed at disrupting receptions (jamming attacks) while others aimed at impersonating legitimate users [3]. In mission-critical applications, such as military systems and autonomous vehicles, the ability to discern between legitimate and rogue waveforms is quite critical to the overall safety and security of the network.

Recently, deep neural networks (DNNs) have been applied to RF signal classification problems, including modulation and coding scheme (MCS) identification [4], [5], unknown signal detection [6], and protocol classification [1]. In contrast to traditional feature-based spectrum sensing, DNN-based classification is data driven, and does not require explicit specification of any technology-dependent features. Different types of DNNs have been considered, including convolutional neural networks (CNNs) [7]–[9] and recurrent neural networks (RNNs) [2], [10]. A CNN employs convolution layers to extract features in multidimensional data. However, it is not effective at capturing temporal dependencies. In contrast, an RNN uses a recurrent structure to capture the memory (time dependency) in the data, which explains its widespread use in forecasting problems, such as language modeling, speech recognition, and trajectory prediction of moving objects. In [7], the authors used a CNN-based model to classify Wi-Fi devices using a 2-by-N matrix as its input, where N is the number of successively received and down-converted samples. Each sample is associated with an in-phase (I) and a quadrature (Q) components. The authors in [10] applied a multi-layer long-short-term-memory (LSTM) network, a class of RNNs, for automatic modulation classification. Their proposed design outperforms the CNN model at high SNRs. In [2], the authors used CNNs and LSTM networks to detect radar signals in the 3.5 GHz band. Their results show that both CNN and LSTM models have the potential to achieve high signal classification accuracy. A combined CNN/LSTM architecture was proposed in [1] to identify Wi-Fi, 5G NR, and LTE signals over the unlicensed 5 GHz bands.

Despite their advantages, DNN-based classifiers are prone to adversarial machine learning (AML) attacks [11]. Such attacks have been studied in object classification/recognition problems (e.g., [12], [13]), and more recently in RF signal classification (e.g., [14]–[16]). The general idea is to train a surrogate DNN classifier, henceforth called the *attacker's classifier*, to produce properly crafted perturbations. When combined with a test input, these perturbations mislead the legitimate classifier, henceforth called the *defender's classifier*, into incorrect labels; see Figure 1. Note that in an object

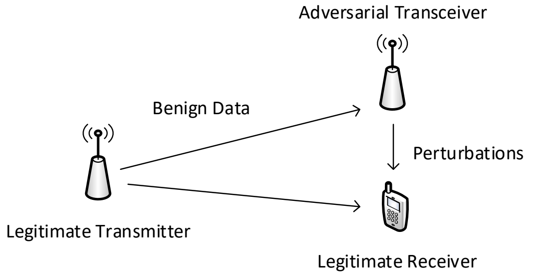


Fig. 1. AML-based attack on the signal classifier of a legitimate receiver.

recognition application, the same training dataset is used at both the attacker's and defender's classifiers. In contrast, for the RF signal classification problem, different datasets may be used to account for differences in the channel conditions between the attacker and defender ("legitimate receiver"). The authors in [14] designed various AML attacks that take into account channel conditions. In [16], the authors showed that the accuracy of a CNN-based classifier used for modulation classification drops tremendously when receiving a slightly perturbed input. In our paper, we study four CNN- and RNN-based classifiers for Wi-Fi, LTE, and 5G NR signals that coexist over the unlicensed 5 GHz band. We first verify the high accuracy of these classifiers under noisy but benign perturbations (i.e., AWGN). We then study the impact of AML-based perturbations on their classification accuracy.

II. System Model

We consider a wireless system that consists of a legitimate transmitter-receiver pair and an adversarial device. The transmitter randomly generates waveforms according to one of several possible protocols in an interleaved manner, (i.e., one transmission at a time). The defender's classifier resides at the legitimate receiver and is trained to identify waveforms based on the *received* baseband I/Q samples. The attacker eavesdrops on ongoing transmissions (called *benign data*) and uses them to train its own classifier. Subsequently, the attacker transmits its perturbations that interfer 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*.

Consider the defender's classifier. Its output can be represented as $z = f(x;\theta)$, where x is the input and θ is the set of learnable DNN parameters, i.e., weight matrix and bias vectors. The input x is a 2-by-N matrix, where N is the window size (number of consecutive samples) and the first (second) row represents the sequence of I (Q) values. By applying an activation function σ, we have the numerical output vector z according to the class number K: $\sigma(z) \subset \mathbb{R}^K$. After that, the classifier assigns the label to the received input $l(x;\theta) = \arg\max_k(\sigma(z)_k)$, where $k \in K$. In this formulation, $\sigma(z)_k$ is the numerical output of classifier f corresponding to the kth protocol type.

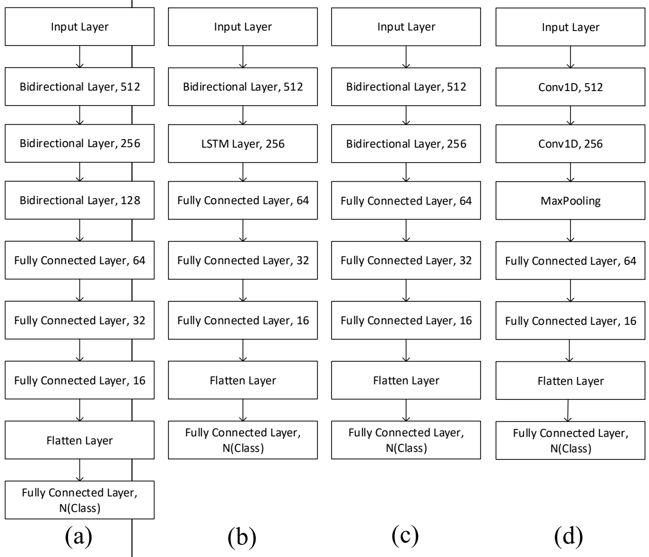

Input Layer
Bidirectional Layer, 512
Bidirectional Layer, 256
Bidirectional Layer, 128
Fully Connected Layer, 64
Fully Connected Layer, 32
Fully Connected Layer, 16
Flatten Layer
Fully Connected Layer, N(Class)
(a)
Input Layer
Bidirectional Layer, 512
LSTM Layer, 256
Fully Connected Layer, 64
Fully Connected Layer, 32
Fully Connected Layer, 16
Flatten Layer
Fully Connected Layer, N(Class)
(b)
Input Layer
Bidirectional Layer, 512
Bidirectional Layer, 256
Fully Connected Layer, 64
Fully Connected Layer, 32
Fully Connected Layer, 16
Flatten Layer
Fully Connected Layer, N(Class)
(c)
Input Layer
Conv1D, 512
Conv1D, 256
MaxPooling
Fully Connected Layer, 64
Fully Connected Layer, 16
Flatten Layer
Fully Connected Layer, N(Class)
(d)

Fig. 2. DNN structures used for waveform classification. Structure (a)-(c) are RNN-based classifiers, whereas structures (d) is a CNN-based classifier.

We define $\mathbf{H}_{td}$ as the channel matrix from the legitimate transmitter to the defender, $\mathbf{H}_{ta}$ as the channel matrix from the legitimate transmitter to the attacker, and $\mathbf{H}_{ad}$ as the channel matrix from the attacker to the defender. We assume AWGN (n) at any receiving device. In the absence of AML perturbations, the defender receives $x_d^t = \mathbf{H}_{td}x_o + n$, where x_o is the transmitted waveform. The attacker receives $x_a^t = \mathbf{H}_{ta}x_o + n$. To launch an AML attack, the adversary uses its received signal x_a^t to generate and transmit the perturbations η. Under this attack, the defender receives $x_d^{t,a} = \mathbf{H}_{td}x_o + \mathbf{H}_{ad}\eta + n$.

To generate η, we consider the *fast gradient sign method* (FGSM) [11], which has been widely applied in AML-based attacks on image classifiers. Specifically, the attacker solves the following problem for η:

$$\begin{aligned} &\max_{\eta} \ \mathbb{I}\big\{l(x_d^t;\theta) \neq l(x_d^{t,a};\theta)\big\} \\ &s.t. \ \|\eta\|_\infty \leq \epsilon \end{aligned} \tag{1}$$

where ϵ is a preset parameter that is used to limit the power of the perturbations and ensure that the attack is hard to detect. $\mathbb{I}$ is an indicator function that reflects the number of misclassified labels in a given training set. We assume the attacker is close to the defender, hence, $\mathbf{H}_{ad}\eta \approx \eta$.

III. Neural Network Structures

We consider four DNN structures for the defender's and attacker's classifiers, as shown in Figure 2. To train and test these networks, we generate a dataset of 15,000 inputs (see Section VI-A), each of which containing 512 noisy I/Q samples. Approximately 60% of the dataset is used for training, 20% for validation and early stopping, and 20% for testing the network. To reduce overfitting, we monitor the categorical cross-entropy with patience of three in the early stopping for all the proposed models.

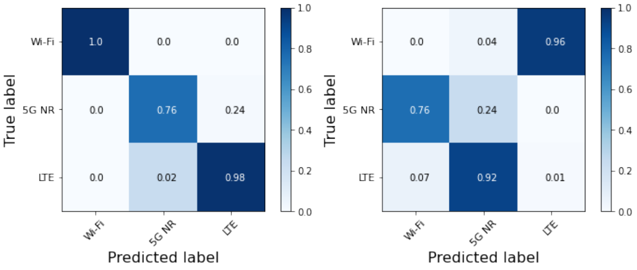
A. Recurrent Neural Networks

We consider a stacked RNN architecture where the output of one RNN layer is used as input to the next-outer RNN layer. The inner layer can be any RNN structure, such as a standard RNN (i.e., the SimpleRNN in *TensorFlow* [17]), a Long Short-Term Memory (LSTM), or a Gated Recurrent Unit (GRU). During training, the various classification outcomes of the inner layer are used as inputs to train the outer layer. Thus, the output at the final layer (i.e., classification layer) is expected to achieve higher classification accuracy than any inner-layer network. This stacked architecture captures temporal correlations at different time scales without using many input samples. To further improve the classification accuracy, we also consider a bidirectional RNN structure for the inner layers [18]. These bidirectional layers connect two hidden layers of opposite directions to receive information from the past (backward) and future (forward) states simultaneously. This layer of bidirectionality makes the network non-causal, where future information can influence the current decision; however, this non-causality is applied only during the training of the RNN network and is not required when the network is evaluated.

There are several considerations that guided the selection of a RNN layer within the architecture. LSTM networks have been widely used for many sequential prediction tasks, due to the efficiency of their gated structure and high accuracy [2], [10]. Therefore, we consider applying the bidirectional stacked LSTM network to classify the RF signals. Such a structure allows the lower layer to transform the raw input into a more suitable feature representation (e.g., removing unrelated samples and disturbances). The higher layers can make a more precise prediction by learning the dependencies in both directions from the refined sequence data. Figure 2(a)-(c) shows three network configurations that are used in Section VI.

B. Convolutional Neural Network

We also use a CNN that has been modified from *LeNet* [19] to benchmark against the RNNs. The original CNN was designed for image classification. Hence, it will not work properly for our task of sequence classification. Therefore, we use a *Conv1D* layer to transform the sequence data. The sequences may need to be padded with zeros depending on the length of the data that are sampled; however, our experiments showed that padding the input sequence at the convolutional layer does not improve the performance. Therefore, we removed the padding layer from *LeNet* and only reported the results for the better-performing CNN. The final CNN configuration is summarized in Figure 2(d). Note that the kernel size for *Conv1D* layer is 2, and its stride is set to 1. The activation functions are scaled exponential linear units for all the *Conv1D* and fully-connected layers. The output layers for all networks in Figure 2 are soft-max.



(a) Benign dataset (b) Adversarial dataset ($\epsilon = 0.3$)

Fig. 3. Confusion matrices for DNN structure (b).

IV. White-box Adversarial Attack

The neural networks in the previous section can be used to generate adversarial perturbations that fool either the same network or the other networks. In the latter situation, the attack samples are transferable if the attacker does not know the defender's network but still negatively impacts the defender's performance. If the attacker knows the defender's data and network, then the attack is known as white-box and is the worst-case scenario for the defender. To generate the adversarial perturbations, we first considered using the FGSM attack [11]. This technique uses the gradients of a neural network to generate a perturbation η and, subsequently, the adversarial data x_{adv}. The defender expects to predict the same class for x and its adversarial perturbation $x_{adv} = x + \eta$ if every element of η is less than the given precision. Hence, a classifier can assign the same class to x and x_{adv} if $||\eta||_\infty < \epsilon$; however, the adversary's goal is to make sure the classifier cannot accurately predict on the perturbed data.

We simplify the NN's mapping function f with parameters θ as $f(x;\theta)$. Even though the difference between the adversarial input x_{adv} and benign input x is a small perturbation η, the output difference $\delta = f(x+\eta;\theta) - f(x;\theta)$ is not a linear increase with η. In fact, the impact of δ can be learned by AML techniques and change the label sign by calculating back-propagated gradients. Therefore, we can expect that small perturbations in the input add up to change the expected label of the original output. The adversarial perturbation is formally given by

$$\eta = \epsilon \text{sign}(\nabla_x L(x, y; \theta)). \tag{2}$$

where $L(x, y; \theta)$ is the loss function of the model with parameters θ [11]. The adversarial data are generated by maximizing the loss function with respect to the classifier's input x based on the gradients $\nabla_x L(x, y; \theta)$. The final adversarial perturbation is given by:

$$x_{adv} = x + \epsilon \text{sign}(\nabla_x L(x, y; \theta)). \tag{3}$$

The gradient ∇_x can be computed via back-propagation, where the loss function of the perturbed signal is $L(x_d^{t,a}, y_d^{\text{true}}; \theta) \approx L(x_d^t, y_d^{\text{true}}; \theta) + \eta^T \nabla_{x_d^t} L(x_d^t, y_d^{\text{true}}; \theta)$. The final optimization of the adversarial perturbation becomes one to maximize the loss function subject to $\eta = \epsilon \nabla_x L(x_d^t, y_d^{\text{true}}; \theta)$.

The scaling factor ϵ controls the power of the perturbation. If ϵ is increased then the perturbation can have larger impact on the input x_d^t which will result in a poor accuracy on the adversarial dataset $x_d^{t,a}$. To show the energy level of the proposed perturbations, we define the Signal to Perturbation Ratio (SPR) between the received signal and the perturbation as $E(x_d^t)/E(\eta)$ in dB. We will estimate the relationship between ϵ and SPR in the next section. As an example for the classification of the perturbed signal, Figure 3 shows the confusion matrix of the proposed bidirectional LSTM structure (b) on the benign and adversarial datasets. The neural network can successfully classify the waveforms from LTE, 5G NR, and Wi-Fi signals on the benign dataset. This model can achieve a relatively high classification performance on each category of waveforms. However, the performance of the same networks drops to 8.3% by adding the FGSM based perturbations with $\epsilon = 0.3$! The confusion matrix in Figure 3(b) shows that the legitimate user classifies the waveforms into the wrong labels on the adversarial dataset. For example, 92% of the received LTE signals are labeled as the 5G NR signals. Moreover, all the Wi-Fi waveforms are classified into the LTE and 5G NR signals. Such misclassification results in a poor accuracy for the legitimate user and increase the packet loss.

V. Adversarial Attacks with Limited Knowledge

The white-box attack scenario, while the most effective attack, is not realistic. Therefore, we consider scenarios where the attacker only has access to partial information from the defender. We divide such knowledge into classifier and data domains. The white-box attack is performed when the adversary knows all the information needed in the classifier and data domain. However, the defender can protect some of their information, and it becomes challenging to eavesdrop. We consider the different levels of knowledge for the attacker in both domains to evaluate the accuracy under limited information leakage scenarios.

A. Limited Knowledge of Defender's Classifier

In real-world environments, the attacker tries to eavesdrop to obtain the information about the classification model so they can generate attacks. When all the information has eavesdropped, it becomes a white-box attack. White-box attacks are a strong assumption of the knowledge of the attacker. Therefore, we consider a more realistic situation, where the attacker learns a classifier $f_a(x;\theta_a)$ based on different knowledge levels of the defender's classifier $f_d(x;\theta_d)$.

Attack Scenario (a): The attacker knows all the hyperparameters of the defender but does not know the trained weights. The classifier is trained with the same architecture for the defender and adversary; however, the final trained classifiers will be different even under the same hyperparameter setting and the same training dataset (i.e., due to random initialization, etc.). As a result, the two classifiers will have different weights even they have similar classification performances. In this case, we use two different random seeds to initialize the models before the training.

Attack Scenario (b): In this attack, the adversary knows the overall structure of the DNN but does not know the other hyperparameters. It is a more realistic attack, where the attacker eavesdrops on the defender and learns the structure instead of all the settings of a model. For example, the attacker may know the defender is using a seven-layer CNN model with *Conv1D* as the first two layers but does not know the filter numbers of these layers. However, such filter number (or the unit number for RNN) can significantly impact gradient backpropagation, forcing the DNN to end up with different weights after the training. Therefore, we consider the attack that knows the layer numbers, types, and orders but does not know the filter numbers of the layers.

Attack Scenario (c): The attacker knows the classifier type (CNN or RNN) but does not know the structure. In this attack, we use a differently structured classifier at the attacker side to generate the adversarial perturbations. Mostly, we consider using the same type of the DNN model but with different layer numbers (e.g., we use a three-layer RNN structure (a) for the defender but use a two-layer structure (c) for the attacker).

Attack Scenario (d): The attacker does not know the classifier type. We use $f(x;\theta)$ to present the DNN. The mapping function f can differ significantly with classifier types, especially if a CNN represents features much differently than an RNN. In this scenario, we consider the situation when the attacker uses RNN based classifier to generate the adversarial perturbations, but the defender uses the CNN-based classifier as the detector and vice versa.

B. Limited Knowledge of Defender's Training Data

In the real environment, benign waveforms received by the attacker are $x_a^t = \mathbf{H}_{ta}x_o + n$, and signals received by the defender are $x_d^t = \mathbf{H}_{td}x_o + n$. Considering the channel impact, the transmissions received by the attacker and the receiver are different. Therefore, the attacker needs to train its own classifier f_a based on the dataset x_a^t. Due to the training data being different from the defender's, the trained parameters θ_a will be different even with the same hyperparameter setting. As a result, the adversarial perturbations must to be generated with $f_a(x;\theta_a)$. The loss function $L(x_d^{t,a}, y_d^{\text{true}};\theta_d)$ is approximated by $L(x_a^t, y_a^{\text{true}};\theta_a) + \eta^T \nabla_{x_a^t} L(x_a^t, y_a^{\text{true}};\theta_a)$. We denote this type of adversarial signal as *Attack Scenario (e)*: The attacker gains a different dataset to train its classifier. The signal is broadcasted by the legitimate transmitter, so the attacker and the defender will receive the waveforms that contain the same bit-level information. Due to the channel impact and the noise, datasets are different in baseband waveforms. We consider the AWGN channel between all the communication nodes, and same levels of SNR for the transmission received defender and the attacker.

VI. Performance Evaluation

A. Data Generation

The *Matlab Communication* and the *5G Toolboxes* are used to generate the waveforms of the LTE, Wi-Fi, and 5G NR signals. A set of signal features, including channel bandwidth,

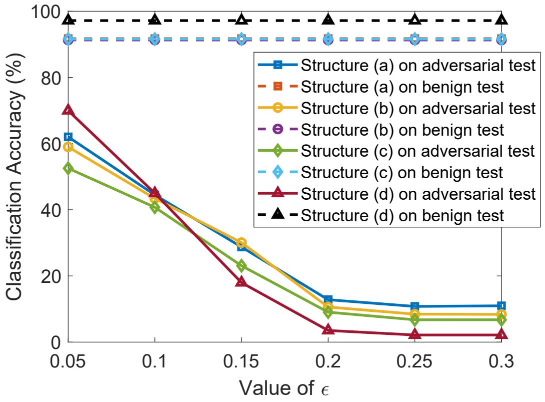


Fig. 4. Accuracy of proposed DNN classifiers under benign and AML-based perturbations (white-box attack).

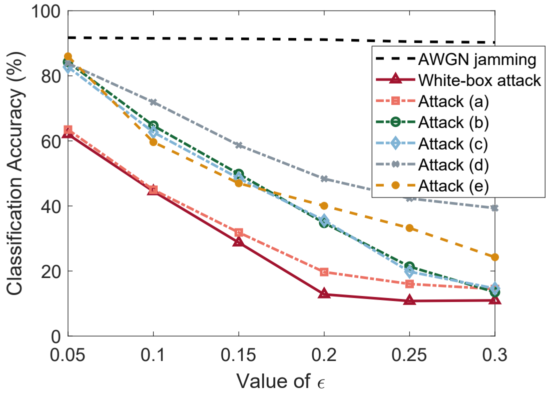


Fig. 5. Accuracy of RNN classifier (structure (a)) under different attacks.

modulation schemes, I/Q imbalance, DC offset, and subcarrier spacing can be adjusted by the protocols. By knowing these features, we simulate waveforms of the three technologies under different parameter settings supported by the standards. Of the various possible features, we consider the baseband I/Q samples at the receiver (with added noise) as input to the classifier. I/Q samples are obtained before decoding the signal, and they provide a rich representation of the actual waveform.

These samples are divided into multiple sequences by applying a sliding window with a step size of one, each consisting of 512 I/Q pairs. These sequences are used as datasets to train and test the classifiers. In this paper, we assume all protocols are transmitted in the same center frequency and have a channel bandwidth of 20 MHz. In addition, we consider the LTE, Wi-Fi, and 5G NR signals that are transmitted under an AWGN channel with SNR = 15 dB. The Wi-Fi waveform is transmitted by generating baseband samples of 802.11 ac (VHT) with BPSK modulation and $1/2$ rate. The LTE waveform is generated by downlink RMC with the reference channel of R.9, which has a 64 QAM modulation. We also generate 5G waveforms using 5G DL FRC with QPSK modulation, a rate of $1/3$ with a subcarrier spacing of 15 kHz.

B. Impact of White-box Adversarial Attack

We evaluate the four neural network architectures in Figure 2 and present the accuracy of the defender's classifier under the benign (i.e, AWGN) and adversarial perturbations. As shown in Figure 4, the RNN-based models (a)-(c) achieve approximately 91% accuracy under benign perturbations, while the CNN structure (d) can achieve 97% accuracy. The three RNN structures (a) and (c) have comparable performance because of their comparable bidirectional LSTM designs. We also observe that structure (a) performs the best when ϵ is larger than 0.15. The accuracy drops for all four classifiers as we increase the magnitude of the adversarial perturbations via ϵ. Even though the CNN achieves the best performance under benign perturbations, it suffers more from the AML attacks. When ϵ is greater than 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, which indicates the white-box attack can mislead the defender's classifier with very limited power control. To show the energy level of the proposed perturbations, we calculate the SPR as shown in Table I. The ratio drops faster with smaller ϵ, and the trend slows down when ϵ becomes larger.

TABLE I
RELATIONSHIP BETWEEN $E(x_d^t)/E(\eta)$ AND ϵ

ϵ	0.05	0.1	0.15	0.2	0.25	0.3
SPR (dB)	24.14	18.32	14.59	12.10	10.17	8.58

C. Impact of Limited-Knowledge Attacks

After testing the white-box attacks, we consider attack scenarios where the attacker has incomplete knowledge of the defender's classifier and/or the training dataset used by the defender. The attack scenarios (a)-(e) are as described in Section V. As shown in Figure 4, structure (a) has the best performance, so we explore its accuracy changes under different attacks and use it to represent RNN models.

1) Attacks on the RNN model: The accuracy for structure (a) is presented in Figure 5. The impact of attack (a) is close to the white-box attack, and it is 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 (b) exchanges the filter number of the first two layers, and attack (c) uses one less layer (e.g., remove the third layer of structure (a)) for the attacker. Both of them show similar performance as the defender, which means these hyperparameters have comparable influences. Attack (d) has the worst attack effect. This is because that the attacker applies the CNN structure (d) to generate the adversarial signals for the RNN model. Even though both types of the classifier can classify the received waveforms accurately, the actual trained weights can differ significantly from the other's. Therefore, a well-crafted perturbation for the CNN may not achieve the expected effect on RNNs. Attack (e) uses the different training datasets to generate the perturbations. Thus, it shows more variance than other attacks. It has an

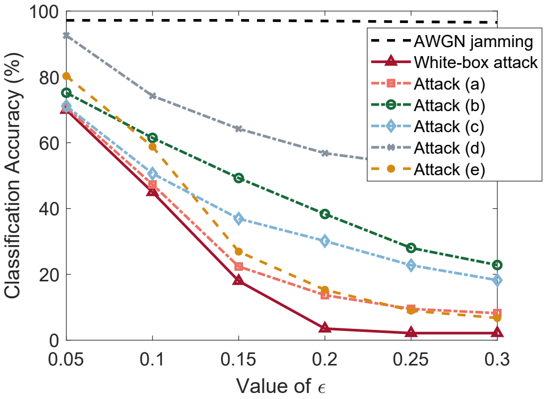


Fig. 6. Accuracy of a CNN-based classifier under different attacks.

equivalent trend with the attack (b) and (c) but slows down when ϵ exceeds 0.15.

2) Attacks on the CNN Classifier: The accuracy for structure (d) is presented in Figure 6. Similar to what was observed in the RNN model, the impact of the attack depends heavily on the attacker's knowledge levels. Attack (a) was the closest one to the white-box attack. In the simulation, attack (b) exchanges the filter number of the two *Conv1D* layers, and attack (c) removes the second *Conv1D* layer at the attacker side. Compared with the RNN model, the layer and filter number setting play a more important role in CNNs. As a result, attacks (c) and (d) show different trends with ϵ. Attack (e) shows strong similarity with the attack (a), which implies the CNN model can have a more severe attack than the RNN model even when the attacker has limited knowledge of the data.

VII. Conclusions

We studied the vulnerability of DNN-based protocol classifiers to AML-based jamming attacks, considering a shared spectrum scenario with Wi-Fi, LTE, and 5G NR transmissions. Several DNN designs were proposed, including a CNN and three RNN structures (with forward and bidirectional LSTM networks). First, we showed that under "benign" (random) noise, all four classifiers exhibit high classification accuracy (above 90%). Replacing this random noise with adversarial FGSM-based perturbations while maintaining almost the same SNR level, all four DNNs were shown to suffer significant reduction in the classification accuracy. 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 different levels of knowledge. In one extreme, the attacker has full knowledge of the defender (white-box attack). We observed that DNNs used for protocol classification are vulnerable to these attacks even the attacker has limited knowledge. Compared to traditional jamming, where the attacker transmits only random noise, the proposed FGSM based attack requires much less transmit power to mislead the classifiers.

VIII. Acknowledgements

This material is based upon work supported by the U.S. Army Small Business Innovation Research Program Office and the Army Research Office under Contract No. W911NF-21-C-0016.

References

[1] 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)*, 2021, pp. 1–10.

[2] 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.

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

[4] T. O'Shea and J. Hoydis, "An introduction to deep learning for the physical layer," *IEEE Transactions on Cognitive Communications and Networking*, vol. 3, no. 4, pp. 563–575, 2017.

[5] T. J. O'Shea, T. Roy, and T. C. Clancy, "Over-the-air deep learning based radio signal classification," *IEEE Journal of Selected Topics in Signal Processing*, vol. 12, no. 1, pp. 168–179, 2018.

[6] 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)*, 2019.

[7] 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)*, 2019, pp. 370–378.

[8] 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.

[9] 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.

[10] 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.

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

[12] K. Eykholt, I. Evtimov, E. Fernandes, B. Li, A. Rahmati, C. Xiao, A. Prakash, T. Kohno, and D. Song, "Robust physical-world attacks on deep learning visual classification," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 1625–1634.

[13] Y. Shi and Y. E. Sagduyu, "Evasion and causative attacks with adversarial deep learning," in *Prof. of the IEEE Military Communications Conference (MILCOM)*, 2017, pp. 243–248.

[14] 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*, 2020.

[15] 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.

[16] 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)*, 2020, pp. 2469–2478.

[17] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving *et al.*, "Tensorflow: A system for large-scale machine learning," in *Proc. of the USENIX symposium on operating systems design and implementation (OSDI)*, 2016, pp. 265–283.

[18] Z. Huang, W. Xu, and K. Yu, "Bidirectional LSTM-CRF models for sequence tagging," *arXiv preprint arXiv:1508.01991*, 2015.

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