2023-06

UTILIZATION OF MACHINE LEARNING TO OPTIMIZE RADIO-FREQUENCY INTERFERENCE IDENTIFICATION FOR U.S. NAVAL COMMUNICATIONS

Garnett, Rorey E.

Monterey, CA; Naval Postgraduate School

NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

UTILIZATION OF MACHINE LEARNING TO OPTIMIZE RADIO-FREQUENCY INTERFERENCE IDENTIFICATION FOR U.S. NAVAL COMMUNICATIONS

by

Rorey E. Garnett

June 2023

Thesis Advisor: Wenschel D. Lan
Co-Advisor: Mark Karpenko

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE	*Form Approved OMB No. 0704-0188*

1. AGENCY USE ONLY *(Leave blank)*	**2. REPORT DATE** June 2023	**3. REPORT TYPE AND DATES COVERED** Master's thesis

4. TITLE AND SUBTITLE UTILIZATION OF MACHINE LEARNING TO OPTIMIZE RADIO-FREQUENCY INTERFERENCE IDENTIFICATION FOR U.S. NAVAL COMMUNICATIONS	**5. FUNDING NUMBERS**
6. AUTHOR(S) Rorey E. Garnett	
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)

The proliferation of electronic devices emitting radio waves has led to Radio Frequency (RF) spectrum congestion. This poses a significant threat to Department of Defense (DOD) environments, especially naval communications heavily reliant on satellite systems, which are susceptible to electromagnetic interference. The lack of sufficient interference identification and characterization capabilities further compounds the operational risks faced by naval units. This thesis investigates the utilization of machine learning (ML) techniques for interference detection in RF transmissions. With their advanced data analysis and pattern-recognition capabilities, ML algorithms can enhance interference detection and mitigation. Two architectures, a basic autoencoder and Long Short-Term Memory (LSTM) autoencoder, were evaluated for their ability to identify anomalous RF data within a dataset. The research methodology involved generating RF data with varying Additive White Gaussian Noise (AWGN) levels in a basic transmission pathway. The ML models were trained using normal RF data and evaluated on their ability to detect and classify signals with and without interference. The results demonstrate that both the basic autoencoder and LSTM autoencoder models could effectively identify interference. The LSTM autoencoders achieved a success rate of about 99%, indicating their potential use as a solution to the capabilities gap for interference identification.

14. SUBJECT TERMS artificial intelligence, machine learning, satellite, frequency, optimization	**15. NUMBER OF PAGES** 117
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	**18. SECURITY CLASSIFICATION OF THIS PAGE**	**19. SECURITY CLASSIFICATION OF ABSTRACT**	**20. LIMITATION OF ABSTRACT**
Unclassified	Unclassified	Unclassified	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.

UTILIZATION OF MACHINE LEARNING TO OPTIMIZE RADIO-FREQUENCY INTERFERENCE IDENTIFICATION FOR U.S. NAVAL COMMUNICATIONS

Rorey E. Garnett
Lieutenant, United States Navy
BS, Lander College, 2007

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN SPACE SYSTEMS OPERATIONS

from the

NAVAL POSTGRADUATE SCHOOL
June 2023

Approved by: Wenschel D. Lan
Advisor

Mark Karpenko
Co-Advisor

James H. Newman
Chair, Space Systems Academic Group

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

The proliferation of electronic devices emitting radio waves has led to Radio Frequency (RF) spectrum congestion. This poses a significant threat to Department of Defense (DOD) environments, especially naval communications heavily reliant on satellite systems, which are susceptible to electromagnetic interference. The lack of sufficient interference identification and characterization capabilities further compounds the operational risks faced by naval units. This thesis investigates the utilization of machine learning (ML) techniques for interference detection in RF transmissions. With their advanced data analysis and pattern-recognition capabilities, ML algorithms can enhance interference detection and mitigation. Two architectures, a basic autoencoder and Long Short-Term Memory (LSTM) autoencoder, were evaluated for their ability to identify anomalous RF data within a dataset. The research methodology involved generating RF data with varying Additive White Gaussian Noise (AWGN) levels in a basic transmission pathway. The ML models were trained using normal RF data and evaluated on their ability to detect and classify signals with and without interference. The results demonstrate that both the basic autoencoder and LSTM autoencoder models could effectively identify interference. The LSTM autoencoders achieved a success rate of about 99%, indicating their potential use as a solution to the capabilities gap for interference identification.

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

ADC	analog to digital converter
AGC	Automatic Gain Control
AI	Artificial Intelligence
ANN	Artificial Neural Network
AWGN	Additive White Gaussian Noise
BPSK	Binary Phase Shift Keying
C2	command and control
CCDR	combatant commander
CDF	cumulative distribution function
CJCSM	Chairmen of the Joint Chiefs of Staff manual
COA	course of action
COTS	commercial off-the-shelf
DAC	digital to analog converter
dB	decibels
DNN	Deep Neural Network
DOD	Department of Defense
DVB-S2	Digital Video Broadcasting-Satellite Second Generation
EA	electronic attack
EHF	extremely high frequency
EMCON	electromagnetic condition
EMI	electromagnetic interference
EMS	electromagnetic spectrum
EMSOE	electromagnetic spectrum operating environment
FCC	Federal Communications Commission
GAN	generative adversarial network
GHz	gigahertz
GSM	Global System for Mobile

Hz	hertz
I&Q	In-phase and Quadrature
IoT	Internet of Things
ITU	International Telecommunications Union
JADC2	Joint all Domain command and control
JEMSO	joint electromagnetic spectrum operations
JFC	joint forces command
JFMO	joint frequency management office
JSIR	joint spectrum interference resolution
JSIR-O	joint spectrum interference resolution online
kHz	kilohertz
LNA	Low Noise Amplifier
LO	local oscillator
LPD	low probability of detection
LPI	low probability of intercept
LSTM	Long Short-Term Memory
LTE	Long Term Emissions
MATLAB	Matrix Laboratory
MHz	megahertz
ML	Machine Learning
NN	Neural Network
NPS	Naval Postgraduate School
NTIA	National Telecommunications and Information Administration
PDF	probability density function
QPSK	Quadrature Phase Shift Keying
RF	radio frequency
RFI	radio frequency interference
RNN	Recurrent Neural Network
RTSO	Real Time Spectrum Operation

RX	Receive
SAA	satellite access agreement
SATCOM	satellite communication
SDR	software defined radio
SHF	super high frequency
SIPRNET	SECRET Internet Protocol Network
SNR	signal-to-noise ratio
THD	total harmonic distortion
TX	transmission
UAV	unmanned aerial vehicles
UHF	ultrahigh frequency

THIS PAGE INTENTIONALLY LEFT BLANK

ACKNOWLEDGMENTS

Heartfelt thanks to my wife, Sally, whose unwavering love and support have been the foundation of my academic journey. Your encouragement and belief in me have been my rock. I am forever thankful for you.

Overwhelmingly I am thankful for my son, Landon. Your infectious enthusiasm and creative spirit have inspired me every step of the way. I hope you can be proud of your old dad and understand that if I can do this, you can do anything you set your mind to.

My sincerest appreciation goes to my Advisors, Dr. Wenschel Lan and Dr. Mark Karpenko, whose expertise, guidance, and valuable insights have been instrumental in shaping my journey. Thank you for looking at all the pages of graphs and data, allowing me to take up your lunch breaks, and answering all my stupid questions, occasionally to the point where you were late to the next class. Thank you so much.

Endless gratitude flows to my Mother, whose unfaltering belief in my abilities has fueled my determination to pursue my passions. Your constant encouragement and faith in me have meant the world.

To my Father, I owe a debt of gratitude for always being there to talk and listen, even when it was evident I was trying to procrastinate on writing this thesis.

Every week, my sister, niece and nephews had infectious smiles and joyful laughter on the phone, and they have been a precious gift, providing a delightful escape and offering a much-needed respite from the stresses of writing this thesis.

Amidst my heartfelt thanks, I cannot overlook the SSAG, as the knowledge and experiences gained while at NPS shaped my understanding and appreciation for space.

Monterey, California, will hold a special place in my heart. The memories and friendships forged during my time here will forever be cherished.

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION

The current technological boom in wireless communications has generated a demand for speed, uninterrupted connectivity, and bandwidth. The Radio Frequency (RF) spectrum has become congested due to the proliferation of electronic devices emitting radio frequency waves, including cellphones, smartwatches, wireless audio and video transmitters, and Internet of Things (IoT) sensors. This congestion is further intensified by the Department of Defense's (DOD's) growing utilization of wireless communication systems and unmanned vehicle systems, adding to the complexity of the spectrum environment. With increasing congestion, the likelihood of interference also rises. While interference has traditionally been viewed as a disruption or nuisance in the civilian sector, it is considered a significant threat in DOD environments. Radio frequency interference (RFI) can lead to latency and overall degradation of communication links, jeopardizing critical missions and operations. Recognizing and mitigating these degradations has become a priority, as outlined in the DOD Electromagnetic Superiority Strategy of 2020 [1].

Naval communications, in particular, are susceptible to RFI effects. While terrestrial units have various options for satellite communication (SATCOM) and land-based command and control, naval communications heavily rely on satellite systems. However, these legacy SATCOM systems struggle to keep up with the increasing demand for bandwidth compared to their terrestrial counterparts. Compounding the issue is the limited availability of resources for naval communicators to localize and identify sources of interference. Traditional methods for interference identification involve sophisticated hardware systems, such as spectrum and frequency monitoring equipment, which are not readily accessible throughout the naval fleet. Although PC-based software exists for identifying interference, it is primarily used in the preplanning and allocation stages and is not operationally responsive when units are actively experiencing interference. An emerging solution to this gap in interference identification capabilities lies in the field of artificial intelligence (AI), particularly machine learning (ML). ML techniques have demonstrated success in anomaly detection in industrial equipment and have a proven track

record in handwriting identification, speech recognition, and pattern identification [2], [3], [4]. Leveraging ML methods to extract information from large datasets and gain insights into RF signals can bridge the gap for naval communicators. By employing ML techniques tailored to RF data analysis, naval communicators can enhance their ability to detect and identify interference, thereby improving their communications systems' overall reliability and efficiency.

A. PROBLEM STATEMENT

This thesis aims to identify alternative methods of detecting the presence of interference in RF transmissions that are not currently utilized on naval surface ships. More specifically, this research will leverage various ML algorithms to evaluate their ability to detect anomalous RF data within an RF data set. Current interference detection capabilities exist through expensive hardware systems that are limited in numbers and not readily accessible to all naval surface ships. However, since all naval surface ships utilize the RF spectrum, having the ability to identify degrading effects can limit the adverse effects of these degradations. Current capabilities exist for more high-value units, such as various spectrum and signal monitoring equipment. Additionally, web- and PC-based software programs such as Real Time Spectrum Operations (RTSO) and Spectrum XXI [5] exist to aid operational planners in the deconfliction of the electromagnetic spectrum (EMS) during preplanning phases of operational readiness. These systems and software are limited in operational environments or unavailable in large quantities to be deployed on more than a few units.

Units lacking spectrum monitoring hardware must rely on external sources with the appropriate capability for EMI identification. Without on-site knowledge of interference characteristics or sources, naval communication officers and radio operators depend on simplistic mitigation efforts. These efforts involve verifying system configuration based on Satellite Access Agreements (SAAs), troubleshooting misconfigured or malfunctioning hardware, and visually confirming connectivity and message transmission. However, these methods introduce significant human error into the EMI identification process.

ML can greatly assist human operators in closing their capabilities gap by offering advanced data analysis and pattern recognition techniques. ML algorithms can process large data volumes, identify patterns, and learn from historical examples to detect complex anomalies. Integrating ML capabilities into existing systems enhances processing and analysis methods. ML algorithms extract valuable insights, identifying abnormal patterns or behaviors that indicate potential threats. These algorithms continually adapt and improve by learning from new data, enabling units to proactively detect emerging anomalies.

Therefore, this research aims to explore the unsupervised autoencoder and Long Short-Term Memory (LSTM) autoencoder ML models as feasible methods for RF interference identification as an available option to close this capabilities gap. As a proof of concept, a simple transmission pathway will transmit a text message, utilizing Software Defined Radios (SDR). Intentional interference in the form of Additive White Gaussian Noise (AWGN) will be placed in the transmission pathway. These ML networks will then be trained, and a resulting model will analyze the transmission's In-phase and Quadrature (I&Q) data and detect any anomalous signals. ML models will be evaluated to deduce effectiveness in making correct identification determinations.

Due to the increased congestion in cellular and wireless communications in the civilian sector, several applications of machine learning to detect and classify RFI were reviewed for their applicability to this problem set. For example, Henarejos et al. [6] designed an RF interference detector based on deep neural network (DNN) autoencoder architectures. The researchers focused on interference detection and then interference classification. Interference detection utilized the raw I&Q data of RF transmissions and statistical analysis to identify anomalous signals. This research demonstrated that the I&Q data could be leveraged as significant information from which to identify and distinguish signals. Researchers coupled a Digital Video Broadcasting-Satellite 2 (DVB-S2) signal with various cellular signals to create the desired interference classes. Utilizing LSTM neural networks, researchers successfully classified the DVB-S2 signals by the cellular standard it was coupled with, regardless of bandwidth or frequency position. Having a baseline knowledge of a DVB-S2 signal without interference, the research demonstrated

how LSTM ML models could classify signals based on deviations from the known baseline signal parameters.

Roy [7] demonstrated the ability of recurrent neural networks (RNN) to distinguish between "trusted" and "counterfeited" RF transmitters by analyzing the details contained within the I&Q data. This research focused on adversarial learning-based approaches, particularly generative adversarial nets (GAN), which allow adversaries to gain knowledge of the feature space and potentially mislead any employed ML processes. Furthermore, it was found that identifiable details are present within the I&Q samples of RF transmissions that enable the identification of individual transmitters from each other. These details can be extracted as features and utilized as training data for neural networks (NN) or treated as raw data points for autoencoders. Coupled with statistical analysis, these techniques offer valuable information for interference identification, allowing for improved detection and mitigation strategies.

Hall et al. [8] conducted research on the viability of using synthetically generated RF signal data sets as training resources for ML models. They successfully demonstrated the creation of an RF signal database, emphasizing the importance of collecting identifying characteristics, such as modulation type, to accurately emulate specific operating environments. Their research focused on various modulation schemes, including Quadrature Phase Shift Keying (QPSK), Binary Phase Shift Keying (BPSK), and Long-Term-Evolution (LTE) emissions subjected to out-of-band emissions. This work highlighted the potential of using synthetic RF signals for ML training purposes, as it offers a practical means of generating training data. ML fields such as image classification, speech recognition, and handwriting and object recognition already utilize large datasets, making synthetic RF signals a valuable addition to the available training resources.

This research aims to achieve similar results previously demonstrated but with a focus on developing deployable models in an operational environment with low-cost equipment. Some significant variances exist between the research discussed and this thesis. ML will be conducted in Matrix Laboratory (MATLAB) instead of Python and conducted natively versus utilizing the transferred learning method. Transfer learning is a machine learning technique that utilizes knowledge gained from one task to enhance the

performance of a related task, reducing the need for extensive training data and accelerating the learning process. By leveraging pre-trained models and their learned features, transfer learning can improve generalization and efficiency when utilizing machine learning [9]. All generated RF data sets in this thesis will be local, i.e., from a single task and not obtained from outside sources. Evaluation metrics will focus on binary classification, detecting whether interference is present or not, vice classifying interference based on type. Additionally, the interference type used will be AWGN in varying levels instead of signals of varying frequency or origin.

B. RESEARCH QUESTIONS

The primary question this research seeks to answer is, can U.S. Navy Satellite electromagnetic spectrum interference (EMI), identification, and mitigation efforts be improved by leveraging ML algorithms hosted locally? Additionally, can trained ML models be trained, developed, and deployed utilizing low-cost commercial off-the-shelf (COTS) equipment to offer a solution for the current interference identification gap that exists?

Secondary research objectives will focus on identifying what details can be extracted from signals identified as anomalous and determining if they provide increased interference reporting information that ultimately increases the understanding of the electromagnetic spectrum operating environment (EMSOE).

C. RESEARCH DESIGN

The scope of this research will focus on the ability of ML algorithms to detect anomalous signals generated in a simplified RF communication system in a lab setting. To represent the communication link between a naval surface ship and a SATCOM asset, two SDRs programmed for QPSK modulation will be used to transmit and receive a message of known length and size. The focus will be on determining which ML model successfully identifies an anomalous signal from a representative "normal" signal. The interference type will be AWGN at various signal-to-noise ratios (SNR). A signal with an SNR of 20 decibels (dB) will represent a normal signal. The autoencoder and LSTM autoencoder architectures will be evaluated in their ability to correctly distinguish signals with varying

SNR from the representative normal signal. The signal data set will have SNR ratios from 5 dB to 20 dB and be characteristic of varying magnitudes of degradation. The evaluation metric will include correctly determining a signal with and without interference. A final metric will include determining how well either of the trained models can identify interference.

1. Limitations/Assumptions

To scope the overall research problem appropriately, several assumptions will be made in this research. The representative communication pathway has no equipment faults, and additional typical factors contributing to degraded satellite communications have been eliminated. These external factors are frequency synchronization, polarization, and atmospheric weather. The primary criterion for interference identification relies solely on the successful transmission of a message. No additional methods for interference identification will be used. Contributing limitations to this research include already established data sets of RF transmissions that are not available, which led to the use of locally generated RF data in a controlled environment. This thesis is a proof of concept for ML model training and deployment in a low-cost, minimally computationally intensive setting. This enables the potential application to be easily reproduced by an operational unit at the tactical edge.

2. Thesis Outline

This thesis is organized to first outline how electromagnetic spectrum management within the DOD occurs. Focusing on RF users' responsibilities when utilizing their allotted portions of the RF spectrum, relevant DOD policies will be identified, as well as how these policies and responsibilities enable Naval operational units to contribute to the electromagnetic spectrum superiority Strategy. Emphasis will be placed on the Joint Spectrum Interference Reporting (JSIR) process. The discussion will center on how the details required by the report can be difficult for units without interference-detecting capabilities to provide. There are specific areas where this thesis research can improve the overall fidelity of JSIR reports, providing increased detail and valuable information to

those collecting the reports. These improvements will ultimately benefit the DOD when characterizing the EMSOE.

Chapter II provides an overview of the fields of AI and how they relate to the ML methods used previously for data analysis and anomaly detection. Focusing on two models, autoencoders and LSTM autoencoders, details of how the mechanics of these models work and can be beneficial in RF data analysis. Then an analysis of an LSTM autoencoder model will be discussed, and how this model can leverage the benefits of both models to be useful for RF interference detection. This analysis lays the foundation for how ML will be leveraged to gain insights from RF data with the goal of contributing to the increased fidelity reporting of the JISR process.

Chapter III will identify what hardware and software were utilized to generate and what functions and processes were used to analyze the data. Chapter IV will describe the process of how the test data was generated, and pre-processed, and describe the development of the LSTM autoencoder and autoencoder models. Chapter V will present the results of the research and discuss any information discovered while conducting the research. Chapter VI will discuss conclusions and potential future work.

THIS PAGE INTENTIONALLY LEFT BLANK

II. BACKGROUND

The following sections will serve as references for relevant points related to this thesis research. First, this section will review EMI and how it affects RF communications. Then a review of how the Joint forces manage and allocate EMS-allotted frequencies. This background section will then focus on how the Navy component manages the EMS and what capabilities exist to help naval communicators maintain command and control (C2) capabilities for commanders focusing on what shortfalls exist in the USN fleet. Finally, select types of artificial intelligence and ML methods that have demonstrated uses in RF-focused operations will be discussed.

A. ELECTROMAGNETIC INTERFERENCE

EMI is "any electromagnetic disturbance that interrupts, obstructs, or otherwise degrades or limits the effective performance of electronics or electro-optical equipment" [10]. EMI can be intentional or unintentional and can be self-induced or the result of adversarial actions. EMI can occur from natural sources such as atmospheric effects and space weather. Additionally, EMI can be natural or manmade and originate in all domains, land, air, and sea. Some examples of EMI are co-channel interference, adjacent channel interference, or signal attenuation and reflection. Co-channel interference is what signals from different sources overlap, leading to data corruption and reduced signal quality. The presence of stronger signals can cause weaker signals to become drowned out and lose connection. Adjacent channel interference occurs when frequency bands that are close to each other bleed over to other channels and cause interference. This can reduce data rates and transmissions. Signal attenuation is the result of a signal losing strength over the length of transmission and can result from a signal traversing through material or over long distances. Contributing to signal attenuation is the reflection or blockage of a signal off materials or objects. Because interference can originate from an abundance of sources, EMI is considered a persistent and recurring threat, set to increase as the operating environment becomes more congested with EM-emitting devices. EMI resolution ensures the rapid delivery and exchange of information during military

operations. Because types of EMI vary widely, the focus will be on EMI to SATCOM Systems.

B. RADIO FREQUENCY COMMUNICATIONS IN THE DEPARTMENT OF DEFENSE

1. Electromagnetic Spectrum

Electromagnetic radiant energy can be grouped into bands based on wavelengths or frequency. The entire range of these bands make up the EMS. NASA describes in [11] that gamma waves, microwaves, visible light, and radio waves are bands with similar properties and characteristics. These characteristics are measured in Hertz (Hz). Radio waves can be broken down into frequency bands ranging from 9 kilohertz (kHz) to 300 gigahertz (GHz) and are used in various communications and data transfer systems. When referring to specific bands of the EMS, various international organizations employ different naming conventions. Figure 1 illustrates distinct naming conventions utilized by specific international organizations to represent their respective bands within the EMS.

2. Federal Spectrum Allocation

The International Telecommunication Union (ITU) regulates the allocation and use of various frequency bands. The Federal Communications Commission (FCC) and the National Telecommunications and Information Administration (NTIA) divide the responsibility of regulating spectrum allocation within the United States. The FCC allocates spectrum portions for non-federal use, which include commercial and state government uses. The FCC website lists the NTIA as the responsible authority for spectrum allocation to federal agencies [12].

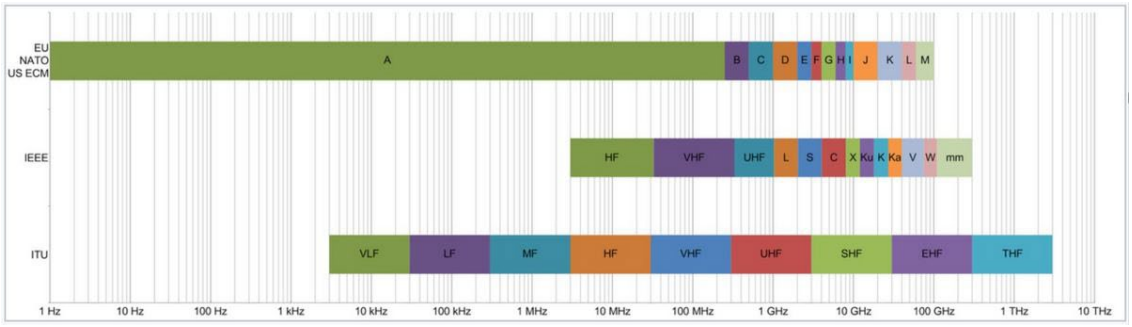


Figure 1. Frequency Bands and the International Organization Associated. Source: [13].

3. Current Military Applications of the Spectrum

The NTIA allocates a portion of the EMS to the DOD. Many systems in the DOD utilize the allocated frequencies within their operating environment. This operating environment is condensed because it is a shared area occupied by government agencies, national governments, and civilian companies that all utilize the RF spectrum. These uses include a multitude of systems; terrestrial-based communication systems, radar systems, submarines, surface vessels, unmanned aerial vehicles (UAVs), and satellites. Some DOD systems operate within the same bands, and in some cases, non-DOD systems, such as commercial satellites or state government, are allocated frequencies. As an increased number of systems emitting EM transmission increase, so do the chances of co-channel interference, blockage, reflection, or adjacent channel interference. This congestion can lead to latency, data loss, signal quality degradation, and reduced coverage areas.

C. SPECTRUM OPERATIONS

Commanders within a geographic region perform various aspects of spectrum operations. Spectrum operations are actions achieved by managing or applying various aspects of the EMS. They range from establishing C2 structures to navigation and signature management.

1. Command and Control

C2 "encompasses the exercise of authority and direction by a commander over assigned and attached forces to accomplish the mission" [14]. Leaders manage available resources to ensure that orders and information gets promulgated and that friendly or enemy forces are identified. These resources can range from SATCOM assets, C2 aircraft such as the E8C Joint Surveillance Target Attack Radar System, to portable radios such as the PRC-111 [15]. Degradation in any of the RF systems available to a commander decreases their ability to maintain C2, resulting in increased vulnerability to adversaries. RF systems compete for limited resources in contested environments; as a result, interference and degradation are common. As technology advances and the number of systems emitting RF signals increases, the operating environment will likely become more congested.

2. Signature Management

Signature management is an important area of spectrum operations, as reducing an observable electromagnetic signature can aid commanders from giving away positions or information regarding assets. Various techniques and processes are applied to signature management, such as emission conditions (EMCON) set aboard naval vessels or changing properties of transmitted signals to reduce the probability of interception or detection (LPI/ LPD) [15].

D. DOD SPECTRUM STRATEGIES AND POLICIES

The DOD released the electromagnetic spectrum superiority strategy in 2020. The goal was to align EMS operations with the National Defense Strategy and National Security Strategies to maintain the Dods' EMS superiority [1]. Within the EMS superiority strategy, objectives were established to guide developments within the DOD and ensure that "Freedom of Action in the Electromagnetic Spectrum is achieved at the time and place of its choosing" [1]. Developing technologies and improvements in the process aim to achieve Joint All-Domain Command and Control (JADC2). The strategy placed heavy importance on developing technologies, methods, and techniques that enable robust electromagnetic battle management capabilities. Identifying that the future battlefield will be rapidly

changing and involves many systems that utilize the RF spectrum makes being able to characterize, adapt, and change of central importance [1].

E. JOINT ELECTROMAGNETIC SPECTRUM MANAGEMENT OPERATIONS IN THE ELECTROMAGNETIC OPERATIONAL ENVIRONMENT

Because the EMSOE is expected to get more congested, there is a need to ensure that the military has unimpeded access to the EMS so that it is available when needed [16]. This need led to the creation of Joint Electromagnetic Spectrum Management Operations (JEMSO). In May 2020, the Joint Staff released a doctrine with respect to JEMSO in which Joint Publication 3-85 was the consolidation of electromagnetic warfare and electromagnetic spectrum management operations under the same umbrella [17]. This consolidation emphasized the importance of the electromagnetic spectrum and solidified its importance in all aspects of military operations. JP 3-85 establishes command-specific policy and guidance for JEMSO.

Any associated activity needed to control and manage the EMS in joint or multinational military operations is the responsibility of the JEMSO, which has two primary goals: ensure EMS-dependent systems can operate in the desired environment and utilize available resources as efficiently as possible. Successful utilization of the EMS is accomplished through the planning and allocating of available resources. Limited capabilities exist in resolving EMS conflicts when deployed, so there is a greater focus in the planning process to mitigate and minimize potential conflicts [14].

1. DOD Management of the Electromagnetic Spectrum

JEMSO are the overarching activities related to the management of the EMS within the Joint Forces. The Combatant Commander (CCDR) performs the specific duties and responsibilities to accomplish JEMSO by establishing specific guidelines for EMS use and a joint frequency management office (JFMO). The JFMO delegates frequency assignments and identifies, analyzes, and evaluates potential spectrum use conflicts and electromagnetic interference. In addition, the resolution of EMI is a central responsibility. This is accomplished through the analysis and collection of reports that detail EMI occurrences in

a geographic area of operations. The overall management of the EMS within the DOD occurs through the Joint Task Force spectrum management life cycle. This life cycle guides joint spectrum managers by detailing steps on information gathering and the development of a spectrum management plan. Spectrum management plans are devised through deconfliction and interference resolution of systems by balancing RF needs with what frequencies are available. Overall, there are 12 steps in the JTF spectrum management life cycle [16]. The life cycle encompasses the complete process of providing spectrum management support to the Joint Forces Command (JFC). This research focuses on the last two areas: resolution and reporting of interference.

2. Navy Communications Component

While the Joint Services Component manages the overarching EMS for the services within an operational geographic area, the Navy has its own management office. Being able to manage the EMS is vital to naval operations. Navy Telecommunication Publication 6F [18] describes how a significant factor in managing the EMS is the timely and accurate identification, reporting, and resolution of EMI. EMI reporting is accomplished through the Joint Spectrum Interference Resolution-Online (JISR-O). This reporting, including both non-hostile and hostile EMI incidents, will pass through each localized command chain of command. As the report makes its way through the chain of command workflow, it is reviewed, and resolution recommendations are made.

3. Joint Spectrum Interference Resolution

JISR is considered the last resort for EMI deconfliction. Reporting procedures are carried out under the Chairmen of the Joint Chiefs of Staff manual (CJCSM) 3320.0E [10]. This last resort is mandatory. It is a centrally managed operation with decentralized execution. When completing interference reporting procedures, the focal point for requesting resolution support is with the JFMO. They will provide guidance for resolving any EMI that cannot be locally accomplished. There are three steps in the JSIR process:

- identification, verification, characterization, and reporting of EMI events;
- geolocation, analysis, course of action (COA) development, and recommendations for corrective actions; and,
- implementation, notification to users, and final closure reporting.

The overall aim of the JSIR process is to understand any EMI events and resolution steps taken. As a result, improving the situational awareness of the EMSOE. The JSIR process is the reporting procedure manual for all of DOD, focusing on the SATCOM EMI. Within the JSIR process, satellite systems include SATCOM, global positioning, and other space systems. EMI can affect SATCOM systems either in the uplink, downlink, or within the crosslinks of systems. Uplink and downlink EMI are reported in the JSIRO portal and help analyze and identify hostile electronic attacks (EA). The JSIR process provides the procedures and methods to resolve EMI at the lowest level within an operational unit's command, as well as detailing specific steps to inform a superior in the command organization if needed. Informing relevant agencies or responsible superior units is aimed at assisting operators in mitigation efforts. Identifying any impact of the DOD's use of EMS will enable decision-makers to have the ability and knowledge on how to effectively communicate or operate EMS systems in a denial environment.

a. Limitations with current expectations

The JSIR procedures are intended to be simple, with the goal of providing sufficient information to aid outside agencies in mitigation efforts. However, there are details of the report that may be difficult for an operating unit to provide. The JSIR content includes information about the affected system, characteristics of the EMI, as well as resolution efforts taken. Since information about the affected system, such as its operating frequency, bandwidth, and polarization, is expected to be readily accessible, this thesis does not focus on improvements in this reporting area. When characterizing the EMI, the reporting operator is first expected to identify if the EMI is a result of solar weather, atmospheric conditions, equipment blockage, or equipment misconfiguration or determine if another unit is the source of the EMI. If EMI is determined to not be the result of any of these

issues, then operators are left to suspect hostile EA. There exist significant gaps in this reporting technique.

Reporting operators are expected to identify characteristic details about the experienced EMI. The steps included in EMI characterization and resolution at the local level can be found in CJCSM 3320.02E [17] and span over 37 steps in 12 categories. Characterizing EMI includes identifying modulation schemes, frequency variances, as well as determining if the EMI is continuous, intermittent, random, or varies. While these characteristics help outside agencies determine the source and mitigation effort of EMI, the ability of the individual unit to determine any of these characteristics with fidelity is limited. As discussed previously, some tools and capabilities, such as RTSO, exist within the Naval surface community; however, these tools and trained operators are not readily available for all units. This capability gap poses a problem as all units are susceptible to EMI and are still expected to report EMI experienced. Without access to the available resources for identifying and reporting EMI, units must rely on outside sources to provide the details needed to complete EMI reporting. This reliance requires communication between the two, and in the case of degraded SATCOM capability, it may not be possible. This thesis seeks to identify a low-cost process and solution to provide the elements of this missing information.

Individual units can enhance their resolution procedures by understanding the features and characteristics of experienced electromagnetic interference (EMI). This understanding facilitates the development of more refined and specific resolution procedures. For example, if there is no structural blockage or changing of filtering methods needed, then changing frequencies just avoids the problem of proper identification and resolution. However, if we can identify the features of suspected EA, over all EMS planning will benefit.

This thesis seeks to find available methods and opportunities to enable individual warfighters to increase their ability to identify and report EMI. Leveraging ML algorithms is a possible solution. ML algorithms are data-centric and can leverage formulas and mathematical properties to identify small features in the numerical properties of RF data. A technique that utilizes ML with RF is known as RF transmitter fingerprinting [19]. Even

individual transmitters utilizing similar frequencies and modulation methods have distinguishable features of their individually transmitted signals. These patterns and the analysis of these unique patterns can help operators identify sources of EMI.

F. ARTIFICIAL INTELLIGENCE

AI is the general ability of computer systems to mimic or emulate human thought processes or actions. It is broad and encompasses a variety of techniques. One of the subsets of AI is ML which focuses on pattern recognition and decision-making based on data sets and related scenarios [20]. Because the range of artificial intelligence-type operations is vast, this section will focus on a discrete path of more refined learning techniques. The learning techniques that are the focus of this section are subsets of Artificial Neural Networks (ANN): LSTM and autoencoders. ANNs are a subset of a larger ML group, which itself is a subset of overall AI.

1. Machine Learning

ML utilizes various mathematical formulas known as algorithms to identify patterns within data to translate and transform these patterns into decision-making processes. ML is different from AI because ML algorithms can improve and learn from the data sets without being specifically programmed to do so. This algorithmic application of pattern recognition is known as "learning." ML learning is categorized into supervised and unsupervised, as seen in Figure 2. An ML model has three major components: data sets, features, and the learning algorithm. The data sets are the entire set of representative information, such as images, text, and signals. Features are the specific characteristics of the data set that can be labeled for learning. Learning algorithms are the mathematical operations that are utilized to execute the learning steps of ML processes [21]. Applying ML algorithms to data sets helps make better decisions and predictions when patterns may not be obvious [22]. As a result, ML can offer efficient alternatives to traditional engineering processes when time and costs are a concern, as described by Simeone in [23]. When ML algorithms are applied to data sets, these data sets are referred to as input. The data resulting from this application is called the output. However, it's important to note

that in addition to the input data set, ML algorithms can also produce outputs in the form of predictions, classifications, or other derived information.

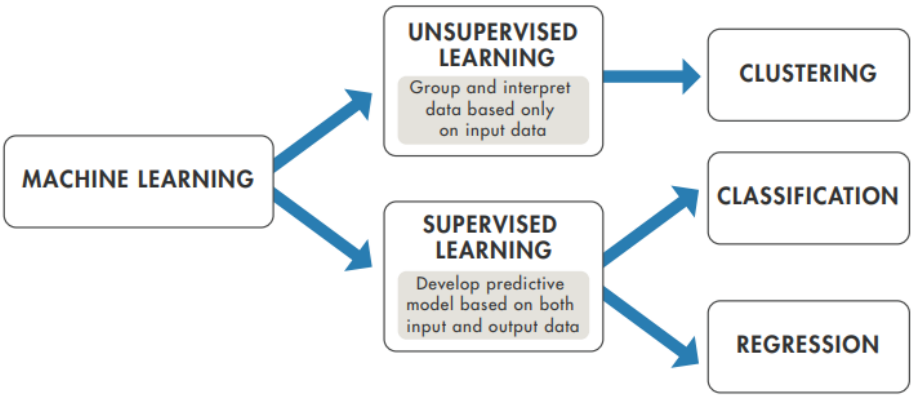


Figure 2. ML Techniques. Source: [22].

2. ML Techniques

a. Supervised learning,

Supervised learning is a learning type of ML where data with known information or labels are used to train a model. The results of applying a supervised learning model to a data set are observed, and predictions are made based on the response. The idea behind supervised learning is that the algorithms are guided along or supervised with correctly labeled inputs, and learning is continuous until a desired level of accuracy is achieved. Within supervised learning techniques, there are classification techniques and regression techniques. Classification techniques are when the output data has a discrete result, such as a decision, whereas regression techniques have output data that is continuous. Supervised learning techniques are recommended when predictions from known inputs are desired, such as echocardiogram interpretation [24].

b. Unsupervised Learning

Unsupervised learning is the other learning type of ML that involves identifying patterns within input data sets that are not labeled, previously identified, or that may be hidden. Unsupervised learning uses clustering, which groups similar output responses

together based on some measure of similarity or shared characteristic. Common applications of unsupervised learning are in image recognition, where models are trained to identify features within images and used to identify them in separate data sets. Unsupervised learning techniques are recommended when a user wants to determine a distinguishing pattern within a group of data and use this pattern to group output data into distinguishable representations of the data.

3. ML Model Construction

a. Artificial Neural Networks

A subset of machine learning is ANN. There are a variety of neural network types and functions, but the basic construct of a NN involves an input layer, a hidden layer, and an output layer, as seen in Figure 3. The input layer of an ANN processes the incoming data through mathematical operations, resulting in an output. This processing takes place in a series of interconnected perceptrons, and the outputs of these perceptrons are then passed to the subsequent layer. Baheti describes in [25] how the construct of a NN seeks to mimic the interconnectivity of the human brain. Utilizing interconnected nodes or neurons in a layered structure. Nodes and neurons within a NN serve as building blocks for learning and modeling intricate non-linear relationships between input and output data [26]. There are terms that identify specific parts of the neural network. Weight, bias, nodes, and features are all terms that identify a function, part, or process associated with an NN.

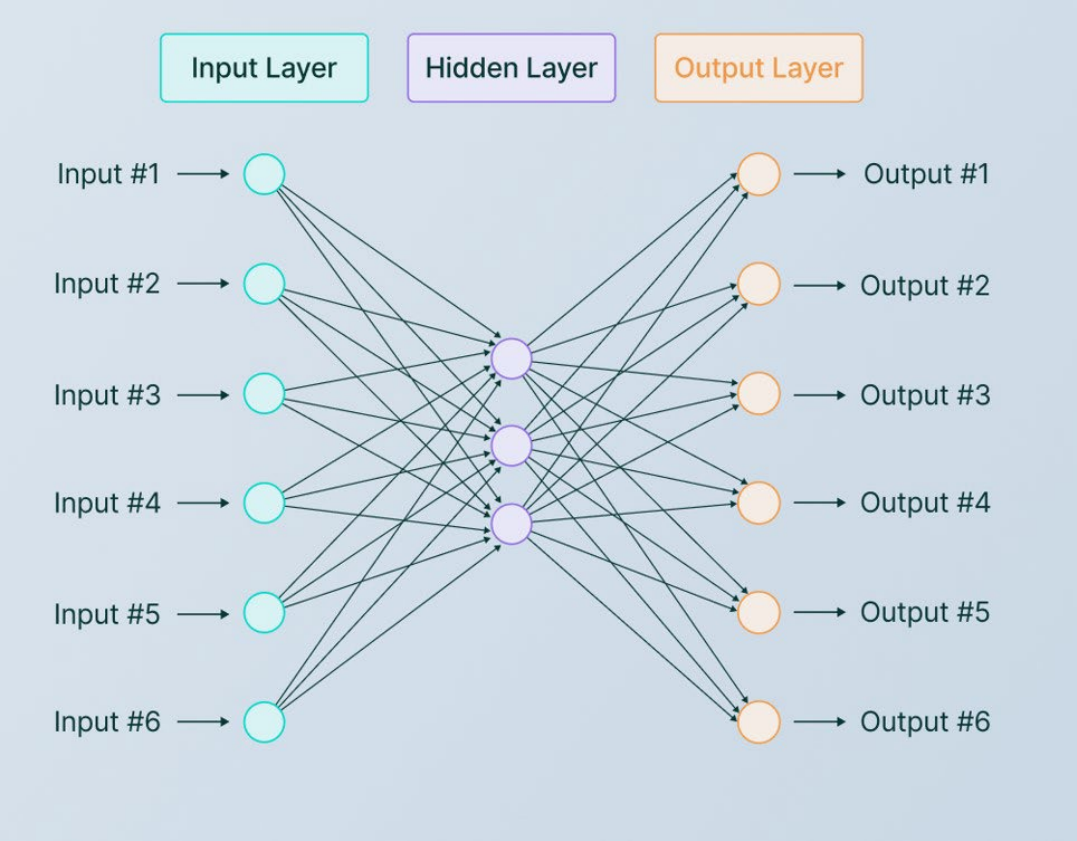


Figure 3. Neural Network Layers. Source: [25].

(1) Input Layer

The input layer takes the data being fed into the model, assigns it an individual node, *x*, and passes the complete information to the hidden layer.

(2) Hidden Layers

The hidden layer takes output data from the input layer nodes, processes it, and sends it to the next layer. As features are fed into the input layer, the weight is determined by attributing importance to features that help the model distinguish the data. Dasaradh [27] describes a simple mathematical formula of what happens at each input layer. At each input, a data value, x_i, is multiplied by a weight value, w_i. Weights are a value that represents how strong an influence a particular input has on the neurons' output. There may be more than one hidden layer in sequence, depending on the complexity of the

network. Hidden layers do the leg work of the computations and extract what features exist in the data. The first step is to take the input values x_i and multiply them by the weight, w_i, associated with the input. The sum of these products is collected.

$$z = \sum(x_i w_i) \quad 1$$

After the sum of the product of the inputs and weights, a bias b is added to the summation and serves to act as a threshold for the activation function, resulting in z [27].

$$z = (x_i * w_i) + b \quad 2$$

After the weighted sum is computed and bias is associated, z is then passed through the activation function, which is how the hidden layer introduces nonlinearity and determines the output of the neuron. The activation function, determined by the task of your ML model, takes the weighted sum with bias and produces an output [27].

$$y = \sigma(z) \quad 3$$

(3) Output Layer

Output data from the hidden layers, y, enter the output layer and are assigned a value. These values make a binary determination or assign the data to a classification group. For binary determination types, yes or no are the output nodes. In classification models, the output nodes correspond to the number of classifications in the data set. The construct of the output layer depends on how the model was built.

b. Autoencoders

Autoencoders are a type of unsupervised neural network that does not require labeled data at the output layer for training. Instead, an autoencoder learns to reproduce the input at its output. The training process of an autoencoder focuses on optimizing the cost function, which measures the discrepancy between the output and the system's input. The objective is to minimize this difference until the output replicates the input accurately. As seen in Figure 4, an autoencoder consists of two main components: an encoder and a decoder. The encoder compresses the input data into a lower-dimensional representation,

capturing its essential features. The decoder then reconstructs the input data from this compressed representation. In some applications, the encoder and decoder portions of the network can be separated and implemented as two distinct systems. By learning to reconstruct the input accurately, autoencoders enable dimensionality reduction and feature extraction in an unsupervised manner

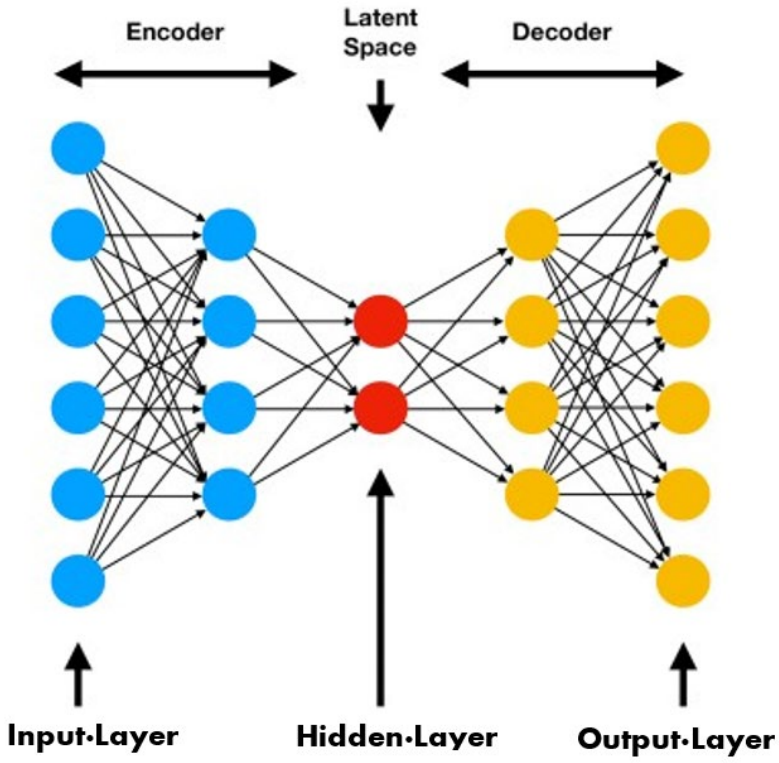


Figure 4. Basic Autoencoder with Associated Layers. Adapted from [28].

(1) Construction of Encoder

Zavrak and Iskefiyeli derive an encoder formula in [29]. At each node, signified by each blue dot in Figure 4, the following function occurs:

$$h = \sigma(w_{xh}x + b_{xh}) \quad 4$$

where σ is the non-linear transformation function, bias is represented by b, and weight is represented by W. x is the representative input data vector that is fed into the initial layer of the autoencoder. The transformation of the input data is h and becomes the hidden representation of that data.

(2) Construction of Decoder

In the decoder, the transformation function is applied to the hidden representations, resulting in an output variable, z. This is described by Zavrak and Iskefiyeli in [29].

$$z = \sigma(w_{hx}h + b_{hx}) \quad 5$$

(3) Reconstruction Error

The metric used to determine the effectiveness of the autoencoder in recreating the input values is the mean squared error (MSE) [30], r, where the output values are subtracted from the input values and averaged over the length of the data set

$$r = MSE = \frac{1}{n}\sum_{i=1}^{n}(x_i - z_i)^2 \quad 6$$

c. Deep Neural Networks

Deep neural networks (DNN) are NN with three or more hidden layers associated with the model [26].

d. Long Short-Term Memory

LSTM is an RNN that specializes in retaining information for long periods. LSTMs are made up of memory cells that contain gates that regulate what information is remembered over time [31]. Each of these gates interacts with an LSTM cell that determines if information about a data set is remembered or forgotten and then what is passed to the next cell. In addition, there are input gates, forget gates, and output gates. The flow of these gates through the cells of the LSTM pathway is illustrated in Figure 5. LSTMs are useful for longer sequences or data with a temporal relationship between their elements. This makes it useful in RF signal analysis.

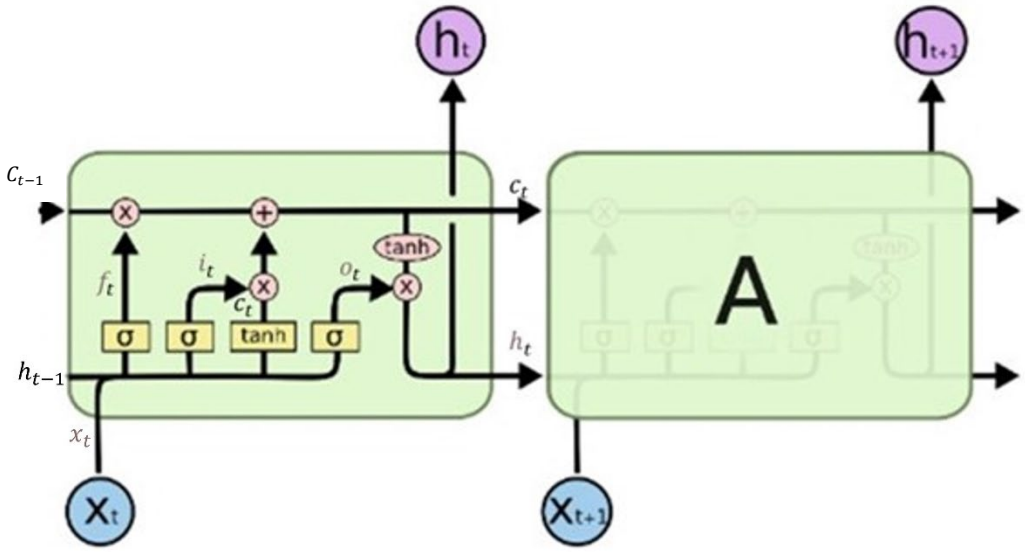


Figure 5. Construction of Long Short-Term Memory Gates. Adapted from [31].

(1) Forget Gate

Forget gates control whether the information is thrown away or forgotten from the cell state. If the information is forgotten, the memory cell is reset to zero, and no information is added to the cell state. The formula for the forget is denoted by

$$f_t = \sigma(W_i * [h_{t-1}, x_t] + b_f) \quad 7$$

where x_t is the input of the cell, h_{t-1} is the hidden information from a previous state. W_t signifies the weight of the current state. Bias is denoted by b_t [31].

(2) Input Gate

Input gates perform two functions: first, they determine whether the cell is updated with new information, then they update the cell state with the information. Olah [31] describes how the input gate determines what values will be used to update the cell.

$$i_t = \sigma(W_i * [h_{t-1}, x_t] + b_f) \quad 8$$

The cell is updated with i_t in addition to the candidate values $\overline{C_t}$ Candidate values denote values derived from applying a $tanh$ activation function and relate to the amount of new information to be stored in the cell.

$$\overline{C_t} = tanh(W_C * [h_{t-1}, x_t] + b_C \quad 9$$

(3) Output Gate

The output gate filters the cell state to select the most important elements for generating the input to the next cell. Using a sigmoid function, the gate creates a vector that is passed through the hyperbolic tangent function to scale the values and only include the most relevant information [31].

$$o_t = \sigma(W_o * [h_{t-1}, x_t] + b_o \quad 10$$

$$h_t = o_t * tanh\,(C_t) \quad 11$$

e. LSTM Autoencoders

The architecture of an LSTM autoencoder model comprises a series of LSTM cells grouped into encoder and decoder layers. The encoder layer compresses the input data into a lower-dimensional representation, capturing essential information and preserving any temporal dependencies. Subsequently, the decoder layer aims to faithfully reconstruct the input data from this compressed representation. Figure 6 illustrates the architecture of an LSTM autoencoder, highlighting the distinct components of the encoder and decoder segments, with each segment featuring an embedded LSTM module.

The LSTM autoencoder model shows promise in RF signal anomaly detection for several reasons. Firstly, using LSTM memory enables the model to capture long-term characteristics within RF signals. This capability can be used to identify anomalies within complex signals. Additionally, the autoencoder component facilitates dimensionality reduction by extracting the most relevant features from the input data. This aspect proves highly advantageous in RF signal analysis, which itself is often high-dimensional data. By effectively extracting relevant features, the LSTM autoencoder model enhances the detection of anomalies and distinguishes them from normal signal patterns.

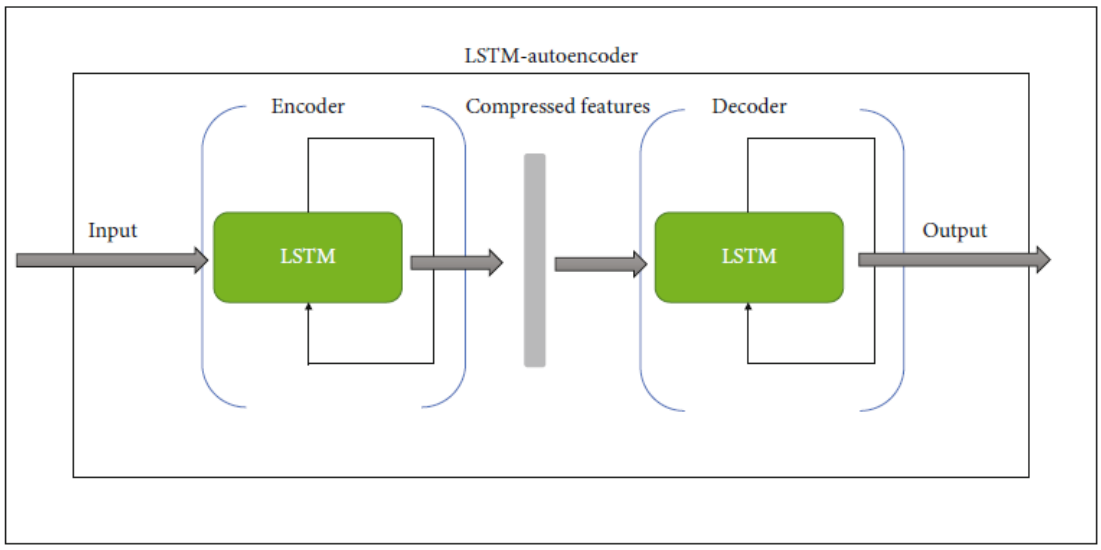


Figure 6. LSTM Autoencoder Hybrid Model Showing Structure for Encoding and Decoding Source: [32].

III. RF HARDWARE AND SOFTWARE

A. INTRODUCTION

This chapter will describe in detail the specific hardware components utilized in this thesis research, as well as the software and software functions utilized to achieve the research objectives. The overarching aim was to achieve similar results with hardware and software that contrasts the expensive, exquisite, and computationally powerful spectrum monitoring equipment that is used in naval communications. SDRs offer flexibility, adaptability, and cost-effectiveness in the RF domain. Having the typical RF hardware components hosted on the SDR enabled rapid changes and alterations without having to manage device compatibility. Mathworks matrix laboratory (MATLAB) software is a common programming software suite utilized in various Naval Postgraduate School (NPS) space systems academic group classes. MATLAB has various ML capabilities and offers interconnectivity with a variety of SDR classes. Simulink is a block diagram-based software that enables the creation of a variety of RF pathways and parameters.

B. HARDWARE

The hardware selected for this research was driven by its availability as inexpensive COTS components. The SDR chosen for data generation and testing was the ADALM-PLUTO SDR, a low-cost SDR that has cross-functionality with MATLAB. These SDRs were programmed directly from Simulink to facilitate data collection. Therefore, the SDRs were not run in a standalone configuration; two computers were used to drive each SDR. Figure 7 illustrates the test bet setup for this thesis. While the ADALM-PLUTO SDR can support a loopback configuration, over-the-air transmissions were desired for this research. As a result, separate SDRs were designated as the transmitter and receiver of the RF chain. Testing occurred in a controlled environment that sought to limit outside RF interference. While not all sources of RF could be completely controlled and eliminated, great lengths were taken to maximize the efficiency of both the transmission (TX) and receive (RX) SDRs during data generation and testing. Both the TX SDR and RX SDRs were equipped with the JCG401 2dBi antenna provided with the systems.

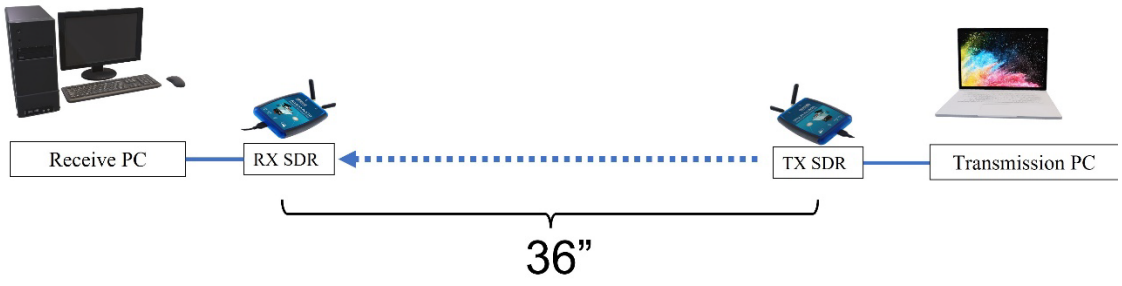


Figure 7. RF Test Bed Setup

1. ADALM-Pluto SDR

The ADALM-PLUTO SDR is the active learning SDR module from Analog Devices frequently used for RF and wireless communications experimentation. Utilizing the AD9363 board, it has a single TX channel and single RX channel that offers full duplex capability. Figure 8 demonstrates the compact size of the ADALM-PLUTO, which fits in the palm of a user’s hand. This small form factor offers versatility, allowing for flexible usage. ADALM-PLUTO boards can transmit analog signals from 325 megahertz (MHz) to 3800 MHz with a maximum sample rate of 61.44 mega samples per second at 20 MHz bandwidth. This offered a wide range of center frequencies to test and evaluate the designed communications pathway [33].

Figure 8. Image of the ADALM-PLUTO Software Defined Radio

2. ADALM-PLUTO Antenna

The antenna that is provided with the ADALM-PLUTO SDR is a 42mm long Global System for Mobile (GSM) antenna. The 2dBi, linearly polarized antenna is capable of both transmitting and receiving in the following ranges: 824 to 960 MHz and 1710 to 2170 MHz. The horizontal and vertical plane radiation patterns of various transmitting frequencies are outlined in Figure 9. It has an impedance factor of 50 ohms [34].

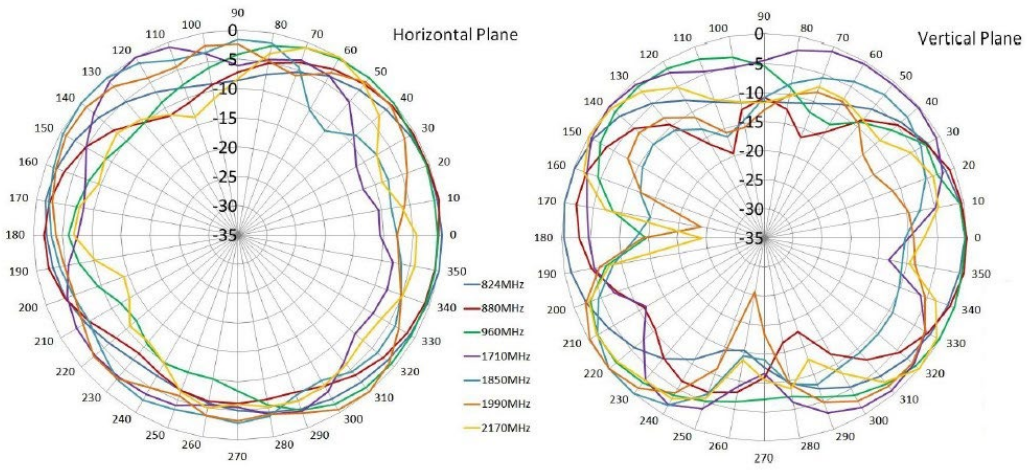


Figure 9. Radiation Patterns for JCG401 Antenna. Horizontal Plane (left) Vertical Plane (right). Source: [34].

3. AD9363

The AD9363 is an RF transceiver that integrates all RF, digital, and mixed-signal blocks within the same device. This integration reduces the need for additional hardware, increasing flexibility and versatility during testing. The analog-to-digital conversion (ADC) at the transmission side and digital-to-analog conversion (DAC) at the receiver side enables the rapid conversion of RF signals to and from digital components prior to passing them to the data interface component [35]. These connections, along with the local oscillators (LO), are all seen in Figure 10. Because the ADALM-PLUTO contains multiple components on the system on a chip (SOC), only the relevant portions that impacted the research of both the receiver and transmitter pathways will be highlighted in the following Receiver and Transmitter sections.

a. Receiver

Figure 10 illustrates the RX path of the larger functional diagram of the AD9363. The receiver operates with a direct conversion system that links its Low Noise Amplifier (LNA) and I&Q amplifiers. Additionally, this is down-converted to baseband for digitization. In this research, the automatic gain controller (AGC) will be enabled by utilizing the slow mode within the Simulink software component. The AGC "helps

stabilize the received signal amplitude to ensure an optimum loop design" [36], benefitting the carrier and symbol synchronizers used in the Simulink software component [36].

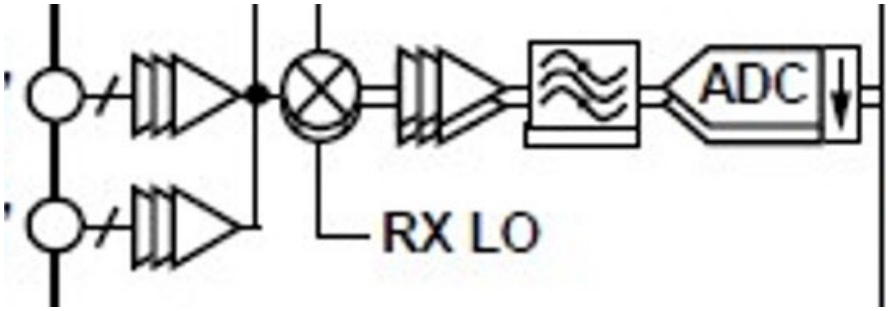


Figure 10. Functional Block Diagram of the AD9363 RX Path. Adapted from [37].

b. Transmitter

The TX section of the AD9363 provides all system blocks needed for the digital to analog-conversion. "Conversion of the signal to baseband analog signals, the I&Q signals are filtered to remove sampling artifacts and provide band shaping, and then are passed to the up-conversion mixers" [37]. This direct conversion architecture, illustrated in Figure 11, enables modulation at high accuracy with low noise degradations [37].

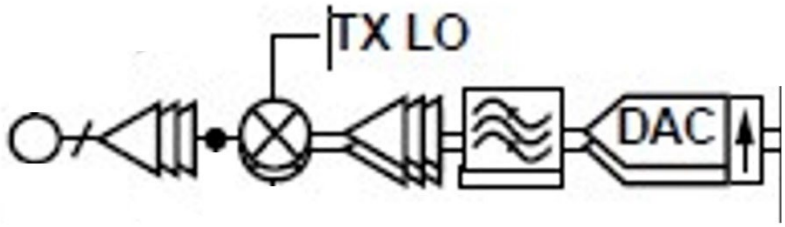


Figure 11. Functional Block Diagram of the AD9363 TX Path. Adapted from [37].

C. SOFTWARE

This section will discuss the software that was used in the research. The Simulink toolbox offered various data collection models compatible with the ADALM-PLUTO SDR within MATLAB. MATLAB served as a readily available software that provided ML models. The ADALM-PLUTO TX SDR was configured and programmed utilizing the 'QPSK Transmitter with ADALM-PLUTO radio model', and the RX SDR was configured utilizing the 'QPSK Receiver with ADALM-PLUTO radio model'. When utilized together, these models provide a complete RF communication system that incorporates the ADALM-PLUTO SDR. These models have default settings that were maintained, with the adjustments and research-relevant portions identified in the following transmitter and receiver sections of this chapter. Both models are available through the MATLAB Communications Toolbox [36], [38]. The additional toolboxes that were installed to add processing, collection, and simulation capabilities include Statistics and Machine Learning, Deep Learning, Signal Processing, Predictive Maintenance, HDL coder, DSP System, and Simulink toolboxes.

1. QPSK Transmitter with ADALM-PLUTO Radio

The transmission pathway of the RF system utilized the 'QPSK Transmitter with ADALM-PLUTO Radio' model, with the addition of an Additive White Gaussian Noise (AWGN) channel highlighted in blue in Figure 12. This transmitter model incorporated a bits generation subsystem, a QPSK modulator block, and a raised cosine transmit filter block. The modulation for this model was Quadrature Phase Shift Keying, which has one of four possible phase carrier shifts (0, 90, 180, 270 degrees). QPSK allows for a 2-bit per symbol modulation technique, maximizing the transmitting bandwidth.[39]. The ADALM-PLUTO transmitter block assigned generated message, post modulation and filtering, to a center frequency prior to its transmission. No additional gain was added to the transmitter model.

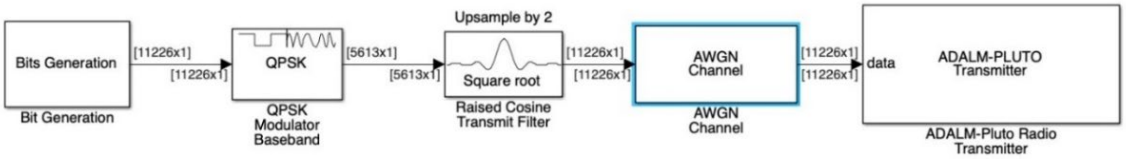


Figure 12. QPSK Transmitter with ADALM-PLUTO Radio Simulink Block Diagram. Adapted from [38].

The Bits Generation block encoded a "Hello World XXX" message, repeated 100 times as a frame payload. Each frame contained 100 of the Hello World messages combined with a Header. The Header was a 13-bit Barker code that served as a sync to better facilitate decoding in the received model. Figure 13 outlines the Simulink block diagram showing the frame size as it passes through the diagram.

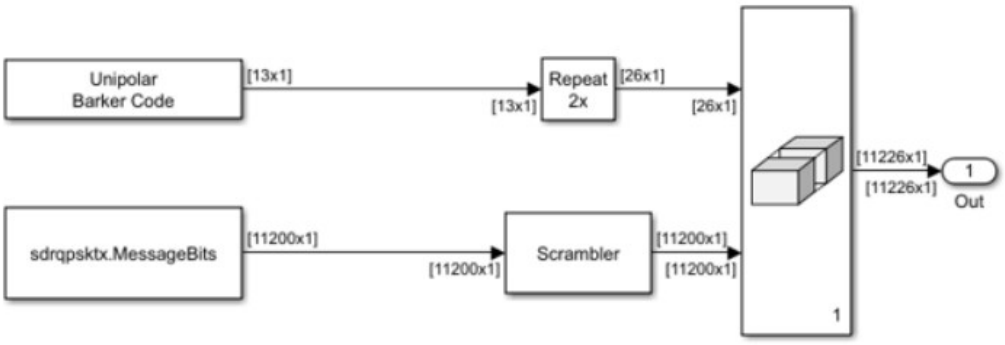


Figure 13. Bits Generation Block Structure. Source: [38].

The AWGN channel allowed for intentional interference as a controlled variable for adjusting the SNR in the signal, as seen in Figure 13. The SNR used to generate the data sets is shown in Table 2. SNR values ranged from 20 dB down to 5 dB. A data collection set was also generated when the RX SDR simulated a collection period, but no accompanying TX signal was transmitted. This data set was used as a reference for the surrounding noise floor and labeled 'EMPTY'.

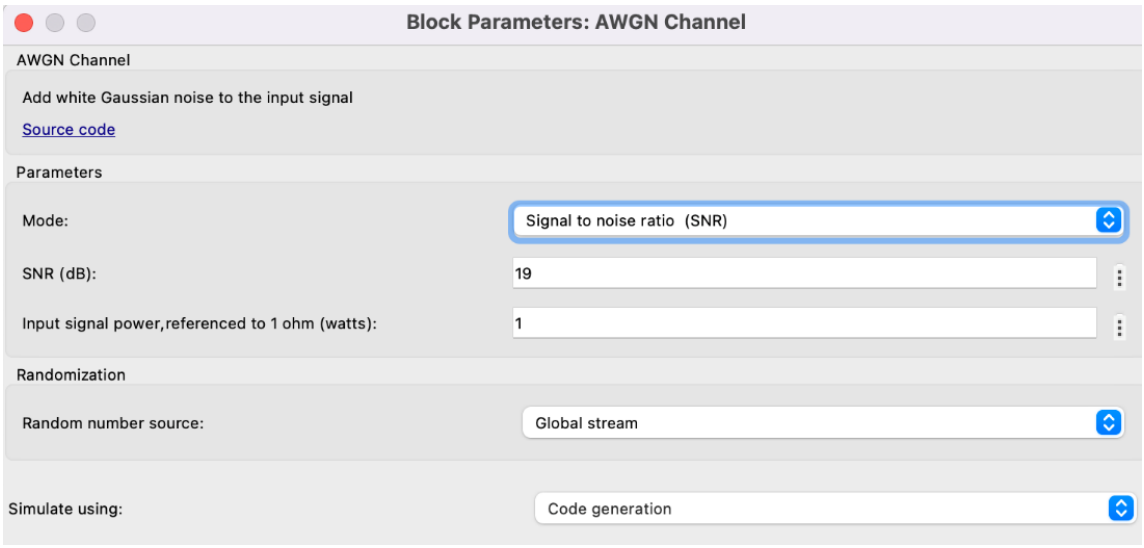


Figure 14. Input Parameter View for AWGN Channel Block.

2. QPSK Receiver with ADALM-PLUTO Radio

The RX portion of the RF system utilized the 'QPSK Receiver with ADALM-PLUTO Radio' model and added several collection and spectrum monitoring areas in the block diagram. The QPSK receiver model has the ADALM-PLUTO receiver block that feeds into the QPSK receiver subsystem. The QPSK receiver subsystem demodulates the received symbols and outputs the received "Hello World" message to the command line.

The bit error rate (BER) was calculated in the BER block. The first display of the BER block was the overall bit error rate of the transmission. The second line in the display was the total number of dropped bits in the transmission. This provides an immediate indicator if the model is experiencing greater than expected interference as the value will continue to increase. The third display in the BER block is the total number of bits received. In Figure 14, a To Workspace block is highlighted blue. This block collected the BER values calculated during the simulation and was saved as a `Structure with Time` element.

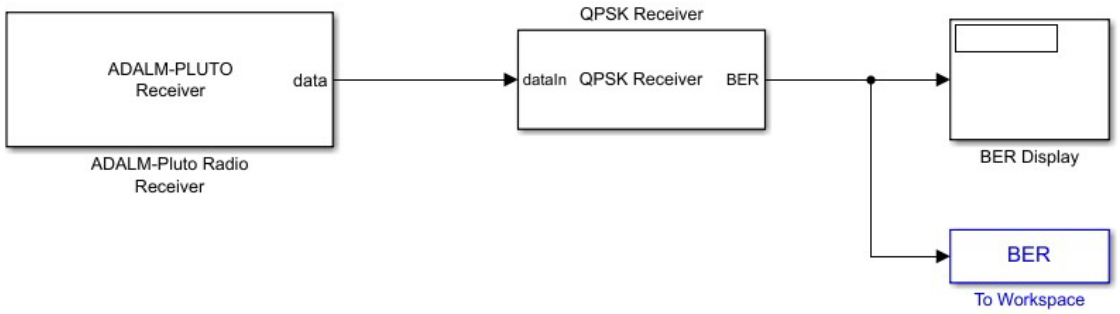


Figure 15. QPSK Receiver with ADALM-PLUTO Radio in Simulink Block Diagram. Adapted from [36].

a. QPSK Receiver Block

The QPSK receiver block, shown in Figure 15, includes several processes that manage the transmission from the transmitter. The AGC block adaptively changes the gain to achieve a constant signal level for the output signal, maintaining the SNR of the transmission. The Raised Cosine receive filter had a filter span of 10 symbols and a roll-off factor of 0.5. The Course Frequency Compensation block compensated for carrier frequency offset experienced in the transmission. The maximum frequency offset was 12500 Hz. The symbol synchronizer resamples the input signal so that symbol decisions are made at the optimum sampling instants. The Carrier Synchronizer compensates for the residual frequency offset as well as any phase offset that may occur in the transmission. The Preamble block detects the 13-bit frame header, and the Frame Synchronizer aligns the frame with the known header block. The Data Decoding subsystem finally demodulates the signal and decodes the transmission displaying its message in the command line. The Complex to Real Image block, highlighted blue in Figure 15, was added to the model to collect the real portion of the I&Q data to serve as the data set. The 'Real2' workspace variable was collected in structure with simulation time format. The complex RF signal was split into the real and imaginary portions as they relate to the in-phase (I) and Quadrature (Q) portions of the signal. Since the in-phase component, or real value, captures the amplitude variations of the signal at a given point in time, it was selected as the portion

of the complex signal to utilize. Additionally, the ML models used in MATLAB for this thesis could not employ complex data formats for analysis.

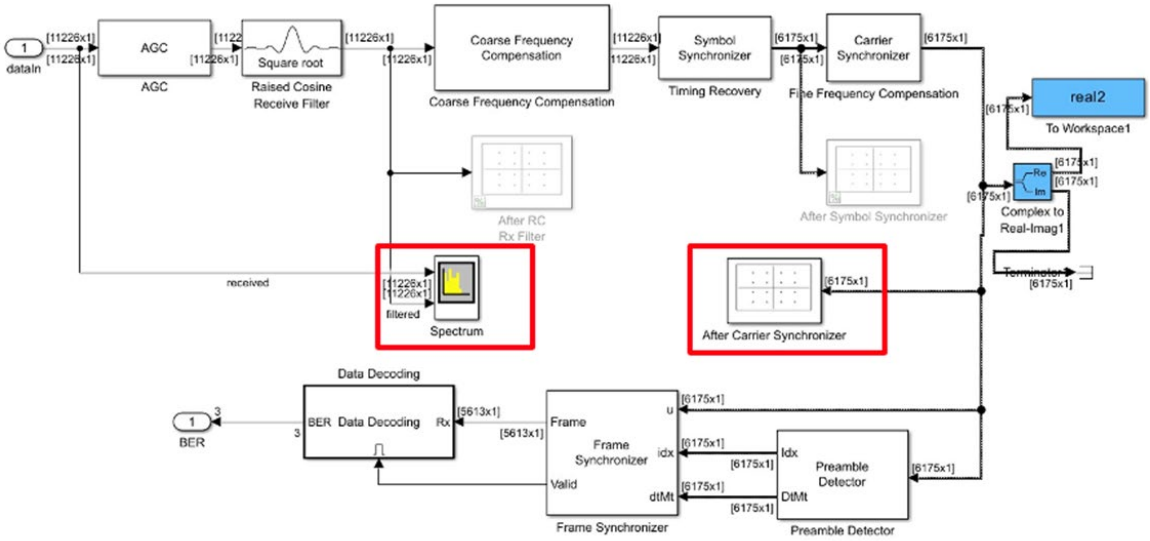


Figure 16. Receiver Block Diagram, I&Q Collection Points in Blue, Display Blocks in Red. Adapted from [37].

The Spectrum display block displayed the received signal in yellow and the filtered signal in blue. During the monitoring of transmissions, the Spectrum Scope display and After Carrier Synchronizer display, highlighted in red in Figure 15, provided a visual confirmation of successful reception. Figure 16 presents a transmission with an SNR of 20 dB, where the After Carrier Synchronizer displayed the QPSK constellation based on the I&Q data. In Figure 17, a transmission with an SNR of 9 is depicted, showing a more dispersed and spread QPSK constellation compared to the normal transmission relating to a higher ratio of interference compared to the normal signal.

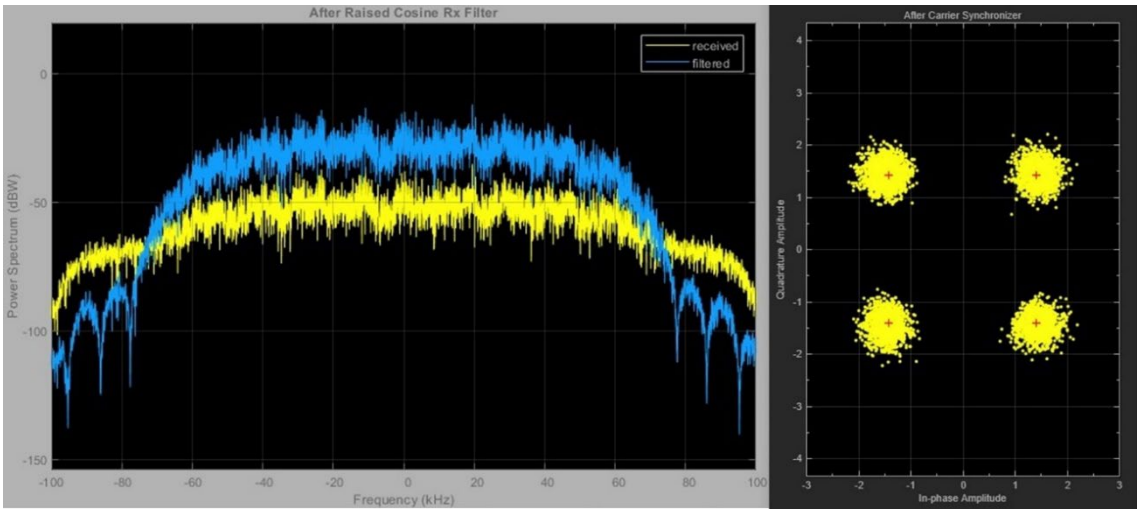


Figure 17. Images of Scope (left) and After Carrier Synchronizer (right) Showing SNR 20 Datasets QPSK Modulation

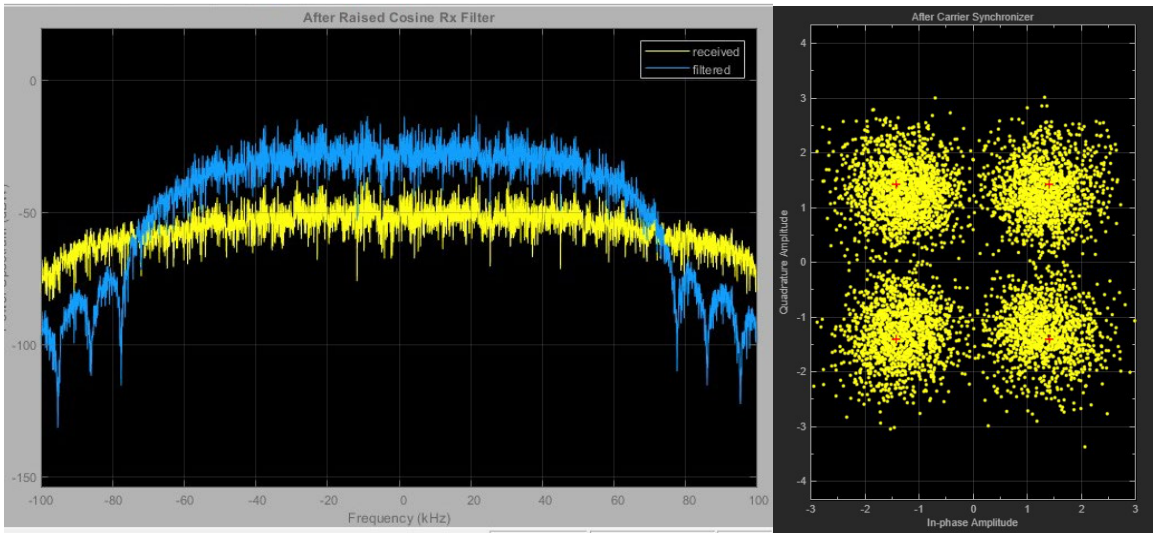


Figure 18. Images of Scope (left) and After Carrier Synchronizer (right) Showing SNR9 Datasets QPSK Modulation

As transmissions were received, the display window produced a decoded numbered 'Hello world' message. Figure 19 illustrates the command window displaying a perfectly received and decoded transmission. The display window allowed for a visual confirmation that the transmission was successful.

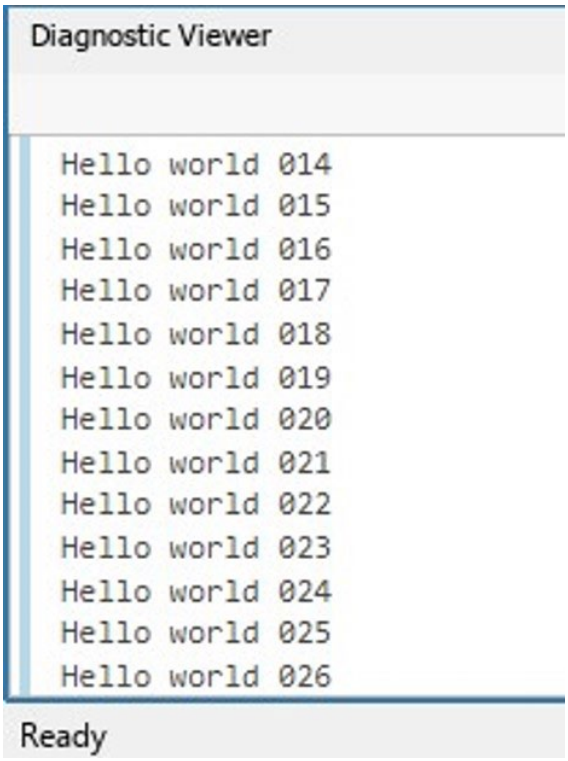


Figure 19. Transmission Payload Display

D. CHAPTER SUMMARY

This chapter reviewed the specific hardware and software that was used in the thesis research. The aim was to achieve comparable results using COTS hardware and software, contrasting expensive, computationally capable systems that are currently used in spectrum monitoring. ADALM-PLUTO SDRs were used for their versatility, flexibility, and adaptability for RF signal generation and collection. MATLAB and Simulink provided the integration software for the SDR and ML model development. The AD9363 SOC found in the ADALM-PLUTO integrated the necessary RF, digital, and conversion blocks that enabled the optimum signal generation and collection test bed. The complex RF signal was split into real and imaginary parts, with the real portions enabling the appropriate variation data used as data for the ML models.

IV. DATA GENERATION

In order to evaluate ML techniques effectively, it is crucial to use a sufficiently large dataset. Tasks such as image classification, handwriting prediction, and voice/audio datasets benefit from large datasets that exist in repositories from various sources. Since a model's effectiveness is related to the data quality used for training [40], it was necessary to establish specific parameters for the data used in this thesis. The data required for this research included RF I&Q data modulated with known parameters. Since the amount of required data was unknown at the outset, all the necessary data was generated locally. This approach provided the most control over quantity but eliminated any external source's responsibility for the quality of the data generated. This chapter will describe the collection parameters of the data generated in the RF path and outline the steps taken to preprocess the data before feeding it to the ML algorithms.

A. RF PORTION

1. RF Transmitter Settings

Table 1 provides the values and formulas for each block of the Simulink RF model. The interpolation factor doubled the frame size resulting in an overall frame length of 11226 bits. In addition, a Raised Cosine Transmit Filter was applied to the modulated data, upsampling the data by 2.

The data was transmitted through the ADALM-PLUTO SDR on a carrier wave with a center frequency of 915 MHz. The front-end sample rate of 200 kHz was identified as the ideal sample rate for this setup. Increasing the sample rate resulted in garbled message displays, making data generation difficult. The SDRs utilized a USB 2.0 connection with the PCs and, during sample rates greater than 200 kHz, became backlogged in transferring the data from the SDR to PC, resulting in garbled transmissions. These signal parameters were utilized to generate consistent data within the specifications of the SDR for analysis with the ML algorithms.Transmitter Settings and Value Derivations

Table 1. Transmitter Settings and Value Derivations

Parameter	Formula	Value
Center Frequency	Set by user	915,000,000 Hz
Symbol Rate	Set by user	100,000 Hz
Transmitter Gain	Set by user	0 dB
Sample Rate	$Symbol\ Rate * Interpolation$	200,000 Hz
Frame Size	$\frac{Header + Payload}{log2(Mod\ Order)}$	5,613 Bits
Frame Length	Frame Size * Interpolation	11,226 Bits
Frame Time	$\frac{Frame\ Length}{Sample\ Rate}$	0.05613 seconds
Modulation Order	QPSK alphabet (0,90,180,270)	4
Interpolation	N/A	2

2. RF Receiver Settings

The symbol synchronization block took frames of fixed length and applied a phased locked loop to correct any timing inconsistencies in the received signal [41]. The symbol synchronizer block utilized a Garder algorithm to output one symbol for every two input symbols. An effect of channel timing delays is that the output sample per symbol may reach a symbol boundary resulting in inconsistent-sized outputs [41]. These inconsistencies are seen in the 11226x1 input converted into a 6175x1 output. Because the expected output is 5613x1 in a perfectly timed and aligned system, there are extra values or omissions of values in the 6175 sample [36]. These must be accounted for and removed, which will be discussed later in the Data Preprocessing section.

Each sample collected was an array 6175x1x1 long. The first dimension, 6175, represented the data points collected for each sample time. The second dimension, 1, represented the data sets collected in each sample. Since there would only be one channel transmitting one set of samples, this dimension remained one no matter the length of the simulation. The final dimension, 1, represented the number of sets collected in each sample. Since the frames arrived at a sample time of 0.05613 seconds, we could determine how long to run the simulation based on the desired number of samples.

The BER calculation utilized the Data Decoding and Frame Synchronizer blocks. These blocks aligned frames by correctly identifying the Barker code and then decoded the symbols into the numbered 'Hello World' Displays accordingly.

Table 2 has the various levels of SNR and associated labels used for the data sets. Both ML models were unsupervised and did not require labeled datasets. However, an evaluation metric utilized labeled datasets for generating a confusion matrix.

Table 2. SNR Levels for Each Collection Point and Associated Label.

Signal to Noise Ratio Collected	Data Label
SNR 20 dB	'Normal'
SNR 19 dB	'Nineteen'
SNR 17 dB	'Seventeen'
SNR 15 dB	'Fifteen'
SNR 13 dB	'Thirteen'
SNR 11 dB	'Eleven'
SNR 9 dB	'Nine'
SNR 7 dB	'Seven'
SNR 5 dB	'Five'
No Signal TX	'EMPTY'

The command window displayed the decoded 'Hello world' transmission. During the RF generation portion, SNR 20, SNR 19, SNR 17, and SNR 15 were 100% of the transmission displayed, making it difficult for visual confirmation of interference in the transmission. While there were some bits that were dropped during the transmission, all four of these SNR levels enabled a complete transmission of data with very few error bits. Figure 20 displays the indistinguishable SNR 20 and SNR 15 command windows. Beginning with SNR 13 and ending with SNR 5, the level to which the 'Hello world' transmission was legible would decrease, enabling visual indication that there was interference present in the signal. Figure 21 illustrates the increasingly scrambled displays of SNR 9 and SNR7.

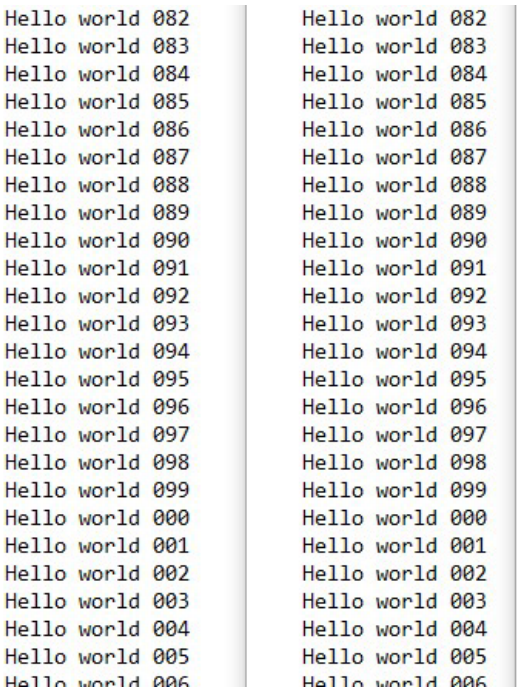


Figure 20. Command Window Display of SNR 20 (left) SNR 15 (right)

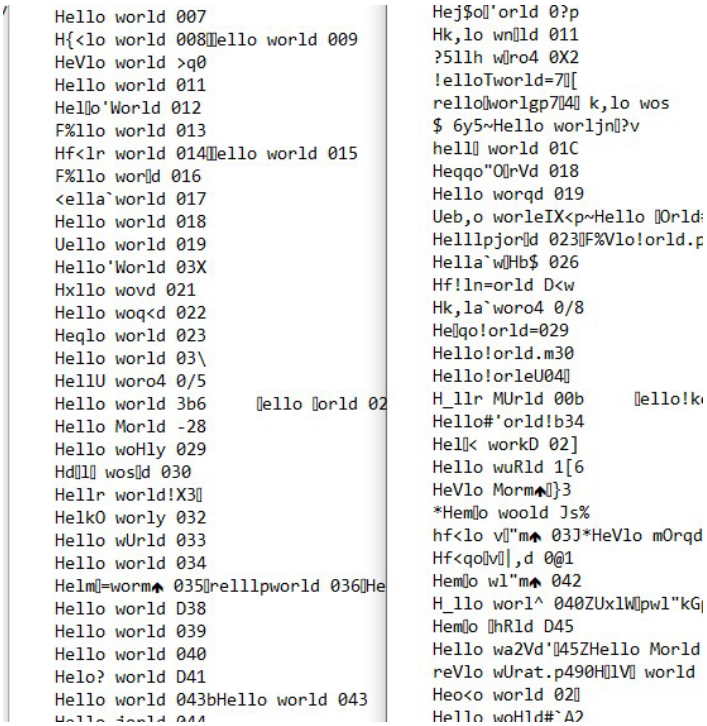


Figure 21. Command Window Display of SNR 9 (left) and SNR 7 (right)

B. DATA PREPROCESSING

After collecting data in the receiver model, it is essential to preprocess the data for utilization in the ML models. The extent of preprocessing required varied depending on the type of autoencoder being employed. Although MATLAB software was utilized for data collection, processing, and ML model training, the data preparation involved various conversions between different data structures or utilizing functions to extract details from the data. In the case of the LSTM autoencoder, the data was transformed into compressed representations called features. These features represent the entire range of data and serve as the input for the ML model.

On the other hand, the basic autoencoder, also an unsupervised learning technique, utilized the raw, unlabeled data as input for the model. Both autoencoder models necessitate data partitioning into training and testing sets, but the LSTM autoencoder model utilized a validation set to assess the performance during training. The evaluation metrics of these ML models will be further discussed in Chapter V.

1. LSTM Autoencoder

Preparing data for the LSTM autoencoder involves trimming each dataset to a uniform size, accommodating any outlier data, and extracting representative features from each dataset. The last step in data preprocessing is partitioning data into separate training, validation, and testing sets. The data sets were partitioned into training, testing, and validation sets. The partitioned percentages were 80, 10, and 10, respectively.

a. Removal of NaN

A visual review of the data sets revealed several cells that contained the term, Not a Number, abbreviated 'NaN.' NaN was a representative term used when data was not calculated or received during a collection. In this case, after the symbol synchronizer resampled the data, in any place that did not have data, the synchronizer block inserted a 'NaN' value in the 6175x5001 array. The first step was to search the array for the first appearance of a NaN value, then trim the array at the appearance of the first NaN value. For all datasets where an actual RF transmission was sent, the last row with a numerical

value was row 5613. This aligned with the frame size being 5613 bits in length. Therefore, each data set was trimmed to 5613x5001 size. Table 3 lists the data set and the corresponding row where the first NaN value was identified. Data set 'SNR EMPTY' was the outlier, as the first appearance of a NaN value was in row 548. Since the synchronizer block did not uniformly insert NaN values as it resampled the data, some samples had a NaN value in row 5613, and some did not. In order to solve this, a mean vector for each data set was created from all the other values in each column. Then in any location where a NaN value occurred, the corresponding mean from the vector was placed to fill in missing data. This ensured all 5001 samples had 5613 values, and in any column that needed a NaN value replaced, a mathematically derived value was used and not ignored or replaced with 0.

Table 3. SNR Data Sets, Data Dimension for Each Set, and First Row that Contained a NaN Value

Data set	# NaN rows found in 6515x5001 array	Trimmed Dimension	# of NaN values found in row 5613
SNR 20.mat	563	5613 x 5001	4
SNR 19.mat	563	5613 x 5001	2
SNR 17.mat	563	5613 x 5001	7
SNR 15.mat	563	5613 x 5001	4
SNR 13.mat	563	5613 x 5001	16
SNR 11.mat	563	5613 x 5001	32
SNR 9.mat	563	5613 x 5001	34
SNR 7.mat	563	5613 x 5001	45
SNR 5.mat	563	5613 x 5001	47
SNR EMPTY.mat	548	5613 x 5001	0

As the SNR decreased, resulting in more significant interference, the ability of the symbol synchronizer to adequately sample the received signal decreased, resulting in greater numbers of NaN values in row 5613. Other than in SNR15 and SNR17 data sets, the NaN count in the rest of the data sets increased as the SNR value decreased. The EMPTY data set had a much lower count of NaN rows, which indicated that the RX symbol synchronizer was inputting values much further into the 6175x5001 array to synchronize a

sample from a signal that did not exist. For uniformity, the EMPTY data set had the same dimensions, 5613x5001, as the other data sets to maintain continuity.

b. Feature extraction

Feature extraction is the process of creating useful information from a set of raw data points. This new data set will be heavily condensed yet still representative of the more extensive raw data set [42]. Features design varies based on the data type or created from standard analysis methods. MATLAB offers a Diagnostic Feature Designer application that enables feature design and extraction from a data set. The features extracted were based on a time-series domain and were a collection of 13 basic statistical functions derived from the data sets. The features were grouped into statistical features, impulsive features, and harmonic features. The statistical features were, mean, root mean square, standard deviation, shape factor, kurtosis, and skewness. The statistical features provide information about the distribution, variability, and any trends present in the data [43]. The impulsive features included crest factor, impulse factor, clearance factor, and peak value. Impulsive features identify sudden changes in the data, spikes, or outliers [43]. The harmonic features included; SNR, total harmonic distortion, and signal to noise to distortion. Harmonic features capture periodic or repetitive patterns and are derived from the frequency domain representations of the data [44].

(1) Diagnostic Feature Designer

The Diagnostic Feature Designer application in MATLAB enables the visualization of raw data so that features can be designed that indicate particular conditions or relationships. In order to utilize this application, a table containing two sets of appropriately labeled raw data for comparison must be loaded, and various analyses or comparisons can be conducted [45].

To maximize the effectiveness of the features identified using the application, features were selected that enabled distinguishing two data points that were not easily achievable through human analysis of the raw data. The diagnostic feature designer refers to a collection of data points as an ensemble. The statistical functions are applied to the ensemble, and histograms are generated. Members reference a single individual data set

where a particular function is applied [46]. After the histogram representations are generated, the diagnostic feature designer application can help determine the top features to apply to the data set by ranking the features in a table. The rankings can be based on simple T-score analysis between two data sets or through unsupervised or supervised rankings. For this thesis, all 13 features were chosen to provide the highest opportunity to distinguish between the two data sets. Histograms are color-coded to identify corresponding data sets. Features with the greatest discrimination between the two datasets are identified by those whose histograms do not overlap and offer the greatest separation, as seen in Figure 22. Those whose histogram representations overlap describe features that are similar in characteristics and, therefore, not easily distinguishable to the human eye. The histograms could be ranked in order by the variance that existed between the two data sets being analyzed in the Diagnostic Feature Designer. It is important to note that when comparing various SNR data sets, such as SNR 19 and SNR 17, the histogram separation will decrease, and therefore the NN's ability to distinguish the histogram separations will become more important.

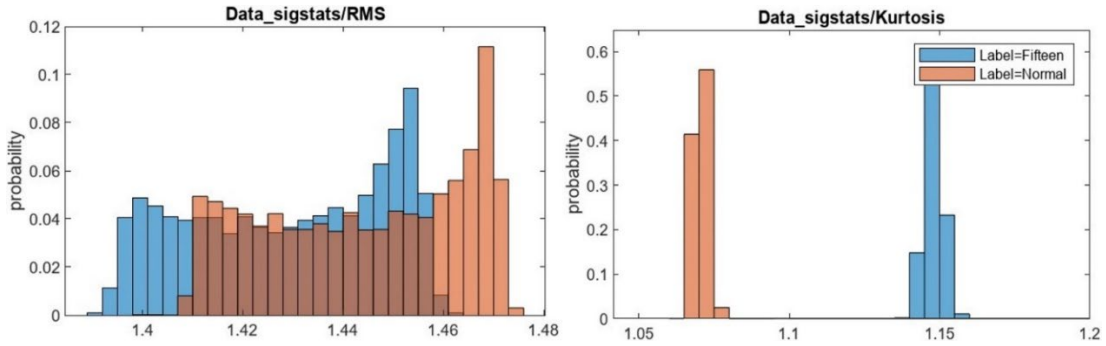
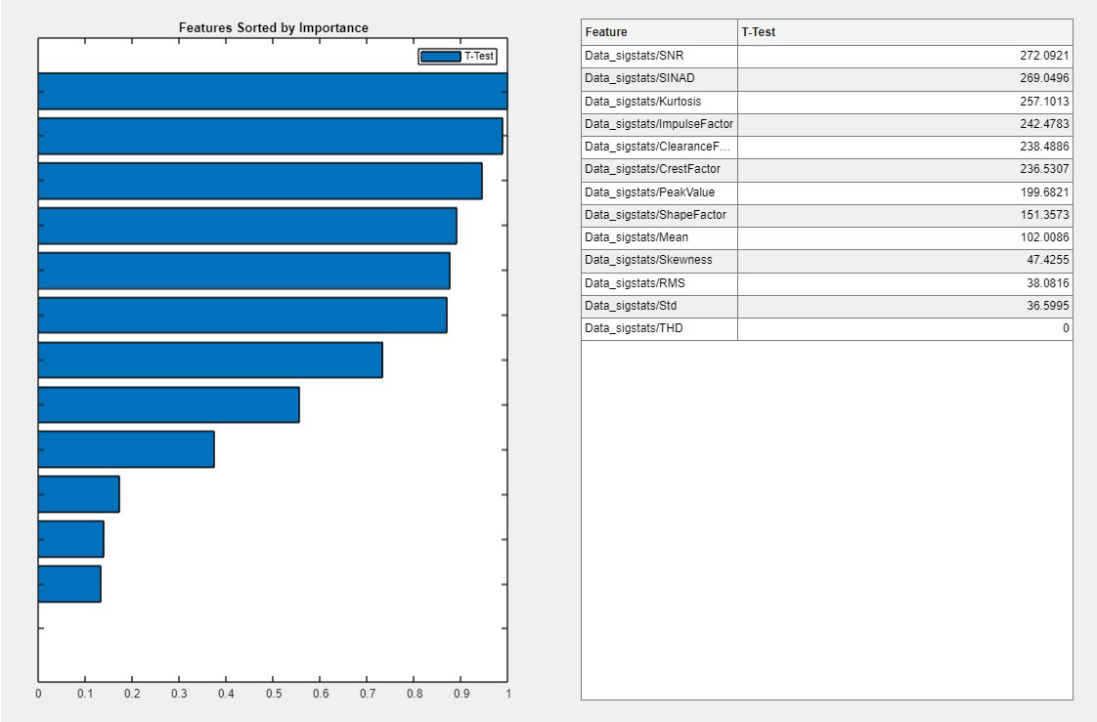


Figure 22. Histogram of Raw Data Sets in the Feature Designer Application. Least Useful (left) Most Useful (right)

Figure 23 shows the ranking, from greatest to least, of feature importance between the SNR 15 and SNR 20 data sets in both a table view and bar graph ranking. The greater the histogram separation, the greater the ranked importance.



Feature	T-Test
Data_sigstats/SNR	272.0921
Data_sigstats/SINAD	269.0496
Data_sigstats/Kurtosis	257.1013
Data_sigstats/ImpulseFactor	242.4783
Data_sigstats/ClearanceF...	238.4886
Data_sigstats/CrestFactor	236.5307
Data_sigstats/PeakValue	199.6821
Data_sigstats/ShapeFactor	151.3573
Data_sigstats/Mean	102.0086
Data_sigstats/Skewness	47.4255
Data_sigstats/RMS	38.0816
Data_sigstats/Std	36.5995
Data_sigstats/THD	0

Figure 23. Feature Rankings Bar Graph View (left) Rank Table (right)

c. Normalization of data

After extracting the features from the data sets, the output will be a 14x5001 array where each sample has a set of 13 feature input values and a desired output label. The features will be the data utilized for model training. The next step in preprocessing the data is to ensure that each feature is normalized between the two data sets. This is referred to as the normalization of data [47]. Normalization is a technique where all data is formatted to a similar scale. Multiple types of normalizing techniques exist. However, the type that is utilized for this data set will be scaling the data to a particular range as described in [47]. Scaling data involves converting data ranges from their natural variation to a range from 0 to 1. This scaling is accomplished through Min-Max normalization, where each feature data point is subtracted from the minimum value and divided by the mean of the feature set [47],[48].

$$x' = (x - x_{min})/(x_{max} - x_{min}) \quad 12$$

In equation 12, x' is the normalized data while x is the feature data point being normalized, and the x_{max} and x_{min} are the upper and lower ranges of the data. Normalization will occur with each feature, so the min and max ranges will be unique to the column feature data. Each data set will be normalized using the x_{min} and x_{max} values of the normal data set to scale all data to the same range. Normalization benefits LSTM autoencoder learning as it attributes the same importance to all the features while also ensuring that all the data lie within the linear range of the network's observations. Even though each feature histogram was ranked to determine the highest degree of discrimination between the data, all available features were selected. Utilizing all 13 features as training data for the model results in each feature having equal importance for learning the LSTM autoencoder layers and collectively contributing to the model's learning.

d. Removal of Infinite Values

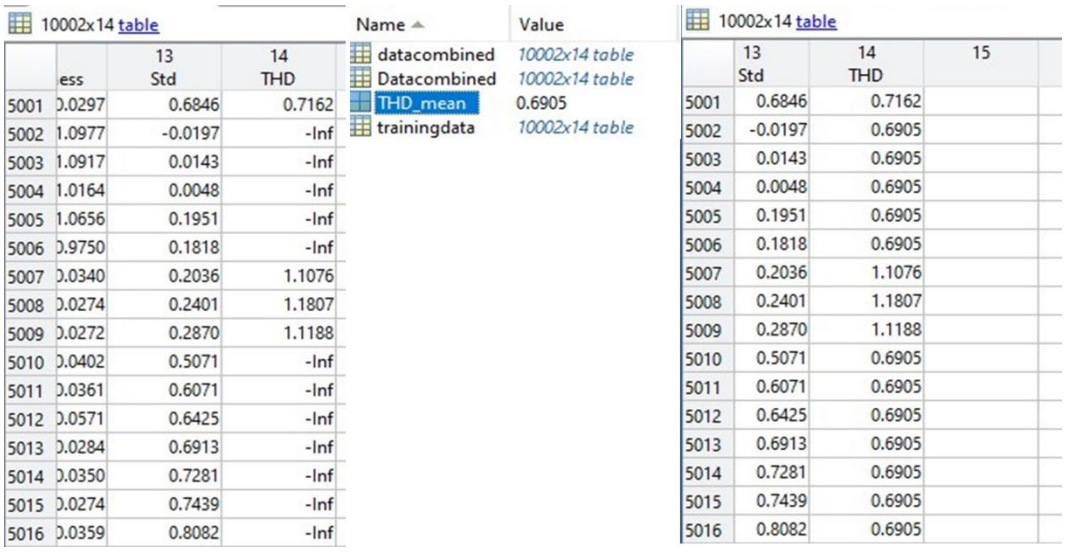
After extracting features from the data, then normalizing the data sets, the last stage of preprocessing is required, which involves checking the data and ensuring that all columns and rows have numerical values. Errors when deriving means or conducting the statistical analysis can arise if non-numerical values are included in the calculations. Feature 13, total harmonic distortion, was a repeat offender in having non-numerical results in all SNR data sets. Total harmonic distortion refers to the amount of distortion a select signal has compared to the fundamental frequency or purely sinusoidal signal. Derived by taking the ratio of RMS of the harmonics of the signal being analyzed to the RMS of the fundamental frequency. The derivation for calculating THD is found in [44].

$$THD = \sqrt{\frac{(V_1+V_2+V_3+V_4 \ldots V_n)}{V_{fundamental}}} \qquad 13$$

THD has been used as an indicator of the quality of a system. It usually is combined with other qualitative factors that dictate transmission quality. This quality determination is outside the scope of the research. When calculating the THD as a feature, some elements have a result of infinity or negative infinity. This is due to one of several factors. When dividing a number by zero or attempting to take the square root of a negative number. Since there are various reasons why infinity or negative infinity are derived when calculating

THD, any instance or presence will be removed or replaced using the same methodology as NaN value replacement.

Not all values in the THD feature column were infinite, and different SNR data sets had various instances of infinity or negative infinity being present. Figure 24 shows a sample of the data from SNR 17 containing a high number of Infinity or negative Infinity entries. Overall, each data set was searched, and every instance of infinity was identified. Next, the column-wise mean calculation was conducted on all finite values within each column. This mean value was used as a replacement value for infinity or negative infinity to complete the data for the THD feature column.



10002x14 table

	ess	13 Std	14 THD
5001	0.0297	0.6846	0.7162
5002	1.0977	-0.0197	-Inf
5003	1.0917	0.0143	-Inf
5004	1.0164	0.0048	-Inf
5005	1.0656	0.1951	-Inf
5006	0.9750	0.1818	-Inf
5007	0.0340	0.2036	1.1076
5008	0.0274	0.2401	1.1807
5009	0.0272	0.2870	1.1188
5010	0.0402	0.5071	-Inf
5011	0.0361	0.6071	-Inf
5012	0.0571	0.6425	-Inf
5013	0.0284	0.6913	-Inf
5014	0.0350	0.7281	-Inf
5015	0.0274	0.7439	-Inf
5016	0.0359	0.8082	-Inf

Name	Value
datacombined	10002x14 table
Datacombined	10002x14 table
THD_mean	0.6905
trainingdata	10002x14 table

10002x14 table

	13 Std	14 THD	15
5001	0.6846	0.7162	
5002	-0.0197	0.6905	
5003	0.0143	0.6905	
5004	0.0048	0.6905	
5005	0.1951	0.6905	
5006	0.1818	0.6905	
5007	0.2036	1.1076	
5008	0.2401	1.1807	
5009	0.2870	1.1188	
5010	0.5071	0.6905	
5011	0.6071	0.6905	
5012	0.6425	0.6905	
5013	0.6913	0.6905	
5014	0.7281	0.6905	
5015	0.7439	0.6905	
5016	0.8082	0.6905	

Figure 24. Infinity Values in THD Column of SNR17 Data Set. (left) THD Mean (center) Infinity Values Replaced (right).

e. LSTM Autoencoder settings

When constructing an LSTM autoencoder through MATLAB, various hyperparameters can be set to design the model as desired. The function trainNetwork(x) is a neural network default function offered through MATLAB. Using the parameters listed in Table 4, the function was converted to an LSMT autoencoder model. There were three bidirectional LSTM layers; an encoder with 32 cells, followed by another layer of 16 cells,

that was followed by a decoder with 32 cells. In between each layer, there is a rectified activation function. The activation function adds non-linearity to the output of the LSTM layers aiding in the model's effectiveness in learning the features of the data as it passes through the layers [49]. Additionally, setting the optimizer function and learning rate parameters enabled the shaping of the neural network into an LSTM autoencoder model.

Table 4. Parameters of the Layer and Training Options for Neural Network Models.

Layer parameter	Value	Name/Function
FeatureDimension	1	
BiLSTM layers	32,16,32	Bidirectional LSTM layers
Activation function	relu1, relu2, relu3	Rectified Activation Function
Training options parameter	Value	Name/Function
Optimizer	Adam	Adaptive moment estimation
Learning rate	0.001	
Mini Batch size	500	
Epoch	200	The number of complete passes over the training set.
Validation data	500x1	10% of the total data set
Validation Frequency	50	Every 50 iterations of training data were validated.
Plots	'training-progress'	Plots mini-batch size and validation accuracy.

(1) Training Options

The optimizer for this LSTM autoencoder model was the Adaptive moment estimation (Adam). Adam optimization is a widely used and well-tuned optimizer function used in neural network training [50]. The learning rate parameter remained the default 0.001, as increasing the value would decrease training time but result in a failure of the model to converge during training resulting in sub-optimization [51]. The mini-batch size represents the subset of the total data fed through the layers in the model. Epoch size refers to the total number of passes the model will go through in the partitioned training data set and is explained in detail in [51]. Increasing the epoch size would result in a longer training time. A training size of 200 was found to produce good results without overfitting.

Validating the data was a parameter that was added to the model, which did not come as a standard option for the trainNetwork(x) model through MATLAB [52]. Validation data was not used to train the model but used while training as a method to monitor the model's performance and prevent any over or underfitting. Validation loss was the metric derived during the training, with the optimal goal to have the validation root mean square error near the resulting training RMSE [53]. Validation checks occurred at every 50 iterations of training.

Visual monitoring of the model's training occurred by plotting the training progress. A black circle indicated every validation step, and the RMSE of the training data was tracked along all epochs and training iterations.

(2) Layer Construction

Two separate model configurations were designed using The LSTM autoencoder model. The first model had a layer construction of 32-16-32. The second model was designed with a much smaller layer construction, 13-6-13. This was to determine if the layers with greater node counts were required to identify interference successfully and to what extent the 13-6-13 could offer results. The typical layer construction used for both LSTM autoencoder models is illustrated in Figure 25.

Figure 25. LSTM Autoencoder Layer Construction

2. Autoencoder

The basic autoencoder used in this thesis was also trained utilizing the MATLAB trainAutoencoder(x) function. The input data for this model did not require feature extraction but did require the removal of the NaN values, and normalization preprocessing

steps. The input matrix for each dataset would remain in the 5613x5001 format after preprocessing.

a. Training Parameters

The hidden layers parameter of the basic autoencoder model played a crucial role in how the model learned and was able to recreate the compressed features of the input data. With additional neurons, the hidden layer can capture finer details and variations in the data, allowing for a more expressive and detailed representation of the input. Adjustments were made to the size of the hidden layers to reach a balance between the size of hidden layers and recreation accuracy without overfitting. The hidden layer values ranged from 4, 13, and 32 neurons in each hidden layer. Table 5 outlines the training parameters for the basic autoencoder models and the length of time for training the model.

Table 5. Training Parameters for Basic Autoencoder Models

Parameter	Model 4	Model 13	Model 32
Epoch	400	400	400
Hidden layer size	4	13	32
Performance metric	Mean Square Error	Mean Square Error	Mean Square Error
Training method	Scaled Conjugate Gradient	Scaled Conjugate Gradient	Scaled Conjugate Gradient
Encoder transfer function	Logsig	Logsig	Logsig
Decoder transfer function	Logsig	Logsig	Logsig
Training time	10 min 27 sec	11 min 25 sec	13 min 34 sec

The training method for the basic autoencoder included the scaled conjugate gradient (SCG) method, an optimization algorithm now commonly used for training neural networks. The SCG optimization was the default training algorithm for the trainAutoencoder (x) function in MATLAB, and the reason this algorithm was utilized versus the Adam training algorithm used in the LSTM autoencoder. Moller [54] describes

how the SCG method incorporates derivative information to estimate step sizes and directions, facilitating faster convergence and improved accuracy in model training.

The encoder and decoder transfer function was the Logistic sigmoid (logsig) function, described in [55] as

$$f(z) = \frac{1}{1+ e^{-z}} \qquad 14$$

where z is the output from the input layer. The logsig function is useful as an activation function by introducing the non-linearity, enabling the network to learn any complex relationships between inputs and outputs.

When using the trainAutoencoder(x) function, there are a number of default settings associated with the training of the algorithm. These default settings will be listed in the Appendices. Only the parameters that were altered were identified in Table 6. Figure 26 illustrates the autoencoder construct, highlighting the hidden layer size and transfer functions of the encoder and decoder.

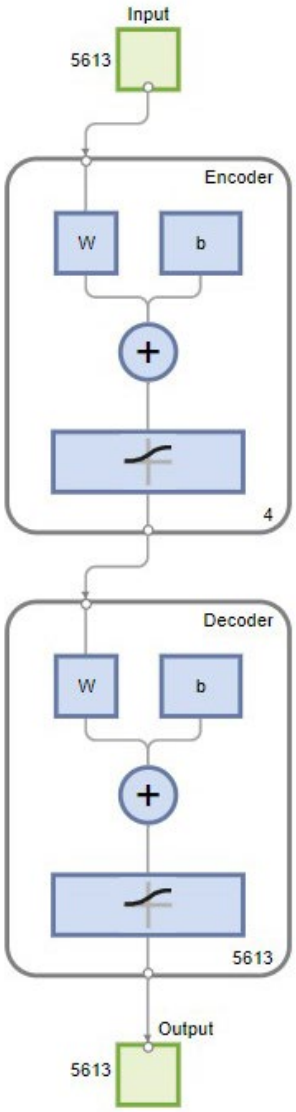


Figure 26. Autoencoder Construct with 4 Hidden Layers.

C. CHAPTER SUMMARY

This chapter reviewed the data processing methods for preparing the data for use in both ML models. The basic autoencoder ML model did not require extensive preparation, as the raw data was utilized for training. Steps included normalizing the data, removing NaN values, and ensuring all samples were 5613 data points in length, resulting in each data set having a 5613 x 5001 size. The LSTM autoencoder model required significant preprocessing steps. Features were extracted from each data set and utilized as the input

data for the LSTM autoencoder model. 13 features were derived utilizing statistical, impulsive, and harmonic characteristics of the data. MATLAB offers the Diagnostic Feature Designer application that can graphically represent features as histograms, enabling visualization of how each feature impacts the representation of the data set. After extracting the features, it was observed that the THD resulted in several infinite calculations, which had to be removed to enable statistical analysis of the data. After removing the infinite values, NaN values and normalizing the data, the resulting data was 14x5001 for each data set. Finally, training parameters and layer construction parameters were set to prepare both ML models for training according to desired parameters.

THIS PAGE INTENTIONALLY LEFT BLANK

V. RESULTS

This chapter presents the results of the data sets as the trained ML models processed them. The basic autoencoder evaluation used two comparison metrics: graphs of the MSE and then statistical analysis values of the MSE. Each data set's MSE was graphed according to its probability density function (PDF) and cumulative distribution function (CDF). This provided a visual interpretation of the data. Statistical analysis of the MSE data sets established numerical thresholds that enabled distinguishing the data sets apart.

Three figures of merit (FOM) evaluated how well the LSTM autoencoder could identify interference in a sample set of signals. The first FOM was a single sample from each dataset evaluated by establishing values for how much error existed between a reconstructed value and the original value for each feature. FOM two was the root mean square error (RMSE) of the data sets and how that compared to the RMSE of the 'Normal' data set. The final FOM was a representative confusion matrix based on the model's success in distinguishing signals with interference from representative normal signals. The first two FOMs required human interpretation for confirmation, while the third FOM provided an automated identification method. Each section will first review the corresponding FOM data, and a discussion on each FOM and its use in answering the research questions will follow.

A. BASIC AUTOENCODER

The data from the basic autoencoder was evaluated in two parts. First, the MSE values derived from the 4,13, and 32 models were graphed according to their PDF and CDF. Then MSE values were evaluated based on the four moments of statistical analysis, measuring the mean, variance, skewness, and kurtosis. The aim of this evaluation was to derive insights from the MSE through numerical comparison. In addition, separations in the PDF and CDF indicated that the statistical properties of these graphical representations could be exploited, indicating the presence of interference in the signal datasets.

1. Probability Density Function Results

In this evaluation, the PDF of the MSE was graphed to assess the reconstruction error of the autoencoder. The PDF represents the probability of an input value falling within a specific range of the input and reconstruction error values. It is a fundamental statistical tool used to analyze and characterize data distribution by quantifying the likelihood of different values occurring [56]. Figures 27 and 28 display the PDF graphs of SNR 20 vs. interference data using Model 4. Figures 29 and 30 display the PDF graphs of the same signal using Model 13, and Figures 31 and 32 display the PDF graphs using Model 32.

a. Model 4

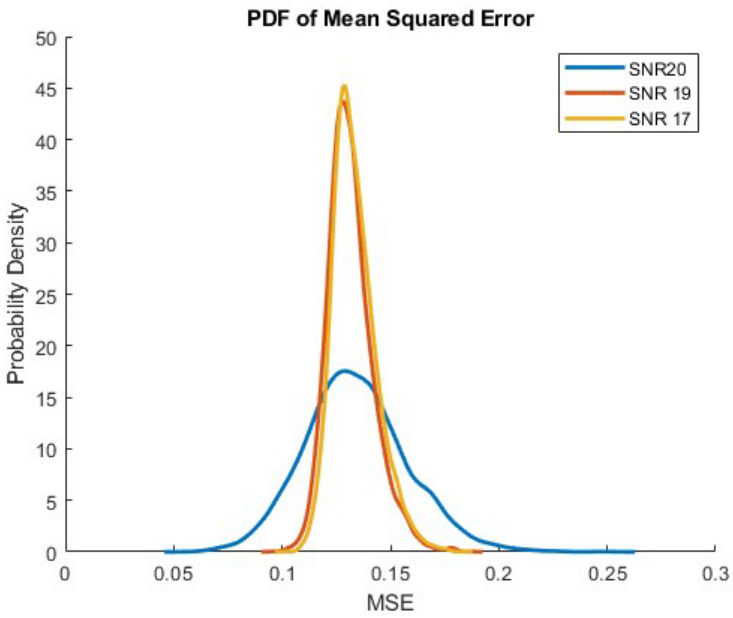


Figure 27. Probability Density Function Graph of SNR 20 and Higher SNR Signals.

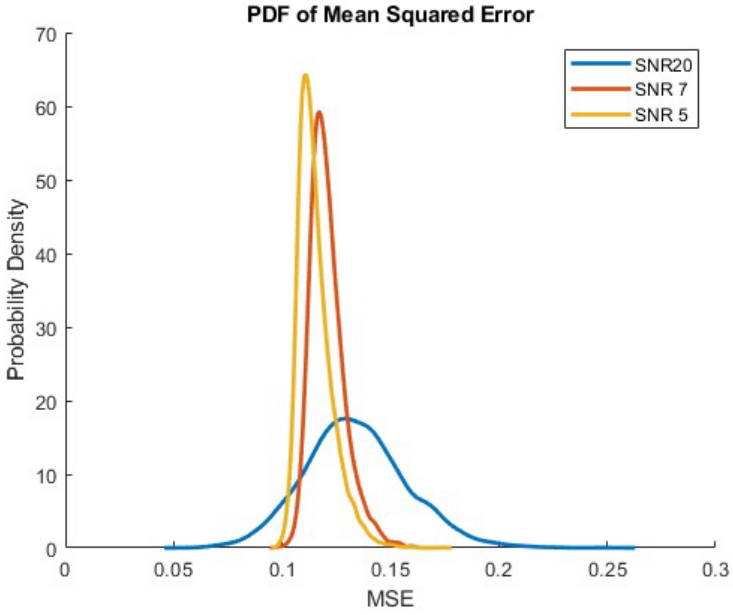


Figure 28. Probability Density Function Graph of SNR 20 and Lower SNR Signals.

b. Model 13

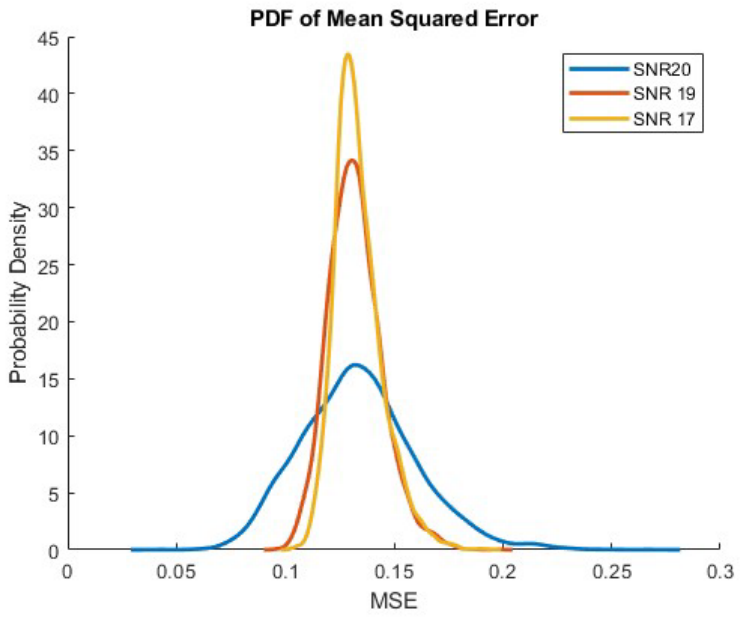


Figure 29. Probability Density Function Graph of SNR 20 and Higher SNR Signals

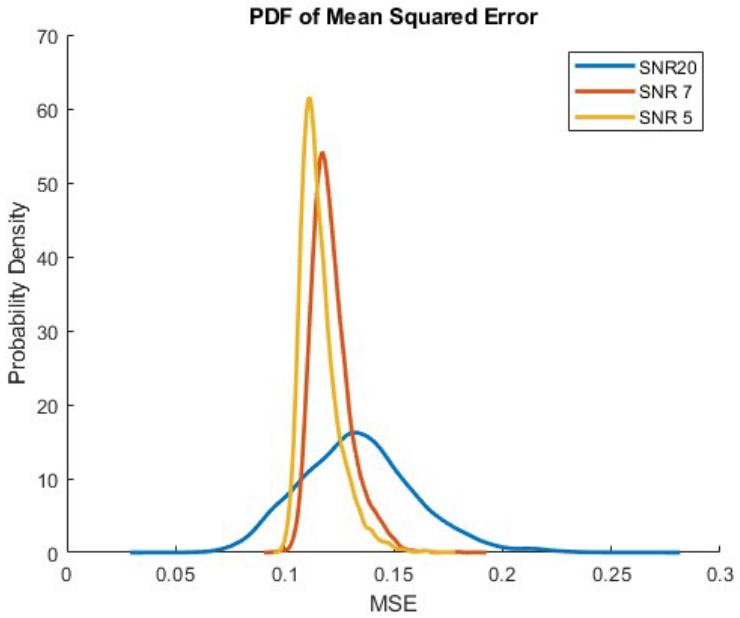


Figure 30. Probability Density Function Graph of SNR 20 and Lower SNR Signals

c. Model 32

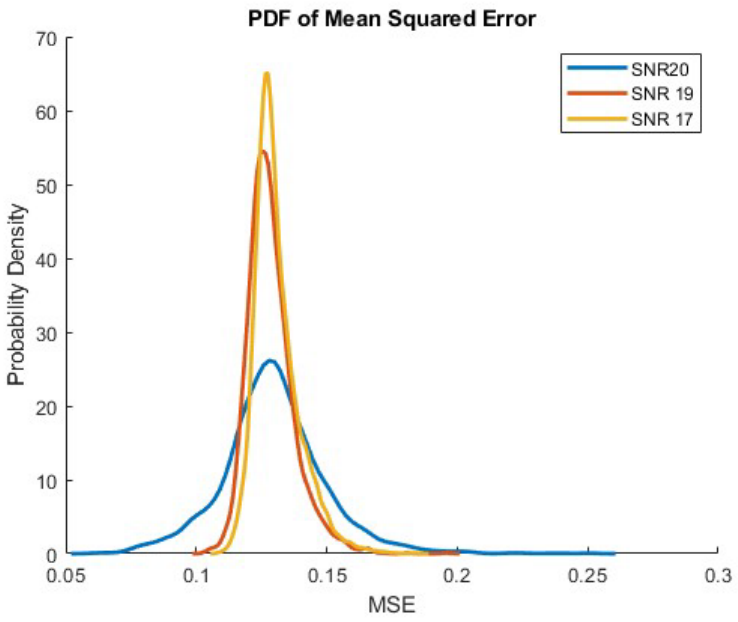


Figure 31. Probability Density Function Graph of SNR 20 and Higher SNR Signals

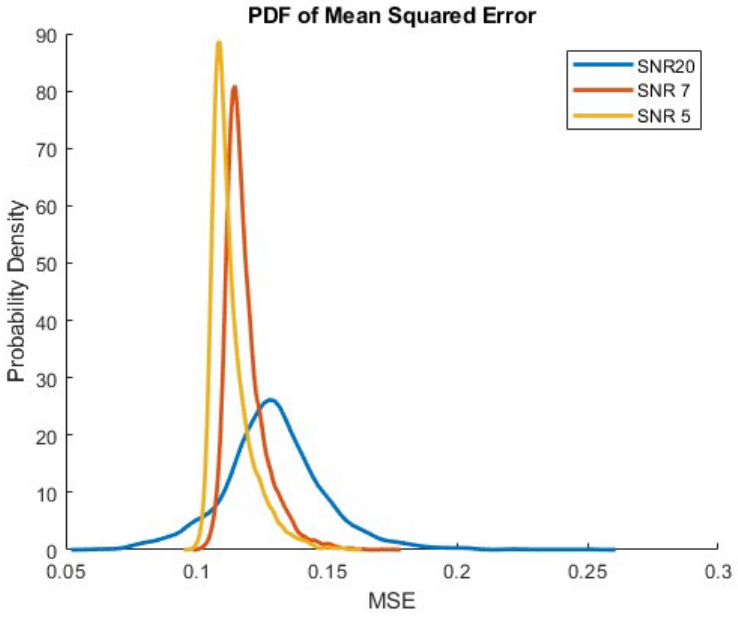


Figure 32. Probability Density Function Graph of SNR 20 and Lower SNR Signals

2. Cumulative Distribution Function Results

The CDF of the MSE gives the cumulative probability that an input value takes on a value less than or equal to a given value. It allows analysis of an entire range of the data set and determines the likelihood that an input value will be below a certain threshold. The CDF is derived from the PDF and is the integration over a range [57]. For CDF, the data sets were placed into two groups to minimize the number of graphs and visualize how similar SNR datasets were distributed as compared to the normal data set. Group 1 included SNR 19, SNR 17, and SNR 15. Group 2 included SNR 13, SNR 11, SNR 9, SNR 7, and SNR 5. Figure 33 and Figure 34 illustrate Model 4 CDF of Group 1 and Group 2. Figure 35 and Figure 36 illustrate the CDF of Group 1 and Group 2 for Model 13. Finally, Figure 37 and Figure 38 depict the CDF of Group 1 and Group 2 for Model 32.

a. Model 4

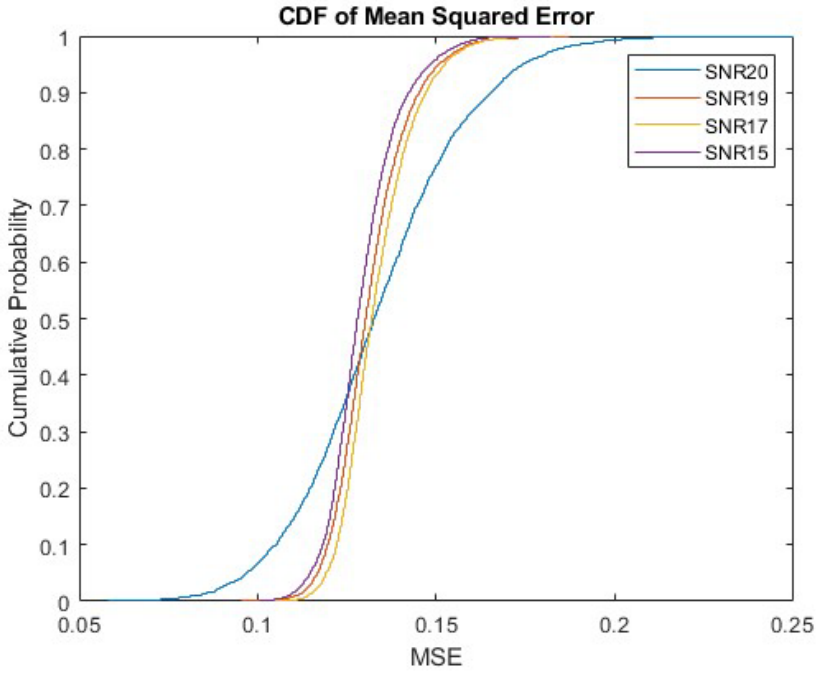


Figure 33. Model 4 Cumulative Distribution Function of Group 1 vs. SNR 20

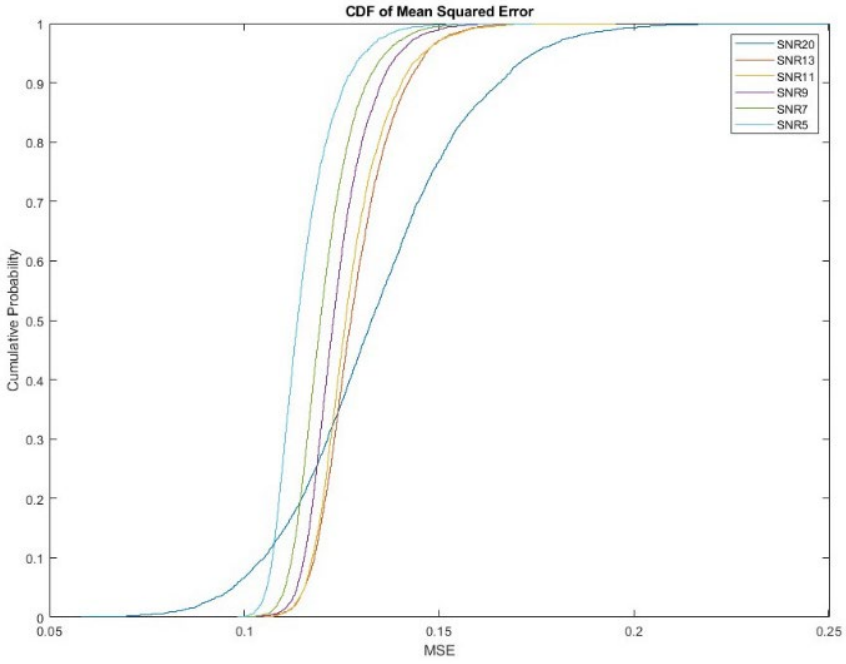


Figure 34. Model 4 Cumulative Distribution Function of Group 2 vs. SNR 20.

b. Model 13

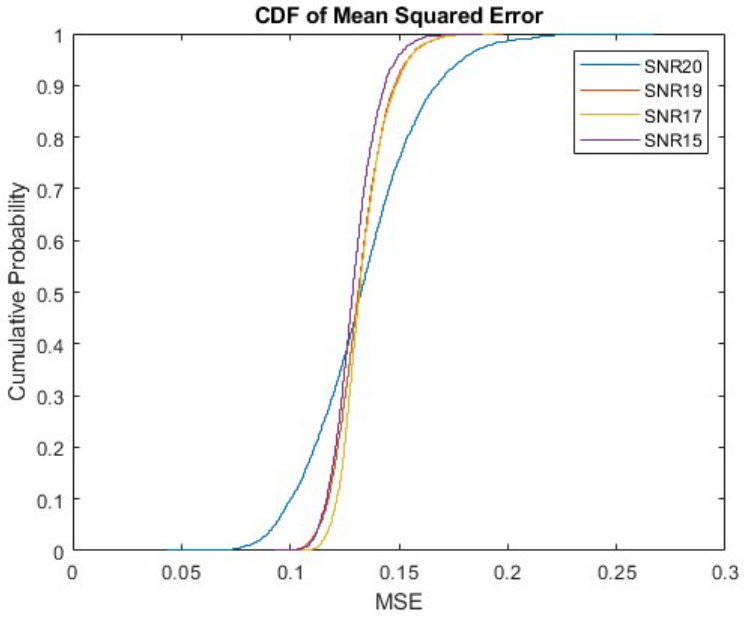


Figure 35. Model 13 Cumulative Distribution Function of Group 1 vs. SNR 20

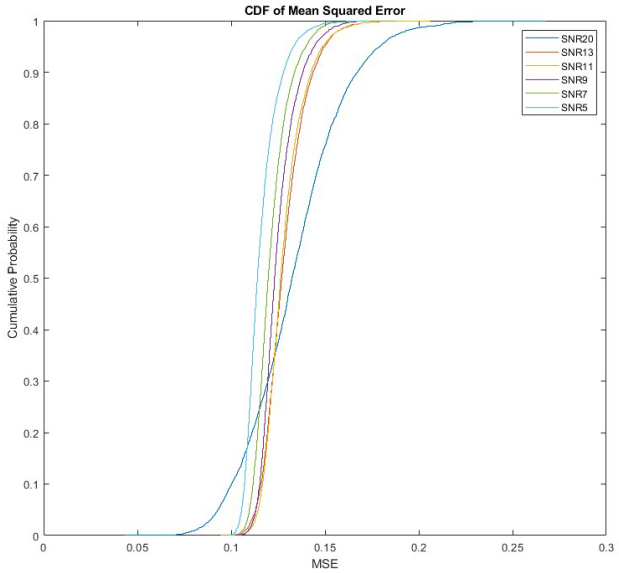


Figure 36. Model 13 Cumulative Distribution Function of Group 2 vs. SNR 20

c. Model 32

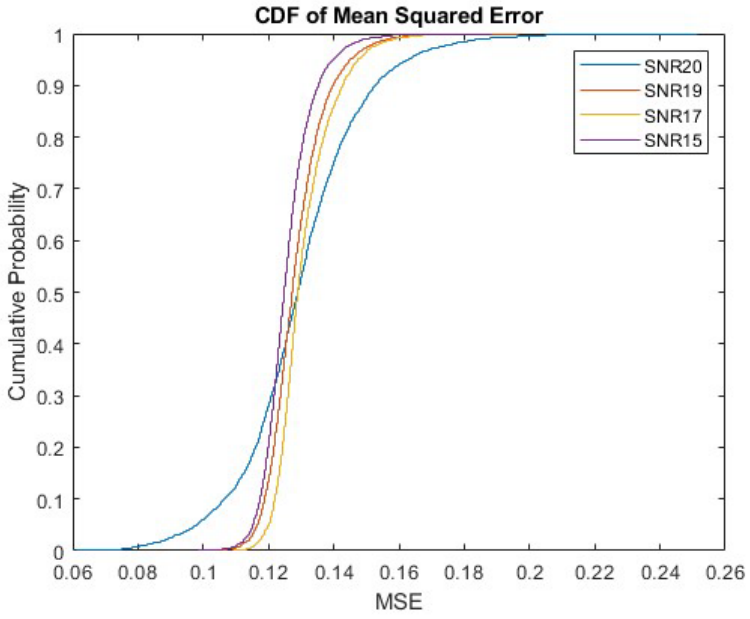


Figure 37. Model 32 Cumulative Distribution Function of Group 1 vs. SNR 20

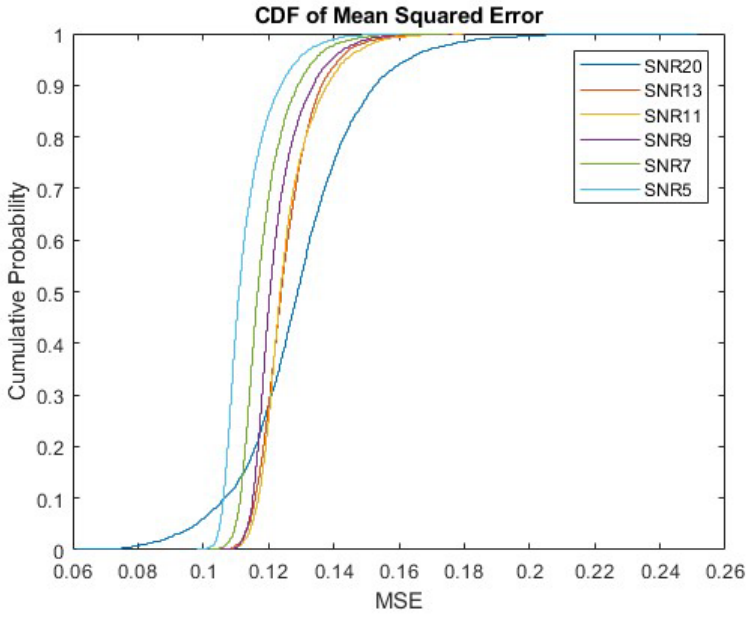


Figure 38. Model 32 Cumulative Distribution Function of Group 2 vs. SNR 20

3. PDF and CDF Discussion

Utilizing both the PDF and CDF graphs to visualize the MSE of the datasets enabled a visual confirmation of existing differences between the normal data set and the data sets with interference.

For PDF, all three models displayed similar graphs for the two groups. Model 4 and Model 13 had similar probability densities for SNR 17 and SNR 19 data sets, peaking slightly above 45%. While Model 32 had a peak of 65% for SNR 17 and above 50% for SNR 19, indicating that the increased hidden layer size of 32 could extract more information from the data set for the ML model. At lower SNR levels, the overall PDF probability percentages were much higher, with Model 4 and Model 13 having a 65% and 60% peak for SNR 5 and SNR 7, respectively. In Model 32, the peaks were much higher, topping out at 90% for SNR 5 and 80% for SNR 7.

Distinguishing between Model 4 and Model 13 overall would be difficult as the PDFs of all three models appeared similar. All three models clearly distinguished between the normal signal and all interference data sets. Data sets with interference had much higher peaks of MSE probability in all three models. Overall utilizing the PDF and CDF enable a basic visual indication of interference. The extent to which the variance or difference exists would need to be examined using the more in-depth four moments statistical analysis approach.

Overall, utilizing the PDF or CDF to visually confirm the presence of interference could prove helpful, as operators only need to plot the output values from a received signal fed through any of the three basic autoencoder models. When aiming to automate the process, further analysis would need to occur. The four moments of statistical analysis can provide this automation capability by establishing numerical thresholds that indicate interference. Utilizing these thresholds as indicators of interference reduces the need for subjective interpretation of visual graphs.

4. Statistical Analysis Results

The formulas for the four moments of statistical analysis are derived from [58]. N is the number of data sets, and σ is the standard deviation. Mean is the average value of the dataset and is represented by

$$\mu = \frac{1}{N}\sum_{j=1}^{N} x_j \quad (15)$$

variance is measuring the spread of the dataset, determining how far each data point is from the mean, and is represented by:

$$Var(x_N) = \frac{1}{N-1}\sum_{j=1}^{N}(x_j - \mu)^2 \quad (16)$$

Skewness is a measure of the symmetry of the distribution of the dataset and is derived by:

$$Skew(x_N) = \frac{1}{N}\sum_{j=1}^{N}\left[\frac{x_j-\mu}{\sigma}\right]^3 \quad (17)$$

Kurtosis is a value that determines how much the data peaks or how flat the data set is in comparison to the normal data distribution. It is derived by:

$$Kurt(x_N) = \left\{\frac{1}{N}\sum_{j=1}^{N}\left[\frac{x_j-\mu}{\sigma}\right]^4\right\} - 3 \quad (18)$$

Tables 6, 7, and 8 display the calculated moments for each data set and the percent change of the values compared to the normal data utilizing Models 4, 13, and 32, respectively.

Table 6. Statistical Moment Values Utilizing Model 4

Data Set	Mean	% Change	Variance	% Change	Skewness	% Change	Kurtosis	% Change
SNR 20	0.134	0	5.6E-04	0	0.320	0	3.431	0
SNR 19	0.132	-1.80%	1.1E-04	-80.21%	0.778	143.51%	4.433	29.19%
SNR 17	0.133	-0.38%	1.0E-04	-81.58%	0.752	135.35%	3.873	12.88%
SNR 15	0.129	-3.46%	1.0E-04	-81.98%	0.718	124.63%	4.151	20.98%
SNR 13	0.129	-3.85%	9.5E-05	-83.04%	0.788	146.46%	4.038	17.68%
SNR 11	0.128	-4.66%	9.3E-05	-83.41%	1.159	262.65%	5.399	57.36%
SNR 9	0.124	-7.25%	6.8E-05	-87.77%	1.015	217.68%	4.772	39.06%
SNR 7	0.121	-9.74%	6.5E-05	-88.34%	1.038	224.89%	4.743	38.23%
SNR 5	0.115	-13.88%	6.3E-05	-88.63%	1.294	304.87%	5.694	65.96%
SNR EMPTY	0.077	-42.82%	2.9E-05	-94.84%	1.713	436.04%	7.622	122.15%

Statistical moments with greatest percentage variance from normal data set highlighted in blue.

Table 7. Statistical Moment Values Utilizing Model 13

Data Set	Mean	% Change	Variance	% Change	Skewness	% Change	Kurtosis	% Change
SNR 20	0.134	0.00%	6.88E-04	0.00%	0.428	0.00%	3.55	0.00%
SNR 19	0.132	-1.12%	1.51E-04	-77.99%	0.537	25.40%	3.70	4.39%
SNR 17	0.134	0.02%	1.23E-04	-82.13%	0.890	107.91%	4.35	22.69%
SNR 15	0.130	-3.02%	1.16E-04	-83.20%	0.610	42.55%	4.08	14.86%
SNR 13	0.129	-3.76%	1.31E-04	-80.95%	0.924	115.96%	4.72	32.89%
SNR 11	0.128	-3.98%	1.20E-04	-82.56%	1.254	193.20%	6.16	73.48%
SNR 9	0.125	-6.32%	9.40E-05	-86.34%	1.227	186.87%	5.27	48.44%
SNR 7	0.122	-9.06%	8.54E-05	-87.59%	1.168	173.01%	5.43	53.12%
SNR 5	0.116	-13.17%	7.94E-05	-88.46%	1.461	241.45%	6.33	78.49%
SNR EMPTY	0.078	-41.81%	4.45E-05	-93.53%	1.695	296.27%	7.10	100.19%

Statistical moments with greatest percentage variance from normal data set highlighted in blue.

Table 8. Statistical Moment Values Utilizing Model 32

Data Set	Mean	% Change	Variance	% Change	Skewness	% Change	Kurtosis	% Change
SNR 20	0.130	0	3.9E-04	0	0.537	0	5.601	0
SNR 19	0.129	-0.90%	8.0E-05	-79.47%	1.102	105.38%	6.003	7.17%
SNR 17	0.131	0.90%	7.5E-05	-80.66%	1.212	125.81%	5.430	-3.07%
SNR 15	0.126	-3.07%	6.0E-05	-84.73%	1.024	90.88%	6.173	10.20%
SNR 13	0.125	-3.33%	7.7E-05	-80.24%	1.275	137.59%	6.190	10.51%
SNR 11	0.126	-3.02%	8.3E-05	-78.72%	1.496	178.74%	6.293	12.35%
SNR 9	0.123	-5.39%	6.6E-05	-83.06%	1.656	208.54%	6.760	20.69%
SNR 7	0.119	-8.49%	6.1E-05	-84.33%	1.652	207.88%	7.140	27.47%
SNR 5	0.113	-12.68%	6.0E-05	-84.66%	1.690	214.97%	6.704	19.69%
SNR EMPTY	0.076	-41.16%	3.8E-05	-90.37%	1.885	251.34%	8.194	46.28%

Statistical moments with greatest percentage variance from normal data set highlighted in blue.

5. Statistical Analysis Discussion

The most significant overall percentage change between the statistical moments and the normal data set in all three models came from the variance and skewness values. The least indicative values came from the data sets' mean and kurtosis values. The regular percent change was utilized in this analysis versus the absolute value of change, as the positivity or negativity of the change also gives insight. Comparing the statistical analysis data, the variance in the negative direction aligns with the PDF graphs showing the slight shift of the peaks to the left, as compared to the Normal data PDF. Additionally, positive skewness, or "right skewed," is distinguished by a shift of the peaks to the left as compared to the mean [59]. The research findings suggest that skewness and variance can be effectively utilized as indicators of a signal with interference.

B. LSTM AUTOENCODER

Two LSTM autoencoder models were constructed. Both LSTM autoencoders had the same configuration as described in Table 9. However, the two ML models varied in hyperparameters. There were two layered constructions, a 32-16-32 layer model and a 13-6-13 layer model. Table 7 includes the layer sizes for the two models and the time each model took to train. Both models had similar validation RMSE, which was derived during the validation portion of the training. For conciseness, the two models will be referred to as the `32-16-32` and the `13-6-13` models, respectively.

Table 9. Layer Parameters for `32-16-32` and `13-6-13` Models

Layer Parameter `32-16-32`		Layer Parameter `13-6-13`	
BiLSTM layer 1	32	BiLSTM layer 1	13
BiLSTM layer 2	16	BiLSTM layer 2	6
BiLSTM layer 3	32	BiLSTM layer 3	13
Time to train model	6 min 56 sec	Time to train model	3 min 13 sec
Validation RMSE	0.029638	Validation RMSE	0.029687

1. First FOM – Single Sample Results

A single sample was selected from each dataset, and the input values for the 13 features were plotted against the output value obtained by running the sample through the trained LSTM autoencoder models. The error was determined by taking the absolute value of the input value minus the "decoded" or output value. The greatest error value was 2, so error values closer to 2 signified the greatest variance. Jammed signals would be expected to have a larger number of features that showed error versus a normal signal who with little to no significant errors. Figure 39 displays SNR 20, SNR 15, and SNR 5 plots for the `13-6-13` model. Table 10 displays the reconstruction error values of each feature for the sample using the `13-6-13` model, with samples appearing in Figure 39 highlighted in blue. Table 11 displays the `32-16-32` model results, with samples appearing in Figure 40 highlighted in blue.

Table 10. Feature Prediction Error Values for the`13-6-13` LSTM Autoencoder Model.

Data set	Feature												
	1	2	3	4	5	6	7	8	9	10	11	12	13
SNR 20	0.005	0.009	0.003	0.003	0.001	0.007	0.015	0.016	0.012	0.002	0.001	0.048	0.041
SNR 19	0.018	0.008	0.016	0.011	0.003	0.074	0.142	0.089	0.130	0.063	0.052	0.137	0.005
SNR 17	0.146	0.212	0.216	0.206	0.794	0.217	0.221	0.059	0.155	0.374	1.266	0.376	0.168
SNR 15	0.084	0.072	0.074	0.304	0.031	0.044	0.260	0.347	0.349	0.078	0.154	0.180	0.430
SNR 13	0.132	0.134	0.093	0.182	0.060	0.177	0.262	0.045	0.102	0.183	0.081	0.605	0.200
SNR 11	0.392	0.307	0.216	0.210	0.991	0.238	0.145	0.018	0.074	0.599	0.761	0.739	0.239
SNR 9	0.436	0.284	0.212	0.154	1.126	0.229	0.224	0.124	0.142	0.604	0.735	0.914	0.258
SNR 7	0.741	0.411	0.366	0.109	1.296	0.351	0.389	0.105	0.627	0.622	0.680	1.122	0.107
SNR 5	0.740	0.320	0.321	0.116	1.232	0.282	0.832	0.031	0.613	0.476	0.688	1.598	0.056
EMPTY	2.873	1.479	2.438	1.844	1.027	0.139	5.254	1.076	1.225	0.611	0.311	5.297	0.033

Select signals featured in Figure 39 highlighted in light blue, features that had highest value of variance highlighted in dark blue.

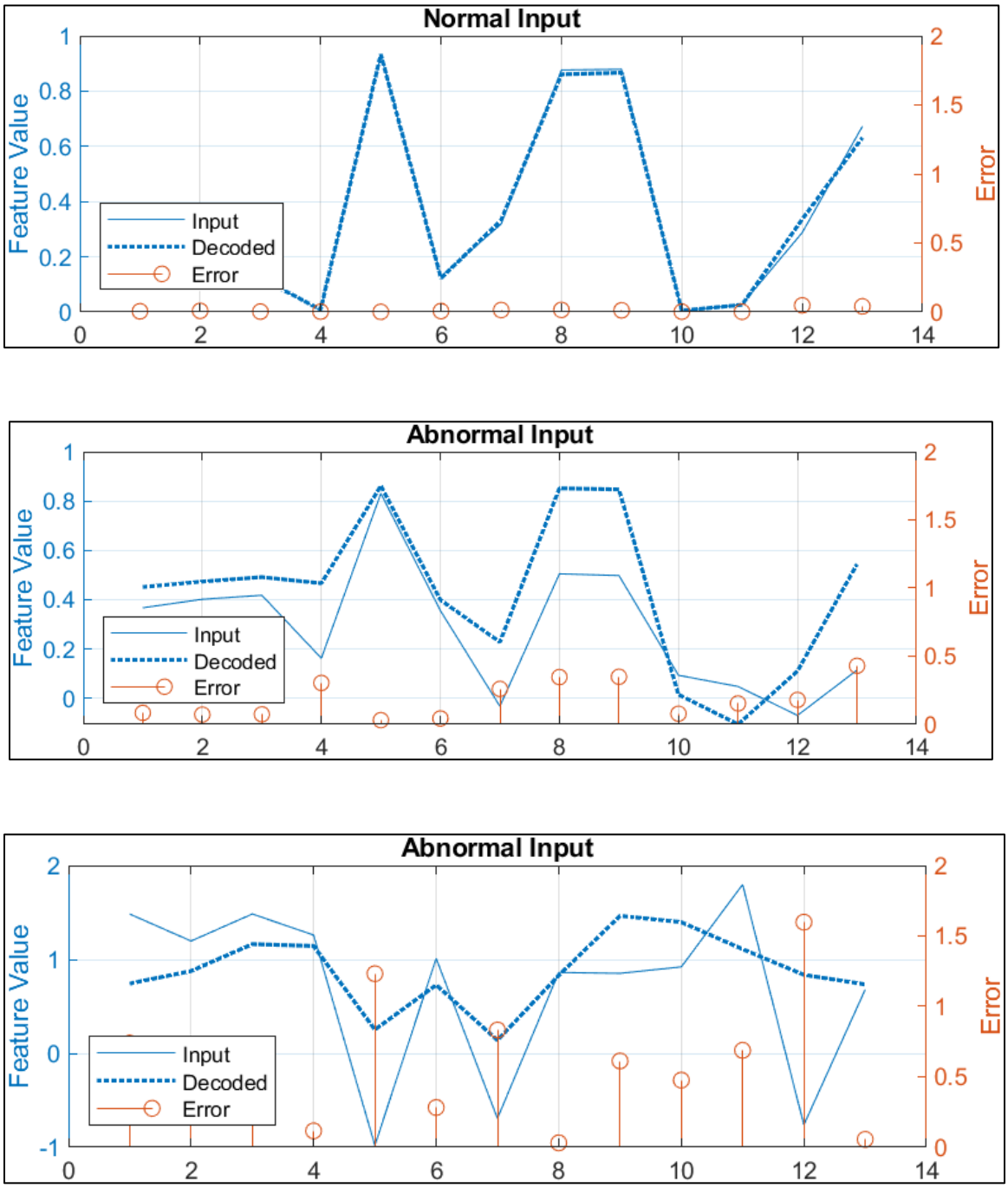


Figure 39. Representative Single Sample Plots for SNR 20 (top), SNR 15 (middle) and SNR 5 (bottom) Using 13-6-13

Table 11. Feature Prediction Error Values `32-16-32` LSTM Autoencoder Model.

Data set	Feature												
	1	2	3	4	5	6	7	8	9	10	11	12	13
SNR 20	0.006	0.007	0.003	0.014	0.007	0.002	0.038	0.011	0.006	0.008	0.001	0.036	0.024
SNR 19	0.006	0.009	0.001	0.011	0.056	0.003	0.210	0.166	0.115	0.033	0.002	0.217	0.015
SNR 17	0.202	0.351	0.370	0.176	1.785	0.093	0.234	0.046	0.094	0.323	1.621	0.397	0.482
SNR 15	0.018	0.001	0.099	0.014	0.069	0.056	0.224	0.184	0.026	0.131	0.324	0.229	0.439
SNR 13	0.038	0.012	0.052	0.026	0.028	0.009	0.176	0.015	0.087	0.129	0.009	0.342	0.218
SNR 11	0.381	0.291	0.475	0.400	1.353	0.156	0.322	0.271	0.415	0.119	1.235	0.861	0.034
SNR 9	0.416	0.257	0.472	0.569	1.438	0.156	0.443	0.408	0.523	0.197	1.229	1.029	0.001
SNR 7	0.463	0.170	0.410	0.585	1.493	0.393	0.809	0.528	0.391	0.434	1.295	1.491	0.020
SNR 5	0.682	0.192	0.484	0.430	1.727	0.442	0.948	0.109	0.211	0.693	1.345	1.556	0.453
EMPTY	3.002	1.255	2.242	1.859	1.271	0.532	5.151	0.108	0.184	1.884	0.807	5.177	0.089

Select signals featured in Figure 40 highlighted in light blue, features that had highest value of variance highlighted in dark blue.

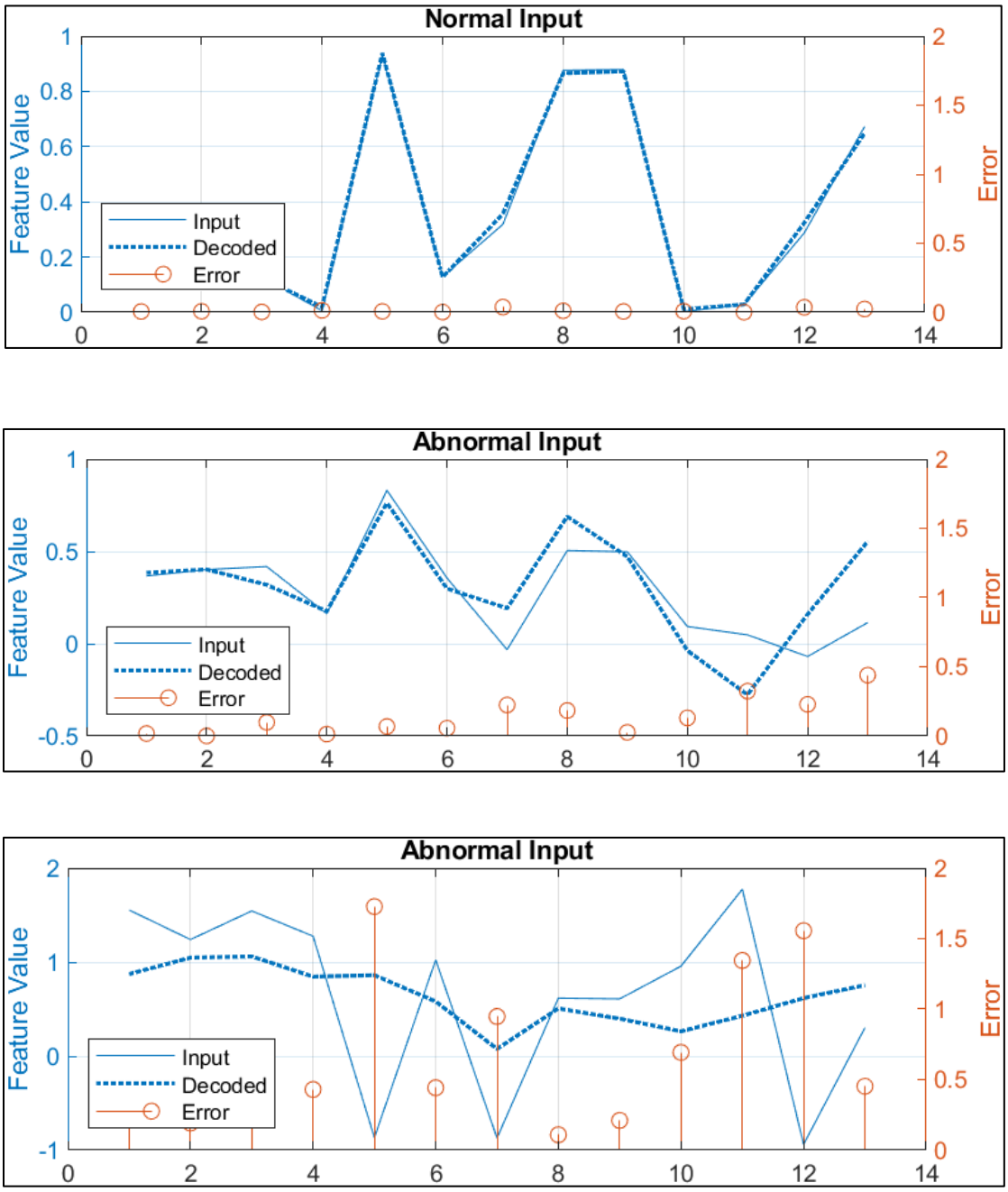


Figure 40. Representative Single Sample Plots for SNR 20 (top), SNR 15 (middle), and SNR 5 (bottom) Using '32-16-32'

2. First FOM – Single Sample Discussion

This FOM quantifies the reconstruction error, which measures the dissimilarity between input and output data after passing through a trained ML model. By providing numerical values, this FOM enables the determination of sample characteristics. Larger reconstruction error values indicate more significant deviations from the input values, indicating abnormal signal characteristics. These values aid in interpreting individual samples and classifying them according to error characteristics.

Creating a database based on this information could facilitate a detailed analysis of how received signals are affected by interference. When deriving feature data from the raw data sets, the features were numbered according to the listing of the features in alphabetical order. Table 12 groups the features and aligns them with the numbering position used in Tables 10 and 11. The Diagnostic Feature Generator's associated ranking is also listed in Table 12.

It is important to note that the reconstruction values of samples selected for the first FOM did not directly correlate to features as the Diagnostic Feature Designer ranked them. Many samples had greater error values in features that had an overall lower ranking of importance which indicates that the neural network processes classified the features differently. Therefore, individual samples cannot represent the entire 5001 sample dataset. This FOM provides detailed information on the extent of variance in a single sample when compared to a normal sample, offering specific characteristics that cannot be inferred from RMSE or the probability of sample anomaly. Utilizing this FOM can contribute to generating a characteristic database or accumulating information over time for previously identified anomalous signals.

Table 12. Feature Rankings and Associated Ranking of Importance.

Feature #	Feature	Feature Ranking
1	Clearance Factor	5
2	Crest Factor	6
3	Impulse Factor	4
4	Kurtosis	3
5	Mean	9
6	Peak Value	7
7	Root Mean Square	11
8	Signal Noise Distortion	2
9	Signal Noise Ratio	1
10	Shape Factor	8
11	Skewness	10
12	Standard Deviation	12
13	Total Harmonic Distortion	13

3. Second FOM – Mean Squared Error Results

The second FOM for evaluating the effectiveness of the trained ML model was to calculate the RMSE across the 13 features used for each dataset. A single RMSE value would then represent the entire dataset. Tables 13 and 14 outlines the calculated RMSE of each data set for each of the ML constructions.

Table 13. `13-6-13` Model RMSE Values

Data set	RMSE
SNR 20	0.08
SNR 19	0.42
SNR 17	1.73
SNR 15	0.78
SNR 13	0.76
SNR 11	1.82
SNR 9	1.99
SNR 7	2.23
SNR 5	2.65
EMPTY	8.91

Table 14. `32-16-32` Model RMSE Values

Data set	RMSE
SNR 20	0.08
SNR 19	0.44
SNR 17	2.62
SNR 15	0.73
SNR 13	0.53
SNR 11	2.52
SNR 9	2.56
SNR 7	2.79
SNR 5	3.06
EMPTY	8.82

The RMSE values were derived by taking the square root of the output vector values of each feature minus the original vector values of each feature in each dataset. The output vector values of each feature were created using MATLAB's predict(x) function[60]. The predict function simply applied the input values to the ML model to complete the output.

4. Second FOM – Mean Squared Error Discussion

In both models, the RMSE rose as the SNR decreased, signifying that the ML models had higher reconstruction error values over the entirety of the data sets as the AWGN strength increased. As a result, human operators could utilize the RMSE as a single identifier for the amount of variance a received signal has, compared to what would be expected, in this case, the values from the SNR 20 dataset. The RMSE value from the lowest level of interference, SNR19, resulted in an RMSE over five times greater than the normal data set. Such a difference in RMSE could be utilized as a threshold metric to classify all signals as interference. This single scalar value does not provide many characteristics for the identified signal as required for reporting; however, it does provide a method for identifying the presence of interference.

There were outliers in FOM two for both models. In the SNR 17 data set for both models, the RMSE value was significantly higher than the similar SNR data sets. Initially, it was believed that the high number of infinity values that were present and replaced in the SNR 17 feature data impacted the RMSE of the entire data set. However, as seen in Table 15, SNR 19 and SNR 5 also had significant infinite values replaced in their data sets. Additionally, there was a significant drop in RMSE between SNR 15 and SNR 13, where only two and 917 infinity values were replaced, respectively. Since the THD of each dataset can be related to the quality of the transmission system [44], it was determined that the quality of the SDR radios and transmission pathways had some impact on the data, but to what extent was not determined. Overall replacing the infinite values with the mean did not directly relate to the spikes or drops in RMSE values.

Table 15. Number of Infinity Values Replaced during Preprocessing

Data set	# replaced
SNR 20	0
SNR 19	4143
SNR 17	3388
SNR 15	2
SNR 13	917
SNR 11	117
SNR 9	80
SNR 7	64
SNR 5	4146
SNR Empty	0

5. Third FOM Test Accuracy and Confusion Matrices Results

FOM three had two associated evaluation criteria: test accuracy and a confusion matrix. The test accuracy was a calculated percentage representing how well the model predicted a signal as having interference or if a signal was a normal signal. The confusion matrix was produced by determining the number of accurate predictions made versus the number of false predictions made. Each SNR data set with interference was evaluated against the 'Normal' data set separately. Signals correctly predicted as having interference would count as a True Normal or True Abnormal, and incorrect predictions counted as False Normal or False Abnormal. Figure 41 illustrates the confusion matrix for SNR 19 with an RMSE threshold of 0.75 using model 32-16-32, and Figure 42 illustrates the results using model 13-6-13.

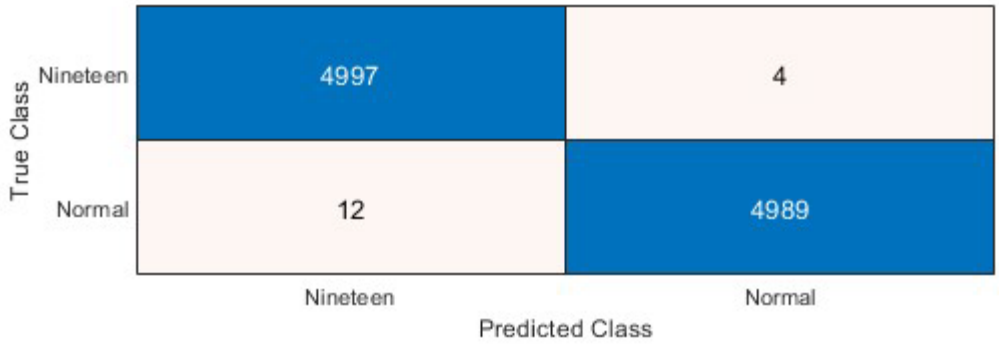


Figure 41. Confusion Matrix for SNR 19 Using a RMSE Threshold of 0.75. Model 32-16-32

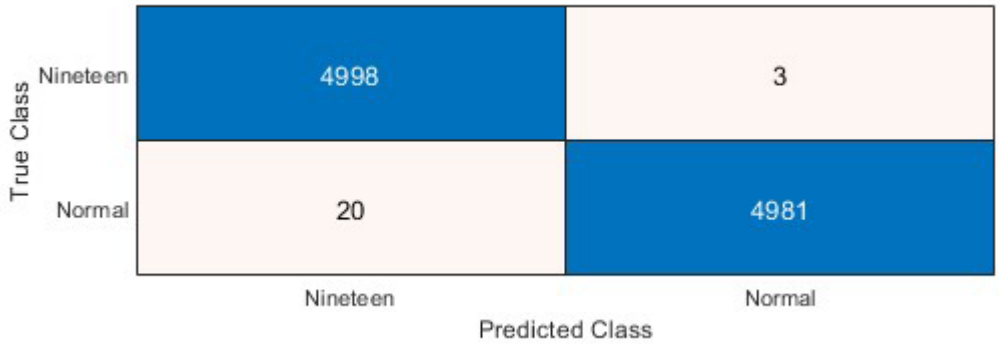


Figure 42. Confusion Matrix for SNR 19 Using a RMSE Threshold of 0.75 Model 13-6-13

The testing accuracy percentage was derived by taking a dataset with interference and combining it with the `Normal` dataset, deemed a testing set, resulting in 10002 total samples. A key value in this evaluation criteria was the threshold value set before evaluating the testing set. The threshold value represented how much of the separation would represent an abnormal signal in the prediction results of the model. If an output value were higher than the set threshold, the output value would result in a predicted abnormal signal. RMSE thresholds ranged from 0.05 to 5.0 representing increasing differences as prediction indicators. Figure 43 depicts the MATLAB script used to calculate test accuracy.

The reconstruction error was plotted for each sample, and anomalies identified with an x were evaluated.

```
>> %graph 3  test accuracy
thresh = 0.5;
XTestAll = helperExtractLabeledData(datacombined, "All");
yHatAll = predict(net, XTestAll);
errorAll = helperCalculateError(XTestAll, yHatAll);
anomalies = errorAll > thresh*mean(errorAll);
```

Figure 43. Graph 3 Testing Accuracy MATLAB Script

In Figure 43, the 'helperExtractLabeledData' embedded function was a modified version of functions used in the MATLAB example "Anomaly Detection in Industrial Machinery Using Three-Axis Vibration Data" [61]. `helperExtractLabeledData` was used to extract all appropriately labeled data from a representative dataset, in this case, the combined dataset was labeled 'data combined.' All data extracted, in this case, consisted of 10002 samples. The 32-16-32 layered model was named 'net' in this script, and the yHatAll variable was the resulting matrix of values passed through the 32-16-32 layered LSTM autoencoder model. The threshold variables were changed, and the values were saved as 'thresh' in the script of Figure 43. Table 16 and Table 17 contain all calculated test accuracy percentage values for each data set, evaluated using each threshold value. Figure 44 plots the SNR 19 test accuracy using a threshold value of 0.75 for model 13-6-13.

Table 16. `13-6-13` Testing Accuracy Percentages and Threshold Values

Data set	Threshold values									
	0.05	0.10	0.25	0.35	0.50	0.75	1.0	2.0	3.0	5.0
SNR 19	50.00	50.29	66.69	82.15	95.77	99.77	99.37	56.19	50.45	50.02
SNR 17	55.90	83.05	99.91	99.91	99.90	99.92	99.93	72.12	50.00	50.00
SNR 15	50.07	54.94	90.32	98.79	99.87	99.89	99.88	60.33	50.04	50.00
SNR 13	50.05	54.51	89.52	98.63	99.88	99.70	95.01	60.01	56.94	50.80
SNR 11	56.9	85.24	99.91	99.91	99.92	99.95	99.96	65.58	50.00	50.00
SNR 9	59.32	88.91	99.91	99.91	99.92	99.95	99.96	67.15	50.00	50.00
SNR 7	63.14	92.98	99.91	99.91	99.93	99.95	99.96	67.88	50.00	50.00
SNR 5	69.89	97.57	99.91	99.92	99.95	99.96	99.98	69.09	50.00	50.00
Empty	99.91	99.92	99.96	100	100	100	100	67.66	50.08	50.04

Highest test accuracy values highlighted in blue.

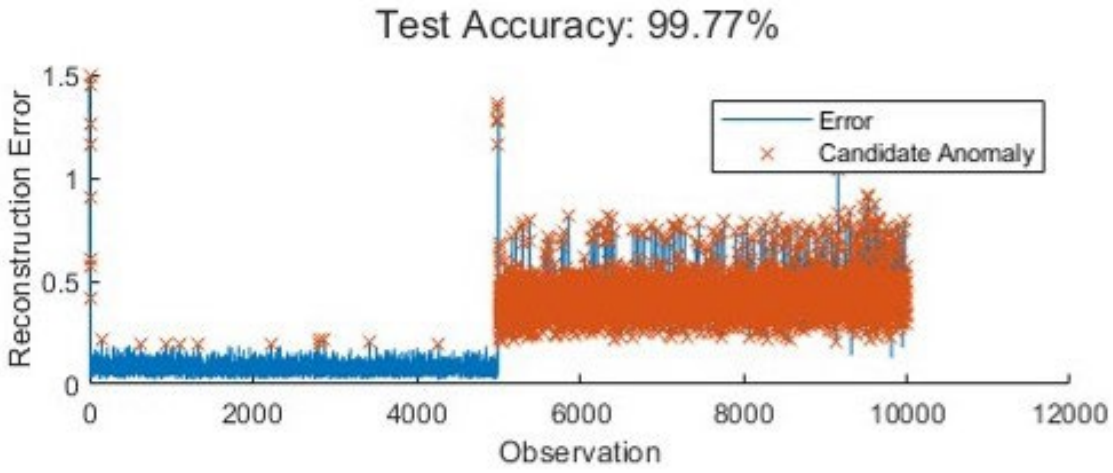


Figure 44. Model 13-6-13 Test Accuracy of SNR 19 Using an RMSE Threshold of 0.75

Figure 45 displays the SNR 19 test accuracy results using an RMSE threshold value of 0.75 for model 32-16-32.

Table 17. `32-16-32` Testing Accuracy Percentages and Threshold Values

Data set	Threshold values									
	0.05	0.10	0.25	0.35	0.50	0.75	1.0	2.0	3.0	5.0
SNR 19	50.00	50.43	71.76	87.04	96.83	99.84	99.30	60.87	50.01	50.02
SNR 17	73.73	97.56	99.91	99.92	99.93	99.93	99.93	59.69	50.00	50.00
SNR 15	50.06	55.18	90.75	98.18	99.86	99.87	99.85	64.58	50.04	50.02
SNR 13	50.00	51.33	80.11	93.03	99.12	96.94	86.03	58.29	57.66	53.61
SNR 11	72.04	96.95	99.91	99.96	99.96	99.96	99.97	68.65	50.00	50.00
SNR 9	72.85	97.22	99.91	99.94	99.96	99.96	99.97	68.74	50.00	50.00
SNR 7	76.75	98.44	99.91	99.95	99.96	99.96	99.97	68.24	50.00	50.00
SNR 5	81.02	99.24	99.92	99.95	99.96	99.96	99.98	68.94	50.00	50.00
Empty	99.90	99.92	99.96	99.98	100	100	100	67.82	50.08	50.04

Highest test accuracy values highlighted in blue.

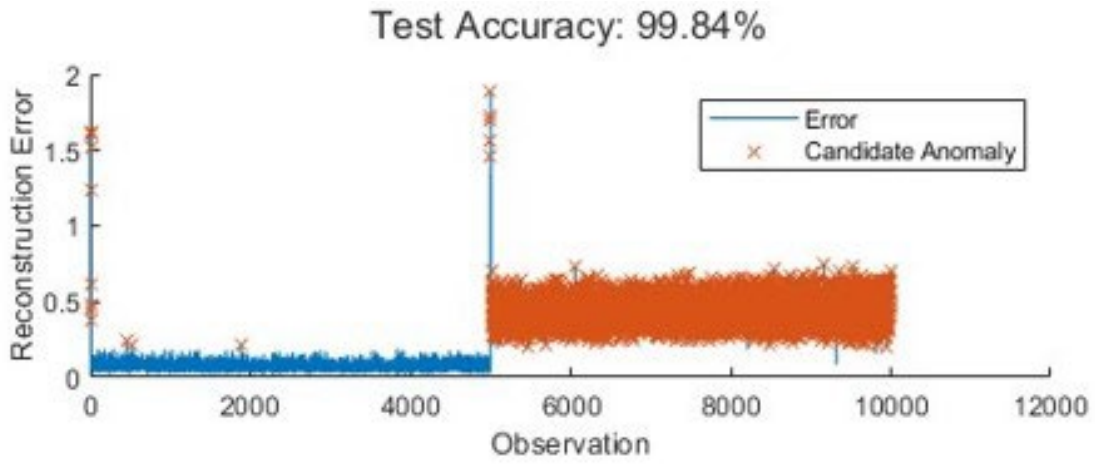


Figure 45. Model 32-16-32 Test Accuracy of SNR 19 Using an RMSE Threshold of 0.75

6. Third FOM Test Accuracy and Confusion Matrices Discussion

This FOM uses RMSE calculation to detect anomalous signal datasets. It classifies a dataset as anomalous or normal by applying a threshold value. Experimental results indicate that thresholds of 0.50, 0.75, and 1.0 yield the highest accuracy rates for correct classification over all data sets. As the SNR decreases and interference becomes more pronounced, higher thresholds result in better rates of successful interference detection.

Results in Table 16 and Table 17 have notable points highlighted in blue. In both models, the highest rates of interference identification were in the threshold ranges of 0.50 to 1.0. The lowest accuracy occurred in the SNR 19 data set, using model 13-6-13, with a test accuracy of 99.77%. Data set SNR 5 had the highest test accuracy at 99.98% in both models. Utilizing the LSTM autoencoder in either model construct provided a useful interference identification method with over 99% accuracy in select thresholds.

C. CHAPTER SUMMARY

Both the basic autoencoder and LSTM autoencoder proved valuable methods for identifying signals with interference compared to those without. The basic autoencoder employed the PDF and CDF of the MSE derived from running the data through the ML model to visually confirm interference. The peaks in the PDF graphs of signals with interference were easily distinguishable, becoming more pronounced as the number of hidden layers increased. The CDF curves also provided a clear separation between the data sets in each model. Statistical analysis, particularly skewness, and variance, served as useful thresholds for an automated identification process.

The LSTM autoencoder was also successful in identifying interference. By extracting features from the data sets and utilizing them as representative data, higher fidelity information was obtained during analysis. The three FOM used in the analysis provided increasing levels of information, with FOM three accurately predicting the presence of interference with a high success rate. The models demonstrated high classification success in different SNR categories. FOM two calculated the RMSE of the data sets, indicating overall variance compared to the expected values of a normal signal. Some outliers were observed, such as the higher RMSE in the SNR 17 data attributed to preprocessing issues. FOM one quantified reconstruction error for each of the 13 features, providing specific details and characteristics of single signal samples, which could be leveraged for developing high-fidelity data on abnormal signals.

VI. CONCLUSION

The DOD has established doctrine and strategy to ensure that operations with a contested and increasingly congested EMOE are secured and ensured. The electromagnetic superiority strategy dictates that efforts be made to ensure that DOD assets can maintain spectrum operations and C2. The U.S. Navy is increasingly vulnerable to the EMI threat of a contested and congested EMSO, and with the existing gap in interference identification capabilities, it must ensure that interference identification and mitigation become a priority.

This thesis sought to determine if the data analysis capabilities of ML autoencoders could be leveraged to fill the capabilities gap in interference identification and EMI characterization. For this research, RF data was generated locally and used to train and evaluate the ML models. RF signals with increasing levels of interference were generated, collected, and then subjected to various levels of processing for utilization in either of the ML models.

A basic autoencoder model was trained on raw I&Q data collected in the RF generation phase, and data sets with interference were run through the trained model. Plotting the PDF and CDF of the MSE from the outputs of the basic autoencoders resulted in a visual indication of signals with interference and without. Further statistical analysis of the MSE outputs resulted in two key features of the signal analysis pointing to potential indicator thresholds for interference identification. Utilizing thresholds in skewness and variance values provided the potential for setting an autonomous method for identifying interference.

An LSTM autoencoder model's capability for interference detection was evaluated on three FOM. The organization of these FOM provided increasingly detailed information, which ultimately offers a viable solution to the primary research question of whether leveraging locally hosted ML algorithms can enhance the U.S. Navy's satellite EMI, identification, and mitigation efforts. FOM 3, test accuracy, demonstrated a 99.74% success rate of the LSTM autoencoder model in identifying interference in the data sets with even the lowest interference levels, peaking at 99.98% in signals with the highest

interference levels. FOM 1 provided a method for characterizing the details of a single signal and evaluating the variance of a received signal from expected values. This FOM could provide a solution to the secondary research question of identifying the details that can be extracted from analyzed signals as anomalous. By characterizing interference signals based on the variance of features in the output versus input data, a database-building process could be created, further enhancing the fidelity of information about an EMSO.

In this research, the results of the basic autoencoder and LSTM autoencoder both provided a useful method for interference identification. Both ML models enabled a visual indication utilizing graphical representations of the output data. Additionally, both ML models enabled the utilization of numerical thresholds in produced data that could serve in a proposed automated interference identification process, where a received signal exceeding a threshold of variance would be accurately classified as interference.

A. FUTURE WORK

Based on the research findings of this thesis, there are several key areas for future work in this area. Evaluating the ML models used in this thesis against various interference types is one area of future work. Interference placed within the center frequency of transmissions, meaconing detection, and identifying counterfeit transmitters would enable the evaluation of ML techniques against other types of EMI. Building ML models with increased interference detection capabilities has the potential to improve ML applications for DOD problem sets.

Additionally, investigating the deployment of trained machine learning models on various edge devices or low-form factor devices could determine the balance between versatility and identification capabilities. This evaluation could leverage cloud computing services and various edge devices as a testing platform while also assessing the potential for building a cloud database of interference characteristics applicable to different EMSOEs. An RF signal database could provide valuable insights for operators by offering an expanded reach of important signal data and improving spectrum managers' capabilities to understand various EMOEs better.

Various ML techniques' effectiveness in RF signal data analysis should also be explored. Techniques such as transfer learning or Generative Adversarial Networks offer distinct advantages. GANs should be explored to subvert the ML interference identification techniques identified in this thesis research. This assessment would help develop improved neural network models that mimic adversaries' responses to the interference mitigation efforts and attempt to evade detection during RF analysis. Additionally, transfer learning techniques offer a reduced learning time; they could serve as a potential use for a cloud database of RF signal characteristics to improve an operational unit's ability to adapt to various EMI sources.

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF REFERENCES

[1] Department of Defense, "Electromagnetic spectrum superiority strategy," Washington, DC, USA, 2020. Available: https://media.defense.gov/2020/Oct/29/2002525927/-1/-1/0/electromagnetic_spectrum_superiority_strategy.pdf

[2] A. Nikitha, J. Geetha and D. S. JayaLakshmi, "Handwritten text recognition using deep learning," *2020 International Conference on Recent Trends on Electronics, Information, Communication & Technology (RTEICT)*, Bangalore, India, 2020, pp. 388–392, doi: 10.1109/rteict49044.2020.9315679

[3] P. Lakkhanawannakun and C. Noyunsan, "Speech recognition using deep learning," *2019 34th International Technical Conference on Circuits/Systems, Computers and Communications (ITC-CSCC)*, JeJu, Korea (South), 2019, pp. 1–4, doi: 10.1109/itc-cscc.2019.8793338

[4] B. Omarov and Y. I. Cho, "Machine learning based pattern recognition and classification framework development," *2017 17th International Conference on Control, Automation and Systems (ICCAS)*, Jeju, Korea (South), 2017, pp. 1–5, doi: 10.23919/iccas.2017.8204282

[5] M. Johnson, "Real time spectrum operations," in *National Spectrum Management Association Annual Spectrum Management Conference*, Arlington, VA, May 2019. Available: https://www.nsma.org/wp-content/uploads/2019/05/real-time-spectrum-operations.pdf

[6] P. Henarejos, M. Á. Vázquez, and A. I. Pérez-Neira, "Deep learning for experimental hybrid terrestrial and satellite interference management," in *2019 IEEE 20th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, Jul. 2019, pp. 1–5. doi: 10.1109/SPAWC.2019.8815532

[7] D. Roy, "Machine learning based RF transmitter characterization in the presence of adversaries," Ph.D. dissertation, Dept. of Comp. Sci, University of Central Florida, Orlando, FL, USA, 2020. Available: https://purls.library.ucf.edu/go/DP0023859

[8] T. A. Hall, R. Caromi, M. Souryal, and A. Wunderlich, "Reference datasets for training and evaluating RF signal detection and classification models," in *2019 IEEE Globecom Workshops (GC Wkshps)*, Waikoloa, HI, USA: IEEE, Dec. 2019, pp. 1–5. doi: 10.1109/GCWkshps45667.2019.9024532. Available: https://ieeexplore.ieee.org/document/9024532/

[9] F. Zhuang *et al.*, "A comprehensive survey on transfer learning." arXiv, Jun. 23, 2020. Available: http://arxiv.org/abs/1911.02685

[10] *Joint Electromagnetic Spectrum Management Operations in the Electromagnetic Operational Environment,* CJCSM 3320.01C, Joint Chiefs of Staff. In CJCSM 3320.01C. Washington, DC, USA, 2019. Available: https://www.jcs.mil/Portals/36/Documents/Library/Manuals/CJCSM%203320.01C.pdf?ver=-F25WY4LZ1IDYK9OB0VQ2W%3d%3d

[11] B. Myers, "Electromagnetic spectrum – Introduction," *Imagine the Universe The Electromagnetic Spectrum*, NASA. March 2013 [Online]. Available: https://imagine.gsfc.nasa.gov/science/toolbox/emspectrum1.html

[12] Federal Communications Commission, "Radio spectrum allocation." Accessed: May 04, 2023 [Online]. Available: https://www.fcc.gov/engineering-technology/policy-and-rules-division/general/radio-spectrum-allocation

[13] TeraSense, "Radio frequency bands." Accessed: Apr. 28, 2023 [Online]. Available: https://terasense.com/terahertz-technology/radio-frequency-bands/

[14] *Joint Campaigns and Operations,* JP-3-0,Joint Chiefs of Staff. Washington, DC, USA, 2022. Available: https://jdeis.js.mil/jdeis/new_pubs/jp3_0.pdf

[15] J. R. Hoehn, K. M. Sayler, and J. C. Gallagher, "Overview of department of defense use of the electromagnetic spectrum," Congressional Research Service, Washington, DC, USA, CRS Report No. R46564, 2021. Available: https://crsreports.congress.gov/product/pdf/R/R46564/13

[16] *Joint Electromagnetic Spectrum Operations,* JP-3-85,Joint Chiefs of Staff. Washington, DC, USA, 2020. Available: https://www.jcs.mil/Portals/36/Documents/Doctrine/pubs/jp3_85.pdf

[17] *Joint Spectrum Interference Resolution Procedures,* CJCSM 3320.02E, Joint Chiefs of Staff. in CJCSM 3320.02E. Washington, DC, USA, 2022. Available: https://www.jcs.mil/Portals/36/Documents/Library/Manuals/cjcsm%203320.02E.pdf?ver=mjhxbfch67vq1qf8tcucya%3d%3d

[18] *Naval Telecommunications Procedures (NTP) 6F Navy Electromagnetic Spectrum (EMS) Guide,*Naval Information Warfighting Development Center. in *Naval Telecommunication Procedures*, no. 6F. Norfolk, VA, USA, 2018.

[19] K. Youssef, L.-S. Bouchard, K. Z. Haigh, H. Krovi, J. Silovsky, and C. P. V. Valk, "Machine Learning Approach to RF Transmitter Identification." arXiv, Nov. 07, 2017. Available: http://arxiv.org/abs/1711.01559

[20] The Fu Foundation School of Engineering and Applied Science, "Artificial intelligence (AI) vs. machine learning." Accessed: May 04, 2023 [Online]. Available: https://ai.engineering.columbia.edu/ai-vs-machine-learning/

[21] Y. Gavrilova, "Artificial intelligence vs. machine learning vs. deep learning: essentials," *Serokell Software Development Company*, Apr. 08, 2020. Available: https://serokell.io/blog/ai-ml-dl-difference

[22] Mathworks, *Machine Learning with MATLAB*. 2022. Available: https://www.mathworks.com/content/dam/mathworks/ebook/gated/machine-learning-ebook-all-chapters.pdf

[23] O. Simeone, "A brief introduction to machine learning for engineers." arXiv, May 17, 2018. Available: http://arxiv.org/abs/1709.02840

[24] J. Zhang *et al.*, "Fully automated echocardiogram interpretation in clinical practice: feasibility and diagnostic accuracy," *Circulation*, vol. 138, no. 16, pp. 1623–1635, Oct. 2018, doi: 10.1161/circulationaha.118.034338

[25] P. Baheti, "The essential guide to neural network architectures," *v7labs*, Jul. 08, 2021 [Online]. Available: https://www.v7labs.com/blog/neural-network-architectures-guide

[26] E. Kavlakoglu, "AI vs. machine learning vs. deep learning vs. neural networks: what's the difference?," *IBM Cloud*, Jan. 19, 2022. Available: https://www.ibm.com/cloud/blog/ai-vs-machine-learning-vs-deep-learning-vs-neural-networks

[27] S. K. Dasaradh, "A gentle introduction to math behind neural networks," *Towards Data Science*, Oct. 2020. Available: https://towardsdatascience.com/introduction-to-math-behind-neural-networks-e8b60dbbdeba

[28] N. Barla, "Understanding representation learning with autoencoder," *neptune.ai*, Jul. 21, 2022. Available: https://neptune.ai/blog/representation-learning-with-autoencoder

[29] S. Zavrak and M. Iskefiyeli, "Anomaly-Based intrusion detection from network flow features using variational autoencoder," *IEEE Access*, vol. 8, pp. 108346–108358, 2020, doi: 10.1109/ACCESS.2020.3001350

[30] "Unsupervised learning: autoencoders," class notes for Machine Learning for Autonomous Operations, Naval Postgraduate School, Monterey CA, USA, winter 2023.

[31] C. Olah, "Understanding LSTM networks," colah's blog. August 2015 [Online]. Available: https://colah.github.io/posts/2015-08-Understanding-LSTMs/

[32] P. Mobtahej, X. Zhang, M. Hamidi, and J. Zhang, "An LSTM-Autoencoder architecture for anomaly detection applied on compressors audio data," *Comput. Math. Methods*, vol. 2022, pp. 1–22, Sep. 2022, doi: 10.1155/2022/3622426

[33] R. Getz, "ADALM-PLUTO overview," Analog Devices Wiki, 2021 [Online]. Available: https://wiki.analog.com/university/tools/pluto

[34] Jinchang Electron, *GSM Antenna JCG401*. Accessed: May 1,2023 [Online]. Available: https://wiki.analog.com/_media/university/tools/pluto/users/jcg401.pdf

[35] D. Koch, "ADI SDR transceivers enable amateur space communication," *AnalogDialogue*, vol. 54, no. 1, Jan. 2020, [Online]. Available: https://www.analog.com/media/en/analog-dialogue/volume-54/number-1/adi-sdr-transceivers-enable-amateur-space-communication.pdf

[36] MATLAB, "QPSK receiver with ADALM-PLUTO radio in Simulink." Accessed: Apr. 23, 2023 [Online]. Available: https://www.mathworks.com/help/supportpkg/plutoradio/ug/qpsk-receiver-with-adalm-pluto-radio-." 1.html#Qpskreceiverwithadalmplutoradiosimulinkexample-16

[37] Analog Devices, *AD9363 Data Sheet*. 2016 [Online]. Available: https://www.analog.com/media/en/technical-documentation/data-sheets/ad9363.pdf

[38] MATLAB, "QPSK transmitter with ADALM-PLUTO radio in Simulink." Accessed: Apr. 23, 2023 [Online]. Available: https://www.mathworks.com/help/supportpkg/plutoradio/ug/qpsk-transmitter-with-adalm-pluto-radio-1.html

[39] M. Viswanathan, "QPSK modulation & demodulation," Gaussian Waves. Accessed: May 05, 2023 [Online]. Available: https://www.gaussianwaves.com/2010/10/qpsk-modulation-and-demodulation-2/#:~:text=Quadrature%20Phase%20Shift%20Keying%20(QPSK,possible%20carrier%20phase%20shift%20states

[40] "Best public datasets for machine learning and data science | AltexSoft," 2019 [Online]. Available: https://www.altexsoft.com/blog/datascience/best-public-machine-learning-datasets/

[41] MATLAB, "QPSK receiver with ADALM-PLUTO radio." Accessed: April 23, 2023 [Online]. Available: https://www.mathworks.com/help/supportpkg/plutoradio/ug/qpsk-receiver-with-adalm-pluto-radio.html

[42] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016 [Online]. Available: https://www.deeplearningbook.org/

[43] MATLAB, "Signal features." Accessed: April 23, 2023 [Online]. Available: https://www.mathworks.com/help/predmaint/ug/signal-features.html

[44] W. Kester, "Understand SINAD, ENOB, SNR, THD, THD+N, and SFDN so you don't get lost in the noise floor," Analog Devices Tutorials, 2009 Accessed: April 30, 2023 [Online]. Available: https://www.analog.com/media/en/training-seminars/tutorials/MT-003.pdf

[45] MATLAB, "Interactively extract, visualize, and rank features from measured or simulated data for machine diagnostics and prognostics." Accessed: April 23, 2023 [Online]. Available: https://www.mathworks.com/help/predmaint/ref/diagnosticfeaturedesigner-app.html

[46] MATLAB, "Import and visualize ensemble data in diagnostic feature designer." Accessed: April 23, 2023 [Online]. Available: https://www.mathworks.com/help/predmaint/gs/import-and-visualize-data-in-diagnostic-feature-designer.html [Accessed: Apr. 23, 2023]

[47] Google Developers, "Normalization" Foundational courses. Accessed: May 04, 2023 [Online]. Available: https://developers.google.com/machine-learning/data-prep/transform/normalization

[48] GeeksforGeeks, "Data normalization in data mining," Jun. 13, 2019 [Online]. Available: https://www.geeksforgeeks.org/data-normalization-in-data-mining/

[49] S. Singh, "Activation functions for deep neural networks," Knowledgehut, Jan. 19, 2023 [Online]. Available: https://www.knowledgehut.com/blog/data-science/activation-functions-in-neural-networks

[50] D. P. Kingma and J. Ba, "Adam: a method for stochastic optimization." arXiv, Jan. 29, 2017 [Online]. Available: http://arxiv.org/abs/1412.6980

[51] MATLAB, "Training options for Adam optimizer." Accessed: May 04, 2023 [Online]. Available: https://www.mathworks.com/help/deeplearning/ref/nnet.cnn.trainingoptionsadam.html?s_tid=doc_ta

[52] MATLAB, "Train deep learning neural network." Accessed: May 04, 2023 [Online]. Available: https://www.mathworks.com/help/deeplearning/ref/trainnetwork.html?s_tid=doc_ta

[53] baeldung, "Training and Validation Loss in Deep Learning," *Baeldung on Computer Science*, Feb. 19, 2022 [Online]. Available: https://www.baeldung.com/cs/training-validation-loss-deep-learning

[54] M. F. Møller, "A scaled conjugate gradient algorithm for fast supervised learning" *Neural Netw.*, vol. 6, no. 4, pp. 525–533, Jan. 1993, doi: 10.1016/S0893-6080(05)80056-5

[55] MATLAB, "Train an autoencoder." Accessed: May 04, 2023 [Online]. Available: https://www.mathworks.com/help/deeplearning/ref/trainautoencoder.html#buyr01b-1

[56] "Probability density function (PDF)," *BYJU'S*. Accessed: May 4, 2023 [Online]. Available: https://byjus.com/maths/probability-density-function/

[57] V. Singh, "PDF vs. CDF: difference between PDF and CDF," *Shiksha Online*, May 2023 [Online]. Available: https://www.shiksha.com/online-courses/articles/difference-between-pdf-and-cdf/

[58] W. H. Press, Ed., *Numerical Recipes in FORTRAN: The Art of Scientific Computing*, 2nd ed. Cambridge [England] ; New York: Cambridge University Press, 1996,pp. 604–609

[59] A. Kujawska, "Statistical moments in data science interviews" *Towards Data Science*, Jun. 2021 [Online]. Available: https://towardsdatascience.com/statistical-moments-in-data-science-interviews-bfecd207843d

[60] MATLAB, "Predict responses of linear regression model." Accessed: May 23, 2023 [Online]. Available: https://www.mathworks.com/help/stats/linearmodel.predict.html

[61] MATLAB, "Anomaly detection in industrial machinery using three-axis vibration data." Accessed: April 23, 2023 [Online]. Available: https://www.mathworks.com/help/predmaint/ug/anomaly-detection-using-3-axis-vibration-data.html

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California