Channel Effects on Surrogate Models of Adversarial Attacks against Wireless Signal Classifiers

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

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

Abstract—**We consider a wireless communication system that consists of a background emitter, a transmitter, and an adversary. The transmitter is equipped with a deep neural network (DNN) classifier for detecting the ongoing transmissions from the background emitter and transmits a signal if the spectrum is idle. Concurrently, the adversary trains its own DNN classifier as the surrogate model by observing the spectrum to detect the ongoing transmissions of the background emitter and generate adversarial attacks to fool the transmitter into misclassifying the channel as idle. This surrogate model may differ from the transmitter's classifier significantly because the adversary and the transmitter experience different channels from the background emitter and therefore their classifiers are trained with different distributions of inputs. This system model may represent a setting where the background emitter is a primary user, the transmitter is a secondary user, and the adversary is trying to fool the secondary user to transmit even though the channel is occupied by the primary user. We consider different topologies to investigate how different surrogate models that are trained by the adversary (depending on the differences in channel effects experienced by the adversary) affect the performance of the adversarial attack. The simulation results show that the surrogate models that are trained with different distributions of channel-induced inputs severely limit the attack performance and indicate that the transferability of adversarial attacks is neither readily available nor straightforward to achieve since surrogate models for wireless applications may significantly differ from the target model depending on channel effects.**

I. Introduction

Deep learning (DL) has been broadly applied to solve complex problems in various domains such as computer vision and speech recognition [1] by learning from and adapting to rich data environments via deep neural networks (DNNs). On the other hand, it is well known that the DNNs are susceptible to the adversarial attacks (also known as evasion attacks) that can fool the DNNs into misclassification by adding small perturbations to the input data samples [2]. When the target DNN model is not readily available to the adversary, it can query the DL system (the target DNN model) with its own input data and use the returned labels to build a surrogate model (another DNN model). Then, the adversary uses this surrogate model to craft adversarial attacks against the target model by relying on transferability of adversarial attacks (namely, the attacks developed using a reliable surrogate model should also work against the target model with high success). In data domains such as computer vision, the adversary has typically the advantage of having (i) control of adding adversarial perturbations directly to the input data and (ii) direct access to training data for the surrogate model that have the same or similar distributions as the training data for the target model.

These two properties do not necessarily hold in wireless communication systems, where DL has found rich applications such as spectrum sensing [3], signal classification [4], and waveform design [5]. Recently, adversarial machine learning has gained attention to understand the emerging attack surface regarding wireless security. Those attacks include exploratory (inference) [6], [7], adversarial (evasion) [8]–[19], poisoning (causative) [20], [21], membership inference [22], Trojan [23], and spoofing [24], [25] attacks.

Adversarial attacks aim to fool a DNN into making misclassification errors by adding adversarial perturbations to the input data. In the wireless domain, it has been shown in [8] that the DNN modulation classifier from [26] is vulnerable to adversarial attacks generated using fast gradient method (FGM) when the adversarial perturbations are transmitted through an additive white Gaussian noise (AWGN) channel. Vulnerabilities of the modulation classifier in the presence of adversarial attacks have been studied in [9]–[13]. Adversarial attacks on the modulation classifier in realistic channel environments such as Rayleigh fading have been studied in [14], [15] by accounting for the adversary's lack of direct control on adversarial perturbations (namely, the adversarial perturbations need to go through complex channel effects before reaching the target classifier at the receiver). In addition to fooling the modulation classifier at one receiver, the requirement has been added in [16] to guarantee that another receiver can still recover the underlying message with high reliability. This concept has been also applied in [17] to enable covert communications without being detected by signal classifiers. Adversarial attacks have been extended in [18] to the use of multiple antennas to craft and transmit perturbations. Adversarial attacks have been also considered against spectrum sensing in [19]–[21], where the attack aims to manipulate the spectrum sensing data collected at the transmitter to make incorrect transmit (spectrum access) decisions using the surrogate DL model built by the adversary.

For adversarial attacks on wireless signal classification, a common assumption has been made that the inputs to

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

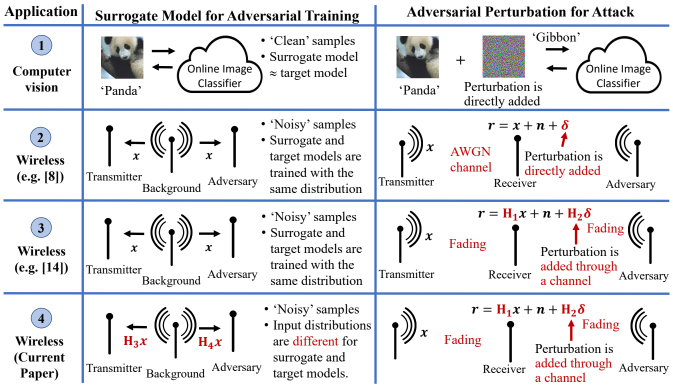
Application
Surrogate Model for Adversarial Training
Adversarial Perturbation for Attack
1
Computer vision
'Panda'
Online Image Classifier
• 'Clean' samples
• Surrogate model ≈ target model
'Panda'
+
Perturbation is directly added
'Gibbon'
Online Image Classifier
2
Wireless (e.g. [8])
Transmitter
Background
Adversary
x
x
• 'Noisy' samples
• Surrogate and target models are trained with the same distribution
$r = x + n + \delta$
Transmitter
AWGN channel
Receiver
Perturbation is directly added
Adversary
3
Wireless (e.g. [14])
Transmitter
Background
Adversary
• 'Noisy' samples
• Surrogate and target models are trained with the same distribution
$r = H_1 x + n + H_2 \delta$
Transmitter
Fading
Receiver
Perturbation is added through a channel
Fading
Adversary
4
Wireless (Current Paper)
Transmitter
$H_3 x$
Background
$H_4 x$
Adversary
• 'Noisy' samples
• Input distributions are different for surrogate and target models.
$r = H_1 x + n + H_2 \delta$
Fading
Receiver
Perturbation is added through a channel
Fading
Adversary

Fig. 1. Adversarial attacks from computer vision to wireless applications.

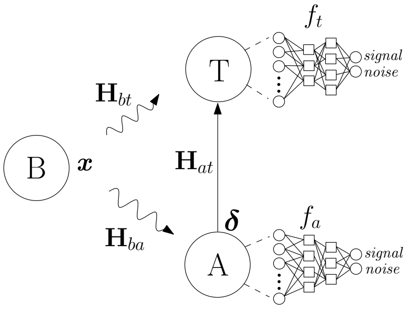

Fig. 2. System model. 'B' stands for the background emitter, 'T' stands for the transmitter, and 'A' stands for the adversary.

the target model and the surrogate model have the same distributions such that the adversarial attacks can be readily transferred from the surrogate model to the target model. However, practical wireless deployments require that the wireless adversary builds the surrogate model by observing the spectrum and collecting the training data over the air. Therefore, the surrogate model and the target model may not have the same input distributions due to the discrepancies in channels experienced by the adversary and the target receiver (e.g., due to different distances from the signal source that they are trying to sense). Fig. 1 summarizes the difference of adversarial attacks and surrogate models in different applications including computer vision and wireless communications.

In this paper, we investigate the channel effects on the surrogate model that is built by the adversary through over-the-air spectrum observations and used to craft adversarial attacks against a DNN-based wireless signal classifier. We consider a wireless communications system with a background emitter, a transmitter, and an adversary. The transmitter collects I/Q data and uses its DNN classifier to detect the ongoing transmission of the background emitter that is further used for channel access decisions. On the other hand, the adversary builds a surrogate model of the DNN classifier used at the transmitter. For that purpose, the adversary collects the I/Q data, namely signals received from the background emitter, by exploiting the broadcast nature of transmissions and obtains the labels by listening to the potential signals from the transmitter. If the transmitter classifies its received signal as 'noise', then it transmits so the adversary obtains the label 'noise'. If not, the label becomes 'signal'. Note that there could be discrepancies between the labels of the target model and the surrogate model that we do not consider in this paper. Instead, we focus on the discrepancies between the input data samples for the target model and the surrogate model.

Using the surrogate model, the adversary generates adversarial perturbations to fool the classifier at the transmitter from label 'signal' to label 'noise'. This setting may model a system where the background emitter is a primary user, the transmitter is a secondary user, and the adversary is trying to fool the secondary user to transmit even though the channel is used by the primary user. Compared to previous works, we make the realistic assumption that the adversarial attacks are generated using the surrogate model that is trained with a different distribution of the training dataset relative to the transmitter's target model since the signals received by the adversary and the transmitter are different due to channel discrepancies. Different topologies of the adversary are considered to investigate how the difference in the distribution of the training datasets affects the performance of the adversarial attack. Also, different approaches to select the transmit power at the adversary node using the surrogate model are considered. Finally, we relax the assumption that the adversary knows the exact input at the transmitter when determining the adversarial perturbation. Results show that the performance of adversarial attacks against a wireless signal classifier heavily relies on the reliability of surrogate model which depends on the difference of channels experienced by the adversary and the transmitter.

The rest of the paper is organized as follows. Section II provides the system model. Section III describes how to generate adversarial attacks against wireless signal classifiers by using surrogate models. Section IV considers different topologies to evaluate the effect of surrogate models. Section V presents simulation results. Section VI concludes the paper.

II. System Model

We consider a wireless communication system that consists of a background emitter, a transmitter, and an adversary as shown in Fig. 2. The background emitter transmits k complex symbols, $\boldsymbol{x} \in \mathbb{C}^k$, and node j (either the transmitter t or the adversary a) receives

$$\boldsymbol{r}_{bj} = \mathbf{H}_{bj}\boldsymbol{x} + \boldsymbol{n}_{bj}, \quad j \in \{t, a\}, \tag{1}$$

where $\mathbf{H}_{bj} = \text{diag}\{h_{bj,1}, \cdots, h_{bj,k}\} \in \mathbb{C}^{k \times k}$ and $\boldsymbol{n}_{bj} \in \mathbb{C}^k$ are the channel gains and complex Gaussian noise from the background emitter to node j, respectively.

Using $\boldsymbol{r}_{bt}$, the transmitter trains its DNN classifier, $f_t(.;\boldsymbol{\theta}_t) : \mathcal{X} \to \mathbb{R}^2$, to determine the existence of ongoing background transmission to utilize the idle bands, where $\boldsymbol{\theta}_t$ is the set of transmitter's DNN parameters and $\mathcal{X} \subset \mathbb{C}^k$. The input $\boldsymbol{r}_{bt}$ is assigned to the label $\hat{l}_t(\boldsymbol{r}_{bt}, \boldsymbol{\theta}_t) = \arg\max_q f_t^{(q)}(\boldsymbol{r}_{bt}, \boldsymbol{\theta}_t)$, where $f_t^{(q)}(\boldsymbol{r}_{bt}, \boldsymbol{\theta}_t)$ is the output of classifier f_t corresponding to the qth label.

Concurrently, the adversary tries to detect the background transmission based on $\boldsymbol{r}_{ba}$ using its own DNN classifier as the *surrogate model*, $f_a(.;\boldsymbol{\theta}_a) : \mathcal{X} \rightarrow \mathbb{R}^2$, where $\boldsymbol{\theta}_a$ is the set of the adversary's DNN parameters. The label corresponding to the input $\boldsymbol{r}_{ba}$ is defined as $\hat{l}_a(\boldsymbol{r}_{ba}, \boldsymbol{\theta}_a) = \arg\max_q f_a^{(q)}(\boldsymbol{r}_{ba}, \boldsymbol{\theta}_a)$, where $f_a^{(q)}(\boldsymbol{r}_{ba}, \boldsymbol{\theta}_a)$ is the output of classifier f_a corresponding to the qth label. If the label of the classifier is 'signal', the adversary transmits a perturbation $\boldsymbol{\delta} \in \mathbb{C}^k$ to change the label of the DNN classifier at the transmitter to 'noise'. Fooling the transmitter's classifier to change the label from 'noise' to 'signal' (by adding some interference) is easier than 'signal' to 'noise' (that requires careful signal cancellation), thus we only consider the more difficult case of changing the label from 'signal' to 'noise'.

If the background emitter transmits $\boldsymbol{x}$ and the adversary transmits $\boldsymbol{\delta}$, the received signal at the transmitter is

$$\boldsymbol{r}'_{bt}(\boldsymbol{\delta}) = \mathbf{H}_{bt}\boldsymbol{x} + \mathbf{H}_{at}\boldsymbol{\delta} + \boldsymbol{n}_{bt}, \tag{2}$$

where $\mathbf{H}_{at} = \text{diag}\{h_{at,1}, \cdots, h_{at,k}\} \in \mathbb{C}^{k\times k}$ is the channel gain from the adversary to the transmitter. The adversary generates the adversarial perturbation $\boldsymbol{\delta}$ to cause misclassification at the transmitter for the input $\boldsymbol{r}_{bt}$ while satisfying the power budget P_{max} for a stealthy attack. Thus, the adversary determines $\boldsymbol{\delta}$ by solving the following optimization problem:

$$\begin{aligned} \arg\min_{\boldsymbol{\delta}} \quad & ||\boldsymbol{\delta}||_2 \\ s.t. \quad & \hat{l}_t(\boldsymbol{r}_{bt}, \boldsymbol{\theta}_t) \neq \hat{l}_t(\boldsymbol{r}'_{bt}(\boldsymbol{\delta}), \boldsymbol{\theta}_t) \\ & ||\boldsymbol{\delta}||_2^2 \leq P_{max}. \end{aligned} \tag{3}$$

However, in reality, the adversary may not have information about the classifier at the transmitter, f_t, and it is not possible to check whether an attack generated at the adversary satisfies the constraint $\hat{l}_t(\boldsymbol{r}_{bt}, \boldsymbol{\theta}_t) \neq \hat{l}_t(\boldsymbol{r}'_{bt}(\boldsymbol{\delta}), \boldsymbol{\theta}_t)$, or not. Therefore, we change this constraint by using the DNN classifier at the adversary, f_a, which can be thought of as a *surrogate model* for the transmitter's classifier. Now, the optimization problem to generate the adversarial perturbation at the adversary is

$$\begin{aligned} \arg\min_{\boldsymbol{\delta}} \quad & ||\boldsymbol{\delta}||_2 \\ s.t. \quad & \hat{l}_a(\boldsymbol{r}_{bt}, \boldsymbol{\theta}_a) \neq \hat{l}_a(\boldsymbol{r}'_{bt}(\boldsymbol{\delta}), \boldsymbol{\theta}_a) \\ & ||\boldsymbol{\delta}||_2^2 \leq P_{max}. \end{aligned} \tag{4}$$

As the non-linearity of the DNNs makes the solution $\boldsymbol{\delta}^*$ to (4) difficult to obtain, the adversarial perturbation is approximated. Fast gradient method (FGM) [27] is an efficient method to craft adversarial attacks by linearizing the loss function, $L_j(\boldsymbol{\theta}, \boldsymbol{x}, \boldsymbol{y})$, of the jth node's DNN classifier in a neighborhood of input $\boldsymbol{x}$, where $\boldsymbol{y}$ is the label vector, and uses linearized loss function for the optimization. In this paper, we consider a targeted attack, specifically the maximum received perturbation power (MRPP) attack in [14], where the adversary's perturbation aims to decrease the loss function of data with label 'noise' and cause a specific misclassification, from label 'signal' to label 'noise', at the transmitter in the presence of background transmission.

III. Adversarial Attack built upon the Surrogate Model of the Adversary

In this section, we introduce how to craft an adversarial perturbation using the surrogate model of the adversary, f_a, to flip the label of the transmitter's classifier. To do so, the adversary first tries to detect the background transmission based on $\boldsymbol{r}_{ba}$ using its own DNN classifier, f_a. Since the adversary tries to change the label from 'signal' to 'noise', the adversary generates an attack if the label $\hat{l}_a(\boldsymbol{r}_{ba}, \boldsymbol{\theta}_a) = \arg\max_q f_a^{(q)}(\boldsymbol{r}_{ba}, \boldsymbol{\theta}_a) =$ 'signal'. Then, the adversary crafts the adversarial perturbation using the MRPP attack. Note that we assume that the adversary knows the channel $\mathbf{H}_{at}$ and the exact input at the transmitter, $\boldsymbol{r}_{bt}$, when creating the adversarial perturbation. We relax the assumption regarding the knowledge of the exact input at the transmitter by using the input at the adversary, $\boldsymbol{r}_{ba}$, instead of $\boldsymbol{r}_{bt}$ later in the paper.

Since the adversary has no information about the transmitter's classifier, it is hard for the adversary to determine the transmit power that is needed to change the label at the transmitter. Therefore, we consider two different ways to decide the transmit power at the adversary.

A. Adversarial perturbation using the maximum power P_{max}

Without knowing the transmitter's classifier, one simple option for the adversary is to use the maximum power for the adversarial perturbation. Therefore, the adversary simply transmits $\boldsymbol{\delta}^* = -P_{max}\frac{\nabla_{\boldsymbol{x}}\mathbf{H}^*_{at}L_a(\boldsymbol{\theta}_a, \boldsymbol{r}_{bt}, \boldsymbol{y}^{target})}{(||\nabla_{\boldsymbol{x}}\mathbf{H}^*_{at}L_a(\boldsymbol{\theta}_a, \boldsymbol{r}_{bt}, \boldsymbol{y}^{target})||_2)}$, where $\boldsymbol{y}^{target}$ is 'noise'.

B. Adversarial perturbation using the surrogate model of the adversary

The transmit power at the adversary should be carefully determined to fool the transmitter's classifier. This has been done in [14], [28] by using the transmitter's classifier. However, the adversary has no information about the transmitter's classifier. Therefore, the adversary determines the transmit power using its own classifier as a surrogate model for the transmitter's classifier. The details of how to generate the adversarial perturbation are presented in Algorithm 1.

C. Adversarial perturbation using r_{ba}

Previously, we assumed that the adversary knows the exact input at the transmitter, $\boldsymbol{r}_{bt}$. However, it is impractical to assume that the adversary knows $\boldsymbol{r}_{bt}$. Thus, the adversary uses the received signal at the adversary, $\boldsymbol{r}_{ba}$, instead of $\boldsymbol{r}_{bt}$ to craft the adversarial perturbation. The same procedure used in Section III-A and Section III-B can be used to generate the adversarial perturbation without the knowledge of the input at the transmitter by changing $\boldsymbol{r}_{bt}$ to $\boldsymbol{r}_{ba}$ in conjunction with using the surrogate model of the adversary.

IV. Topologies to Characterize the Effects of Surrogate Models on Adversarial Attacks

We consider different topologies to investigate the effects of the surrogate models (used by the adversary at different locations) on the performance of adversarial attacks. We assume

Algorithm 1: Adversarial perturbation using $\boldsymbol{r}_{ba}$

```
Inputs: $\boldsymbol{r}_{bt}$, desired accuracy $\varepsilon_{acc}$ and $P_{max}$
Initialize:
 $\varepsilon \leftarrow 0, \varepsilon_{max} \leftarrow \sqrt{P_{max}}, \varepsilon_{min} \leftarrow 0, \boldsymbol{y}^{target} \leftarrow$ 'noise'
$\boldsymbol{\delta}_{norm} = \frac{\nabla_{\boldsymbol{x}} \mathbf{H}^*_{at} L_a(\boldsymbol{\theta}_a, \boldsymbol{r}_{bt}, \boldsymbol{y}^{target})}{(||\nabla_{\boldsymbol{x}} \mathbf{H}^*_{at} L_a(\boldsymbol{\theta}_a, \boldsymbol{r}_{bt}, \boldsymbol{y}^{target})||_2)}$
if $\hat{l}_a(\boldsymbol{r}_{ba}) ==$ 'noise' then
    while $\varepsilon_{max} - \varepsilon_{min} > \varepsilon_{acc}$ do
        $\varepsilon_{avg} \leftarrow (\varepsilon_{max} + \varepsilon_{min})/2$
        $\boldsymbol{x}_{adv} \leftarrow \boldsymbol{r}_{bt} - \varepsilon_{avg} \mathbf{H}_{at} \boldsymbol{\delta}_{norm}$
        if $\hat{l}_a(\boldsymbol{x}_{adv}) ==$ 'noise' then  $\varepsilon_{min} \leftarrow \varepsilon_{avg}$
        else $\varepsilon_{max} \leftarrow \varepsilon_{avg}$
    end
end
$\varepsilon = \varepsilon_{max}$, $\boldsymbol{\delta}^* = -\varepsilon \boldsymbol{\delta}_{norm}$
```

that the channel between node i and node j is Rayleigh fading with path-loss and shadowing, i.e., $h_{ij,k} = K(\frac{d_0}{d_{ij}})^{\gamma} \psi h^{ray}_{ij,k}$, where $K = 1, d_0 = 1, \gamma = 2.7, \psi \sim \text{Lognormal}(0, 8)$, $h^{ray}_{ij,k} \sim \text{Rayleigh}(0, 1)$ and d_{ij} is the distance between node i and node j. Thus, if the location of the adversary changes with respect to the background emitter, the classifier at the adversary is trained with a different input distribution that changes with respect to the distance from the background emitter. Also, if the location of the adversary changes with respect to the transmitter, the usage of the power needed at the adversary changes, i.e., the adversary needs more power if the distance between the transmitter and adversary increases. To assess the performance of the classifier at the adversary when the location changes, we consider the following adversary locations while we fix the distance between the background emitter and the transmitter as 1 in all topologies.

A. Fixed distance between the background emitter and the adversary

We first consider topologies where the distance between the background emitter and the adversary is fixed for all cases and the only difference among adversary locations is the distance to the transmitter as shown in Fig. 3, i.e., the adversary uses the same surrogate model for these topologies. Specifically, the distance between the background emitter and the adversary, namely d_{ba}, is 0.5 for adversary locations A1-A4 in Fig. 3 and the distance between the transmitter and the adversary, namely d_{ta}, is $0.5, 1, \sqrt{1.25}, 1.5$ for adversary locations A1-A4 in Fig. 3, respectively. In this setting, the classifier of the adversary remains the same when its location changes, since the distributions of received signals from the background emitter are the same. The only difference among the adversary locations is the distance to the transmitter.

B. Fixed distance between the transmitter and the adversary

Next, we consider the topology in Fig. 4, where the distance between the adversary and the transmitter is fixed for all cases and the difference among the adversary locations is the distance to the background emitter, i.e., the surrogate models of the adversary are different, whereas the channel effects on the adversarial perturbations have the same distributions. Specifically, d_{ta} is 0.5 for adversary locations A1, A5, A6, and A7 in Fig. 4 and d_{ba} is $0.5, 1, \sqrt{1.25}, 1.5$ for adversary locations A1, A5, A6, and A7 in Fig. 4, respectively. Since the locations of the adversary are different with respect to the background emitter, the classifiers of the adversary at different locations are trained with different distribution of inputs. Note that the classifier at the transmitter is trained with the same distribution of inputs as the adversary at position 5 (A5) and differently from other locations.

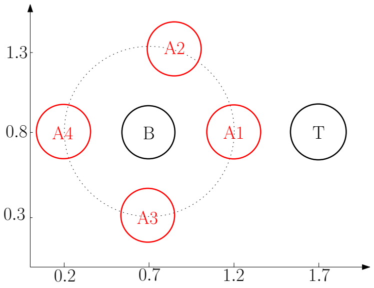

Fig. 3. Fix the distance from the background emitter 'B' to the adversary (A1-A4 correspond to different locations of the adversary).

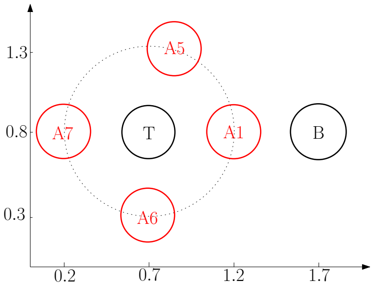

Fig. 4. Fixed distance between the transmitter 'T' and the adversary (A1, A5-A7 correspond to different locations of the adversary).

V. Performance Evaluation

In this section, we investigate how different topologies described in Section IV and the corresponding channels affect the performance of the adversarial perturbation built upon the adversary's surrogate model. We assume that there is QPSK signal transmission in the background and the classifiers at the transmitter and the adversary are both a convolutional neural network (CNN), where the input to the CNN is of two dimensions (2,16) corresponding to 16 in-phase/quadrature (I/Q) data samples. We investigate the adversarial perturbation performance when the classifier at the adversary is the same as or different from the classifier at the transmitter. In all cases, the adversary trains a surrogate model that is different from the model of the transmitter's classifier.

The default CNN structure used in the paper consists of a convolutional layer with kernel size $(1, 3)$, a hidden layer with dropout rate 0.1, ReLu activation function at convolutional and hidden layers and softmax activation function at the output layer that provides the label 'signal' or 'noise'. We apply

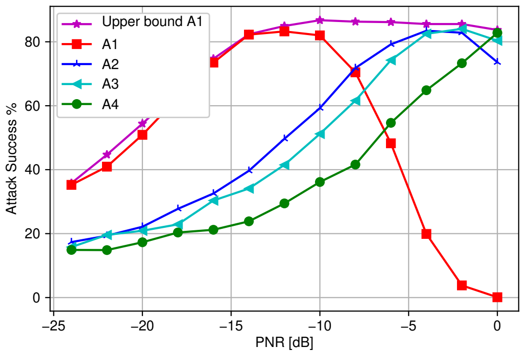

Fig. 5. Attack performance of the adversary using maximum power when d_{ba} is fixed.

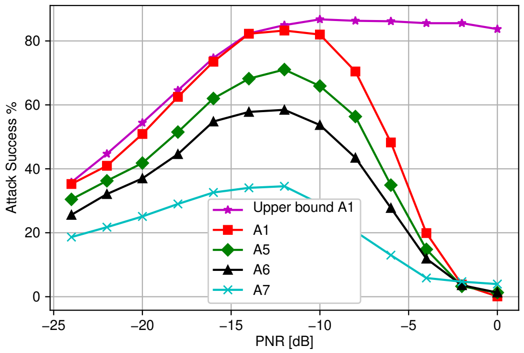

Fig. 6. Attack performance of the adversary using maximum power when d_{ta} is fixed.

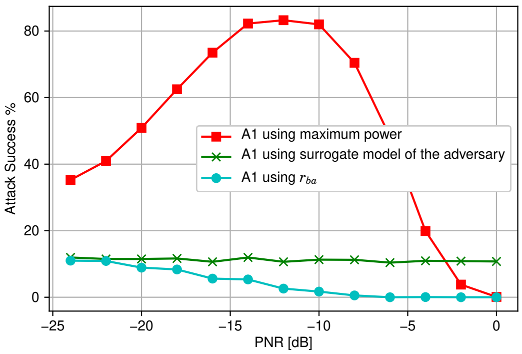

Fig. 7. Attack performance of adversary using maximum power, surrogate model of the adversary, or $\boldsymbol{r}_{ba}$.

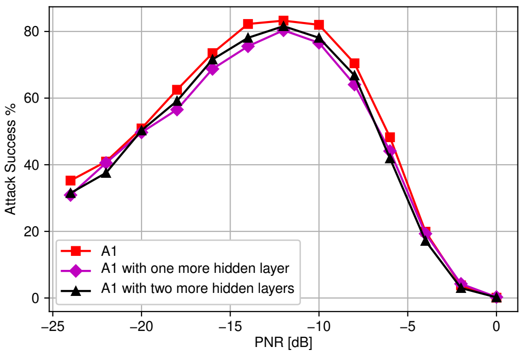

Fig. 8. Attack performance of adversary with different classifier architectures.

the backpropagation algorithm with Adam optimizer to train the CNN using cross-entropy as the loss function. The CNN is implemented in Keras with TensorFlow backend. We start with the same classifier architecture for the transmitter and the adversary. Even in this case, each classifier is trained with a different input data distribution due to the different channel between the background emitter and each node. We use the perturbation-to-noise ratio (PNR) metric that is also used in [14], [28] to represent the relative perturbation power with respect to the noise. Note that as the PNR increases, it becomes easier to detect the adversary's transmission.

In Fig. 5, we first consider the topology depicted in Fig.3, where the distance between the background emitter and each adversary location is the same. We use the same classifier for the adversary at all locations in this topology since the input distributions for the adversary at all these locations are the same and the only difference is its distance to the transmitter. The case where the classifier of the adversary is the same as the transmitter's classifier is considered as an upper bound where the power is determined using Algorithm 1. As the distance between the adversary and the transmitter increases, the peak of the curve shifts to the right meaning that it needs more power to fool the classifier at the transmitter. Moreover, the attack success first increases as the PNR increases and then decreases after exceeding some PNR value. This is because the adversary transmits the adversarial perturbation with the maximum power that fails to fool the classifier at the transmitter into classifying the received signal as 'noise' when too much power is used at the adversary. Therefore, the adversary should select the right amount of power to transmit the adversarial perturbation. When the surrogate model at the adversary is the same as the transmitter's classifier, the attack success increases up to some PNR value and then saturates after exceeding it.

Next, we consider a different topology, where the distance between the transmitter and each adversary location is the same as shown in Fig. 4. Since the distance between the background emitter and each adversary location is different, the distributions of the received signal at different adversary locations are different. Thus, we train the classifiers for the adversary at different locations differently and use these classifiers in the simulations. In Fig. 6, as the distance between the background emitter and the adversary increases, the peak of the attack success decreases. In addition, even though the adversary at location 5 is trained with the same distribution as the transmitter's classifier, the classifier of the adversary at location 1 that is closer to the background emitter performs better. This observation suggests that the adversary should be located closer to the background emitter when the distance to the transmitter is fixed to increase the attack performance.

The performance of the adversarial attack using maximum power, the surrogate model, or r_{ba} is compared in Fig. 7. Even though the adversary precisely decides the transmit power based on its own classifier, this attack performs poorly compared to the case when the adversary uses maximum power. When we relax the assumption of knowing r_{bt} and instead use r_{ba} to generate the adversarial attack at the adversary, the overall performance drops compared to other methods and decreases when the PNR increases.

Finally, we apply a DNN architecture for the surrogate model at the adversary that is different from the transmitter's classifier. The results are shown in Fig. 8. Although the number of hidden layers increases compared to the CNN with one hidden layer, the impact on the attack success is negligible. The transferability [29] property holds since the performance has not changed although the architecture changes at the adversary. However, the surrogate models may be significantly different from the target model as the channels experienced by the transmitter and the adversary may differ. There may be other channel differences, such as multipath, intersymbol interference (ISI), and mobility (Doppler), present in practice. These differences may further increase the difference between the target model and the surrogate model at the adversary, and reduce the attack success. Therefore, the transferability argument is not readily applicable in practical wireless applications of over-the-air adversarial attacks.

VI. Conclusion

We considered a wireless communication system where an adversary transmits a perturbation signal to fool the DNN classifier at a transmitter into classifying the ongoing background transmission as noise. The adversary trains its surrogate model by observing the spectrum and uses this model to design the adversarial attack. Through different topologies, we showed how the adversary's location significantly affects the attack performance as the surrogate model may differ from the target model due to channel discrepancies. In particular, the attack success against the transmitter drops when the adversary moves away from the background emitter (and the surrogate model becomes less reliable) although the adversary does not necessarily move closer to the transmitter.

References

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

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

[3] K. Davaslioglu and Y. E. Sagduyu, "Generative adversarial learning for spectrum sensing," in *IEEE International Conference on Communications (ICC)*, 2018.

[4] Y. Shi, K. Davaslioglu, Y. E. Sagduyu, W. C. Headley, M. Fowler, and G. Green, "Deep learning for signal classification in unknown and dynamic spectrum environments," in *IEEE International Symposium on Dynamic Spectrum Access Networks (DySPAN)*, 2019.

[5] T. Erpek, T. O'Shea, Y. E. Sagduyu, Y. Shi, and T. C. Clancy, "Deep learning for wireless communications," in *Development and Analysis of Deep Learning Architectures*. Springer, Cham, 2020, pp. 223–266.

[6] Y. Shi, Y. E. Sagduyu, T. Erpek, K. Davaslioglu, Z. Lu, and J. H. Li, "Adversarial deep learning for cognitive radio security: Jamming attack and defense strategies," in *IEEE International Conference on Communications (ICC)*, 2018.

[7] 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.

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

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

[10] S. Kokalj-Filipovic, R. Miller, and G. M. Vanhoy, "Adversarial examples in RF deep learning: Detection and physical robustness," in *IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, 2019.

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

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

[13] Y. Lin, H. Zhao, Y. Tu, S. Mao, and Z. Dou, "Threats of adversarial attacks in DNN-based modulation recognition," in *International Conference on Computer Communications (INFOCOM)*, 2020.

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

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

[16] M. Z. Hameed, A. Gyorgy, and D. Gunduz, "Communication without interception: Defense against modulation detection," in *IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, 2019.

[17] B. Kim, Y. E. Sagduyu, K. Davaslioglu, T. Erpek, and S. Ulukus, "How to make 5G communications "invisible": Adversarial machine learning for wireless privacy," in *Asilomar Conference on Signals, Systems, and Computers*, 2020.

[18] B. Kim, Y. E. Sagduyu, K. Davaslioglu, T. Erpek, and S. Ulukus, "Adversarial attacks with multiple antennas against deep learning-based modulation classifiers," in *IEEE Global Communications Conference (Globecom)*, 2020.

[19] Y. E. Sagduyu, Y. Shi, and T. Erpek, "IoT network security from the perspective of adversarial deep learning," in *IEEE International Conference on Sensing, Communication, and Networking*, 2019.

[20] Y. Shi, T. Erpek, Y. E. Sagduyu, and J. Li, "Spectrum data poisoning with adversarial deep learning," in *IEEE Military Communications Conference (MILCOM)*, 2018.

[21] Y. E. Sagduyu, T. Erpek, and Y. Shi, "Adversarial deep learning for over-the-air spectrum poisoning attacks," *IEEE Transactions on Mobile Computing*, no. 1, pp. 2–14, Feb. 2021.

[22] Y. Shi, K. Davaslioglu, and Y. E. Sagduyu, "Over-the-air membership inference attacks as privacy threats for deep learning-based wireless signal classifiers," in *ACM WiSec Workshop on Wireless Security and Machine Learning (WiseML)*, 2020.

[23] K. Davaslioglu and Y. E. Sagduyu, "Trojan attacks on wireless signal classification with adversarial machine learning," in *IEEE DySPAN Workshop on Data-Driven Dynamic Spectrum Sharing*, 2019.

[24] Y. Shi, K. Davaslioglu, and Y. E. Sagduyu, "Generative adversarial network for wireless signal spoofing," in *ACM Workshop on Wireless Security and Machine Learning (WiseML)*, 2019.

[25] Y. Shi, K. Davaslioglu, and Y. E. Sagduyu, "Generative adversarial network in the air: Deep adversarial learning for wireless signal spoofing," *IEEE Transactions on Cognitive Communications and Networking*, vol. 7, no. 1, pp. 294–303, March 2021.

[26] T. J. O'Shea, T. Roy, and T. C. Clancy, "Over-the-air deep learning based radio signal classification," *IEEE Journal of Selected Topics in Signal Processing*, vol. 12, no. 1, pp. 168–179, January 2018.

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

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

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