Generalized Wireless Adversarial Deep Learning

Francesco Restuccia, Salvatore D'Oro, Amani Al-Shawabka, Bruno Costa Rendon, Kaushik Chowdhury, Stratis Ioannidis and Tommaso Melodia
Institute for the Wireless Internet of Things, Northeastern University, Boston, MA, USA

ABSTRACT

Deep learning techniques can classify spectrum phenomena (*e.g.*, waveform modulation) with accuracy levels that were once thought impossible. Although we have recently seen many advances in this field, extensive work in computer vision has demonstrated that an adversary can "crack" a classifier by designing inputs that "steer" the classifier away from the ground truth. This paper advances the state of the art by proposing a generalized analysis and evaluation of adversarial machine learning (AML) attacks to deep learning systems in the wireless domain. We postulate a series of adversarial attacks, and formulate a Generalized Wireless Adversarial Machine Learning Problem (GWAP) where we analyze the combined effect of the wireless channel and the adversarial waveform on the efficacy of the attacks. We extensively evaluate the performance of our attacks on a state-of-the-art 1,000-device radio fingerprinting dataset, and a 24-class modulation dataset. Results show that our algorithms can decrease the classifiers' accuracy up to 3x while keeping the waveform distortion to a minimum.

CCS CONCEPTS

• **Theory of computation → Adversarial learning**; • **Security and privacy** → *Mobile and wireless security*.

KEYWORDS

Adversarial Learning, Security, Wireless networks, Deep Learning

ACM Reference Format:
Francesco Restuccia, Salvatore D'Oro, Amani Al-Shawabka, Bruno Costa Rendon, Kaushik Chowdhury, Stratis Ioannidis and Tommaso Melodia. 2020. Generalized Wireless Adversarial Deep Learning. In *2nd ACM Workshop on Wireless Security and Machine Learning (WiseML '20), July 13, 2020, Linz (Virtual Event), Austria.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3395352.3402625

1 INTRODUCTION

Recent advances in wireless deep learning have now clearly demonstrated its great potential for applications such as for radio fingerprinting [16] and signal/traffic classification [11, 17–19], among many others [7]. However, it has been extensively proven that neural networks are prone to be "hacked" by carefully crafting small-scale perturbations to the input that are ultimately able to "steer" the neural network away from the ground truth. This activity is also known in literature [2, 5, 9, 14] as *adversarial machine learning* (AML). The degree to which malicious wireless agents can find adversarial examples is critically correlated to the applicability of neural networks to problems in the wireless domain.

The above reasons clearly show the timeliness and urgency of a rigorous investigation into the *robustness* of wireless deep learning systems. Prior work (see Section 6) is limited by small-scale simulation-based scenarios, which has left several fundamental questions unanswered. The key reason that sets AML apart in the wireless domain is that waveforms are inevitably affected by the stochastic nature of the channel [4]. This implies that the channel action must necessarily be factored into the crafting process of the AML attack. Therefore, the key contribution of this paper is to provide a comprehensive modeling and experimental evaluation of adversarial machine learning (AML) attacks to state-of-the-art wireless deep learning systems.

We summarize our core technical contributions as follows:

• We propose a novel AML threat model (Section 2) where we consider a "whitebox" scenario (*i.e.*, the adversary has access to the neural network). The primary advance of our model is that our attacks are derived for arbitrary channels, waveforms, and neural networks, and thus generalizable to any state-of-the-art wireless deep learning system;

• Based on the proposed model, we formulate an *AML Waveform Jamming* (Section 3.1) and an *AML Waveform Synthesis* (Section 3.2) attack. Next, we propose a *Generalized Wireless Adversarial Machine Learning Problem* (GWAP) where an adversary tries to steer the neural network away from the ground truth while satisfying constraints such as bit error rate, radiated power, and other relevant metrics below a threshold (Section 4);

• We evaluate the proposed attacks on (i) a deep learning model for radio fingerprinting [16] trained on a 1,000-device dataset of WiFi and ADS-B transmissions collected *in the wild*; and (ii) a model [11] trained on the widely-available RadioML 2018.01A modulation dataset. Our algorithms are shown to decrease the accuracy up to 3x, while keeping the waveform distortion to a minimum.

2 WIRELESS AML MODEL

We use boldface upper and lower-case letters to denote matrices and column vectors, respectively. For a vector $\mathbf{x}$, x_i denotes the i-th element, $\|\mathbf{x}\|_p$ indicates the l_p- norm of $\mathbf{x}$, $\mathbf{x}^\top$ its transpose, and $\mathbf{x} \cdot \mathbf{y}$ the inner product of $\mathbf{x}$ and $\mathbf{y}$. For a matrix $\mathbf{H}$, H_{ij} will indicate the (i,j)-th element of $\mathbf{H}$. The notation $\mathbb{R}$ and $\mathbb{C}$ will indicate the set of real and complex numbers, respectively.

System Model. The top portion of Figure 1 summarizes our system model, where we consider a receiving node R, an attacker node A, and a set $\mathcal{L}$ of N legitimate nodes communicating with R. We assume that R hosts a target neural network (TNN) used to classify waveforms coming from nodes in $\mathcal{L}$.

WiseML '20, July 13, 2020, Linz (Virtual Event), Austria
© 2020 Association for Computing Machinery.
ACM ISBN 978-1-4503-8007-2/20/07…$15.00
https://doi.org/10.1145/3395352.3402625

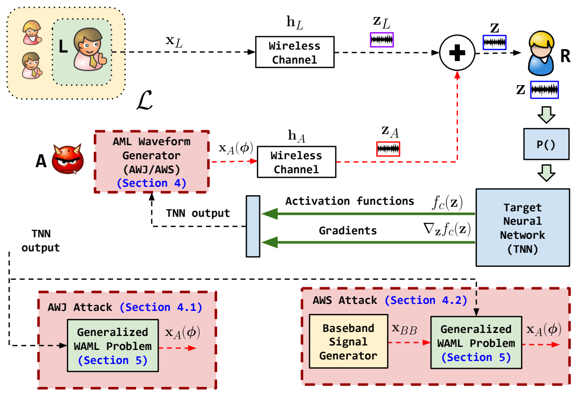

Figure 1: Overview of AML Waveform Jamming (AWJ) and AML Waveform Synthesis (AWS).

Let $\Lambda > 1$ be the number of layers of the TNN, and C be the set of its classes. We model the TNN as a function F that maps the relation between an input $\mathbf{x}$ and an output $\mathbf{y}$ through a Λ-layer mapping $F(\mathbf{x};\theta) : \mathbb{R}^i \rightarrow \mathbb{R}^o$ of an input vector $\mathbf{x} \in \mathbb{R}^i$ to an output vector $\mathbf{y} \in \mathbb{R}^o$. The mapping happens through Λ transformations: $\mathbf{r}_j = F_j(\mathbf{r}_{j-1}, \theta_j)$ $0 \leq j \leq \Lambda$, where $F_j(\mathbf{r}_{j-1}, \theta_j)$ is the mapping carried out by the j-th layer. The vector $\boldsymbol{\theta} = \{\theta_1, \dots, \theta_\Lambda\}$ defines the whole set of parameters of the TNN. We assume the last layer of the TNN is *dense*, meaning that $F_{\Lambda-1}(\mathbf{r}_{j-1}, \theta_j) = \sigma(\mathbf{W}_j \cdot \mathbf{r}_{j-1} + \mathbf{b}_j)$, where σ is a softmax activation function, $\mathbf{W}_j$ is the *weight* matrix and $\mathbf{b}_j$ is the *bias* vector. Notice that the mapping $F(\mathbf{x};\theta) : \mathbb{R}^i \rightarrow \mathbb{R}^o$ can be any derivable function, including recurrent networks. We evaluate the activation probabilities of the neurons at the last layer of the TNN. Let $c \in C$ be a generic class in the classification set of the TNN. We denote $f_c(\mathbf{x})$ as the *activation probability* of the neuron corresponding to class c at the output layer of the TNN when input $\mathbf{x}$ is fed to the TNN. It follows that $f_c(\mathbf{x}) = F_{\Lambda,c}(\mathbf{r}_{\Lambda-1}, \theta_\Lambda)$.

By taking as reference [11], we assume that the input of the TNN is a series of I/Q samples received from the radio interface. We assume that the I/Q samples may be processed through a *processing function* $P()$ before feeding the I/Q samples to the TNN. Common examples of processing functions $P()$ are equalization, demodulation or packet detection.

Threat Model. We assume the adversary A may or may not be part of the legitimate set of nodes in $\mathcal{L}$. We call the adversary respectively *rogue* and *external* in these cases. We consider a *white-box* scenario where the adversary A has knowledge of the TNN activation functions F_j, meaning that A has access to the output layer F_Λ but also to the weight vector $\boldsymbol{\theta}$ (and thus, its gradient as a function of the input). The adversary then may choose different strategies to craft adversarial samples over tuples $(\mathbf{x}, y)$ obtained from querying the TNN. We leave for future work the investigation of blackbox scenarios. By referring to prior work, we consider both *targeted* [23] and *untargeted* [9] attacks.

Wireless Model. To be effective, the attacker must be within the transmission range of R, meaning that A should be sufficiently close to R to emit waveforms that compromise (to some extent) ongoing transmissions between any node $l \in \mathcal{L}$ and R. This scenario is particularly compelling, since not only can A eavesdrop wireless transmissions generated by R (*e.g.*, feedback information such as ACKs or REQs), but also emit waveforms that can be received by R – and thus, compromise the TNN.

We illustrate the effect of channel action in Figure 1, which can be expressed through well-established models for wireless networks. Specifically, the waveform transmitted by any legitimate node $L \in \mathcal{L}$ and received by R can be modeled as

$$\mathbf{z_L} = \mathbf{x_L} \circledast \mathbf{h_L} + \mathbf{w_L}, \tag{1}$$

where $\mathbf{x_L}$ represents the waveform transmitted by node L, $\circledast$ is the convolution operator; $\mathbf{h_L}$ and $\mathbf{w_L}$ are the fading and noise characterizing the channel between node L and the receiver R.

Similarly, let $\mathbf{x_A}$ be the waveform transmitted by node A, and let $\boldsymbol{\phi}$ be an attack strategy of A. The attacker utilizes $\boldsymbol{\phi}$ to transform the waveform $\mathbf{x_A}$ and its I/Q samples. For this reason, the waveform transmitted by A can be written as $\mathbf{x_A}(\boldsymbol{\phi})$. For the sake of generality, in this section we do not make any assumption on $\boldsymbol{\phi}$. However, in Section 3 we present two examples of practical relevance (*i.e.*, jamming and waveform synthesis) where closed-form expressions for the attack strategy $\boldsymbol{\phi}$ and $\mathbf{x_A}(\boldsymbol{\phi})$ are derived. The waveform $\mathbf{z_A}$ can be written as

$$\mathbf{z_A} = \mathbf{x_A}(\boldsymbol{\phi}) \circledast \mathbf{h_A} + \mathbf{w_A}. \tag{2}$$

Without loss of generality, we model $\mathbf{h}_i$ as a Finite Impulse Response (FIR) filter with $K > 0$ complex-valued taps. Notice that (1) and (2) do not assume any particular channel model, nor any particular attack strategy. Therefore, our formulation is very general in nature and able to model a rich set of real-world wireless scenarios.

3 WIRELESS AML ATTACKS

With the help of Figure 1, we now introduce the *AML Waveform Jamming* (Section 3.1), and *AML Waveform Synthesis* (Section 3.2).

3.1 AML Waveform Jamming (AWJ)

In AWJ, an adversary jams the waveform of a legitimate device to confuse the TNN. Since the TNN takes as input I/Q samples, the adversary may craft a jamming waveform that, at the receiver side, causes a slight displacement of I/Q samples transmitted by the legitimate device, thus pushing the TNN towards a misclassification.

As shown in Figure 1, the waveform $\mathbf{x_A}$ generated by the attacker node A is aimed at jamming already ongoing transmissions between a legitimate node L and the receiver R. In this case, the signal received by R can be written as $\mathbf{z} = \mathbf{z_A} + \mathbf{z_L}$, where $\mathbf{z_A}$ and $\mathbf{z_L}$ are defined in (1) and (2), respectively.

Attack objectives ad strategies. The attacker aims at computing $\mathbf{x_A}$ so that $C(\mathbf{z}) \neq C(\mathbf{z}_L)$. Moreover, this attack can be either *targeted* (*i.e.*, A generates jamming waveforms whose superimposition with legitimate signals produce $C(\mathbf{z}) = c_T$, with c_T being a specific target class in C), or *untargeted* (*i.e.*, it is sufficient to obtain $C(\mathbf{z}) \neq c_L$). In the jamming case, $\mathbf{x}_A(\boldsymbol{\phi}) = \boldsymbol{\phi}$. That is, the transmitted waveform corresponds to the actual attack (jamming) strategy. Specifically, we have

$$\mathbf{x}_A(\boldsymbol{\phi}) = (\phi_n^{\Re} + j\phi_n^{\Im})_{n=1,\dots,N_J}, \tag{3}$$

where (i) $a^{\Im} = \text{Im}(a)$ and $a^{\Re} = \text{Re}(a)$ for any complex number a; and (ii) $N_J > 1$ represents the length of the jamming signal in terms of I/Q samples. Since N_J might be smaller than the TNN input N_I, we assume that the adversary periodically transmits the sequence of N_J I/Q samples so that they completely overlap with legitimate waveforms and have the same length. However, it is worth to notice that we do not assume perfect superimposition of the jamming signal with the legitimate signal, and thus, adversarial signals are not added in a precise way to the legitimate waveform.

Addressing non-stationarity. An adversary cannot evaluate the channel $\mathbf{h_L}$ in (1) – which is node-specific and time-varying. Also, waveforms transmitted by legitimate nodes vary according to the encoded information, which is usually a non-stationary process. It follows that jamming waveforms that work well for a given legitimate waveform $\mathbf{z_L}$, might not be equally effective for any other $\mathbf{z'_L} \neq \mathbf{z_L}$. Thus, rather than computing the optimal jamming waveform for each $\mathbf{z_L}$, we compute it over a set of consecutive S legitimate input waveforms, also called *slices*.

Let $\rho \in \{0, 1\}$ indicate whether or not the attacker node belongs to the legitimate node set $\mathcal{L}$ (*i.e.*, a rogue node). Specifically, $\rho = 1$ if the attacker node is a rogue device and $A \in \mathcal{L}$, $\rho = 0$ if the attacker is external (*i.e.*, $A \notin \mathcal{L}$). Also, let c_L and c_A be the correct classes of the waveforms transmitted by nodes L and A, respectively.

Untargeted AWJ. The adversary aims at jamming legitimate waveforms such that (i) these are misclassified by the TNN; (ii) malicious activities are not detected by the TNN; and (iii) attacks satisfy hardware limitations (*e.g.*, energy should be limited). These objectives and constraints can be formulated through the following *untargeted* AWJ problem (AWJ-U):

$$\underset{\boldsymbol{\phi}}{\text{minimize}} \quad \frac{1}{S}\sum_{s=1}^{S}\left[f_{c_L}(\mathbf{z}_s) + \rho \cdot f_{c_A}(\mathbf{z}_s)\right] \qquad \text{(AWJ-U)}$$

$$\text{subject to } \mathrm{BER}_L(\mathbf{z}_s) \leq \mathrm{BER}_{\max}, \quad s = 1, 2, \ldots, S \qquad \text{(C1)}$$

$$\|\mathbf{x_A}(\boldsymbol{\phi})\|_2^2 \leq \mathrm{E}_{\max}, \quad s = 1, 2, \ldots, S \qquad \text{(C2)}$$

where $\mathbf{z}_s = \mathbf{z_A} + \mathbf{z_{L_s}}$, $\mathbf{z_{L_s}}$ represents the s-th slice (or input) of the TNN; Constraint (C1) ensures that the BER experienced by the legitimate node is lower than the maximum tolerable BER threshold $\mathrm{BER}_{\max}$; while (C2) guarantees that the energy of the jamming waveform does not exceed a maximum threshold $\mathrm{E}_{\max}$. In practice, Constraints (C1) and (C2) ensure that jamming waveforms do not excessively alter the position of legitimate I/Q samples. This is crucial to avoid anti-jamming strategies such as modulation and frequency hopping, among others. Although Problem (AWJ-U) takes into account Constraints (C1) and (C2) only, in Section 4 we extend the formulation to larger set of constraints. The term $\rho f_{c_A}(\mathbf{z}_s)$ in Problem (AWJ-U) is crucial to generate jamming waveforms that hide features of the rogue node known by the TNN.

Targeted AWJ. By defining $c_T \in C$ as the target class, we formulate the *targeted* AWJ as

$$\underset{\boldsymbol{\phi}}{\text{maximize}} \quad \frac{1}{S}\sum_{s=1}^{S}\left[f_{c_T}(\mathbf{z}_s) - \left(f_{c_L}(\mathbf{z}_s) + \rho \cdot f_{c_A}(\mathbf{z}_s)\right)\right] \qquad \text{(AWJ-T)}$$

$$\text{subject to Constraints (C1), (C2)}$$

When compared to Problem (AWJ-U), Problem (AWJ-T) differs in terms of the objective function. One naive approach would see the adversary maximize the term $\frac{1}{S}\sum_{s=1}^{S} f_{c_T}(\mathbf{z}_s)$ only. However, the objective of the adversary is to produce misclassifications, so the adversary should try to reduce the activation probability of the jammed class c_L and adversarial class c_A, while maximizing the activation probability for the target class c_T. It is expected that the TNN has high accuracy and by simply maximizing $\frac{1}{S}\sum_{s=1}^{S} f_{c_T}(\mathbf{z}_s)$ does not necessarily mean that the TNN would not be able to still correctly classify transmissions from the legitimate device L (i.e., the activation probability f_{c_L} might still be high). *In other words, to effectively fool the TNN, the attacker must generate waveforms that (i) suppress features of class c_L; (ii) mimic those of class c_T; and (iii) hide features of the attacker's class c_A.* These objectives can be formulated via the objective function in Problem (AWJ-T).

3.2 AML Waveform Synthesis (AWS)

In this attack – illustrated in the bottom-right side of Figure 1 – an adversary A transmits synthetic waveforms trying to imitate features belonging to a target class c_T. In contrast to the AWJ, in this case $\mathbf{z} = \mathbf{z_A}$ and synthetic waveforms $\mathbf{x_A}(\boldsymbol{\phi})$ are generated so that $C(\mathbf{z}) = c_T$ and the waveform received by node R is still intelligible. By definition, this attack is **targeted** only.

Let $c_T \in C$ be the target class. The (targeted) AWS problem (AWS) is formulated as

$$\underset{\boldsymbol{\phi}}{\text{maximize}} \quad \frac{1}{S}\sum_{s=1}^{S}\left[f_{c_T}(\mathbf{z}_{A_s}) - \rho f_{c_A}(\mathbf{z}_{A_s})\right] \qquad \text{(AWS)}$$

$$\text{subject to Constraints (C1), (C2)}$$

This attack maps well to scenarios such as radio fingerprinting, where a malicious device aims at generating a waveform embedding impairments that are unique to the target legitimate device [16]. In other words, the attacker *cannot generate random waveforms as in the AWJ*, but should transmit waveforms that contain decodable information. To this end, FIR filters are uniquely positioned to address this issue. More formally, a FIR is described by a finite sequence $\boldsymbol{\phi}$ of M *filter taps, i.e.*, $\boldsymbol{\phi} = (\phi_1, \phi_2, \ldots, \phi_M)$. For any input $\mathbf{x} \in \mathcal{X}$, the filtered n-th element $\hat{x}[n] \in \hat{\mathbf{x}}$ can be written as

$$\hat{x}[n] = \sum_{m=0}^{M-1} \phi_m x[n-m] \qquad (4)$$

It is easy to observe that by using FIRs, *the adversary can manipulate the position in the complex plane of the transmitted I/Q symbols.* By using complex-valued filter taps, *i.e.*, $\phi_m \in \mathbb{C}$ for all $m = 0, 1, \ldots, M-1$, Eq. (4) becomes:

$$\begin{aligned}\hat{x}[n] &= \sum_{m=0}^{M-1}(\phi_m^{\Re} + j\phi_m^{\Im})(x^{\Re}[n-m] + jx^{\Im}[n-m]) \\ &= \hat{x}^{\Re}[n] + j\hat{x}^{\Im}[n] \end{aligned} \qquad (5)$$

For example, to rotate all I/Q samples by $\theta = \pi/2$ radiants and halve their amplitude, we may set $\phi_1 = \frac{1}{2}\exp^{j\frac{\pi}{2}}$ and $\phi_k = 0$ for all $k > 1$. Similarly, other complex manipulations can be obtained by fine-tuning filter taps. It is clear that complex FIRs can be effectively used by the attacker node to fool the TNN through AWS attacks.

By using a FIR $\boldsymbol{\phi}$ with M complex-valued taps, the waveform $\mathbf{x}_A(\boldsymbol{\phi})$ transmitted by the attacker can be written as

$$\mathbf{x}_A(\boldsymbol{\phi}) = \mathbf{x}_{BB} \circledast \boldsymbol{\phi} \qquad (6)$$

where $\mathbf{x}_A(\boldsymbol{\phi}) = (x_A[n](\boldsymbol{\phi}))_{n=1,\ldots,N_I}$, $x_A[n](\boldsymbol{\phi})$ is computed as in (5), $\mathbf{x}_{BB} = (x_{BB}[n])_{n=1,\ldots,N_I}$ is an intelligible signal (e.g., a portion of a WiFi packet) and $\boldsymbol{\phi} = (\phi_n^{\Re} + j\phi_n^{\Im})_{n=1,\ldots,N_I}$ is the FIR used to generate a synthetic waveform.

4 GENERALIZED WAML PROBLEM (GWAP)

Notice that Problems (AWJ-U), (AWJ-T) and (AWS) are similar in target. Thus, we propose the following Generalized Wireless AML problem (GWAP) formulation

$$\underset{\boldsymbol{\phi}}{\text{maximize}} \quad \sum_{s=1}^{S} \sum_{c \in C} \omega_c f_c(\mathbf{z}_s) \tag{GWAP}$$

$$\text{subject to } \mathbf{g}(\mathbf{z}_s) \leq 0, \; s = 1, ..., S \tag{7}$$

where $\mathbf{g}(\mathbf{z}) = (g_1(\mathbf{z}), \ldots, g_G(\mathbf{z}))^\top$ is a generic set of constraints that reflect BER, energy and any other constraint that the attack strategy $\boldsymbol{\phi}$ must satisfy (*e.g.*, upper and lower bounds); and ω_c takes values in $\{-\rho, -1, 0, 1, \rho\}$ depending on the considered attack. As an example, Problem (AWJ-T) has $\omega_{c_T} = 1$, $\omega_{c_L} = -1$, $\omega_{c_A} = -\rho$ and $\omega_c = 0$ for all $c \neq c_L, c_T, c_A$.

Problem (GWAP) is non trivial since (i) the functions f_c have no closed-form and depend on millions of parameters; (ii) both the objective and the constraints are highly non-linear and non-convex; (iii) it is not possible to determine the convexity of the problem. Despite the above challenges, in whitebox attacks *the adversary has access to the gradients of the TNN* (Figure 1). In the following, we show how an attacker can effectively use gradients to efficiently compute AML attack strategies.

The received waveform is $\mathbf{z} = \mathbf{z}_A + \mathbf{z}_L$. Since $\mathbf{z}_L$ cannot be controlled by the attacker node, we have $f_c(\mathbf{z}) = f_c(\mathbf{z}_A)$. Figure 1 shows that the TNN provides the gradients $\nabla_{\mathbf{z}} f_c(\mathbf{z})$, hence the attacker can compute the gradients $\nabla_{\boldsymbol{\phi}} f_c(\mathbf{z})$ of the activation probability corresponding to the c-th class of the TNN with respect to the attacker's strategy $\boldsymbol{\phi}$ by using the well-known chain rule of derivatives. Specifically, the gradients are

$$\nabla_{\boldsymbol{\phi}} f_c(\mathbf{z}) = J_{\boldsymbol{\phi}}(\mathbf{z})^\top \cdot \nabla_{\mathbf{z}} f_c(\mathbf{z}) \tag{8}$$

where $J_{\boldsymbol{\phi}}(\mathbf{z})$ is the $N_I \times M$ Jacobian matrix of the input $\mathbf{z}$ with respect to the attacker's strategy $\boldsymbol{\phi}$, $\top$ is the transposition operator, and $\cdot$ stands for matrix dot product.

We define the input of the TNN as a set of N_I consecutive I/Q samples, *i.e.*, $\mathbf{z} = (z[n])_{n=0,\ldots,N_I-1}$, where $z_n \in \mathbb{C}$ for all $n = 0, \ldots, N_I - 1$. The attacker's waveform is defined as a sequence of M complex numbers, *i.e.*, $\mathbf{x}_A(\boldsymbol{\phi}) = (x_A[m](\boldsymbol{\phi}))_{m=0,\ldots,M-1}$ whose values depend on the attack strategy $\boldsymbol{\phi}$. With this information at hand, we observe the gradient $\nabla_{\boldsymbol{\phi}} f_c(\mathbf{z})$ has dimension $2M \times 1$, while the gradients with respect to real and imaginary parts of the m-component are

$$\frac{\partial f_c(\mathbf{z})}{\partial \phi_m^{\Re}} = \sum_{n=1}^{N_I} \left(\frac{\partial f_c(\mathbf{z})}{\partial z^{\Re}[n]} \frac{\partial z^{\Re}[n]}{\partial \phi_m^{\Re}} + \frac{\partial f_c(\mathbf{z})}{\partial z^{\Im}[n]} \frac{\partial z^{\Im}[n]}{\partial \phi_m^{\Re}} \right) \tag{9}$$

$$\frac{\partial f_c(\mathbf{z})}{\partial \phi_m^{\Im}} = \sum_{n=1}^{N_I} \left(\frac{\partial f_c(\mathbf{z})}{\partial z^{\Re}[n]} \frac{\partial z^{\Re}[n]}{\partial \phi_m^{\Im}} + \frac{\partial f_c(\mathbf{z})}{\partial z^{\Im}[n]} \frac{\partial z^{\Im}[n]}{\partial \phi_m^{\Im}} \right). \tag{10}$$

From (2), (3) and (6), we have that the gradients for AWJ and AWS attacks can be computed respectively as

$$\frac{\partial z^{Z'}[n]}{\partial \phi_m^{Z''}} = h_{A_{n-m}}[n] \tag{11}$$

and

$$\frac{\partial z^{Z'}[n]}{\partial \phi_m^{Z''}} = \sum_{k=0}^{K-1} h_{A_k}[n] \left(\sum_{m=0}^{M-1} x_{BB}[n-m-k] \right). \tag{12}$$

If compared to fast gradient sign methods (FGSM) [3] (where adversarial inputs are tailored for a specific input and channel condition), we take into account multiple inputs to find a single FIR filter that can synthesize many adversarial inputs that are effective against multiple channel conditions.

Due to space limitations, we omit the detailed solution of Problem GWAP which, however, consists in relaxing C1 and C2 via Lagrangian Duality and applying the Non-linear Conjugate Gradient (NCG) method [16] to iteratively compute the attack solution.

5 EXPERIMENTAL RESULTS

We first describe the datasets and learning architectures in Section 5.1, followed by the results for AWJ (Section 5.2), AWS (Section 5.3).

5.1 Datasets and Learning Architectures

5.1.1 Radio Fingerprinting. We consider (i) a dataset of 500 devices emitting IEEE 802.11a/g (WiFi) transmissions; and (ii) a dataset of 500 airplanes emitting Automatic Dependent Surveillance – Broadcast (ADS-B) beacons[1]. ADS-B is a surveillance transmission where an aircraft determines its position via satellite navigation. For the WiFi dataset, we demodulated the transmissions and trained our models on the derived I/Q samples. To demonstrate the generality of our AML algorithms, the ADS-B model was instead trained on the unprocessed I/Q samples. We use the CNN architecture in [20], where the input is an I/Q sequence of length 288, followed by two convolutional layers (with ReLu and 2x2 MaxPool) and two dense layers of size 256 and 80. We refer to the above CNN models as **RF-W (WiFi) and RF-A (ADS-B) TNN architectures**.

5.1.2 Modulation Classification (MC). We use the RadioML 2018.01A dataset, **publicly available for download at http://deepsig.io/datasets**. The dataset is to the best of our knowledge the largest modulation dataset available, and includes 24 different analog and digital modulations generated with different levels of signal-to-noise ratio (SNR). Details can be found in [11]. For the sake of consistency, we also consider the neural network introduced in Table III of [11], which presents 7 convolutional layers each followed by a MaxPool-2 layer, finally followed by 2 dense layers and 1 softmax layer. The dataset contains 2M examples, each 1024 I/Q samples long. In the following, this model will be referred to as the **MC TNN architecture**. We considered the same classes shown in Figure 13 of [11]. Confusing classes in Fig. 3 ($\epsilon = 0.2$) of our paper and Figure [11] in are the same (*i.e.*, mostly M-QAM modulations). Notice that $\epsilon = 0$ corresponds to no attack.

5.1.3 Data and Model Setup. For each architecture and experiment, we have extracted two distinct datasets for *testing* and *optimization* purposes. The optimization set is used to compute the attack strategies $\boldsymbol{\phi}$ as shown in Sections 3 and 4. The computed $\boldsymbol{\phi}$ are then applied to the testing set and then fed to the TNN. To understand the impact of channel conditions, we simulate a Rayleigh fading channel with AWGN noise $\mathbf{h}_A$ that affects all waveforms that node A transmits to node R. We consider high and low SNR scenarios with path loss equal to 0dB and 20dB, respectively. Moreover, we also consider a baseline case with no fading. To train our neural networks, we use an ℓ_2 regularization parameter $\lambda = 0.0001$. We

[1]Due to stringent contract obligations, we cannot release these datasets to the community. We hope this will change in the future.

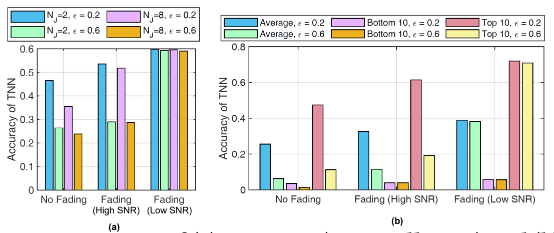
Figure 2: Accuracy of (a) MC TNN (originally 60%) and (b) RF-W TNN (originally 40%) under the AWJ-U attack for different jamming lengths and ϵ values.

also use an Adam optimizer with a learning rate of $l = 10^{-4}$ and categorical cross-entropy as a loss function. All architectures are implemented in Keras. Since all datasets are static, we cannot present results on BER values whose study will be the focus of future work.

5.2 AML Waveform Jamming (AWJ) Results

5.2.1 Untargeted AWJ (U-AWJ). Figure 2(a) shows the accuracy of the MC TNN (original accuracy of 60%) under the AWJ-U attack, for different channel conditions $\mathbf{h}_A$, jamming waveform length N_J and ϵ values. Figure 2 shows that the adversary always reduces the accuracy of the TNN even when N_J and ϵ are small. We notice that high SNR fading conditions allow the adversary to halve TNN's accuracy, while the best performance is achieved in no-fading conditions where the attacker can reduce TNN's accuracy by 3x.

Figures 3 and 4 show the confusion matrices and the corresponding accuracy levels of the AWJ-U attack to the MC TNN model in the low SNR regime. Here, increasing ϵ also increases the effectiveness of the attack, demonstrated by the presence of high values outside the main diagonal of the confusion matrix.

Figure 2(b) shows the accuracy of the RF-W TNN for different attack strategies, constraints and fading conditions. To better understand the impact of AWJ-U, we have extracted the 10 least (*i.e.*, *Bottom 10*) and most (*i.e.*, *Top 10*) classified devices out of the 500 devices included in the WiFi dataset. Interestingly, AWJ-U attacks are extremely effective when targeting the top devices. In some cases, the attacker can drop the accuracy of these devices from 70% to a mere 20% in the high SNR regime. Since the bottom 10 devices are classified with a low accuracy already, it takes minimal effort to alter legitimate waveforms and activate other classes.

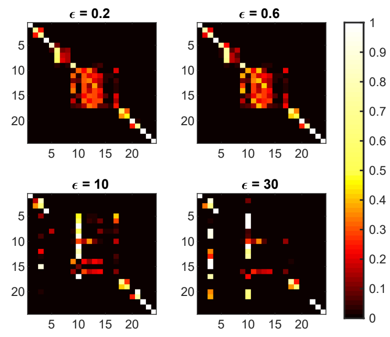
Figure 3: Confusion matrix of MC TNN under the AWJ-U attack in low SNR regime for different ϵ values.

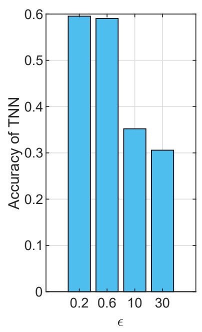
Figure 4: Accuracy of MC TNN in Fig. 3 (originally 60%).

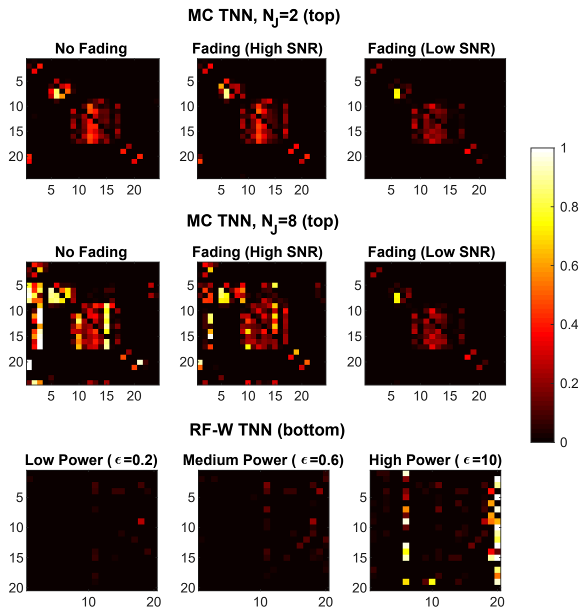
Figure 5: (top) Fooling matrix of MC TNN under AWJ-T for N_J and ϵ values; (bottom) Fooling matrix of RF-W TNN under AWJ-T for different ϵ values and no fading.

5.2.2 Targeted AWJ (AWJ-T). Compared to untargeted jamming, AWJ-T requires smarter attack strategies as the adversary needs to (i) jam an already transmitted waveform, (ii) hide the underlying features of the jammed waveform and (iii) mimic those of another class. The top portion of Figure 5 show the fooling matrices of AWJ-T attacks against MC TNN. Notice that the higher the fooling rate, the more successful the attack is. We notice that the adversary is able to effectively target a large set of modulations from 1 to 17 and 24 (*i.e.*, OOK, M-QAM, M-PSK, ASK). However classes from 18-23 (*i.e.*, AM, FM and GMSK) are hard to be targeted and show low fooling rate values. The bottom portion of Figure 5 shows the results concerning the AWJ-T attack against RF-W TNN.

5.3 AML Waveform Synthesis (AWS) Results

Let us now evaluate the performance of AWS attacks in the case of rogue nodes. In this case, the attacker strategy $\boldsymbol{\phi}$ consists of M complex-valued FIR taps (Section 3.2) that are convoluted with a baseband waveform $\mathbf{x}_{BB}$. To simulate a rogue device, we extract $\mathbf{x}_{BB}$ from the optimization set of the rogue class. This way we can effectively emulate a rogue class that needs to hide its own features and imitate those of the target classes.

Figure 6 depicts the fooling matrices of AWS attacks against the RF-W TNN. We notice that (i) increasing the number of FIR taps increases the fooling rate; and (ii) the bottom classes (1-10) are the ones that the attacker is not able to imitate. However, the same does not hold for the top 10 classes (11 to 20), which can be imitated with high probability (*i.e.*, 28%, 35%, 62% for classes 11,15,20, respectively). Figure 6 gives us an interesting insight on AWS attacks as it shows that the attacker is unlikely to attack those classes that are misclassified by the TNN.

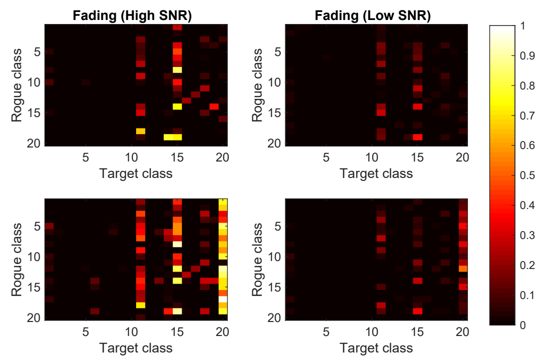

Figure 6: Fooling matrix of RF-W TNN under AWS for different values of M ($M = 4$: top; $M = 8$: bottom).

The same behavior is also exhibited by the RF-A TNN. Figure 7 shows the fooling matrix when $\epsilon = 0.5$ and $M = 4$. Our results show that the attacker is not able to properly imitate classes 1-10 (*i.e.*, the bottom classes). Classes 11-20 (*i.e.*, the top classes) can instead be imitated to some extent. This is because it is unlikely that a unique setup of ϵ and M will work for all classes (both rogue and target).

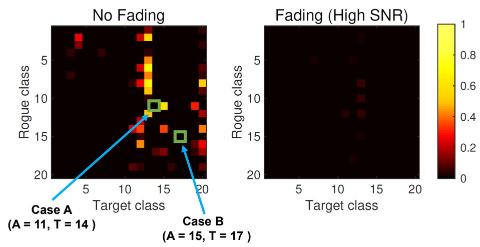

Figure 7: Fooling matrix of RF-A (original accuracy 60%) TNN under AWS with $M = 4$ and $\epsilon = 0.5$.

6 RELATED WORK

Szegedy *et al.* [23] first pointed out the existence of *targeted* adversarial examples: given a valid input x, a classifier C and a target t, it is possible to find $x' \sim x$ such that $C(x') = t$. More recently, Moosavi-Dezfooli *et al.* [9] have further demonstrated the existence of so-called *universal perturbation vectors*, such that for the majority of inputs x, it holds that $C(x + v) \neq C(x)$. Carlini and Kruger [2] evaluated a series of adversarial attacks that are shown to be effective against defensive neural network distillation [15]. Although the above papers have made significant advances, they can only be applied to stationary learning contexts such as computer vision. The presence of non-stationarity makes wireless AML significantly more challenging and thus worth of additional investigation.

Only very recently has AML been approached by the wireless community [13]. Adversarial attacks for modulation classification systems is studied in [1], [6], and [8]. Shi *et al.* [22] propose the usage of a generative adversarial network to spoof a targeted device. However, the evaluation is only conducted through simulation without real dataset. Sadeghi *et al.* [21] proposed two AML algorithms based on a variation of the fast gradient methods [5] and tested on the 11-class RadioML *2016.10A* dataset [12] and with the architecture in [10]. In this paper, we instead consider the much larger RadioML *2018.01A* dataset [11], which has 24 classes.

ACKNOWLEDGEMENT

This work was supported in part by ONR under Grants N00014-19-1-2409 and N00014-20-1-2132 and by the National Science Foundation (NSF) under Grant CNS-1923789.

REFERENCES

[1] Samuel Bair, Matthew DelVecchio, Bryse Flowers, Alan J. Michaels, and William C. Headley. 2019. On the Limitations of Targeted Adversarial Evasion Attacks Against Deep Learning Enabled Modulation Recognition. In *Proceedings of the ACM Workshop on Wireless Security and Machine Learning (WiseML)*. 25–30.

[2] Nicholas Carlini and David Wagner. 2017. Towards Evaluating the Robustness of Neural Networks. In *IEEE Symposium on Security and Privacy (S&P)*. 39–57.

[3] Yinpeng Dong, Fangzhou Liao, Tianyu Pang, Hang Su, Jun Zhu, Xiaolin Hu, and Jianguo Li. 2018. Boosting Adversarial Attacks With Momentum. In *Proceedings of IEEE CVPR*. 9185–9193.

[4] Andrea Goldsmith. 2005. *Wireless communications*. Cambridge university press.

[5] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. 2015. Explaining and Harnessing Adversarial Examples. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

[6] Muhammad Zaid Hameed, Andras Gyorgy, and Deniz Gunduz. 2019. The Best Defense Is a Good Offense: Adversarial Attacks to Avoid Modulation Detection. *arXiv: Learning* (2019).

[7] Jithin Jagannath, Nicholas Polosky, Anu Jagannath, Francesco Restuccia, and Tommaso Melodia. 2019. Machine Learning for Wireless Communications in the Internet of Things: A Comprehensive Survey. *Ad Hoc Networks* 93 (2019).

[8] Brian Kim, Yalin E Sagduyu, Kemal Davaslioglu, Tugba Erpek, and Sennur Ulukus. 2020. 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)*. IEEE, 1–6.

[9] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, Omar Fawzi, and Pascal Frossard. 2017. Universal Adversarial Perturbations. In *Proceedings of IEEE CVPR*.

[10] Timothy J O'Shea, Johnathan Corgan, and T Charles Clancy. 2016. Convolutional Radio Modulation Recognition Networks. In *International Conference on Engineering Applications of Neural networks*. Springer, 213–226.

[11] Timothy J. O'Shea, Tamoghna Roy, and T. Charles Clancy. 2018. Over-the-Air Deep Learning Based Radio Signal Classification. *IEEE Journal of Selected Topics in Signal Processing* 12, 1 (Feb 2018), 168–179.

[12] Timothy J O'shea and Nathan West. 2016. Radio Machine Learning Dataset Generation with Gnu Radio. In *Proceedings of the GNU Radio Conference*, Vol. 1.

[13] Luca Pajola, Luca Pasa, and Mauro Conti. 2019. Threat is in the Air: Machine Learning for Wireless Network Applications. In *Proceedings of the ACM Workshop on Wireless Security and Machine Learning (WiseML)*. ACM, 16–21.

[14] Nicolas Papernot, Patrick McDaniel, Ian Goodfellow, Somesh Jha, Z. Berkay Celik, and Ananthram Swami. 2017. Practical Black-Box Attacks Against Machine Learning. In *Proceedings of ASIA CCS*. ACM, 506–519.

[15] Nicolas Papernot, Patrick McDaniel, Xi Wu, Somesh Jha, and Ananthram Swami. 2016. Distillation as a Defense to Adversarial Perturbations Against Deep Neural Networks. In *Proceedings of IEEE S&P*. 582–597.

[16] Francesco Restuccia, Salvatore D'Oro, Amani Al-Shawabka, Mauro Belgiovine, Luca Angioloni, Stratis Ioannidis, Kaushik Chowdhury, and Tommaso Melodia. 2019. DeepRadioID: Real-Time Channel-Resilient Optimization of Deep Learning-based Radio Fingerprinting Algorithms. In *Proceedings of ACM MobiHoc*.

[17] Francesco Restuccia and Tommaso Melodia. 2019. Big Data Goes Small: Real-time Spectrum-Driven Embedded Wireless Networking through Deep Learning in the RF Loop. In *Proceedings of IEEE INFOCOM*.

[18] Francesco Restuccia and Tommaso Melodia. 2020. DeepWiERL: Bringing Deep Reinforcement Learning to the Internet of Self-Adaptive Things. In *Proceedings of IEEE INFOCOM*.

[19] Francesco Restuccia and Tommaso Melodia. 2020. PolymoRF: Polymorphic Wireless Receivers Through Physical-Layer Deep Learning. In *Proceedings of ACM MobiHoc*.

[20] Shamnaz Riyaz, Kunal Sankhe, Stratis Ioannidis, and Kaushik Chowdhury. 2018. Deep Learning Convolutional Neural Networks for Radio Identification. *IEEE Communications Magazine* 56, 9 (Sept 2018), 146–152.

[21] Meysam Sadeghi and Erik G. Larsson. 2019. Adversarial Attacks on Deep-Learning Based Radio Signal Classification. *IEEE Wireless Communications Letters* 8, 1 (Feb 2019), 213–216.

[22] Yi Shi, Kemal Davaslioglu, and Yalin E. Sagduyu. 2019. Generative Adversarial Network for Wireless Signal Spoofing. In *Proceedings of the ACM Workshop on Wireless Security and Machine Learning (WiseML)*. ACM, 55–60.

[23] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. 2013. Intriguing Properties of Neural Networks. *Proceedings of ICLR* (2013).