2022-12

# SORTING OF RADIO SIGNALS USING ADVERSARIAL MACHINE LEARNING

Valeske, Jessica M.

Monterey, CA; Naval Postgraduate School

https://hdl.handle.net/10945/71559

# NAVAL POSTGRADUATE SCHOOL

**MONTEREY, CALIFORNIA**

# THESIS

**SORTING OF RADIO SIGNALS USING ADVERSARIAL MACHINE LEARNING**

by

Jessica M. Valeske

December 2022

Thesis Advisor: Frank E. Kragh
Co-Advisor: James W. Scrofani

**Approved for public release. Distribution is unlimited.**

THIS PAGE INTENTIONALLY LEFT BLANK

| REPORT DOCUMENTATION PAGE | | | | *Form Approved OMB No. 0704-0188* |
|---|---|---|---|---|
| Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington, DC, 20503. | | | | |
| **1. AGENCY USE ONLY** ***(Leave blank)*** | **2. REPORT DATE** December 2022 | **3. REPORT TYPE AND DATES COVERED** Master's thesis | | |
| **4. TITLE AND SUBTITLE** SORTING OF RADIO SIGNALS USING ADVERSARIAL MACHINE LEARNING | | | | **5. FUNDING NUMBERS** |
| **6. AUTHOR(S)** Jessica M. Valeske | | | | |
| **7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)** Naval Postgraduate School Monterey, CA 93943-5000 | | | | **8. PERFORMING ORGANIZATION REPORT NUMBER** |
| **9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)** N/A | | | | **10. SPONSORING / MONITORING AGENCY REPORT NUMBER** |
| **11. SUPPLEMENTARY NOTES** The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government. | | | | |
| **12a. DISTRIBUTION / AVAILABILITY STATEMENT** Approved for public release. Distribution is unlimited. | | | | **12b. DISTRIBUTION CODE** A |

**13. ABSTRACT (maximum 200 words)**

Signals Intelligence depends on signal classification accuracy. Artificial intelligence is a tool that allows for the fast and accurate identification of communications signals. Neural networks utilize a set of training data to learn patterns in datasets for recognition and classification. This learning is pivotal to the performance of the neural network and is dependent on the accuracy of the training data used to train. In this thesis, a strong and realistic communications training dataset is developed using MATLAB. It incorporates realistic and real-world factors that more accurately represent a radio frequency (RF) communication signal, then tests the neural network against the newly developed signals to prove the accuracy of the technology. The dataset is also varied in modulation type to fully represent the spectrum of signals to be analyzed by the neural networks.

| **14. SUBJECT TERMS** | | | **15. NUMBER OF PAGES** |
|---|---|---|---|
| generative adversarial networks, GAN, identification and classification of signals, radio frequency, RF | | | 59 |
| | | | **16. PRICE CODE** |
| **17. SECURITY CLASSIFICATION OF REPORT** Unclassified | **18. SECURITY CLASSIFICATION OF THIS PAGE** Unclassified | **19. SECURITY CLASSIFICATION OF ABSTRACT** Unclassified | **20. LIMITATION OF ABSTRACT** UU |

NSN 7540-01-280-5500 Standard Form 298 (Rev. 2-89) Prescribed by ANSI Std. 239-18

THIS PAGE INTENTIONALLY LEFT BLANK

**Approved for public release. Distribution is unlimited.**

**SORTING OF RADIO SIGNALS USING ADVERSARIAL MACHINE LEARNING**

Jessica M. Valeske
Lieutenant, United States Navy
BS, United States Naval Academy, 2016

Submitted in partial fulfillment of the
requirements for the degree of

**MASTER OF SCIENCE IN ELECTRICAL ENGINEERING**

from the

**NAVAL POSTGRADUATE SCHOOL**
**December 2022**

Approved by: Frank E. Kragh
Advisor

James W. Scrofani
Co-Advisor

Douglas J. Fouts
Chair, Department of Electrical and Computer Engineering

THIS PAGE INTENTIONALLY LEFT BLANK

# ABSTRACT

Signals Intelligence depends on signal classification accuracy. Artificial intelligence is a tool that allows for the fast and accurate identification of communications signals. Neural networks utilize a set of training data to learn patterns in datasets for recognition and classification. This learning is pivotal to the performance of the neural network and is dependent on the accuracy of the training data used to train. In this thesis, a strong and realistic communications training dataset is developed using MATLAB. It incorporates realistic and real-world factors that more accurately represent a radio frequency (RF) communication signal, then tests the neural network against the newly developed signals to prove the accuracy of the technology. The dataset is also varied in modulation type to fully represent the spectrum of signals to be analyzed by the neural networks.

THIS PAGE INTENTIONALLY LEFT BLANK

# TABLE OF CONTENTS

# LIST OF FIGURES

THIS PAGE INTENTIONALLY LEFT BLANK

# LIST OF TABLES

THIS PAGE INTENTIONALLY LEFT BLANK

# LIST OF ACRONYMS AND ABBREVIATIONS

| | |
|---|---|
| COMINT | Communications Intelligence |
| CNN | Convolutional Neural Network |
| GAN | Generative Adversarial Network |
| Hz | Hertz |
| MSI | Mobile Subscriber Identifier |
| NN | Neural Network |
| QPSK | Quadrature Phase Shift Keying |
| RF | Radio Frequency |
| RNN | Recurrent Neural Network |
| SNR | Signal to Noise Ratio, $E_b/N_o$ |

THIS PAGE INTENTIONALLY LEFT BLANK

# I. INTRODUCTION

Wireless communication has become vital and accessible to almost everyone. From the everyday user to the military, there have been thousands of technological advances in the field that have allowed users all over the world to take advantage of the radio frequency (RF) spectrum. However, the increase in users has caused congestion of the spectrum and an increase in interference. Receivers must now sift through a multitude of signals within their observed frequency band, as well as compensate for the interference and noise that comes inherently with the technology. As all this information flows in, it has become time consuming and difficult to pinpoint pertinent information. In military operations, it is important to be able to identify these vital pieces of information as signals of interest (SOI) and disseminate them to the appropriate organization or units. The process of intercepting and analyzing communication information is the military field of communications intelligence (COMINT) [1]. With the increase in technical ability of our advisories, this field has become increasingly important to military operations; however, as the wireless communication environment becomes saturated and interference increases, it is increasingly difficult for the COMINT community to accomplish their mission quickly and successfully.

One of the major technologies that has been developed to aid in the classification of wireless communication signals is machine learning in the form of neural networks (NN)[2]. These machine learning processes can learn from large amounts of training datasets to quickly identify incoming signals without the need for human effort. The technology itself is promising; however, obtaining the suitable training data has proven to be a challenge to classification success. The NN uses the training data to make future decisions on unknown signals, so if the training data is inaccurate or not sufficiently robust, the training will suffer and the accuracy of the NN may be unacceptable [2].

## A. PROBLEM STATEMENT

The use of NN shows promise; however, it must be tested to the full extent for real-world application. To fully incorporate these realities into the training data, there must be a realistic analysis of all possible distortions experienced by the RF signal and these need to be accurately incorporated into the signal generation for the NN to use. The signals that are to be analyzed by the generative adversarial network (GAN) in real-world scenarios are going to be large, unordered datasets among a sea of thousands of other signals. Signals are sorted by frequency bands; however, even within the frequency band, there are thousands of other signals being transmitted. Finding a singular signal of interest hidden within a large number of signals is the problem.

Historically this problem was solved via human analysis, but with the increase of signals and their complexity, it has become too costly and time consuming. Prior work addressed this problem using a generative adversarial network; however, we wish to do so with a dataset that is a higher fidelity representation of actual collected signals [3]. We envision a collection system that identifies signals just after collection and down conversion, but prior to the demodulation. The dataset will incorporate unintended signal variations inherent to the down conversion and sampling process. The receiver records the phase changes of incoming signals, but there is no way for it to know the initial phase for reference. Since the collection equipment does not synchronize to the incoming signal, the initial sample time may not occur exactly coincident with the beginning of the signal. The frequencies produced by local oscillators in transmitters and the collection equipment vary slightly from their design values causing small carrier frequency errors. All these signal variations inherent to the equipment and reordering process are experienced by the signals and must be incorporated to test the ability of the GAN to classify realistic collected signals.

This research creates a robust training dataset that accurately simulates realistic signals and demonstrates that a GAN-based sorter can sort signals of interest from a wider set of signals automatically with high accuracy. The technology works well and should be tested in an operationally relevant environment using signals collected over the air to verify the technology is ready for operational use.

## B. THESIS ORGANIZATION

The technique for signal recognition presented in this thesis has two main components, neural networks, and RF signals. Neural networks are a vital piece of our classification process; however, the primary focus of the research will be on dataset generation and real-world variations incorporated. Chapter II is broken into three sections, describing NN and RF signal modulation and signal variations. Section II A follows a building block approach that begins with the neuron and culminates in a description of the generative adversarial network and how it is used in this research. Section II B begins by providing background on the theory of RF signals and how they are used in communications. In section II C, the QPSK signal is introduced, followed by the modulation and demodulation processes. The chapter ends with the section on contributors to noise and signal variations within the QPSK signal at the receiver. Chapter III includes an explanation of the dataset generation and the implementation of signal variations. This chapter also details the experiment setup and the variables implemented in each experiment. Chapter V reports the results of NN accuracy in recognizing signals in each experiment and an analysis of those results. Chapter VI summarizes the research and results, then gives suggestions and topics for future research.

THIS PAGE INTENTIONALLY LEFT BLANK

# II. THEORY

Machine learning is a recent science that has been utilized across all scientific disciplines. From the medical field to operations research, machine learning has proven to be an exceptional tool for solving complicated problems quickly. Machine learning incorporates many types and styles of algorithms that accomplish the tasks required. One subset of machine learning is the neural network.

## A. NEURAL NETWORKS

Neural network, or NN, are an artificial intelligence technique loosely modeled after the human brain. The brain is comprised of layers of neurons that take inputs (dendrites) and produce an output signal (axon) to pass onto other neurons [2]. The input/output device concept is applied to the computer science environment in the form of neurons in a neural network.

### 1. The Neuron

An artificial neuron is the base unit for NN. They are the singular unit that take an input, then produce an output to pass downstream to the next neuron. Figure 1 shows a biological neuron in the human brain versus a simple single layer computational neuron.

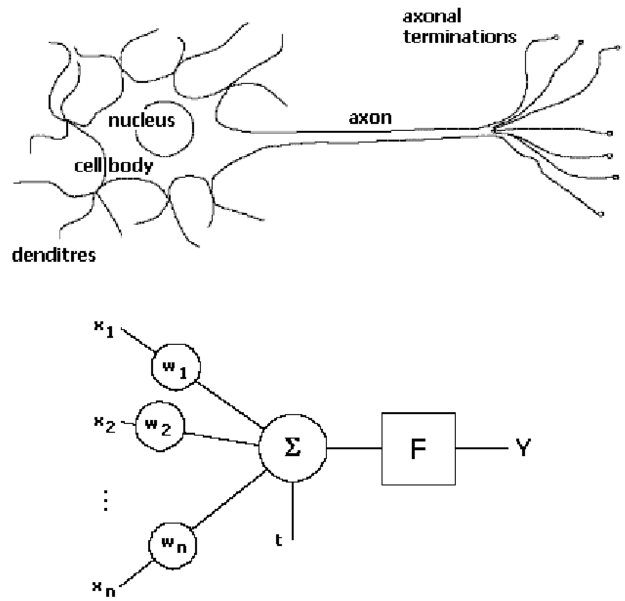


Figure 1. Biological Neuron vs. Computational Neuron. Source: [2].

The artificial neuron can process multiple inputs. Each input is assigned a specific weight. In Figure 1, these weights are $w_1$ through $w_n$. These weights can be altered and adjusted to improve the accuracy of the decision. A larger weight will be applied to an input that has been deemed more significant in the decision-making process. The flexibility and adaptability of the weights can allow a neuron to "learn" or adjust the projected output to identify patterns more accurately when given an unknown input [4].

Each of the inputs are then summed together, then sent to the activation function. The activation function is the threshold of a neuron. It dictates whether the neuron will send an output signal or not. The output of an activation is often either 1 and 0, or 1 and-1. This allows for backpropagation. Backpropagation is how the neural network learns by comparing the current error rate to that of the previous layer [4]. When the activation function is limited to 1 and 0, or 1 and -1, it is much faster and easier to compare the error rates, making backpropagation possible.

### 2. Layers within Networks

A singular neuron can have an output that is broadcast to multiple follow-on neurons. When the output of one neuron is connected to the input of another, they create a web of decision points that can take multiple inputs and have multiple outputs. These larger, interconnecting webs are called neural networks. There are three parts of a NN, the input layer, the hidden layer, and the output layer [2]. A layer refers to a set of neurons, often visualized in a vertical stack, that have similar inputs, but do not feed to each other. The input layer is the set of neurons that take the initial input signals. The hidden layers are the layers in between the input and output layer. The output layer gives the result of the network. Figure 2 shows a feed forward multilayer NN. This network is feed forward because all the connections are going in the forward direction and there are no feedback loops [2].

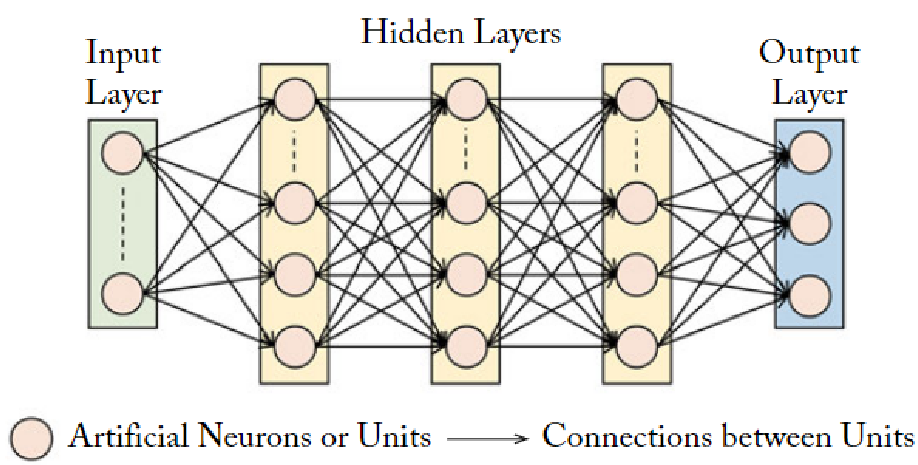


Figure 2. Multi-Layer Feed Forward Neural Network. Source: [2].

The utility of these networks is the variability within each neuron and the ability to identify distinguishing features of an input and make a calculated decision regarding the particular parameter. Each path between neurons and layers represents a characteristic of a signal or dataset being analyzed and categorized to result in an accurate output analysis.

### 3. Training the Network

The decision-making process of the neurons and the network as a whole is based on the weight at each input node and the backpropagation used to adjust the weights; however, in order to determine and categorize attributes of the input, the network must first be trained. There are two types of training methods for neural networks. The first is supervised training. This is accomplished by providing the network with inputs as well as the associated outputs [5]. The network then adjusts the weights within each neuron to recognize specific characteristics and patterns associated with the inputs. This is traditionally used for pattern recognition [2]. The second type of training is unsupervised. Unsupervised training involves feeding the network inputs, but not providing the defined outputs [2]. Instead of identifying specific patterns, it will cluster similar characteristics together and group inputs accordingly.

### 4. Convolutional Neural Networks

One specific type of neural network is the convolutional neural network (CNN). The CNN has filters within the layers that are created during the training phase. When an unknown input is seen, these filters convolve with the input to create a new dataset to be passed to the activation function [6]. It is much easier to think of this convolution as a grid of numbers. The filter would be like a matrix of a specific size and values determined by the learning process. Figure 3 shows this process visually. The filter slides through the input grid, much like a scanning pattern. With each position of the scan, the input is convolved with the filter to produce a separate output.

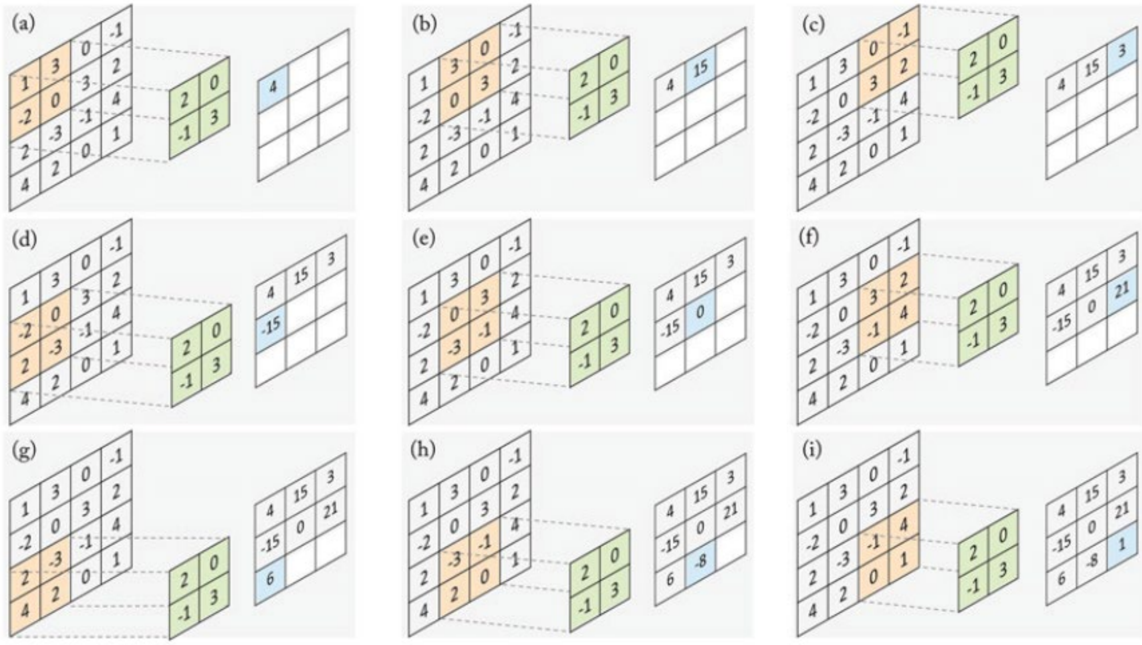


Figure 3. Convolution of Filter Matrix with Input Grid. Source: [2].

Convolutional neural networks can take the original input, normalize with respect to the mean of the input, then scan each part of the input with the filter to pinpoint characteristics [6]. This greatly increases the accuracy of the network and allows for faster classification.

### 5. Generative Adversarial Network

Now that the basic building blocks of neural networks have been established, it is time to look more in depth into the technique used for this thesis experiment. It was first theorized by Ian Goodfellow in 2014 [7] to develop a way for networks to better visualize and categorize images. In more recent years, it has been adapted to various disciplines where classification is the goal.

#### *a. Adversarial nets*

The generative adversarial network is a process that utilizes two networks where each work to fool the other. There is the generator network and the discriminator network. During training, the generator takes the training dataset, recognizes patterns, then develops its own signal to send to the discriminator. The discriminator works as the deciding factor to classify the signal from the generator. It calculates the probability of the input coming from the generator or from the real source. The goal of this two-player game is to pit each network against each other. The job of the generator is to fool the discriminator, and the job of the discriminator is to accurately decide what input is real and what is fake [5].

The game allows for each network to train the other further than human-defined data could. It would take a huge amount of false training data input into the network to improve the accuracy of the NN. With the GAN, it develops its own false data to practice classifying and learns from its own processes. During the training phase, the generator and discriminator play the game until both reach at least 50% accuracy [5]. Once this is accomplished, the networks are considered at peak performance and fully trained, i.e., ready for outside input classification.

## B. RF SIGNALS

Now that the neural networks have been explained, it is time to go in depth into the data set used. Neural networks are a great resource; however, it is a “garbage in, garbage out” situation. If the training data input is not robust enough to accurately depict real-life signals and noise scenarios, then the accuracy and simulated performance will not be accurate when real-world signals are the input. The structure and characteristics of a Radio Frequency wave must be understood to accurately simulate them and create usable training data.

### 1. The Spectrum and Communications

The use of radio waves to transmit data was first established in the 1890s by Guglielmo Marconi [8]. Radio wave utilization has since grown in size, adaptability, and usage. The communications application of radio waves has allowed for the dissemination of large volumes of information over long distances using analog and digital communications.

The frequency spectrum is the backbone of RF communications. There is, at all times, electric and magnetic energy traveling through the air in a wave-like path called propagation. This energy propagation is characterized by the frequency that it travels. The spread of frequency ranges that energy travels in is called the frequency spectrum. Figure 4 shows a visual representation of the spectrum and some common applications utilizing the frequencies.

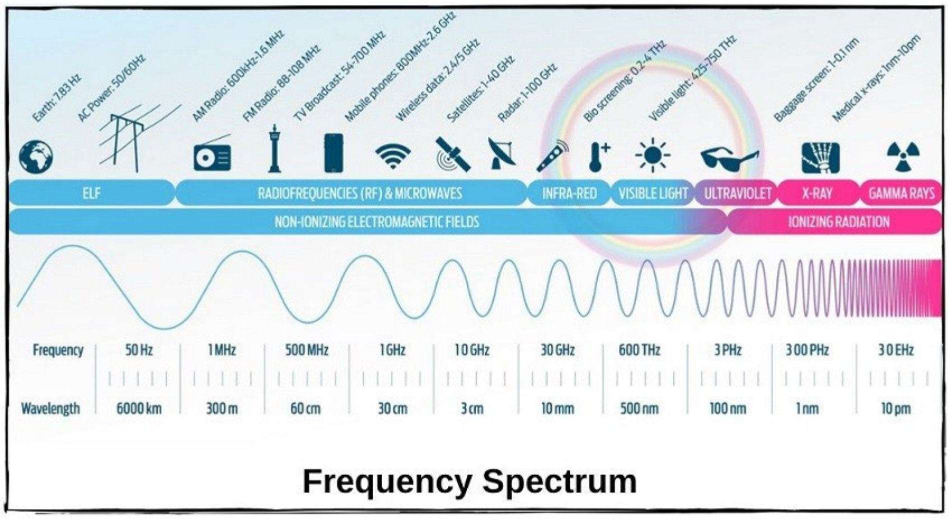


Figure 4. Frequency Spectrum. Source: [6].

When a means of communication is being utilized, it is described by its frequency band. For example, radio waves operate at a frequency between 3 kHz to 100 GHz. Communication engineers can embed communications within the oscillations of radio waves by varying the characteristics of the wave.

Digital communication is the modification of a radio wave to represent a message in the form of ones and zeros. These ones and zeros are the symbols that are then translated to the original message. The greatest example of the first form of digital communications is the telegraph. The telegraph used a simple electric pulse to send a “dot” or “dash.” These dots and dashes represent one letter at a time. A short time period when no signal was present (a short blank) was used to separate the dots from the dashes, and longer blanks were used between letters. Figure 5 shows the representations of dot/dash/blank combinations and what they represent, or the International Morse code.

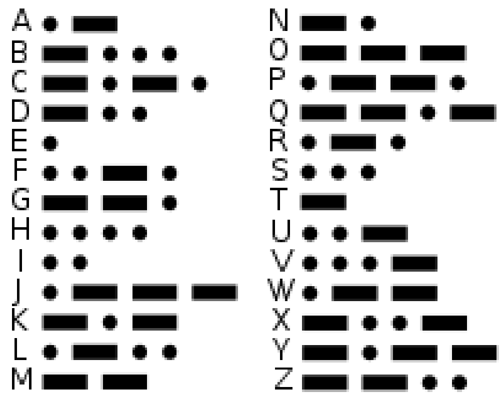


Figure 5. International Morse Code

Computer language is much like Morse code in that it has a dot/blank in the form of ones and zeros. This communication via ones and zeros is digital communication. Analog is the varying wave carrying the digital signal. So how can digital signals be represented on an analog wave?

### 2. Modulation

Communication engineers can represent ones and zeros through modifications of the analog wave. Modifying a particular wave at a particular frequency is called modulation. Modulation is done by altering one of the major characteristics of a wave: frequency, amplitude, or phase. Figure 6 shows how modifying a wave can represent the ones and zeros of digital communications.

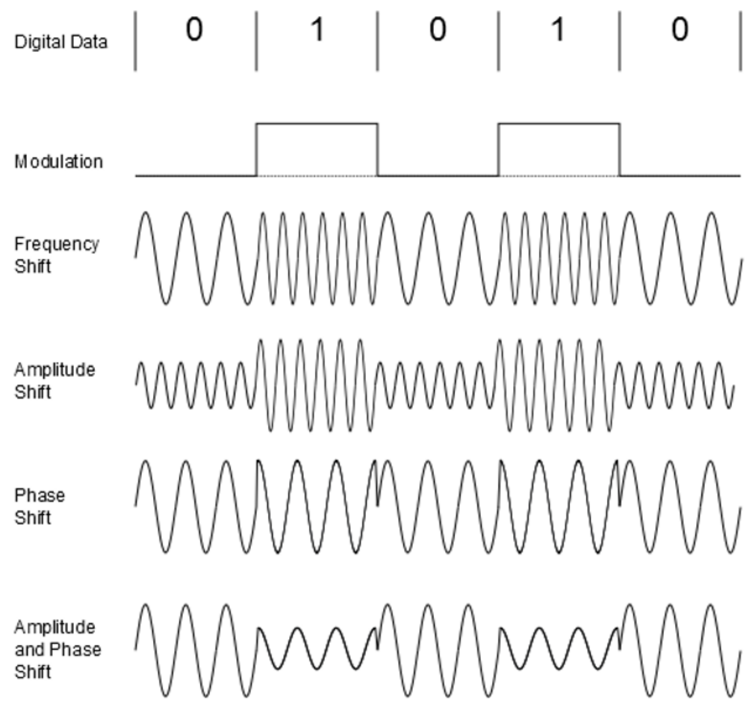

Figure 6. Basic Signal Modulation Techniques. Source: [15].

The form of modulation can be a singular technique, or a combination of techniques. The type of modulation depends on the frequency range of the analog signal used and the environment it is operating in.

### 3. BPSK Modulation

Binary Phase Shift Keying (BPSK) is the process of representing ones and zeros as two phases of a sinusoid 180° apart from each other [9]. For every bit time, the transmitted signal will either remain at the current phase, or switch to the other. The receiver will translate a constant phase as a 0, or an altered phase as a 1. Figure 7 shows how the sinusoidal waveform is affected by each phase change.

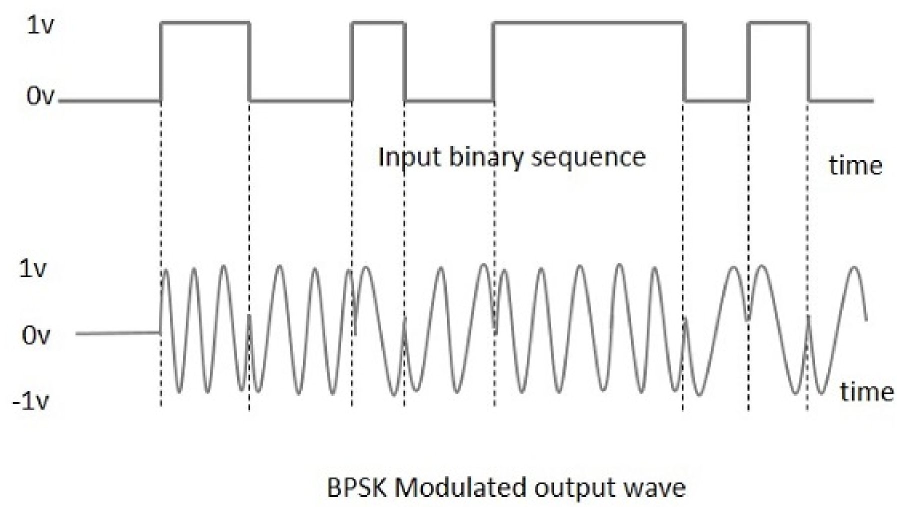


Figure 7. BPSK Output Wave. Source: [10].

### 4. QPSK Modulation

Quadrature Phase Shift Key (QPSK) modulation is done by modifying the phase of the transmitted sinusoidal signal by four different phases. Like BPSK, the receiver will see the different phase changes; however, for QPSK, the phase variations are specific for the pairs of bits they represent. The phase variation for this thesis will be π/4, 3π/4, 5π/4, and 7π/4, but a QPSK signal can have a phase shift of any multiple of π/4. Each phase signifies a set of bits; 00, 01, 10, and 11. Figure 8 shows a simplified version of how the sinusoidal waveform is affected by each phase shift.

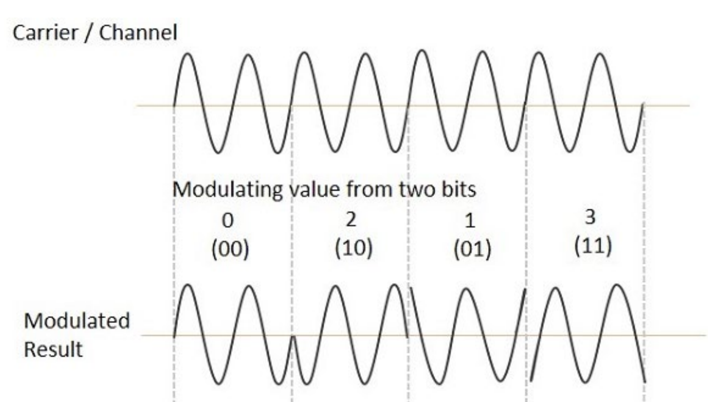


Figure 8. QPSK Modulated Waveform. Source [10].

A symbol is the pulse or change in waveform that represents a set of bits. For QPSK, the symbol is the change in phase, but each symbol represents two bits. Figure 9 shows each phase and its binary representation.

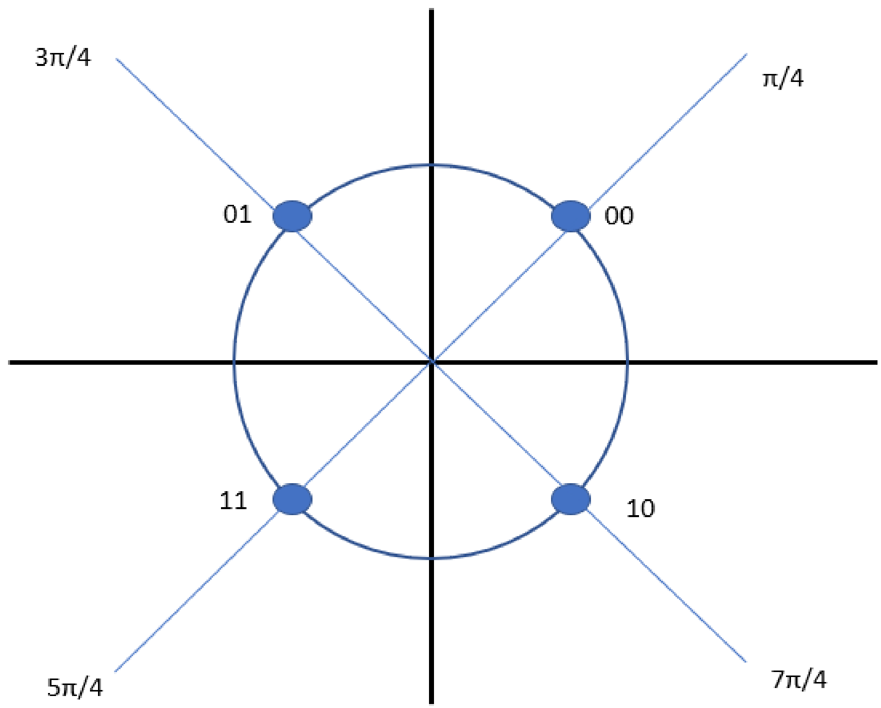


Figure 9. QPSK Scatter Plot

Now that there is an associated phase shift for bits transmitted, those phases need to be represented in a sinusoidal waveform to propagate across the spectrum.

The basic mathematical representation of a waveform is the cosine wave. The characteristics of the cosine wave can vary in its amplitude, frequency, and phase. After considering the characteristics of a QPSK signal, the transmitted QPSK signal can be described using the following equation:

$$s_i(t) = \begin{cases} \sqrt{\dfrac{2E}{T}} \cos\left[2\pi f_c t + (2i-1)\dfrac{\pi}{4}\right], & 0 \le t \le T \\ 0, & elsewhere \end{cases}, \quad (1)$$

where $T$ is the symbol duration, $E$ is the transmitted signal energy per symbol, $f_c$ is the carrier frequency, and $i$ is 1, 2, 3, or 4, representing the four possible phases of a QPSK signal [11]. The combination of symbol duration and symbol time prior to the signal start is a contributor to the amplitude A used later. This is the ideal form of the signal that is seen at the receiver.

### 5. QPSK Demodulation

The extent of this thesis will focus on the signal prior to demodulation, which is referred to as the pre-demodulated signal. Our envisioned system will collect the pre-demodulated (pre-D) signals. The GAN sorts signals based on their representation prior to demodulation in our collection database. However, the demodulation process is useful to understand the structure of a communications signal and how it is processed. The incoming QPSK-modulated signal represented in (1) undergoes demodulation at the receiver. The demodulation process is represented in Figure 10 in the form of a block diagram.

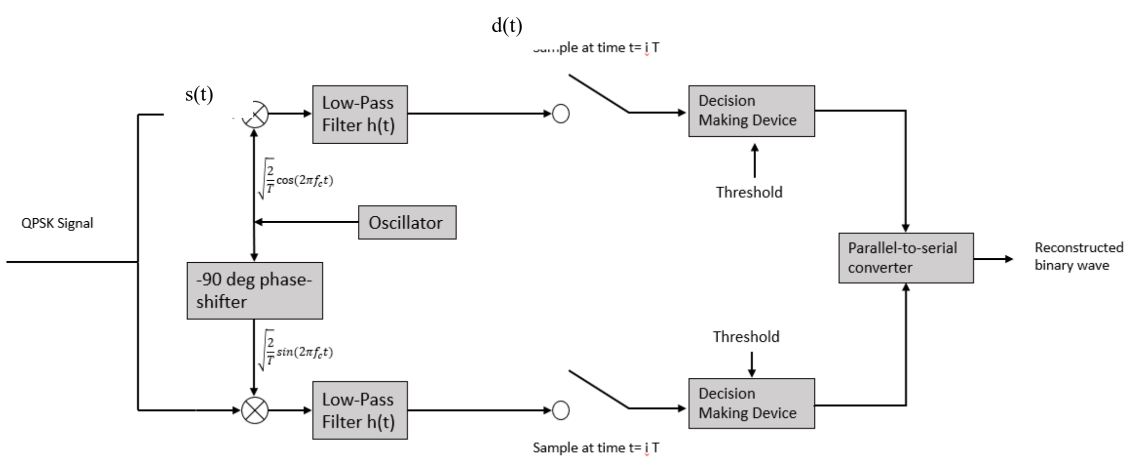


Figure 10. Demodulation for Coherent QPSK Receiver. Source [6].

The QPSK signal comes into the receiver as a complex waveform and is mathematically represented in the form of (1). To demodulate the signal, it is first multiplied by the orthogonal carriers:

$$\phi_1(t) = \sqrt{\frac{2}{T_s}} \cos\left(2\pi f_c t\right), \tag{2}$$

and

$$\phi_2(t) = \sqrt{\frac{2}{T_s}} \sin\left(2\pi \mathrm{f_c t}\right). \tag{3}$$

This separates the incoming signal into the in-phase channel and quadrature channel using the locally generated signal that has the known carrier phase information. When the receiver has the known carrier information, the detector is considered a coherent detector. The oscillator creates the basis function using the carrier information [12]. Multiplying by the carriers yields

$$\begin{aligned} &\textit{In Phase} \quad s_I(t) = A\cos\left(2\pi \mathrm{f_c t}\right)\cos\left(2\pi \mathrm{f_c t} + \theta\right) \\ &\textit{Quadrature Phase} \quad s_q(t) = A\sin\left(2\pi \mathrm{f_c t}\right)\sin\left(2\pi \mathrm{f_c t} + \theta\right) \end{aligned} \tag{4}$$

Where θ is the phase difference between the incoming signal and locally generated signal from the oscillator and A is the amplitude of the incoming signal. Using trigonometric identities, the above equations can be written as [13]

$$s_1(t) = \frac{1}{2}A\left[\cos\left(2\pi f_c t + \theta\right) + \cos(\theta)\right], \tag{5}$$

$$s_q(t) = \frac{1}{2}A\left[\sin\left(2\pi f_c t + \theta\right) + \sin(\theta)\right]. \tag{6}$$

The signals are then sent to the low pass filter or matched filter. The purpose of the matched filter is to maximize the SNR at the sampling instant. The filter can be mathematically represented using either a frequency domain, (7), or time domain, (8) approach as:

$$H(f) = \begin{cases} exp\left(-j2\pi f t_o\right), & -B \le f \le B \\ 0, & |f| > B \end{cases}, \tag{7}$$

$$h(t) = \int_{-B}^{B} exp\left[j2\pi f\left(t - t_o\right)\right]df \quad , \tag{8}$$

where B is the bandwidth of the filter. The incoming signal is filtered to produce the demodulated signals

$$d_I(t) = \frac{1}{2}A_I\cos(\theta) \quad , \tag{9}$$

$$d_q(t) = \frac{1}{2}A_q\sin(\theta) \quad , \tag{10}$$

Where θ is the phase associated with the QPSK transmitted shift of π/4, 3π/4, 5π/4, and 7π/4. The parallel to series converter takes the real and quadrature decisions and compiles them into the demodulated binary signal and the message has been successfully received and demodulated.

## C. SIGNAL VARIATIONS/NON-IDEAL FACTORS

The demodulation process involves many mathematical steps and equipment that are dependent on knowing the exact time and frequency of the transmitted signal. In an ideal setting, the incoming signal would look exactly like the transmitted signal, and the process of demodulation would add no imperfections. The real-world scenario; however, includes imperfections in the equipment, noise, and differences in timing. These imperfections and variables will be referred to as non-ideal factors contributing to variations in the received signal. Non-ideal factors and discrepancies must be incorporated into the signal generation input into the GAN to accurately depict a real-world scenario.

### 1. Vary SNR

The first kind of interference to be analyzed is noise. A communication signal propagates through a medium. In the case of QPSK communication signals, that medium is air; however, the atmosphere is filled with exterior signals and atmospheric variations. These variations can be summed up in the form of noise. Noise is the added signal at the receiver that is not the intended message.

Noise can be modeled as a linear function of time and then simply added to the transmitted signal

$$r(t) = s(t) + n(t)\,. \tag{11}$$

Noise is generally described with respect to the original signal energy. $E_b$ is the energy per bit of the signal. The ratio of bit energy to one-sided noise power spectral density is $E_b/N_o$, also called Signal to Noise Ratio (SNR). When noise power spectral density is higher than the bit energy, the SNR is less than one and the signal fidelity is diminished. When the SNR is high, there is less noise, and a higher accuracy of signal transmission is achieved. The power of a communications signal can be described by simply squaring the signal level; therefore, the average power of the incoming QPSK modulated signal in (1) can be calculated by (12) and (13) [11].

$$\text{if} \quad s(t) = A\cos\left(2\pi f_c t + \theta\right), \text{ adapted from (1)}$$

$$E\left[s^2(t)\right] = \frac{1}{T}\int_0^T \left(A\cos\left(2\pi f_c t + \theta\right)\right)^2 \mathrm{d}t \tag{12}$$

$$= \frac{A^2}{2} \tag{13}$$

(13) gives the power of the signal. The transmitted signal, however, has two bits per symbol, so the energy in a symbol is the power of the signal times the symbol time, $T_{sym}$

$$\textit{Energy in a symbol} = E_s = \frac{A^2}{2} T_{sym} \tag{14}$$

To break it down even further, we need to find the energy per bit to accurately calculate the SNR. Because there are two bits per symbol, we can simply divide the energy in a symbol by 2

$$\textit{Energy in a bit} = E_b = \frac{E_s}{2} = \frac{A^2}{4} T_{sym} \tag{15}$$

Now that we have accurately defined the energy per bit for a QPSK signal, with the above definition of energy per bit, the SNR can be represented as:

$$E_b/N_0 = \frac{A^2 T_{sym}}{4N_0}. \tag{16}$$

### 2. Arbitrary Initial Phase

The incoming signal is multiplied by the signal from the local oscillator set to the original carrier frequency. This allows the phase difference to be calculated with respect to the original phase. In coherent detectors, the phase of the original signal is known by the receiver; however, there is no specific reference phase to initiate a start and stop. This means when the demodulator reads the input signal phase, it does not have an initial zero' phase to reference.

### 3. Arbitrary Start time

For uniform sampling, the collection system samples at a constant sampling rate, $R_{samp} = \frac{1}{T_{samp}}$. The sampling instants, while they follow a constant pattern, are not coordinated with the timing of the signals. If we consider the signal to start at some time $t_{start}$, then the initial sample will be at some time $t_{start} + T_{offset}$ where $T_{offset}$ is a uniform random variable ranging between zero and $T_{samp}$. This is very significant as two identical signals will be represented by different vectors of sample values due to the random offset time. We need to design a GAN that still classifies these different signal vectors as the same signal.

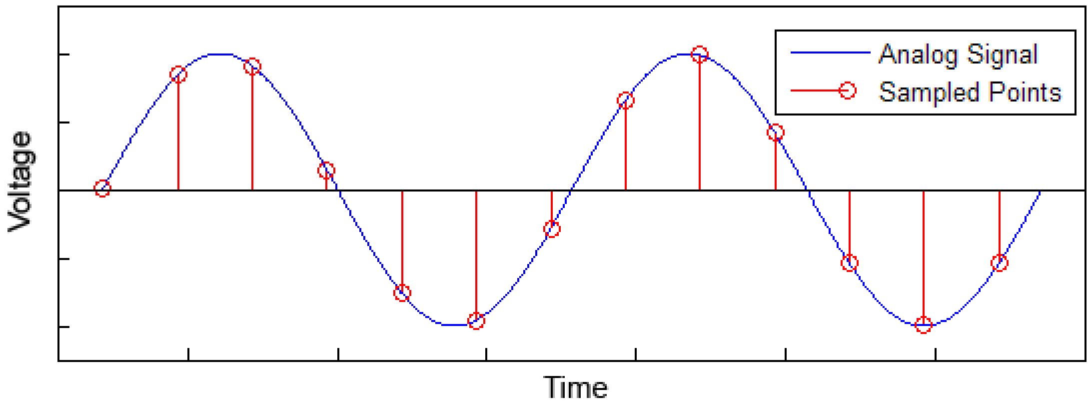


Figure 11. Digital Sampling of Analog Signal. Source [9].

The sampling process has limitations. There are a minimum number of samples that must be received within a time period in order to collect enough samples to combat aliasing. Aliasing is when the sampling does not happen as quickly as the analog signal is changing. The samples do not occur fast enough to accurately recreate the incoming signal, causing major distortion. Figure 12 gives a visual of this occurrence. The blue line is the incoming analog signal, and the red line is the recreated signal from the sampled values.

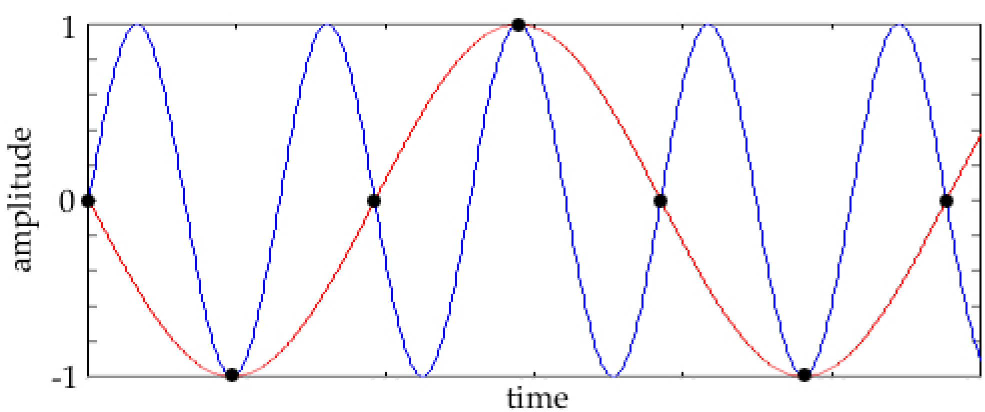


Figure 12. Aliasing of a Signal

To accurately sample the original signal, the sample time needs to be twice the signal maximum frequency. This is called the Nyquist rate. The Nyquist rate is defined with respect to the bandwidth as

$$f_s > 2f_{max} = 2B, \tag{17}$$

where $f_s$ is the sampling frequency, $f_{max}$ is the maximum frequency of the original signal, and B is the signal bandwidth. With a QPSK signal, the signal period corresponds to the symbol period, not the bit period.

# III. DATASET AND EXPERIMENT ARCHITECTURE

## A. SOFTWARE OVERVIEW

### 1. Google Colab/Tensorflow

The neural network in this work was developed using Tensorflow. Tensorflow is a free, open-source program that allowed for the design of the backbone of the neural network [14]. It works as the structure to be able to consolidate data, clear and standardize it, and preprocess for desired analysis. Tensorflow is a library of machine learning algorithms to develop learning programs [14]. Keras is an application programming interface within Tensorflow that allows the user to access even more high-level neural network applications [15].

Google Colab was used in this work to write code utilizing the tools from Tensorflow and Keras [16]. Colab, short for colaboratory, was created by Google to allow any user to write python code and process it through their high-power processing resources. It allows access to thousands of programming toolboxes and is ideal for developing machine learning software [16].

While the GAN was developed in Colab, the QPSK dataset was developed using MATLAB. MATLAB was used because it was the more familiar program, was easiest to develop the QPSK signal, and worked well with Colab.

## B. SYNTHETICALLY GENERATED PRE-D RF DIGITAL SIGNALS

To first develop the transmitted signal, the baseband signal comprised of a series of ones and zeros needed to be developed. In the context of a wireless signals, the baseband signal consists of the preamble, mobile subscriber identity field, and message bit stream. The preamble is the bit stream common to every signal and signifies the start of the transmission. Within our generated dataset, all of the signals have the same preamble bit sequence. The mobile subscriber identity field is specific to which transmitter is transmitting, so it will vary between transmitters. The GAN must learn to use the part of the signal reflecting the mobile subscriber to identify and sort the signals of interest from non-signals of interest. This process is automatic and done with no prior knowledge of the signal structure.

To run experiments on both BPSK and QPSK simulated signals, the bit stream pairs were mapped according to the number of phases for modulation type. Figure 13 shows the scatter plot for the QPSK signal generated without addition of non-ideal factors.

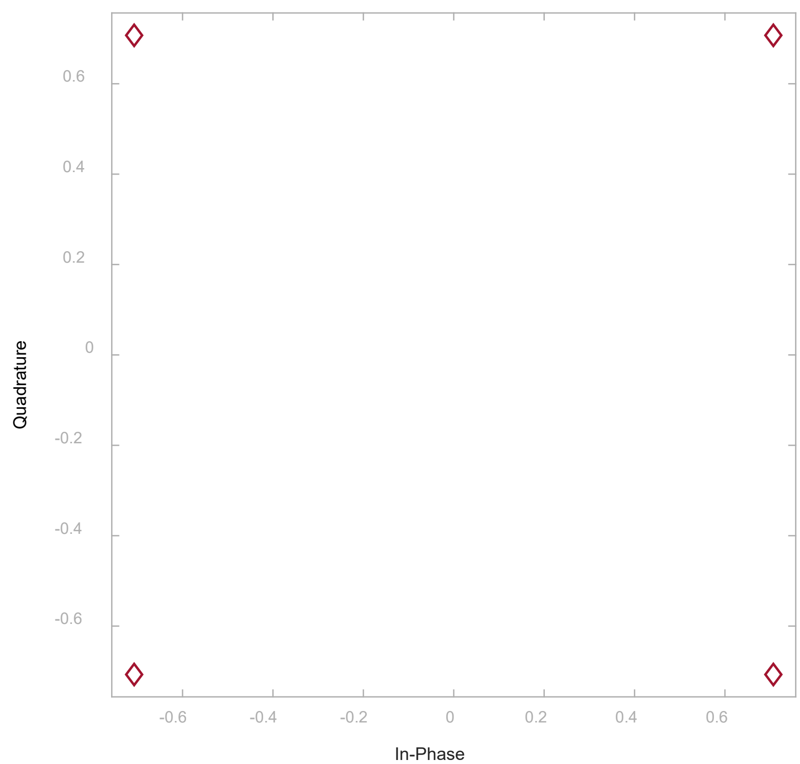


Figure 13. QPSK Bit Scatterplot

### 1. Incorporation of Non-Ideal Factors

The signal dataset was developed to incorporate the above-mentioned non-ideal factors. To accurately develop the real-world dataset, each factor was analyzed and included in specific areas of the signal generation work-flow corresponding with the location affected by real world variations in the received signal.

#### *a. White Gaussian Noise*

The first non-ideal factor considered was the varying SNR. MATLAB's native function to add additive white noise to a signal, AWGN, was unsuitable for the purposes of this thesis due to it lacking the input for the signal power. The function restricts input signal power to 0dB [17], which limits the ability to model SNR variability. To visually see and confirm the effect of the " AWGN " function, Figure 14 shows how a simple bit stream is affected by the added white Gaussian noise.

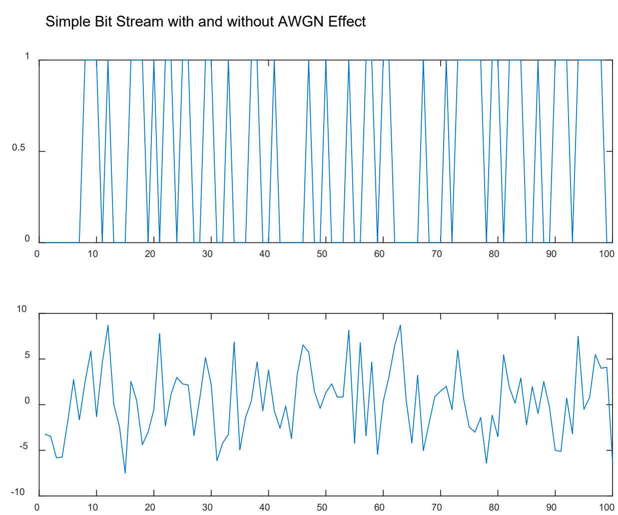


Figure 14. AWGN Effect on Sample Bit Stream

When considering a more accurate representation of an SNR, the signal power must be considered. A dataset was created with varying SNR from -18 to 20dB in 2dB increments.

To vary the SNR, the dB value of the experimental SNR is converted from the logarithmic relative value of a decibel to the absolute value. This means the value of decibel was converted using $10^{\frac{dB}{10}}$, then (16) is used to calculate $N_o$ , noise power. The variance, $\sigma^2$ , is calculated using

$$\sigma^2 = \frac{1}{4} N_o f_{samp}. \tag{18}$$

Then, standard deviation of the noise is calculated

$$\sigma = \sqrt{\sigma^2}. \tag{19}$$

The standard deviation is then used to generate the random values of the noise to be added to the signal.

To fully evaluate the production of noise and test all variations against the GAN, the experiments include a set of runs utilizing the AWGN function, then a set of runs using a more accurate noise generation function.

#### *b. Arbitrary Initial Phase*

When the signal is modulated, the phases are varied to represent the symbols; however, the receiver only sees a series of sinusoidal signals without knowing the initial phase of the signal. The variations in phase are based on an initial reference phase, but that is unknown to the receiver. Due to this, there is a variation within the phases of the sent and received signals. To incorporate this, an initial phase $\phi$ is randomly selected from a uniformly distributed random variable between 0 and $2\pi$. This allowed for even more randomness in the signal development to match the randomness of real-world signal variation. The $\phi$ is expressed as an exponential phase due to Euler's Identity, then multiplied to the initial signal to alter the phase. This can be mathematically explained as

$$\text{Euler's Identity: } e^{j\phi} = \cos\phi + j\sin\phi,$$

$$\text{Law of Exponents: } e^{m} \cdot e^{n} = e^{m+n},$$

$$e^{j2\pi f_c t} \cdot e^{j\phi} = e^{j(2\pi f_c t + \phi)}. \tag{20}$$

#### *c. Arbitrary Initial Start Time*

The start and stop times of the signal are initiated using the preamble; however, much like the initial phase, the timing of the transmitted signal depends on the synchronization of the transmitter and receiver. In real-world communications, this synchronization is imperfect. Because of this, a small delay is added to simulate the random relation between the start time of the signal and the initial sample instant. This is done by using the rand operation to add a small number to the time t uniformly distributed between 0 and $T_{samp}$. This shifts the signal slightly to simulate variation in the initial start sample time.

#### *d. Small Frequency Error*

Transmitters and receivers have imperfect local oscillators, resulting in generated carriers that are slightly different in frequency, even when they are supposed to be the same frequency. Even within a single transmitter or receiver, the local oscillator frequency will vary slowly over a small range. Therefore, two signals carrying identical bits will generally be represented by two distinct signal vectors in our collection. Despite this distinction, our GAN needs to be able to automatically learn that these two distinct signals must be classified as the same. Therefore, the simulated collected signals all include independent small frequency errors selected from a standard normal random variable between 0 and 1, multiplied by one thousandth of the frequency error.

After all the alterations have been implemented, Figure 15 shows the scatterplot of the training data.


Figure 15. Received QPSK Signal with Non-Ideal Factors Implemented

### 2. GAN Architecture

The structure of the GAN used is that previously used by Ellison in his thesis on development for signal classification [3]. The two main segments of the GAN are the discriminator and the generator. The structure of the GAN is made up of two identically structured CNN, each with an input layer, three convolutional layers, and an output layer. The discriminator and the generator each use LeakyReLu activation functions. Figure 16 and Figure 17 are the flow charts for the structure of CNN as detailed in [3].

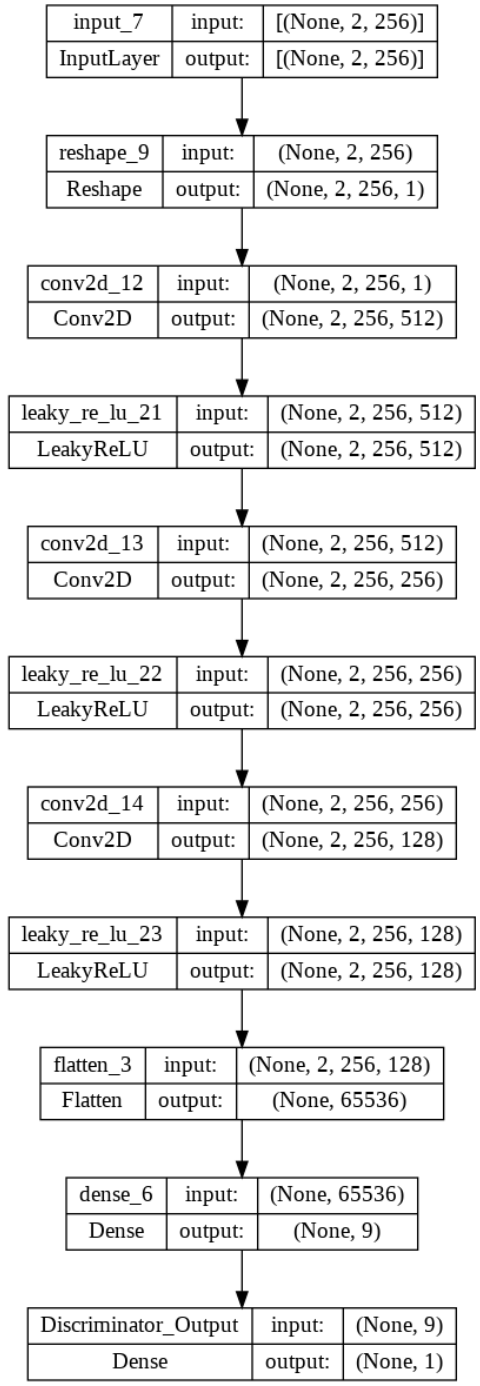


Figure 16. Discriminator Layer Composition

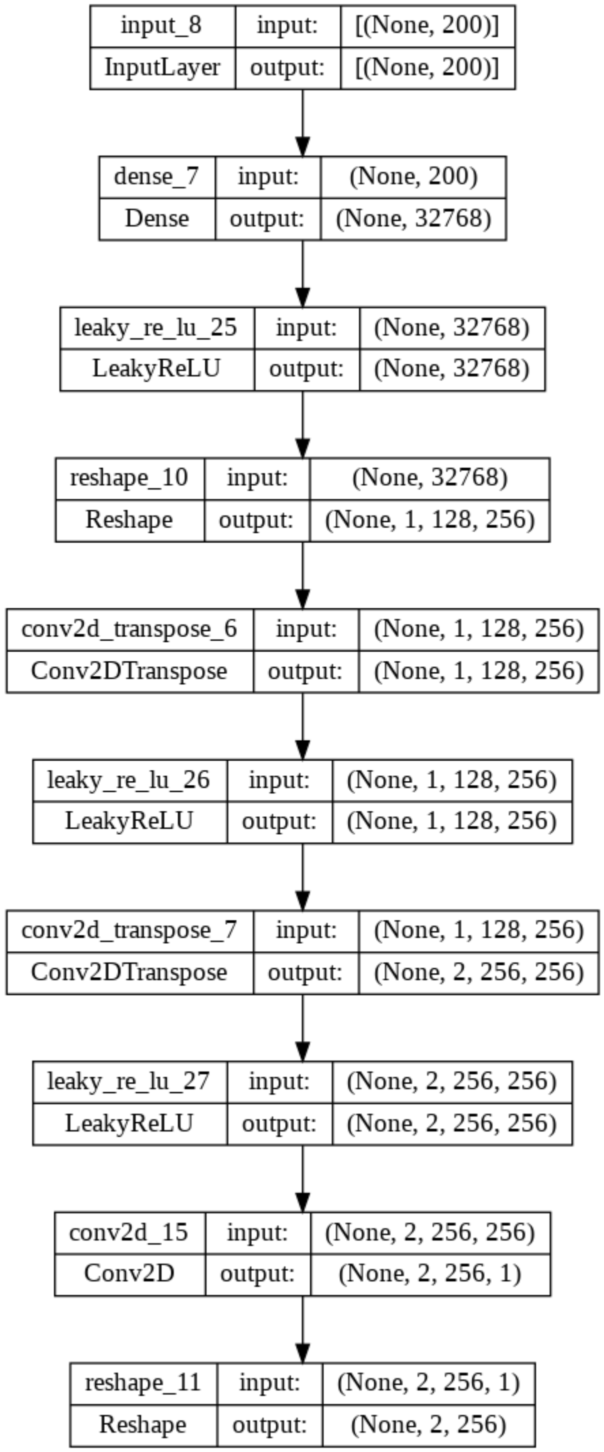


Figure 17. Generator Layer Composition

The environment and conditions of the experiment were those of the optimum classification characteristics of the GAN concluded in [3]. The structure of the GAN was unchanged.

The performance evaluation of the GAN in previous work indicated the size of the dataset to be used. When generating a dataset for a NN, a percentage of it is used for training. This means the NN is given the known signals of interest features and their identity labels to learn the stochastic distributions associated with SOIs and distinguish between the non-SOIs and signals of interest. From [3], the optimum size of the training dataset was reported as 75%, meaning of all the training dataset created, 75% of it was used to train the GAN.

## C. EXPERIMENT SETUP

The dataset generation for this thesis was developed to simulate both BPSK and QPSK signals to be able to accurately recreate the results of previous work and solidify consistency in experiments.

### 1. Dataset Composition

The dataset generation began with nine separate signal sets, each representing one distinct transmitter. There is one SOI set, then eight other transmitted signal sets labeled 1–8. Each signal has the same general data structure, all starting out with the preamble that is 16 bits (8 symbols) for each. Then, each signal set has its own specific mobile subscriber identifier specific to the signal that is 72 bits long (36 symbols). Each signal then has a random message that is a bit stream of 100 bits (50 symbols). There are 94 symbols in each signal. When sampled at eight samples per symbol, each signal is 752 samples long. In summary, all signals share the same 16-bit preamble; all signals in each set share the same 72-bit mobile subscriber identifier, which is distinct from the MSI for any other set; and all signals have a randomly selected data field. So, the GAN will need to learn to sort signals based upon the MSI, but without demodulating the signals and without any information underlying the signal structure.

To create a large enough dataset to train the network, 1000 signals were generated for each set. Therefore, there are 1000 different SOI signals, each having the same preamble and mobile subscriber identifier, but having different message bit streams. For the purposes of this thesis, the 1000 signals of one signal type will be referred to as a batch. The goal of the GAN is to accurately identify the SOIs from the signals in the 8 other batches.

### 2. Dataset Variation

The white Gaussian noise simulated in the dataset varied from -18 dB to 20 dB in increments of 2 dB. A copy of each signal batch was made for each SNR, then the SNR was applied. This means each batch consists of 20 sets of noisy signals, each of the 20 sets has the identical signal set.

Two types of modulation are simulated. As a control group, Dataset 1 consists of the nine signal sets previously discussed, but with BPSK modulation and only added white Gaussian noise. The white Gaussian noise implemented is from the noise generating function developed in this thesis. This allowed for the inclusion of signal power when varying the SNR. Dataset 2 has QPSK signals with only added white Gaussian noise. Dataset 3 has BPSK signals with small frequency error, arbitrary start time, and arbitrary initial phase added along with white Gaussian noise. Dataset 4 has QPSK signals with small frequency error, arbitrary start time, arbitrary initial phase, and white Gaussian noise.

# IV. RESULTS AND ANALYSIS

Each experiment involved running the dataset through the GAN and displaying accuracy. Table 1 shows the parameters associated with each dataset.

Table 1. Dataset 1–4 Characteristics

| | Dataset 1 | Dataset 2 | Dataset 3 | Dataset 4 |
|---|---|---|---|---|
| Signal Type | BPSK | QPSK | BPSK | QPSK |
| Arbitrary Phase Error | OFF | OFF | ON | ON |
| Small Frequency Error | OFF | OFF | ON | ON |
| Timing Offset | OFF | OFF | ON | ON |

Each dataset had a percentage of it used for training, then the rest was used for testing.

## A. DATASET 1

The first dataset was used to train the GAN and then to evaluate the GANs ability to classify the signals. Figure 18 is the percent accuracy of the GAN classification at the number of training iterations.

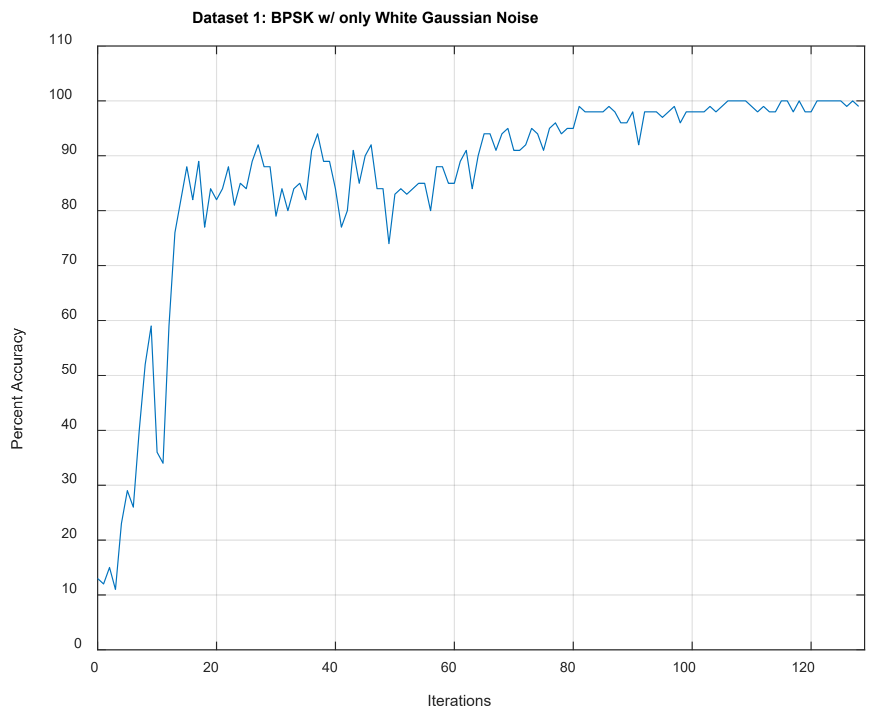


Figure 18. BPSK with only White Gaussian Noise

Due to the time it took to train, the experiments were run until the 129th iteration for continuity and to show the affect the noise, modulation, and non-ideal factors have on the signals. For those that did not reach 100% accuracy by the 129th iteration, the experiment was run again to see how long it would take. For Dataset 1, the GAN reached 100% accuracy by the 108th iteration.

## B. DATASET 2

The second dataset was the QPSK modulation with the White Gaussian Noise only and no other factors. Figure 19 is the percentage accuracy per iteration.

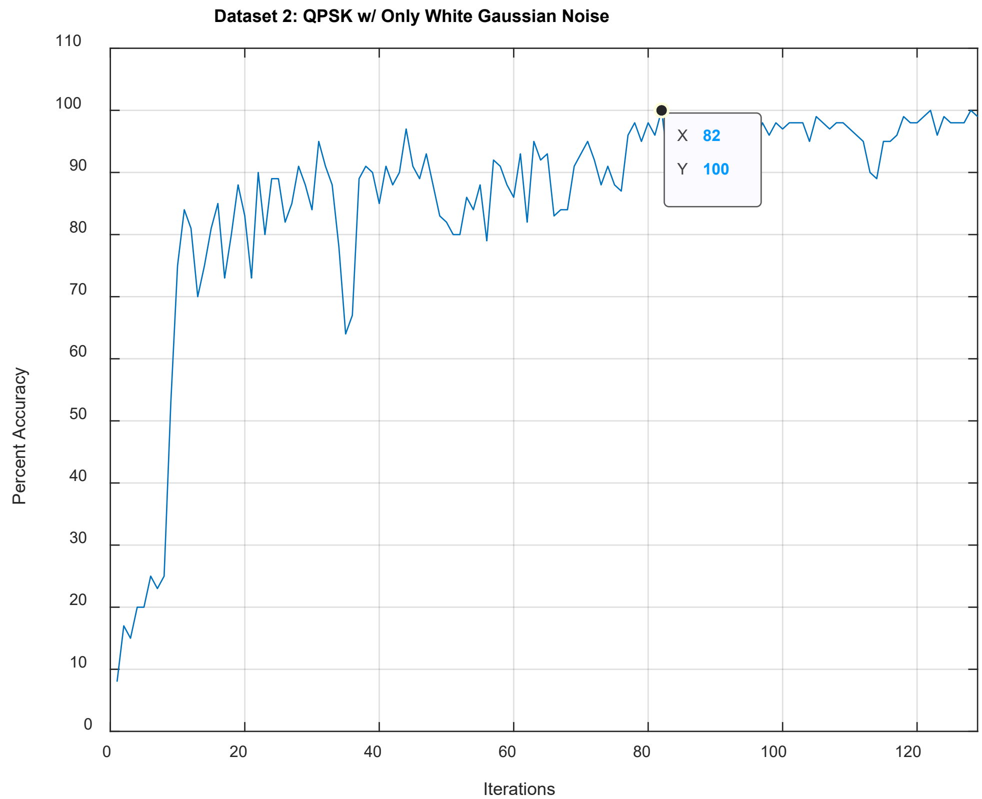


Figure 19. QPSK with Only White Gaussian Noise

The QPSK dataset reached 100% accuracy by the 82nd iteration. The variations in percent accuracies while training were between 5% and 20%, meaning as the GAN was learning, it reached an accuracy, but then did not perform as well in the next iteration and the accuracy decreased.

## C. DATASET 3

The third dataset is the first to run with the collection variations implemented. The dataset was developed with BPSK modulation, white Gaussian noise, arbitrary initial phase, arbitrary start time, and small frequency error. Figure 20 graphs the GAN percent accuracy per iteration. The learning was halted at 160 iterations to save computing time.

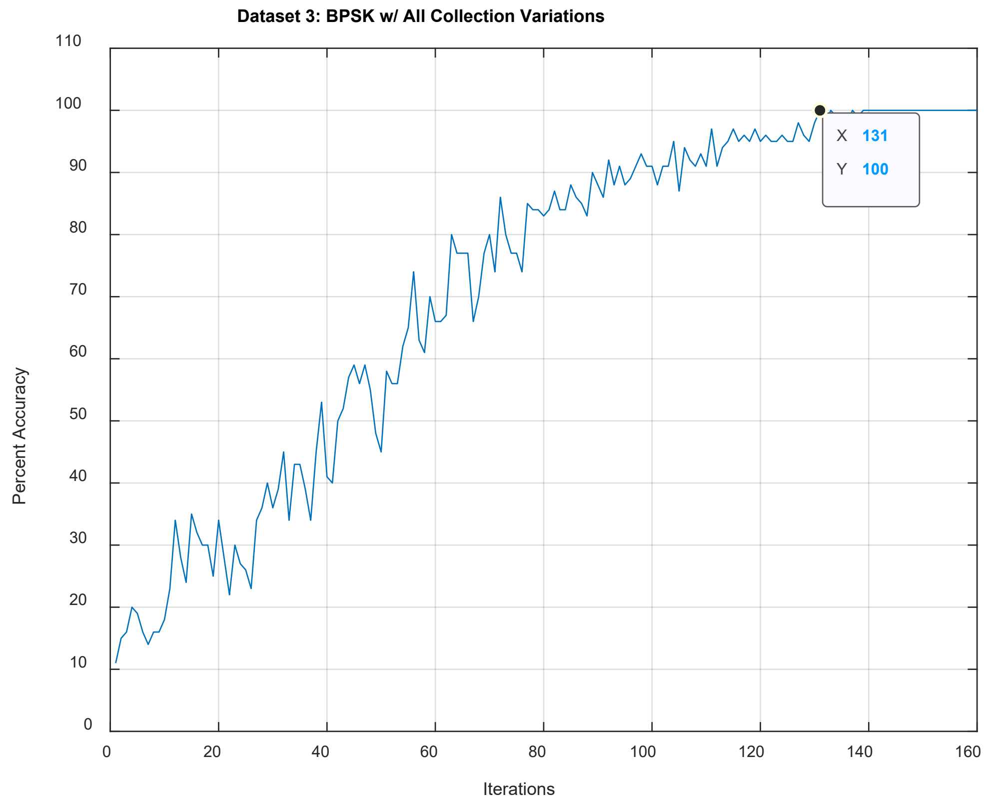


Figure 20. BPSK with White Gaussian Noise and All Collection Variations

The percent accuracies took a slower, more gradual path to reach 100% with no significant drop in accuracies. The GAN did not reach 100% accuracy until the 131st iteration.

## D. DATASET 4

The final dataset is based on QPSK modulation with white Gaussian noise and all non-ideal factors included. Figure 21 graphs the GAN percentage accuracy per iteration with 70 percent of the dataset used for training and 30 percent used for testing.

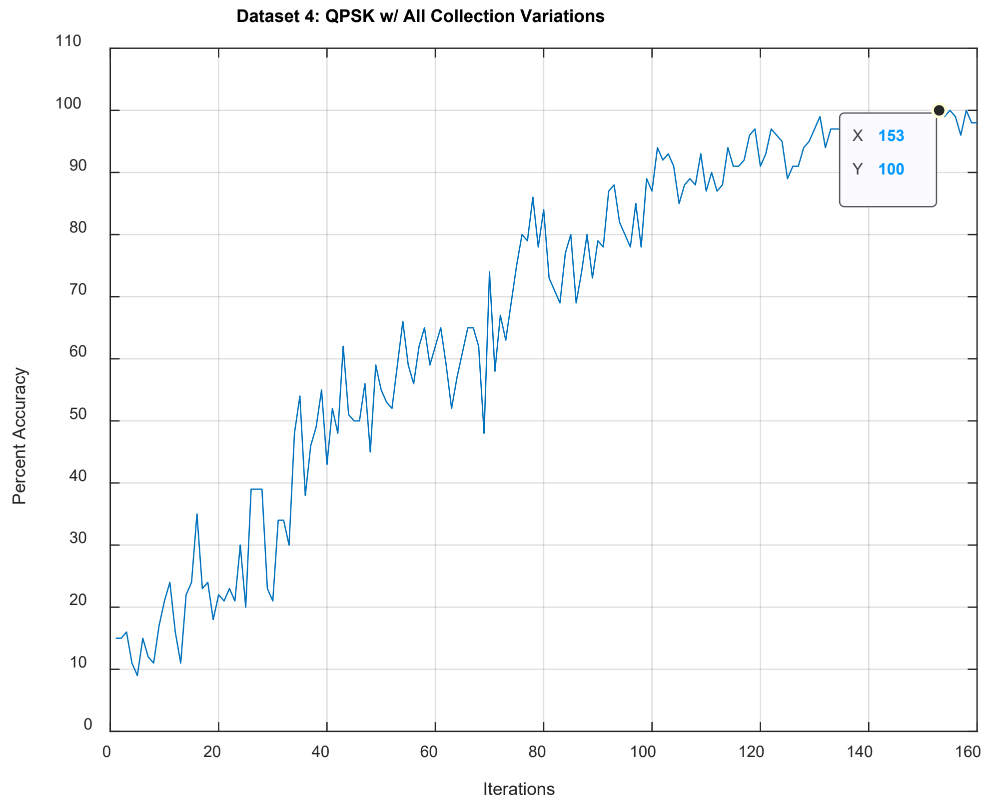


Figure 21. QPSK Signal with All Non-Ideal Factors and White Gaussian Noise

The QPSK results have significant dips in accuracies. While they are at different locations than the dataset with only White Gaussian Noise, the dips are clear and drop more than 20%.

## E. CONSOLIDATED DATASET

All dataset percentage accuracies are graphed together in Figure 22 to compare the performance of the GAN against different modulation types, noise implemented, and the affect real-world non-ideal factors have on the percent accuracy.

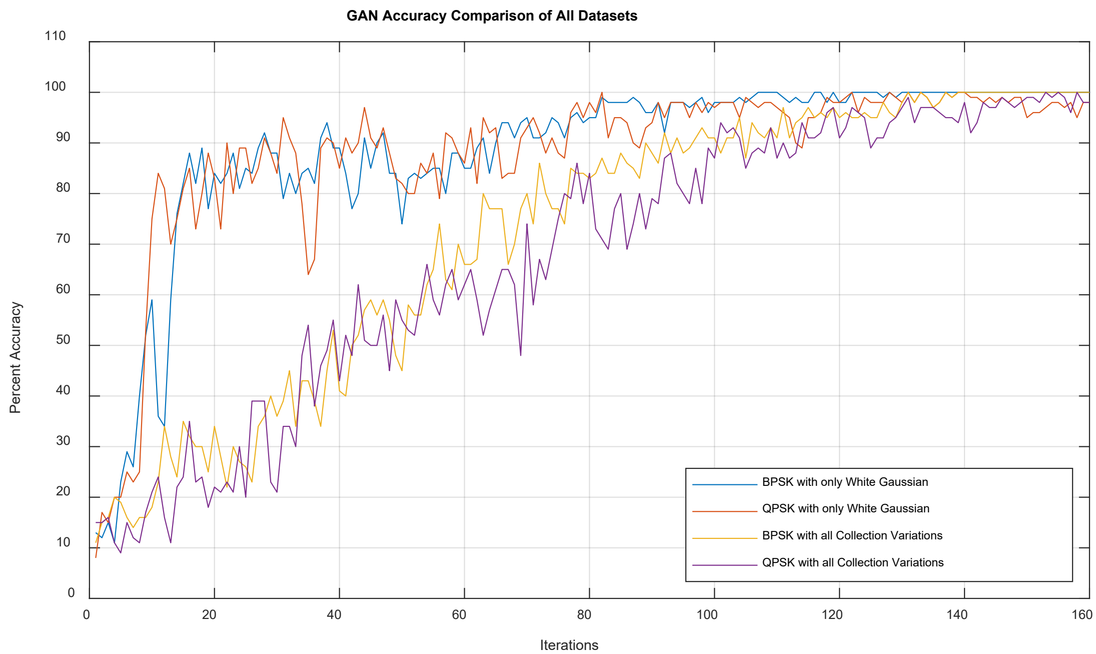


Figure 22. All Datasets Compared

The signals with all the collection variations added took more iterations to reach 100% accuracy; however, the GAN was still able to achieve 100% accuracy of the signals.

## F. ANALYSIS

The GAN was able to classify signals withs 100% accuracy for each dataset. The most rapidly trained case was the case using BPSK signals with only white Gaussian noise. This is to be expected because the complexity of BPSK modulation is less than that of QPSK and therefore creates a simpler stochastic distribution within the dataset, allowing the GAN to more easily distinguish patterns and identify the SOIs.

QPSK modulation had a large effect on the time it took the GAN to reach 100% accuracy. This is to be expected because QPSK has a stochastic distribution that is more complicated than BPSK. It is important to note that the GAN was able to maintain an accuracy of 95% and above after 129 iterations. Recall that the dataset included signals with an SNR of -20 dB. It is plausible that the GAN is classifying signals significantly better than a human analyst.

# V. CONCLUSIONS AND FUTURE WORK

## A. CONCLUSIONS

The feasibility of using a GAN used to classify signals is realistic and attainable. The GAN maintained a high degree of accuracy despite the varying modulation types. While the time to train was longer with the non-ideal factors implemented, it is imperative that these variations be included to simulate real-world collected signals.

The performance of the GAN can be analyzed with respect to how it performed against different modulation types. The accuracy was maintained; however, the speed was greatly affected by the complicated modulation type. As the modulation type becomes more complicated, the GAN maintains its efficacy but requires more training time.

This thesis shows that a GAN can be trained to automatically sort realistic collection signals, identifying nearly all signal types and signals of interest, in spite of highly varying signal to noise ratios and many other unavoidable signal variations induced by the collection system. This would allow the consideration of many more signals than human analysts would be able to sort by hand. This technology is ready for testing against relevant signals collected-off-the-air in an operational environment with relevant collection and computing hardware. If tests are successful, this could be applied to operational use.

## B. FUTURE WORK

While the goal of this thesis was to develop a robust, realistic dataset to prove the feasibility of using a GAN to identify and classify communication signals , this dataset can be utilized by a variety of thesis topics involving communication transmission and analysis. The realistic nature of the dataset could be insightful for BPSK and QPSK modulation, or design and development of hardware where design sensitivity and analysis is needed.

When considering identification and classification of BPSK and QPSK signals, other neural network architectures could be used. There are other types of deep neural networks that account for time varying systems that could be used. For example, recurrent neural networks are often used for natural language processing because of their input capacity. This could show promise in the classification of communications signals where the length of the input signal may vary.

Furthermore, the structure of the GAN in this thesis involved both the generator and discriminator be the same type of neural network. More work and research could be done to explore different structures for the generator and discriminator. Other supervised learning classifiers could be implemented and compared to the benchmarks achieved in this thesis. For example, random forest methods, and their various adaptations, like adaptive and gradient boosting.

# LIST OF REFERENCES

[1] Author(s) J R Runyon, "COMINT (communications intelligence) vs COMSEC (Communications Security) - a vital issue," *COMINT (Communications Intelligence) vs COMSEC (Communications Security) - A Vital Issue | Office of Justice Programs*. [Online]. Available: https://www.ojp.gov/ncjrs/virtual-library/abstracts/comint-communications-intelligence-vs-comsec-communications. [Accessed: 28-Nov-2022].

[2] S. Khan, H. Rahmani, S. A. A. Shah, and M. Bennamoun, "A guide to convolutional neural networks for computer vision," Synthesis Lectureers on Computer Vision, vol 8, no. 1, pp. 1–207, Feb. 2018, doi: 10.220/S00822ED1V01Y201712COV015/.

[3] B. D. Ellison, "Identification and Classification of Signals Using Generative Adversarial Networks," thesis, Naval Postgraduate School, Monterey, 2021.

[4] L. V. Fausett, Fundamentals of neural networks: architectures, algorithms, and applications. Upper Saddle River, Nj: Prentice Hall, 1994.

[5] Y.-J. Cao et al., "Recent Advances of Generative Adversarial Networks in Computer Vision," IEEE access, vol. 7, pp. 14985–15006, 2019, doi: 10.1109/ACCESS.2018.2886814.

[6] G. Boesch, "Deep Neural Network: The 3 popular types (MLP, CNN and RNN)," *viso.ai*, 10-Jul-2022. [Online]. Available: https://viso.ai/deep-learning/deep-neural-network-three-popular-types/. [Accessed: 04-Nov-2022].

[7] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and Harnessing Adversarial Examples," 2014.

[8] "1890s – 1930s: Radio." *Elon University*, https://www.elon.edu/u/imagining/time-capsule/150-years/back-1890-1930/.

[9] B. Sklar, *Digital Communications: Fundamentals and Applications*. Upper Saddle River, NJ: Prentice Hall PTR, 2016.

[10] "Digital Communication - phase shift keying," *Tutorials Point*. [Online]. Available: https://www.tutorialspoint.com/digital_communication/digital_communication_phase_shift_keying.htm#. [Accessed: 17-Nov-2022].

[11] S. Haykin, *An introduction to analog and digital communications*. Chichester: John Wiley & Sons, 2007.

[12] Chandan P. et al. (2012) Implementation of QPSK Based Communication System on TMS320C6713 DSK. World Research Journal of Ad Hoc and Ubiquitous Computing, ISSN: 2320–3382 & E-ISSN: 2320–5660, Volume 1, Issue 1, pp.-11-15.

[13] D. B. Ghate, "Implementation of a digital communication system using QPSK modulation," thesis, Naval Postgraduate School, Monterey, CA, 1995.

[14] "Introduction to tensorflow," *TensorFlow*. [Online]. Available: https://www.tensorflow.org/learn. [Accessed: 03-Dec-2022].

[15] K. Team, "Simple. flexible. powerful.," *Keras*. [Online]. Available: https://keras.io/. [Accessed: 03-Dec-2022].

[16] "Making the Most of your Colab Subscription," *Google colab*. [Online]. Available: https://colab.research.google.com/. [Accessed: 03-Dec-2022].

[17] "awgn," *Add white Gaussian noise to signal - MATLAB*. [Online]. Available: https://www.mathworks.com/help/comm/ref/awgn.html. [Accessed: 17-Nov-2022].

[18] "WLAN frequency bands ⋆ IPCISCO," *IPCisco*, 18-May-2020. [Online]. Available: https://ipcisco.com/lesson/wlan-frequency-bands/. [Accessed: 01-Oct-2022].

[19] P. Frantz, C. J. Ganier, E. Welsh, and A. Valenzuela, *Microcontroller and Embedded Systems Laboratory*. Houston, TX: Rice University, 2006.

## INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California