2022-12-01

# Radio Frequency Fingerprinting And Its Application To Scada Environments

Evan White
*University of Texas at El Paso*

RADIO FREQUENCY FINGERPRINTING AND ITS APPLICATION TO SCADA ENVIRONMENTS

EVAN MARCUS WHITE

Master's Program in Computer Science

APPROVED:

Deepak Tosh, Ph.D., Chair

Sai Mounika Errapotu, Ph.D.

Palvi Aggarwal, Ph.D.

Abel Gomez, Ph.D.

Stephen Crites, Ph.D.
Dean of the Graduate School

# RADIO FREQUENCY FINGERPRINTING AND ITS APPLICATION TO SCADA ENVIRONMENTS

by

EVAN MARCUS WHITE, B.S.

THESIS

Presented to the Faculty of the Graduate School of

The University of Texas at El Paso

in Partial Fulfillment

of the Requirements

for the Degree of

MASTER OF SCIENCE

Master's Program in Computer Science

THE UNIVERSITY OF TEXAS AT EL PASO

DECEMBER 2022

# Abstract

With the introduction of IoT into ICS and smartgrid environments there has been a modernization of communication protocols through the internet. This has led to the use of features such as TCP/IP but with it comes modernized attack vectors against these systems. These attacks can be Man In the Middle (MITM), rogue device communication and device cloning. To prevent these attacks, this thesis deploys Radio Frequency Fingerprinting (RFF) techniques to verify the uniqueness and legitimacy of known devices. It is crucial to employ security measures within ICS that do not add to the network complexity as this effects the availability of critical resources. RFF aims to solve this by establishing itself a physical layer authentication method. It does not add network complexity as it focuses on the analysis of existing wireless transmissions amongst devices in the ICS network. RFF has improved significantly through Convolutional Neural Networks (CNN) and this thesis presents a case study on the feasibility of deploying these new techniques on Remote Terminal Unit (RTU) devices.

It has been found that a RFF CNN model can run alongside the normal duties of an RTU. Directly this thesis shows that the increased responsibility is possible on low end devices with a 64-bit architecture, which means that devices like the SIMANTIC S7-1500 controller can utilize RFF in the field. The trained accuracy of the CNN has a detection rate of 84% when handling the dataset gathered in this thesis. This is a promising result given the fact the computer intensive RFF mechanism is being executed on a resource constrained environment like a RTU.

# Table of Contents

# Chapter 1

# Introduction

## 1.1 Internet of Things

Billions of Internet of Things (IoT) devices have arrived in the consumer space and are dominating in the Information Technology (IT)/Operational Technology (OT) space within Industrial Control Systems (ICS). These devices are able to deliver useful data with the luxuries provided by TCP/IP and modern routing at a low cost. Other luxuries included with IoT are IEEE 802.11, HART and WiMAX which allow for the communication of data wirelessly. While this is convenient and cost effective compared to previously closed loop systems, it adds attack vectors to an ICS network.

Man In The Middle (MITM) attacks have been prolific throughout the years and are used as a means to deliver replay attacks and distribution of unwanted software. Since IoT devices are becoming widely adopted within existing networks and will be integrated into smartgrid networks, the operators must be concerned with looming threats [15]. These threats are device cloning and rogue devices used for MITM attacks. This is caused by the exposure of these field devices to the commercial consumer space. These networks are no longer physically air gapped and thus opening the attack demographic from insider attacks to public entities.

## 1.2 Smartgrid

The smartgrid is an electrical grid that employs modern technologies such as the internet and IoT device to manage electrical demands. As one can imagine, constant communication

and connectivity is required to maintain devices with these new capabilities. Compared to conventional electric grids, the IoT devices represent nodes with their own identities. These may manage jobs like actuators to turn off electric flow or sensors like smart- meters and provide the control and information needed to make educated accurate decisions. The scoped deployment of smart grids target deployment within existing infrastructure including residential areas to utilize devices like the electric flow sensors for measurements as shown in Figure 1.1

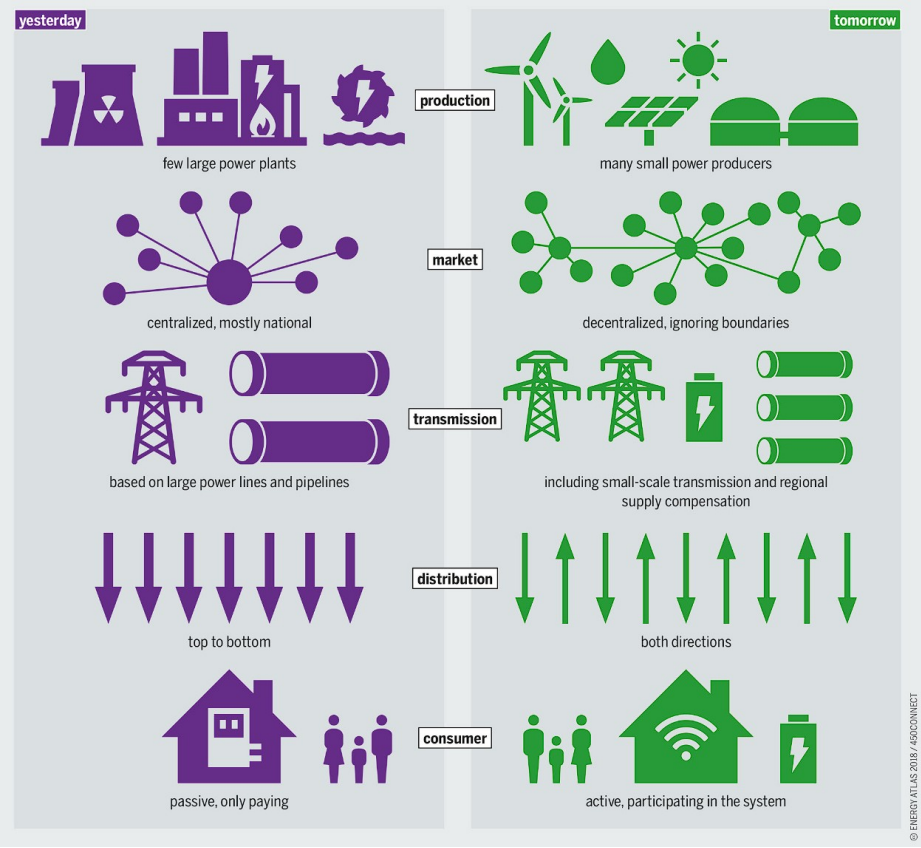


Figure 1.1: Visual demonstration of how each portion of older electric grid systems map to the expanding smartgrid [19]

Due to the sheer scale of this network, copious amounts of data needs to be maintained for this system to function. Since the smartgrid is essentially an ICS network, it also bears the hindrance of an OT network. This means that all communication has to

maintain availability as its core pillar as opposed to traditional security which focuses on confidentiality.

## 1.3 IT Security vs. OT Security

Over the years IT and OT have begun to blend over the years due to the adoption of IT solutions into this field. These adoptions were made with the intention of creating improved connectivity and the ability to use remote access. While this has helped close the differences between the two, the core security values of each of these networks are different. [17] As illustrated in Figure 1.2, we see that OT values availability above all else.

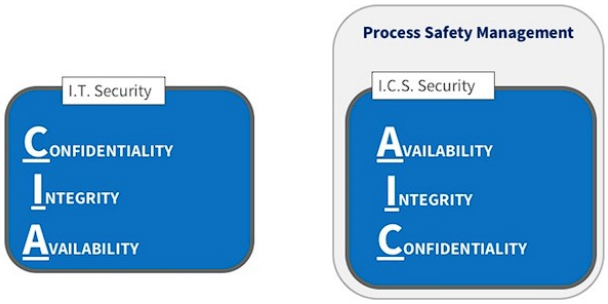


Figure 1.2: Differentiation between traditional security CIA and ICS CIA [21].

ICS is designed to remain operational at all times since it is a production environment providing national core services. Within OT there is also a battle for resources such as network and computing resources. Consider [2] where the authors propose to use heartbeat packets within a Supervisory Control and Data Acquisition (SCADA) System, which is unsuitable as it adds complexity to the network and software. Since OT now has IT devices that add internet connectivity, and the smartgrid is placing devices in non-industrial settings, there is a physical disparagement. Consequently, there is a concern about who has access to these end devices as they are no longer inaccessible and air-gapped. We must be able to ensure our physical security and prevention against attacks such as MITM and cloning.

Not all devices that are in ICS systems utilize the same communication protocol. These protocols were derived from proprietary analog communication standards such as modbus and DNP3. Originally the devices that communicated with these protocols were air-gapped and did not have to worry about network security. OT adopted TCP/IP creature comforts such as routing and adding application level implementations of Modbus and DNP3. However, security was not implemented until later iterations like Modbus+. The desire for a single secure authentication protocol arises from the fact that not all devices are unionized under one secure protocol.

## 1.4 Security through Radio Frequency Fingerprinting

To overcome the imperfections in existing security coverage a physical layer authentication method is beneficial. This comes in the form of Radio Frequency Fingerprinting (RFF), this methodology uses a radio's 'biometric' fingerprint to identify the device regardless of the types of packets it is sending. This fingerprint is derived from manufacturing imperfections and is difficult to replicate feasibly (Section 2.2). This fingerprint is seen across the wireless transmissions of devices and can be used to continuously authenticate the device. This continuous authentication is possible since the fingerprint can be perceived across the wavelength of a transmission through specific analysis (Section 2.3.1). The dataset required is pre-existing in ICS networks with wireless communications, all that is needed is the ability to harvest these communications for RFF use. Since wireless communications are being recorded, there is not an introduction of more packets into the network. Once this data has been collected, unique fingerprints can be derived using computer-accelerated techniques for authentication. These accelerated techniques come in the form of Machine Learning and Neural Networks. These technologies allow for the superior visual learning ability to be applied to a large and complex dataset feasibly with low implementation overhead.

## 1.5 Research Problem

In spite of the fact that RFF is a robust tool for physical layer authentication, there are some issues it poses that need to be addressed.

- How can spectral data be harvested? (Section 4.1)
- Which analysis techniques are robust and reliable? (Section 2.3)
- Which Neural Network/Machine Learning architecture can offer optimal results in classifying the wireless devices? (Section 2.4)
- How can the RFF-based authentication be implemented into ICS systems? (Section 3)

## 1.6 Research Goal

While there does exist a robust set of neural networks for edge devices, the context has been found to be clinical thus far. The capturing of data has usually been in a controlled environment where most variables can be controlled. The devices used are low powered edge devices however, the implementation of RFF onto Remote Terminal Unit (RTU) or Programmable Logic Controller (PLC) devices is sparse. Due to this sparseness, a methodology that details how these techniques will work in the real world is not currently present.

The main goal of this thesis is to determine if RFF can be run on an RTU device for physical layer security in ICS systems. The sub goals that support this are the recording IQ data from wireless ICS communications, an application of RFF onto RTU devices and a proposed future topology to utilize these methods in a real world scenario.

# Chapter 2

# Radio Frequency Fingerprinting: Background

This chapter provides a brief introduction into RFF theory and modern approaches in implementing this technique [9]. Related works will also be addressed and the contributions of this thesis will be covered.

## 2.1 What are Radio Frequencies?

Informally radio frequencies are a set of waves that are used to communicate data over wireless mediums. A radio transmission will consist of a set of waves known as In-phase/Quadrature waves, or simply *I/Q signals*. Any points on this wave are known as an IQ point or sample. The I portion of the signal is a cosine wave and the Q portion is a sine wave. These waves can be altered or manipulated in certain ways to convey information. How this information is conveyed as bits is determined through what are known as *modulation techniques*. These modulation techniques can alter the waves through different techniques such as amplitude shifts, frequency shifts or phase shifts. This can be seen visually later in Figure 2.1, each give unique characteristics or *spectral features*. Analysis of these features without accelerated computational assistance will be referred to as *Traditional Radio Frequency Fingerprinting Approaches*.

## 2.2 What are Radio Frequency Fingerprints?

Just as the name would infer, a radio frequency fingerprint is a unique identifier that is derived from the broadcast of a transmitting device. All devices have this identifier, this is due to the construction of the transmitter circuit of the device. This circuit consists of a power supply, electronic oscillator, modulator, RF amplifier and antenna tuner. Each of these contain components that are manufactured in mass with a given tolerance level. Given that each analog component is not produced exactly identically to one another this gives a unique attribute to each fully assembled device. These characteristics lead to differences in the way the frequencies are transmitted such as alterations to amplitudes or slight delays in transmissions. Just like humans, a fingerprint is very difficult to replicate even if the wave form can be collected. Since the replication of this fingerprint is not practical, this allows for 'biometric' security. Due to the fact that this feature is always carried by the transmission wave, this can be used to continuously identify the device. This enables the physical layer of security. This is essentially a Physical Unclonable Function (PUF). A PUF is a physical object that exhibits specific characteristics due to its physical construction, this enables a 'fingerprint' to be made. As mentioned before, this is what RFF is, a fingerprint based off of physical manufacturing imperfections, so it is a PUF based off of these imperfections.

### 2.2.1 Biometric Example

Take a smartphone for example. This device utilizes the users unique body to authenticate signing in to the device. This can be a thumb print or the users face. It is something that is always carried with them and can be guaranteed to always identify them. Their characteristics may be a wide jaw and a stout nose. A RF fingerprint can be thought of in a similar manner except its facial features will be amplitude and wavelengths.

## 2.3 Traditional RFF Approaches

Traditional RFF approaches consist of analyzing spectral features and deriving unique characteristics. These approaches utilize different parts of the wave such as magnitude and phase. Popular techniques are known as Modulation (Steady-State)-based detection, statistical and transient-based analysis.

### 2.3.1 Steady-state Analysis

Modulation detection is based on the premise of analyzing spectral features, or magnitude and phase characteristics of a frequency. Since some modulation frequencies communicate bits in vastly different ways, it is easy for a human to pick out which signal is different from one another. Figure 2.1 visually demonstrates the differences between Amplitude-Shift Keying (ASK) and Phase-Shift Keying (PSK) modulations, so even the untrained eye can decipher the two.

Though this example in Figure 2.1 demonstrates differentiation between modulations, the meat of the technique utilizes IQ offsets, magnitude and phase errors to detect differences between homogeneous modulated signals. Since these detection methods are based on how close each wave is from one another, methods like K-Nearest Neighbors (KNN) and Support Vector Machines (SVM) can be utilized to accelerate the differentiation. According to [4], success was found with 97% accuracy with a Signal to Noise Ratio (SNR) of 30 decibels (dB) with diminishing returns at 0dB with an accuracy of 66%. Signal to Noise Ratios (SNR) are critical to the extraction of features especially if they rely on finding clear cut patterns like in steady state detection. As we see in Figure 2.2, the signal to noise ratio can disrupt the discernible pattern of the waveform signal.

This is why results can be diminished when signal to noise ratios are poor. Typically, higher RF power translates to better SNR, so higher decibels mean higher signal quality; the acceptable range for this is between 25-40dB. The reason steady state is not viable in all scenarios is due to not all transmitters possessing the “steady-state” part of the

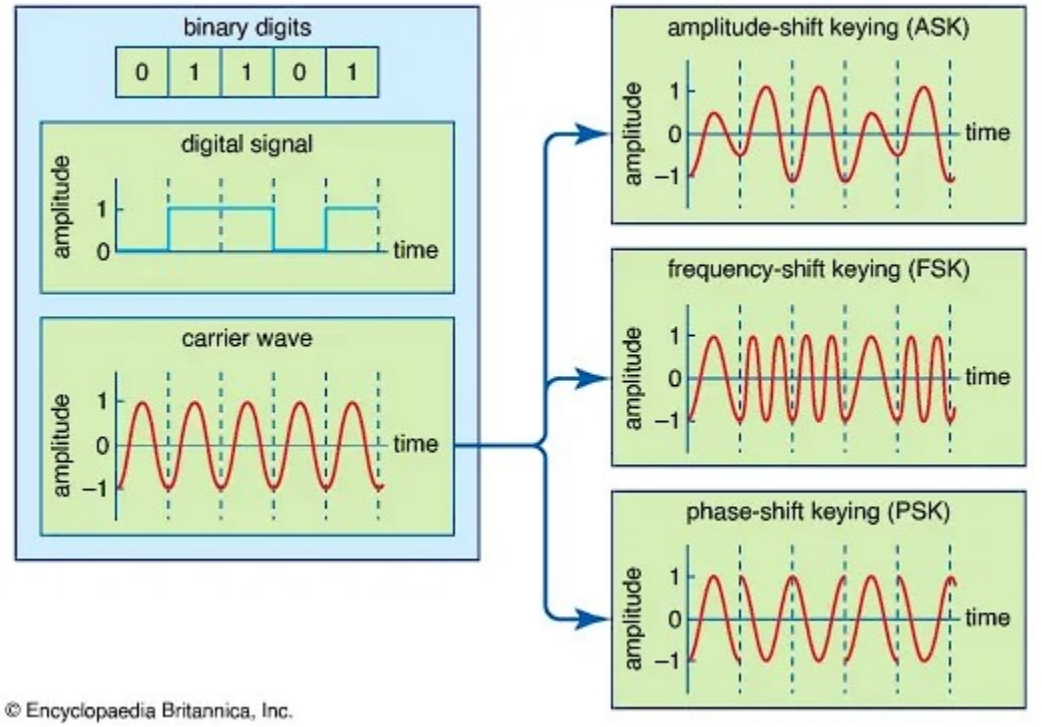


Figure 2.1: Visual demonstration of how different modulation techniques convey bits [20].

transmission signal. This is what made room for other techniques like transient analysis.

### 2.3.2 Transient Analysis

Transient analysis focuses on the "turn-off" to "turn-on" portion of the signal. This takes place before the actual transmission of the data, so this is applicable to all transmitting devices since this phenomenon is recurring. To utilize this technique effectively you will need the start point and a duration of a stable signal in order to extract features.

Once this is determined, extraction of features through the Bayesian Step Change Detection (BSCD) or Phase Detection (PD) can commence. This method started in the early 90's and these techniques are still applicable today through probabilistic neural networks [9]. Although this traditional method has seen success, it relies on the ability to capture the signal with the pre-transmission and a stable signal for a fixed duration of time. This

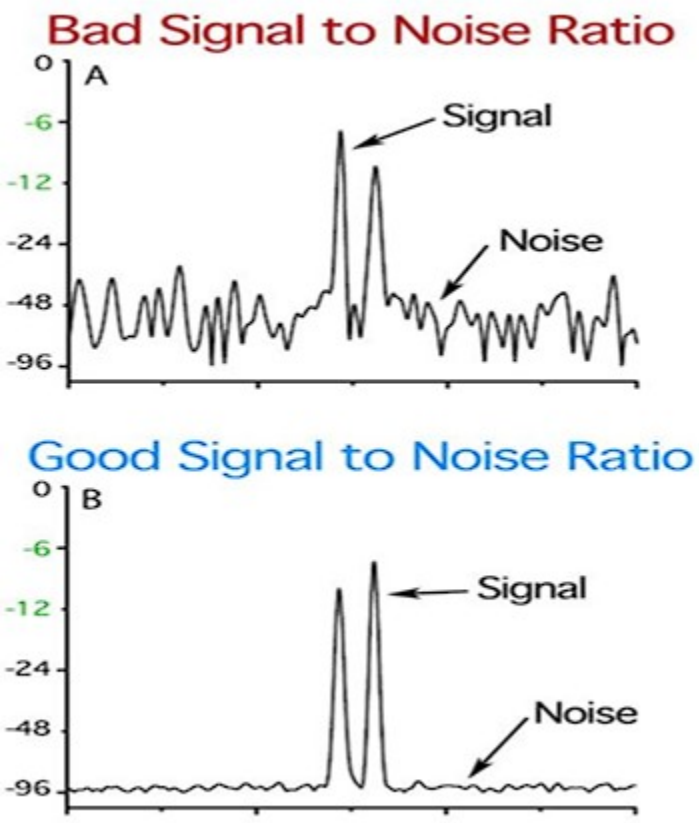


Figure 2.2: Visual demonstration of bad and good SNR [22].

may not always be feasible in networks with heightened noise or multiple devices operating with overlapping sub-bands. This technique relies on the knowledge of when the signal is being communicated since the time before transmission is required for analysis.

## 2.4 Deep Learning Approaches

Traditional RFF approaches such as transient analysis and steady-state modulation have opportunities to be accelerated through neural networks, so these techniques still are being leveraged today. Since computers are excellent with handling convoluted inputs, the analysis of raw IQ points can be harvested and fed into these learning architectures. Through

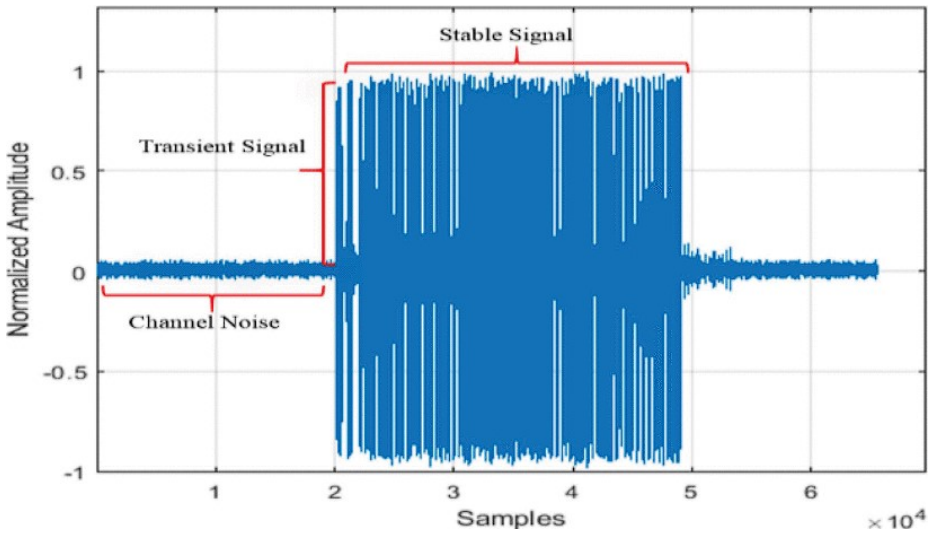


Figure 2.3: Visual demonstration of how each part of a transmission is understood in transient analysis [5].

the assistance of neural networks we can place the interpretation of discerning potentially harder inputs into the machine. Inputs with lesser Signal to Noise Ratios (SNR) may be easier to discern for a machine rather than a human performing the manual analysis. There are many neural networks to choose from that can be applicable to RFF such as Convolutional Neural Networks (CNN), Feed-forward Neural Networks (FNN), Recurrent Neural Networks (RNN) and transformers. Each of these papers, [1], [3], [4], [14], implement traditional approaches that have been accelerated in some way through a neural network. They have been utilized in high end servers that utilize Nvidia V100 Graphical Processing Units (GPUs) and in lower end devices such as Field Programmable Gate Arrays (FPGAs), smartphones and IoT edge devices.

Each of these poses unique insights into different approaches and deployments. Specifically in [1], they show the differentiation of signals between 'identical' Unarmed Aerial Vehicles (UAVs), more specifically DJI Mavic drones. The importance of this lies in SCADA deployments where multiple devices from the same vendor could potentially exist in the same network area. Differentiating between devices with the same design is usually more

difficult due to the similar transceiver circuits being used. The resulting differences or features between the waves are nearly homogeneous in observation.

## 2.5 Related Works

There have been approaches in this field that encompass the use of neural networks. The authors in [1] detail how a 1D CNN can be applied to learning the fingerprints or drones. It was demonstrated that a fingerprints characteristics can be altered dramatically based on the movement of the drone. Their original classifications without drone movement yielded an accuracy of 97%. Once the drones were in motion this accuracy dropped dramatically to the 20-30% range. This drop was circumvented by implementing an aggregate neural network that took the samples of 12 neural networks to yield a combined accuracy of 93%. This is impressive considering that each of the transmitting drones are identical. The goal of this work was to classify drones, so the coverage of stationary devices was not large.

The authors in [3] specifically target the deployment of a model onto edge devices such as FPGAs, a Samsung Galaxy S10 and Nvidia TX2. Specifically pruning was used to yield a high accuracy with an increased efficiency of 27.3 times better results compares to a standard CNN. This efficiency gain was accomplished through compression of convolutional layers and through the use of pruning techniques. This is how they were able to meet the aggressive target of low end devices like a smartphone. This points out that the deployment of a deep neural network is possible with efficacy onto low end devices, though the deployment onto an RTU is not detailed. Each of these low end devices were solely running the neural network model.

The authors in [4] demonstrate a steady state approach where k-Nearest Neighbors classifiers are used. In good SNR recordings (above 25db) there is a perceived 97% accuracy. This decreases in low SNR recordings, at 0db there is a 66% yielded accuracy. It is to be noted that noise will decrease discernible features but adding more bins may help classification. This covers the idea that imperfections in the original data can lead to loss

in performance of RFF techniques.

The authors in [11] cover the effects of pre-proccessing and data collection on the discernibility of captured radio waves. The principles state that the higher the sample points the more likely features will be able to be recognized. Smooth filtering will also increase discernibility by removing bogus points with low pass filtering and high frequency attenuation.

# Chapter 3

# Experimental Methodologies

## 3.1 Proposed Topology

The framework that is going to be used can be visualized in Figure 3.1. This figure maintains the separation of layers similar to Purdue reference model for ICS/SCADA networks. Since the CNN server should not handle any data other than gathered IQ points, there should only be communication between the RTU and the server which only contains IQ data. Since the RTU carries the responsibility of IQ capture, the link between the server and RTU should either be a dedicated link separate from the OT network or a physical transfer through a storage medium. If the RTU device is reachable and has the ability to connect to physical mediums, transfer through a physical medium is preferable. This transfer from server to RTU is only necessary twice. The process for exchanging data shall be as follows:

1. RTU collects intial IQ samples.
2. These samples are sent to the server for model training.
3. The model shall be sent back to the RTU for deployment
4. The RTU deploys this model.
5. The RTU will continue to collect samples for validation against the deployed model.

This process of data exchange will only need to occur once per set of data, if a new device is introduce this process must occur again. This topology was used for the experiments

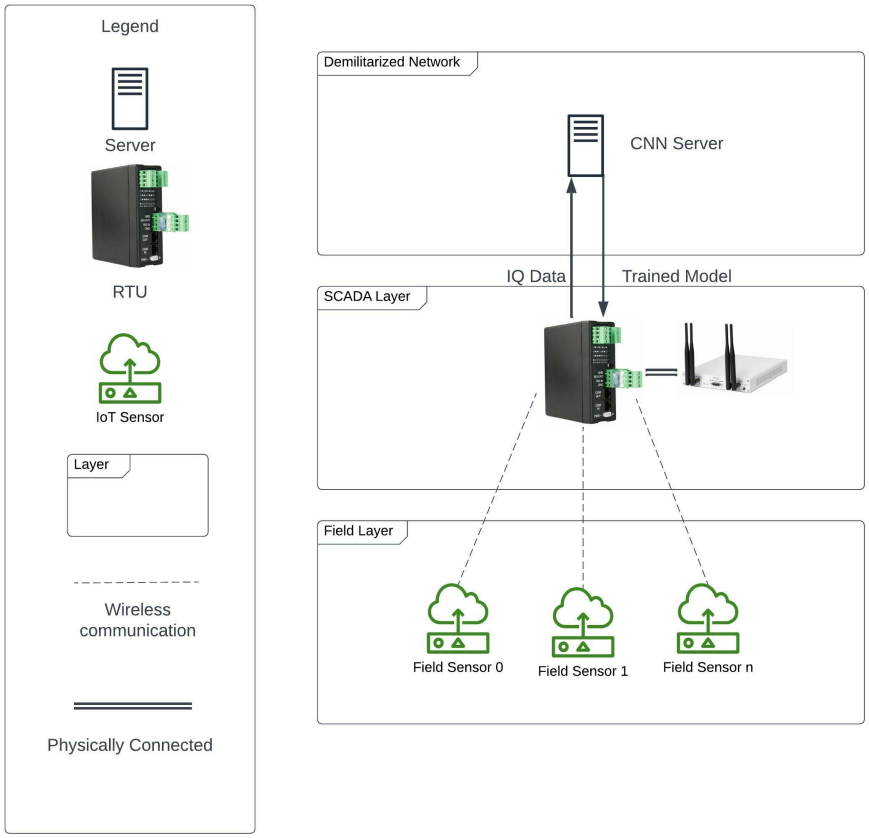


Figure 3.1: Visual representation of the proposed topology

in this thesis. The physically connected USRP device collects the wireless communication from the field sensors, the IQ data gathered is sent to the server and a trained model is sent back for deployment.

## 3.2 Approaches Used

As mentioned in section 2.4, many of the traditional methods are accelerated through neural networks. This work will focus on the use of steady-state analysis through a Convolutional Neural Network. Since 1D CNNs have a superior recognition in applications such as biomedical data classifications [16]. Since they only perform scalar multiplications

and additions, implementations on lower end devices are feasible. This is in part due to the reduction in run-time complexity compared to 2D networks. It is reduced to $O(NK)$ compared to $O(N^2K^2)$ where N represents the dimensions to convolve and K is the kernel used to convolve them. Their success can be seen in [1], [3], [4] where each accomplished a final percentage above 90%.

A 1DConvolutional model will be used since it yields great accuracy and efficiency. The traditional method that will be used in tandem will be steady-state analysis. Steady state is chosen to leverage the availability of raw IQ signals and the visual learning advantages seen in 1D Convolutional networks. The flow of this interpretation can be seen in Figure 3.2. The work found in [1] will be leveraged and updated from TF1.12 to TF2.X so it can run on modern hadrware.

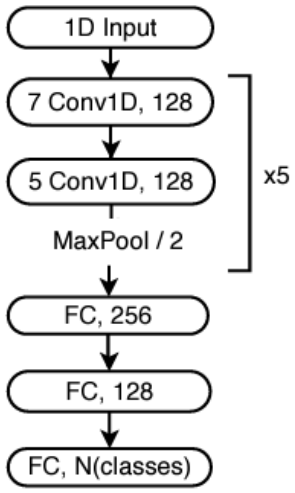


Figure 3.2: Visual representation of how the 1D Convolutional architecture [1].

## 3.3 Limitations

The work found in [1] is explicitly built for the detection of new UAVs, so this work can be used for the detection of new rogue devices like we might see in cloning attacks. This is also

leveraging the lighter 1DAlexnet [16] which will allow for the prospect of deployment on low end devices. This application should allow for the use of RFF on edge with detection of new devices. The capture of frequency broadcasts can be accomplished through Universal Software Radio Peripherals (USRP) devices and GNURadio. These two tools will allow for the capture of broadcasted signals on specific channels that our device will be broadcasting on.

# Chapter 4

# Experiment Setup

As we have learned from our techniques, our physical testbed must reflect the ideals shown in Figure 3.1. With the physical testbed, we capture RF transmissions from field devices, analyze their transmissions with our CNN model and deploy the trained model into the field for authentication on an RTU device. This trained model will include the analyzed fingerprints from the training set and will be compared against gathered transmissions. For our physical testbed, we must understand and break up our complete setup into logical pieces that reflect the real world scenario. The pieces consist of developing a dataset, analyzing this dataset and deploying the dataset. It is to be noted that the key differences between the original and the updated model is as follows: the meaningful differences is the data being fed into the nerual network and the new model is written in TF 2 instead of TF 1.12 of the original model. In the updated model we are feeding in our gathered dataset to get its respective results.

## 4.1 Building the Dataset

In our experiment, we will be utilizing two datasets: an established dataset in a clinical environment [1] and a dataset that we gathered ourselves from our own physical testbed. The first dataset consists of Ettus N210 USRP devices with VERT2450 antennas over the 2.432GHz band. There were a total of ten unique USRP devices and ten separate respective thirty-second transmission captures.

For our dataset, we followed the similar procedures from the clinical capture to eliminate variance and to establish a known metric. These procedures are the thirty-second

transmission intervals and capture these devices' transmissions individually. In order to build a dataset that is similar to the data found in [1], we use multiple transceivers to fingerprint. This was accomplished by utilizing ten Wi-Fi transceivers (TL-WN722N). Each of these transceivers are identical devices, but they maintain hardware uniqueness from their manufacturing imperfections. Considering the original dataset consisted of identical DJI Mavic drones, this should work well. These transceivers will be connected to a Raspberry Pi 4 that will communicates over Wi-Fi to a router. The pi with different transceivers will represent field devices that will have their transmissions captured. Both of the datasets resulted in large binary files that stored the IQ samples byte-by-byte in little endian format.

Up until now, we presented the details of the physical setup that will be used to transmit our data. Now, we describe how we capture this transmission data. We will use our own USRP device (Ettus B210 with VERT245 Antennas) in tandem with GNURadio to capture the Wi-Fi transmission. This capture will utilize the flow shown in (Figure 4.1). As we can see from the flow, it captures packets based on a channel that we specify. This channel is found by analyzing the Wi-Fi subbands that are being utilized by our transceivers (Figure 4.2). It is ensured that the communicating device will have its own dedicated band that no one else will be able to interfere with for the capture. The capture sampling rate was set to 20MHz. The rate of capture is important since it dictates how much information can be used to discern differences from transmission to transmission. The data density of the transmission can be compared to the number of pixels per inch (PPI) in a photo; 20MHz corresponds to a high PPI photo and 5MHz to a low PPI photo (Figure 4.3).
Our procedure for capture is as follows:

1. Establish connection between our router and Raspberry Pi with transceiver
2. Find which Wi-Fi subband is in use and set the USRP to receive on that subband
3. Start a thirty-second transmission window and start receiving simultaneously
4. Conclude capture has been recorded in a binary (.bin)

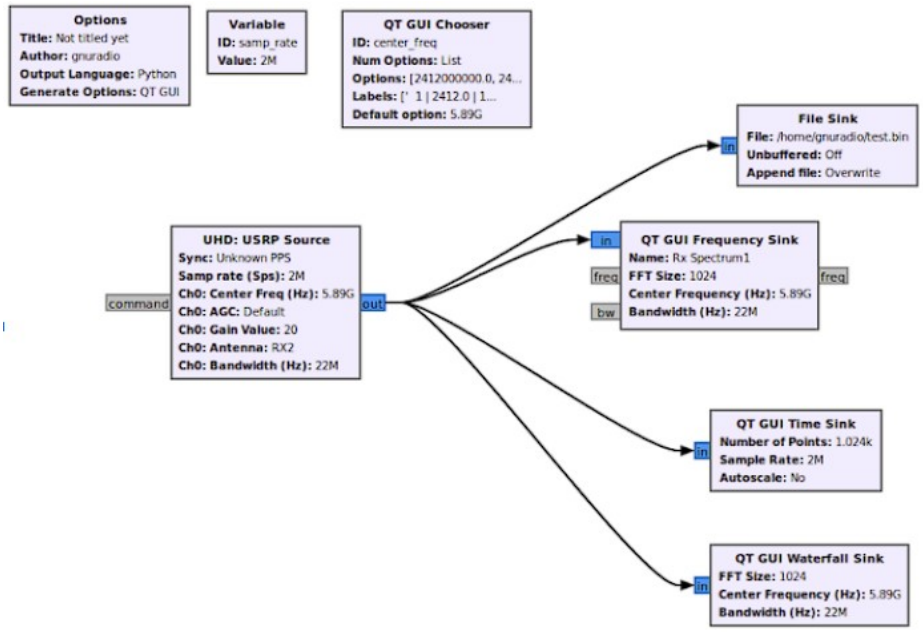


Figure 4.1: Visual analogy of how a higher sampling rate can be compared to a lower sampling rate

5. Repeat the previous steps for every transceiver in our set

The subband we captured over was subband 6 (2426-2448MHz, Figure 4.2).

## 4.2 Analyzing the Dataset

Now that our datasets have been established we can move onto the analysis of our data with our CNN. The structure of our model is a regurgitation of the established model used by RFDatafactory[1]. Their model was originally used for UAVs and it utilized Tensorflow (TF) 1.12. To accomplish all of our goals, we modernized this model by using TF 2.X. The server hardware used is as follows: Threadripper 2970WX, 128GB ECC RAM, and RTX 3080. To utilize the RTX 3000 series, TF 2.X is the only version that allows for hardware acceleration over the Ampere architecture. In the TF 1.12 testbed, [1] utilized Compute Unified Device Architecture (CUDA) 10.0 capable cards. The structure of the CNN can be

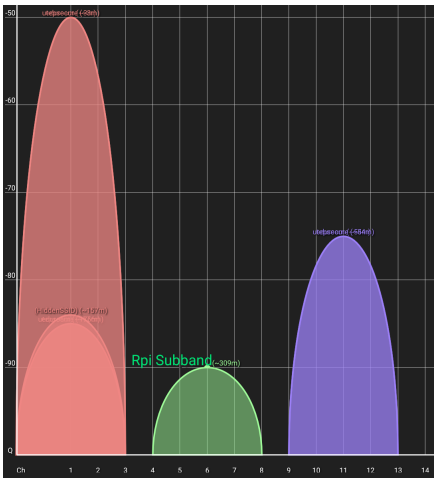


Figure 4.2: Visual analogy of how a higher sampling rate can be compared to a lower sampling rate

seen in Section 4.4.1, where AlexNet is being utilized. The model took a quicker amount of time to train and test compared to the original model's deployment as seen in Figure 4.4. This is due to the faster hardware used in our setup.

After preprocessing, we trained and tested the 1DAlexNet model using 100 epochs, early stopping, and a slice size of 200; 80% of the dataset was for training and 20% for testing. These settings were used across the original model and the updated model, the difference being the datasets used. The original used the dataset from [1] and the updated used the dataset that was gathered.

### 4.2.1 Preprocessing

Preprocessing is a necessary step in the the ingestion of data into our neural network. It cleans the data with different methodologies to ensure that the output data is as true to the original as it should be. In our case our methods include band equalization and slicing. Band equalization ensures that the resulting waves are within a known acceptable range, this removes bogus data points that can harm the learning process. Slicing is an important step in feeding data to a CNN specifically, this architecture requires a fixed input size.

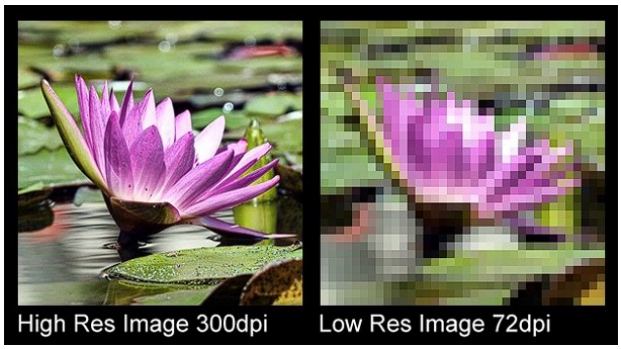


Figure 4.3: Visual analogy of how a higher sampling rate can be compared to a lower sampling rate

| AlexNet1D | Original Model | Updated Model |
|---|---|---|
| CPU Only 100 epoch (h) | N/A | 32 |
| GPU 10 epoch (h) | N/A | 11 |
| GPU 50 epoch (h) | N/A | 13.9 |
| GPU 100 epoch (h) | 23 | 16.2 |
| No. of Parameters | ~1.1 Million | ~1.1 Million |
| Final accuracy | 97% | 84% |

Figure 4.4: Training and validation times and best final accuracies of the original model and the updated model

Slicing ensures that every portion of the data that is being fed into the network are of identical and equal sizes.

## 4.3 Deploying the Dataset

Once the training and testing from our CNN is complete we will convert the TF model into a TFLite model for deployment on edge devices. Once we have this model converted, we can run this on the Raspberry Pi as our simulated PLC device. The Raspberry Pi has an instance of OpenPLC with a traffic light ladder logic program running to simulate the work of an RTU (Figure 4.5). There is also an instance of GNURadio running that

is capturing RF samples in real time for testing through the TFLite model. To make simultaneous running instances possible, we utilize a Google Coral USB accelerator, which is a TPU that is purpose-built for accelerating TFLite models on edge devices. In addition to USB, Google Coral comes in multiple form factors such as m.2, mini pci-e and dedicated development boards. This ensures that regardless of RTU I/O there is a form factor that can suit its needs. To summarize the steps, the following is the deployment flow procedure:

1. The RTU device will capture RF samples.
2. These samples will be written to a file for the TFLite model.
3. The TFLite model will ingest this data and output its confidence.
4. If the confidence is sufficient, we continue; otherwise we raise a flag.
5. Repeat this process until stopped.

## 4.4 Results

In this section, we present our observations of performance metrics from running the model on our edge device after training the neural network on the CNN server.

### 4.4.1 CNN Server

The first result we look at is our CNN TF model accuracy in comparison to its original model seen in Figure 4.4. The original model accuracy was obtained by using the dataset provided from [1] and the updated model used the gathered dataset. The updated model was able to correctly identify the gathered transmissions 84% of the time. The experiment shows that the CNN can feasibly replicate passable results across multiple datasets including the captured dataset. Though as seen in Figure 4.4, the results were not perfectly consistent with the original model. Since the settings were consistent across the models, the issue must

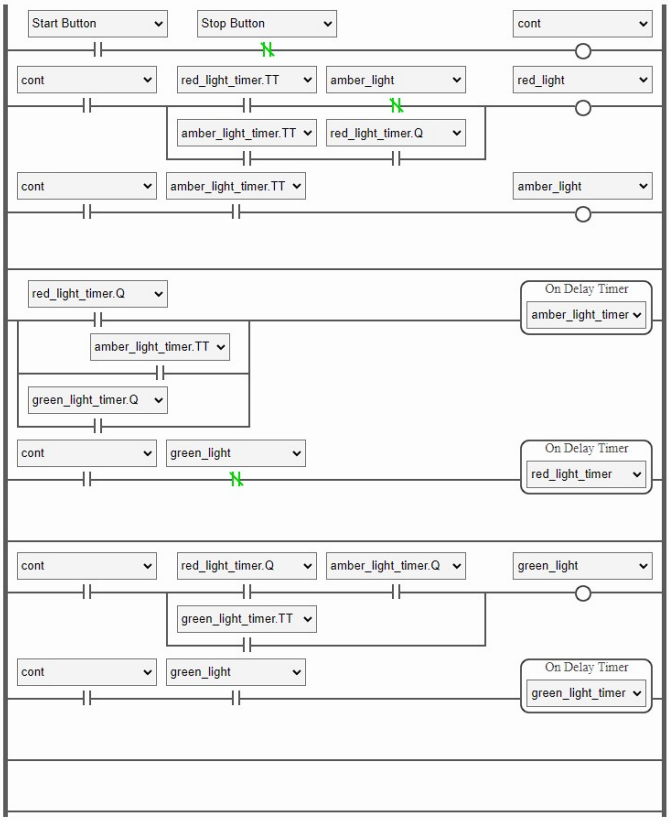


Figure 4.5: Ladder logic program that is ran in OpenPLC

lie in the collection of data or the devices themselves. This claim is backed up through Figure 4.6, this eliminates the possibility of the neural network having insufficient hyper parameters.

In the first runs it is noted that under-fitting of data may have occurred since the accuracy improved as epochs increased. The same hyper parameters were also utilized in the original model which leads to an issue in data. During the time of collection the distance between the devices may have been sub optimal as there were metallic objects and computers near and between the broadcasting devices. With this information, we can move on to analyzing the feasibility of running this model on an edge device. The final structure of the model can be seen in Table 4.1, this shows the distribution of parameters per layer of the architecture. This shows how many values the algorithm can alter independently as

Table 4.1: This table shows the structure of the CNN. Each row is a layer with its respective shape and number of parameters

| Layer (type) | Output Shape | Param Num. |
|---|---|---|
| conv1d$_1$($Conv1D$) | (None, 256, 128) | 1920 |
| conv1d$_2$($Conv1D$) | (None, 256, 128) | 82048 |
| max pooling1d$_1$($MaxPooling1D$) | (None, 128, 128) | 0 |
| conv1d$_3$($Conv1D$) | (None, 128, 128) | 114816 |
| conv1d$_4$($Conv1D$) | (None, 128, 128) | 82048 |
| max pooling1d$_2$($MaxPooling1D$) | (None, 64, 128) | 0 |
| conv1d$_5$($Conv1D$) | (None, 64, 128) | 114816 |
| conv1d$_6$($Conv1D$) | (None, 64, 128) | 82048 |
| max pooling1d$_3$($MaxPooling1D$) | (None, 32, 128) | 0 |
| conv1d$_7$($Conv1D$) | (None, 32, 128) | 114816 |
| conv1d$_8$($Conv1D$) | (None, 32, 128) | 82048 |
| max pooling1d$_4$($MaxPooling1D$) | (None, 16, 128) | 0 |
| conv1d$_9$($Conv1D$) | (None, 16, 128) | 114816 |
| conv1d$_1$0($Conv1D$) | (None, 16, 128) | 82048 |
| max pooling1d$_5$($MaxPooling1D$) | (None, 8, 128) | 0 |
| flatten$_1$($Flatten$) | (None, 1024) | 0 |
| dense$_1$($Dense$) | (None, 256) | 262400 |
| dense$_2$($Dense$) | (None, 128) | 32896 |
| dense$_3$($Dense$) | (None, 7) | 903 |

it learns. The greater the number, the longer it will take to compute.

| AlexNet1D | Updated Model Accuracy | Early Stopping |
|---|---|---|
| CPU Only 100 epoch (h) | 15% | No |
| GPU 10 epoch (h) | 45% | No |
| GPU 100 epoch (h) | 84% | Yes |

Figure 4.6: Training and validation times and best final accuracies of the original model and the updated model

### 4.4.2 Edge Device

After loading all of the necessary components onto the edge device, performance metrics were gathered to see the resource utilization. This is reflected in Figure 4.7, here it can be observed that there is still headroom available on the device with TPU acceleration. It is to be noted that the classification of data was using existing pre-proccessed data instead of the IQ data being gathered in real time.

Lastly, analysis of the run time accuracy must be observed. The RF sample capture speed had to be throttled down to 5MHz compared to the 20MHz used for capture in Section 4.1 as the edge device could not analyze the larger stream of data due to the constraints placed on the CPU by the Neural Network. The Random Access Memory (RAM) utilization was not near the full capacity of the device and is notated through Figure 4.8. The RAM utilization figure is important because some RTU devices only function on a 32-bit architecture which has a limit of 4GB of RAM.

Our simulated RTU also shares this 4GB RAM limitation for continuity. Swap partitions are being utilized for both TensorflowLite and GNURadio to reduce RAM utilization. This leads to the load being shifted onto the storage medium rather than volatile memory.

Since the CPU and RAM utilization are not fully in use, we can conclude that it is possible to run RFF on a simulated RTU device. The architecture used on this device is ARM based which is popular amongst cost effective devices. The CPU used is a Cortex A72 with a 64-bit architecture. This performance can also be found in SIMANTIC S7-1500 controllers, these devices are able to run 64-bit versions of windows server. These versions

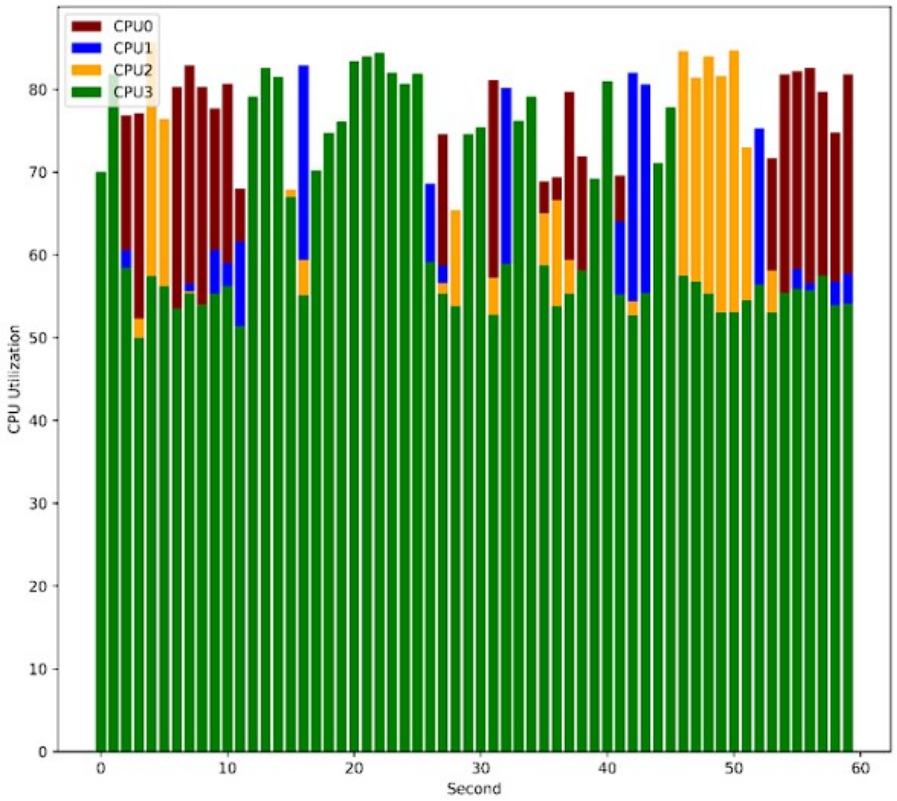


Figure 4.7: CPU Utilization on all four cores of the simulated RTU device

of windows require at least 512MB of RAM whilst our utilization is sitting below it at 492MB. For these controllers the findings are transferable based on our architectures, CPU and RAM utilization.

## 4.5 Comparison to another Neural Network Architecture

It is to be noted that CNNs are no longer the cutting edge of Neural Networks, Visual Transformers (ViT) have also seen success in image processing in different ways than CNNs. They are also lighter than CNNs and have been seen to use less parameters in [17]. This

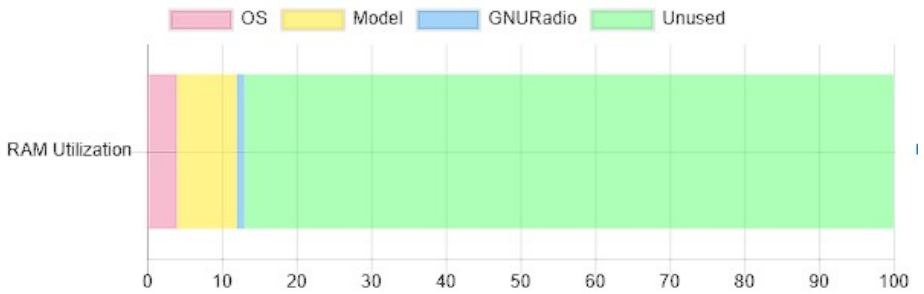


Figure 4.8: RAM Utilization bar graph of the simulated RTU device. The total amount of memory used is 492.52MB or 12% of total RAM available

paper exclusively uses ViT for RFF with results in the 90th percentile. CNNs are still popular and can be used in tandem with ViTs for better performance [18], this has yet to be seen within the RFF field.

# Chapter 5

# Edge Devices with RFF - Future

## 5.1 Conclusions

As we have seen from our results on our edge server, the Cortex A72 processor (processor found in the Raspberry Pi 4) is able to handle a model with over a million parameters at a passable accuracy on a less clinically gathered dataset. This shows the validity of deploying an RFF model onto an edge RTU device. Since other edge devices are getting access to performant hardware, like the Wago Touch Panels 600, there is a direct applicability to testing and deploying this on the field. The Wago Touch Panels possess a Cortex A9 processor, this is a somewhat similar chip to the tested A72 chipset. It allows for the use of 4GB of RAM, 4 logical cores for multitasking and the ability to use docker to deploy images. Due to these similarities, it is likely that we can use the existing work on real hardware because of compatibility. In our tests, we did not exceed the 4GB of RAM on this device, and it was able to handle a rather large data model.

## 5.2 Research Questions and Shortcomings

The procedure used to capture our dataset was successful in capturing meaningful data from broadcasted wi-fi signals, however it could have contributed to the lower accuracy when we passed it through the CNN. This could be due to other environmental factors such as electromagnetic interference, other miscellaneous signals and general noise. Though the procedure we utilized through GNURadio was functional, it can be improved for better testing and training accuracy. Though the results are not as high as the original model due

to our captured dataset imperfections, the training portion and testing was successful. This demonstrates the power of new hardware through the reduction of training and testing time which in turn will become cheaper in both time and money as this improves. As mentioned in section 4.3-4.4, there is a successful deployment of a CNN onto an edge RTU device. Since the RAM utilization is well below the 4GB threshold, implementation onto 32-bit operating systems found in modern RTUs is absolutely feasible. The concern would be the CPU utilization on 32-bit systems, a dense network may be cumbersome but efficient implementations of transformer-based architectures may still be possible. To address the data collection methodologies, this leaves room to question; how can data be collected properly in a real world environment where similar obstructions may exist? There is also the shortcomings of only being able to capture at a sampling rate of 5MHz, this will be a challenge for actual deployment since it is necessary to capture all details of a RF transmission for a high degree of accuracy. This can potentially be overcome by reducing resource usage of the neural network by using a more optimized model. There is also the challenge of how this is deployed onto existing hardware. There is also the issue of what data is being collected to be fed into the neural network. There could be sensitive data that is captured as a wave, then a static copy of this data exists, this is why a secure topology is necessary to transfer this data. The proposition of exchanging data with either a dedicated link or through a physical medium from data collection to server is paramount. This proposed topological view was utilized for the experimental setup and shows viability but it is yet to be tested in a real ICS system environment. This entire implementation is also relying on the existing high end and newly released RTU devices. Not all networks will be able to adopt this proposed methodology if the hardware does not exist to deploy this on. Lastly, the capture of data is facilitated through costly USRP devices, this cost can be overcome if RTU devices allow for direct access to their wireless antennas for raw IQ capture.

## 5.3 Future Work

Since the scope of this paper was only concerning one edge device with one type of architecture, it would be important to start to test the abilities of other Neural Networks such as transformers. As we have seen in [18], transformers are a less dense architecture with less parameters to achieve passable results. With this information, it is likely that integration can take place on actual hardware despite the performance delta between them. Once this has been integrated onto real hardware with a leaner architecture, the real time performance can be evaluated. It would be ideal to test this in an isolated controlled environment as a proof of concept. This test would consist of regular communication between the RTU and its known trained devices as a baseline and the introduction of a rogue device attempting to communicate with the RTU. Considering that Wi-Fi is not the only RF communication standard, the exploration of fingerprinting other common radio frequencies must be taken into account. After this has been completed, the construction of a physical/virtualized testbed would be necessary for the final component. The final component would be to establish an Intrusion Detection System that uses the real time reports gathered from our edge device to alert operators to suspicious activities. This will also use the topology proposed to be implemented into the ICS network. The work accomplished in this paper is the first step towards the future integration of an Intrusion Detection System based on RFF.

# References


[1] Nasim Soltani, Guillem Reus-Muns, Batool Salehi, Jennifer Dy, Stratis Ioannidis, and Kaushik Chowdhury, "RF Fingerprinting Unmanned Aerial Vehicles with Non-standard Transmitter Waveforms," Accepted in IEEE Transactions on Vehicular Technology, Nov. 2020.

[2] Y. Peng, *Active and passive control auto-switching research based on heartbeat protocol for long-distance natural gas pipeline SCADA system*, Proceedings of the 2016 International Forum on Energy, Environment and Sustainable Development.

[3] T. Jian et al., "Radio Frequency Fingerprinting on the Edge," in: *IEEE Transactions on Mobile Computing*, , vol. 21, no. 11, pp. 4078-4093, 1 Nov. 2022.

[4] I. O. Kennedy, P. Scanlon, F. J. Mullany, M. M. Buddhikot, K. E. Nolan, and T. W. Rondeau, *Radio transmitter fingerprinting: A steady state frequency domain approach,* in: *Proc. IEEE 68th Veh. Technol. Conf.*, pp. 1–5, 2008.

[5] N. Soltanieh, Y. Norouzi, Y. Yang and N. C. Karmakar, "A Review of Radio Frequency Fingerprinting Techniques," in: *IEEE Journal of Radio Frequency Identification*, vol. 4, no. 3, pp. 222-233, Sept. 2020.

[6] O. Ureten and N. Serinken, "Wireless security through RF fingerprinting," Can. J. Elect. Comput. Eng., vol. 32, no. 1, pp. 27–33, 2007.

[7] S. Stone, M.l Temple, *Radio-frequency-based anomaly detection for programmable logic controllers in the critical infrastructure,*, in: *International Journal of Critical Infrastructure Protection,* Volume 5, Issue 2, 2012, Pages 66-73, ISSN 1874-5482.

[8] T. Bartman and K. Carson, *Securing communications for SCADA and critical in-*

*dustrial systems,* in: *2016 69th Annual Conference for Protective Relay Engineers (CPRE)*, 2016, pp. 1-10.

[9] A. Jagannath, J. Jagannath, P. S. Pattanshetty Vasanth Kumar, "A comprehensive survey on radio frequency (RF) fingerprinting: Traditional approaches, deep learning, and open challenges," Computer Networks, Volume 219, 2022, 109455, ISSN 1389-1286.

[10] G. Shen, J. Zhang, A. Marshall, M. Valkama and J. Cavallaro, "Radio Frequency Fingerprint Identification for Security in Low-Cost IoT Devices," in: *2021 55th Asilomar Conference on Signals, Systems, and Computers*, 2021, pp. 309-313.

[11] W. Wang and L. Gan, "Radio Frequency Fingerprinting Improved by Statistical Noise Reduction," in: *IEEE Transactions on Cognitive Communications and Networking*, vol. 8, no. 3, pp. 1444-1452, Sept. 2022.

[12] E. -K. Lee, M. Gerla and S. Y. Oh, *Physical layer security in wireless smart grid,* in: *IEEE Communications Magazine* , vol. 50, no. 8, pp. 46-52, August 2012.

[13] E. Uzundurukan, Y. Dalveren, A. Kara., *A Database for the Radio Frequency Fingerprinting of Bluetooth Devices,* 2020.

[14] G. Shen, J. Zhang, A. Marshall, L. Peng, X. Wang, *Radio Frequency Fingerprint Identification for LoRa Using Spectrogram and CNN,* in: *Proc. IEEE INFOCOM,* 2021, accepted.

[15] Wlazlo, P., et al.:, *Man-in-the-middle attacks and defence in a power system cyber-physical testbed* IET Cyber-Phys. Syst., Theory Appl. 6( 3), 164– 177 (2021).

[16] S. Kiranyaz, O. Avci, O. Abdeljaber, T. Ince, M. Gabbouj, D. J. Inman, *1D convolutional neural networks and applications: A survey, Mechanical Systems and Signal Processing,* Volume 151, 2021, 107398, ISSN 0888-3270.

[17] 2020 Smart Grid System Report. Department of Energy . (2022, January).

[18] J. Guo, and K. Han, W.Han, Y. Tang, X. Chen, Y. Wang, C. Xu, *CMT: Convolutional Neural Networks Meet Vision Transformers,* in: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* , pp. 12175-12185, 2022.

[19] T. Dinh *Energy consumption forecasting in energy management systems* , Jan. 2021

[20] J. S. Lehnert, W. E. Stark, D. E. Borth *Modulation* , Encyclopaedia Britannica

[21] S. Singh *IT vs ICS* , Infosec, 2019

[22] *SIGNAL AND NOISE: COSMIC MUONS* Quarknet, 2019

# Curriculum Vita

Evan White received his Bachelor in Computer Science degree from UTEP in Fall 2021. He has served as a research assistant under the instruction of Dr. Deepak Tosh since 2019 under UTEP's Department of Computer Science. In the summers of 2021 and 2022 he served as an intern for Sandia National Laboratories. During his studies at UTEP, he has co authored one paper that was published:

Rivera, Abel White, Evan Acosta, Jaime Tosh, Deepak. (2022). Enabling Device Trustworthiness for SDN-Enabled Internet -of- Battlefield Things. 1-7. 10.1109DSC54232.2022.9888903.

He will be seeking a PhD after completion of his Masters at UTEP. He will hope to further his findings from this paper in his doctoral research.