Multi-subcarrier Physical Layer Authentication Using Channel State Information and Deep Learning

Ken St. Germain, Frank Kragh
Department of Electrical and Computer Engineering
Naval Postgraduate School
{ kenneth.stgermain, fekragh} @nps.edu

Abstract

Strong authentication is crucial as wireless networks become more widespread and relied upon. The robust physical layer features produced by advanced communication networks lend themselves to accomplishing physical layer authentication by using channel state information (CSI). The use of deep learning with neural networks is well suited for classification tasks and can further the goal of enhancing physical layer security. To that end, we propose a semi-supervised generative adversarial network to differentiate between legitimate and malicious transmitters and accurately identify devices for authentication across a range of signal to noise ratio conditions. Our system leverages multiple input multiple output CSI across orthogonal frequency division multiplexing subcarriers using a small percentage of labeled training data.

1. Introduction

Efficient and effective security measures are required to ensure modern and future communication systems perform to their full potential. Security is commonly and successfully accomplished using key-based cryptography, however as networks grow and become more complex, key distribution and management may not scale without causing undue user delays [1, 2]. The latest iterations of IEEE 802.11 WiFi standards and 5th generation New Radio (5G-NR) mobile networks leverage several techniques at the physical layer to provide high data rate communications to multiple users [3]. The features created by these techniques create opportunities for the employment of physical layer security.

Physical layer security is accomplished at the bottom of the Open Systems Interconnection (OSI) stack. Thus, when an illegitimate device attempts to create a secure connection, the expenditure of undue resources is reduced in the higher layers. This is especially important for devices that are computationally or energy constrained, such as those in the Internet of Things (IoT) realm [4].

In this work, we explore physical layer authentication and deep learning to ensure strong authentication on a variety of devices with disparate applications. The goal in this paper is to prevent illegitimate devices from authenticating and then correctly identify legitimate devices based on features observed at the physical layer. These features are realized in a particular environment based on the system characteristics, such as carrier frequencies, using multiple transmitter and receiver antennas, and the use of multiple subcarriers through such methods as orthogonal frequency-division multiplexing (OFDM). Although no approach can guarantee prevention of an intruder authenticating, using the uniqueness of CSI for physical layer authentication can improve the overall security of a network. Devices that use no authentication strategy at all offer an attractive vulnerability for an adversary.

In this paper we use a generative adversarial network (GAN) to determine which of several transmitters should be authenticated or denied access. Specifically, we employ a GAN model with semi-supervised learning known as a semi-supervised GAN (SGAN) [5] where only a small portion of the training dataset is labeled.

The literature proposes two broad categories to distinguish legitimate from illegitimate devices at the physical layer without the use of a pre-shared secret, cryptography, user-provided credentials, or higher OSI-layer processing. The first category relies on unique imperfections of the transmitter hardware that manifest as radio frequency (RF) fingerprints or signatures [6]. Based on manufacturing processes and designs, the transmitted signal will be uniquely distorted from device to device, even if only slightly. The second category leverages the stochastic nature of the wireless channel to take advantage of multi-path fading environments. The temporally and spatially-unique impulse or frequency response can be used to identify the transmitter [7].

Our proposed method is based on research using the second category. The effects of the multipath channel can be described in the channel state information (CSI) matrix. The elements of the CSI matrix are attributes of the fading channel and are therefore unique to the pairwise position of the receiver and transmitter in line-of-sight (LOS) and non-line-of-sight (NLOS) multipath environments. While there is merit in using the RF fingerprinting method in the first category, these characteristics are observable by a malicious actor and can be spoofed. Contrast with using the channel-based approach, an adversary cannot directly measure the CSI between two entities and create a transmission to mimic a legitimate signal. In dynamic conditions with a mobile transmitter, receiver, and/or significant reflective or absorbing objects, the CSI is also time-variant. The focus of this paper is on the static case, and using a technique as described by [8] can be adopted to account for scenarios where motion is expected to change the CSI. We use the static channel here to explore the use of both the magnitude and phase of the CSI.

To simulate multiple-input multiple-output (MIMO) millimeter wave OFDM subchannels, we take advantage of the DeepMIMO dataset [9], based on ray-tracing data from the Remcom Insite tool [10]. The DeepMIMO dataset is configurable to a variety of wireless applications, and we use it to create samples that train and test the SGAN in a 4×4 MIMO environment operating with a 60 GHz carrier frequency with 16 pilot subcarriers from a 512 OFDM subcarrier system in an urban setting. Although the Deep MIMO dataset scenario is based on an urban environment, more fully appropriate use-cases for static channels might include an uninhabitable industrial setting, or a in a remote, deployed sensor network.

In this paper, we use a SGAN to identify malicious users who attempt to spoof the CSI of legitimate users and prevent them from authenticating. We also identify legitimate users and correctly categorize them. The contributions of this paper are:

- We introduce analysis illustrating how the received CSI matrix elements and measurement error over multiple subcarriers can be used for physical layer authentication.
- We propose a discriminative model that processes both legitimate samples from a trusted source and faked samples created by the generative model to determine whether a transmitter should be authenticated using complex-valued CSI elements.
- We propose four classifier neural network models that determine the identity of a transmitter based on associated CSI matrix samples at various levels of signal to noise ratio (SNR)
- In Section 6, we use a SGAN architecture to accurately classify MIMO CSI as a basis of physical layer authentication for multiple transmitters. We use small amounts of labeled training data to perform physical layer authentication. By not relying on a fully labeled dataset, we benefit by using a method to reduce authentication overhead.
- Distinguishing from previous research, our proposal takes advantage of the SGAN-trained discriminator's performance across different SNR levels to prevent illegitimate transmitters from authenticating while minimizing the number of required labeled samples. The SGAN-trained classifier then accurately identifies samples from legitimate transmitters.

This paper discusses the channel model and previous work in the application of machine learning within the RF domain in Section 2. Next we present the development of the SGAN in Section 3. Section 4 describes the DeepMIMO dataset and how we use it to create CSI samples. The system model for the SGAN is illustrated in Section 5, and simulation results are shown in Section 6. Finally, we summarize our observations and discuss future work in Section 7. With respect to notation, unless otherwise addressed, vectors are indicated with bold lower-case letters, and matrices with bold upper-case letters.

2. Background and related research

This section introduces the channel model, the use of GANs for RF tasks, the model for measured CSI, and the architecture of a semi-supervised GAN.

2.1. Channel Model

By taking advantage of the randomness and uniqueness inherent in the RF communication channel, physical layer information provided by the channel can be used to conduct authentication [11]. The nature of the wireless medium affects the transmitted signal as it propagates to the receiver. The single subcarrier narrowband model of the wireless channel is given by

$$\boldsymbol{y} = \boldsymbol{H}\boldsymbol{x} + \boldsymbol{n} \tag{1}$$

where $\boldsymbol{y}$ is the received signal, $\boldsymbol{x}$ is the transmitted signal, $\boldsymbol{H}$ is the time-varying CSI or channel response, and $\boldsymbol{n}$ is the noise vector. For a Rayleigh faded channel, $\boldsymbol{H}$ is an

$N \times M$ matrix of circularly symmetric complex-valued Gaussian random variables. The number of transmitter antennas is M and the number of receiver antennas is N. Each complex element within $\boldsymbol{H}$, $h_{n,m}$, is composed of real and imaginary zero-mean independent Gaussian random variables with identical variance [12]. Jakes' uniform scattering model [13] states that antennas spatially separated more than two carrier wavelengths from each other will observe sufficiently independent fading channels due to rapid decorrelation of the signal envelope among receivers.

Although the CSI elements may be independent for a single subcarrier, that does not necessarily hold true when there are multiple subcarriers, such as in an OFDM system. For K subcarriers, we extend (1) by adding an additional dimension as a superscript where $k = 1, 2, \dots, K$ is the sampled subcarrier, resulting in

$$\boldsymbol{y}^k = \boldsymbol{H}^k \boldsymbol{x}^k + \boldsymbol{n}^k \tag{2}$$

where $\boldsymbol{H}^k$ is a three-dimensional tensor of size $N \times M \times K$. Fading across the subcarrier channels will be correlated if the coherence bandwidth is large [14], resulting in correlated CSI elements across subchannels, for example, $h^k_{1,1}$. To illustrate this, Fig. 1 depicts the magnitude of $h^k_{1,1}$ over 512 subcarriers in an OFDM system with 16 pilot subcarriers with correlated fading. There is correlation between adjacent OFDM channels, but not across the entire band of subcarriers or adjacent pilots channels.

2.2. Authentication with measured CSI

A receiver continues to authenticate a transmitter if the received CSI varies less than a threshold applied to the received CSI from previous transmissions. This requires some method of initial authentication, such as the use of cryptographic methods or physical layer authentication using RF fingerprinting from transmitter imperfections. During initial authentication, the receiver makes CSI measurements of the channel and stores that information for future authentication.

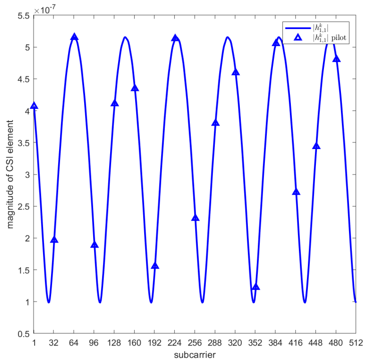


Figure 1. CSI element magnitude vs. subcarrier.

During channel measurement, even in a stable static environment, the receiver imparts noise to the received signal, resulting in variation to the measured CSI elements. This error, $\boldsymbol{\epsilon}$, is modeled as an additive complex zero-mean Gaussian process on each subcarrier, $\mathcal{CN}(0, \boldsymbol{\Sigma}_\epsilon)$, where the covariance of the sample mean is $\boldsymbol{\Sigma}_{\overline{\epsilon}} = \boldsymbol{\Sigma}_\epsilon / s$ for s samples during the measurement. Therefore, the tth CSI measured by the receiver at subcarrier k, $\hat{\boldsymbol{H}}^k_t$, is given as

$$\hat{\boldsymbol{H}}^k_t = \boldsymbol{H}^k + \boldsymbol{\epsilon}^k_t \qquad t = 1, 2, \dots, s \tag{3}$$

where $\boldsymbol{H}^k$ is the true CSI from (2) and $\boldsymbol{\epsilon}^k_t$ is a complex $N \times M \times K$ tensor with independent identically distributed elements. Since $\boldsymbol{\epsilon}^k_t$ is zero-mean, $\boldsymbol{H}^k$, can be estimated with a variety of techniques including least squares estimation, minimum mean-square error estimation, and through successive measurements and element-wise averaging of $\hat{\boldsymbol{H}}^k_t$ for $t = \{1, 2, \dots, s\}$ as demonstrated in [15].

2.3. Machine learning applied to the RF Domain

Since the received CSI relies on the position of the transmitter, the receiver, and reflecting and absorbing materials in the environment, researchers have used machine learning to successfully resolve localization challenges. By collecting and recording CSI and known transmitter locations in advance, machine learning provided accurate a posteriori transmitter positions [16, 17, 18]. The use of machine learning and location information has been used to make an authentication decision using CSI [19, 20, 21]. In [21], Pan et al. showed that authentication performance was better in scenarios with stationary systems, abundant multi-path effects, and separation of transmitter antennas by more than one-half wavelength.

Introduced by Goodfellow et al. [22], the GAN framework trains two artificial neural network models called the discriminator and the generator as they compete against each other in an adversarial competition. Although used extensively in image processing fields, GANs have been shown useful for investigations in the RF field as well.

O'Shea et al. [23] used a GAN to determine the optimal modulation scheme in a given channel, showing how GANs can allow for adaptation to the RF environment. Based on earlier work by O'Shea et al. for radio modulation classification [24], Li et al. [25] employed a SGAN to classify 11 different modulation types and improve the classification performance over a convolutional neural network model. In an adversarial situation such as jamming and spoofing, Roy [26] proposed the use of GANs for building a robust system that can determine legitimate transmitters from illegitimate ones based on the imbalance of in-phase and quadrature components of a symbol constellation.

Our contribution to this area is the application of an SGAN to the complex-valued CSI elements for classification. Whereas previous works that research physical layer authentication with CSI use a channel gain coefficient matrix or use a method that normalizes the channel response, we retain the real and imaginary parts of the CSI elements in an effort to retain as much information as possible from the channel. We also examine the performance of the SGAN and additional neural networks across a range of SNR.

3. Semi-Supervised GAN

Semi-supervised learning for neural networks requires that only a portion of the training data be labeled. As opposed to supervised learning where all the training data is labeled or unsupervised learning where there are no labels and the networks must find their own way to organize the data, semi-supervised algorithms attempt to correctly identify samples when only a small portion of the training data is labeled. This can be very helpful when the dataset is large and it would be laborious and time-intensive for an expert to correctly label every sample manually. In our case, the labeled data comes from the CSI samples recorded during the initial authentication session. If more data is required, the overhead will increase.

When training a vanilla GAN in an unsupervised learning architecture, the discriminative model, $\mathcal{D}$, is a binary classifier that receives unlabeled authentic samples from the training data or fake samples generated by the generative model, $\mathcal{G}$. The generative model creates fake samples based on a function with random seed input, and the parameters in $\mathcal{G}$. The discriminative model assigns a probability from zero to one based on its perception of whether the sample is fake (0.0) or authentic (1.0). The value function that describes this relationships from the original work by Goodfellow [22] is given by

$$\min_G \max_D V(D,G) = \mathbb{E}_{x\sim p_{data}(x)}[\log D(x)] + \mathbb{E}_{z\sim p_z(z)}[\log(1 - D(G(z)))] \tag{4}$$

where $D(x)$ is the probability that x came from the data distribution $p_{data}(x)$ containing authentic training samples, and $D(G(z))$ is the estimate of the probability that the discriminator incorrectly identifies the fake instance as authentic. The generator network attempts to maximize $D(G(z))$, while the discriminator network tries to minimize it. The generator creates samples, $G(z)$, based on the parameter values in G and the random seed values z provided to the generator consistent with $p_z(z)$. Adversarial data, the output of the generator, is not random, however the generator makes use of random data to produce an output. The generator output is shaped by the parameters of the generator and the binary cross entropy loss based on the discriminator output when the generator creates samples.

With semi-supervised learning, a small percentage of the training data is labeled, and instead of using a binary classifier, the discriminator is a multi-class classifier. For N classes, the model requires $N+1$ outputs to account for all the authentic classes plus one additional class for the fake generated class. This can be implemented in a variety of ways. Following Salimans et al. [27], we can build an N-class classifier network, $\mathcal{C}$, with output logits $\{l_1, l_2, \ldots, l_N\}$ prior to the *softmax* activation for $\mathcal{C}$. The logits vector is then used as the input to the activation function for $\mathcal{D}$, which is given as $D(x) = \frac{Z(x)}{Z(x)+1}$, where $Z(x) = \sum_{n=1}^{N} \exp[l_n(x)]$. Because $\mathcal{D}$ and $\mathcal{C}$ share the same weights, both networks act as a single network, $\mathcal{D}/\mathcal{C}$, that is updated during backpropagation based on their respective loss functions, $J^{(\mathcal{D})}$ and $J^{(\mathcal{C})}$. The generator loss function is given by $J^{(\mathcal{G})}$.

Fig. 2 shows a functional depiction of a SGAN in training. The training dataset is partially labeled and provided to the $\mathcal{D}/\mathcal{C}$ model for classification by $\mathcal{C}$. The remainder of the training dataset as well as the generated samples from $\mathcal{G}$ are used as input to $\mathcal{D}/\mathcal{C}$ for discrimination where $\mathcal{D}$ will predict whether the sample

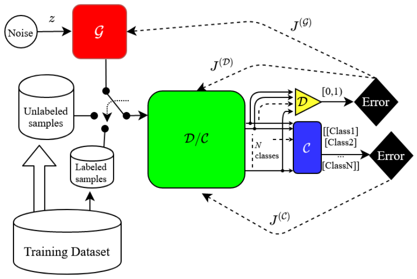


Figure 2. Training a semi-supervised generative adversarial network with N classes.

came from the training dataset or if it was created by $\mathcal{G}$.

4. The DeepMIMO dataset

Using the Remcom Insite ray-tracing tool [10], Alkhateeb developed the DeepMIMO dataset generation framework [9]. The framework allows researchers to tailor parameters in a MATLAB [28] program to suit the need of their machine learning based wireless application. This section discusses the setting we use to obtain our training and testing data, and the parameters we selected for our model.

4.1. Target data and labels

The target data for our proposed SGAN is the received 4×4 MIMO CSI from a transmitter to 14 distinct user locations, $\{User0, User1, \ldots, User13\}$, across 16 subcarriers. The transmitter and receivers operate at 60 GHz using 512 OFDM subcarriers. Each of the 16 pilot subcarriers are evenly spaced 32 subcarriers apart. A single target sample consists of 256 complex numbers, accounting for 16 CSI elements in each of their respective 16 subcarriers.

For each CSI target sample created in the dataset, there is an accompanying target label, denoting the user. During training, the goal for the SGAN classifier will be to differentiate among the legitimate users. The SGAN discriminator will attempt to categorize these legitimate users as "Real", and categorize the generator-created samples as "Fake".

During testing for the discriminator, an additional user, representing a malicious actor will be added to the test dataset. The discriminator will not have seen this data during testing, but will need to identify samples from the malicious user's CSI as "Fake" to prevent authentication. The classifier will attempt to assign the correct label for each user's CSI from the test dataset, however the classifier will not be exposed to the malicious user's CSI, since in application, the discriminator would have already prevented authentication.

4.2. DeepMIMO scenario

The setting used for the scenario is denoted "O1" and is described in detail in [9]. The "O1" scenario is an outdoor urban setting with a variety of possible transmitter base station locations and user locations on the streets surrounded by buildings of various heights. Fig. 3 shows the position of the 14 legitimate users denoted by blue circles and the red square indicating the malicious user in the white patch above the "User Grid 3" label. Base station 7 (BS7) is the transmitter for our case, circled in white and is across the intersection from the users. Both streets are 40 m wide, and the user positions are centered in the street going in the X direction and 7 m from the street going in the Y direction. There are 10 cm between adjacent users, and each user as well as BS7 has four antennas.

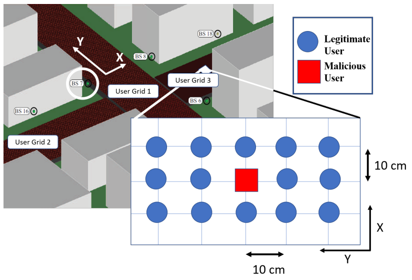


Figure 3. DeepMIMO scenario "O1". After [9]

.

Table 1. DeepMIMO dataset parameters

Base Station	7
First row of users	4528
Last row of users	4531
Center frequency	60 GHz
Antenna spacing	1 wavelength
System bandwidth	8.64 GHz
OFDM channels	512
OFDM channel interval	32
Number of paths	3

4.3. DeepMIMO parameters

The parameters chosen were made to emulate advanced wireless communication technologies, but they are not intended to model any specific standard. Table 1 summarizes the parameters used. Advanced technologies refers to communications systems using millimeter wavelengths and multiple antennas for transmitting and receiving signals. We believe that such devices will become adopted and more commonplace in the future. While the parameters chosen do not match any particular technology, they are analogous to those described by IEEE standards recently released or currently in draft.

5. System model

We consider a 4×4 wireless MIMO communications channel using 512 OFDM subchannels with 16 pilots. There are 14 trusted users and some unknown number of untrusted users, some of the latter group are malicious adversaries. The adversaries have resources available to change their antenna characteristics, transmitter RF path timing, output power, and/or present reflectors between themselves and the receiver. Thus, they are able to change their CSI as measured by the receiver and may have an accomplice receiver to provide feedback as described by Shi et al. in [29]. Although the adversaries have the ability to change their CSI, they do not have accurate advanced knowledge of the CSI required to spoof the user. A user becomes a victim if the malicious adversary is able to create CSI that is authenticated as the transmitter by the user.

To defeat this scenario, the discriminative model at the receiver is trained by a generative model that creates authentic looking CSI samples. By training with increasingly high quality "fake" samples, the discriminative network learns the features of transmitters that should be authenticated and the features of those that should not be authenticated. Parallel to the adversarial training between $\mathcal{D}$ and $\mathcal{G}$, the classifier, $\mathcal{C}$, learns the correct labels assigned to the 14 trusted users.

During an initial authentication session by other means, the pilot subcarriers from the transmitters are measured and recorded for training the SGAN. Initially authenticating by other means, higher protocol layers are used, however for subsequent packet transfers, authenticating at the physical layer reduces the workload on these higher-layer protocols as discussed in [30]. An attack during the initial authentication is certainly a vulnerability, but corruption in the training data may cause authentication to fail shortly after communication begins. This is an aspect to be explored in future research.

The classifier provides identification only after the discriminator successfully authenticates. The discriminator authenticates when a sample is assessed to be "Real". The discriminator may make an incorrect authentication decision (denying authentication when the sample is "Real" or authenticating when the sample was actually faked), therefore we explored how SNR can affect discriminator accuracy.

5.1. SGAN architecture

The adversarial competition in the SGAN is a minimax game described by (4) where the discriminative model attempts to correctly identify authentic training samples from a distribution produced by CSI matrix elements, $p_{data}(h_{n,m})$, and fake training samples created by the generator.

As $\mathcal{D}$ and $\mathcal{G}$ adversarially train each other, they learn to improve their individual performance. When the discriminative model correctly identifies fake samples created by the generative model, the generative network will update its parameter weights through backpropagation to make more realistic samples. Likewise, the discriminative model will update its parameter weights when it incorrectly identifies real or fake samples. The results of this training are a generator neural network adept at creating data that closely mimics training data, a discriminator neural network that can identify all but the best fakes, and a classifier neural network that can determine which trusted transmitter produced the received CSI.

Additionally, $\mathcal{C}$ is trained on labeled samples from the training dataset. Although $\mathcal{C}$ does not directly receive unlabeled authentic or fake samples, the weights of $\mathcal{C}$ are affected since it shares weights with $\mathcal{D}$ in the $\mathcal{D}/\mathcal{C}$ implementation.

Best practices from GAN researchers [31] were used to create the SGAN. The architecture for the discriminator and classifier was chosen to enable feature extraction from the input tensor. A reverse architecture was used for the generator to create realistic-looking samples.

5.2. Discriminative Model

The discriminator estimates the probability that a sample came from the training data, rather than the generator. When training begins, the discriminator won't know $p_{data}(x)$, so the accuracy of correctly assigning authentic and fake samples will be near 0.5. The accuracy will increase with more iterations of samples and backpropagation as the authentic data distribution is learned until the generator network creates samples such that the fake sample distribution, $p_g(z)$ optimally matches $p_{data}(x)$. At this point, the accuracy of correctly assigning authentic and fake samples will return to 0.5 since for the optimal discriminator D^*, and fixed generator, G, $D^*_G(x) = \frac{p_{data}(x)}{p_{data}(x)+p_z(z)}$. When $p_{data}(x) = p_z(z)$, $D^*_G(x) = 0.5$. [22]

5.3. Generative Model

Without having direct access to $p_{data}(x)$, the generator attempts to capture this distribution through feedback based on the probabilities the discriminator assigns to generated fake samples [22]. The weights of the generator network are updated via the loss function $J^{(G)}$ so that the generator will create better samples.

5.4. Classifier Model

The classifier shares all but the final activation layer with the discriminator and is trained to determine which of the 14 trusted transmitters will be authenticated. The weights of $\mathcal{D}/\mathcal{C}$ are iteratively updated as $\mathcal{D}$ and $\mathcal{C}$ are trained.

6. Simulation

This section describes the simulation of the system model from Section 5. The dataset for the SGAN is described and results are presented.

6.1. Dataset

A dataset of 224,000 authentic samples was created, where each sample was a $4 \times 4 \times 16$ complex tensor. The training dataset was allocated 70% of the greater dataset, while the remaining 30% was set aside for testing. Every sample started as a position-dependent $4 \times 4 \times 16$ tensor created by the DeepMIMO dataset. For each of the samples, 1,000 additional samples of measurement error in the form of 16 different levels of SNR were generated. Simulating thermal noise in the receiver, decreasing amounts of AWGN were combined with the original signal to produce the 16 different levels of SNR ranging from -10 dB to 20 dB in steps of 2 dB. This was inspired by the technique used by O'Shea et al. to create distortion for the modulation classification task in [24], except we used MATLAB instead of Python and GNU Radio libraries.

The SGAN processed 16 subcarriers in a MIMO 4×4 configuration with 14 trusted transmitters. Therefore, the classifier model would need to have 16×16 complex inputs and 1 real output for each transmitter label. However, we separated the real and imaginary parts for processing through the neural networks, resulting in inputs tensors of shape $16 \times 2 \times 16$. The discriminative model also has inputs of shape $16 \times 2 \times 16$ and 1 real output denoting "Real" or "Fake", while the generative model has 1 real input and $16 \times 2 \times 16$ outputs. Additionally, the values of the real and imaginary parts are preprocessed to scale $[-1, 1]$ to allow for the $tanh$ activation function range in the generator network as mentioned in the Section 6.2.

To mimic the malicious user's attempt to fool a legitimate user, CSI is generated for a user position in the center of the group of legitimate users, as shown in Fig. 3. This sample is preprocessed as before, to include creating 1,000 samples of 16 different levels of SNR. These samples are then added to the testing dataset, remaining unknown to the SGAN until testing following the completion of training.

6.2. SGAN development

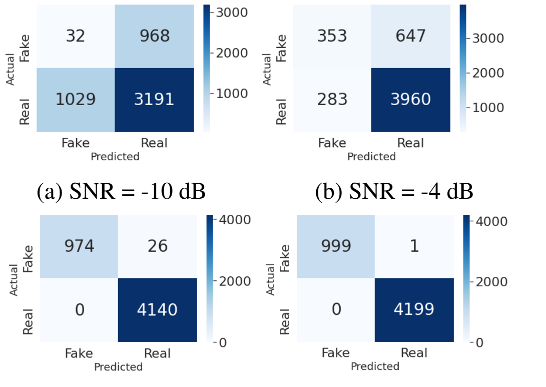
The SGAN was implemented using the Python programming language, Keras [32] front-end, and Tensorflow [33] back-end. Additionally, Numpy, and Matplotlib Python libraries were used. The dataset was created using MATLAB and Python.

The discriminator/classifier network, $\mathcal{D}/\mathcal{C}$, is a dense or fully connected deep neural network (DNN) with 16 inputs of size 2×16 merged into one *Concatenated* layer. Each input has 2 nodes to accommodate the real and imaginary parts of the complex CSI matrix element. Nine additional fully connected layers with *LeakyReLU* activations (*alpha* = 0.3) follow. All hidden layers use *Dropout* of 0.5 to prevent overfitting. Prior to the output layers, a fully connected layer of 14 is used to capture the number of transmitters to be classified. The discriminator output layer of size 1 is fully connected and uses a custom activation $D(x) = \frac{Z(x)}{Z(x)+1}$, where $Z(x) = \sum_{n=1}^{N} \exp[l_n(x)]$ to provide values [0.0, 1.0) as discussed in Section 3. The classifier output is a *softmax* activation connected to the 14 node layer. The learning rate for $\mathcal{D}/\mathcal{C}$ was 0.00009 using the *Adam* [34] optimizer and training was done with batches of 128 samples.

The generator network, $\mathcal{G}$, has a single input with 5 nodes fully connected to the first hidden layer of size 16. Seven additional hidden layers are again fully connected using *LeakyReLU* (*alpha* = 0.3). The last hidden layers are 16 fully connected layers of size 32 followed by *tanh* activations. Finally, the output is reshaped to produce 16 output layers of size 2×16. The learning rate for $\mathcal{G}$ was 0.00009 using the *Adam* optimizer.

6.3. Results

Training was conducted over the course of 20 epochs. Of the 156,800 samples in the training dataset, just over 10% (15,988) were labeled. These labeled samples trained the classifier to identify the legitimate user. When selecting the labeled samples, care was taken to ensure an equal distribution of samples for each of the 14 legitimate users, however the SNR levels in the samples for each of the users was left to chance. All the training samples as well as those created by the generator were used to train the discriminator. Following training, the classifier and discriminator networks and their respective weights were saved. For testing, the classifier and discriminator networks and weights were reloaded and presented with the test dataset.



(a) SNR = -10 dB (b) SNR = -4 dB

(c) SNR = 2 dB (d) SNR = 4 dB

Figure 4. SGAN dense discriminator performance with SNR levels at (a) -10 dB, (b) -4 dB, (c) 2 dB, (d) 4 dB.

The test dataset for the discriminator contained additional samples associated with the malicious user. Figs. 4 and 5 show that the discriminator performs well for SNR levels greater than 2 dB. The confusion matrices in Fig. 4 show the discriminator labeled the malicious user's CSI as "Real" for low SNR levels, but gradually began to correctly categorize them as "Fake" as the SNR level increases. At each SNR level, there are 1,000 "Fake" samples, however the number of "Real" samples varies slightly due to the random split of the original dataset into training and testing components. For authentication, if there is an error it is likely more favorable to have a false negative rather than a false positive. Although this can be frustrating for the authentic user denied authentication and result in lower throughput rates because of restarting the authentication process, malicious users are kept out of the system.

The SGAN-trained densely connected discriminator was accurately able to differentiate "Real" from "Fake" at SNR values above 4 dB. This result shows the limitations of the SGAN approach. The quality of the generator is one aspect that determines how well the discriminator will perform. Training a traditional standalone neural network to differentiate "Real" from "Fake" without a robust generator requires "Fake" samples from another source. While this can be obtained, it is likely not feasible to sample every possible fake CSI sample. The SGAN does not need these negative examples because it creates its own and still performs well provided a sufficient SNR.

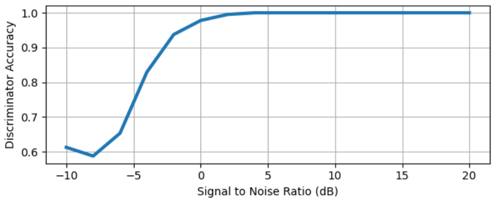
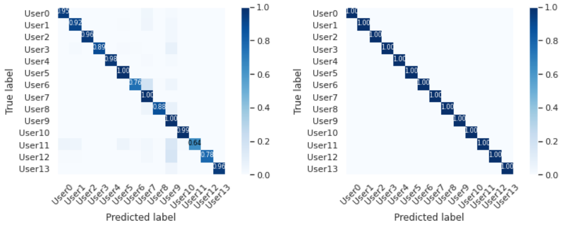
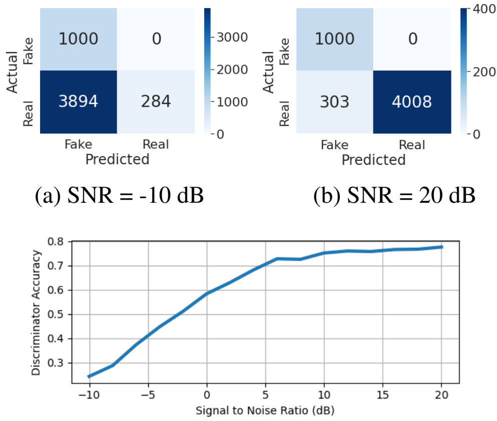


Figure 5. SGAN dense discriminator accuracy vs SNR



(a) SNR = -10 dB (b) SNR = -4 dB

Figure 6. SGAN dense classifier performance with SNR levels at (a) -10 dB, (b) -4 dB.



(a) SNR = -10 dB (b) SNR = 20 dB

(c)

Figure 7. SGAN CNN discriminator performance for (a) -10 dB, (b) 20 dB, (c) -10 dB to 20 dB.

The test dataset without the CSI samples from the malicious user was then used to obtain the performance for the classifier. As shown in Fig. 6, the confusion matrices indicate accurate classification performance even at low SNR values. Accuracy is measured by dividing the correctly classified samples by the sum of the correctly and incorrectly classified samples. Fig. 6(a) shows that the classifier attained classification accuracy above 90% for most of the users at -10 dB SNR, and Fig. 6(b) shows 100% accuracy at -4 dB SNR.

6.4. Additional networks

To compare the performance of the SGAN dense classifier, we constructed three additional networks. First, we used another SGAN classifier, but use convolutional layers instead of fully connected layers. This gives us a SGAN convolutional neural network

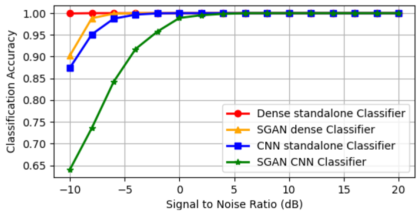


Figure 8. Classifier accuracy vs SNR.

(CNN) classifier. Next, instead of training in a SGAN architecture, we created a standalone dense classifier. This classifier uses the same parameters as our SGAN dense classifier, $\mathcal{C}$. Finally, we implemented a standalone CNN classifier, using the same parameters of the SGAN CNN classifier.

By training the SGAN CNN classifier we also trained a SGAN CNN discriminator. Unfortunately the CNN discriminator did not perform as well as the SGAN dense discriminator. As shown in Fig. 7, the CNN discriminator did not correctly identify all the legitimate users as “Real”. However, at all SNR values, the CNN was able to identify the malicious user CSI as “Fake”, so there may be a use case where this is desirable behavior even though it prevents some number of users from successfully authenticating when they should.

Fig. 8 shows the results of the various classifier testing after training. All the neural networks reach 100% accuracy with sufficient SNR. The standalone dense classifier trained for 125 epochs and obtained almost 100% accuracy for all SNR levels except -10 dB. At -10 dB, the standalone dense classifier was 99.929% accurate. The standalone CNN classifier trained for 667 epochs and had very similar performance to the SGAN dense classifier. Finally, the SGAN CNN classifier trained for 30 epochs, and lagging the others, reached 100% accuracy at 6 dB.

Where the discriminators were not able to differentiate between legitimate and generator-produced samples with increased noise levels, the classification results show that the classifiers are able to differentiate among users at these same SNR values. The reason for this is that the generators’ samples at low SNR closer approximate the sample distribution from the authentic dataset, making training difficult for the discriminators. Contrast with the classifiers’ training where they only receive samples from the dataset and learns the features relevant to the 14 transmitters’ CSI.

7. Conclusion and future work

We showed how the use of a SGAN can be used to discriminate and classify transmitters by multiple-subcarrier MIMO CSI as a method to provide physical layer authentication. Our simulation results illustrated that with a very small percentage of labeled CSI samples, accurate discrimination between legitimate and adversary transmitters as well as classification can be made with a SGAN dense classifier for SNR values greater than 4 dB. An adversary may achieve a high degree of accuracy when spoofing a legitimate transmitter, but by retaining the magnitude and phase of the CSI elements, we have shown that our system can differentiate transmitter CSI from positions 10 cm apart.

We saw that the SGAN-trained classifiers required less epochs to train, however the standalone dense classifier had the best performance overall. However, the standalone classifiers only classify legitimate transmitters, while the SGAN is able to first discriminate and then classify. We explored the use of a GAN-trained classifier to discover if this is a more accurate method for classification based on CSI at varying SNR levels. Future research includes interference by an adversary during the authentication process where the original training dataset would be built. Additionally, we intend to explore transfer learning from a SGAN-trained classifier to a standalone network in an attempt to reduce the training time of the standalone classifer, while increasing overall performance.

References

[1] R. Fantacci, L. Maccari, T. Pecorella, and F. Frosali, “Analysis of secure handover for IEEE 802.1x-based wireless ad hoc networks,” *IEEE Wireless Communications*, vol. 14, no. 5, pp. 21–29, Oct. 2007.

[2] X. Wang, P. Hao, and L. Hanzo, “Physical-layer authentication for wireless security enhancement: current challenges and future developments,” *IEEE Communications Magazine*, vol. 54, no. 6, pp. 152–158, Jun. 2016.

[3] D. Fang, Y. Qian, and R. Q. Hu, “Security for 5G Mobile Wireless Networks,” *IEEE Access*, vol. 6, pp. 4850–4874, 2018.

[4] F. Meneghello, M. Calore, D. Zucchetto, M. Polese, and A. Zanella, “IoT: Internet of Threats? A Survey of Practical Security Vulnerabilities in Real IoT Devices,” *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 8182–8201, Oct. 2019.

[5] A. Odena, “Semi-Supervised Learning with Generative Adversarial Networks,” *arXiv:1606.01583 [cs, stat]*, Oct. 2016, arXiv: 1606.01583. [Online]. Available: http://arxiv.org/abs/1606.01583

[6] A. C. Polak, S. Dolatshahi, and D. L. Goeckel, “Identifying Wireless Users via Transmitter Imperfections,” *IEEE Journal on Selected Areas in Communications*, vol. 29, no. 7, pp. 1469–1479, Aug. 2011.

[7] N. Al Khanbashi, N. Al Sindi, S. Al-Araji, N. Ali, Z. Chaloupka, V. Yenamandra, and J. Aweya, “Real

time evaluation of RF fingerprints in wireless LAN localization systems," in *2013 10th Workshop on Positioning, Navigation and Communication (WPNC)*, Mar. 2013, pp. 1–6.

[8] L. Xiao, L. Greenstein, N. Mandayam, and W. Trappe, "A Physical-Layer Technique to Enhance Authentication for Mobile Terminals," in *2008 IEEE International Conference on Communications*, May 2008, pp. 1520–1524, iSSN: 1550-3607, 1938-1883.

[9] A. Alkhateeb, "DeepMIMO: A Generic Deep Learning Dataset for Millimeter Wave and Massive MIMO Applications," in *Proc. of Information Theory and Applications Workshop (ITA)*, San Diego, CA, Feb. 2019.

[10] Remcom, "Wireless InSite," https://www.remcom.com/wireless-insite.

[11] L. Xiao, L. Greenstein, N. Mandayam, and W. Trappe, "Fingerprints in the Ether: Using the Physical Layer for Wireless Authentication," in *2007 IEEE International Conference on Communications*, Jun. 2007, pp. 4646–4651.

[12] K. Yu and B. Ottersten, "Models for MIMO propagation channels: a review," *Wireless Communications and Mobile Computing*, vol. 2, no. 7, pp. 653–666, 2002. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/wcm.78

[13] W. C. Jakes, Ed., *Microwave mobile communications*, nachdr. ed., ser. An IEEE Press classic reissue. New York, NY: IEEE Press [u.a.], 1995, oCLC: 249569885.

[14] A. Goldsmith, *Wireless Communications*. Cambridge University Press, Aug. 2005.

[15] Y. Chapre, A. Ignjatovic, A. Seneviratne, and S. Jha, "CSI-MIMO: Indoor Wi-Fi fingerprinting system," in *39th Annual IEEE Conference on Local Computer Networks*, Sep. 2014, pp. 202–209, iSSN: 0742-1303.

[16] C. Nerguizian, C. Despins, and S. Affes, "Geolocation in mines with an impulse response fingerprinting technique and neural networks," in *IEEE 60th Vehicular Technology Conference, 2004. VTC2004-Fall. 2004*, vol. 5, Sep. 2004, pp. 3589–3594 Vol. 5, iSSN: 1090-3038.

[17] J. Xiao, K. Wu, Y. Yi, and L. M. Ni, "FIFS: Fine-Grained Indoor Fingerprinting System," in *2012 21st International Conference on Computer Communications and Networks (ICCCN)*, Jul. 2012, pp. 1–7, iSSN: 1095-2055.

[18] X. Wang, L. Gao, S. Mao, and S. Pandey, "CSI-Based Fingerprinting for Indoor Localization: A Deep Learning Approach," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 1, pp. 763–776, Jan. 2017.

[19] Q. Wang, H. Li, D. Zhao, Z. Chen, S. Ye, and J. Cai, "Deep Neural Networks for CSI-Based Authentication," *IEEE Access*, vol. 7, pp. 123 026–123 034, 2019, conference Name: IEEE Access.

[20] R.-F. Liao, H. Wen, J. Wu, F. Pan, A. Xu, Y. Jiang, F. Xie, and M. Cao, "Deep-Learning-Based Physical Layer Authentication for Industrial Wireless Sensor Networks," *Sensors (Basel, Switzerland)*, vol. 19, no. 11, May 2019. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6603790/

[21] F. Pan, Z. Pang, M. Luvisotto, X. Jiang, R. N. Jansson, M. Xiao, and H. Wen, "Authentication Based on Channel State Information for Industrial Wireless Communications," in *IECON 2018 - 44th Annual Conference of the IEEE Industrial Electronics Society*, Oct. 2018, pp. 4125–4130.

[22] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative Adversarial Nets," in *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, ser. NIPS'14. Cambridge, MA, USA: MIT Press, 2014, pp. 2672–2680, event-place: Montreal, Canada. [Online]. Available: http://dl.acm.org/citation.cfm?id=2969033.2969125

[23] T. J. O'Shea, T. Roy, N. West, and B. C. Hilburn, "Physical Layer Communications System Design Over-the-Air Using Adversarial Networks," in *2018 26th European Signal Processing Conference (EUSIPCO)*, Sep. 2018, pp. 529–532.

[24] T. J. O'Shea, J. Corgan, and T. C. Clancy, "Convolutional Radio Modulation Recognition Networks," in *Engineering Applications of Neural Networks*, ser. Communications in Computer and Information Science, C. Jayne and L. Iliadis, Eds. Cham: Springer International Publishing, 2016, pp. 213–226.

[25] M. Li, G. Liu, S. Li, and Y. Wu, "Radio Classify Generative Adversarial Networks: A Semi-supervised Method for Modulation Recognition," in *2018 IEEE 18th International Conference on Communication Technology (ICCT)*, Oct. 2018, pp. 669–672, iSSN: 2576-7828.

[26] D. Roy, T. Mukherjee, M. Chatterjee, E. Blasch, and E. Pasiliao, "RFAL: Adversarial Learning for RF Transmitter Identification and Classification," *IEEE Transactions on Cognitive Communications and Networking*, pp. 1–1, 2019.

[27] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, "Improved Techniques for Training GANs," in *Advances in Neural Information Processing Systems 29*, D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, Eds. Curran Associates, Inc., 2016, pp. 2234–2242. [Online]. Available: http://papers.nips.cc/paper/6125-improved-techniques-for-training-gans.pdf

[28] MATLAB, *version 9.6.0.1072779 (2019a)*. Natick, Massachusetts: The Mathworks Inc., 2019.

[29] Y. Shi, K. Davaslioglu, and Y. E. Sagduyu, "Generative Adversarial Network for Wireless Signal Spoofing," in *Proceedings of the ACM Workshop on Wireless Security and Machine Learning - WiseML 2019*. Miami, FL, USA: ACM Press, 2019, pp. 55–60. [Online]. Available: http://dl.acm.org/citation.cfm?doid=3324921.3329695

[30] L. Xiao, A. Reznik, W. Trappe, C. Ye, Y. Shah, L. Greenstein, and N. Mandayam, "PHY-Authentication Protocol for Spoofing Detection in Wireless Networks," in *2010 IEEE Global Telecommunications Conference GLOBECOM 2010*, Dec. 2010, pp. 1–6.

[31] S. Chintala, "How to train a GAN," *NIPS 2016 Workshop on Adversarial Training*, 2016. [Online]. Available: https://github.com/soumith/ganhacks,https://www.youtube.com/watch?v=myGAju4L7O8

[32] F. Chollet, et al., *Keras*, 2015. [Online]. Available: https://keras.io

[33] M. Abadi, et al., "TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems," 2015. [Online]. Available: https://www.tensorflow.org/

[34] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," *arXiv:1412.6980 [cs]*, Jan. 2017, arXiv: 1412.6980. [Online]. Available: http://arxiv.org/abs/1412.6980