Over-the-Air Adversarial Attacks on Deep Learning Based Modulation Classifier over Wireless Channels

Brian Kim[1], Yalin E. Sagduyu[2], Kemal Davaslioglu[2], Tugba Erpek[2], and Sennur Ulukus[1]

[1]Department of Electrical and Computer Engineering, University of Maryland, College Park, MD 20742, USA
[2]Intelligent Automation, Inc., Rockville, MD 20855, USA

***Abstract*—We consider a wireless communication system that consists of a transmitter, a receiver, and an adversary. The transmitter transmits signals with different modulation types, while the receiver classifies its received signals to modulation types using a deep learning-based classifier. In the meantime, the adversary makes over-the-air transmissions that are received as superimposed with the transmitter's signals to fool the classifier at the receiver into making errors. While this evasion attack has received growing interest recently, the channel effects from the adversary to the receiver have been ignored so far such that the previous attack mechanisms cannot be applied under realistic channel effects. In this paper, we present how to launch a realistic evasion attack by considering channels from the adversary to the receiver. Our results show that modulation classification is vulnerable to an adversarial attack over a wireless channel that is modeled as Rayleigh fading with path loss and shadowing. We present various adversarial attacks with respect to availability of information about channel, transmitter input, and classifier architecture. First, we present two types of adversarial attacks, namely a targeted attack (with minimum power) and non-targeted attack that aims to change the classification to a target label or to any other label other than the true label, respectively. Both are white-box attacks that are transmitter input-specific and use channel information. Then we introduce an algorithm to generate adversarial attacks using limited channel information where the adversary only knows the channel distribution. Finally, we present a black-box universal adversarial perturbation (UAP) attack where the adversary has limited knowledge about both channel and transmitter input. By accounting for different levels of information availability, we show the vulnerability of modulation classifier to over-the-air adversarial attacks.**

I. Introduction

Advances in *deep learning* (DL) based on *deep neural networks* (DNNs) have supported numerous applications to learn from complex data domains such as in computer vision and speech recognition [1]. Following the success of these applications, DL has been applied to wireless communications, where channel, interference, and traffic effects jointly contribute to the high complexity of the spectrum data [2].

Machine learning (ML) in the presence of adversaries have been studied in the context of *adversarial machine learning* [3]. In particular, DNNs are known to be highly susceptible to adversarial attacks, as demonstrated with applications in computer vision domain [4].

This effort is supported by the U.S. Army Research Office under contract W911NF-17-C-0090. The content of the information does not necessarily reflect the position or the policy of the U.S. Government, and no official endorsement should be inferred.

Recently, adversarial ML has been studied in wireless communication systems using DNNs. *Exploratory (inference) attacks* have been considered in [5], where an adversary builds a DNN to learn the transmission pattern in the channel and jam transmissions that would be otherwise successful. Over-the-air spectrum *poisoning (causative) attacks* have been considered in [6], where an adversary poisons (falsifies) a transmitter's spectrum sensing data over the air by transmitting during the short spectrum sensing period of the transmitter. *Trojan attacks* have been studied in [7] against a signal classifier, where an adversary slightly manipulates training data by inserting Trojans in terms of modifying the phases and the labels of only few training data to a target label, and then transmits signals with the same phase shift in the inference time to fool the signal classifier.

Built upon adversarial ML, *adversarial attacks* (a.k.a *evasion attacks*) correspond to small modifications of the original input to the DNNs that make DL algorithm to misclassify the input. These small modifications are not just random noise but carefully designed in a way that changes the decision of the DL algorithm. As an evasion attack, [8] has showed that the end-to-end autoencoder communication systems, proposed in [9], are vulnerable to adversarial attacks in an additive white Gaussian noise (AWGN) channel environment, where the attack increases the block-error-rate at the receiver. Also, adversarial attacks have been studied for modulation classification of wireless signals in [10], where fast gradient method (FGM) [11] is used to generate adversarial attacks. Specifically, targeted FGM attack has been used by enforcing the DNNs to misclassify the input signals to a target label. Here, the target is decided by searching over all possible target labels and selecting the one with the least perturbation required to enforce misclassification. It has been shown that the modulation classifier used in [12] incurs major errors due to adversarial attacks in the AWGN channel.

Similar evasion attacks and corresponding defense mechanisms have been studied in [13]–[18]. Previous work has considered the AWGN channel from the transmitter to the receiver only, but has not considered channel effects (path loss or fading) from the adversary to the receiver. However, even a small channel effect would significantly reduce the impact of adversarial attacks by reducing the received perturbation power just below the necessary level such that the adversarial attack fails in changing classification decision over the air.

In this paper, we consider a wireless communication system where a DNN is used to classify wireless signals to modulation types as in [10], and show how to make this classifier vulnerable to adversarial attacks even in the presence of *realistic channel effects* from the adversary to the receiver. For that purpose, we design adversarial attacks with a power constraint that decreases the accuracy of detecting modulation type at the receiver. We first propose two white-box attacks, a *targeted attack* with minimum power and a *non-targeted attack*, subject to channel effects known by the adversary. We show that the adversarial attack fails if the channel between the adversary and the receiver is not considered (as in [10], [13]–[17]) when designing the adversarial attack. Then we show how to design the adversarial attack by accounting for known channel effects.

Next, we relax the assumption that the adversary knows the exact channel condition and present a white-box adversarial attack with *limited channel information* available at the adversary. Finally, we design a *black-box universal adversarial perturbation (UAP) attack*, where the adversary has limited information about the transmitter input, channel conditions, and the classifier architecture. All these attacks demonstrate the importance of channel effects on attack performance and raise the need to utilize channel information in designing adversarial attacks and launching them over the air.

The rest of the paper is organized as follows. Section II explains the system model. Sections III and IV describe targeted and non-targeted white-box adversarial attacks, respectively, using channel information. Section V considers the white-box adversarial attack with limited channel information. Section VI describes a universal adversarial perturbation. Section VII presents simulation results. Section VIII concludes the paper.

II. System Model

We consider a wireless communication system that consists of a transmitter, a receiver, and an adversary. The transmitter transmits signals with one of the modulation types. The receiver applies a pre-trained DL-based classifier on the received signals to classify the modulation type that is used at the transmitter. The adversary launches an attack by transmitting over the air to cause misclassification at the receiver.

The DNN classifier at the receiver is denoted by $f(\cdot;\boldsymbol{\theta}) : \mathcal{X} \rightarrow \mathbb{R}^C$, where $\boldsymbol{\theta}$ is the parameters of the DNN and C is the number of modulation types. Note $\mathcal{X} \subset \mathbb{C}^p$, where p is the dimension of the complex-valued (in-phase/quadrature) inputs that can be also represented by concatenation of two real-valued inputs. The classifier f assigns a modulation type $\hat{l}(\boldsymbol{x},\boldsymbol{\theta}) = \arg\max_k f_k(\boldsymbol{x},\boldsymbol{\theta})$ to every input $\boldsymbol{x} \in \mathcal{X}$. In this formulation, $f_k(\boldsymbol{x},\boldsymbol{\theta})$ is the output of classifier f corresponding to the kth modulation type.

There exist the channel $\boldsymbol{h}_{tr}$ from the transmitter to the receiver and the channel $\boldsymbol{h}_{ar}$ from the adversary to the receiver, where $\boldsymbol{h}_{tr} = [h_{tr,1}, h_{tr,2}, \cdots, h_{tr,p}]^T \in \mathbb{C}^{p\times 1}$ and $\boldsymbol{h}_{ar} = [h_{ar,1}, h_{ar,2}, \cdots, h_{ar,p}]^T \in \mathbb{C}^{p\times 1}$. If the transmitter transmits $\boldsymbol{x}$, the receiver receives $\boldsymbol{r}_t = \mathbf{H}_{tr}\boldsymbol{x} + \boldsymbol{n}$, if there is no adversarial attack, or receives $\boldsymbol{r}_a = \mathbf{H}_{tr}\boldsymbol{x} + \mathbf{H}_{ar}\boldsymbol{\delta} + \boldsymbol{n}$, if the transmitter launches an adversarial attack by transmitting the perturbation signal $\boldsymbol{\delta}$, where $\mathbf{H}_{tr} = \text{diag}\{h_{tr,1}, \cdots, h_{tr,p}\} \in \mathbb{C}^{p\times p}$, $\mathbf{H}_{ar} = \text{diag}\{h_{ar,1}, \cdots, h_{ar,p}\} \in \mathbb{C}^{p\times p}$, $\boldsymbol{\delta} \in \mathbb{C}^{p\times 1}$ and $\boldsymbol{n} \in \mathbb{C}^{p\times 1}$ is complex Gaussian noise. For a stealth (i.e., difficult to detect) attack, the adversarial perturbation $\boldsymbol{\delta}$ is restricted as $||\boldsymbol{\delta}||_2^2 \leq P_{max}$ for some suitable power P_{max}. The adversary obtains the adversarial perturbation $\boldsymbol{\delta}$ for the input $\boldsymbol{x}$ and classifier f by solving the following optimization problem:

$$\begin{aligned} \max_{\delta} \quad & \mathbb{I}\{\hat{l}(\boldsymbol{r}_t,\boldsymbol{\theta}) \neq \hat{l}(\boldsymbol{r}_a,\boldsymbol{\theta})\} \\ \text{subject to} \quad & ||\boldsymbol{\delta}||_2^2 \leq P_{max}, \end{aligned} \tag{1}$$

where $\mathbb{I}\{\cdot\}$ is an indicator function.

In practice solving (1) is difficult. Thus, different methods have been proposed (primarily in the computer vision domain) to approximate the adversarial perturbation such as FGM. FGM is a computationally efficient method for crafting adversarial attacks by linearizing the loss function of the DNN classifier. Let $L(\boldsymbol{\theta},\boldsymbol{x},\boldsymbol{y})$ denote the loss function of the model, where $\boldsymbol{y} \in \{0,1\}^C$ is the label vector. Then FGM linearizes the loss function in a neighborhood of $\boldsymbol{x}$ and uses this linearized function for optimization.

There are two types of attacks called *targeted attacks* and *non-targeted attacks* that involve different objective functions to optimize. In a targeted attack, the adversary is trying to generate a perturbation that causes the classifier at the receiver to have a specific misclassification, e.g., the classifier classifies QPSK modulation as QAM16, whereas in non-targeted FGM attack, the adversary is searching for a perturbation that causes any misclassification (independent of target label). We will further explain these two types of attacks in the next section.

Our goal in this paper is to design an attack to fool the classifier at the receiver while considering the channel effects and satisfying the power constraint at the adversary. For the white-box adversarial attacks, we assume that the adversary knows the architecture ($\boldsymbol{\theta}$ and $L(\cdot)$) of the classifier at the receiver. Also, we assume that the adversary knows the input at the receiver and consequently the channel $\boldsymbol{h}_{ar}$ between the adversary and the receiver. We will relax these assumptions in the later part of the paper.

III. Targeted White-Box Adversarial Attacks using Channel Information

For the targeted attack, the adversary minimizes $L(\boldsymbol{\theta},\boldsymbol{r}_a,\boldsymbol{y}^{target})$ with respect to $\boldsymbol{\delta}$, where $\boldsymbol{y}^{target}$ is one-hot encoded desired target class. FGM is used to linearize the loss function as $L(\boldsymbol{\theta},\boldsymbol{r}_a,\boldsymbol{y}^{target}) \approx L(\boldsymbol{\theta},\boldsymbol{r}_t,\boldsymbol{y}^{target}) + (\mathbf{H}_{ar}\boldsymbol{\delta})^T\nabla_{\boldsymbol{x}}L(\boldsymbol{\theta},\boldsymbol{r}_t,\boldsymbol{y}^{target})$ that is minimized by setting $\mathbf{H}_{ar}\boldsymbol{\delta} = -\alpha\nabla_{\boldsymbol{x}}L(\boldsymbol{\theta},\boldsymbol{r}_t,\boldsymbol{y}^{target})$, where α is a scaling factor to constrain the adversarial perturbation power to P_{max}.

The adversary can generate different targeted attacks with respect to different $\boldsymbol{y}^{target}$ that causes the classifier at the receiver to misclassify the received signals to $C-1$ different modulation types. Thus, as in [10], the adversary can create targeted attacks for all $C-1$ modulation types and chooses the target modulation that uses the least power. However, [10]

only considered the AWGN channel, i.e., $\mathbf{H}_{ar} = \mathbf{I}$, which falls short from representing channel conditions encountered in real wireless communication systems. We call the targeted attack perturbation in [10] as $\boldsymbol{\delta}^{NoCh}$, which is an optimal targeted attack under the AWGN channel. The detailed algorithm is given in Algorithm 1 by setting $\mathbf{H}_{ar} = \mathbf{I}$. In the following subsections, we propose three targeted adversarial attacks to overcome the random effects of the channel.

A. Channel Inversion Attack

We first begin with a naive attack, where the adversary designs its attack by inverting the channel in the optimal targeted attack $\boldsymbol{\delta}^{NoCh}$, which is obtained using Algorithm 1 with the AWGN channel. Since the adversarial attack goes through channel $\boldsymbol{h}_{ar}$, the ith element of the perturbation $\boldsymbol{\delta}$ is simply designed as $\delta_i = \frac{\delta_i^{NoCh}}{h_{ar,i}}$ so that after going through the channel it has the same direction as δ_i^{NoCh} for $i = 1, \cdots, p$. Furthermore, in order to satisfy the power constraint P_{max}, we introduce a scaling factor α so that $\boldsymbol{\delta}^{div} = -\alpha\boldsymbol{\delta}$, where $\alpha = \frac{\sqrt{P_{max}}}{||\boldsymbol{\delta}||_2}$ to satisfy the power constraint at the adversary. Thus, the attack received at the receiver is $\mathbf{H}_{ar}\boldsymbol{\delta}^{div} = -\alpha\boldsymbol{\delta}^{NoCh}$.

B. Minimum Mean Squared Error (MMSE) Attack

In the MMSE attack, the adversary designs the perturbation $\boldsymbol{\delta}^{MMSE}$ so that the distance between the attack after going through the channel and the optimal targeted attack over AWGN channel is minimized. By designing the attack in this way, the received attack at the receiver is close to the optimal targeted attack as much as possible while satisfying the power constraint at the adversary. However, since the classifier is sensitive to not only the direction but also the power of perturbation, the squared error criterion might penalize the candidates of $\boldsymbol{\delta}^{MMSE}$, which have more power with the direction of $\boldsymbol{\delta}^{NoCh}$, i.e., $\boldsymbol{\delta}^{MMSE} = \gamma\boldsymbol{\delta}^{NoCh}$. Therefore, we formulate the optimization problem to select the perturbation $\boldsymbol{\delta}^{MMSE}$ as

$$\begin{aligned} \min_{\boldsymbol{\delta}^{MMSE}} \quad & ||\mathbf{H}_{ar}\boldsymbol{\delta}^{MMSE} - \gamma\boldsymbol{\delta}^{NoCh}||_2^2 \\ \text{subject to} \quad & ||\boldsymbol{\delta}^{MMSE}||_2^2 \leq P_{max}, \end{aligned} \tag{2}$$

where γ is optimized by line search. We can write (2) as

$$\begin{aligned} \min_{\delta_i^{MMSE}} \quad & \sum_{i=1}^{p} ||h_{ar,i}\delta_i^{MMSE} - \gamma\delta_i^{NoCh}||_2^2 \\ \text{subject to} \quad & \sum_{i=1}^{p} ||\delta_i^{MMSE}||_2^2 \leq P_{max}. \end{aligned} \tag{3}$$

We solve the convex optimization problem (3) by using Lagrangian method. The Lagrangian for (3) is given by

$$\mathcal{L} = \sum_{i=1}^{p} ||h_{ar,i}\delta_i^{MMSE} - \gamma\delta_i^{NoCh}||_2^2 + \lambda\left(\sum_{i=1}^{p} ||\delta_i^{MMSE}||_2^2 - P_{max}\right), \tag{4}$$

where $\lambda \geq 0$. The KKT conditions are given by

$$h_{ar,i}^*(h_{ar,i}\delta_i^{MMSE} - \gamma\delta_i^{NoCh}) + \lambda\delta_i^{MMSE} = 0, \tag{5}$$

Algorithm 1 MRPP attack

Inputs: input $\boldsymbol{r}_t$, desired accuracy ε_{acc}, power constraint P_{max} and model of the classifier
Initialize: $\boldsymbol{\varepsilon} \leftarrow \mathbf{0}^{C\times 1}$
for class-index c in range(C) **do**
  $\varepsilon_{max} \leftarrow P_{max}, \varepsilon_{min} \leftarrow 0$
  $\boldsymbol{\delta}_{norm}^c = \frac{\mathbf{H}_{ar}^*\nabla_{\boldsymbol{x}}L(\boldsymbol{\theta},\boldsymbol{r}_t,\boldsymbol{y}^c)}{(||\mathbf{H}_{ar}^*\nabla_{\boldsymbol{x}}L(\boldsymbol{\theta},\boldsymbol{r}_t,\boldsymbol{y}^c)||_2)}$
  while $\varepsilon_{max} - \varepsilon_{min} > \varepsilon_{acc}$ **do**
    $\varepsilon_{avg} \leftarrow (\varepsilon_{max} + \varepsilon_{min})/2$
    $\boldsymbol{x}_{adv} \leftarrow \boldsymbol{x} - \varepsilon_{avg}\mathbf{H}_{ar}\boldsymbol{\delta}_{norm}^c$
    if $\hat{l}(\boldsymbol{x}_{adv}) == l_{true}$ **then**
      $\varepsilon_{min} \leftarrow \varepsilon_{avg}$
    else
      $\varepsilon_{max} \leftarrow \varepsilon_{avg}$
    end if
  end while
  $\varepsilon[c] = \varepsilon_{max}$
end for
$target = \arg\min \boldsymbol{\varepsilon},\ \boldsymbol{\delta}^{MRPP} = -\sqrt{P_{max}}\boldsymbol{\delta}_{norm}^{target}$

for all $i = 1, \cdots, p$. From KKT conditions, we obtain the perturbation of the MMSE attack as

$$\delta_i^{MMSE} = -\frac{\gamma h_{ar,i}^*\delta_i^{NoCh}}{h_{ar,i}^*h_{ar,i} + \lambda}, \tag{6}$$

for all $i = 1, \cdots, p$, where λ is determined by the power constraint at the adversary. Note that the received perturbation at the receiver is $\mathbf{H}_{ar}\boldsymbol{\delta}^{MMSE} = -\boldsymbol{\alpha}^T\boldsymbol{\delta}^{NoCh}$ where $\boldsymbol{\alpha} \in \mathbb{R}^{p\times 1}$ and each element of $\boldsymbol{\alpha}$ is $\alpha_i = \frac{\gamma h_{ar,i}h_{ar,i}^*}{h_{ar,i}^*h_{ar,i}+\lambda}$.

C. Maximum Received Perturbation Power (MRPP) Attack

In the MRPP attack, the adversary selects the perturbation $\boldsymbol{\delta}^{MRPP}$ to maximize the received perturbation power at the receiver and analyzes how the received perturbation power affects the decision process of the classifier. To maximize the received perturbation power and effectively fool the classifier into making a specific classification error, the adversary has to fully utilize the channel between the adversary and the receiver. Thus, if the targeted attack δ_i^{target} is multiplied by the conjugate of the channel, $h_{ar,i}^*$, then the received perturbation after going through the channel becomes $||h_{ar,i}||_2^2\delta_i^{target}$. In this attack, not only the direction is unaffected after going through the channel but also the power is maximized by utilizing the channel. Finally, the adversary generates targeted attack for every possible modulation type to decide the target class and calculate the scaling factor to satisfy the power constraint at the adversary. The details are presented in Algorithm 1.

IV. Non-Targeted White-Box Adversarial Attacks using Channel Information

In this section, the adversary designs the attack based on the non-targeted attack and its objective is to maximize the

Algorithm 2 Crafting naive non-targeted attack

Inputs: number of epochs E, power constraint P_{max}, true label $\boldsymbol{y}^{true}$ and model of the classifier
Initialize: Sum of gradient $\Delta \leftarrow 0$, $\boldsymbol{x} \leftarrow \boldsymbol{r}_t$
for epoch e in range(E) **do**
  $\boldsymbol{\delta}_{norm} = \frac{\nabla_{\boldsymbol{x}} L(\boldsymbol{\theta},\boldsymbol{x},\boldsymbol{y}^{true})}{(||\nabla_{\boldsymbol{x}} L(\boldsymbol{\theta},\boldsymbol{x},\boldsymbol{y}^{true})||_2)}$
  $\boldsymbol{x} \leftarrow \boldsymbol{x} + \sqrt{\frac{P_{max}}{E}} \mathbf{H}_{ar} \boldsymbol{\delta}_{norm}$
  $\Delta \leftarrow \Delta + \sqrt{\frac{P_{max}}{E}} \boldsymbol{\delta}_{norm}$
end for
$\boldsymbol{\delta}^{naive} = \sqrt{P_{max}} \frac{\boldsymbol{\Delta}}{||\boldsymbol{\Delta}||_2}$

loss function $L(\boldsymbol{\theta}, \boldsymbol{r}_a, \boldsymbol{y}^{true})$, where $\boldsymbol{y}^{true}$ is the true label of $\boldsymbol{x}$. FGM is used to linearize the loss function as $L(\boldsymbol{\theta}, \boldsymbol{r}_a, \boldsymbol{y}^{true}) \approx L(\boldsymbol{\theta}, \boldsymbol{r}_t, \boldsymbol{y}^{true}) + (\mathbf{H}_{ar}\boldsymbol{\delta})^T \nabla_{\boldsymbol{x}} L(\boldsymbol{\theta}, \boldsymbol{r}_t, \boldsymbol{y}^{true})$ that is maximized by setting $\mathbf{H}_{ar}\boldsymbol{\delta} = \alpha \nabla_{\boldsymbol{x}} L(\boldsymbol{\theta}, \boldsymbol{r}_t, \boldsymbol{y}^{true})$, where α is a scaling factor to constrain the adversarial perturbation power to P_{max}. Based on FGM for the non-targeted attack, we propose non-targeted adversarial attacks to effectively attack the classifier at the receiver.

A. Naive Non-Targeted Attack

As in the targeted attacks, we begin with the naive non-targeted attack. First, the adversary divides its power P_{max} into E epochs and uses $\frac{P_{max}}{E}$ amount of power to the gradient of loss function to tilt the transmitted signal from the transmitter. Next, the adversary calculates the gradient again with respect to the transmitted signal from the transmitter and added perturbation. Then the adversary adds another perturbation with power $\frac{P_{max}}{E}$ using the new gradient. This scheme generates the best direction to increase the loss function at that specific instance. Finally, the adversary repeats this procedure E times and sums all the gradients of the loss function that were added to the transmitted signal from the transmitter since the adversary can send only one perturbation at a time over the air. Finally, a scaling factor is introduced to satisfy the power constraint at the adversary. The details of this algorithm are presented in Algorithm 2.

B. Minimum Mean Squared Error (MMSE) Attack

The non-targeted MMSE attack is designed similar to the targeted MMSE attack. The adversary first obtains $\boldsymbol{\delta}^{NoCh}$ from the naive non-targeted attack with $\mathbf{H}_{ar} = \mathbf{I}$ and uses it to solve problem (2). Thus, the solution is the same as the solution to (2) except that it has the opposite direction to maximize the loss function, whereas the loss function is minimized for the targeted attack case. Therefore, the perturbation selected by the MMSE scheme for non-targeted attack is $\boldsymbol{\delta}^{MMSE} = \boldsymbol{\alpha}^T \boldsymbol{\delta}^{NoCh}$, where $\boldsymbol{\alpha} \in \mathbb{R}^p$ and each element of $\boldsymbol{\alpha}$ is $\alpha_i = \frac{\gamma h^*_{ar,i}}{h^*_{ar,i} h_{ar,i} + \lambda}$.

C. Maximum Received Perturbation Power (MRPP) Attack

As we have seen in the targeted MRPP attack, the attack should be in the form of $\boldsymbol{\delta}^{MRPP} = \mathbf{H}^*_{ar} \boldsymbol{\delta}^{target}$ to maximize the received perturbation power at the receiver. Thus, the naive non-targeted attack is changed to create the MRPP non-targeted attack by changing $\boldsymbol{\delta}_{norm}$ in Algorithm 2 to $\frac{\mathbf{H}^*_{ar} \nabla_{\boldsymbol{x}} L(\boldsymbol{\theta},\boldsymbol{x},\boldsymbol{y}^{true})}{(||\mathbf{H}^*_{ar} \nabla_{\boldsymbol{x}} L(\boldsymbol{\theta},\boldsymbol{x},\boldsymbol{y}^{true})||_2)}$.

Algorithm 3 Crafting adversarial attack with limited channel information

Inputs: N channel realization $\{\mathbf{H}^{(1)}_{ar}, \cdots, \mathbf{H}^{(N)}_{ar}\}$, input $\boldsymbol{r}_t$ and model of the classifier
Initialize: $\Delta \leftarrow 0$
for n in range(N) **do**
  Find $\boldsymbol{\delta}^{(n)}$ from white-box attack algorithm using $\boldsymbol{r}_t$ and $\mathbf{H}^{(n)}_{ar}$
  Stack $\boldsymbol{\delta}^{(n)}$ to Δ
end for
Compute the first principle direction $\boldsymbol{v}_1$ of Δ using PCA
$\Delta = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T$ and $\boldsymbol{v}_1 = \mathbf{V}\boldsymbol{e}_1$
$\boldsymbol{\delta}^{limited} = \sqrt{P_{max}} \boldsymbol{v}_1$

V. White-box Adversarial Attack with Limited Channel Information

The adversarial attacks that are designed in the previous sections use the exact channel information. However, this may not always be the case in practical scenarios. Therefore, in this section, we propose an algorithm to generate adversarial attacks using principal component analysis (PCA) with limited channel information, i.e., distribution of the channel. PCA was also used in [10] for the AWGN channel case only. PCA is performed by eigenvalue decomposition of the data covariance matrix or singular value decomposition of a data matrix and is used to obtain the principal component which has the largest variance. In other words, PCA finds the principal component that provides the most information about the data with reduced dimension by projecting the data onto it.

To generate an adversarial attack with limited channel information, we first generate N realizations of the channel between the adversary and the receiver $\{\mathbf{H}^{(1)}_{ar}, \mathbf{H}^{(2)}_{ar}, \cdots, \mathbf{H}^{(N)}_{ar}\}$ from a known distribution. Then we generate N adversarial attacks using white-box attack algorithms from the previous sections, either targeted or non-targeted, using N realizations of the channel and the known input at the classifier $\boldsymbol{r}_t$. Finally, we stack N generated adversarial attacks in a matrix and find the principal component of the matrix to use it as the adversarial attack with limited channel information. The details are presented in Algorithm 3.

VI. Universal Adversarial Attack

In the previous sections, the adversary designs a white-box attack with the assumptions that it knows the architecture of the classifier at the receiver, the channel between the adversary and the receiver, and the exact input at the receiver. However, these assumptions are not always practical in real wireless communications systems. Thus, in this section, we relax these assumptions and propose UAP attacks.

A. Universal Adversarial Attack with Pre-Collected Input at the Receiver

Here, we first relax the assumption that the adversary knows the exact input of the classifier. The adversary in the previous attacks generates an input-dependent perturbation, i.e., $\boldsymbol{\delta}$ is designed given the exact input $\boldsymbol{r}_t$. This requires the adversary to always know the input of the classifier, which is not a practical assumption to make due to synchronization issues. Thus, it is more practical to design an input-independent UAP.

We propose an algorithm to design the UAP using PCA. We assume that the adversary collects some arbitrary set of inputs $\{\boldsymbol{r}_t^{(1)}, \boldsymbol{r}_t^{(2)}, \cdots, \boldsymbol{r}_t^{(N)}\}$ and associated labels. The adversary generates perturbations $\{\boldsymbol{\delta}^{(1)}, \boldsymbol{\delta}^{(2)}, \cdots, \boldsymbol{\delta}^{(N)}\}$ with respect to the obtained arbitrary set of inputs and the exact channel information using schemes from the previous sections. To reflect the common characteristics of $\{\boldsymbol{\delta}^{(1)}, \boldsymbol{\delta}^{(2)}, \cdots, \boldsymbol{\delta}^{(N)}\}$ in the UAP, we stack these perturbations into a matrix and perform PCA to find the first component of the matrix with the largest eigenvalue. Hence, we use the direction of the first principal component as the direction of UAP for channel $\mathbf{H}_{ar}$. The algorithm for the UAP with N pre-collected input data is similar to Algorithm 3. The difference is that there are N pre-collected data inputs instead of N realizations of the channel.

B. Universal Adversarial Attack with Limited Channel Information

Now, we further relax the assumption that the adversary knows the exact channel between the adversary and the receiver, and assume that the adversary only knows the distribution of this channel. To design the UAP knowing the distribution of the channel, we first generate random realizations of the channel $\{\mathbf{H}_{ar}^{(1)}, \mathbf{H}_{ar}^{(2)}, \cdots, \mathbf{H}_{ar}^{(N)}\}$ from the distribution. Then we generate $\boldsymbol{\delta}^{(n)}$ using $\boldsymbol{r}_t^{(n)}$ and $\mathbf{H}_{ar}^{(n)}$ instead of using the real channel $\mathbf{H}_{ar}$ and real input $\boldsymbol{r}_t$. Again, we use PCA to find the first component of the matrix and use it as our direction of UAP. The algorithm for UAP with limited channel information is analogous to Algorithm 3 except that we have pre-collected input data as opposed to real input data in Algorithm 3.

C. Black-box Universal Adversarial Attack

The last assumption that we will relax is the information about the classifier at the receiver. To relax this assumption, we use the well-known transferability property of adversarial attacks [19]. This property states that the adversarial attack crafted to fool a specific DNN can also fool other DNNs with different architectures, with high probability. Therefore, the adversary generates UAPs using a substitute DNN and uses them to fool the actual DNN at the receiver.

VII. Simulation Results

We compare the performance of attacks proposed in this paper and another attack from [10]. We use VT-CNN2 classifier used in [9] and [10], and train it with GNU radio ML dataset RML2016.10a [20]. The dataset contains 220,000 samples. Each sample corresponds to one specific modulation scheme at a specific signal-to-noise ratio (SNR). There are

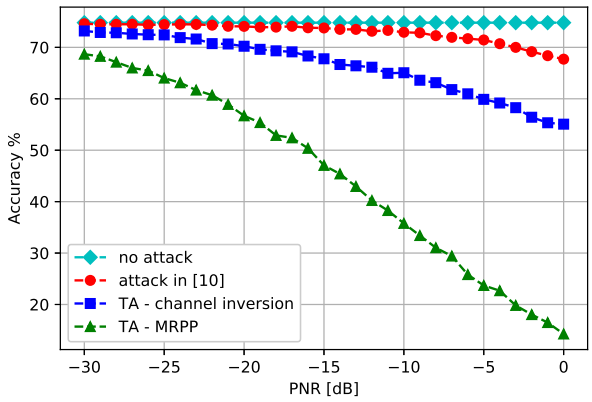

Fig. 1. Classifier accuracy with and without considering wireless channel when SNR = 10 dB.

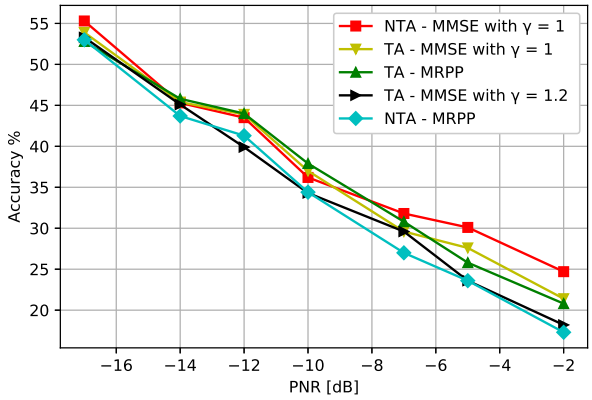

Fig. 2. Classifier accuracy under different white-box attack schemes when SNR = 10 dB.

11 modulations: BPSK, QPSK, 8PSK, QAM16, QAM64, CPFSK, GFSK, PAM4, WBFM, AM-SSB and AM-DSB. Also, we follow the same setup of [9], using Keras with TensorFlow backend, where the modulation classifier at the receiver estimates the modulation after receiving 128 I/Q (in-phase/quadrature) channel symbols. We assume that the channel between the adversary and the receiver is Rayleigh fading with path-loss and shadowing, i.e., $\boldsymbol{h}_{ar} = K(\frac{d_0}{d})^{\gamma}\psi\boldsymbol{h}_{ray}$ where $K = 1, d_0 = 1, d = 10, \gamma = 2.7, \psi \sim \text{Lognormal}(0, 8)$ and $\boldsymbol{h}_{ray} \sim \text{Rayleigh}(0, 1)$.

Here, we use perturbation-to-noise ratio (PNR) metric from [10] that shows the relative perturbation power with respect to the noise and measure how the increase in the PNR affects the accuracy of the classifier. Note that as the PNR increases, it is more likely to be detected by the receiver. In the figures, we denote targeted attack by TA and non-targeted attack NTA.

Fig. 1 presents the accuracy of the classifier versus PNR under the proposed targeted white-box adversarial attacks with exact channel information and the adversarial attack studied in [10]. As expected, the white-box attack in [10] considering only the AWGN channel has poor performance that is close to no attack case in low PNR region. The reason is that the

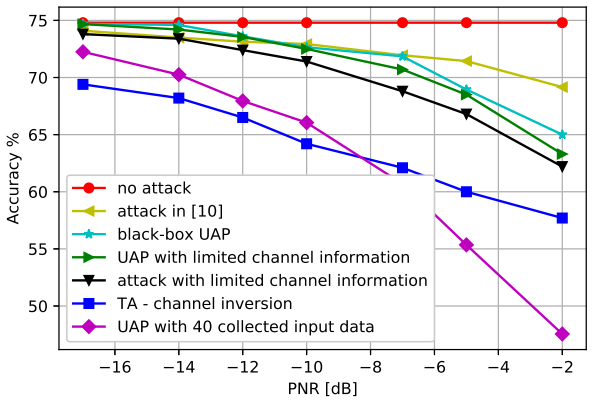

Fig. 3. Classifier accuracy using the UAP with different levels of information availability when SNR = 10 dB.

wireless channel changes the phase and the magnitude of the perturbation at the receiver. Further, we see that the targeted channel inversion attack does not perform well compared to the targeted MRPP attack, indicating the importance of the received power at the classifier.

The performance of the proposed white-box attacks is compared in Fig. 2. As discussed in Section IV.B, λ can be optimized by using a line search method and it can be seen that the targeted MMSE attack performs better with $\lambda = 1.2$ compared to $\lambda = 1$. Furthermore, we observe that the non-targeted MRPP attack outperforms other attacks. This can be explained by the freedom of the direction that the non-targeted adversarial attack can take. For targeted attacks, we can only have 10 different directions since we have 11 modulation types, however, the non-targeted attacks do not have such restriction. Thus, it is more likely that the non-targeted attacks choose a better direction to enforce misclassification. Moreover, the computation complexity for non-targeted attacks is lower compared to the targeted attacks that involve iterations to reach the desired accuracy.

In Fig. 3, we investigate the performance of the adversarial attacks with respect to different levels of information availability. First, we observe that the UAP with 40 pre-collected inputs, where the adversary knows the exact channel information, outperforms other attacks with limited information. This result shows the importance of the channel state information over the exact input data when crafting an adversarial attack. Note that the UAP with 40 pre-collected input data even outperforms the targeted channel inversion attack in the high PNR region, where the adversary knows not only the exact channel but also the exact input at the receiver. Furthermore, similar performance of the UAP with limited channel information and the black-box UAP shows transferability of adversarial attack, where for black-box UAP we use the same structure of the classifier but train it differently.

VIII. Conclusion

We considered the wireless communication system, where DL algorithms are used to classify radio signals and showed that adversarial attacks against a modulation classifier are effective even when there are channel effects beyond the AWGN channel. Specifically, we considered both targeted attack and non-targeted attacks, and observed in the simulation results that DNNs used for modulation classification are vulnerable to these attacks. Furthermore, even with limited information, we show that the UAP can be generated to enforce misclassification at the receiver.

References

[1] I. Goodfellow, Y. Bengio, and A. Courville, "Deep learning." MIT press, 2016.

[2] T. Erpek, T. O'Shea, Y. E. Sagduyu, Y. Shi, and T. C. Clancy, "Deep learning for wireless communications. in development and analysis of deep learning architectures." Springer, 2020, pp. 223–266.

[3] Y. Vorobeychik and M. Kantarcioglu, "Adversarial machine learning," *Synthesis Lectures on Artificial Intelligence and Machine Learning*, vol. 12, no. 3, pp. 1–169, December 2017.

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

[5] T. Erpek, Y. E. Sagduyu, and Y. Shi, "Deep learning for launching and mitigating wireless jamming attacks," *IEEE Transactions on Cognitive Communications and Networking*, vol. 5, no. 1, pp. 2–14, March 2019.

[6] Y. E. Sagduyu, T. Erpek, and Y. Shi, "Adversarial deep learning for over-the-air spectrum poisoning attacks," *IEEE Transactions on Mobile Computing*, no. 1, pp. 2–14, 2019.

[7] K. Davaslioglu and Y. E. Sagduyu, "Trojan attacks on wireless signal classification with adversarial machine learning," in *IEEE Workshop on Data-Driven Dynamic Spectrum Sharing of IEEE International Symposium on Dynamic Spectrum Access Networks (DySPAN)*, 2019.

[8] M. Sadeghi and E. G. Larsson, "Physical adversarial attacks against end-to-end autoencoder communication systems," *IEEE Commun. Lett.*, vol. 23, no. 5, pp. 847–850, May 2019.

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

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

[11] A. Kurakin, I. Goodfellow, and S. Bengio, "Adversarial examples in the physical world," in *ICLR*, 2017.

[12] T. J. O'Shea, J. Corgan, and T. C. Clancy, "Convolutional radio modulation recognition networks," in *Int. Conf. on Engineering Applications of Neural Networks*, 2016.

[13] B. Flowers, R. M. Buehrer, and W. C. Headley, "Evaluating adversarial evasion attacks in the context of wireless communications," 2019, available on arXiv:1903.01563.

[14] S. Bair, M. Delvecchio, B. Flowers, A. J. Michaels, and W. C. Headley, "On the limitations of targeted adversarial evasion attacks against deep learning enabled modulation recognition," in *ACM WiSec Workshop on Wireless Security and Machine Learning (WiseML)*, 2019.

[15] S. Kokalj-Filipovic and R. Miller, "Adversarial examples in RF deep learning: detection of the attack and its physical robustness," 2019, available on arXiv:1902.06044.

[16] S. Kokalj-Filipovic and R. Miller, "Targeted adversarial examples against RF deep classifiers," in *ACM WiSec Workshop on Wireless Security and Machine Learning (WiseML)*, 2019.

[17] S. Kokalj-Filipovic, R. Miller, and G. M. Vanhoy, "Adversarial examples in RF deep learning: Detection and physical robustness," in *GlobalSIP*, 2019.

[18] M. Z. Hameed, A. Gyorgy, and D. Gunduz, "Communication without interception: Defense against deep-learning-based modulation detection," 2019, available on arXiv: 1902.10674.

[19] N. Papernot, P. McDaniel, and I. Goodfellow, "Transferability in machine learning: from phenomena to blackbox attacks using adversarial samples," 2016, available on arXiv: 1605.07277.

[20] T. J. O'Shea and N. West, "Radio machine learning dataset generation with GNU radio," in *Proc. of the 6th GNU Radio Conf.*, 2016.