Detection of Rogue RF Transmitters using Generative Adversarial Nets

Debashri Roy*, Tathagata Mukherjee†, Mainak Chatterjee*, Eduardo Pasiliao‡

* Computer Science
University of Central Florida
Orlando, FL 32826
{debashri, mainak}@cs.ucf.edu

† Computer Science
University of Alabama
Huntsville, AL 35899
{tm0130}@uah.edu

‡ Munitions Directorate
Air Force Research Laboratory
Eglin AFB, FL, 32542
{eduardo.pasiliao}@us.af.mil

***Abstract*—Understanding and analyzing the radio frequency (RF) environment have become indispensable for various autonomous wireless deployments. To this end, machine learning techniques have become popular as they can learn, analyze and even predict the RF signals and associated parameters that characterize a RF environment. However, classical machine learning methods have their limitations and there are situations where such methods become ineffective. One such setting is where active adversaries are present and try to disrupt the RF environment through malicious activities like jamming or spoofing. In this paper we propose an *adversarial learning technique* for identifying rogue RF transmitters and classifying trusted ones by designing and implementing generative adversarial nets (GAN). The GAN exploits the in-phase (I) and quadrature imbalance (i.e., the *IQ imbalance*) present in all transmitters to learn the unique high dimensional features that can be used as "fingerprints" for identifying and classifying the transmitters. We implement a generative model that learns the sample space of the IQ values of the known transmitters and use the learned representation to generate fake signals that imitate the transmissions of the known transmitters. We program 8 universal software radio peripheral (USRP) software defined radios as trusted transmitters and collect over-the-air raw IQ data from them using a RTL-SDR in a laboratory setting. We also implement a discriminator model and show that the discriminator is able to discriminate between the trusted transmitters from fake ones with 99.9% accuracy. Finally, the trusted transmitters are classified using convolutional neural network (CNN) and fully connected deep neural networks (DNN). Results reveal that the CNN and DNN are able to correctly discriminate between the 8 trusted transmitters with 81.6% and 96.6% accuracies respectively.**

Keywords: RF fingerprinting, GAN, machine learning, deep neural network, IQ imbalance, USRP, confusion matrix.

I. Introduction

Localization, identification, and characterization of radio frequency (RF) signal sources (aka RF transmitters) are indispensable for applications such as locating a cell phone, identifying a jammer, detecting the presence or absence of a signal, tracking objects, etc. Localization of a RF transmitter is a well studied problem where the received power at a receiver is utilized to estimate the distance to the transmitter, given some known transmitters and path-loss models (see [1] and the references therein). With more and more autonomous deployments of wireless networks, identification of transmitters has also become important. For example, a wireless sensor network relies on trustworthy signals; however, malicious transmitters can contaminate the signals and jeopardize the utility of the sensor network. Existence of such threats underscore the need for techniques that recognize and authenticate transmitter identity irrespective of the network protocols and the communication technologies being used. However, correctly identifying a transmitter and being able to characterize it in real time remains a challenging problem.

Use of secure mechanisms to authenticate RF transmitters has been a common way to identify malicious transmitters. However, implementation of such secure mechanisms adds to the computation and communication overhead for real time systems such as connected autonomous vehicles. Recent developments in radio frequency machine learning (RFML) systems have given rise to the possibility of using these methods for automated real time authentication of RF transmitters. These methods can also be used in adversarial settings for tasks such as the identification of malicious transmitters by through the use of learning powered transmitter forensics [2].

Unlike the image or speech processing domain, where machine learning techniques have been widely successful, learning in the RF domain is just beginning to see some breakthroughs. For processing images, spatial correlations and knowledge of previously observed objects make future predictions possible. Similarly, in speech recognition recognizable patterns emerge from known sequences that can be used to synthesize phrases and words. In essence such techniques are built on prior distributions and patterns that make generalizations possible. However, such techniques cannot be easily extended to the RF domain because of the unpredictable and varied nature of the RF signals. To make matters worse, the presence of adversaries make it even more difficult to learn and characterize RF signals. For one smart adversaries can spoof transmitters and introduce noise in the transmission channels making it harder to learn unique characteristics of the transmitters, in essence manipulating the learning phase to render the detection model ineffective. Thus, application of RFML systems for RF synthesis and recognition has become an interesting and open research area.

For machine learning techniques to be effective, one must choose an attribute or feature that is unique to a transmitter, irrespective of the signals it transmits. One such commonly used feature is the *rise time* signature that is generated by slight variations of the component values during the manufacturing process. Though the rise time signature has mostly been used

978-1-5386-7646-2/19/$31.00 ©2019 IEEE

for signal intelligence and by regulatory agencies, it performs poorly in the presence of malicious entities. Moreover, the rise time signature is commercially unavailable , hence cannot be used as a standard signature. '*IQ imbalance*' is another feature that is affected at the time of manufacturing due to the use of noisy mixers, oscillators and unbalanced low pass filters [3]. This is the imbalance between the in-phase (I) and quadrature (Q) components of a signal which is the result of interaction of the radio frequency with the local oscillator frequency which is required to get the intermediate frequency. Though there are techniques to compensate for this imbalance [4], the fact remains that all transceivers exhibit this *unique* IQ imbalance. The IQ imbalance depends on the choice of the hardware components used, and is an unwanted byproduct of the manufacturing process that is hard to imitate. This imbalance can be used as a basis for feature engineering (automated or otherwise) for transmitter identification and recognition. However, this fact is also known by the adversaries. Thus, their target would be to learn and estimate the probability distribution of the training data used for model creation, given a particular sample space. The adversaries can use a *generative model* to generate fake signals so as to spoof the transmission of known transmitters. Generative Adversarial Nets (GAN) [5] uses a generative model which enables the realistic generation of samples from a given distribution which can then be used to train a discriminator for identifying real samples drawn from fake ones obtained from the generator.

In this paper, we use generative adversarial nets (GAN) to detect rogue transmitters. Unlike most machine learning techniques, GAN has been designed to learn in adversarial situations. We propose and implement a generative model and a discriminative model to (i) detect unknown and/or rogue transmitters and (ii) classify the known transmitters. We leverage the fact that the IQ imbalance for every radio transmitter is unique and could be exploited by the GAN to generate unique features or fingerprints. The main contributions of this paper are:

- We propose a generative model that uses a deep neural network (DNN) for generating (fake) signals that very closely resemble real signals. The generator proceeds by reducing the parameter space, replicates the time-invariant features and serves as a compact front-end for fake radio signal generation.
- We propose a discriminative model using a deep neural network that takes input from the known signal transmitters as well as the fake signals from the generative model. The purpose of this discriminative model is to distinguish the real signals from the fake ones. The outcome of the decision process is fed to the generative model, allowing the adversary (generator) to update its model so as to better generate fake signals.
- Once we detect the trusted transmitters from the rogue ones, we use supervised training models to classify the trusted transmitters. We design a convolution neural network (CNN) for that purpose, which leverages the correlation between the complex-valued IQ data constellations. We design another deep neural network to improve the accuracy of the CNN.
- Our models have been validated on a laboratory test bed consisting of several universal software radio peripheral (USRP) B210s [6] and one RTL-SDR receiver [7]. The USRPs transmitted signals on a particular frequency which were received by the RTL-SDR. The collected dataset had 1024 complex IQ samples per timestamp, generating 2048 features. The generative and discriminative models were trained and tested on the collected dataset. The unique pattern of variation of the IQ imbalances for each radio is captured as features by the multiple layers of the neural network.
- The novelty of the proposed work lies in accurately modeling and implementing the proposed generative and discriminative models on real hardware using raw IQ data. To the best of our knowledge, this is the first paper that uses GANs to identify adversarial RF signals and for fingerprinting radio transmitters.

Next we describe the current and previous work in machine learning based transmitter identification.

II. Background and Related Work

During the process of designing and manufacturing cheap radio hardware certain imperfections have become the norm. The IQ imbalance is one such imperfection that is unique to different radio hardware and are caused by imperfections in local oscillators and mixers. As a result of this, the in-phase (I) and quadrature (Q) components of the modulator are not orthogonal. When a signal is transmitted using a particular radio transmitter, some IQ imbalance is imposed over the complex-valued IQ data during transmission [8] as shown in Fig. 1. IQ imbalance leads to performance degradation for higher order modulations because the symbol rotation becomes more sensitive with increasing number of constellations towards I and Q branches [9].

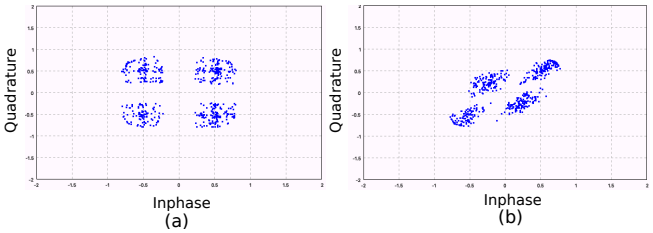


Fig. 1. IQ Imbalance for QPSK: (a) Before (b) After 45° Phase Imbalance

In recent years, there have been some efforts at using machine learning techniques for fingerprinting RF transmitters. In [10], O'Shea et al presented a radio modulation classification method using naively learned features. They have shown that blind temporal learning on densely encoded time series using convolutional neural networks is a viable approach. However, this method did not perform well in the low signal to noise ratio (SNR) regime. In [11], the authors have presented an unsupervised learning technique

using convolutional autoencoders which can eventually learn the basics of modulation functions and then leverage that to recognize different digital modulation schemes. They also proposed a method to evaluate the quantitative metrics on reconstructed data to recognize the schemes. Another study for modulation classification using raw IQ samples was presented in [12]. A method for modulation classification was proposed in [13] for a distributed wireless spectrum sensing network. The authors proposed a recurrent neural network model using long short term memory (LSTM) cell, yielding 90% accuracy for synthetic dataset [14].

An in-depth study on the performance of deep learning based radio signal classification was proposed in [15]. The authors considered 24 modulation schemes with a rigorous baseline method that uses higher order moments and strong boosted gradient tree classification for detection. The authors also applied their method on real over-the-air data collected by Software Defined Radios (SDRs). An approach based on the concept of adversary was proposed in [2] for synthesizing new physical layer modulation and coding schemes. The adversarial approach is targeted to learn the channel response approximations in any arbitrary communication system, enabling the design of a smarter channel autoencoder. All these proposed approaches demonstrate how difficult it is to make machine learning techniques effective in the RF domain.

Motivated by the above mentioned works, we focus on transmitter identification in the presence of adversaries. The idea of training discriminative models via an adversarial process was first proposed by Goodfellow [5]. Since then, generative adversarial nets (GAN) have been adopted for various fields and applications, particularly for image processing where GANs have proved their efficacy [16]–[18].

III. Proposed GAN for RF Fingerprinting

Recent advances in neural networks have made it possible to obtain robust models with low generalization errors by training "deep" neural architectures efficiently. The "depth" signifies the number of iterative operations performed on the input data using each layer's transfer function and deeper architectures allow the network to learn robust feature representations from the input data. Though such techniques demand higher computation and involve complicated layer-by-layer backpropagation, nevertheless, most deep learning systems are able to perform training on deep networks using some variation of gradient decent with adaptive learning rates (e.g., *Adam* [19]), regularization to avoid overfitting (e.g., *Dropout* [20]) and the use of backpropagation. Our intention is to design neural network models that can train in the presence of adversaries and discriminate between fake and known transmitters through automatic fingerprinting.

A. Proposed GAN Architecture

The GAN framework has two primary models, a generative model ($\mathcal{G}$) that generates fake data from a given data distribution, and a discriminative model ($\mathcal{D}$) that estimates the probability that a sample came from the training data rather than

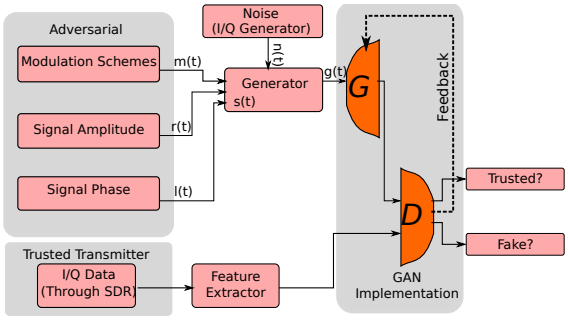


Fig. 2. Proposed GAN architecture

$\mathcal{G}$. The proposed GAN architecture is shown in Fig. 2. $\mathcal{G}$ and $\mathcal{D}$ are the generative and discriminative models respectively. The adversary generates random modulation scheme ($m(t)$), signal amplitude ($r(t)$), and phase ($l(t)$) and mixes additive white Gaussian noise ($n(t)$) with that. The generated signal ($g(t)$) which is initially random in nature improves over time as the generator learns from the discriminator and improves on its accuracy to imitate real data. On the other hand, the discriminator ($\mathcal{D}$) gets input from both the generator ($\mathcal{G}$) and `Trusted` transmitters. This helps it to learn to differentiate between real and fake inputs. The known transmitter data is collected and fed to the discriminator ($\mathcal{D}$) as raw IQ values.

Overall, the target is to train $\mathcal{G}$ in such a way that will maximize the probability of $\mathcal{D}$ making a mistake. $\mathcal{G}$ tunes its hyper parameters with the feedback from $\mathcal{D}$. We argue, GAN is an efficient way to generate correlated data samples and thereby approach an accurate generative model– something the rogue transmitters aim to achieve. Once the model is trained, the generated signals are synthesized to mimic rogue transmitters based on the sample space of IQ signal data from the known transmitters.

B. The Generative Model

As far as the generator is concerned, the overall problem can be treated as an N-class decision problem where the input is a complex base-band time series representation of the received signal. That is, the dataset is the in-phase and quadrature components of a radio signal obtained at discrete time periods through analog to digital conversion with a carrier frequency to obtain a $1 \times N$ complex valued vector. Classically, this is written as:

$$s(t) = c_1 m(t) + c_2 r(t) + c_3 l(t) \tag{1}$$

where $s(t)$ is a continuous time series signal modulated onto a sinusoid with either varying frequency, phase, amplitude, trajectory, or some permutation of multiple parameters. Here, $m(t)$, $r(t)$, and $l(t)$ are the time series continuous signals for modulation, amplitude, and phase respectively, selected randomly by the generator. The coefficients c_1, c_2, and c_3 are some path loss or constant gain terms associated with $m(t)$, $r(t)$, and $l(t)$ respectively. The output $g(t)$ is obtained as:

$$g(t) = s(t) + n(t) \tag{2}$$

where $n(t)$ is the additive Gaussian white noise. The output $g(t)$ is then fed to a generator which is used as an unsupervised learning tool as a part of the generative network. The generator learns the probability distribution $p_g(\mathbf{x})$ over sample space $(\mathbf{x})$ of the input. In this case, $\mathbf{x}$ is the sample space of IQ values.

C. The Discriminative Model

The discriminative model learns by minimizing a cost function during training. The cost function, $C(\mathcal{G};\mathcal{D})$, depends on both the generator $(\mathcal{G})$ and the discriminator $(\mathcal{D})$. It is formulated as $C(\mathcal{G};\mathcal{D}) = \mathbb{E}_{p_{data}(\mathbf{x})} \log \mathcal{D}(\mathbf{x}) + \mathbb{E}_{p_g(\mathbf{x})} \log(1-\mathcal{D}(\mathbf{x}))$, where $p_g(\mathbf{x})$ is the generator's distribution over $\mathbf{x}$, $p_{data}(\mathbf{x})$ is the data distribution over $\mathbf{x}$, $\mathcal{D}(\mathbf{x})$ is the probability that $\mathbf{x}$ came from $p_{data}(\mathbf{x})$ than $p_g(\mathbf{x})$. The training is formulated as:

$$\max_{\mathcal{D}} \min_{\mathcal{G}} C(\mathcal{G};\mathcal{D}) \tag{3}$$

For the GAN framework there is an unique optimal discriminator for a fixed generator, $\mathcal{D}^*(\mathbf{x}) = \dfrac{p_{data}(\mathbf{x})}{p_{data}(x) + p_g(\mathbf{x})}$. It is also inferred that $\mathcal{G}$ is optimal when $p_g(\mathbf{x}) = p_{data}(\mathbf{x})$, i.e., the generator is optimal when the discriminator cannot distinguish real samples from fake ones. Similarly, the $\mathcal{D}$ is optimal when the discriminator can recognize each real sample from fakes.

IV. Testbed Evaluation

In order to validate the proposed GAN, we implemented the generator and discriminator models using data from universal software radio peripheral (USRP) and conducted indoor experiments to distinguish between 8 *similar* transmitters.

A. Signal Generation and Data Collection

In order to learn the features of *similar* transmitters, we used eight USRPs of the same kind, namely B210 from Ettus Research [6]. The signal generation and reception are shown in Fig. 3. The B210s were programmed to transmit random data on 904 MHz using Quadrature Phase Shift Keying (QPSK) modulation. Then the modulated signal was transmitted through the USRP sink block. We used GNURadio [21] for signal processing and data transmission. The flow graph is presented in Fig. 4. For the receiver, we used a RTL-SDR [7] which captured over-the-air raw IQ data and stored them on file.

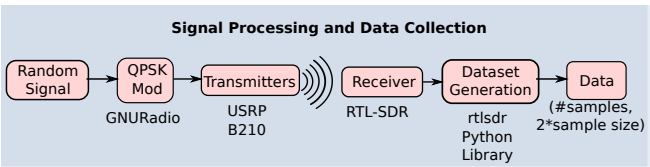


Fig. 3. Signal Generation and Data Collection Technique

We collected the IQ signal data with a sample size of 1024. Each data sample had 2048 entities consisting of the I and Q values for the 1024 samples. A larger sample size would mean more training examples for the neural network. The choice of 1024 samples was sufficient to capture the unique pattern of IQ imbalances and at the same time it was not computationally

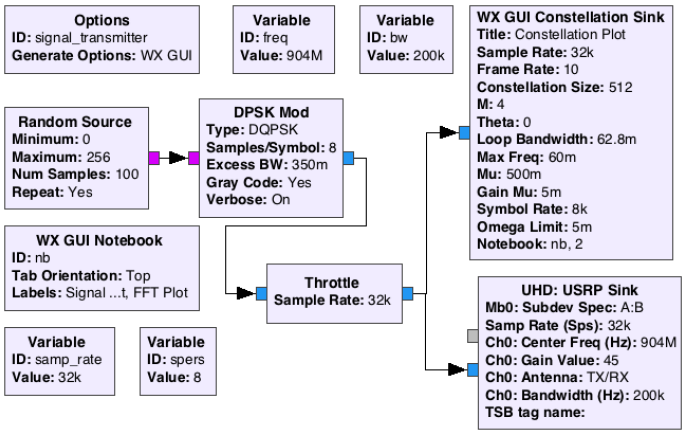


Fig. 4. GNU Radio Flow Graph for Data Collection

Parameters	Values
Transmitter Gain	45 dB
Transmitter Frequency	904 MHz (ISM)
Bandwidth	200 KHz
Sample Size	1024
Samples/Transmitter	40,000
# Transmitters	4 and 8

TABLE I
TRANSMISSION CONFIGURATION PARAMETERS

expensive. We collected 40,000 training examples from each transmitter to avoid the data skewness problem observed in machine learning. The configuration parameters are given in Table I. We had two sets of data: (i) using 4 transmitters: 6.8 GB size, 160K rows and 2048 columns and (ii) using 8 transmitters: 13.45 GB size, 320K rows and 2048 columns.

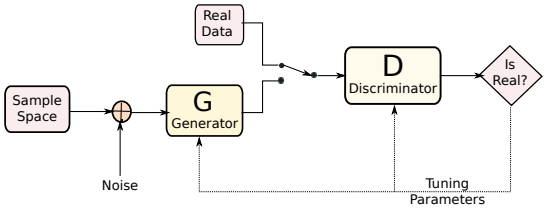


Fig. 5. A Simplified View of GAN Implementation

B. Machine Learning Libraries

There are several libraries and tools that implement deep learning frameworks with support of immensely concurrent GPU architecture that reduce the burden of programming the traditional routines for training of larger neural networks. We use *Keras* [22] as the frontend and *Tensorflow* [23] as the backend. *Keras* is an overlay on neural network primitives with Tensorflow [23] or Theano [24] that provides a customizable interface for quick deployment of complex neural networks. We also use *Numpy*, *Scipy*, and *Matplotlib* Python libraries.

C. Experimental Setup and Performance Metrics

We conducted the experiments on a Ryzen 8 Core system having 64 GB RAM and a GTX 1080 Ti GPU unit having 11 GB memory. We focused on three main aspects:

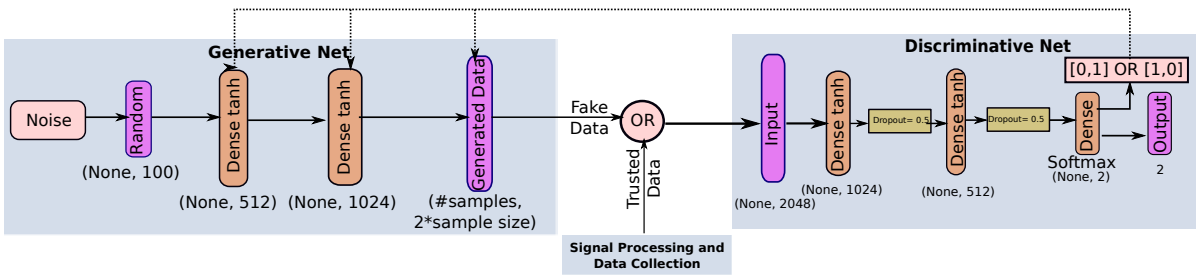


Fig. 6. GAN Implementation for rogue Transmitter Detection

- Designing a generative adversarial net (GAN) to distinguish rogue transmitters from trusted ones.
- Designing a convolutional neural network (CNN) to exploit the correlation in collected signal data of the trusted transmitters.
- Designing a deep neural network (DNN) to classify the trusted transmitters for fingerprinting.

To measure the effectiveness of the proposed neural networks, "accuracy" is used as the typical performance metric. However, accuracy can sometimes be misleading and incomplete when data is skewed. A confusion matrix overcomes this problem by showing how confused the classification model is on its predictions. It provides more insights on the performance by identifying not only the number of errors, but more importantly the types of errors.

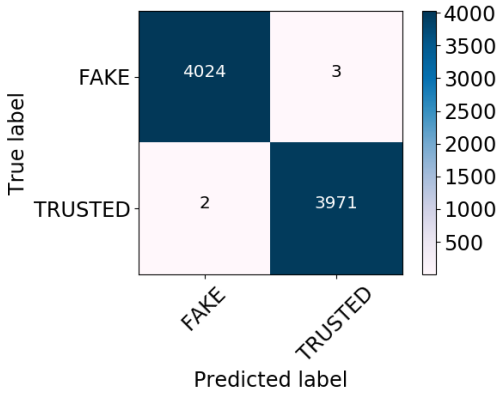


Fig. 7. Confusion Matrix for Determining Trusted and Fake Transmitters

V. GAN Implementation

For implementing the GAN, we use the over-the-air data collected from the trusted transmitters. The generator ($\mathcal{G}$) generates fake data from the same sample space to impersonate as a transmitter. Trusted and fake data are fed to the discriminator ($\mathcal{D}$) with an equal and unbiased probability. We design the discriminator and generator separately, as shown in Fig. 5.

The generator starts with randomly generating data within the sample space [-1,1]. Two *dense* layers of size 512 and 1024 are applied with *tanh* activation function. Then one *dense* layer of $2\times$ sample size (2048 in this case) is invoked with the *sigmoid* activation function. $\mathcal{G}$ continues to learn the data distribution (p_g) and generating fake samples of size 2048 within the signal IQ values sample space. $\mathcal{D}$ consists of one input layer of 2048 nodes, two hidden layers of 1024 and 512 nodes respectively, and finally a *softmax* output layer of 2 nodes to classify an input as either `Fake` or `Trusted`. We use *tanh* as activation function at the hidden layer and added *Dropout* [20] of 0.5 in between those layers for regularization. The overall GAN implementation is shown in Fig. 6.

We train both the generator and discriminator through iterative sequential learning to strengthen the generative model over time. We use categorical cross-entropy training on *Adam* [19] optimizer for gradient based optimization. We notice that the discriminator was able to detect the fake transmitters with 50% accuracy before the adversarial training. After several epochs (< 50) of adversarial training, the optimal discriminator ($\mathcal{D}^*$) is able to detect the `Fake` transmitters with about 99.9% accuracy, as shown in the confusion matrix in Fig. 7. Note that one epoch consists of a forward and a backward pass through the designed model over the entire dataset. It is clear from the confusion matrix that the number of false negatives and false positives are very low and well within acceptable range [25]. Once the GAN is trained, it will be able to detect rogue transmitters from over-the-air reception of raw IQ data.

A. CNN Implementation

The main motivation for implementing a convolution neural network was to capture the correlation between IQ values of samples. The CNN has three *Conv2D* layers of 1024, 512 and 256 filters, a *Flatten* operation, and three fully connected (*FC*) layers of 512, 256 and 8 nodes as shwon in Fig. 8. We use Dropout [20] of 0.25 and 0.5 after each *conv2D* and *dense* layer respectively. We use kernel size of (2,3) and stride of (2,2) at each *Conv2D* layer. We also apply a pooling layer *MaxPooling2D* after each *Conv2D* layer with pool size of (2,2) and stride of (2,2). We use *ReLU* [26] activation for all convolution and fully connected layers, other than the *softmax* layer of the output nodes.

We obtain 89.07% and 81.6% accuracy for 4 and 8 transmitter classification respectively using the aforementioned CNN. The low accuracies are due to the poor correlation among samples from the same radio. The accuracy plots and confusion matrices for both the cases are shown in Figs. 9 and 10. Both training and validation accuracy increase with the number of

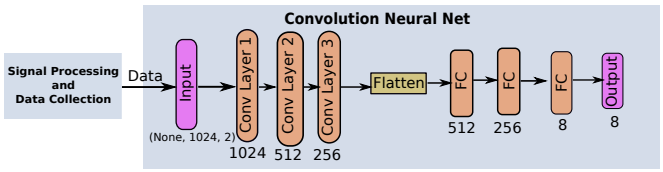
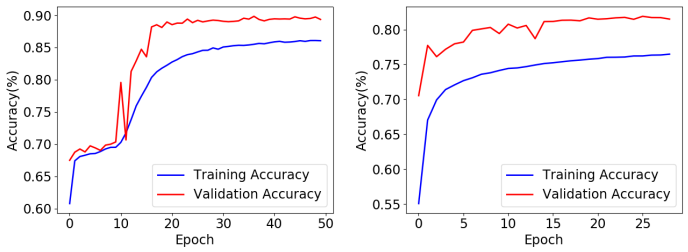


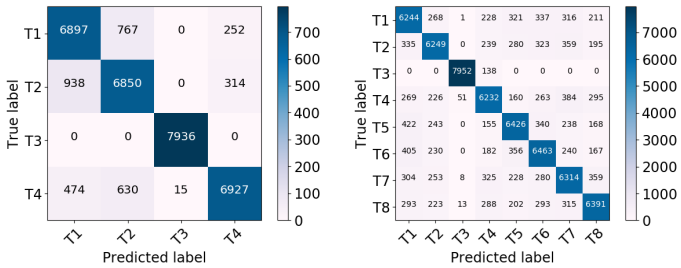
Fig. 8. CNN Implementation for Transmitter Classification

epochs. With the CNN, the number of false positives and false negatives are somewhat high for the predicted versus the true labels. So, we proceed to build a deep network to achieve better accuracy for a trusted transmitter classifier.



(a) 4 Transmitters (b) 8 Transmitters

Fig. 9. Accuracy Plot for Transmitter Classification using CNN



(a) 4 Transmitters (b) 8 Transmitters

Fig. 10. Confusion Matrix for Transmitter Classification using CNN

B. DNN Implementation

Once the discriminator recognizes the trusted transmitters from the fake ones, we feed the trusted transmitter data to a deep neural network for its classification. The implementation of the DNN is similar to the discriminator model of GAN and is shown in Fig. 11. The only difference is that the *softmax* output layer has 8 nodes to recognize the 8 classes. We use biases and regularization to avoid under- and over-fitting. We use *Adam* [19] based optimization with categorical cross-entropy training. The DNN yields an accuracy of 97.21% for 4 transmitters, and 96.6% for 8 transmitters. The accuracy and confusion matrices are shown in Figs. 12 and 13 respectively. It is evident that the number of false positives and false negatives in the confusion matrices are significantly low for the DNN.

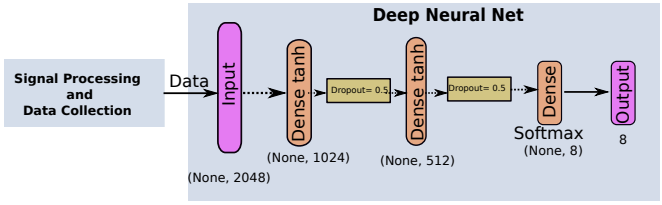
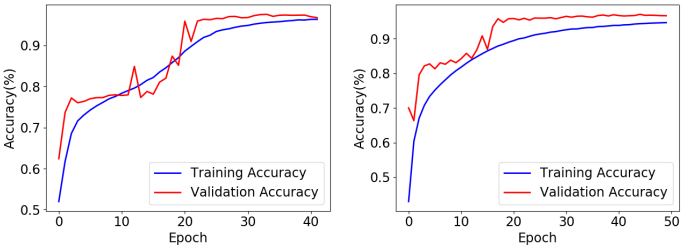


Fig. 11. DNN Implementation for Transmitter Classification



(a) 4 Transmitters (b) 8 Transmitters

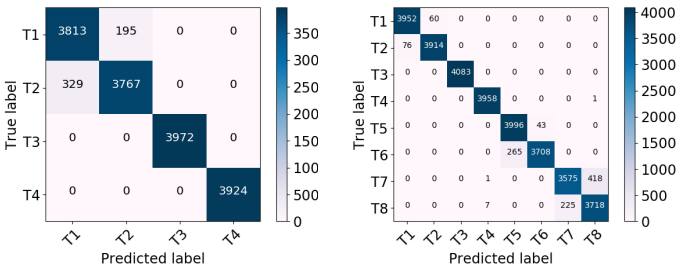
Fig. 12. Accuracy for Transmitter Classification using DNN

C. Comparisons of GAN/CNN/DNN Implementations

Once a transmitter is found to be `Trusted` via the proposed GAN, we use the DNN or the CNN to identify it. We used 90%, 5%, and 5% to train, validate, and test respectively. The overall accuracy of different implementations is shown in Table II. We find that the CNN does not exhibit the best accuracy for transmitter classification, which clearly depicts the lack of correlation between the data. We also conducted experiments by varying the number of transmitters from 2 to 8. In Fig. 14, we present how training and testing accuracy changes with increasing number of transmitters. As expected, the accuracy decreases when there are more classes that a transmitter needs to be mapped to.

The GAN based deep neural network achieves an acceptable accuracy for transmitter identification proving the feasibility of the proposed idea. In summary,

1) GAN network is able to distinguish between `Trusted` and `Fake` RF signals.
2) Convolution neural network yields 81%-86% accuracy



(a) 4 Transmitters (b) 8 Transmitters

Fig. 13. Confusion Matrix for Transmitter Classification using DNN

Dataset (GB)	#Trans	Method	#Parameters	Acc (%)
6.8	4	DNN (5 layers)	6.8 M	97.21
13.45	8	DNN (5 layers)	6.8 M	96.6
6.8	4	CNN (6 layers)	38 M	89.07
13.45	8	CNN (6 layers)	38 M	81.59
6.8	4	GAN (DNN)	3.6 M ($\mathcal{G}$) 6.8 M ($\mathcal{D}$) 10.4 M (GAN)	99.9
13.45	8	GAN (DNN)	3.6 M ($\mathcal{G}$) 6.8 M ($\mathcal{D}$) 10.4 M (GAN)	99.9

TABLE II
ACCURACY FOR DIFFERENT IMPLEMENTATIONS

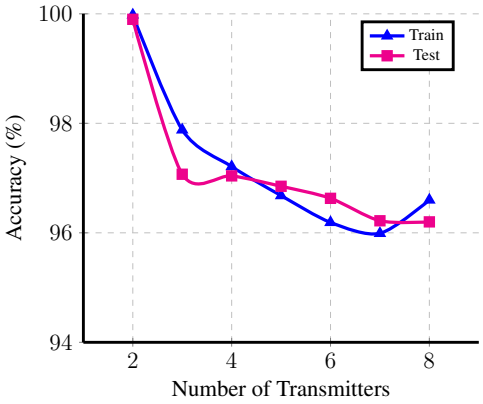


Fig. 14. Training and Accuracy with Increasing numbers of Transmitters

for discriminating between the known transmitters.

3) Deep neural network yields excellent accuracy for known transmitter classification.
4) DNN or CNN can be used for transmitter fingerprinting after the discriminator is able to distinguish `Trusted` and `Fake` signals.

It is to be noted that traditional machine learning techniques suggest different neural networks for different kinds of classification problems. However, we have used a single neural network architecture for both 4 and 8 class classifications, for providing an end-to-end solution. One can use the same neural networks for classifying up to 8 transmitters without changing any parameter or hyper-parameter of the proposed models.

VI. CONCLUSIONS

In this paper, we address the problem of identifying RF transmitters of similar types in the presence of adversarial signals. We argue that most machine learning techniques would not be effective in adversarial situations and that breakthroughs in generative adversarial nets (GAN) can be instrumental in detection of rogue transmitters and subsequently the accurate identification of known ones. We propose and implement a generative model and a discriminative model for the GAN. We collected over-the-air raw IQ data using USRP B210 and used that to train the GAN. The discriminator was able to detect rogue transmitters with an accuracy of ∼99.9%. As for transmitter classification, we first implemented a convolution neural network (accuracy ∼89%) for exploiting the correlation between IQ data. Then we designed and implemented a deep neural network that showed an accuracy of ∼97% for transmitter identification. Our overall implementation framework provides an end-to-end solution for transmitter fingerprinting and identification using raw IQ data.

REFERENCES

[1] I. Amundson and X. D. Koutsoukos, "A Survey on Localization for Mobile Wireless Sensor Networks," in *Mobile Entity Localization and Tracking in GPS-less Environnments*, 2009, pp. 235–254.
[2] T. J. O'Shea *et al.*, "Physical Layer Communications System Design Over-the-Air Using Adversarial Networks," *CoRR*, vol. abs/1803.03145, 2018.
[3] M. Valkama, M. Renfors, and V. Koivunen, "Advanced Methods for I/Q Imbalance Compensation in Communication Receivers," *IEEE Transactions on Signal Processing*, vol. 49, no. 10, pp. 2335–2344, 2001.
[4] J. Tubbax *et al.*, "Compensation of IQ imbalance in OFDM systems," in *IEEE International Conference on Communications*, 2003, pp. 3403–3407.
[5] I. Goodfellow *et al.*, "Generative Adversarial Nets," in *Advances in Neural Information Processing Systems*, 2014, pp. 2672–2680.
[6] Ettus Research, "USRP B210," https://www.ettus.com/product/details/UB210-KIT, 2018.
[7] NooElec, "USRP B210," http://www.nooelec.com/store/sdr/sdr-receivers/nesdr-mini-rtl2832-r820t.html, 2018.
[8] M. D. L. Angrisani and M. Vadursi, "Clustering-based method for detecting and evaluating I/Q impairments in radio-frequency digital transmitters," *IEEE Transactions on Instrumentation and Measurement*, vol. 56, no. 6, pp. 2139–2146, 2007.
[9] C.-L. Liu, "Impacts of I/Q Imbalance on QPSK-OFDM-QAM Detection," *IEEE Transactions on Consumer Electronics*, vol. 44, no. 3, pp. 984–989, 1998.
[10] T. J. O'Shea, J. Corgan, and T. C. Clancy, "Convolutional Radio Modulation Recognition Networks," in *Engineering Applications of Neural Networks*, 2016, pp. 213–226.
[11] T. J. O'Shea, J. Corgan and T. C. Clancy, "Unsupervised Representation Learning of Structured Radio Communication Signals," in *First International Workshop on Sensing, Processing and Learning for Intelligent Machines (SPLINE)*, 2016, pp. 1–5.
[12] T. OShea and J. Hoydis, "An Introduction to Deep Learning for the Physical Layer," *IEEE Transactions on Cognitive Communications and Networking*, vol. 3, no. 4, pp. 563–575, 2017.
[13] S. Rajendran *et al.*, "Deep Learning Models for Wireless Signal Classification With Distributed Low-Cost Spectrum Sensors," *IEEE Transactions on Cognitive Communications and Networking*, vol. 4, no. 3, pp. 433–445, 2018.
[14] radioML, "RFML 2016," https://github.com/radioML/dataset, 2018.
[15] T. J. OShea, 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, 2018.
[16] E. L. Denton *et al.*, "Deep Generative Image Models using a Laplacian Pyramid of Adversarial Networks," in *Advances in Neural Information Processing Systems*, 2015, pp. 1486–1494.
[17] J. Zhu *et al.*, "Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks," *CoRR*, vol. abs/1703.10593, 2017.
[18] D. Berthelot, T. Schumm, and L. Metz, "BEGAN:Boundary Equilibrium Generative Adversarial Networks," *CoRR*, vol. abs/1703.10717, 2017.
[19] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," *CoRR*, vol. abs/1412.6980, 2014.
[20] N. Srivastava *et al.*, "Dropout: A Simple Way to Prevent Neural Networks from Overfitting," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.
[21] GNURadio, "GNU Radio," https://www.gnuradio.org, 2018.
[22] F. Chollet *et al.*, "Keras: The Python Deep Learning library," https://keras.io, 2015.
[23] M. Abadi *et al.*, "Tensorflow: Large-scale machine learning on heterogeneous distributed systems," *CoRR*, 2016.
[24] R. Al-Rfou *et al.*, "Theano: A python framework for fast computation of mathematical expressions," *CoRR*, 2016.
[25] C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag, 2006.
[26] V. Nair and G. E. Hinton, "Rectified Linear Units Improve Restricted Boltzmann Machines," in *Proceedings of International Conference on International Conference on Machine Learning*, 2010, pp. 807–814.