Adversarial Filters for Secure Modulation Classification

Alex Berian, Kory Staab, Gregory Ditzler, Tamal Bose, Ravi Tandon
Department of Electrical and Computer Engineering
University of Arizona, Tucson, Arizona, USA
{*berian, kstaab, ditzler, tbose, tandonr*}@email.arizona.edu

***Abstract*—Classification (MC) is the problem of classifying the modulation format of a wireless signal. In the wireless communications pipeline, MC is the first operation performed on the received signal and is critical for reliable decoding. This paper considers the problem of secure MC, where a transmitter (Alice) wants to maximize MC accuracy at a legitimate receiver (Bob) while minimizing MC accuracy at an eavesdropper (Eve). This work introduces novel adversarial learning techniques for secure MC. We present adversarial filters in which Alice uses a carefully designed adversarial filter to mask the transmitted signal, that can maximize MC accuracy at Bob while minimizing MC accuracy at Eve. We present two filtering-based algorithms, namely gradient ascent filter (GAF), and a fast gradient filter method (FGFM), with varying levels of complexity. Our proposed adversarial filtering-based approaches significantly outperform additive adversarial perturbations (used in the traditional machine-learning (ML) community and other prior works on secure MC) and have several other desirable properties. In particular, GAF and FGFM algorithms are a) computational efficient (allow fast decoding at Bob), b) power-efficient (do not require excessive transmit power at Alice); and c) SNR efficient (i.e., perform well even at low SNR values at Bob).**

I. Introduction

In recent years, machine learning (ML), particularly neural networks (NNs), has shown great promise in many applications and has even surpassed humans in image classification [1]. In the domain of wireless communication, ML has been adapted for many applications [2]–[4]. One important use of ML in wireless communication is modulation classification (MC), where ML models are used to classify the modulation format of signals [5]. In MC, it is common to use extracted features for classification, however, there has been recent work showing that the in-phase/quadrature (IQ) samples passed directly into a NN can achieve competitive classification accuracy [6]–[8].

Despite the benefits of using ML for classification, Adversarial Learning (AL) shows that these algorithms are susceptible to *adversarial examples*, which are typically created by strategically crafting additive perturbations to add to input samples [9]–[12]. A Universal Adversarial Perturbation (UAP) is a perturbation that can create an adversarial example from any input to a fixed ML model [13]. It has also been shown that AL attacks can have detrimental effects on MC [14], [15].

In this paper, we consider an eavesdropper scenario where a transmitter (Alice), has an intended receiver (Bob), and there is an eavesdropper (Eve) listening in on the transmission. The communication system uses blind modulation, so both Bob and Eve need to classify the modulation format before they can decode information. There exist applications where decoding may not be desired by Eve, and simply classifying the modulation format is sufficient for the identification of other parties. Encrypting the data before transmission does not impair Eve's ability to classify the modulation format of a signal. It is, therefore, necessary to use physical layer approaches to prevent Eve's ability to classify the modulation format.

We consider a scenario where Alice uses AL to send adversarially perturbed signals instead of an unperturbed signal. Bob is assumed to already have a securely pre-shared key before communication to help him undo the adversarial attack, and he has a minimum signal-to-noise ratio (SNR) constraint to ensure decodability. The secrecy of the scheme is measured in terms of Eve's classification accuracy which must remain low.

In this paper, we present a filtering-based framework to create adversarial examples instead of additive perturbations. Filter-based adversarial attacks do not suffer from drawbacks that additive perturbations face in this eavesdropper scenario. It is much easier for Bob to undo the effect of the adversarial filter as opposed to additive adversarial perturbations.

The main contributions of this paper are as follows. Two novel methods of creating adversarial filters are proposed. The first approach is an iterative optimization technique where filter taps are effectively trained: we call this approach the gradient ascent filter (GAF). Two methods to create a GAF that has a stable inverse are proposed. The second adversarial filter is an approximate analytical solution to the optimization problem of maximizing loss with respect to the filter's taps. The proposed approaches are for finite impulse response (FIR) linear filters, where the inverse filters are infinite impulse response (IIR) filters that can be easily solved. This paper analyses an eavesdropper scenario for MC, where secrecy is measured by the reduction in classification accuracy at the eavesdropper by applying adversarial attacks at the transmitter. We also present simulation results to compare traditional additive perturbations versus the proposed adversarial filtering in terms of classification accuracy at the eavesdropper and the intended receiver, available transmit power at the transmitter, and SNR requirement at the intended receiver.

II. System Model & Problem Statement

Consider The system model shown in Figure 1, where Alice wants to send a signal $\vec{s}$ to Bob which is a $[d \times 1]$ vector of complex values ($\vec{s} \in \mathbb{C}^d$). There are C different modulation classes with which Alice may choose to transmit information to Bob. $\vec{s}$ is a signal with modulation

This project was partially supported by the Department of Energy (DOE) #DE-NA0003946, the Broadband Wireless Access and Applications Center (BWAC), NSF Award #1822071, and NSF Career #1943552

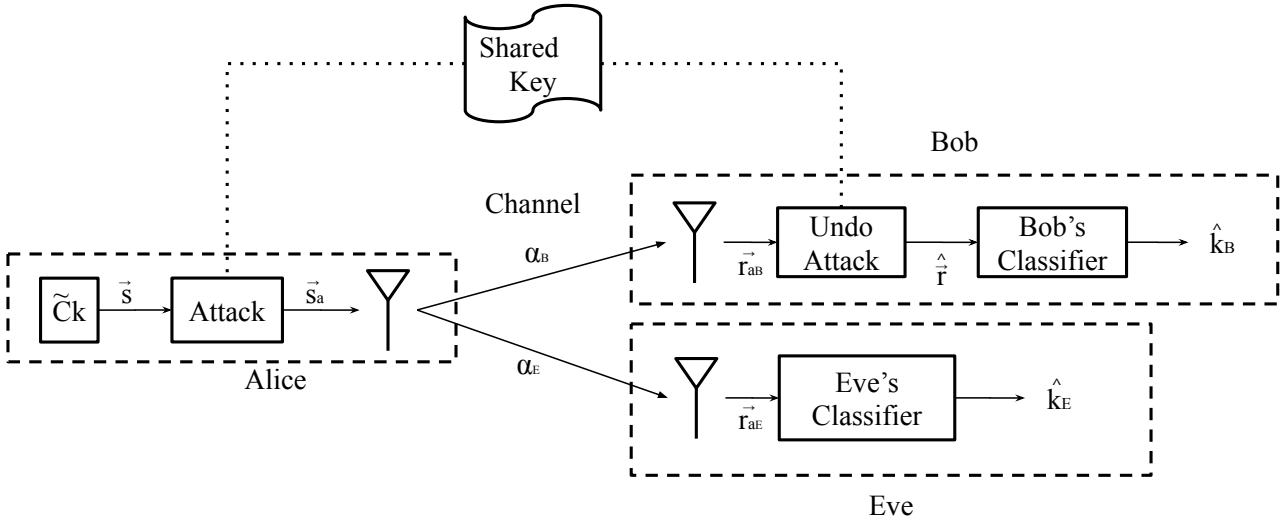

Fig. 1. Assumed Communication system model with Transmitter (Alice), receiver (Bob), and eavesdropper (Eve)

format k, and $\tilde{C}_k$ is the set of all signals from class k (i.e., $\vec{s} \in \tilde{C}_k$).

The communication channel from Alice to Bob and Alice to Eve are assumed to be Additive White Gaussian Noise (AWGN) channels with a fixed noise power P_N and an attenuation of α_B and α_E respectively. The received signal by Bob $\vec{r}_B$ and Eve $\vec{r}_E$ when Alice send $\vec{s}$ are given by

$$\vec{r}_B = \alpha_B \vec{s} + \vec{N}, \tag{1}$$

$$\vec{r}_E = \alpha_E \vec{s} + \vec{N}, \tag{2}$$

where α_B, α_E are the real-valued positive scalar channel attenuation for Bob and Eve's channels respectively, and $\vec{N}$ is a complex AWGN vector with zero mean.

Eve can predict the modulation class k from $\vec{r}_E$. Alice wants to keep k secure from Eve. Using AL, Alice creates an adversarial example $\vec{s}_a$ from $\vec{s}$. Using some adversarial attack function f with an $[m \times 1]$ vector of parameters $\vec{\delta}$,

$$\vec{s}_a = f(\vec{s}, \vec{\delta}). \tag{3}$$

Alice transmits $\vec{s}_a$ and has a finite transmission power P_T, this introduces the constraint

$$\|\vec{s}_a\|^2 \leq P_T, \tag{4}$$

where $\|\vec{v}\|$ denoted the L2 norm. When Alice sends $\vec{s}_a$ instead of $\vec{s}$, The received signals by Bob and Eve are given by

$$\vec{r}_{aB} = \alpha_B \vec{s}_a + \vec{N}, \tag{5}$$

$$\vec{r}_{aE} = \alpha_E \vec{s}_a + \vec{N}A \tag{6}$$

Alice and Bob are assumed to have a securely pre-shared key. In this scenario, the key is assumed to already have been securely shared between both Alice and Bob with no overhead. The key in this problem is the vector of parameters $\vec{\delta}$ Bob uses this key to try to undo the adversarial attack to get $\hat{\vec{r}}_B$ which is his estimation of $\vec{r}_B$. Bob then passes $\hat{\vec{r}}_B$ into his classifier h_B to get an estimate $\hat{k}_B$ of true modulation format k:

$$\hat{k}_B = h_B(\hat{\vec{r}}_B). \tag{7}$$

Once Bob chooses the modulation format $\hat{k}_B$, he can use the appropriate demodulation scheme. $\hat{\vec{r}}_B$ must satisfy a minimum SNR for reliable communication. Eve does not have this key. Eve's classification accuracy is the probability that her classifier h_E acting on her received signal outputs the correct class:

$$P_E = Pr\{h_E(\vec{r}_{aE}) = k\} = Pr\{\hat{k}_E = k\}. \tag{8}$$

Since Bob has his estimate $\hat{\vec{r}}_B$, Bob's classification accuracy is

$$P_B = Pr\{h_B(\hat{\vec{r}}) = k\} = Pr\{\hat{k}_B = k\}. \tag{9}$$

For additive attacks, the attack function in (3) is defined as

$$f_+(\vec{s}, \vec{\delta}_+) = \vec{s} + \vec{\delta}_+, \tag{10}$$

and $m = d$ to satisfy dimensionality constraints. For additive attacks δ is denoted by $\vec{\delta}_+$, which is the additive perturbation in additive attacks. In a filter-based attack, (3) can be rewritten as follows:

$$f_\circledast(\vec{s}, \vec{\delta}_\circledast) = \vec{s} \circledast \vec{\delta}_\circledast, \tag{11}$$

and $m \neq d$, $\vec{\delta}_\circledast$ denotes the taps of the FIR filter that generates the adversarial sample, and $\circledast$ is convolution between two finite vectors. This work analyses the performance of additive and linear filter forms of the adversarial attack $f(\vec{s}, \vec{\delta})$.

To correctly classify the received signal and decode, Bob must reverse the adversarial attack from $\vec{r}_{aB}$ to get $\hat{\vec{r}}_B$. That is

$$\hat{\vec{r}} = f^{-1}(\vec{r}_{AB}, \vec{\delta}). \tag{12}$$

The formulation of $f^{-1}(\vec{r}_{aB}, \vec{\delta})$ depends on how the key is shared between Alice and Bob. The key is never altered, nor shared during communication between Alice and Bob. The function $f(\cdot, \cdot)$ is known by both Bob and Eve.

Alice and Bob's goal is to design an adversarial attack and shared key that keeps Eve's classification accuracy P_E low and Bob's classification accuracy P_B high while satisfying the minimum SNR constraint of recovered signal $\hat{\vec{r}}$.

III. Main Results & Discussion

A. Reversing Adversarial Attacks at Bob

When Bob reverses an additive attack, he subtracts $\vec{\delta}_+$ from $\vec{r}_{aB}$ with the proper attenuation. However, Bob cannot perfectly find α_B, so he must make an estimate $\hat{\alpha}_B$ of the

true α_B. This results in the following expression of Bob's recovered signal:

$$\hat{\vec{r}}_{+B} = \alpha_B \vec{s} + \vec{N} + (\alpha_B - \hat{\alpha}_B)\vec{\delta}_+. \tag{13}$$

From (13) we see that unless Bob perfectly estimates $\vec{\alpha}$, there will always be remnants of the adversarial perturbation $\vec{\delta}_+$. This is particularly bad for Bob because many AL algorithms that generate $\vec{\delta}_+$ are designed to create perturbations that ruin classification accuracy even when the power assigned to the perturbation is extremely small [9], [10], [16]. In real-time, Alice must repeatedly retransmit the finite perturbation. For Bob to undo this attack he must sync his removal of $\vec{\delta}_+$ with Alice's repeated transmission.

Reversing an FIR linear filter attack is significantly simpler, as there is no need to estimate the attenuation α_B, or synchronize the additive perturbation. This paper focuses on FIR filters where $\delta_{\circledast}^{-1}$ is the impulse response of the inverse filter to $\vec{\delta}_{\circledast}$. The recovered signal under a filter-based attack is

$$\hat{\vec{r}}_{\circledast B} = \vec{\delta}_{\circledast B}^{-1} * \vec{r}_{aB} = \alpha_B \vec{s} + \vec{\delta}_{\circledast}^{-1} * \vec{N}. \tag{14}$$

Therefore, when Bob undoes the filtering-based attack, he colors the noise.

Figure 2 shows the SNR of the recovered signals at Bob when additive and filter attacks are used. The recovered signal SNR can be expressed as a function of transmit power P_T, perturbation power $P_{\vec{\delta}}$, the channel attenuation coefficient α_B, and the noise power $P_{\vec{N}}$. Note that $P_{\vec{\delta}}$ is only applicable if an additive perturbation is used. Assuming Bob perfectly eliminates the additive perturbation, the SNR of the recovered signal $\hat{\vec{r}}_{+B}$, expressed in (13), for additive perturbations is given as

$$SNR_+ = \frac{\alpha_B^2 P_T - P_{\vec{\delta}}}{P_{\vec{N}}}. \tag{15}$$

As for filter-based adversarial attacks used by Alice, there is no need to allocate transmit power to a perturbation, so the SNR at Bob $SNR_{\circledast}$ for filter attacks is

$$SNR_{\circledast} = \frac{\alpha_B^2 P_T}{P_{\vec{N}}}. \tag{16}$$

Alice need not exceed the SNR requirement at Bob. For additive attacks, Alice can allocate just enough power to $P_{\vec{s}}$ to satisfy the SNR requirement, and allocate the rest of P_T to $P_{\vec{\delta}}$ to minimize Eve's classification accuracy. If there is barely enough transmit power to satisfy the SNR requirement, little to no power can be allocated to the perturbation. When P_δ is very small, Eve can achieve a high classification accuracy. An adversarial filter will not suffer from this drawback.

Bob needs to reverse the adversarial attack. Alice cannot pre-share every possible $\vec{\delta}_j$ for every possible signal $\vec{s}_j$ she sends. Therefore, it is critical that Alice uses a UAP that will work for all $\vec{s}_j$. In addition to attacks being universal, they can also be class-specific. Instead of $\vec{\delta}$ being designed to work on every input $\vec{s}_j \in \tilde{C}$, $\vec{\delta}^{(k)}$ can be designed to work especially well for inputs from the k^{th} class $\vec{s}_j \in \tilde{C}_k$. There are only a finite number of classes, so it is feasible for Alice to pre-share every $\vec{\delta}^{(k)}$ with Bob.

One important aspect regarding undoing the filter at Bob is the invertibility of the filter used by Alice. If the inverse filter at Bob $\vec{\delta}_{\circledast}^{-1}$ is unstable, then the noise at Bob's receiver will be amplified and no communication will be possible. All the zeros of the filter $\vec{\delta}_{\circledast}$ must be outside the unit circle for $\vec{\delta}_{\circledast}^{-1}$ to be stable.

The filters used at both Bob and Alice must have an overall gain of 1 to ensure that signal power is preserved.

$$\frac{1}{2\pi}\int_{-\pi}^{\pi} |D_{\circledast}(\omega)|^2 d\omega = 1, \tag{17}$$

where $D_{\circledast}(\omega)$ is the Discrete-Time-Fourier-Transform (DTFT) of the filter $\vec{\delta}_{\circledast}$. Using Parseval's theorem, the constraint in (17) can be rewritten as the L2 norm of the magnitude:

$$\|abs(\vec{\delta}_{\circledast})\|^2 = 1. \tag{18}$$

Here the magnitude operation $abs(\cdot)$ is element-wise for the complex vector within.

B. Classification Accuracy and Secrecy

A common metric for secrecy is the mutual information between recovered information by the eavesdropper and information for the intended receiver. In this problem, the information is the modulation class k, and Eve's guess $\hat{k}_E$.

We can treat the true modulation class as a random variable K, and Eve's recovery as another random variable $\hat{K}_E$. Let us assume K has an equal probability of being any of the C possible modulation formats. Using Fano's inequality [17], we can make the lower-bound on mutual information

$$I(K; \hat{K}_E) \geq log(C) - H(P_E) - P_E log(C), \tag{19}$$

where $H(P_E)$ is the binary entropy function for Eve's probability of error $P_E = P(\hat{K}_E \neq K)$. From (19) it is clear that by increasing P_E Alice improves the secrecy of her communication to Bob, therefore Eve's classification accuracy is a si metric for secrecy.

IV. Adversarial Filtering Algorithms

A. Gradient Ascent Filter

The first novel AL algorithm in this work is the gradient ascent filter (GAF). GAF can be summarized as an optimization approach where a seed vector $\vec{\Delta}$ is trained such that the adversarial filter $\vec{\delta} = G(\Delta)$ empirically lowers classification accuracy on a targeted ML model. The seed vector $\vec{\Delta}$ of size $[l \times 1]$ is randomly initialized, then converted into a the adversarial filter $\vec{\delta}$ of size $[m \times 1]$ using some filter generation function $\vec{\delta} = G(\vec{\Delta})$. The desired number of filter taps m and the generation function G determine the size of the seed vector l.

During optimization, the seed vector $\vec{\Delta}$ is updated iteratively to increase loss as shown in Figure 3. The seed vector $\vec{\Delta}$ is trained with any optimization process as desired (i.e., Adagrad, RMS prop, Adam [18]). The formal algorithm for training the GAF with stochastic gradient ascent is formally described in algorithm 1.

B. GAF variants

This subsection proposes three filters for GAF. All three proposed filters ensure that the power preserving constraint in (18) is satisfied. Of the three presented filters, only the First Tap Constrained GAF and the Root Training GAF have a guaranteed stable inverse. Although their inverse may not be finite, there is a tuneable parameter β that can guarantee it decreases significantly and can effectively be finite in practice.

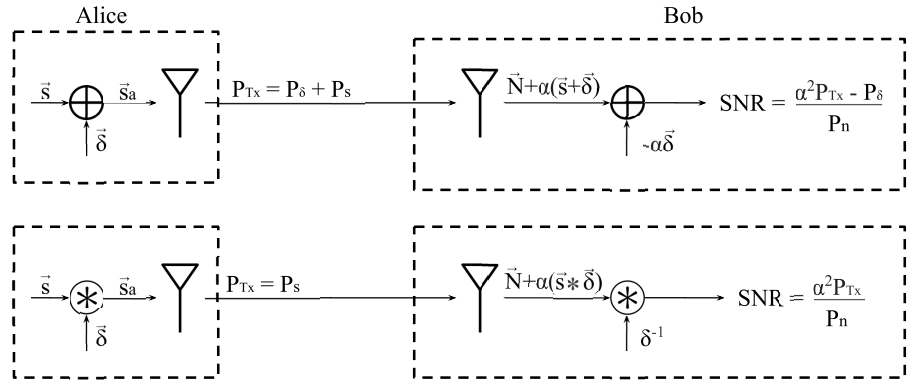

Fig. 2. Illustration of how Alice and Bob use AL learning in their communication system. We see that there is a loss in SNR at Bob when an additive perturbation is employed compared to filtering based attacks

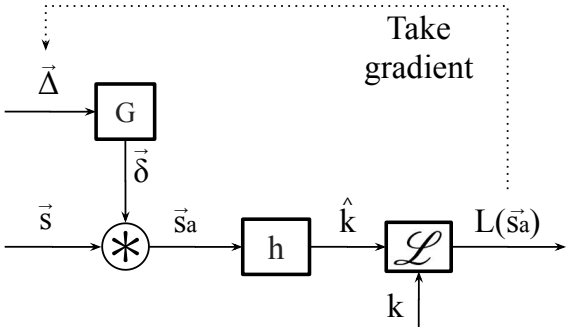

Fig. 3. Flowgraph for Training Gradient Ascent Filter (GAF).

Algorithm 1: Algorithm for creating $\vec{\Delta}$ using the GAF technique

Inputs:

- number of filter taps m
- filter generation function G
- model h
- number of training epochs T
- learn rate η
- loss function $\mathscr{L}$
- training inputs set $\boldsymbol{s} = \{\vec{s}_i : \forall i\}$
- training labels set $\boldsymbol{k} = \{k_i : \forall i\}$

Output:

- The adversarial filter $\vec{\delta}_{GAF}$

Initialize $\vec{\Delta}_0 \sim \mathcal{N}(0^{[1\times l]}, I^{[l\times l]})$
for $t = 1$ to T **do**
  $\vec{\delta} = G(\vec{\Delta}_{t-1})$
  $\boldsymbol{s}_a = \{\vec{s}_{a_i} : \vec{s}_{a_i} = \vec{s}_i \circledast \vec{\delta}, \forall i\}$
  $\hat{\boldsymbol{k}} = \{\hat{k}_i : \hat{k}_i = h(\vec{s}_{a_i}), \forall i\}$
  $\boldsymbol{L} = \{L_i : \mathscr{L}(\hat{k}_i, k_i), \forall i\}$
  $L = \sum_{\forall i} L_i$
  $\vec{\Delta}_t = \vec{\Delta}_{t-1} + \eta \nabla_{\vec{\Delta}_{t-1}} L$
end for
return $\vec{\delta}_{GAF} = G(\vec{\Delta}_T)$

1) Unconstrained GAF: One approach to create GAF is to treat $\vec{\Delta}$ as an unconstrained filter, and set $\vec{\delta}$ to be a power preserving version of $\vec{\Delta}$. The unconstrained GAF generation function is given by

$$\vec{\delta} = G_u(\vec{\Delta}) = \frac{\vec{\Delta}}{\|abs(\vec{\Delta})\|}, \tag{20}$$

where $\vec{\Delta}$ is complex valued, and $l = m$ for dimensionality. The unconstrained GAF is named so, because this filter has no constraints on the filter taps besides power preservation, and it does not guarantee an FIR filter with a stable inverse.

2) First Tap Constrained GAF: Cauchy's argument principle states that if $D(\omega)$ does not wrap around the origin in the complex plane, then $\vec{\delta}$ has a stable inverse. One way to ensure that $D(\omega)$ does not wrap around the origin it to ensure it's real part is always greater than 0. The proposed approach to satisfy this constraint is to set the taps of an intermediate filter $\vec{h}_{ftc}$ be

$$\vec{h}_{ftc,k} = \begin{cases} \beta + 1 & k = 0 \\ \frac{\vec{\Delta}_{k-1}}{\sum_{i=0}^{m-2} abs(\vec{\Delta}_i)} & k \in [1, m-1] \end{cases}, \tag{21}$$

where $\beta > 0$ is a real constant to ensure stability as opposed to marginal stability, and $l = m - 1$. Now the filter generation function for the first tap constrained GAF is

$$\delta = G_{ftc}(\vec{\Delta}) = \frac{\vec{h}_{ftc}}{\|abs(\vec{h}_{ftc})\|}. \tag{22}$$

3) Root Training GAF: The third approach to creating a GAF with a stable inverse seeks to treat $\vec{\Delta}$ as a vector of zeros of the adversarial filter. The full pipeline for the root training GAF filter is shown in Figure 4. Of course, during optimization, the values in $\vec{\Delta}$ are unpredictable, so a function (far left block in Figure 4) for creating a vector of zeros of an invertible FIR filter from a vector with any values is used. Vieta's formula [19] is used to expand the polynomial described by the filter's zeros, and get the filter taps. Finally, the filter taps are normalized so that the filter preserves power. For more detail on the root training GAF, we refer the reader to [20].

C. Fast Gradient Filter Method

The second novel filtering AL algorithm presented in this paper is the fast gradient filter method (FGFM). This algorithm is similar to the Fast gradient method (FGM) because it is based on choosing $\vec{\delta}$ to take a small step in the direction of the gradient with respect to loss. Loss $L(\vec{s})$

Fig. 4. Flowgraph of the filter creation function $G_{rt}(\vec{\Delta})$ for the root training GAF.

for a particular input is the differentiable metric for how wrong the trained classifier h is when observing $\vec{s}$. Like the FGM, the FGFM also creates an input specific $\vec{\delta}$.

The adversarial filter created by the FGFM is

$$\vec{\delta}_{FGFM} = \epsilon\vec{\delta} + \vec{v}, \tag{23}$$

where $\vec{v}$ is an $[m \times 1]$ vector of zeros with a 1 in the center. In the case of $m = 3$, $\vec{v} = [0, 1, 0]^T$. ϵ is a real scalar which determines how much the filter alters the input signal $(\vec{s})$. In the case where $m = 3$ and $d = 5$, $\vec{\delta}$ is given by

$$\vec{\delta} = \epsilon \begin{bmatrix} \vec{s}_1 & \vec{s}_2 & \vec{s}_3 & \vec{s}_4 & 0 \\ \vec{s}_0 & \vec{s}_1 & \vec{s}_2 & \vec{s}_3 & \vec{s}_4 \\ 0 & \vec{s}_0 & \vec{s}_1 & \vec{s}_2 & \vec{s}_3 \end{bmatrix} (\nabla_{\vec{s}} L(\vec{s})). \tag{24}$$

(24) is specific to $m = 3$ and $d = 5$, however, the form of $\vec{\delta}$ generalizes for other values of d and m. The structure of Toeplitz matrix depends on how the vector convolution operation $\circledast$ is defined, but its dimensionality is always $[d \times m]$. For further details on the derivation for the FGFM, we refer the reader to [20].

The $\vec{\delta}_{FGFM}$ is specifically designed for a particular input $\vec{s}$. As previously stated in Section III, a universal attack is needed for Alice to effectively communicate with Bob. One can simply use existing UAP algorithms to aggregate many filters generated by the FGFM into a universal filter. It is important to note that the FGFM may not have a stable inverse, hence Bob may not be able to get Alice's signal back.

V. Experiments

In this section, we present simulation results to show the efficacy of the proposed methods and compare them with existing ones. All experiments were conducted purely in simulation using Python. The TensorFlow and Keras libraries were used as the platform for implementing DL algorithms with graphics processing unit (GPU) acceleration. Many useful functions from the SciPy and Matplotlib libraries were used. The ML algorithm analyzed in these simulations is a deep convolutional neural network with inputs that have normalized power and mean. There are 4 convolutional layers with varying filter sizes smaller than $[7 \times 1]$, and each convolutional layer has 128 output channels. Following the convolutional layers are two dense layers with outputs of size 256, and C respectively. The same DL architecture was used for both Bob and Eve. Although this is not a practical assumption, it is shown in many AL papers that an adversarial attack that works on one DL model will usually work on another similar deep learning model. Only white-box attacks were considered, however it has been shown in many AL works that AL attacks are transferable between DL models. The channel uses trivial attenuation $\alpha = 1$. To implement the FGFM, Fast gradient method (FGM), and Fast gradient sign method (FGSM) as universal or modulation specific attacks, the PCA-based algorithm for creating UAP's from Sadeghi et al was used [14]. The FGM and the FGSM are the only additive attacks from prior works we compare the filter attacks with. All adversarial filters in these experiments have 5 taps. Classification accuracy in these experiments is given by averaging the probability of correct classification over all classes. A dataset of clean waveforms was created for this experiment using Python simulations. The waveforms to be classified all have 4 samples per symbol, and we consider $C = 8$ where the modulation classes 8FSK, 2FSK, 16QAM, 64QAM, 4PAM, QPSK, 8PSK, BPSK.

Figure 5 compares two additive attacks (FGM and FGSM) with the root training GAF (labeled as rtGAF) and FGFM. The minimum SNR requirement at Bob is -10 dB. In this experiment, it is assumed that Bob can estimate the attenuation of the channel α perfectly, so the plot shows that Bob's classification accuracy does not drop in additive attacks. The lowest Tx power shown in the plots indicates that P_T is just enough to satisfy the SNR requirement at Bob. This means that additive attacks such as the FGSM or FGM cannot have any power allocated to them (i.e., $P_T = P_{\vec{s}} + P_{\vec{\delta}}$ and $P_{\vec{\delta}} = 0$). Eve's classification accuracy is equal to Bob's classification accuracy when P_T is very low because of this. We observe that with very little extra transmit power at Alice, the FGSM lowers Eve's classification abilities significantly. This is because the FGSM is optimal in the sense that it guarantees misclassification with a minimum infinite norm of the perturbation. There is a short-lived increase in classification accuracy for the FGM likely because the direction of the FGM perturbation has overstepped a local minimum in the loss space. The root training GAF (labeled as rtGAF) reduces Eve's classification accuracy at any P_T because there is no need to dedicate transmit power for the attack. Bob's classification accuracy remains constant under the additive attacks because Alice is allocating as much transmit power to the perturbation as possible. Bob's classification accuracy raises with Alice's transmit power when filter attacks are used because Alice can allocate more power to her signal and Bob can almost perfectly recover the un-attacked signal.

Figure 6 compares Eve's and Bob's classification accuracy when Alice uses different filtering attacks proposed in this paper. The FGFM produces a very effective filter that fools Eve; however, its inverse is unstable. Therefore, Bob's classification accuracy suffers as well. The unconstrained GAF (labeled as uGAF) was the most effective at fooling Eve, due to being able to optimize filter taps unconstrained, but its inverse is even more unstable than the FGFM which causes Bob's classification accuracy to be worse. The first tap constrained GAF (labeled as ftcGAF) has a stable inverse, so Bob's classification accuracy is good. However, the first tap constrained GAF retains a large portion of Alice's original signal because of the large real-valued first tap, so Eve is not fooled effectively. β for the first tap constrained GAF was 0.9 for this experiment. The root training GAF (labeled as rtGAF) also has a stable inverse, and it is significantly less constrained than the first tap constrained GAF, so it is intuitive that Eve's classification accuracy is very low with this filter. β for the root training GAF was 0.5 for this experiment.

Figure 7 shows an experimental comparison of Eve/Bob's classification accuracy when Alice employs universal versions versus modulation specific versions of

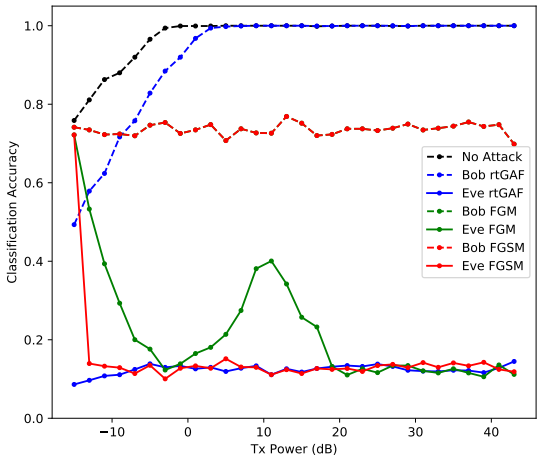

Fig. 5. The relationship between Alice's transmit power and Eve/Bob's classification accuracy when different attack are applied by Alice and Bob. The minimum SNR requirement at Bob is -10dB, and the noise power is 5dB.

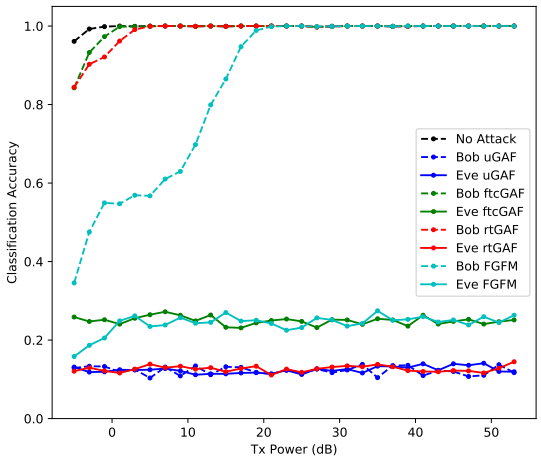

Fig. 6. The relationship between Alice's transmit power and Eve/Bob's classification accuracy when different filtering attack are applied by Alice and Bob. The minimum SNR requirement at Bob is 0dB, and the noise power is 5dB.

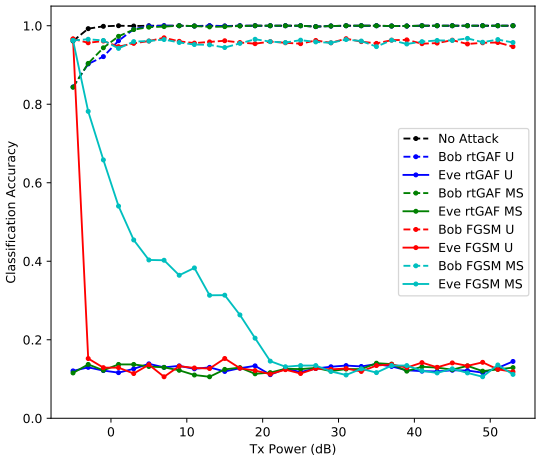

Fig. 7. The relationship between Alice's transmit power and Eve/Bob's classification accuracy when root training GAF and the FGSM applied by Alice and Bob as modulation specific attacks (MS) and universal attacks (U). The minimum SNR requirement at Bob is 0dB, and the noise power is 5dB.

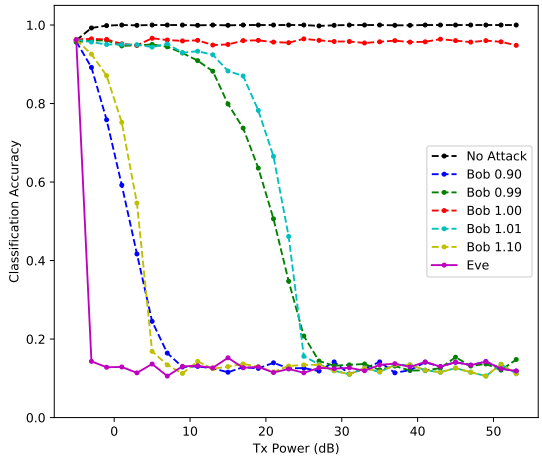

Fig. 8. The importance of Bob estimating the channel attenuation α. The numbers in the legend are the ratio between Bob's channel attenuation estimate and the true channel attenuation $\hat{\alpha}/\alpha$. In this experiment, the SNR requirement at Bob is 0dB, and the noise power is 5dB

the root training GAF and the FGSM. We observe no difference in performance for the root training GAF, but the modulation-specific version of the FGSM is not as secure as the universal.

Figure 8 shows Bob's classification accuracy when his estimate $\hat{\alpha}$ is not perfect, and Alice uses the FGSM. The number used to label each curve is the ratio between Bob's estimate and the true channel attenuation coefficient $\hat{\alpha}/\alpha$. The perfect estimate is when $\hat{\alpha}/\alpha = 1$ where Bob's classification accuracy is highest. As $\hat{\alpha}/\alpha$ deviates further from 1, Bob's classification accuracy drops detrimentally as P_T increases. Eve's classification accuracy is not affected by Bob's inability to estimate α.

VI. Conclusion

This paper examines the security of MC in a point-to-point communication system with an eavesdropper (Eve), where the transmitter (Alice) has finite transmission power, and the intended receiver (Bob) has a minimum SNR requirement that must be met for reliable communication. The channel model explored in this work is an AWGN channel model with attenuation of the transmitted signal. Secrecy is measured in the sense that Eve's classification accuracy is low, Bob's classification accuracy is high. Alice uses an adversarial attack on her transmitted signal to lower Eve's classification accuracy and shares a key with Bob so that he can undo the adversarial attack. Eve does not know the key, so she cannot undo the adversarial attack. This paper shows that additive adversarial attacks fall short in many ways for this application. If Alice uses filters to generate adversarial examples instead of additive perturbations, these shortcomings are mitigated.

This paper also presents two novel filter-based AL algorithms to generate adversarial examples. The first of which is an optimization approach called the gradient ascent filter (GAF). This paper proposes three methods of creating a GAF, two of which have a stable inverse which is necessary for Bob to decode information. The second presented adversarial filter algorithm is similar to the fast gradient method (FGM), called the fast gradient filter method (FGFM). The FGFM does not guarantee a stable inverse. In simulations with a convolutional neural network, the root training GAF was the most effective AL algorithm in this system model.

References

[1] R. Ewerth, M. Springstein, L. A. Phan-Vogtmann, and J. Schütze, ""are machines better than humans in image tagging?"-a user

study adds to the puzzle," in *European Conference on Information Retrieval*. Springer, 2017, pp. 186–198.

[2] Yong-Woon Kim and Dong-Jo Park, "Nonlinear channel equalization using new neural network model," in *IJCNN'99. International Joint Conference on Neural Networks. Proceedings (Cat. No.99CH36339)*, vol. 2, July 1999, pp. 827–830 vol.2.

[3] M. Nazzal, A. R. Ektí, A. Görçín, and H. Arslan, "Exploiting sparsity recovery for compressive spectrum sensing: A machine learning approach," *IEEE Access*, vol. 7, pp. 126 098–126 110, 2019.

[4] A. Irawan, G. Witjaksono, and W. K. Wibowo, "Deep learning for polar codes over flat fading channels," in *2019 International Conference on Artificial Intelligence in Information and Communication (ICAIIC)*, Feb 2019, pp. 488–491.

[5] O. A. Dobre, A. Abdi, Y. Bar-Ness, and W. Su, "Survey of automatic modulation classification techniques: classical approaches and new trends," *IET Communications*, vol. 1, no. 2, pp. 137–156, April 2007.

[6] N. E. West and T. O'Shea, "Deep architectures for modulation recognition," in *2017 IEEE International Symposium on Dynamic Spectrum Access Networks (DySPAN)*, March 2017, pp. 1–6.

[7] G. Vanhoy, N. Thurston, A. Burger, J. Breckenridge, and T. Bose, "Hierarchical modulation classification using deep learning," in *MILCOM 2018 - 2018 IEEE Military Communications Conference (MILCOM)*, Oct 2018, pp. 20–25.

[8] S. Zhou, Z. Yin, Z. Wu, Y. Chen, N. Zhao, and Z. Yang, "A robust modulation classification method using convolutional neural networks," *EURASIP Journal on Advances in Signal Processing*, vol. 2019, no. 1, p. 21, 2019.

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

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

[11] N. Papernot, P. McDaniel, S. Jha, M. Fredrikson, Z. B. Celik, and A. Swami, "The limitations of deep learning in adversarial settings," in *2016 IEEE European symposium on security and privacy (EuroS&P)*. IEEE, 2016, pp. 372–387.

[12] S.-M. Moosavi-Dezfooli, A. Fawzi, and P. Frossard, "Deepfool: a simple and accurate method to fool deep neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2574–2582.

[13] S. Moosavi-Dezfooli, A. Fawzi, O. Fawzi, and P. Frossard, "Universal adversarial perturbations," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017, pp. 86–94.

[14] M. Sadeghi and E. G. Larsson, "Adversarial attacks on deep-learning based radio signal classification," *IEEE Wireless Communications Letters*, vol. 8, no. 1, pp. 213–216, 2018.

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

[16] N. Carlini and D. Wagner, "Towards evaluating the robustness of neural networks," in *2017 IEEE Symposium on Security and Privacy (SP)*, May 2017, pp. 39–57.

[17] T. M. Cover and J. A. Thomas, *Elements of information theory*. Wiley-India, 2010.

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

[19] M. Hazewinkel, *Encyclopaedia of Mathematics: Viète Theorem*. Springer Science & Business Media, 1994.

[20] A. Berian, K. Staab, N. Teku, G. Ditzler, T. Bose, and R. Tandon, "Adversarial filters for secure modulation classification," 2020.