Robust Automatic Modulation Classification in the Presence of Adversarial Attacks

Rajeev Sahay, David J. Love, and Christopher G. Brinton
School of Electrical and Computer Engineering, Purdue University
{sahayr,djlove,cgb}@purdue.edu

***Abstract*—Automatic modulation classification (AMC) is used in intelligent receivers operating in shared spectrum environments to classify the modulation constellation of radio frequency (RF) signals from received waveforms. Recently, deep learning has proven capable of enhancing AMC performance using both convolutional neural networks (CNNs) and recurrent neural networks (RNNs). However, deep learning-based AMC models are susceptible to adversarial attacks, which can significantly degrade the performance of well-trained models by adding small amounts of interference into wireless RF signals during transmission. In this work, we present a two-fold defense mechanism to withstand adversarial interference on modulated radio signals. Specifically, our method consists of (1) correcting misclassifications on mild attacks and (2) detecting the presence of an adversary on more potent attacks. We show that our proposed defense is capable of withstanding adversarial interference injected into RF signals while maintaining false positive detection rates on CNNs and RNNs as low as 3%.**

I. Introduction

Automatic modulation classification (AMC) has become increasingly of interest for shared spectrum environments, where overcrowded radio channels inhibit wireless efficiency. AMC aims to directly classify the modulation scheme of radio frequency (RF) signals in a wireless channel using received waveforms. Recently, deep learning (DL) has demonstrated cutting-edge performance on AMC tasks, without requiring computationally expensive derivations of statistical decision boundaries on manually engineered features [1]. Yet, despite these advantages, DL-based AMC models are highly susceptible to *adversarial attacks* [2]. In such noise injection attacks, the adversary introduces small perturbations into wireless signals during transmission resulting in erroneous, yet high-confidence, predictions from deep learning AMC classifiers. As a result, an adversary can inject additive adversarial interference into an RF signal during transmission to inhibit reliable communications.

In this work, we develop a two-fold defense methodology to mitigate the effects of an adversary on deep learning-based AMC classifiers. Specifically, we propose a detection and a mitigation strategy, where the former rejects identified adversarial signals for classification while the latter mitigates the effects of subtle adversarial interference bypassed by the detector (thus, reducing misclassification on perturbed signals). In applying our methodology on two custom trained deep learning architectures, we find that it is able to detect and mitigate wireless adversarial interference effectively under different bounds of noise injection perturbations.

This work was supported in part by the Naval Surface Warfare Center Crane Division and in part by the National Science Foundation (NSF) under grants AST-2037864, CNS1642982, CCF1816013, and EEC1941529.

Related Work: Both convolutional neural networks (CNNs) [3] and recurrent neural networks (RNNs) [1], [4] have achieved robust AMC performance using different architectures such as AlexNet [5] and ResNet [6] on clean RF signals (i.e., signals not corrupted by adversarial interference). However, AMC DL classifiers are known to be vulnerable to RF signals injected with adversarial interference [7], [8], and as a result, adversarial susceptibility has been cited as one of the primary challenges to widespread deep learning adoption in wireless communications [9]. Defense mechanisms such as denoising autoencoders [10] and Gaussian smoothing [11] have been proposed to strengthen AMC DL models in the presence of adversarial attacks, but they are still susceptible to high-powered adversarial interference signals. Our proposed methodology exposes the presence of adversarial interference in such cases and provides mitigation when the interference has reached perceptible levels.

In computer vision, adversarial retraining [12] has been proposed as an effective defense mechanism against visually imperceptible adversarial attacks. Although adversarial retraining is known to be effective on mild attacks, classifiers trained on adversarial examples demonstrate degraded performance on potent attacks, making direct adoption of retraining in defending AMC DL classifiers ineffective. Manifold learning methods have also been proposed to detect subtle perturbations for image classification [13], but they have been shown to be ineffective when the adversary can account for the detection model [14]. In this work, we will show that adversarial retraining in conjunction with manifold learning-based adversarial detection is effective for both low and high-powered additive interference experienced in wireless communication channels.

Outline and Summary of Contributions: In this paper, we begin by defining our target AMC DL classifiers and comparing their training accuracy and resource utilization (Sec. II-B and III-B). We then demonstrate the susceptibility of each trained target model to two distinct adversarial attacks (Sec. II-C). Finally, we propose a novel two-step mechanism for defending and detecting adversarial DL-

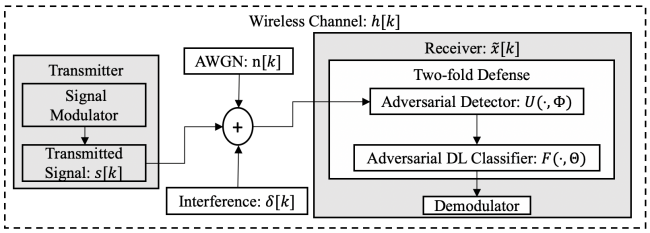

Fig. 1: Our wireless system model and proposed receiver architecture for detecting and mitigating adversarial interference.

based wireless interference (Sec. II-D) and demonstrate its effectiveness against l_2 and l_∞-bounded perturbations (Sec. III-C and III-D).

II. AMC Model and Defense Methodology

In this section, we present our wireless channel model (Sec. II-A), the deep learning-based AMC models under attack (Sec. II-B), and the adversarial interference process (Sec. II-C). Finally, we present our proposed defense and detection methods (Sec. II-D), which mitigate the effects of the adversarial attacks. Our overall system model is depicted in Fig. 1; the AMC model and two-fold defense are contained in the receiver.

A. Signal Modeling

We consider an ℓ-dimensional modulated signal, $\mathbf{s} = [s[0], \ldots, s[\ell-1]]^T$, which is transmitted over a wireless channel. The channel introduces sample rate offset (SRO), center frequency offset (CFO), selective fading, and additive white Gaussian noise (AWGN) to the signal. The received signal for sample k, denoted by $x[k]$, is modeled as

$$x[k] = s[k] * h[k] + n[k], \tag{1}$$

where $*$ denotes convolution, $h[k]$ is the channel's impulse response and includes radio imperfections, and $n[k]$ is complex AWGN with each noise sample distributed as $\mathcal{CN}(0, N_0)$. Note that although all signals are complex, i.e., $\mathbf{x} \in \mathbb{C}^\ell$, we will follow prior AMC work and represent all signals as two-dimensional reals, i.e., $\mathbf{x} \in \mathbb{R}^{\ell \times 2}$, using the in-phase (I) and quadrature (Q) components of the signal, where ℓ denotes the length of the observation window.

B. Target Classifier Architectures

After aggregating a set of modulated data signals, $\mathcal{X} \subset \mathbb{R}^{\ell \times 2}$, where each input, $\mathbf{x} \in \mathcal{X}$, belongs to one of C modulation constellations, we train a DL classifier, which we denote as $f(\cdot;\boldsymbol{\theta})$: $\mathbb{R}^{\ell \times 2} \rightarrow \mathbb{R}^C$, where $\boldsymbol{\theta}$ denotes the model parameters. The trained classifier assigns each input $\mathbf{x} \in \mathcal{X}$ a label $\hat{\mathcal{C}}(\mathbf{x}, \boldsymbol{\theta}) = \text{argmax}_i\, f_i(\mathbf{x}, \boldsymbol{\theta})$, where $f_i(\mathbf{x}, \boldsymbol{\theta})$ is the classification probability that $\mathbf{x}$ belongs to the i^{th} modulation constellation for $i = 1, \ldots, C$. We consider two DL architectures for $f(\cdot, \boldsymbol{\theta})$ consisting of both convolutional and recurrent-based layers.

Convolutional Neural Nets: CNNs consist of one or more *convolutional layers*, which extract spatially correlated information from each layer's input. The primary hyperparameters of each layer include the number of filters per layer, denoted as F, and the filters' kernel size in each layer. We denote the filter kernels as $m_p \in \mathbb{R}^{L \times W}$, where L and W are the length and width of the filter, respectively. The output of a convolutional layer produces F outputs (termed feature maps). Each element in the p^{th} feature map of a convolutional layer, for $p = 1, \ldots, F$, is given by

$$y_p[j,k] = \sum_{l=0}^{L-1} \sum_{w=0}^{W-1} x[j+l, k+w] m_p[l,w], \tag{2}$$

where the input, $\mathbf{x}$, and the p^{th} filter, m_p, are cross-correlated to produce the output $\mathbf{y}$ indexed at j and k. The parameters of each filter, m_p, in each layer are estimated from the training data. Our proposed CNN architecture along with its training details are presented in Tables I and III.

Recurrent Neural Nets: RNNs create memory to correlate earlier input features to delayed features in the input signal. Long-short-term-memory (LSTM) cells [15] extend recurrence in neural networks by implementing three gates for learning. *Input gates* prevent irrelevant features from entering the recurrent layer while *forget gates* eliminate irrelevant features altogether. *Output gates* produce the LSTM layer output, which is inputted into the subsequent network layer. The gates are used to recursively calculate the internal state of the cell, denoted by $\mathbf{z}_c^{(t)}$ at time t (a specific recursive iteration) for cell c, which is then used to calculate the cell output, defined as

$$\mathbf{q}^{(t)} = \tanh(\mathbf{z}_c^{(t)}) \sigma(\mathbf{p}^{(t)}), \tag{3}$$

where $\mathbf{p}^{(t)}$ is the parameter obtained from the output gate of the cell and $\sigma(\cdot)$ is the logistic sigmoid function given by $\sigma(p_i^t) = 1/(1+e^{-p_i^t})$ for the i^{th} element in $\mathbf{p}^{(t)}$. The model parameters are estimated during training. Our proposed LSTM-based RNN architecture and the training details we employ are shown in Tables II and III, respectively.

TABLE I: Proposed CNN architecture. The shapes of the convolutional layers correspond to $L \times W \times F$.

Layer	Dropout Rate (%)	Activation	Shape
Input	-	-	$2 \times \ell \times 1$
Conv 1	20	ReLU	$2 \times 5 \times 256$
Conv 2	20	ReLU	$1 \times 4 \times 128$
Conv 3	20	ReLU	$1 \times 3 \times 64$
Conv 4	20	ReLU	$1 \times 3 \times 64$
FC 1	-	ReLU	128
Output	-	Softmax	C

TABLE II: Proposed RNN architecture consisting of an LSTM layer with 256 units followed by a fully connected layer with 128 units.

Layer	Activation	Output Shape
Input	-	$2 \times \ell$
LSTM	-	256
FC	ReLU	128
Output	Softmax	10

TABLE III: Model training parameters. The CCE cost function refers to categorical cross entropy.

Parameter	CNN	RNN
Cost Function	CCE	CCE
Optimizer Algorithm	Adam	Adam
Learning Rate	0.001	0.001
Batch Size	256	64
Epochs	100	100

Subsampling: To evaluate the tradeoff between training efficiency and the trained model's performance, we will evaluate the training times and classification performances of both the CNN and RNN in Section III using various subsampled input representations. A subsampling rate of d, $0 < d \leq 1$, yields the signal of $d \cdot l$ evenly spaced points from the original observation window. We will see that each model is susceptible to adversarial interference, and we will evaluate the effectiveness of our defense on different subsampling rates.

C. Adversarial Attack Models

Adversarial interference injected into transmitted signals is intended to alter the classification decision of $f(\cdot;\boldsymbol{\theta})$. We model the additive adversarial interference as $\boldsymbol{\delta} \in \mathbb{R}^{\ell \times 2}$, with the perturbed signal given by

$$\tilde{\mathbf{x}} = \mathbf{x} + \boldsymbol{\delta}. \tag{4}$$

We focus on crafting $\boldsymbol{\delta}$ according to the l_p norm-constrained class of noise injection attacks, which are common in AMC settings [7]. In general, multiple methodologies exist to craft adversarial interference signals and, therefore, a chosen $\boldsymbol{\delta}$ may not necessarily be a unique perturbation. Formally, $\boldsymbol{\delta}$ is calculated by solving an optimization problem of the form

$$\begin{aligned} \min_{\boldsymbol{\delta}} \quad & ||\boldsymbol{\delta}||_p \\ \text{s. t.} \quad & \hat{\mathcal{C}}(\mathbf{x},\boldsymbol{\theta}) \neq \hat{\mathcal{C}}(\mathbf{x}+\boldsymbol{\delta},\boldsymbol{\theta}) \quad \text{and} \quad \mathbf{x}+\boldsymbol{\delta} \in \mathcal{X}. \end{aligned} \tag{5}$$

In (5), $||\cdot||_p$ denotes the l_p norm of the perturbation, and the constraints induce misclassification while keeping $\tilde{\mathbf{x}}$ in the same space as $\mathbf{x}$. In this work, we consider perturbations crafted using the fast gradient sign method (FGSM) [16] under both an l_∞-norm constraint and an l_2-norm constraint. We assume the most vulnerable exposure to the adversary, known as a *white box* threat model, where the adversary has full access to the trained model and its parameters. Specifically, the adversary is completely aware of $f(\cdot;\boldsymbol{\theta})$ including its architecture and parameters.

l_∞-bounded attack: The l_∞-bounded FGSM adds a small perturbation, $\epsilon \in \mathbb{R}^{\ell \times 2}$, to each feature of the input sample in the direction of the sign of the classifier's cost function (categorical cross entropy in our case), $J(\mathbf{x}, y, \boldsymbol{\theta})$, which is a function of the input sample, $\mathbf{x}$, its ground truth label, y, and the model parameters, $\boldsymbol{\theta}$. Formally, the l_∞ crafted perturbation is given by

$$\tilde{\mathbf{x}} = \mathbf{x} + \epsilon \cdot \text{sgn}(\nabla_x J(\mathbf{x}, y, \boldsymbol{\theta})). \tag{6}$$

TABLE IV: Convolutional autoencoder architecture. The shapes of the convolutional layers correspond to $L \times W \times F$.

Layer	Activation	Shape
Input	-	$2 \times \ell \times 1$
Conv 1	Linear	$3 \times 3 \times 64$
Conv 2	Linear	$3 \times 3 \times 32$
Conv 3	Linear	$3 \times 3 \times 16$
Conv 4	Linear	$3 \times 3 \times 32$
Conv 5	Linear	$3 \times 3 \times 64$
Conv 6	Linear	$3 \times 3 \times 1$

Employing (6) moves the sample, $\mathbf{x}$, in the direction of the high-parameter neural network's decision boundary, and misclassificaton is induced when the sample crosses the decision boundary. The adversary here is not limited to adding an imperceptible perturbation, and therefore, the ϵ bound added to the sample can vary widely.

l_2-bounded attack: The l_2-bounded FGSM is natural to consider for wireless signals as it corresponds to the signal power of the transmission. Specifically, a perturbation, $\alpha \in \mathbb{R}^{\ell \times 2}$, is added to the signal, $\mathbf{x}$, by

$$\tilde{\mathbf{x}} = \mathbf{x} + \alpha \frac{\nabla_x J(\mathbf{x}, y, \boldsymbol{\theta})}{||\nabla_x J(\mathbf{x}, y, \boldsymbol{\theta})||_2}, \tag{7}$$

where J refers to the classifier's cost function as in (6). In Sec. III-C and III-D, we demonstrate both the potency of the FGSM attack and the effectiveness of our proposed defense across a wide range of perturbation bounds.

D. Adversarial Detection and Mitigation

We now develop our two-fold defense strategy against adversarial interference. Given a dataset of modulated signals, $\mathcal{X} = \{\mathbf{x} \in \zeta\}$, where ζ is the universal set containing all signals, $\mathbf{x}$, we perform a random split to attain two disjoint subsets, $\mathcal{X}_{\text{train}}$ and $\mathcal{X}_{\text{test}}$ at the receiver. The subset $\mathcal{X}_{\text{train}} \subset \mathcal{X}$ is used for optimizing the defense algorithms, and $\mathcal{X}_{\text{test}} \subset \mathcal{X}$ is used to evaluate the efficacy of our proposed defense. Our defense consists of a mitigation and a detection component.

Mitigation: The mitigation portion of our proposed method is concerned with correctly classifying adversarially perturbed inputs. To achieve this, we re-train $f(\cdot;\boldsymbol{\theta})$ using adversarial inputs generated on $\mathcal{X}_{\text{train}}$ using (6) for different ϵ bounds and arrive at $F(\cdot;\boldsymbol{\Theta})$. $F(\cdot;\boldsymbol{\Theta})$ can withstand adversarial perturbations to a greater extent than $f(\cdot;\boldsymbol{\theta})$ due to its augmented training set, which includes inputs artificially injected with adversarial interference. Algorithm 1 describes our method used to generate $F(\cdot;\boldsymbol{\Theta})$. In this algorithm, we select a set of upper perturbation bounds, which are used to train $F(\cdot;\boldsymbol{\Theta})$ to mitigate multiple bounds of injected interference.

Detection: For detection, we propose a manifold learning approach using a convolutional autoencoder. We denote this autoencoder as $U(\cdot;\boldsymbol{\Phi})$, where $\boldsymbol{\Phi}$ denotes the parameters of the autoencoder. We use an encoding function, $h : \mathbb{R}^{\ell \times 2} \rightarrow \mathbb{R}^{k \times 2}$ to map an input signal $\mathbf{x}_i$ to a latent space representation. Then, a decoding function, $g : \mathbb{R}^{k \times 2} \rightarrow \mathbb{R}^{\ell \times 2}$, is used

Algorithm 1 AMC Adversarial Mitigation

input: $f(\cdot;\boldsymbol{\theta})$: trained classifier
$\mathcal{X}_{\text{train}}$: training data set
$\boldsymbol{\eta}$: upper perturbation bounds
initialize: $\tilde{\mathcal{X}}[n] \leftarrow 0 \ \ \forall n$
for $\mathbf{x}_n \in \mathcal{X}_{\text{train}}$ **do**
  for $\eta_i \in \boldsymbol{\eta}$ **do**
    $\tilde{\mathbf{x}}_n \leftarrow \mathbf{x}_n + \eta_i \cdot \text{sgn}(\nabla_x J(\mathbf{x}_n, y, \boldsymbol{\theta}))$
    $\tilde{\mathcal{X}}[n] \leftarrow \tilde{\mathbf{x}}_n$
  end for
end for
$\mathcal{X}_{\text{total}} \leftarrow [\mathcal{X}, \tilde{\mathcal{X}}]$
$F(\cdot;\boldsymbol{\Theta}) \leftarrow$ train $f(\cdot;\boldsymbol{\theta})$ on $\mathcal{X}_{\text{total}}$
return $F(\cdot;\boldsymbol{\Theta})$

Algorithm 2 Constructing AMC Adversarial Detector

input: $\mathcal{X}_{\text{train}}$: training data set
ϵ: upper perturbation bound
initialize: $\tilde{\mathcal{X}}[n] \leftarrow 0 \ \ \forall n$
$E[n] \leftarrow 0 \ \ \forall n$
$U(\cdot;\boldsymbol{\Phi}) = \underset{h,g}{\text{minimize}} || \frac{1}{N} \sum_{\mathbf{x}_i \in \mathcal{X}_{\text{train}}} (\mathbf{x}_i - g(h(\mathbf{x}_i)))||^2$
for all $\mathbf{x}_n \in \mathcal{X}_{\text{train}}$ **do**
  $\tilde{\mathbf{x}}_n \leftarrow \mathbf{x}_n + \epsilon \cdot \text{sgn}(\nabla_x J(\mathbf{x}_n, y, \boldsymbol{\theta}))$
  $E[n] \leftarrow || \frac{1}{N} \sum_{i=1}^{N} (x_i - U(\tilde{x}_i;\boldsymbol{\Phi}))||^2$
end for
$T = \max(\text{E})$
return $T, U(\cdot;\boldsymbol{\Phi})$

to reconstruct an approximation of the input, $\hat{\mathbf{x}}_i \in \mathbb{R}^{\ell \times 2}$. Intuitively, the reconstruction cost will be higher when an adversary has injected noise into the input.

The parameters of the encoder and decoder are simultaneously optimized, using the mean squared error function, to produce the autoencoder given by

$$U(\cdot;\boldsymbol{\Phi}) = \underset{h,g}{\text{minimize}} \left|\left| \frac{1}{N} \sum_{\mathbf{x}_i \in \mathcal{X}_{\text{train}}} (\mathbf{x}_i - g(h(\mathbf{x}_i))) \right|\right|^2. \quad (8)$$

The model is trained using the Adam optimizer, and its architecture is shown in Table IV. After optimizing $U(\cdot;\boldsymbol{\Phi})$, it is used for building and deploying the adversarial detector using the procedures described in Algorithms 2 and 3, respectively. Specifically, the reconstruction loss of $U(\cdot,\boldsymbol{\Phi})$ is used to measure the distance of a sample from the training data manifold, where samples beyond a threshold, T, are considered adversarial. In Algorithm 3, assigning an input sample to a value of one is equivalent to a positive adversarial detection.

Algorithm 3 Applying AMC Adversarial Detector

input: T: pre-determined threshold
$U(\cdot;\boldsymbol{\Phi})$: trained autoencoder
$\mathbf{x}_i$: input sample
$T_i = || \sum_{i=1}^{N} (\mathbf{x}_i - U(\mathbf{x}_i;\boldsymbol{\Phi}))||^2$
if $T_i \geq T$ **then**
  $\mathbf{x}_i \leftarrow 1$
else if $T_i < T$ **then**
  $\mathbf{x}_i \leftarrow 0$
end if

III. Evaluation Results and Discussion

In this section, we begin by discussing the dataset employed to evaluate our proposed defense methodology (Sec. II-A). We then evaluate our AMC models in the absence of an attack (Sec. II-B) and show the ability of our proposed defense to mitigate adversarial interference on both CNNs (Sec. II-C) and RNNs (Sec. II-D).

A. Dataset and Evaluation Procedure

To evaluate our methodology, we leverage the RadioML 2016.10b dataset [17], which consists of 60,000 128×2 wireless IQ signals at different signal to noise ratios (SNRs). Each signal is modulated according to one of the following ten schemes (eight digital (D) and two analog (A) constellations): BPSK (D), QPSK (D), 8PSK (D), QAM16 (D), QAM64 (D), GFSK (D), CPFSK (D), PAM4 (D), WB-FM (A), and AM-DSB (A). The modulated signals range from 0 to 18 dB in increments of 2 dB, and they are normalize to unit energy.

We perform an 80/20 split of the signals at each SNR to construct $\mathcal{X}_{\text{train}}$ and $\mathcal{X}_{\text{test}}$, respectively. We evaluate each model in terms of accuracy, computational overhead, and susceptibility to adversarial perturbations. These metrics are measured using the classification accuracy on an unperturbed testing set, the classifier training time, and the classification accuracy on different perturbation bounds employed on $\mathcal{X}_{\text{test}}$, respectively.

B. AMC Classifier Performance

We first analyze the AMC performance of the CNN and RNN without the presence of adversarial interference. We consider the effect of different SNRs and subsampling rates on the resulting performance. The trained model accuracies and training times are shown in Fig. 2 and Table V, respectively. Both the CNN and RNN follow similar classification accuracy trends in that (i) the classifiers trained on the full observation window achieve similar performance to the classifiers trained on signals subsampled by $1/2$, and (ii) the performance degrades with more aggressive subsampling rates. Higher downsampling results in faster model training times but degrades the performance until the modulated signal does not contain enough relevant features for effective classification. Although the RNN requires significantly less training time (about 75% lower) than the CNN on the full observation window, the benefit of its computational efficiency is hindered by its poorer accuracy (about 18% lower) compared to the CNN. This indicates that convolutional architectures tend to perform better than recurrent

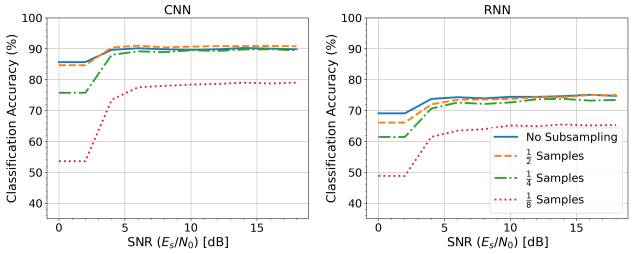

Fig. 2: CNN and RNN classification accuracies at various SNRs. Downsampling and lower SNR degrades accuracy.

TABLE V: Model training times per epoch (in seconds) for different subsampling rates on the CNN and RNN.

Subsamp. Rate	CNN	RNN
None	246	60
1/2	127	55
1/4	70	53
1/8	42	52

architectures when training time is not a factor, but recurrent architectures may be preferred to convolutional models when computational resources for model training are scarce.

C. Defending the CNN-based AMC Classifier

We now evaluate our two-fold defense methodology on the CNN architecture, considering both FGSM attacks and different subsampling rates. The results for the l_∞-bounded perturbation are shown in Fig. 3. The first part of our proposed defense, $F(\cdot, \boldsymbol{\Theta})$, effectively mitigates the misclassification induces from lower perturbation bounds. In particular, when no subsampling is employed, $F(\cdot; \boldsymbol{\Theta})$ is able to achieve an improvement in classification accuracy by a factor of up to 5x for $\epsilon \in [0.00, 0.01]$ (where $\mathbf{x}$ has unit energy) in comparison to the baseline classifier, where no attempt at mitigation is made. Classifiers trained on subsampled inputs show similar performance for small perturbations, but, for more aggressive subsampling rates, the performance of the defense classifier, $F(\cdot; \boldsymbol{\Theta})$, slightly falls below the baseline classifier. This suggests that $F(\cdot; \boldsymbol{\Theta})$ is better suited for defending models trained at a higher dimensionality.

As the perturbation bound grows, the performance of $F(\cdot; \boldsymbol{\Theta})$ degrades and becomes equivalent to the baseline classifier's performance regardless of the subsampling rate. In this scenario, the second part of our proposed defense, $U(\cdot; \boldsymbol{\Phi})$, detects adversarial interference, indicating to the receiver that the prediction of $F(\cdot; \boldsymbol{\Theta})$ may be unreliable. In Fig. 3, once $\epsilon = 0.01$, we achieve high detection rates (above 80%) for each subsampling rate. This exemplifies both portions of the proposed defense working together; the defense metrics show that $F(\cdot; \boldsymbol{\Theta})$ is able to mitigate, to a great extent, the adversarial perturbation on lower bounds when $U(\cdot; \boldsymbol{\Phi})$ has a lower detection rate, whereas $U(\cdot; \boldsymbol{\Phi})$ is able to confidently detect adversarial interference at high bounds when the performance of $F(\cdot; \boldsymbol{\Theta})$ degrades. As a result, $U(\cdot; \boldsymbol{\Phi})$ prevents any adversary from adding a large amount of interference in order to avoid detection. In the limited operating region remaining to the adversary, our proposed defense, $F(\cdot; \boldsymbol{\Theta})$, significantly increases the classification performance in comparison to using the baseline classifier, $f(\cdot; \boldsymbol{\theta})$.

TABLE VI: Attack independent false positive adversarial detection rates for each considered CNN and RNN model.

Subsamp. Rate	CNN FP (%)	RNN FP (%)
None	2.48%	2.24%
1/2	5.69%	5.45%
1/4	18.85%	12.54%
1/8	10.44%	8.21%

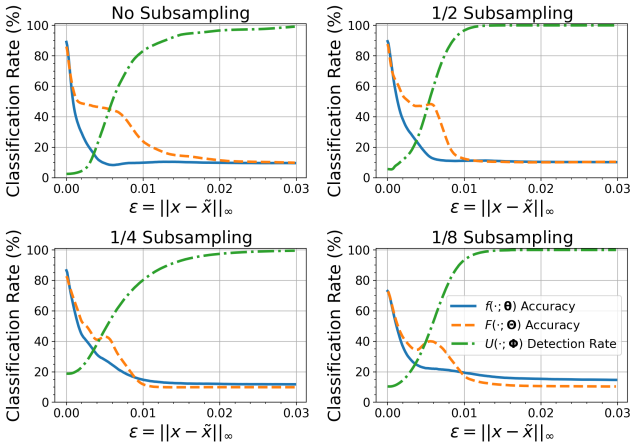

Fig. 3: Defense effectiveness on the CNN against an l_∞-bounded perturbation. For $\epsilon < 0.01$, $F(\cdot, \boldsymbol{\Theta})$ is able to obtain significant improvements in classification performance over $f(\cdot, \boldsymbol{\theta})$, while for $\epsilon > 0.01$, $U(\cdot, \boldsymbol{\Phi})$ has a high detection rate.

Our proposed defense retains similar performance against l_2 bounded attacks on CNNs. Specifically, as shown in Fig. 4, $F(\cdot; \boldsymbol{\Theta})$ achieves higher classification performance than $f(\cdot; \boldsymbol{\theta})$ on lower bounded attacks ($\alpha < 0.1$) followed by degraded performance on higher bounded attacks ($\alpha > 0.1$) in the region where $U(\cdot; \boldsymbol{\Phi})$ achieves high detection rates. This trend is consistent over different subsampling rates. Furthermore, as in Fig. 3, classifiers trained on lower dimensional signals exhibit weaker performance on lower perturbation bounds.

D. Evaluation of Defense on RNN

Fig. 5 shows the result of applying our two-fold defense to the RNN for the l_∞-bounded FGSM attack. Despite the lower baseline classification performance of the RNN, compared to the CNN, the RNN is equivalently susceptible to adversarial attacks. Further, our proposed defense mitigates the FGSM attack similarly on the RNN as with the CNN. Specifically, in each subsampled representation, $F(\cdot; \boldsymbol{\Theta})$ achieves a higher classification performance compared to $f(\cdot; \boldsymbol{\theta})$ on lower bounded attacks ($\epsilon < 0.01$) whereas $U(\cdot; \boldsymbol{\Phi})$ attains nearly perfect detection rates on higher bounded

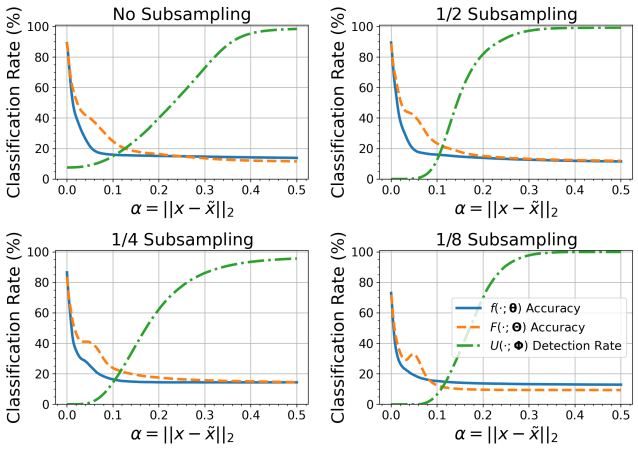

Fig. 4: Defense effectiveness on the CNN against an l_2-bounded perturbation. $U(;\boldsymbol{\Phi})$ again achieves nearly 100% detection at high perturbation bounds while $F(\cdot,\boldsymbol{\Theta})$ increases the classification rate at low perturbation bounds.

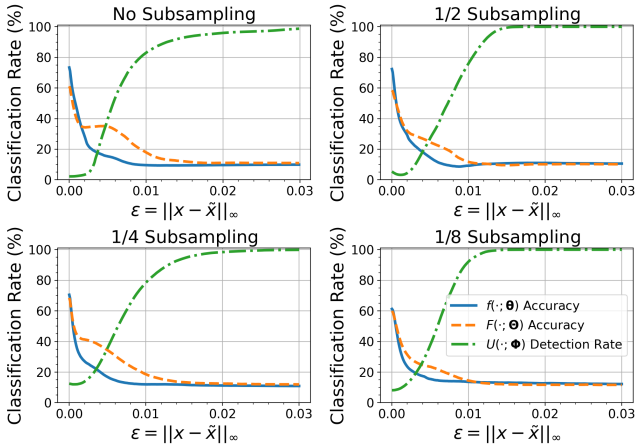

Fig. 5: Defense effectiveness on the RNN against an l_∞-bounded perturbation. Similar to the CNN, the adversary's operating region here is limited to $\epsilon < 0.010$ to avoid probable detection, and in this range of ϵ our mitigation achieves noticeable gains in classification performance.

attacks ($\epsilon > 0.01$). The robust detection rate at high bounded perturbations is critical to inhibiting an adversary from adding a large amount of interference.

The FP rate of $U(\cdot;\boldsymbol{\Phi})$ on each subsampled signal representation is presented in Table VI. $U(\cdot;\boldsymbol{\Phi})$ achieves the lowest FP rate on the full observation window, and the rate increases for higher subsampling rates on both the CNN and RNN. Furthermore, the detectors for both classifiers attain higher FP rates for signals with $1/4$ subsampling compared to $1/8$ subsampling. Thus, although more computationally costly, models trained on the full observation window result in the most secure classifiers with the lowest FP rates on unperturbed RF signals.

IV. Conclusion and Future Work

Deep learning automatic modulation classification (AMC) classifiers have been shown to be susceptible to adversarial interference. In this work, we proposed a two-fold defense strategy capable of detecting and mitigating RF signals perturbed with adversarial interference. We demonstrated the effectiveness of our method on both CNNs and RNNs against two l_p constrained attacks. In future work, we anticipate investigating the effectiveness of our defense in additional threat models in which the adversary may be limited in system knowledge. We also anticipate evaluating our defense in the presence of additional sources of interference such as the superposition of multiple waveforms.

References

[1] 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.

[2] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," *arXiv:1312.6199*, 2013.

[3] F. Meng, P. Chen, L. Wu, and X. Wang, "Automatic modulation classification: A deep learning enabled approach," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 11, pp. 10 760–10 772, 2018.

[4] S. Hu, Y. Pei, P. P. Liang, and Y.-C. Liang, "Robust modulation classification under uncertain noise condition using recurrent neural network," in *2018 IEEE Global Communications Conference (GLOBECOM)*, 2018, pp. 1–7.

[5] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems (NeurIPS)*, 2012, pp. 1097–1105.

[6] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, 2016, pp. 770–778.

[7] 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, 2018.

[8] B. Kim, Y. E. Sagduyu, K. Davaslioglu, T. Erpek, and S. Ulukus, "Over-the-air adversarial attacks on deep learning based modulation classifier over wireless channels," in *2020 54th Annual Conference on Information Sciences and Systems (CISS)*, 2020, pp. 1–6.

[9] Y. Arjoune and S. Faruque, "Artificial intelligence for 5g wireless systems: Opportunities, challenges, and future research direction," in *10th Annual Computing and Communication Workshop and Conference (CCWC)*, 2020, pp. 1023–1028.

[10] S. Kokalj-Filipovic, R. Miller, N. Chang, and C. L. Lau, "Mitigation of adversarial examples in rf deep classifiers utilizing autoencoder pre-training," in *International Conference on Military Communications and Information Systems (ICMCIS)*, 2019, pp. 1–6.

[11] B. Kim, Y. E. Sagduyu, K. Davaslioglu, T. Erpek, and S. Ulukus, "Channel-aware adversarial attacks against deep learning-based wireless signal classifiers," *arXiv preprint arXiv:2005.05321*, 2020.

[12] F. Tramèr, A. Kurakin, N. Papernot, I. Goodfellow, D. Boneh, and P. McDaniel, "Ensemble adversarial training: Attacks and defenses," *arXiv:1705.07204*, 2017.

[13] D. Meng and H. Chen, "Magnet: a two-pronged defense against adversarial examples," in *ACM SIGSAC conference on computer and communications security*, 2017, pp. 135–147.

[14] N. Carlini and D. Wagner, "Magnet and 'efficient defenses against adversarial attacks' are not robust to adversarial examples," *arXiv:1711.08478*, 2017.

[15] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

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

[17] T. J. O'Shea and N. West, "Radio machine learning dataset generation with gnu radio," in *GNU Radio Conference*, vol. 1, no. 1, 2016.