2021 IEEE International Symposium on Technologies for Homeland Security (HST) | 978-1-6654-4152-0/21/$31.00 ©2021 IEEE | DOI: 10.1109/HST53381.2021.9619834

Machine-Learning PUF-based Detection of RF Anomalies in a Cluttered RF Environment

James Lu
School of Engineering
Brown University
Providence, USA
james_lu@brown.edu

Todd Morehouse
Dept of Elec & Comp Engineering
University of Masschusetts
Dartmouth, USA
tmorehouse@umassd.edu

Jiawei Yuan
Dept of Comp & Info Science
University of Masschusetts
Dartmouth, USA
jyuan@umassd.edu

Ruolin Zhou
Dept of Elec & Comp Engineering
University of Masschusetts
Dartmouth, USA
rzhou1@umassd.edu

***Abstract*—With the emergence of software defined radio (SDR) where a computer program defines transceivers' physical layer functions, waveforms can change dynamically. SDR benefits new protocol deployment, enabling smart wireless communication applications. However, SDR makes it easier to mimic authorized transmission, leaving wireless networks vulnerable to spoofing attacks. This work explores ways to detect such radio frequency (RF) anomalies. Specifically, a machine-learning structure called convolutional neural network (CNN) possesses merits of local perception and shift invariance, matching the characteristics of our sampled SDR data. Therefore, we design a CNN for detection of RF anomalies. Furthermore, a physical unclonable function (PUF) provides physical-layer security by identifying a device analogous to human fingerprint. Our CNN extracts waveform features as well as PUFs of transmission devices, from which we train and validate a classification model. The trained model can detect and identify spoofed signals. As proof-of-concept experiments, we generate RF signals with Ettus Universal Software Radio Peripherals (USRPs) and GNU Radio software. We then use the dataset to train our CNN classification model that analyzes features of the RF signals and the USRPs' PUFs. To expand the robustness of our CNN model in cluttered RF environments typical in the Internet of Things (IoT), we generate satellite signals of Automatic Dependent Surveillance – Broadcast (ADS-B) for aircraft tracking. The testing results confirm the promise of machine-learning PUF-based security enforcement in cluttered RF environments.**

Keywords—Cyber security, wireless communication security, physical-layer security, physical unclonable function (PUF), machine learning to spoofing detection, detecting radio frequency (RF) anomalies, securing software defined radio (SDR), protecting integrity of the Internet of Things (IoT).

I. INTRODUCTION

Many wireless communication technologies operate in the same band of radio frequency (RF) spectrum: e.g., Wi-Fi, Bluetooth, and Zigbee occupy the same 100 MHz band from 2.4 GHz to 2.5 GHz. Radio interference occurs frequently. Ever growing devices such as commercial drones and personal wearables connect to the Internet of Things (IoT), cluttering RF environments. Software defined radio (SDR) solves interference issues with complex spectrum sharing. As illustrated in Figure 1, an aircraft uses a surveillance radar called Mode-S to share its information derived from Global Positioning System (GPS) with other aircrafts and a ground station.

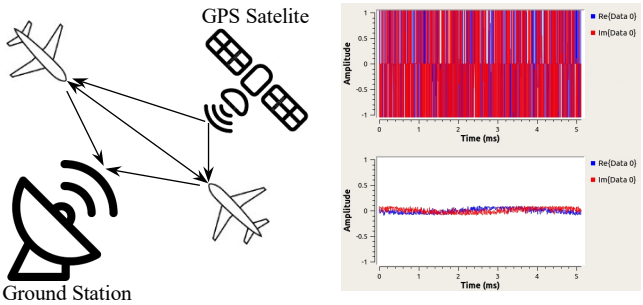

Figure 1: An Example of a Cluttered RF Environment in Mode-S

However, SDR plays a double-edged sword. Its ability to switch channel and modify behavior makes the IoT vulnerable to spoofing attacks. Moreover, physical-layer security becomes paramount to safeguard wireless communications in an era when IoT surround our world. Analog to human interactions, a listener identifies a speaker by physical characteristics, such as unique voice features, rather than contents spoken. The latter can be fabricated at higher/software layers despite using cryptographic memory authentication.

Functioning as a human fingerprint, *RF fingerprint* is a radio transmission characteristic of a device, which depends on the transmitter chain due to its unique manufacturing process imperfections. It differs from a *wireless channel fingerprint*, a random mapping based on temporal factors such as location and propagation characteristics. RF fingerprint has been utilized for *keyless authentication*, a receiver (Rx) identifying a transmitter (Tx) without the need for the two devices to share a secret key. This unique and static input-output characteristic is hardly replicable, even by the same manufacturer. RF fingerprint authentication algorithms have two categories: transient and modulation. *Transient* implementations classify transmitted signals by amplitude/phase characterization of the signal envelope. *Modulation* implementations classify by frequency offset, sync correlation, etc. However, *the simulatability condition* limits the performance of RF fingerprint authentication, i.e. authentication is possible if and only if the adversary cannot simulate the legitimate channel. While evaluating such basic limits of RF fingerprint authentication using Information Theory, Gungor and Koksal [1] devised a graphical approach to check the simulatability region and recommended several methods to enhance the security strength.

Any rate less than Shannon capacity can achieve reliable communication. Erasure and error probabilities of impersonation and substitution attacks correlate to RF channel statistics. Additional analysis at Rx can aid authentication.

Like dynamic biometric augmenting fingerprint in human identification, *Physical Unclonable Function (PUF)* is a black-box challenge-response system, $r = f(c)$. It maps input domain to output range called *challenge-response pairs (CRPs)*, but the internal parameters of *f(.)* are hidden from users. Such parameters represent physical characteristics of the integrated circuit (IC) from manufacturing process imperfections, such as the variability of a circuit's internal gate delay. PUF applicability in security relies on the difficulty of measuring and estimating these parameters as well as the difficulty of manufacturing chips with the same set of parameters. The domain of *f(.)* or the number of unique challenges *c* that a PUF can process sets apart two types of PUFs, each with different security applications. *Weak PUFs* support a small number of challenges (in some cases only a single challenge) and can be applied for Secure Key Generation, which is out of the scope of this paper. *Strong PUFs* support a large number of challenges (ideally, exponential in the number of challenge bits) so that complete determination/measurement of all CRPs within a limited timeframe becomes infeasible. Therefore, strong PUFs can provide low-cost authentication [2].

Figure 2 depicts a two-phase protocol of an authenticator (Rx) identifying devices with strong PUF (Txs). First for bootstrap, Rx directly contacts each Tx to build a table of CRPs. When ready for authentication, Rx issues a *c* for *r*s, checks match to identify Txs, and removes the *c* from the CRP table to prevent replay attacks. Comparing to traditional cryptographic authentications, PUF authentication does not require Txs with secure nonvolatile memory, anti-tamper mechanism, or additional circuitry for crypto acceleration. However, basic PUF authentications still need secure storage for CRPs on Rx, posting the same vulnerability as traditional cryptographic memory authentications. Additionally, the restriction of using each CRP once demands a large memory to store the CRP table. Likewise, PUF authentications face side-channel attacks. Besides the two factors of computational intractability aforementioned, the security of a strong PUF requires an additional difficulty of predicting PUF behavior based on past CRPs [2].

To eliminate PUF's scalability problem with CRP table, Rx adopts a compact model from machine learning to emulate PUF challenge-response behavior rather than stores a CRP table [2]. Tx data that we sampled exhibit localized characteristics and shift invariant, in accordance with the merits of a specific machine learning model called convolutional neural networks (CNN). Although an adversary can spoof an authentication sequence from observations, Rx chooses a one-time random challenge and computes the response matching Tx's. Morehouse and Zhou [3] built and trained a PUF-based CNN model for Rx that can identify RF Txs with accuracy above 90%. This work demonstrates the ability of Morehouse-Zhou model to detect RF anomalies in a cluttered RF environment. We use universal software radio peripherals (USRPs) along with GNU Radio software to form a RF environment. An attacker imitates one of USRPs. An Rx using our CNN classification model can detect the attacker with accuracy at least 97%.

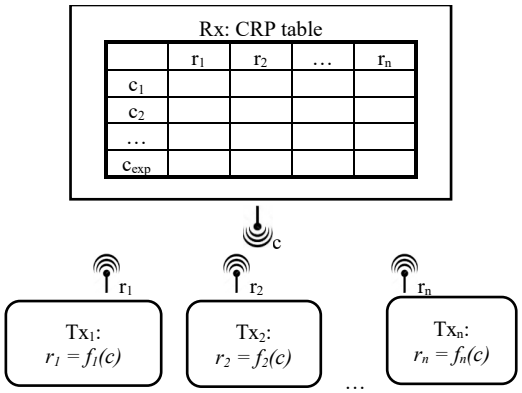

Figure 2: Strong PUF for basic authentication

Our major contributions include:

1) A single central Rx effectively detects RF anomalies among multiple distributed Txs in a cluttered RF environment. The setting serves well to IoT applications.
2) There is no extra hardware for PUF implementation at Tx by exploiting a device's inherent variations resulting from process variability (on-chip) and component tolerance (on-board) for each Tx. The model at Rx compensates Rx non-ideality and accounts for variability of data channel.
3) Our proof-of-concept experiments confirm the alignment of CNN's locality with the characteristics of our dataset, suitable to wireless communication security. The work justifies the call for data-centric machine learning.

The rest of the paper is organized as follow: Section II describes related work. Section III introduces the adopted system model. Section IV designs a CNN architecture. Section V describes the proof-of-concept experiments while VI discusses the results. Section VII concludes the paper.

II. Related Work

Traditional approaches to RF fingerprinting focused on algorithms, relying on domain experts to extract features of RF transmitter imperfections. *Expert-driven* RF fingerprinting was neither reliable (affected by environment distortions) nor scalable (unable to consider all possible scenarios). *Data-driven* approaches, specifically deep learning, can learn features from RF signals, achieving better performance and higher scalability. Youssef et al [4] explored the efficacy of machine learning to RF signal processing, particularly for PF fingerprinting. Four ML algorithms were evaluated: support vector machines (SVM), deep neural nets (DNN), convolutional neural nets (CNN), and DNN with multi-stage training (MST). Their first machine learning algorithm, SVM with 2 different configurations, is non-deep-learning while the rest three are deep-learning, resulting five models. We focus on their deep learning models in our review. Starting from a conventional DNN of two fully-connected hidden layers as the base model trained with the first-order stochastic gradient, Youssef et al extended it, respectively, in model structuring to CNN of two convolutional layers and in model training to MST with a second-order update called Levenberg-Marquardt (LM) method. Their goal of DNN and CNN tests the ability to distinguish among known transmitters while the goal of MST

978-1-6654-4152-0/21/$31.00 ©2021 IEEE

tests the ability to extend the model to capture novel devices via incremental learning, a special kind of transfer learning. *Transfer learning* takes a model trained to perform task A as a starting point and retrains it for a new model to perform another task B. *Incremental learning* enables the new model to perform both tasks A and B, adapting new data without forgetting its existing knowledge. Youssef et al ranked various deep learning approaches to study RF domain in performance (accuracy) and scalability (CPU time) using a methodical and repeatable real-world experimentation and commercial-off-the-shelf (COTS) WiFi transceiver platform. Future work includes extending to more challenging conditions, testing robustness, and addressing complex valued artificial neural networks.

Chatterjee et al [5] combined PUF concept with RF fingerprinting. A RF-PUF framework, for a Machine Learning equipped Receiver (ML-Rx), was proposed to authenticate wireless transmitters (Tx) in real time by exploiting manufacture process imperfections in Tx. It is the first work in an asymmetric IoT network of multiple distributed Txs and a single central Rx for low-cost, preamble-less, intrinsic PUF-based authentication of IoT nodes. The feasibility study of RF-PUF showed that the inherent RF properties arising from the manufacturing process in a wireless node can be exploited as a strong PUF for device authentication in asymmetric IoT networks without any additional hardware at the Tx. The Rx, using in-situ light-weight supervised machine learning, can detect up to 10,000 Txs with about 99% accuracy. They also validated RF-PUF by physical implementation with two software defined radios (SDR) to emulate multiple unique Txs and an on-board microprocessor in Rx to deploy artificial neural networks. Future work includes improving Rx signature compensation, circuit techniques to implement erasability and certifiability, formal or experimental validation of protection against attacks, and stability analysis.

Physical-layer characteristics used in spoofing detection schemes for wireless communications belong to two categories: RF/hardware features and channel/location variances. *RF/hardware* category, based on distinctive patterns in modulation domain of RF signals that different transceivers emit like I/Q origin offset, performs well but needs a high-end signal analyzer. *Channel/location* category, taking channel state information (CSI) and location-specific features such as received signal strength (RSS), costs less but cannot detect attackers close to the legitimate transmitter. Wang et al discovered that Signal-to-Noise Ratio (SNR) traces obtained in sector level sweep (SLS) process are different even for the same type of IEEE 802.11ad devices [6]. Such work is the first to explore SLS SNR traces in 60GHz mmWave off-the-shelf devices to detect spoofing attack for IEEE 802.11ad networks. Machine-learning classification has been applied to detect spoofing attacks with high efficacy at low cost. SLS SNR traces are influenced by both Tx location and hardware impairment while readily obtainable without extra circuits. The machine learning framework stacks a backpropagation network (BN) with a forward propagation network (FN) as generative adversarial networks (GANs) for small sample learning and fast model construction, achieved 98% detection accuracy.

Morehouse and Zhou [3] demonstrated the feasibility of using convolutional neural networks (CNN) to identify RF devices by classifying raw baseband signals without the need of data preprocessing. The CNN architecture in [3] included convolution for feature extraction, batch normalization to increase training speed and accuracy, ReLU for activation, pooling for data reduction, fully connected for dimension reduction, and softmax for classification. The prototype generated a dataset by using three transmitters (one USRP N210 and two PLUTO SDRs), each 80K frames received by a USRP N210 were used to train its CNN in MATLAB for 1.5 hours. The experiment yielded 92.5% testing accuracy at identifying different types of SDR devices (USRP and PLUTO) and individual devices of the same type (two PLUTOs), all radios being identified above 85% accuracy with the range in their accuracy differences about 8%. Future work includes real-time identification of new devices by incremental learning, testing the CNN's prediction confidence, and vulnerability analysis of the CNN to spoofing and jamming.

CNN is the most widely used deep learning technique for grid-like data such as image segmentation and computer vision. CNN's advantages of local/neighboring data correlation and global/hierarchical feature combination enables CNN to learn features automatically from raw data. CNN's another advantage of weight sharing makes it efficient in terms of memory and computation complexity. As an efficient and automatic feature extractor, CNN is also widely used for time-series data, outperforming traditional machine learning techniques on speech recognition tasks and natural language processing. The basic CNN structure has three stages of components. The first is multiple layers of *convolutional kernels* where each neuron computes a weighted sum of input tensors by sliding kernel, *pooling* to down sample for reduction in feature-map size by extracting a combination of invariant features, and *fully connected* at the end of the network to globally analyze the local features from the preceding layers and non-linearly combine selected features for classification. The second component is mapping functions, often called *activation function* that serves as a decision function with non-linearity added to learn intricate patterns. The last component is regulatory units: *batch normalization* unifying the distribution of feature-map values by setting them to zero mean and unit variance; *dropout* eliminating the cause of overfitting by randomly skipping some connections within a network. Building and training a CNN is more like an art than a science currently. It involves many aspects, to name a few: modification of processing units, optimization of parameters and hyper-parameters, design of pattern blocks, selection of layers and their connectivity, and choice of architectures. The last aspect of CNN architectures can have seven categories: spatial exploitation, depth based, multi-path, width-based, feature-map (channel on features), channel boosting (channel on input), and attention [7]. Therefore, we adopt a CNN to learn features of RF signals and devices' PUFs. By trial and error, we modify and retrain Morehouse-Zhou PUF-based CNN classification model to search for a good model that detects RF anomalies in a cluttered RF environment.

While powerful for modeled PUF authentication, machine learning is also a double-edged sword. Sharma et al simulated the catastrophes caused by attacks on a machine learning model in the Internet-of-Vehicles with adversarial examples [8]. Recent vulnerability studies of PUFs to modeling attacks are alarming. Particularly according to Khalafalla et al [9],

978-1-6654-4152-0/21/$31.00 ©2021 IEEE

modeling attacks with deep learning easily broke the security of strong PUFs immune to traditional machine learning models such as support vector machines and single-layer artificial neural networks, with accuracy about 99% trained in minutes. This research shaken the ground of security applications rest on strong PUF's unpredictability and unclonability. Therefore, new PUF architectures were suggested to countermeasure deep learning attacks.

III. Adopted System Model

We adopt a CRP model trained from machine learning, instead of a CRP table in Rx, for a system model to detect spoofing attacks. As shown in Figure 3, our system contains a CRP model in Rx that identifies Txs with strong PUFs. An adversary (A) launches attacks, for example, by spoofing Tx_1 signals after observing its behavior to fool Rx. As part of handshaking before a communication session, Rx issues a challenge c, and a Tx_i that wants to join the session presents its credential for authentication by sending its response $r_i = f_i(c)$. As per the prior agreement $F_i(.)$, Rx accepts the legitimate response r_i from Tx_i matching the specific CRP $F_i(c)$ while rejects A's spoofed response $\tilde{r}_1$.

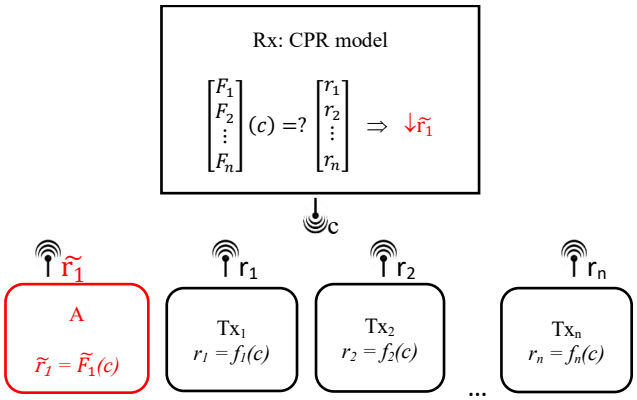

Figure 3: System Model to detect spoofing attacks

The system involves a two-phase protocol as a model-based authenticator. Before the authentication phase described above, Rx first goes through its bootstrapping phase by generating a database of CRPs with Txs in a secure environment. The CRP dataset is used to train and validate a machine learning model. After testing the trained model to assess its efficacy, Rx discards the CRP dataset and deploys the CRP model ready for the authentication phase.

A, on the other hand, also goes through its bootstrapping phase by stealthily collecting the RF signals and CRPs of the targeted device, say Tx_1. Likewise, A deploys its model ready to launch spoofing attacks $\tilde{r}_1 = \tilde{F}_1(c)$. An arms-race runs between Rx and A; A fools Rx for acceptance as Tx_1 while Rx detects the presence of A. Both A and Rx improve its own functionality, similar to a generative adversarial network (GAN) where A as generator and Rx as discriminator are trained independently.

IV. Proposed CNN Architecture

We propose Morehouse-Zhou PUF-based CNN model [3] for the CRP model at Rx. Figure 4 shows the architecture of our CNN-Rx. A *convolution layer (CL)* for automatic feature extraction is followed with a regulatory unit of *batch normalization*, a mapping function of ReLU *activation*, and a *pooling*. This process repeats six times, each increasing the filter size to explore correlation among neighboring inputs at coarser granularity. The process ends with a *fully connected layer (FC)* for non-linear combination before the final *softmax* activation to classify Tx_i or Abnormal. The depth of CNN-Rx architecture is 1 CL × 6 + 1 FC = 7.

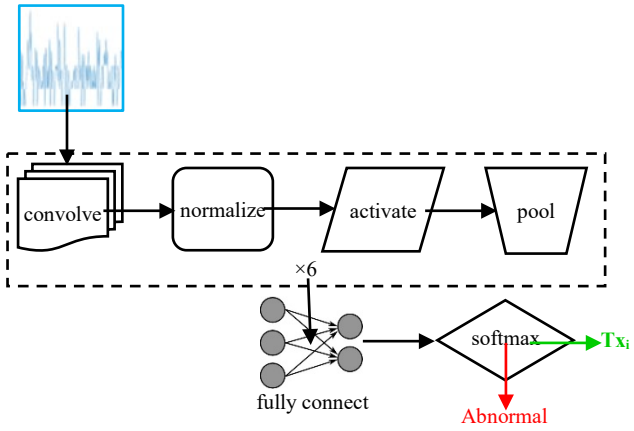

Figure 4: CNN-Rx Architecture

We retrain Morehouse-Zhou CNN model on our new PUFs dataset including RF anomalies for CNN-Rx to improve its detection accuracy. The initial imbalance between normal and abnormal data may affect the performance of CNN-Rx with bias. As more anomaly data becomes available, CNN-Rx will improve its detection rate.

V. Proof-of-Concept Experiments

Figure 5 illustrates the experiment setup. Five USRPs on the UMassD SDR Server are used, as labeled in the figure. CNN-Rx uses a USRP N210 for collecting CRPs with three licensed Txs at the bootstrapping phase and for hosting the trained CNN model for the authentication phase. MATLAB deep learning toolbox runs on a Linux server to train the CNN model with the CRP dataset collected at the bootstrapping phase. The three licensed Txs are USRP N210s. A also uses a USRP N210 to eavesdrop the RF signals of the targeted device, Tx_1 in this study, and retransmits the signals to fool CNN-Rx.

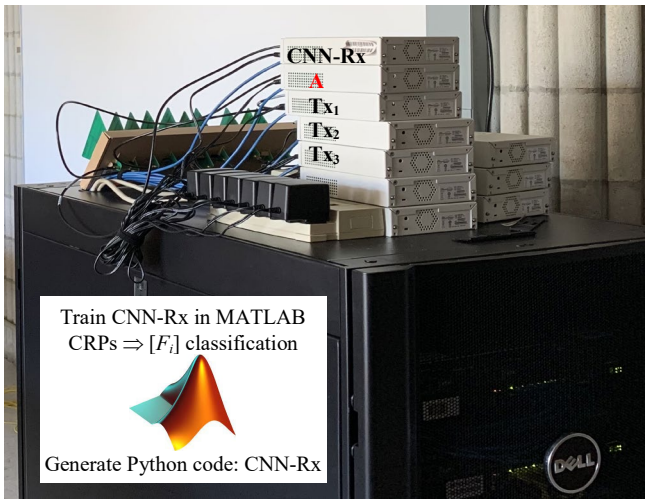

Figure 5: Experiment Setup

978-1-6654-4152-0/21/$31.00 ©2021 IEEE

In designing experiments, we follow the machine learning workflow that starts with collecting data to train a model for a system and ends with deploying that model in the system. This process spirals to assure a trustworthy AI system. Our work observes the trend of data-centric AI that shifts the focus from tuning hyperparameters as model-centric approach to improving data quality [10]. Utilizing the team's domain knowledge in signal processing and cyber security, we collect and prepare good data instead of big data. This new paradigm also helps us gain better insights of the data and the problems at hand.

A. Data Collection and Preparation

We use five USRP N210s to operate as one receiver and four transmitters. Each transmitter sends the same signal multiple times: a frame of 1,024 Automatic Dependent Surveillance–Broadcast (ADS-B) sample signals for 40,000 times. Signals are not collected simultaneously in order to differentiate the PUFs of individual devices. Figure 6 shows a data sample.

DataSet | DataSet.USRPSet1

DataSet.USRPSet1

Fields	data	modType	label
1	2x1024 dou...	'BPSK'	1x1 categori...
2	2x1024 dou...	'BPSK'	1x1 categori...
3	2x1024 dou...	'BPSK'	1x1 categori...
4	2x1024 dou...	'BPSK'	1x1 categori...
5	2x1024 dou...	'BPSK'	1x1 categori...
6	2x1024 dou...	'BPSK'	1x1 categori...
7	2x1024 dou...	'BPSK'	1x1 categori...
8	2x1024 dou...	'BPSK'	1x1 categori...
9	2x1024 dou...	'BPSK'	1x1 categori...
10	2x1024 dou...	'BPSK'	1x1 categori...
11	2x1024 dou...	'BPSK'	1x1 categori...
12	2x1024 dou...	'BPSK'	1x1 categori...
13	2x1024 dou...	'BPSK'	1x1 categori...
14	2x1024 dou...	'BPSK'	1x1 categori...

Figure 6: Data Sample

We use GNU Radio Companion, a framework to design, simulate, and deploy SDR systems. Figure 7 shows the flowgraphs of our entire system model for the USRPs to send and receive signals. GNU Radio Companion offers a versatile coding GUI where blocks can be linked together to program an SDR. Each flowgraph can be converted to a Python code. The File Source inputs the premade ADS-B signals into GNU Radio. This signal is repeated until the user manually stops the program. The fading model implements a frequency shift and can simulate a variety of channel impediments. The phase noise generator adds Gaussian noise to the signal given a specified magnitude and alpha. The USRP sink transmits the signal to the USRP source which writes the output as a binary file. The sink and source blocks have parameters determining the SDR IP address, sampling rate, and channel center frequency among many others. A CNN trained in MATLAB then determines which signal came from which SDR to identify spoofs.

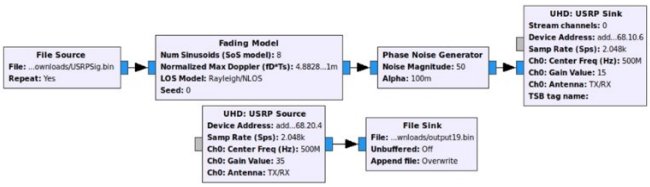

Figure 7: GNU Radio Flowgraphs

The four transmitters are labeled "USRP1", "USRP2", "USRP3", and "Unknown". After basic data collection, we prepare datasets progressively for increasing impediments to boost model resilience. Six versions of datasets, shown in Table 2, are prepared. The first version takes three USRPs to categorize which radio a signal comes from. The second version adds fading model, simulating the doppler effect for the three transmitters moving relative to the receiver. The third version adds the "Unknown" to the second for anomaly detection. The fourth version includes phase noise generator to further distort the signal. Various magnitudes of Gaussian noise are applied to the same signal, but it concludes that such additions yield little effect on network accuracy. Given a wide range of noise, the CNN's decisions remained similar. We further investigate the independency of the devices' RF features and PUFs from their labeling. In fifth and sixth versions, one device changes its label to "USRP1"; CNN-Rx still can distinguish this device and the other device despite their same label.

Each of the trainings converges quickly and consistently reaches ~99% after five epochs. As shown later in Figure 8 of Section VI Results, the training and validation curves closely follow each other in both accuracy and loss, which indicates the high quality of our data as sufficient and similar.

B. Model Training and Deployment

Each version of our datasets is randomly divided into 94-3-3% parts for training, validation, and testing. Table 1 shows the hyperparameter setting.

Table 1: Hyperparameters for Training CNN-Rx

Hyperparameter	Value
Initial Learning Rate	0.02
Learn Rate Drop Period	1
Learn Rate Drop Factor	10%
Learning Rate Schedule	piecewise
L2 Regularization	10^{-4}
Momentum	0.9
Shuffle	every epoch
Number of Epochs	5
Mini Batch Size	256
Validation Data	{Signals, Labels}
Validation Frequency	$\left\lfloor \frac{\#\text{ of Training Labels}}{\text{Mini Batch Size}} \right\rfloor$
Execution Environment	auto

Table 2 summarizes our training scenarios for six versions of our datasets. Each training took more than 40 minutes to converge. Each dataset contains 120,000 or 160,000 frames as each device provides 40,000.

978-1-6654-4152-0/21/$31.00 ©2021 IEEE

Table 2: Training Scenarios

DV #	Dataset Version	Trained Network
1	3 USRPs & 40,000 frames each	trainedNetwork.mat
2	3 USRPs & 40,000 frames each, doppler effect	trainedNetwork1.mat
3	4 USRPs & 40,000 frames each, doppler effect	trainedNetwork2.mat
4	4 USRPs & 40,000 frames each, doppler effect, 50mag Gaussian noise	trainedNetwork3.mat
5	4 USRPs & 40,000 frames each, doppler effect, 50mag Gaussian noise, change 1 USRP	trainedNetwork4.mat
6	3 USRPs & 40,000 frames each, doppler effect, 50mag Gaussian noise, change 1 USRP	trainedNetwork5.mat

VI. Results and Discussion

Table 3 shows the training results from all six datasets, agreeing with our hypotheses of Sub-Section V-A summarized in Table 2. Dataset 4 yields the most interesting Network 3.

Table 3: Results from All Dataset Versions

Dataset Version #	Trained Network	Accuracy
1	trainedNetwork.mat	0.9999
2	trainedNetwork1.mat	0.9995
3	trainedNetwork2.mat	0.9142
4	trainedNetwork3.mat	0.9009
5	trainedNetwork4.mat	0.9192
6	trainedNetwork5.mat	0.9324

Figure 8 is a training screenshot with Dataset 4 for Network 3: validation closely follows training in both accuracy and loss.

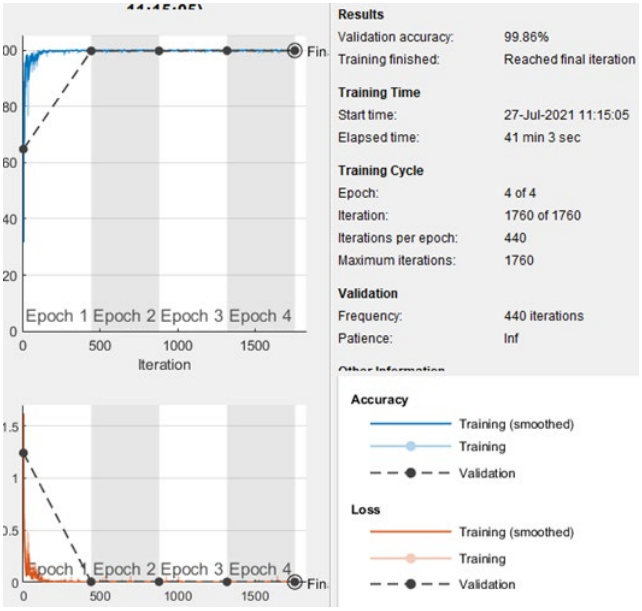

Figure 8: Training Curve for Network 3

Figure 9 shows the confusion matrix of trainedNetwork3's results in details. The matrix indicates that the CNN's predictions often align with the labeled ground truth when validating on 120 of each class. Though fairly accurate, the network sometimes miscategorized USRP2 or Unknown. This may be due to USRP2 and Unknown having similar PUFs. USRP3's high accuracy may indicate that its PUF is most dissimilar from the other devices.

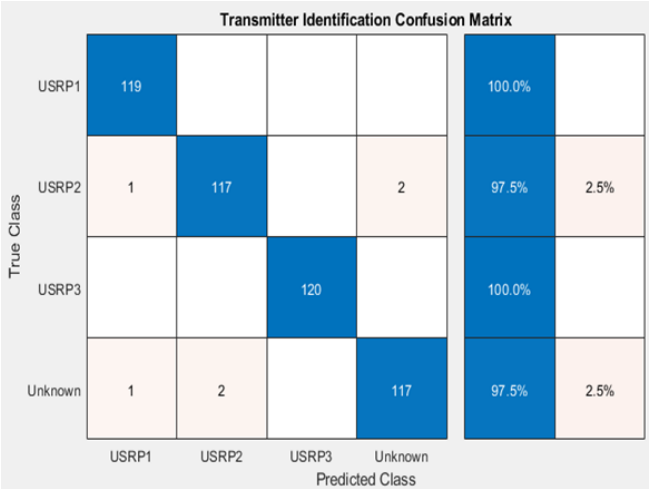

Figure 9: Confusion Matrix for Network 3

To test our CNN models trained with six versions of datasets. We add another USRP N210 and label it "Unknown2" as a new attacker. Each CNN network is tested against several short sample signals. The test cases evaluate if the network can identify the three specified transmitters as well as two attackers from a signal of less than 5 seconds. "Unknown1" is an attacker each CNN is trained on, but "Unknown2" is an attacker a CNN has never seen. Each sample in the signal is categorized so confidence is determined as the percent of the CNN's final prediction to all samples in the signal. Table 4 summarizes trainedNetwork3's predictions and confidence to the signal test cases. Most of the specified devices are correctly identified. The unknown devices are also confidently categorized. Various signal lengths are evaluated and have little effect on network performance.

Table 4: Signal Test Cases & Results for Network 3

<5 seconds output9	identified USRP1 as USRP1, 1
<5 seconds output10	identified USRP2 as USRP1, 1
<5 seconds output11	identified USRP3 as USRP3, 0.6254
<5 seconds output13	identified USRP3 as USRP3, 0.9867
<5 seconds output7	identified Unknown1 as Unknown, 0.9029
>5 seconds output12	identified Unknown2 as Unknown, 1

VII. Conclusion and Future Work

This work, as a sequel of Morehouse-Zhou PUF-based CNN model that identifies RF devices [3], updates their CNN model for detection of RF anomalies in a cluttered RF environment. We form an aircraft tracking system with USRPs that generate ADS-B signals carrying the information about aircrafts'

978-1-6654-4152-0/21/$31.00 ©2021 IEEE

locations and velocities. Data collected provide a new dataset to train a customized CNN for classifying both RF signals from each Tx and its unique USRP PUF. The experimental results demonstrate the effectiveness of Morehouse-Zhou PUF-based CNN model as an authentication system with the presence of attackers in a cluttered RF environment. The system expects work well for securing IoT applications besides the airport example environment.

The merits of such a detection system include no extra hardware for the additional physical-layer security because each Tx inherits PUF as device variation resulting from process variability (on-chip) and component tolerance (on-board). The only cost to the ability of detecting RF anomalies among many distributed Txs is software deployment at a single central Rx, a setting typical to IoT applications. Our proof-of-concept experiments also confirm the alignment of the locality merits possessed by CNN model with the characteristics of the dataset collected, a finding explorable to other wireless communication security services. Our work also demonstrates the assurance of trustworthy products by shifting from model-centric towards data-centric machine learning.

In future development, we will achieve several goals. Firstly, we will develop and test our detection system in real time using new form of machine learning hardware like field-programmable gate arrays (FPGA). Next, we will optimize our PUF-based CNN model, specifically trainedNetwork3, to reduce error rate. Thirdly, we will explore the deployment stage of machine learning lifecycle by designing experiments to test our CNN model in real physical setting of drones. We will also extend our model's resilience for anomalies detection against modeling attacks with machine learning by exploring GAN. Last but not the least, our work will contribute to the field of machine learning in general for deep learning interpretability through data-centric approach.

Acknowledgement

This work was partially supported by the National Science Foundation (NSF) Research Experiences for Undergraduates (REU) Site at the University of Massachusetts Dartmouth: Secure, Robust, and Resilient AI-enabled System Engineering. It was also sponsored by the University of Massachusetts Dartmouth's Marine and Undersea Technology (MUST) Research Program funded by the Office of Naval Research (ONR) under Grant No. N00014-20-1-2170.

References

[1] O. Gungor and C. E. Koksal, "On the Basic Limits of RF-Fingerprint-Based Authentication," *IEEE Transactions on Information Theory,* vol. 62, no. 8, pp. 4523 - 4543, August 2016.

[2] C. Herder, M.-D. Yu, F. Koushanfar and S. Devadas, "Physical Unclonable Functions and Applications: A Tutorial," *Proceedings of the IEEE,* vol. 102, no. 8, pp. 1126 - 1141, August 2014.

[3] T. Morehouse and R. Zhou, "RF Device Identification using CNN based PUF," in *2020 IEEE 63rd International Midwest Symposium on Circuits and Systems (MWSCAS)*, Springfield, MA, USA, 2020.

[4] K. Youssef, L. Bouchard, K. Haigh, J. Silovsky, B. Thapa and C. V. Valk, "Machine Learning Approach to RF Transmitter Identification," *IEEE Journal of Radio Frequency Identification,* vol. 2, no. 4, pp. 197 - 205, December 2018.

[5] B. Chatterjee, D. Das, S. Maity and S. Sen, "RF-PUF: Enhancing IoT Security through Authentication of Wireless Nodes using In-Situ Machine Learning," *IEEE Internet of Things Journal,* vol. 6, no. 1, pp. 388 - 398, February 2019.

[6] N. Wang, L. Jiao, P. Wang, W. Li and K. Zeng, "Machine Learning-based Spoofing Attack Detection in MmWave 60GHz IEEE 802.11ad Networks," in *IEEE INFOCOM 2020 - IEEE Conference on Computer Communications*, 2020.

[7] A. Khan, A. Sohail, U. Zahoora and A. S. Qureshi, "A Survey of the Recent Architectures of Deep Convolutional Neural Networks," *Artificial Intelligence Review,* vol. 53, no. 8, p. 5455 – 5516, December 2020.

[8] P. Sharma, D. Austin and H. Liu, "Attacks on Machine Learning: Adversarial Examples in Connected and Autonomous Vehicles," in *IEEE International Symposium on Technologies for Homeland Security (HST)*, Woburn, MA, 2019.

[9] M. Khalafalla, M. A. Elmohr and C. Gebotys, "Going Deep: Using deep learning techniques with simplified mathematical models against XOR BR and TBR PUFs (Attacks and Countermeasures)," in *IEEE International Symposium on Hardware Oriented Security and Trust (HOST)*, San Jose, CA, 2020.

[10] R. Ashmore, R. Calinescu and C. Paterson, "Assuring the Machine Learning Lifecycle: Desiderata, Methods, and Challenges," *ACM Computing Surveys,* vol. 54, no. 5, pp. 111:1-39, May 2021.

978-1-6654-4152-0/21/$31.00 ©2021 IEEE