Received 1 September 2022; revised 27 September 2022; accepted 7 October 2022. Date of publication 12 October 2022; date of current version 24 October 2022.

Digital Object Identifier 10.1109/OJCOMS.2022.3213371

Countering Physical Eavesdropper Evasion with Adversarial Training

KYLE W. McCLINTICK[1] (Student Member, IEEE), JACOB HARER[2] (Member, IEEE), BRYSE FLOWERS[3] (Student Member, IEEE), WILLIAM C. HEADLEY[4] (Senior Member, IEEE), AND ALEXANDER M. WYGLINSKI[1] (Senior Member, IEEE)

[1]Department of Electrical and Computer Engineering, Worcester Polytechnic Institute, Worcester, MA 01609, USA

[2]Group 0551 Cyber Physical Systems, MIT Lincoln Laboratory, Lexington, MA 02421, USA

[3]Department of Electrical and Computer Engineering, University of California at San Diego, La Jolla, CA 92093, USA

[4]Hume Center, Virginia Polytechnic Institute and State University, Blacksburg, VA 24061, USA

CORRESPONDING AUTHOR: K. W. McCLINTICK (e-mail: kwmcclintick@wpi.edu)

This work was supported by MIT LL Group 0551 Cyber Physical Systems.

ABSTRACT Signal classification is a universal problem in adversarial wireless scenarios, especially when an eavesdropping radio receiver attempts to glean information about a target transmitter's patterns, attributes, and contents over a wireless channel. In recent years, research surrounding the idea of Machine Learning (ML)-based signal classification has focused on modulation classification, with the downstream objective of demodulation. However, while the computer vision data domain has made significant progress in ensuring robust classification of images despite crafted perturbations, this success has not been translated to secure modulation classification. In this work, we perform the first-ever physical test of an eavesdropping ML-based modulation classifier radio, which we trained offline using a ensemble of *i.i.d.* models. Each model is trained with a weighted mixture of data perturbed by iterative, "least likely" white box attacks and non-attacked data. We then tested the ensemble online using coaxial-connected Software Defined Radios (SDRs). We conducted a case study comparing our results to the state-of-the-art computer vision approaches to investigate the presence of "label leaking", model capacity sensitivity, understand the viability of parallel and sequential variations on perturbation training, and assess the effectiveness of iterative attack training. Our results show that perturbations can result in guessing-level classification performance from eavesdroppers, and that varying levels of robustness can be achieved against all presented attacks. These findings confirm that any receiver presents a new attack vector by utilizing ML techniques for classification tasks, and can be vulnerable to evasion attacks at little-to-no cost to transmitters. Consequently, we argue for the use of our training scheme in all ML-based classifying radios where security is a concern.

INDEX TERMS Adversarial perturbations, adversarial training, modulation classification, supervised learning, software defined radio.

I. INTRODUCTION

IN 2014, Goodfellow [1] presented a picture of a panda that the world's state-of-the-art Machine Learning (ML) (Acronym Appendix) algorithms confidently decided is a gibbon. Utilizing the classifier's gradient, an 8-bit integer resolution-bounded noise image was computed and added to the original panda image. Ever since, an arms race has been ongoing between crafting adversarial perturbations and developing countermeasures [2], [3], [4], [5], [6], [7], [8], [9], [10], [11], [12], [13], [14], [15]. It is important that high-risk wireless communications systems employed in applications such as autonomous vehicles [16], agricultural Internet-of-Things (IoT) [17], and military networks [18] are secure from perturbation attacks because incorrect classifications can result in catastrophic financial and/or human costs. Thus the design of trained ML algorithms for these wireless networks must prevent the introduction of additional attack surfaces used by potential adversaries, which requires an understanding of effective perturbation designs in realistic, physical scenarios.

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

As a relatively new field of research, wireless adversarial perturbation papers present many open challenges that have yet to be resolved. Papers that synchronously add physical perturbations to other physical transmissions to fool physical-layer classification systems [19] do not model realistic synchronization errors between the two, with the exception of the time shifted Universal Adversarial Perturbation (UAP) [6] attacks simulated in [20]. Works that simulate transmitters that add optimized perturbations to their own signals [21], [22] generously assume a white box eavesdropper and have not made attempts to design deterministic perturbations that improve various loss metrics. Additionally, several works [21], [22], [23], [24] have noted the frequently used dataset for these studies are generated with several ratio and labeling errors. There has also yet to be an investigation on the effects of the wireless channel applied to the ML-based detection and isolation of these perturbations [7], commonly performed by a statistical distribution estimating algorithms such as Variational Auto Encoders (VAEs) [25] or Generative Adversarial Network (GANs) [26]. Additionally, there remains a need for experimentation to confirm the overwhelmingly simulated works published so far with respect to realistic real-world scenarios involving clock drift, Radio Frequency Front End (RFFE) noise, and other real-world phenomena. Finally, many simulated countermeasures [27], [28], [29] do not consider state-of-the-art adversarial attacks published in the computer vision domain.

A. RELATED WORK

The study of adversarial perturbations in the wireless data domain is relatively new and lagging behind that of leading data domains such as computer vision. Sadeghi and Larsson [20] synchronously added Fast Gradient Sign Method (FGSM) [1] and UAP [6] attacks to received signals over an Additive White Gaussian Noise (AWGN) channel model, highlighting how considerably less power is needed to fool a Convolutional Neural Network (CNN) with perturbations when compared with random jamming signals, and that the UAP [6] attack is robust to time shifts between it and the received signal that simulates synchronization errors. Kim et al. [30] analyzed the same threat model and explored how the adversary can use the channel state information matrix to synchronously deliver power and error-optimized white and black box perturbations. Flowers et al. [24] use a different threat model, where FGSM [1] perturbations were added by a transmitter to its own signal to fool a CNN-based eavesdropper, and investigated the trade-off between BER and adversarial accuracy. The authors found that perturbations strong enough to be effective at lowering eavesdropper classification accuracy came at the cost of a significant number of communication errors (especially higher order modulation signals), that frequency and timing errors possessed a small effect on perturbation effectiveness, and that relatively large, single-step perturbations did not always increase loss because of unstable gradient ascent. These lessons motivated Flowers et al. [21] to design a feedback loop to the transmitter from the adversary to optimize the multi-objective loss functions, which design the perturbations to minimize power consumption, minimize BER, and minimize eavesdropper accuracy. DelVecchio et al. [22] similarly optimized the frequency-domain power and bandwidth of perturbations to maximize communication effectiveness without increasing eavesdropper accuracy. Lin et al. [31], [32], [33] performed an analysis of many state-of-the-art attacks such as Projected Gradient Descent (PGD) [34] against simulated modulation classification datasets. Bao et al. [35] diverged from the modulation classification use-case to analyze the effectiveness of state-of-the-art perturbations used to disrupt (IoT) networks performing device identification. Maroto et al. [27] implemented adversarial training robust to iterative attacks, but experienced label leaking and weak models due to crafting ground-truth-based perturbations that are overly correlated to trained models, ground truth class, and the non-adversarial data. Zhang et al. [28] performed defensive distillation to protect the network from single-step adversarial perturbations, but the process of fooling these networks is well understood [36], [37]. Finally, Sahay et al. [29] performed a 4-class modulation classification adversarial training simulation using both time and frequency domain features, showing a clear improvement over using time-based features alone. However, they did not show evidence that their novel feature extraction offered improvement upon the moment- and cumulant-based features used in state-of-the-art works [38].

In this work, we explore defense approaches against adversarial perturbations in a white box attack regime, as seen in the state-of-the-art. Traditionally, the term “white box” is used in adversarial perturbation scenarios to describe the weights and architecture of the target ML classifier as fully observable by the agent who is crafting perturbations. In our white box scenario, the adversary can observe not only the trained model and its weights, but all aspects of the eavesdropper’s radio and ML systems, such as perturbation detection networks or ensemble classification schemes. We implement our methodology with this assumption because an attack or defense executed with an informational advantage is trivial to study, as it will usually succeed. Additionally, if white box classifier knowledge has been obtained, as in the state-of-the-art, via malware or reverse engineering, it is unclear to us why the perturbation defense sub-system would be unavailable.

Consequently, we do not investigate the use of semi-supervised perturbation detection algorithms [7] because it has been shown they increase the attack surface of the classifier when the adversary is aware of them [4]. We do not investigate the use of gradient masking [5] or defensive distillation [9] because the process of fooling these networks is well understood [36], [37], [39]. Finally, we do not investigate the use of network verification [40], [41], [42], [43] as these computational methods are still prohibitively expensive for all but the smallest datasets and models. While

our classification architectures are relatively small (see Section II-A and Section II-B), we show in Section III-E that making them any smaller such that network verification would be possible, will make them more vulnerable to adversarial perturbations.

Adversarial training has been described as a powerful regularization method [34] that performs a similar function to L_1 regularization on the activations of linear classifiers [1]. When a model is overfitting, adversarial training, defensive distillation, and gradient masking schemes have all doubled as defenses against adversarial perturbations, as well as regularizers that increase the classification PPV of non-adversarial test data.

B. RESEARCH CONTRIBUTIONS AND ORGANIZATION

In this work, we present a number of defensive contributions to the state-of-the-art. These contributions confirm or deny state-of-the-art best practices from the adversarial computer vision domain, as well as establish new ones for wireless communication scenarios through both simulated and experimental demonstrations to make defenses more robust to knowledgeable attackers. Those contributions are as follows:

- Evaluation of the effectiveness of perturbation training to mitigate attacks on a modulation classification model as measured by PPV, ensuring we measure common defense pitfalls discovered in other contexts such as "label leaking";
- Improving network architectures for Radio Frequency Machine Learning (RFML) signal classification in more realistic settings by training models to be robust to attacks and to avoid common defense pitfalls;
- Validation of the state-of-the-art and proposed techniques in a physical setting utilizing unsynchronized radios, such that real-world data demonstrates impact and implementability.

This study impacts wireless communication privacy implications significantly by providing an example of real-world perturbations disrupting an eavesdropper's demodulation efforts in the presence of channel and hardware noise between two unsynchronized radios. A study of the eavesdropper's trade-space in pursuit of demodulation despite those disruptions is also provided.

This paper is organized as follows:

- In Section II, we introduce the real-world physical scenario that motivates adversarial attack of a signal classifier and define metrics of success for such an adversary.
- In Section III, we present several novel studies of state-of-the-art computer vision adversarial training schemes applied to the problem of wireless adversarial perturbation defense.
- In Section IV, we review our contributions and suggest several open challenges to the wireless security community.

II. SYSTEM MODEL

The state-of-the-art perturbation approaches assume one of two diffrent three-player scenarios. In the first scenario, a transmitter and receiver communicate while a reactive adversary eavesdrops, computes the proper perturbation, then synchronously transmits those perturbations to fool the receiving radio, which classifies the observed sum of signals [20], [30]. Alternatively, a transmitter adds pre-channel perturbations to fool an adversarial eavesdropper, which classifies the transmissions while maximizing communication capabilities between the transmitter and receiver [21], [22], [24]. In this paper, we investigate the latter scenario (Fig. 1) while leaving the former for future work.

A. METHODOLOGY

State-of-the art methodology trends are as follows. Adversarial perturbations in the wireless communications domain are typically generated to fool classifiers trained on the RML2016.10A [44] simulated dataset, its successor the RML2018.01A [38] dataset, or datasets modeled after the RML datasets that tune or fix the various channel model parameters, meta-data, or SNR [23]. Previous papers that apply adversarial attacks to this dataset use the same or similar supervised learning model to compute perturbations, the VT-CNN or VT-CNN2 models [45]. Finally, previous papers typically compute white box attacks such as the FGSM attack.

In this work, we create variations inspired by the RML2018.01A [38] datasets in Section III. In a related work [23], the RML2016.10A [44] dataset experiences significant multicollinearity (correlated input data indices), uses a pulse shaping filter without zero-crossings, and does not properly compute energy per symbol ratios. These issues, respectively, limit classifier performance, do not allow bit estimation, and produce SNR-shifted classifier performance results. Consequently, we chose to create a new dataset similar to RML2016.10A in which we increase the number of samples per signal capture from 128 to 4096 to reduce multicollinearity by providing more symbols and symbol transitions per example. Finally, we implement a slightly different Finite Impulse Response (FIR) Root Raised Cosine [46] (RRC) with a rolloff $\alpha = 0.35$ and 12 taps since the RML2016.10A does not possess zero crossings in its RRC filter, as the dataset is not designed for bit estimation.

Our version of the dataset (Fig. 2) implements the cumulative random walk of truncated Gaussian samples for Symbol Rate Offset (SRO) and the cumulative random walk of truncated Gaussian samples for CFO as:

$$x_{O_i} = \mathcal{F}^{-1}\left\{\mathcal{F}\{x\}\left(\cos\left(\frac{2\pi i}{N}\sum_{k=1}^{i}\mathrm{SRO}_k\right)\right.\right.$$

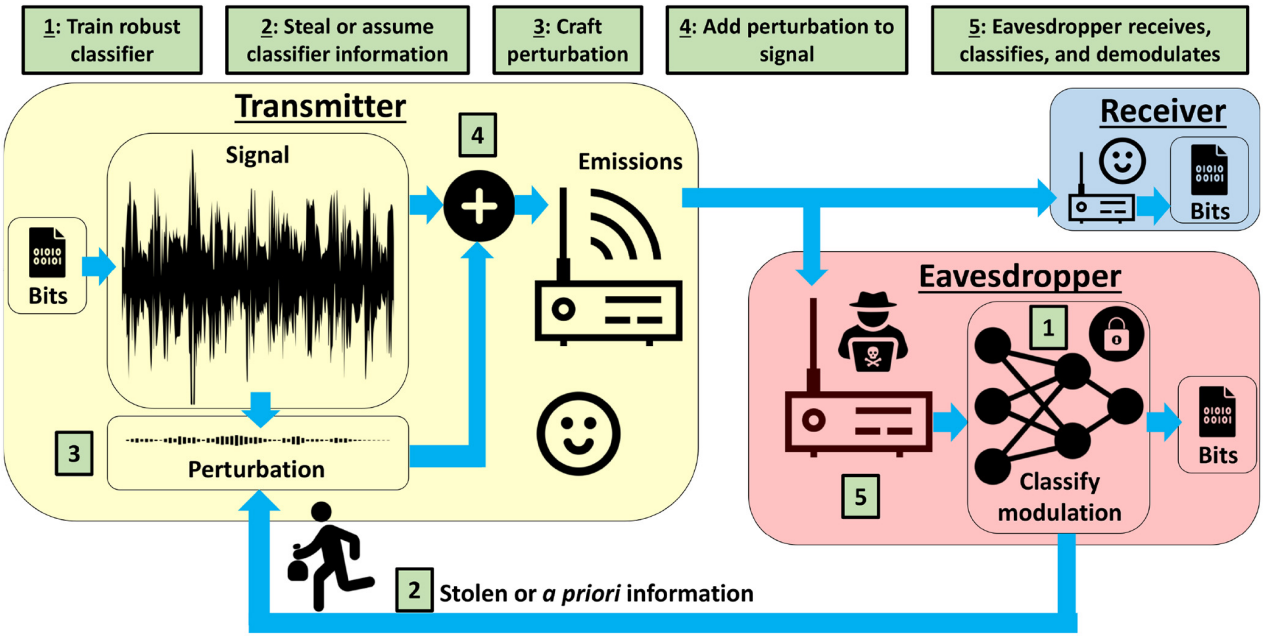


FIGURE 1. The transmitter, given both a signal and a perturbation power constraint, strategically amplifies certain samples of signals such that an adversarial eavesdropper cannot correctly classify the modulation scheme of the observed signals. When successful, the transmitter avoids being demodulated correctly and its bits estimated by the eavesdropper are random and lack any information. We measure the success of the perturbations by how low of a BER they achieve with the intended receiver and by how low of a classification accuracy measured by the Positive Predictive Value (PPV) the eavesdropper achieves in this dual-objective scenario. Conversely, we measure the success of the eavesdropper by how high of a classification PPV it can achieve on observed signals and how many bit errors it can force the transmitter to make in order to avoid correct demodulation.

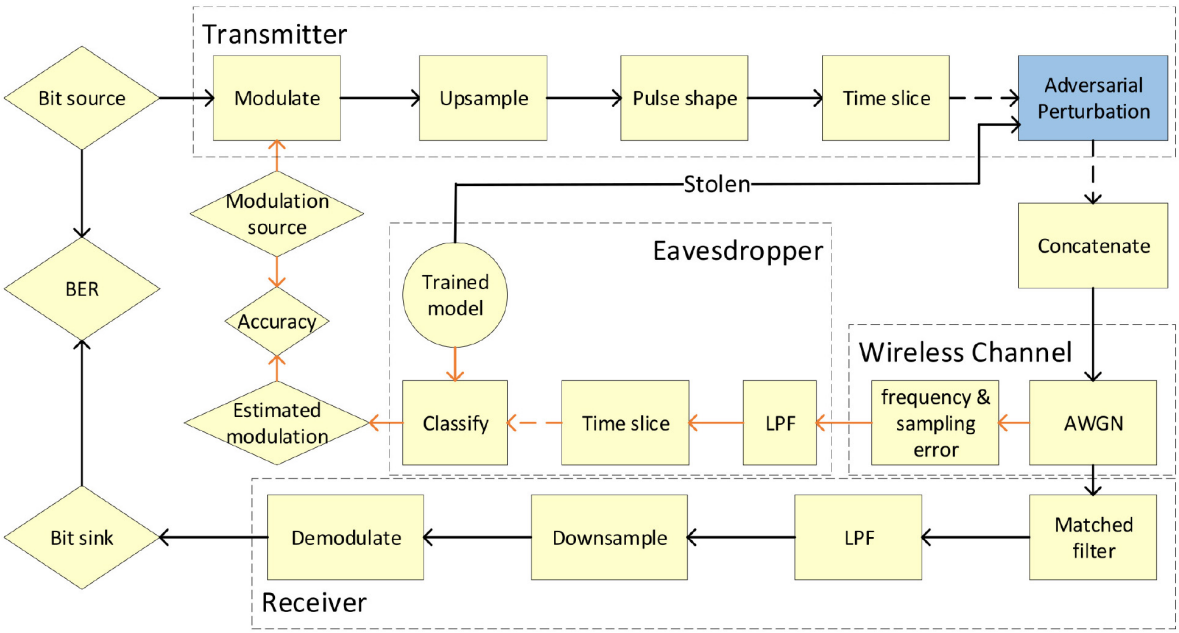


FIGURE 2. A summary of how we generate data for training and inference, evaluate the dual objective, implement attacks, and build robust defenses.

$$\left.-1j\sin\left(\frac{2\pi i}{N}\sum_{k=1}^{i}\mathrm{SRO}_k\right)\right)\Bigg\}$$
$$\times\ \exp\left(\frac{-2j\pi i}{f_s}\sum_{k=1}^{i}\mathrm{CFO}_k\right), \quad (1)$$

for sample index $i = 1, \ldots, N$, vector length $N = 4096$, and sample rate $f_s = 200$ MHz, which are all equal to the values used in [44].

Finally, we implemented an 11th order FIR Butterworth [47] filter with a normalized frequency cutoff of 0.65 to isolate the signals from OOB noise.

TABLE 1. A summary of the mathematical terms found throughout this letter.

I	SPS	$\mathcal{L}(\cdot)$	Cross entropy loss
σ	Standard deviation	λ	Adversarial learning weight
y	Label	k	Adversarial examples per batch
$\hat{y}$	Estimated label	m	Minibatch size
ρ_{mp}	Multipath magnitudes	x	Unperturbed signal
E_p	Perturbation energy	x^*	Perturbed signal
f_s	Sample rate	η	Perturbation
E_s	Signal energy	α	RRC rolloff
E_s/N_0	Energy per bit	α_{iter}	iterLL step size
p	Norm order	α_{adam}	Adam learning rate
ϵ	Perturbation scale	β	Adam average coefficients
$f(\cdot)$	NN forward pass	ϵ_{adam}	Adam stability coefficient
N	Time series length	γ	Adam momentum coefficient
δ	Perturbation constraint	ρ	Filter bank scaling factor
M	Mean layer weight	L	Weights per layer

B. ADVERSARIAL TRAINING WAVEFORMS AND CLASSIFIER

To address the issue of multicollinearity, we used time slices of 4096 complex IQ samples instead of 128. By trial-and-error, we found that by using more samples per signal, we did not need to generate as many signals to achieve the same test PPV, such that our training dataset contains 1.4 million signals instead of 2 million [38]. We additionally implemented the dataset with the following differences from the GR channel model presented in [38]: $\alpha \sim \mathcal{U}(0.35, 0.45)$ instead of $\alpha \sim \mathcal{U}(0.1, 0.4)$, $\sigma_{\text{clk}} = 0.005$ instead of $\sigma_{\text{clk}} = 0.01$, $\tau = [0.0]$ instead of $\tau = [0.0, 0.5, 1.0, 2.0]$, and SNR in the range $E_s/N_0 \in [0, 30]$ dB instead of $E_s/N_0 \in [-20, 30]$ dB. For a summary of parameters, coefficients, and other mathematical terms, see Table 1.

Since the training of models to be robust to perturbations is an adjacent task to the training models that generalize well to test data that differs statistically from training data [34], we generated several physical test sets to see how well our adversarial training schemes perform as regularizers. These test sets are comprised of 1408 signals, which are also called examples, where each of the 88 GR channel models generate 16 signals, as opposed to the training sets wherein 2728 GR channel models generate 512 signals each. Each GR channel model has an independent, fixed modulation class, SNR, and Samples Per Symbol (SPS), I. To generate the data, two USRP N210 SDRs (Fig. 3) are connected via coaxial cable. No digital gain or digital attenuation is used, the radios sample captures using a 1 MHz bandwidth and 20 MHz carrier frequency. We add perturbations to these streams of data via a synchronous Out-Of-Tree (OOT) block in GR implemented before the USRP sink block, which loads the trained Pytorch model, predicts and computes the gradient using the time slice of data, and adds the perturbation to the output. For consistency, we enforce a unit energy constraint on all datasets before classification and perturbation crafting as:

$$x_{norm} = \frac{x\sqrt{N}}{||x||_2}, \tag{2}$$

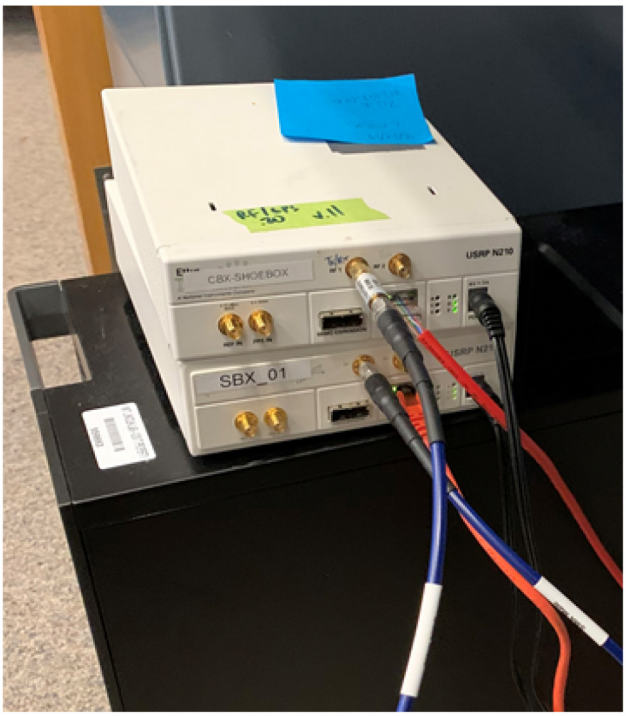


FIGURE 3. USRP N210 SDRs, their coaxial connection, and host computer. The connection employs a 10 dB attenuator.

where N is the length of x. The received test signals do not have an observable DC offset, such that mean subtraction is not implemented in simulated data.

For our IQ data modulation classification adversarial training presented in Section III, we implemented a model deeper than VT-CNN2 in Pytorch, which possesses a higher learning capacity necessary for this work (see Section III-E) when performing adversarial training and training using a dataset with a higher number of classes. We used, as found by trial-and-error, a deep model inspired by the Visual Geometry Group (VGG) 10 [48] CNN model comprised of 9 convolutional layers with ReLU activations [49] and the following number of filters per layer: [64, 64, 128, 128, 256, 256, 256, 256, 256]. The model is terminated with two dense layers with 512 neurons and 22 outputs from the second dense layer. Max pooling is implemented every two layers with stride 2 and size 2. All convolutional layers used have stride 1 and kernel size 3, totaling 18.2×10^6 parameters. All weights are initialized using Kaiming initialization [50]. We did not find that dropout [51] and weight regularization improve classification performance.

The model is optimized in Pytorch by minimizing log softmax plus categorical cross entropy loss via the Adam [52] quasi-Newton method over 20 epochs in mini-batches of size 256. The following Adam parameters are used: $\alpha_{\text{adam}} = 0.0442$, $\beta = (0.9, 0.98)$, $\epsilon_{\text{adam}} = 1 \times 10^{-9}$, $\gamma = 0.1$ with 4,000 warm up steps. No early stopping is implemented, and batch normalization [53] is used. Currently, the literature has observed the strongest perturbations are

those crafted exploiting Neural Networks (NNs) with skip connections, also known as ResNets [54], as a new attack surface [55]. Consequently, we opt to forgo skip connections, despite their advantage in training deep models that are robust to vanishing gradient problems.

C. ADVERSARY GOALS AND DESCRIPTION

An adversarial perturbation is defined as a signal that is added to another signal which is given to a ML model during either training or testing with the intent of causing incorrect estimation or classification during inference. This interaction can be generally described as:

$$x^* = x + \epsilon\eta, \tag{3}$$

where the perturbation, η, is scaled by ϵ and added to the original signal x to form a adversarial example, x^*. If the trained ML model's predictions are described as $f(x) = \hat{y}$, then the perturbations are crafted using the observed or estimated prediction loss function of the model given a signal:

$$\mathcal{L}\big(f(x^*), y\big) > \mathcal{L}(f(x), y), \tag{4}$$

with the expectation that increasing the loss will decrease the performance metrics of the deployed model (i.e., F1-score, precision, recall, AUC, ROC, IoU, mAP). The reason for the scalar ϵ was originally to craft computer vision perturbations that are imperceptible to the human eye, but generally used to minimize the perturbation according to the metric and scheme of choice. Surveys of adversarial attacks and countermeasures are available on these topics from references [2], [3], [4], [5].

In this work, we implemented several different attacks to confirm, deny, or establish best practices presented in leading ML data domains. Due to the variation of perturbations and non-adversarial signals, we define a generalized, adaptive scaling factor based on perturbation energy E_p for all attacks:

$$\epsilon = \frac{\sqrt{10^{\frac{E_s}{E_p}/10}\sum_{i=1}^{n}|x_i|^2}}{||\eta||_2}, \tag{5}$$

where x is the information signal and η is the perturbation signal, which achieves the desired signal (E_s) to perturbation energy ratio:

$$\frac{E_s}{E_p} = 10\log_{10}\left(\frac{\sum_{i=1}^{n}|x_i|^2}{\sum_{i=1}^{n}|\eta_i|^2}\right). \tag{6}$$

The choice of $\frac{E_s}{E_p}$ represents the importance placed by the transmitter on each of the two objectives, being receiver BER and eavesdropper classification PPV.

Given a finite power constraint, it is intuitive that amplifying all samples equally would result in the lowest BER. Yielding some of that power to strategically amplify some samples more than others grants the transmission a measure of obfuscation from fragile ML-based classifiers, at a cost to BER proportional to the power given up. If the transmitter has an objective eavesdropper PPV, the optimal choice for the $\frac{E_s}{E_p}$ ratio cannot be determined without a PPV feedback loop (see [21], [22]) from the eavesdropper to the transmitter, even in a white box scenario where all ML weights and classification rules are known. However, if the wireless channel is well known, as it is in many full duplex links, a BER objective could be used to choose a necessary signal energy, while using the remaining power constraint for perturbation energy.

The attacks used in this work include FGSM [1]:

$$x^* = x + \epsilon\text{sign}(\nabla_x J(x, y_{\text{true}})), \tag{7}$$

where y_{true} is the ground truth label of x, a relatively simple and efficient attack when compared to the others in this work which minimizes $p(y_{\text{true}}|x^*)$. Additionally, we use the One-Step Least Likely (stepLL) attack [34]:

$$x^* = x - \epsilon\text{sign}(\nabla_x J(x, y_{\text{LL}})), \tag{8}$$

where y_{LL} is the least likely predicted class of x as determined by a classifier, which uses the least likely class of the signal according to the class scores of the model to maximize $p(y_{\text{LL}}|x^*)$. This attack is used for adversarial training [34] because FGSM [1] perturbations are substantially deterministic and correlated to the true label. Consequently, adversarial models trained with FGSM attacks classified adversarial data more accurately than non-adversarial test data, while those trained with y_{LL} attacks does not. We visualize some stepLL perturbations in Fig. 4. Finally, we use the Iterative Least Likely (iterLL) attack [34]:

$$\begin{aligned} x^*_{j+1} &= \text{Clip}_{x,-\epsilon,\epsilon}\left\{x^*_j - \alpha_{\text{iter}}\text{sign}(\nabla_x J(x, y_{\text{LL}})\right\}, \\ x^*_0 &= x, \;\; j = 0, \ldots, N \end{aligned} \tag{9}$$

which achieves more powerful perturbations than its one step equivalent by recomputing the direction of the gradient multiple times. In our work, we sample the number of iterations $N \sim \mathcal{U}(2, 10)$ and compute the iteration step size as a ratio, $\alpha_{\text{iter}} = \frac{2\epsilon}{N}$. We leave the investigation of Projected Gradient Descent (PGD) to future work.

III. ADVERSARIAL TRAINING

In this section, perturbation countermeasures are studied by implementing the adversarial training scheme outlined in [34] using our RML2018.01A [38] inspired dataset and the VGG10 [48] inspired modulation classifier from Section II-B, and all attacks presented in Section II-C. We do so using perturbations crafted after first training a non-adversarial model, as in [56], such that we transfer the knowledge of the end results of training. The idea behind adversarial training is to train the model using mini-batches with both perturbations and non-adversarial signals:

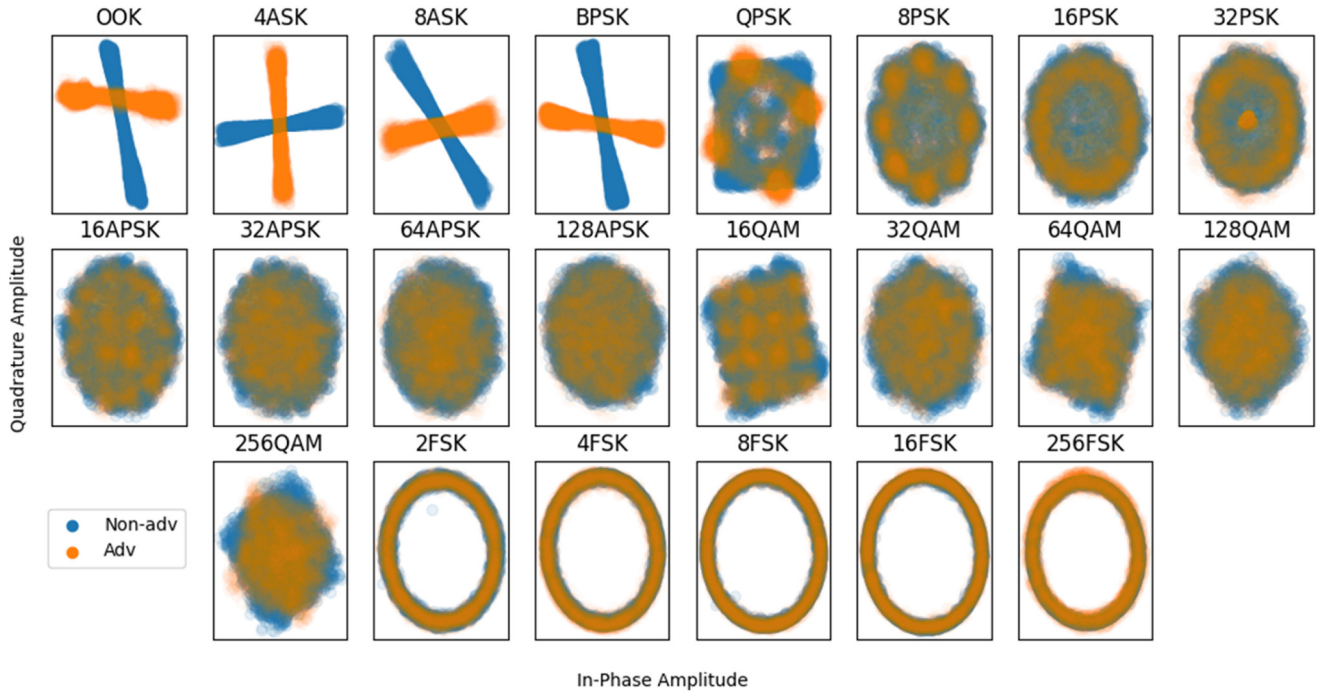


FIGURE 4. One *i.i.d.* (including phase offset) captured perturbation and non-adversarial signal for each modulation class from the connected USRP N210 SDRs from our implementation of the RML2018.01A [38] dataset. These scatter visualizations of the time-series data are over sampled by 8 SPS.

$$\text{Loss} = \frac{1}{(m-k)+\lambda k}\left(\sum_{i=1}^{m-k}\mathcal{L}(x, y_{\text{true}}) + \lambda\sum_{i=1}^{k}\mathcal{L}\big(x^*, y_{\text{true}}\big)\right), \quad (10)$$

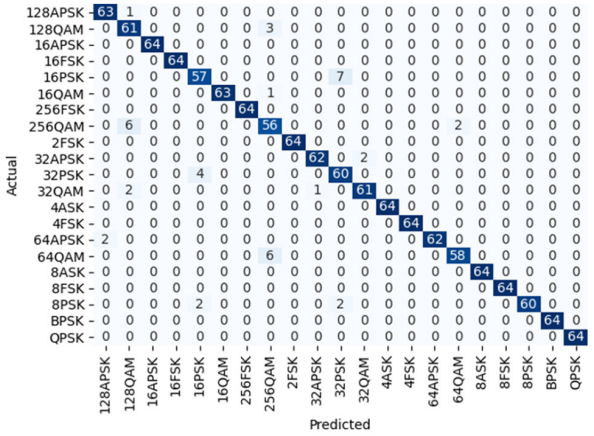
where m is the mini-batch size, k is the number of adversarial examples per mini-batch, $\mathcal{L}(\cdot)$ is categorical cross entropy loss, and λ is the weighting of learning step size for adversarial versus non-adversarial training examples. In this work, we use $m = 256$, $k = 38$, and $\lambda = 1$ such that we achieve what is an effectively equivalent training scheme as seen in [34], who choose $m = 32$, $k = 16$, and $\lambda = 0.3$. We quantify the similarity of these parameter choices as $\frac{m\lambda}{k} = 0.15$. As in [34], we randomly vary perturbation strength such that the adversarial trained model generalizes well to test-stage perturbations of different strengths. We accomplish this variation using a truncated Gaussian distribution as:

$$\epsilon^* = \text{Clip}_{\epsilon,0,1}\{\epsilon + \delta\}, \quad \delta \sim \mathcal{N}(0, 1/2), \quad (11)$$

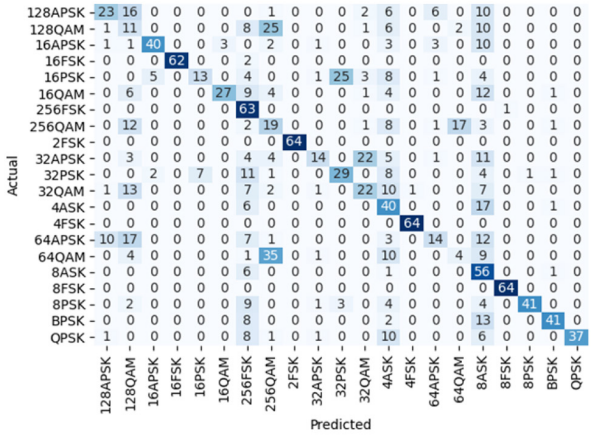
and refer to the value of E_s/E_p for this scheme as "sweeping". We perform the costly, relative to computer vision, training schemes presented in this section using a Intel Xeon Gold 6248 CPU node with 20 cores and 192 GB of RAM, and one NVIDIA Volta V100 GPU node with 32 GB of RAM.

A. EVALUATION OF NON-ADVERSARIAL MODEL

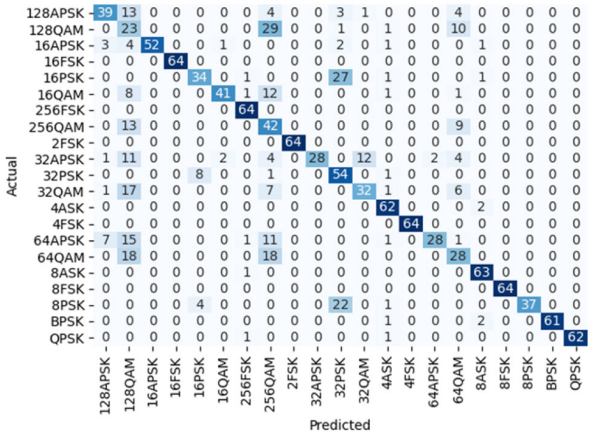
We first evaluate the non-adversarial model as a base line, unprotected classifier. In evaluating the non-adversarial training scheme, we made a number of discoveries. We found that Frequency Shift Keying (FSK) modulation classes are the most difficult to fool, with only three false positives across all modulation orders of FSK in an FGSM attack. This is due to the frequency shifts between each symbol being large. We found that FSK modulation with smaller shifts were easier to fool. The crafting of frequency-domain perturbation is the subject of ongoing research and will be the focus of a subsequent publication. When stronger attacks, deeper models, or larger perturbation energy are used, more FSK signals are fooled. We found that FGSM attacks perform better than stepLL attacks, because they lower the class score of the true class rather than increased the score of the least likely class. We also observed the iterLL attack is the most effective attack because it most accurately ascends the gradient due to taking multiple, smaller steps. Additionally, most test sets showed that, when attacked, they attempt to fool all classifications to be one of a few classes. For instance, 57% of false positives caused by iterLL attacks on the non-adversarial trained model belonged to the 256FSK class, 23% to the 8 Amplitude Shift Keying (ASK) class, and 20% to all other classes. Finally, we observed that increasing perturbation strength decreases modulation classification PPV, which is to be expected. Specifically, $E_s/E_p = 0$ dB stepLL attacks are required to approach a PPV equal to that of a zero rule classifier, and that $E_s/E_p > 35$ dB stepLL attacks had no effect on physical test PPV. On average, perturbations



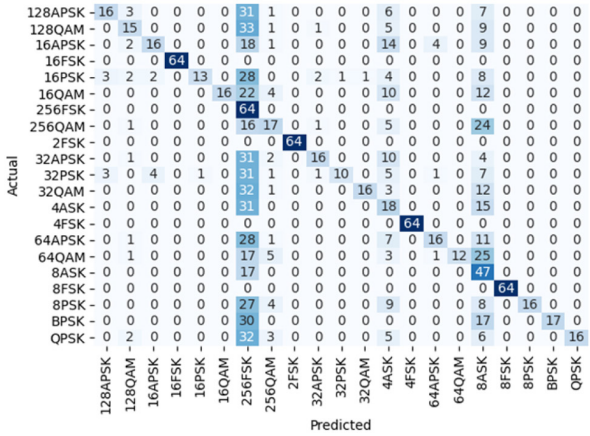
(a) Physical test



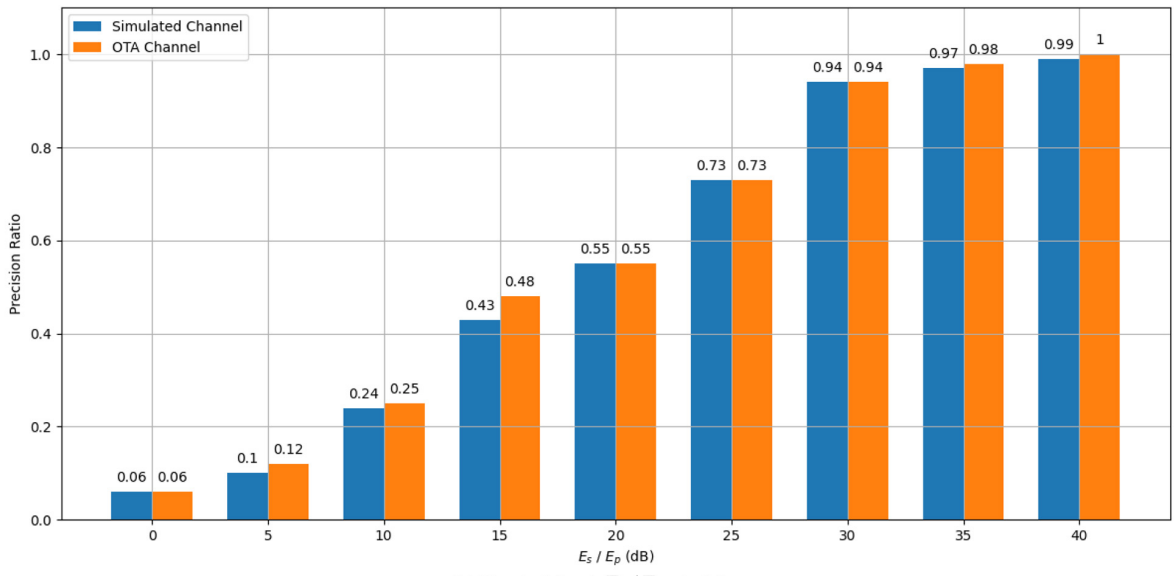
(b) Physical sweeping E_s/E_p FGSM test



(c) Physical sweeping E_s/E_p stepLL test



(d) Physical sweeping E_s/E_p iterLL test



(e) Physical fixed E_s/E_p stepLL

FIGURE 5. **A class-by-class analysis of the effectiveness of each attack and strength of attack on the non-adversarial trained VGG10 model. Most false positives belong to the same one or two classes. IterLL attacks are the strongest, followed by FGSM, and stepLL. FSK classes are the most difficult to fool due to large frequency shifts between each symbol. Perturbations sent over a physical channel are slightly less effective than perturbations transmitted over a simulated wireless channel.**

sent over a physical channel are slightly less effective, relative to non-adversarial PPV, than perturbations transmitted over a simulated wireless channel (Fig. 5). This is due to a covariate shift between training phase simulated channels and test phase physical channels. The size of the PPV ratio gap is proportional to that covariate shift.

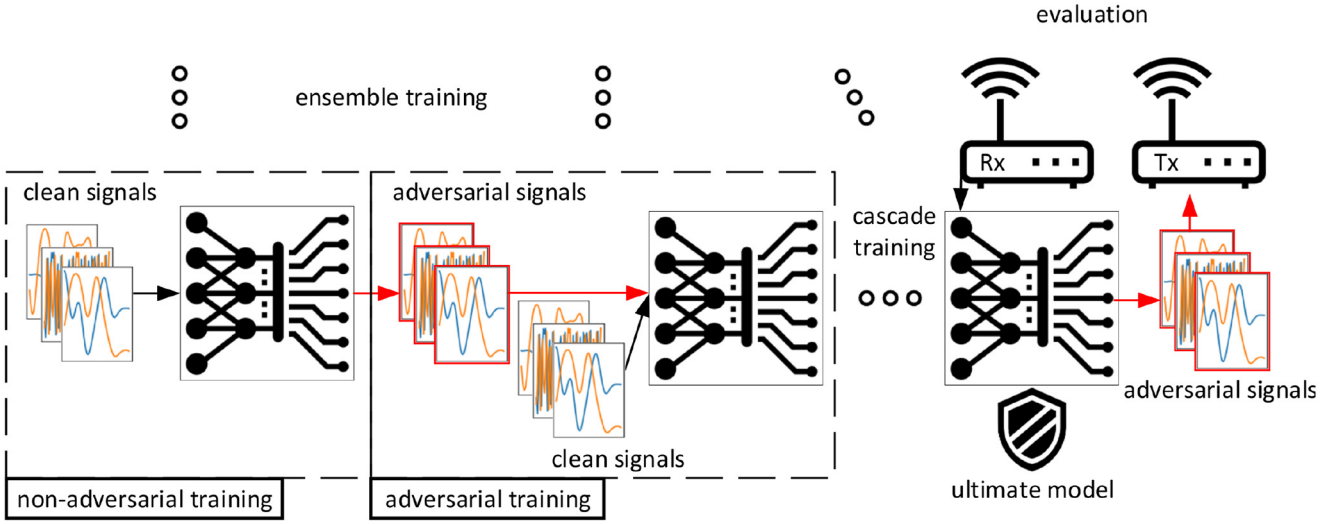


FIGURE 6. Our offline adversarial training framework mostly follows that outlined in [34], although we decouple training by only generating perturbations from already trained models, as in [2]. Additionally, unlike any other paper, we evaluate our model using perturbations crafted from gradients computed from the ultimate model, and do so using online, physical signal captures. Our reasoning is that if our system is vulnerable to an attack once, it can be attacked again, and to assume that the attack is done without knowledge of our countermeasure is overly optimistic. Each model and dataset is *i.i.d.*, and the training of the ultimate model is always done with the same number of weight updates as outlined in Section II-B. For instance, if we produce a parallel set of adversarial training data using three models, we would train the ultimate model using three sets of 1.4/3 million signals for 20 epochs each.

TABLE 2. Effect of various adversarial training schemes on the modulation classification PPV of different partitions of data. step_{i-1} perturbations refers to testing models using perturbations from the same distribution as training set perturbations, where step_i perturbations refers to testing model using perturbations crafted after adversarial training. The adversarial training maintains ∼ 26% of its protection against current step physical attacks compared to physical attacks crafted during training. Furthermore, as in [2], the model trained by the parallel training scheme is more accurate when evaluated on adversarial data at the cost of non-adversarial accuracy. In [2], this gain is seen only for black box attacks, not white box attacks. Our current step white box attacks are analogous to black box attacks from the perspective of adversarial training because test-phase perturbations are crafted from a different set of weights than that from which training perturbations are crafted. Finally, we observed that the cascade adversarial training scheme follows the same trend as the parallel scheme but with greater magnitude.

	Training Scheme			
PPV	Clean	Adversarial	Parallel	Cascade
Training	99.85	99.85	**99.90**	**99.90**
Clean Test	**99.22**	99.15	98.22	95.74
Clean physical Test	96.35	**96.80**	96.21	92.71
stepLL_{i-1} Test	-	76.07	-	-
stepLL_{i-1} physical Test	-	75.22	-	-
stepLL_i Test	70.45	73.58	70.53	**77.63**
stepLL_i physical Test	68.67	70.39	71.88	**81.53**

B. EVALUATION OF CASCADE AND PARALLEL MODELS

Here we evaluate the performance of the protected classifier, as well as the parallel and series extensions of that protection scheme. Parallel [2] (cascade [56]) adversarial training is a parallel (sequential), method of decoupling the generation of adversarial training examples from the model being trained. The theory behind parallel decoupling is that perturbations are transferable between models and that parallel adversarial training schemes will achieve a better approximation of the underlying distribution of perturbations than adversarial training using perturbations crafted from a single pre-trained model, providing greater protection against black box attacks or new white box attacks generated by the fully trained model. The knowledge transferred by a parallel set of perturbations is statistically diverse and high variance, competing with non-adversarial training data for learning capacity in small models [2], such that under fitting occurs if the model size is not increased appropriately.

The theory behind cascade adversarial schemes is that each iteration of training transfers additional information about how perturbations are crafted from already trained models to the ultimate model. We hypothesize there is some number of cascade training iterations and parallel set size that is optimal for a given scenario, and seek to identify the performance trends of these schemes via physical experimentation on models trained offline.

The number of training samples and number of training epochs for the ultimate model were held constant across all of these schemes (Fig. 6) such that the resulting PPV of each scheme will be the result of the knowledge transferred by training perturbations and not the duration of training or quantity of data.

In Table 2, adversarial training maintains about 26% of its protection against current step attacks compared to attacks used in training. Additionally, the ultimate models trained using the parallel training scheme perform worse in all scenarios except for attacks crafted using a model other than that used in adversarial training, or that their robustness is transferable at the cost of regularization. Finally, these models trained using the cascade scheme follow the same trends, but to a greater magnitude than parallel training schemes.

C. LABEL LEAKING

Here we ensure that our protection scheme does not over fit the classifier to depend on perturbations for good

TABLE 3. An investigation of "label leaking" [34] occurring when using FGSM adversarial training schemes, justifying the use of the stepLL attack in training over the use of the FGSM attack. While we do not see evidence of label leaking for this dataset, we find that stepLL training yielded higher protection against iterative and FGSM attacks than FGSM training, which are the most dangerous attacks.

	Training Scheme		
PPV	Clean	FGSM	stepLL
Training	**99.85**	99.73	**99.85**
Clean physical Test	96.35	96.14	**96.80**
stepLL physical Test	68.67	**76.70**	70.39
iterLL physical Test	44.42	42.12	**44.49**
FGSM physical Test	59.23	60.94	**64.96**

TABLE 4. IterLL attacks are significantly more effective than stepLL attacks. StepLL training offer almost no defense against iterLL attacks. We are able to achieve iterLL trained models with a small level of defense against iterLL attacks, and higher defense against stepLL and FGSM attacks with no significant loss to non-adversarial performance.

	Training Scheme		
PPV	Clean	stepLL	iterLL
Training	99.85	99.85	**99.89**
Clean physical Test	96.35	**96.80**	96.65
FGSM physical Test	59.23	64.96	**65.06**
stepLL physical Test	68.67	70.39	**76.21**
iterLL physical Test	44.42	44.49	**45.88**

performance. Label leaking is described in [34] as when adversarial training with the use of ground truth labels in attacks such as FGSM [1] results in a trained model that tests better on adversarial data than non-adversarial data for an individual signal, with and without its added perturbation. Specifically, a label has leaked for a test signal if x^* is classified correctly but x is not. Label leaking is not possible in our experiments since we disjoint crafting by discarding x when we craft x^*, as in [2], which is one of the reasons we have used such a technique. However, we can still interpret the modulation classification PPV obtained on *i.i.d.* populations of adversarial and non-adversarial test signals to determine if models have been over trained with perturbations. This is because the intuition behind label leaking is that ground truth based attacks perform a deterministic transform on data that is highly correlated to the ground truth. As a consequence, if we define the PPV ratio of a model as the PPV of adversarial data divided by the PPV of non-adversarial data, then test sets with leaked labels will achieve a PPV ratio > 1.

To validate the presence and severity of label leaking in wireless experiments and contrast those findings with those in relatively high dimension, zero noise computer vision works [34], we implement the adversarial training methodology presented by Fig. 6 with FGSM attacks. In Table 3, we do not observe any evidence of label leaking, but we do see evidence that stepLL training resulted in more robust models against iterative and FGSM attacks than FGSM training.

D. EVALUATION OF MODELS TRAINED WITH ITERATIVE ATTACKS

Here we investigate the trade-space of computational cost and attack effectiveness against our protected model. In [34], the authors found that adversarial training with iterative attacks did not train models robust to iterative attacks. They hypothesized that they did not have the computational resources to train their Inception v3 [57] model on ImageNet [58] data with a large enough learning capacity to learn the complex distribution of iterative attacks. In [59], the authors reduced the computational cost of iterative Projected Gradient Descent (PGD) [34] attack training by generating Canadian Institute for Advanced Research (CIFAR)-10 and CIFAR-100 [60] adversarial perturbations during training by using the gradient computed for SGD, rather than re-computing. They achieve a moderate level of protection at a very low computational cost.

In this work, we performed iterLL adversarial training using a RML2018.01A inspired dataset to see what degree of protection we may obtain from iterLL and other attacks. We do so without the dual-use of the gradient as in [59] because crafting perturbations during training rather than after does not result in disjoint crafting as in [2]. Additionally, we hypothesized that our relatively low dimension data (i.e., 8192 features/example for the RML2018.01A inspired dataset versus 544509 average features/example for ImageNet [58]), relatively smaller model (i.e., 18.2×10^6 parameters in our VGG10 inspired model versus 24×10^6 parameters in Inception v3), and several years of computational resource advancements (i.e., Volta 100 versus Tesla K80 Graphics Processing Units (GPUs)) will render the dual-use unnecessary.

In Table 4, we observed that iterLL attacks are 206% more effective than stepLL attacks for our dataset, model, and attack parameters. Additionally, stepLL training offered no significant defense against iterLL attacks, prompting the need for an iterLL training scheme. The results of our iterLL training are very positive, showing an increased defense against all attacks without losing non-adversarial performance. Most notably, it is the only training scheme that achieved any level of protection against iterative attacks.

E. MODEL CAPACITY

Here we ensure that our protection scheme does not under fit because it lacks enough trainable parameters to learn both perturbed and non-perturbed data distributions. In other works [34], the authors were unable to find a model deep enough to over fit in the presence of adversarial training using the stepLL method. We scale model width by increasing the number of convolutional filters in every convolutional layer by a factor ρ. While our model utilizes batch normalization to some effect, we do not find dropout to improve test-stage PPV.

In this work, we investigated the effectiveness of stepLL adversarial training as a regularizer in wireless experiments. We hypothesized the relatively low dimension data, relatively small models, and several years of computational resource improvements will make it more feasible to scale to extreme ρ values.

TABLE 5. Effect of model capacity on adversarial training, evaluated using physical test data. We find that adversarial training prevents overfitting from occurring when training our VGG10 model scaled by $\rho = 4$. We additionally find that stepLL perturbations crafted after adversarial training are more effective against deeper models, indicating a model capacity trade-off between non-adversarial and adversarial test classification PPV. Models that are too shallow additionally make lower confidence classifications than deep models, such that they are easier to fool. "Clean" is short hand for non-adversarial data.

	Training Scheme							
	$\rho = 0.5$		$\rho = 1$		$\rho = 2$		$\rho = 4$	
PPV	Clean	stepLL	Clean	stepLL	Clean	stepLL	Clean	stepLL
Training	99.71	99.71	99.85	99.85	99.89	99.88	99.90	99.89
Clean physical Test	95.46	96.06	96.35	96.80	97.47	97.49	97.40	**97.54**
stepLL physical Test	-	58.04	-	**70.39**	-	58.89	-	37.80

In Table 5, we were able to scale $\rho \in [0.5, 4]$ before running out of memory. We found that at $\rho = 4$ the non-adversarial trained VGG10 began to over fit to training data because it had a lower physical test data classification PPV than the $\rho = 2$ non-adversarial trained model. However, with adversarial training, the model is regularized and physical test data classificaiton PPV continues to increase with ρ. Additionally, deeper models were more vulnerable to adversarial perturbations, which can be explained by [1], where it was shown that FGSM perturbations increased the magnitude of activations by $\epsilon \times L \times M$, where M is the average value of weights in a layer and L is the number of weights in a layer. We hypothesized that by increasing ρ, we are increasing L, such that perturbations, all else equal, will have a greater impact on classification PPV. We tested this hypothesis by computing the ratio of mean class score magnitudes between clean physical and stepLL physical test data for adversarial trained models with $\rho = 1$ and $\rho = 4$. We obtained resulting ratios of 0.39 and 0.33, failing to reject our hypothesis that perturbations increase the magnitude of class scores, on average, proportional to the number of weights in each layer of a CNN.

We observed the shallow $\rho = 0.5$ model is also more vulnerable to attacks. One potential explanation for this is it made lower confidence classifications that are easier to fool. To test this, we computed for physical test sets the average difference in class scores between the largest and second largest class scores for $\rho = 0.5$ and $\rho = 1$ adversarial trained models. We found that they had an average top (second top) class score difference of 71.97 (91.11), failing to reject our hypothesis that the shallow model makes less confident classifications.

Consequently, we determined that model width must be carefully managed in adversarial training schemes to ensure that the model is deep enough to learn the non-adversarial and adversarial datasets, deep enough to make high-confidence classifications that require large changes to class scores to cause false positives, and shallow enough as not to become vulnerable to the compounding attribute of attacks. Additionally, we concluded this trade-off is relatively advantageous for adversarial training of wireless spectrum sensing, signal classification, and modulation classification when compared to computer vision tasks, which tend to require much deeper models to learn relatively high dimension data distributions that have large state spaces.

IV. CONCLUSION

We performed in Section III, and outlined the details in Section II-B, the first physical adversarial ML-based modulation class eavesdropping experiment. Given the significant research interest in modulation classification [38], [44], [45], [61], [62], [63], [64] and adversarial wireless ML [20], [21], [22], [23], [24], [27], [28], [29], [30] this novel experiment is a significant real-world validation for many theoretical works that have experimented largely with simulated channel models and signals.

These simulations and experiments yielded a number of findings and confirmations to the state-of-the-art, including:

1) Training a CNN offline using channel models can achieve high accuracy modulation classification performance on physical signals.
2) Physical Adversarial perturbations of a transmitter can reduce the classification accuracy of an eavesdropping receiver's trained ML classifier to as low as guessing despite phase, frequency, and amplitude noise sources from both the RFFE and the channel.
3) Adversarial training of the eavesdropping receiver using simulated channel models can achieve some level of defense against adversarial perturbations, where the best results are achieved when adversarial training is done using perturbations crafted from a fully trained, *i.i.d.* non-adversarial model.
4) Label leaking does not appear to occur in low-dimensional data domains.
5) Parallel and cascade adversarial training schemes over-emphasize adversarial examples during training, reducing testing accuracy for non-adversarial data. This defeats the primary objective of adversarial training, which is to increase robustness without sacrificing non-adversarial performance
6) A measure of protection against iterative attacks is possible with iterLL training.
7) The model width of the eavesdropping receiver must be carefully managed to achieve an "elbow" point in the trade-off between non-adversarial and adversarial test performance. Specifically, we found the CNN must

be wide enough to make correct and high confidence classifications, wide enough to have the learning capacity for both adversarial and non-adversarial PDFs, and thin enough as not to compound the increase to the loss function caused by perturbations.

APPENDIX

ASK	Amplitude Shift Keying
AWGN	Additive White Gaussian Noise
BER	Bit Error Rate
BPF	Band Pass Filter
CFO	Carrier Frequency Offset
CIFAR	Canadian Institute for Advanced Research
CNN	Convolutional Neural Network
FGSM	Fast Gradient Sign Method
FIR	Finite Impulse Response
FSK	Frequency Shift Keying
GAN	Generative Adversarial Network
GPU	Graphics Processing Unit
GR	GNU Radio Companion
IoT	Internet-of-Things
iterLL	Iterative Least Likely
IQ	In-phase Quadrature
LPF	Low Pass Filter
ML	Machine Learning
NN	Neural Network
OOB	Out-of-Band
OOT	Out-of-Tree
PGD	Projected Gradient Descent
QoS	Quality of Service
ReLU	Rectified Linear Unit
RFFE	Radio Frequency Front End
RFML	Radio Frequency Machine Learning
RRC	Root Raised Cosine
SDR	Software Defined Radio
SGD	Stochastic Gradient Descent
SNR	Signal-to-Noise-Ratio
SPS	Samples Per Symbol
SRO	Symbol Rate Offset
stepLL	One-Step Least Likely
UAP	Universal Adversarial Perturbation
USRP	Universal Software Radio Peripheral
VAE	Variational Auto Encoder
VGG	Visual Geometry Group

REFERENCES

[1] I. J. Goodfellow, J. Shlens, and C. Szegedy. "Explaining and harnessing adversarial examples." 2014. [Online]. Available: https://arxiv.org/abs/1412.6572.

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

[3] X. Yuan, P. He, Q. Zhu, and X. Li, "Adversarial examples: Attacks and defenses for deep learning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 9, pp. 2805–2824, Sep. 2019.

[4] N. Papernot, P. McDaniel, S. Jha, M. Fredrikson, Z. B. Celik, and A. Swami, "The limitations of deep learning in adversarial settings," in *Proc. IEEE Eur. Symp. Secur. Privacy (EuroSP)*, 2016, pp. 372–387.

[5] N. Papernot, P. McDaniel, I. Goodfellow, S. Jha, Z. B. Celik, and A. Swami, "Practical black-box attacks against machine learning," in *Proc. ACM Asia Conf. Comput. Commun. Secur.*, 2017, pp. 506–519.

[6] S.-M. Moosavi-Dezfooli, A. Fawzi, O. Fawzi, and P. Frossard, "Universal adversarial perturbations," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 1765–1773.

[7] J. H. Metzen, T. Genewein, V. Fischer, and B. Bischoff, "On detecting adversarial perturbations," 2017, *arXiv:1702.04267*.

[8] N. Papernot, P. D. McDaniel, I. J. Goodfellow, S. Jha, Z. B. Celik, and A. Swami, "Practical black-box attacks against machine learning," 2016. [Online]. Available: http://arxiv.org/abs/1602.02697.

[9] N. Papernot, P. McDaniel, X. Wu, S. Jha, and A. Swami, "Distillation as a defense to adversarial perturbations against deep neural networks," in *Proc. IEEE Symp. Secur. Privacy (SP)*, 2016, pp. 582–597.

[10] O. Poursaeed, I. Katsman, B. Gao, and S. J. Belongie, "Generative adversarial perturbations," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 4422–4431.

[11] N. Akhtar, J. Liu, and A. Mian, "Defense against universal adversarial perturbations," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 3389–3398.

[12] A. Fawzi, O. Fawzi, and P. Frossard, "Analysis of classifier's robustness to adversarial perturbations," *Mach. Learn.*, vol. 107, no. 3, pp. 481–508, 2018.

[13] W. Zhou et al., "Transferable adversarial perturbations," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 471–486.

[14] A. Athalye, L. Engstrom, A. Ilyas, and K. Kwok, "Synthesizing robust adversarial examples," in *Proc. 35th Int. Conf. Mach. Learn.*, vol. 80, 2018, pp. 284–293. [Online]. Available: https://proceedings.mlr.press/v80/athalye18b.html

[15] D. Hendrycks, K. Zhao, S. Basart, J. Steinhardt, and D. Song, "Natural adversarial examples," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 15262–15271.

[16] V. L. Thing and J. Wu, "Autonomous vehicle security: A taxonomy of attacks and defences," in *Proc. IEEE Int. Conf. Internet Things (iThings) IEEE Green Comput. Commun. (GreenCom) IEEE Cyber Phys. Soc. Comput. (CPSCom) IEEE Smart Data (SmartData)*, 2016, pp. 164–170.

[17] S. Prasanna and S. Rao, "An overview of wireless sensor networks applications and security," *Int. J. Soft Comput. Eng.*, vol. 2, no. 2, pp. 2231–2307, 2012.

[18] M. Winkler, K.-D. Tuchs, K. Hughes, and G. Barclay, "Theoretical and practical aspects of military wireless sensor networks," *J. Telecommun. Inf. Technol.*, vol. 2, no. 2, pp. 37–45, 2008.

[19] K. W. Mcclintick, G. M. Wernsing, P. V. R. Ferreira, and A. M. Wyglinski, "Parameter estimation and classification via supervised learning in the wireless physical layer," *IEEE Access*, vol. 9, pp. 164854–164886, 2021.

[20] M. Sadeghi and E. G. Larsson, "Adversarial attacks on deep-learning based radio signal classification," 2018, *arxiv.org/abs/1808.07713*.

[21] B. Flowers, R. M. Buehrer, and W. C. Headley, "Communications aware adversarial residual networks for over the air evasion attacks," in *Proc. IEEE Mil. Commun. Conf. (MILCOM)*, 2019, pp. 133–140.

[22] M. DelVecchio, V. Arndorfer, and W. C. Headley, "Investigating a spectral deception loss metric for training machine learning-based evasion attacks," *Proc. 2nd ACM Workshop Wireless Security Mach. Learn.*, Jul. 2020, pp. 43–48. [Online]. Available: http://dx.doi.org/10.1145/3395352.3402624

[23] K. W. McClintick and A. M. Wyglinski, "Reproduction o "evaluating adversarial evasion attacks in the context of wireless communications" and" convolutional radio modulation recognition networks," " in *Proc. Workshop Benchmarking Cyber-Phys. Syst. Internet Things*, 2021, pp. 1–5.

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

[25] D. Meng and H. Chen, "MagNet: A two-pronged defense against adversarial examples," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, 2017, pp. 135–147. [Online]. Available: https://doi.org/10.1145/3133956.3134057

[26] G. Jin, S. Shen, D. Zhang, F. Dai, and Y. Zhang, "APE-GAN: Adversarial perturbation elimination with GAN," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP)*, 2019, pp. 3842–3846.

[27] J. Maroto, G. Bovet, and P. Frossard, "SafeAMC: Adversarial training for robust modulation recognition models," 2021. [Online]. Available: https://arxiv.org/abs/2105.13746.

[28] L. Zhang, S. Lambotharan, G. Zheng, B. A. Sadhan, and F. Roli, "Countermeasures against adversarial examples in radio signal classification," *IEEE Wireless Commun. Lett.*, vol. 10, no. 8, pp. 1830–1834, Aug. 2021.

[29] R. Sahay, C. G. Brinton, and D. J. Love, "A deep ensemble-based wireless receiver architecture for mitigating adversarial attacks in automatic modulation classification," *IEEE Trans. Cogn. Commun. Netw.*, vol. 8, no. 1, pp. 71–85, Mar. 2022.

[30] 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," 2020. [Online]. Available: https://arxiv.org/abs/2002.02400

[31] Y. Lin, H. Zhao, Y. Tu, S. Mao, and Z. Dou, "Threats of adversarial attacks in DNN-based modulation recognition," in *Proc. IEEE INFOCOM IEEE Conf. Comput. Commun.*, 2020, pp. 2469–2478.

[32] Y. Lin, H. Zhao, X. Ma, Y. Tu, and M. Wang, "Adversarial attacks in modulation recognition with convolutional neural networks," *IEEE Trans. Rel.*, vol. 70, no. 1, pp. 389–401, Mar. 2021.

[33] H. Zhao, Y. Lin, S. Gao, and S. Yu, "Evaluating and improving adversarial attacks on DNN-based modulation recognition," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, 2020, pp. 1–5.

[34] A. Kurakin, I. Goodfellow, and S. Bengio, "Adversarial examples in the physical world," 2016. [Online]. Available: https://arxiv.org/abs/1607.02533.

[35] Z. Bao, Y. Lin, S. Zhang, Z. Li, and S. Mao, "Threat of adversarial attacks on DL-based IoT device identification," *IEEE Internet Things J.*, vol. 9, no. 11, pp. 9012–9024, Jun. 2022.

[36] N. Carlini and D. Wagner, "Defensive distillation is not robust to adversarial examples," 2016, *arXiv:1607.04311*.

[37] J. H. Cho and B. Hariharan, "On the efficacy of knowledge distillation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 4793–4801.

[38] T. J. O'Shea, T. Roy, and T. C. Clancy, "Over-the-air deep learning based radio signal classification," *IEEE J. Sel. Topics Signal Process.*, vol. 12, no. 1, pp. 168–179, Feb. 2018.

[39] A. Athalye, N. Carlini, and D. Wagner, "Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 274–283.

[40] Y.-S. Wang, T.-W. Weng, and L. Daniel, "Verification of neural network control policy under persistent adversarial perturbation," 2019, *arXiv:1908.06353*.

[41] C. R. Serrano, P. M. Sylla, and M. A. Warren, "Generate and verify: Semantically meaningful formal analysis of neural network perception systems," 2020, *arXiv:2012.09313*.

[42] D. Gopinath, G. Katz, C. S. Pasareanu, and C. Barrett, "DeepSafe: A data-driven approach for checking adversarial robustness in neural networks," 2017, *arXiv:1710.00486*.

[43] G. Katz, C. Barrett, D. L. Dill, K. Julian, and M. J. Kochenderfer, "Reluplex: An efficient SMT solver for verifying deep neural networks," in *Proc. Int. Conf. Comput. Aided Verificat.*, 2017, pp. 97–117.

[44] T. J. O'Shea, J. Corgan, and T. C. Clancy, "Convolutional radio modulation recognition networks," in *Proc. Int. Conf. Eng. Appl. Neural Netw.*, 2016, pp. 213–226.

[45] N. E. West and T. O'Shea, "Deep architectures for modulation recognition," in *Proc. IEEE Int. Symp. Dynamic Spectr. Access Netw. (DySPAN)*, 2017, pp. 1–6.

[46] N. S. Alagha and P. Kabal, "Generalized raised-cosine filters," *IEEE Trans. Commun.*, vol. 47, no. 7, pp. 989–997, Jul. 1999.

[47] I. W. Selesnick and C. S. Burrus, "Generalized digital butterworth filter design," *IEEE Trans. Signal Process.*, vol. 46, no. 6, pp. 1688–1694, Jun. 1998.

[48] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*.

[49] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks," in *Proc. 14th Int. Conf. Artif. Intell. Statist.*, 2011, pp. 315–323.

[50] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 1026–1034.

[51] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," 2012, *arXiv:1207.0580*.

[52] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.

[53] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 448–456.

[54] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, "Inception-v4, inception-resnet and the impact of residual connections on learning," in *Proc. 31st AAAI Conf. Artif. Intell.*, 2017, pp. 4278–4284.

[55] D. Wu, Y. Wang, S.-T. Xia, J. Bailey, and X. Ma, "Skip connections matter: On the transferability of adversarial examples generated with resnets," 2020, *arXiv:2002.05990*.

[56] T. Na, J. H. Ko, and S. Mukhopadhyay, "Cascade adversarial machine learning regularized with a unified embedding," 2017, *arXiv:1708.02582*.

[57] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 2818–2826.

[58] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2009, pp. 248–255.

[59] A. Shafahi et al., "Adversarial training for free!" 2019, *arXiv:1904.12843*.

[60] A. Krizhevsky. "Learning multiple layers of features from tiny images." 2009. [Online]. Available: http://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf

[61] H. V. Trees, *Optimum Array Processing: Part IV of Detection, Estimation, and Modulation Theory*, (Detection, Estimation, and Modulation Theory). Hoboken, NJ, USA: Wiley, 2004, pp. 1–1433. [Online]. Available: https://books.google.com/books?id=K5XJC_fMMAwC

[62] H. L. V. Trees, *Detection, Estimation, and Modulation Theory: Radar-Sonar Signal Processing and Gaussian Signals in Noise*. Melbourne, FL, USA: Krieger Publ. Co., Inc., 1992.

[63] C. Park, J. Choi, S. Nah, W. Jang, and D. Y. Kim, "Automatic modulation recognition of digital signals using wavelet features and SVM," in *Proc. 10th Int. Conf. Adv. Commun. Technol.*, vol. 1, 2008, pp. 387–390.

[64] A. K. Nandi and E. E. Azzouz, "Algorithms for automatic modulation recognition of communication signals," *IEEE Trans. Commun.*, vol. 46, no. 4, pp. 431–436, Apr. 1998.

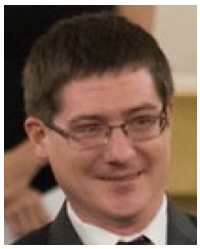
KYLE W. McCLINTICK (Student Member, IEEE) received the B.S. degree in electrical and computer engineering from the Rose-Hulman Institute of Technology in 2017, and the M.S. degree in electrical and computer engineering from Worcester Polytechnic Institute in 2019, with a thesis on supervised learning in wireless communications funded by The MITRE Corporation. He is currently pursuing the Ph.D. degree with the Department of Electrical and Computer Engineering with a data science and computer science minor, generously funded by MIT Lincoln Laboratory. He has been awarded the Best Paper Award for his GlobalSIP 2019 paper, and was awarded the 2019 SMART Ph.D. Fellowship, sponsored by NAWCWD Point Mugu, CA, USA.

JACOB HARER (Member, IEEE) received the Ph.D. degree in computer science from Boston University in 2019, where he focused on Machine Learning for time series data. He is a member of the Technical Staff with MIT Lincoln laboratory. He is part of the Cyber Physical Group with Lincoln, and focuses his research on RF classification/generation, side channel analysis, and anomaly detection.

BRYSE FLOWERS (Student Member, IEEE) received the B.S. and M.S. degrees in computer engineering from Virginia Tech, Blacksburg, VA, USA, in 2014 and 2019, respectively. He is currently pursuing the Ph.D. degree in computer engineering with the University of California at San Diego, San Diego, CA, USA.

In 2013 and 2014, he interned as an Engineer with Qualcomm. From 2015 to 2017, he was an Engineer with Qualcomm, San Diego, CA, USA, where his work focused on multi-networking for applications, such as VoWifi, hardware accelerated protocol processing, and intelligent network selection. In 2018 and 2020, he interned with MIT Lincoln Laboratory and HRL Laboratories. His research interests span the intersection of digital signal processing and machine learning.

Mr. Flowers was awarded the Bradley Masters Fellowship by the Bradley Department of Electrical and Computer Engineering, Virginia Tech, in 2017. In 2018, he received the Hume Graduate Recruiting Fellowship by the Hume Center for National Security and Technology, the Association of Old Crows (AOC) Electronic Warfare Scholarship by the AOC Capitol Club, and was named a Collins Aerospace (formerly UTC Aerospace Systems) Scholar. In 2019, he was awarded the Powell Fellowship by the Jacobs School of Engineering, UCSD.



WILLIAM C. HEADLEY (Senior Member, IEEE) received the B.S., M.S., and Ph.D. degrees in electrical engineering from Virginia Tech. He is the Associate Director for the Electronics Systems Lab with VT Hume Center, where he has served as a Principal or a Co-Principal Investigator on a multitude of government and commercial projects totaling over $12M. Within the Lab, he primarily oversees the Radio Frequency Machine Learning portfolio which is at the forefront of this emerging field. Through his courtesy appointment within Virginia Tech's Electrical and Computer Engineering Department, he also serves as a Mentor and an Advisor to both undergraduate and graduate student researchers, providing them with hands-on research opportunities through these projects as well as guiding them toward their degree requirements. He has written over 30 conference/journal publications in areas, such as spectrum sensing, radio frequency machine learning, and wireless communication educational opportunities.

ALEXANDER M. WYGLINSKI (Senior Member, IEEE) received the B.Eng. degree in electrical engineering from McGill University, Montreal, Canada, in 1999, the M.Sc. degree (Eng.) in electrical engineering from Queen's University, Kingston, Canada, in 2000, and the Ph.D. degree in electrical engineering from McGill University in 2005. He is the Associate Dean of Graduate Studies and a Professor of Electrical and Computer Engineering with Worcester Polytechnic Institute (WPI), Worcester, Mass, USA, where he is also the Director of Wireless Innovation Laboratory. He has published over 48 peer-reviewed journal papers, over 125 peer-reviewed conference papers, and 3 textbooks throughout his academic career. His current research interests are in wireless communications, cognitive radio, machine learning for wireless systems, software defined radio prototyping, connected and autonomous vehicles, and dynamic spectrum sensing. He has been sponsored by both Government Agencies and Industry, such as the National Science Foundation, Office of Naval Research, Air Force Research Laboratory, MIT Lincoln Laboratory, Toyota InfoTechnology Center USA, Verizon, MITRE, Analog Devices, and Raytheon. He served as the President of the IEEE Vehicular Technology Society from 2018 to 2019.