Deep Learning Based Automatic Modulation Classification

Group Members

AHMAD MUJTABA	**191203**
MUHAMMAD TALHA	**191227**
MUHAMMAD ALI TAHIR	**191241**

Bachelor of Computer Engineering (2019-2023)

Project Supervisor

DR MUMAJJED MUDASSIR

ASSISTANT PROFESSOR

Co Supervisor

MUHAMMAD USAMA ZAHID

MANAGER NESCOM

DEPARTMENT OF COMPUTER ENGINEERING

FACULTY OF ENGINEERING

AIR UNIVERSITY, ISLAMABAD

A Final Year Project

presented to

AIR UNIVERSITY

in partial fulfilment of the requirements

for the degree of

Bachelor of Computer Engineering

Deep Learning Based Automatic Modulation Classification

SUBMITTED BY:

AHMAD MUJTABA	**191203**
MUHAMMAD TALHA	**191227**
MUHAMMAD ALI TAHIR	**191241**

(2019-2023)

DEPARTMENT OF COMPUTER ENGINEERING

May ,2023

Approval for Submission

it is to certified that project report titled

"DEEP LEARNING BASED AUTOMATIC MODULATION CLASSIFICATION"

has met the required standard for submission in partial fulfillment

of the requirements for the award of degree of Bachelor of

Computer Engineering at Air University, Islamabad

Project Supervisor

Dr Mumajjed Mudassir

(Assistant Professor)

Head of Department

Dr. Hafiz Ashiq

Acknowledgement

We would like to offer our heartfelt gratitude and appreciation to everyone who helped and contributed to the successful completion of my Final Year Project (FYP). Their great advice, aid, and encouragement have been vital in developing our endeavor. First and foremost, we would like to extend my heartfelt appreciation to my supervisor, Dr. Mumajjed, and my co-supervisor Muhammad Usama Zahid, for their unwavering support, expert knowledge, and continuous guidance throughout this project. Their valuable insights, constructive feedback, and dedication have been invaluable in shaping the direction and quality of this research. Finally, we would want to thank all of the researchers, writers, and academics whose work we have cited in this effort. Their contributions to the area established the groundwork for our study and were critical in molding my knowledge of the subject.

Abstract

In wireless communication systems, automatic modulation classification (AMC) involves identifying the modulation scheme being used by a radio frequency signal. Modulation techniques vary in characteristics such as amplitude, frequency, and phase. Methods for AMC traditionally rely on feature engineering and classical machine learning algorithms, which fail to capture complex patterns. This research provides two new deep learning-based architectures for classifying different modulation techniques. These deep learning architectures have been tested against previous neural networks and show better performance on real-world datasets, even at low SNR.

Nomenclature

AM Amplitude Modulation

AMC Automatic Modulation Classification

CNN Convolutional Neural Network

DL Deep Learning

FB Likelihood-based methods

FM Frequency Modulation

LB Feature-based methods

$LSTM$ Long Short Term Memory

ML Machine Learning

PSK Phase Shift Key

PSK Phase Shift Keying

RNN Recurrent Neural Network

Table of Contents

List of Figures

Chapter 1
INTRODUCTION

Wireless technology is getting better and there are different ways to send information. Automatic Modulation Classification (AMC) is a crucial step in detecting and decoding signals, but it's getting harder to do. It is very important in lots of every day and army-related situations, like checking for radio problems and deciding who can use certain frequencies of communication such as explained in [1]. Identifying the modulation scheme on a received signal is a fundamental task in wireless communication systems. Digital data is encoded onto a carrier signal through modulation. In the military, we need to make sure we receive friendly messages safely. We also need to quickly figure out when someone is sending an unfriendly message, even if we didn't know it was coming. If there are situations like this, we need to use very smart technology that can handle signals and figure out what kind of modifications have been made without being told directly. There are several modulation schemes available, including amplitude modulation (AM), frequency modulation (FM), phase-shift keying (PSK), and quadrature amplitude modulation (QAM) as discussed in [2]. To ensure reliable data transmission, it is necessary to accurately classify modulation schemes in a variety of applications, including wireless networking, satellite communication, and radar. In military communication, we can use the type of signal to figure out who is sending it. AMC can be used to do many things by finding signals that are spread out, like recognizing targets, discovering what's causing interference, and deciding how to divide the available spectrum. DL is a technology that has become very popular because it's good at extracting and showing information. It's been used to create new and exciting changes in things like making computers understand images and language. Deep learning is being used more and more in wireless communication and signal processing for things like identifying radio transmitters, estimating channels, and allowing multiple devices to use the same frequency. Deep learning has proven to be effective in AMC. Many existing AMC systems cannot ensure that all the samples they receive have the same strength. This makes it difficult for AMC algorithms to handle signals with varying strength levels. Therefore, improving the algorithms to be adaptable to changing signal strengths is a big challenge.

1.1 PROBLEM STATEMENT

Automatic Modulation Classification (AMC) with conventional methods is not accurate at low SNRs and handles fewer modulation schemes. Recent advancements in deep learning lead to highly sophisticated neural networks which are computationally intense and require GPU and good performance which has expensive hardware.

1.2 OBJECTIVE

1. To achieve an AMC which can classify different types of Modulated Signals.
2. Classifier which can Perform signal Decoding and analysis
3. A fast and reliable AMC model for various applications mainly military purposes

1.3 BACKGROUND

Before, AMC used numbers and statistics to analyze signals. Intelligent methods for automated modulation categorization which enhances wireless signal detection and spectrum monitoring, were proposed by certain academics such as [3]. However, these approaches are often limited by the need for extensive feature engineering and may not be able to capture complex patterns in the data like those used in [4, 5]. Lately, new ways of using computers to learn have been helping a lot in the field of AMC. leveraging the power of deep neural networks to automatically learn hierarchical representations of the input data. This way of combining features and training the model on it has proven not very success full in the process of modulation classification. In feature-based, it is important to keep track of every situation as a signal can change its type at any time. The paper published in [5] developed a features method for (AMC) using Convectional Neural Network, yielding a **92%** accuracy. However, this strategy encountered difficulties due to the computational price involved while integrating characteristics and training the model on them. AMC is a critical signal processing method used at the physical layer of wireless communication networks. Its major purpose is to detect the modulated format of a signal that arrives reliably in blind mode at the receiver's end. It is used to increase spectrum use efficiency. Recent studies in AMC methods are either complex or not very accurate for a variety of reasons, one of which is the use of raw or noisy datasets [6]. In this study [6] RNN and CNN were both used but the accuracy did not above 50 percent in the presence of noisy data. Residual Networks (ResNet) [7] and Densely Connected Networks (DenseNet) [8] were recently created to improve deep neural network feature propagation by building shortcut channels across network layers. When the bypass connections are added, an identity mapping is generated, allowing the deep network to learn simple functions. In [9], it was demonstrated that a Res-Net architecture was capable of differentiating between 24 different modulation types. Dense-Net performed well in picture recognition but was not employed in modulation recognition. A Convolutional Long Short-Term Deep Neural Network (CLDNN) was recently introduced in [10], which combines the designs of CNN and Long Short-Term Memory into a deep neural network by leveraging the complementary of CNNs, LSTMs, and conventional deep neural network architectures. The LSTM unit is a Recurrent Neural Network memory unit. RNNs are neural networks with memory that can learn sequence tasks like speech recognition and handwriting recognition. LSTM optimizes the gradient vanishing problem in RNNs by incorporating a forget gate into its memory cell, allowing it to learn long-term dependencies. The authors in [11] incorporated LSTM units into the neural network model and demonstrated good classification accuracy for a variety of modulation schemes. In this report, we describe two distinct architectures that perform better than the CNN introduced in [12].

1.4 MOTIVATION

The primary undertaking of this project was to acquire knowledge and skills pertaining to deep learning neural networks within an academic context. Deep learning has exhibited a significant effectual performance in various fields of informatics, specifically in wireless networks and applied in communication technology. A multitude of advanced modulation and coding techniques have been developed, leveraging deep neural networks, to surmount the limitations of preceding methodologies.

1.5 MODULATION

Modulation means changing a repeating sound or signal called the carrier signal by using another signal called the modulation signal that carries information to be sent. The modulation signal can be different things, like a sound from a microphone, a moving image from a video camera, or a bunch of computer data. The main frequency is faster than the signal's frequency. Modulation means putting information on a signal that is sent to a different place. Radio communication sends a signal through the air to a radio receiver more about this in [4]. A modulator is a tool that changes signals to make them better. A demodulator is a circuit that undoes modulation. There are many types of digital Modulation some of the commonly used modulation types are:

1. Phase-shift keying (PSK)
2. Binary PSK (BPSK), using M=2 symbols
3. Quadrature PSK (QPSK), using M=4 symbols
4. 8PSK, using M=8 symbols
5. 16PSK, using M=16 symbols
6. Differential PSK (DPSK)
7. Differential QPSK (DQPSK)

8. Offset QPSK (OQPSK)

9. π/4–QPSK

1.6 MODULATION CLASSIFICATION

Prior to transmission, all radio communication signals, including those used for television, telephony, and other applications, undergo modulation. The recognition and classification of modulations are fundamental requirements for ensuring precise demodulation. The present technology exhibits different implementations across diverse domains, including but not limited to military, intelligence, and civilian sectors. In order to tackle pertinent concerns, substantial attention has been directed towards the advancement of signal processing and artificial intelligence methodologies in contemporary times such as discussed in [4]. Within the realm of signal processing, AMC is an intermediate procedure located between signal recognition and demodulation. The acquisition of signals by a recipient who is unfamiliar with the signals is seen as a vital procedure. The importance of this phenomenon may be seen in a variety of applications, most notably spectrum administration and interference detection. There has been a substantial corpus of scientific effort on the subject of signal processing for several decades. These studies are divided into two categories: likelihood-based approaches (LB) and feature-based methods (FB).

1.7 LIKELIHOOD BASED METHOD

In order to accomplish modulation classification, it is generally observed that FB algorithms exhibit a straightforward implementation and a substantial degree of resilience towards model inconsistencies such as timing errors and phase offsets. However, these algorithms may not represent an optimal solution from a Bayesian perspective and often necessitate an offline training phase for the classification system. In contrast to procedures involving an offline phase, likelihood algorithms possess the advantageous quality of being capable of optimizing the Bayesian framework. However, this advantage is accompanied by the drawback of greater computational intricacy. The present algorithms are designed to establish probability functions for signals that have been either received or retrieved through a multitude of modulation hypotheses. The algorithms are then utilized to make classification determinations, which are primarily based on selecting the maximum value from said functions. The algorithms themselves can be categorized into two distinct groups according to how the data and unidentified parameters are processed. The present taxonomy encompasses two distinct algorithmic approaches, namely those founded upon the Average Likelihood Ratio Test (ALRT) and those rooted in the Hybrid Likelihood Ratio Test (HLRT). The former perspective entails the treatment of both the data and the unknown parameters as random variables possessed of Probability Density Functions (PDFs) that necessitate averaging across them. Consequently, a requisite for their implementation is an antecedent acquaintance with the probability density functions (PDFs) of the aforementioned unknowns, an undertaking that, in circumstances of obscure communication parameters, may not be feasible or accessible. The algorithms that are based on Hybrid LRT (HLRT), in contrast, are perceived as more pragmatic in nature, due to the fact that these algorithms perceive the unknown parameters as uncertain determinants that are subject to estimation, whereas data is treated solely as random variables meant for integration. More about this can be read in this [13].

1.8 FEATURE BASED METHOD

The feature-based method sorts a signal by looking at its special parts. There are five AMC features that people often use. These are features that show what is happening in real-time, features that show changes in frequency, features that look at patterns, features that look at shapes, and features that show where the signal crosses zero. Instantaneous features show how loud, high-pitched, or fast a sound is by looking at its amplitude, phase, and frequency variations. In simpler words: People often use different tools to make signals easier to understand, such as smoothing or filtering. They might change the signal by using things like Fourier or Wavelet domains. Scientists use different statistical features like HOMs, HOCs, HOCCs, and cyclo-stationary to find out what shapes a group of constellation points makes to detect the signal. We will use different features taken from our algorithm for our project.

Chapter 2

LITERATURE REVIEW

We have learned so far that there are several ways to detect the modulated signal at the receiver side. The modulator is software based so the better the algorithm the better the classification. Now let's go one by one and take a look at some of the ways that have been adopted before and how they perform.

2.1 MODELS

2.1.1 RECURRENT NEURAL NETWORK

RNNs are very famous among deep learning algorithms. This was a two-layer network in which the i/q samples are broken down into two Gated Recurrent Units (GRU) and fed to one after the other. More about this in [14]. This method has an accuracy of over 91%.

Results: The new method is better than the old one when there is less noise. This makes the accuracy of recognizing things better.

2.1.2 CONVOLUTIONAL NEURAL NETWORK

A CNN-based AMC is proposed in [15], Which automatically extracts features from a lengthy symbol rate data sequence coupled with the calculated signal-to-noise ratio (SNR.

Results:

This was a CNN model that works in two steps. First, it trained the model for a set of parameters, and using that knowledge it trained a newer model. So basically the concepts of transfer learning were used.

2.1.3 FAST DEEP LEARNING AMC

In this work, a fast approach was taken toward classification by reducing the data set. The Original data set was created from GNU radio and then broken down into smaller subparts. Then a Convolution Neural Network (CNN), Convolution Long Short-term Deep Neural Network (CLDNN), a Long Short-Term Memory neural network (LSTM), and a deep Residual Network (ResNet) were implemented in [16].

Results:

This approach gave an accuracy value of around 90%. This model aims at making the best classification as fast as possible by using minimum data set for training the model. In real life, we don't have a limited data set but a lot of information is being transferred so no compromises should be made on training the model with minimum data and increasing accuracy that way.

2.1.4 FEATURE FUSION TECHNIQUE

This Model [5] presents a scheme of using combinations of different features for classification. Their model achieved a 92% accuracy which is a very good number but only at -4 and above.

Results:

This way of combining features together and training the model on it has proven very success full in the process of modulation classification. In feature-based, it is important to keep track of every situation as a signal can change its type at any time.

2.1.5 MULTI-STREAM NEURAL NETWORK

In [17] this type of neural network, the authors have created a neural network that works by extracting features one by one and moving from one layer to the next with the features of the previous layer method. This method increases the features using a stream method. To achieve the same effect as a big convolution kernel, the stream network superimposes a tiny kernel with fewer parameters. The network layer number is lowered, and the issue of overfitting is successfully avoided.

Results:

In this type of study when using not just one layer for feature extraction but using multi-layer has enhanced the performance to a great extent. This algorithm was complicated and the complexity level is very high. Due to the higher levels of complexity increasing the accuracy score is a tough and time-consuming task.

2.1.6 DEEP CASCADING NETWORK ARCHITECTURE

This method introduced two ways to enhance the modulation classification technique. It uses two detectors one for SNR and the other for modulation recognition.

Results:

Using this method an accuracy of 91.0% was achieved. This system is incomplete and can be completed to achieve a higher accuracy score.

2.1.7 RESNET-50 AND INCEPTION RESNET V2

This network is [18] used images of 8 modulated schemes to create a network that worked on classifying using images. It was a network that worked on the probability of the best guess to classify the modulation scheme.

Results:

The process of figuring out the type of modulation used was done in three steps. Two models were taught using pictures that were created of different colors. This network does not perform well in my opinion because it was only classifying between 8 schemes instead of 11 and even after that it was unable to identify between two common types of modulation which are 16QAM and 64QAM.

Year	Author	Title	Contribution	Modulation Schemes
2017	D. Hong, Z. Zhang, and X. Xu	Automatic modulation classification using recurrent neural network	Use of RNN which proved to be very computationally expensive and accuracy didn't go beyond 91%	11
2018	F. Meng, P. Chen, L. Wu, and X. Wang	Automatic modulation classification: A deep learning-enabled approach	Uses transfer learning with CNN way too many parameters	4
2019	S. Ramjee, S. Ju, D. Yang, X. Liu, A. E. Gamal, and Y. C. Eldar	Fast deep learning for automatic modulation classification	Used multi-model but only reach an accuracy of 90% at only few SNR	10
2020	Kumar, M. Sheoran, G. Jajoo, and S. K. Yadav	Automatic modulation classification based on constellation density using deep learning	used images of modulated signals for detection	8
2023	Proposed	Deep Learning Based Automatic Modulation Classification	Uses light convnet and stacked deep neural network with fewer parameters and better accuracy than previous models	12

Table 2.1: Literature Review Table

Table 2.1 shows our findings in a table. All these methods were discuses in chronological order. Some of them were likelihood-based and some others were feature based. In order for our project, we have decided to take multi approaches from feature extraction models from the time domain as well as frequency domain using the help of deep learning neural networks and in the end, we will see what kind of model gives us good accuracy as well as reliability.

Chapter 3

METHODOLOGY AND SYSTEM DESIGN

This chapter will discuss the methodology and how we designed our deep learning-based classifier from the basic design to the architecture level.

3.1 DATASET CREATION

Most of the available open-source dataset does not reflect real-world scenario. The scenario includes a multi-path environment and inter-symbol interference. To address this limitation we generate our own dataset in practical settings. We collected data by receiving a high carrier signal from our receiver, converting it to a baseband signal, then passing that baseband signal to an ADC for digital conversion. After digitizing the signal, we generated chain digital in-phase quadrature (IQs) through the recording application.

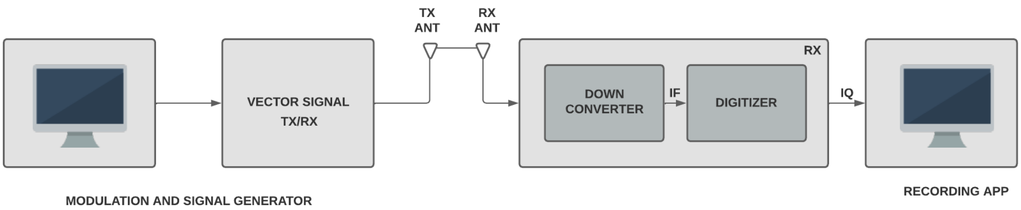

Figure 3.1: Block-Diagram for Data Collection

The procedure used for IQ sample creation and data collection can be seen in figure 3.1. We collected data by receiving a high carrier signal from our receiver, converting it to a baseband signal, then passing that baseband signal to an ADC for digital conversion. After digitizing the signal, we generated chain digital in-phase quadrature (IQs) through the recording application.

Figure 3.2: PXIe chassis and Receiver

Figure 3.2 shows PXle-1075 chassis along with PXle-5667 (receiver) which was used for data collection. For the data Acquisition, we have used PXI Vector Signal Transceiver. The PXI 5841 VST is a cool gadget that does two things: it can change its settings like a computer program, and it sends and receives strong signals like a fancy radio machine. The VST tests cell phone and wireless standards, and the small PXI Express can grow to fit more inputs and outputs. We used two chassis 1075 with vector signal transceivers for modulation signal generation and one for a receiver with 5667 receivers shown in figure 3.2. IQ rate was set to 50 kilo-samples per second. The channel bandwidth was set to 25kHz. A 25kHz signal sample was recorded and then down-sampled from a receiver to be sent to the mixer for IQ generation. In this AMC, a signal model has been used to generate training data for the neural network and to verify the accuracy of the classification. The signal model described how the signal has been modulated and the parameters that characterized it. The dataset used for training and evaluation is made up of 12 different modulation schemes *(QAM-16, QAM-32, QAM-64, QAM-128, QAM-256, FSK-2, FSK-4, FSK-8, FSK-16, FSK-32, BPSK, QPSK)* that are different variants of *5 Classes of QAM, 5 Classes of FSK and 2 Classes of PSK*. For training samples of **60%** and validation set of **40%** the training samples for each class are 1338 per class whereas for validation 892 samples per class.

3.1.1 QAM MODULATION

In the Quadrature Amplitude Modulation schemes, the signal can be represented using a constellation diagram in the complex plane [19]. The location of each point in the diagram represents a particular symbol, and the distance between the points represents the difference in amplitude and/or phase between the symbols. To model QAM signals, the constellation diagram can be represented mathematically using the following equation:

$$s(t) = I(t)\cos(2\pi f_c t) - Q(t)\sin(2\pi f_c t) \tag{3.1}$$

where $s(t)$ is the modulated signal at time t, $I(t)$ and $Q(t)$ are the in-phase and quadrature components of the signal, respectively, f_c is the carrier frequency, and $\cos(2\pi f_c t)$ and $\sin(2\pi f_c t)$ represent the carrier signals in the in-phase and quadrature channels, respectively.

3.1.2 FSK MODULATION

For FSK (Frequency Shift Keying) modulation schemes, the signal can be represented using a time-domain model that describes the signal as a sum of sinusoids with different frequencies, amplitudes, and phases [19]. The signal can be expressed mathematically as:

$$s(t) = \sum_{n=1}^{N} a_n \cos(2\pi f_c t + \phi_n) + b_n \sin(2\pi f_c t + \phi_n) \tag{3.2}$$

where N is the number of symbols in the FSK modulation scheme, a_n and b_n are the amplitude of the cosine and sine components of the nth symbol, respectively, f_c is the carrier frequency, and ϕ_n represents the phase offset of the nth symbol.

3.1.3 PSK MODULATION

PSK (Phase Shift Keying) modulation is a digital modulation scheme that represents digital data by varying the phase of a carrier signal. In PSK, the carrier signal is modulated by discrete phase shifts to encode binary symbols. The basic equation for PSK modulation can be given as:

$$s(t) = A \cdot \cos(2\pi f_c t + \theta) \tag{3.3}$$

where $s(t)$ is the modulated signal at time t, A is the amplitude of the carrier signal, f_c is the carrier frequency, and θ represents the phase shift, which depends on the binary symbol being transmitted. In binary PSK (BPSK), the phase shift is selected from two possible values, typically 0 and π radians. These phase shifts correspond to the binary symbols 0 and 1, respectively. Therefore, the phase shift can be expressed as:

$$\theta = \begin{cases} 0, & \text{if the input symbol is 0} \\ \pi, & \text{if the input symbol is 1} \end{cases} \tag{3.4}$$

The resulting BPSK signal can be represented as:

$$s(t) = A \cdot \cos(2\pi f_c t + k\pi) \tag{3.5}$$

where k represents the binary symbol being transmitted at time t.

3.2 PREPROCESSING METHODS

We generated a CSV dataset with 12 modulation schemes. Then, for preprocessing, we use the Savitzky Golay filter and Wavelet Transform to remove noise from our dataset. Following are the details of the pre-processing technique.

3.2.1 WAVELET TRANSFORM

The method has been extensively utilized and improved in many disciplines, and it has a unique time and frequency analytical property. The wavelet transform improves on the traditional Fourier transformation. It may investigate the time and frequency characteristics of the signal [20]. The wavelet function system is a set of functions used to describe a signal that is created by extending and translating the underlying function. More about this is [21]. The WT of the signal x(t) can be described as:

$$W(a,b) = \frac{1}{\sqrt{a}} \int x(t)\rho \frac{t-b}{a} dt \tag{3.6}$$

where W(a, b) denotes the WT coefficient, x(t) the wavelet function, a and b the stretch and translation factors, and δ(t) the conjugate of x(t). More about this in [21]. While the dimension of the window remains constant, the stretch and translation are superior to the Fourier transform. The inverse wavelet transform may be used to reconstruct the signal x(t) as follows:

$$x(t) = \frac{1}{C_\rho} \int \int \frac{1}{a^2} W(a,b)\rho \frac{t-b}{a} dadb \tag{3.7}$$

Following threshold quantification, the signal without noise is rebuilt using the inverse wavelet transform. Given the fact that threshold choice has a direct influence on denoising performance, selecting a proper threshold is crucial [21]. The signal samples from our dataset were challenging to utilize for training and validation prior to adopting any form of denoising algorithm, as seen below:

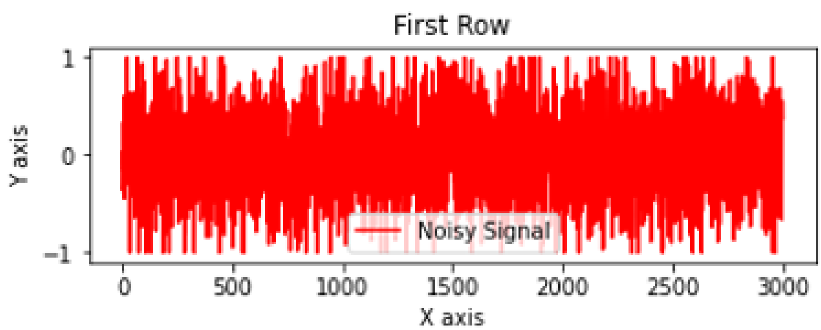

Figure 3.3: Signal Samples before denoising

Figure 3.3 shows signal samples before denoising which are very much noisy and cannot be used for training and testing of our model. After using the wavelet transform for denoising our signal samples the filtered samples can be seen as:

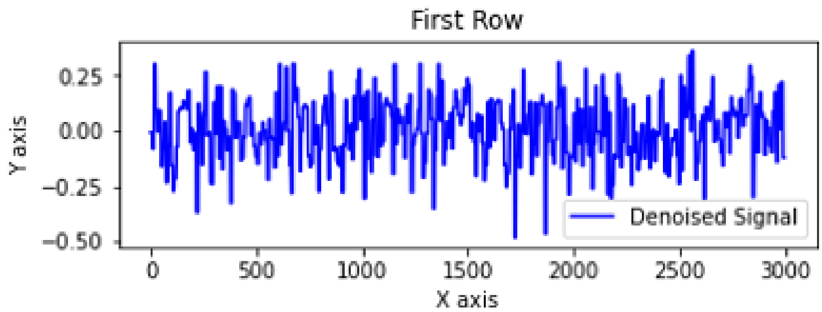

Figure 3.4: Signal Samples after denoising

Figure 3.4 shows signal samples after filtering which are less noisy and can be used for training and testing of our model.

3.2.2 SAVITZKY GOLAY FILTER

This filter is a type that may be used for a series of digital information points to smooth the information, thereby increasing accuracy without altering the signal trend. Convolution is performed by employing the linear least squares approach to fit successive subsets of neighboring data points with a low-degree polynomial. In instances where data points are uniformly spaced, it is possible to determine an analytical solution to the least-squares equations through the utilization of a singular set of ”convolution coefficients”. These coefficients can be utilized across all data subsets to obtain smoothed estimations of signal, or the derivatives of said signal, at the midpoint of each subset, more about this in [22]. We need to use filters on the data before we train the model so that it can be more accurate. This filter is typically used to make signals smoother and get rid of unwanted noise. This is a special computer tool that smooths out data by making a curve from nearby points and then using that curve to figure out a more accurate value for the center point. The filter is defined by the following equation:

$$y_j = \sum_{i=-m}^{m} c_i x_{j+i} \tag{3.8}$$

where y_j is the output value for the j-th data point, x_{j+i} are the input values in the $2m+1$ window around the j-th data point, and c_i are the filter coefficients. The filter coefficients are determined by minimizing the least-squares error between the smoothed values and the original data points [22]. Where $N_s = \frac{1}{s}N$. Multi-scaling is a pre-processing stage that transforms a raw noise signal into various coarse-grained signals of different resolutions.

Scaling is a preliminary processing stage that converts a raw noise signal into a number of coarse-grained signals with varying resolutions [22]. By averaging consecutive samples based on the scale value specified, the multi-scaling through coarse-graining technique reduces noise, smooths down the contour, and forms fine patterns. The scale value must be appropriately set since a greater scale value may result in the removal of crucial information required for capturing the pattern via feature/parameter extraction. The baseband signal for $n = 1, 2, \ldots, N_s$.

The baseband signal defined by (3.9):

$$\mathbf{x}_s = [x_s(1), x_s(2), \ldots, x_s(N_s)]. \tag{3.9}$$

Before using any kind of denoising technique the signal samples from our dataset were difficult to use for training and validation which can be seen as :

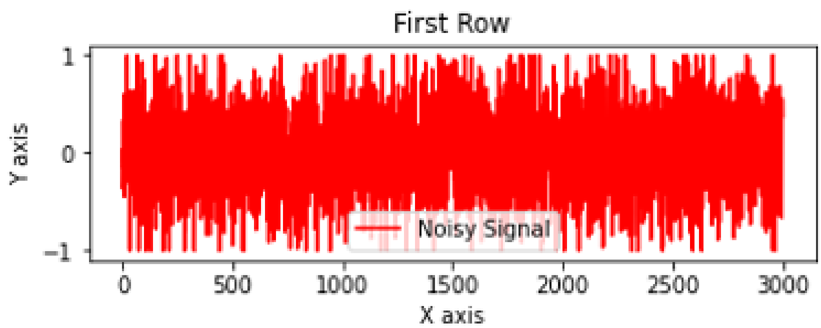

Figure 3.5: Signal Samples before filtering

Figure 3.5 shows signal samples before filtering which are very much noisy and cannot be used for training and testing of our model. After using the Savitzky-Golay filter for denoising our signal samples the filtered samples can be seen as:

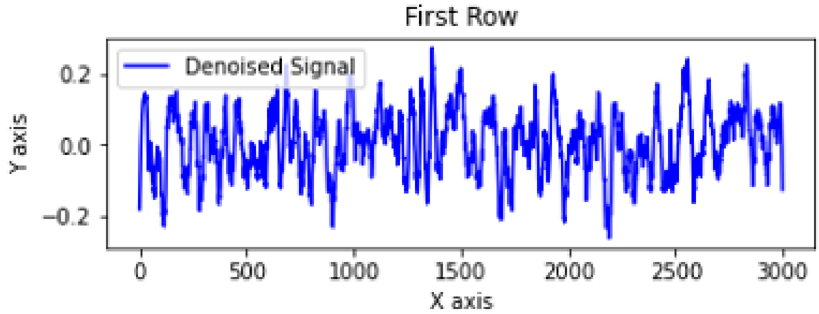

Figure 3.6: Signal Samples after filtering

Figure 3.6 shows signal samples after filtering which are less noisy and can be used for training and testing of our model.

3.3 PROPOSED DEEP LEARNING MODELS

For the algorithms, we decided to take two approaches. We have two architectures. Architecture 1 consists of a 1D CNN model and architecture two is a hybrid functional model consisting of combined three algorithms which are 1D CNN, LSTM, and autoencoders. The flowchart shows our data collection process, followed by pre-processing using the Savitzky-Golay filter, and classification using a deep stacked neural network of CNN, LSTM, and Autoencoder and model 2 of light 1D-ConvNet.

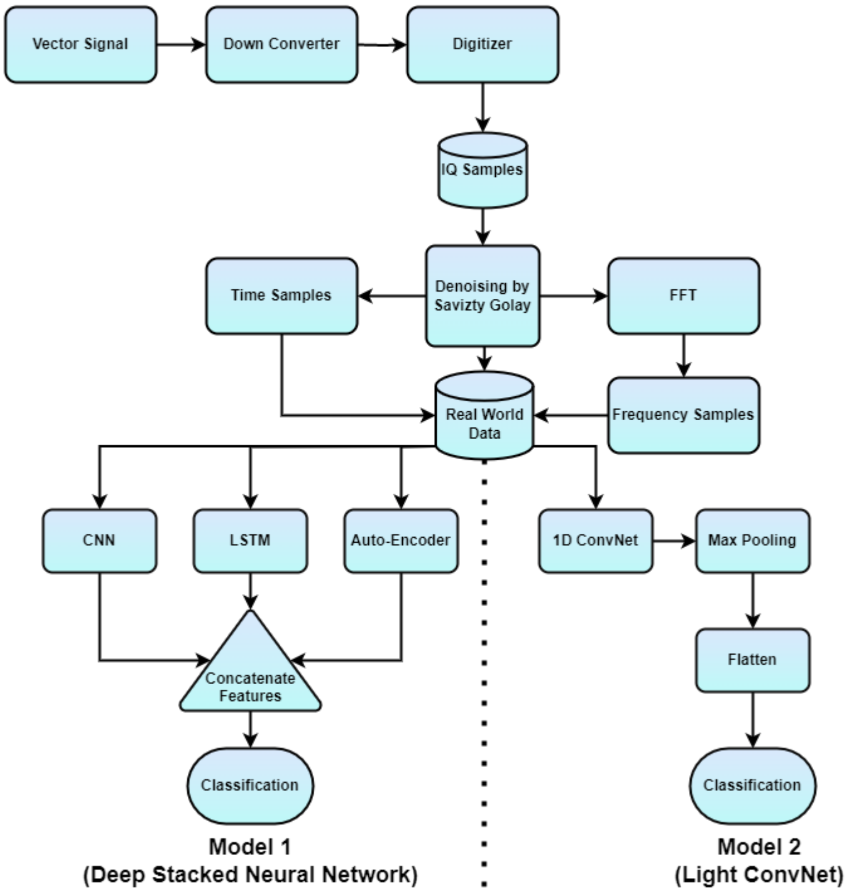

Figure 3.7: Models Flow chart

Figure 3.7 shows the following flow for our model which includes data collection, preprocessing, data concatenation, and then finally function hybrid model combination for classification and predictions.

3.3.1 LIGHT CONVNET

A CNN is a type of algorithm that is used to understand pictures in deep learning. When we talk about neural networks, we usually talk about doing math with grids of numbers. But with ConvNet, that's not how it works. It uses a method called Convolution. Convolution is a way of combining two shapes to create a new shape. It's like putting one shape on top of another and seeing how they change. Convolutional neural networks are created by multiple layers of fake brain cells. Artificial neurons are like little calculators that take in information and give out a value, which is similar to how real neurons in our brains work. When you feed an image into a CNN, each layer generates several activation functions, which are then passed on to the next layer, more about this in [23]. Usually, the first layer finds simple things like lines going straight or across. This information is sent to another layer that finds things like corners or edges that are made up of different shapes. As we go further into the network, we can understand more complicated things like objects and people's faces. The last part of the computer program looks at the picture and decides how probable it is that the picture belongs to a certain group. It uses scores between 0 and 1 to show this.

Sequential 1D CNN is a simple model for time-series analysis that is based on the convolutional neural network architecture. It is a feed-forward network that takes input as a sequence of data and outputs the classification or prediction of the next step in the sequence, more about this in [24]. The architecture consists of multiple layers of Conv1D, MaxPooling1D, and Dropout layers. We used over own collected dataset for sequential 1D CNN, then denoised it using discrete wavelet transform and used that denoise data as an input for a convolutional neural network. Our model required less computation time to train than previously proposed modulation classification models. The Conv1D layer applies a 1D convolution operation on the input sequence with a fixed-size kernel, which slides along the sequence to extract the features. The output of the Conv1D layer is a feature map that represents the learned features,more about this in [25]. The MaxPooling1D layer downsamples the feature map to reduce the spatial size of the output and control over-fitting. The Dropout layer randomly drops out some of the neurons to prevent over-fitting. The architecture of our Sequential light convnet is as follows:

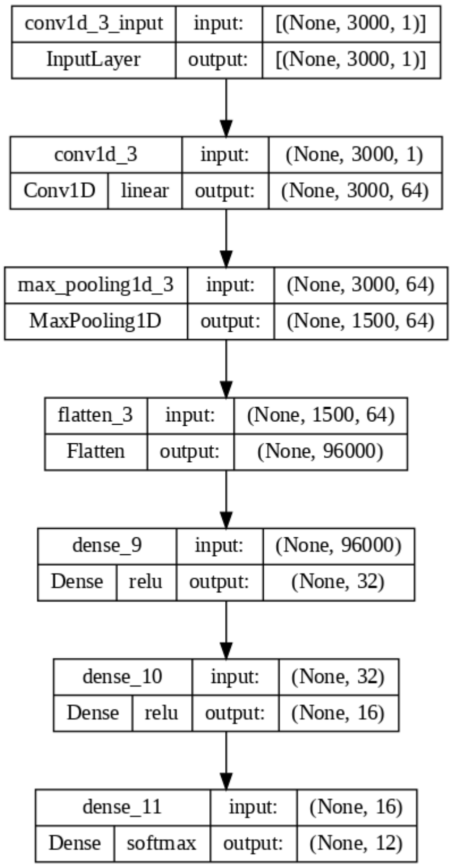

Figure 3.8: Model Summary of Proposed Light ConvNet

Figure 3.8 shows the implemented Architecture for our convolutional neural network labeled with layer names and shape of inputs for layer and lastly shows the 12 predicted modulation schemes for the output layer.

3.3.2 STACKED DEEP NEURAL NETWORK

The Stacked Deep Neural Network of CNN, LSTM, and Autoencoder is a deep learning model that combines the strengths of CNN, LSTM, and Autoencoder models. The model is designed for time-series analysis and is capable of classifying both the time and frequency features of the data. The dataset we used for the functional hybrid API model was a concatenated form of denoised time series data and denoised frequency domain samples so that our proposed model can classify modulated signals even in their frequency domain. The same concatenated data samples were used as input samples over

CNN, LSTM, and autoencoder, the results of which were then flattened and merged to pass from the output dense layer. The architecture for our proposed hybrid functional model is given as follows from its model plot:

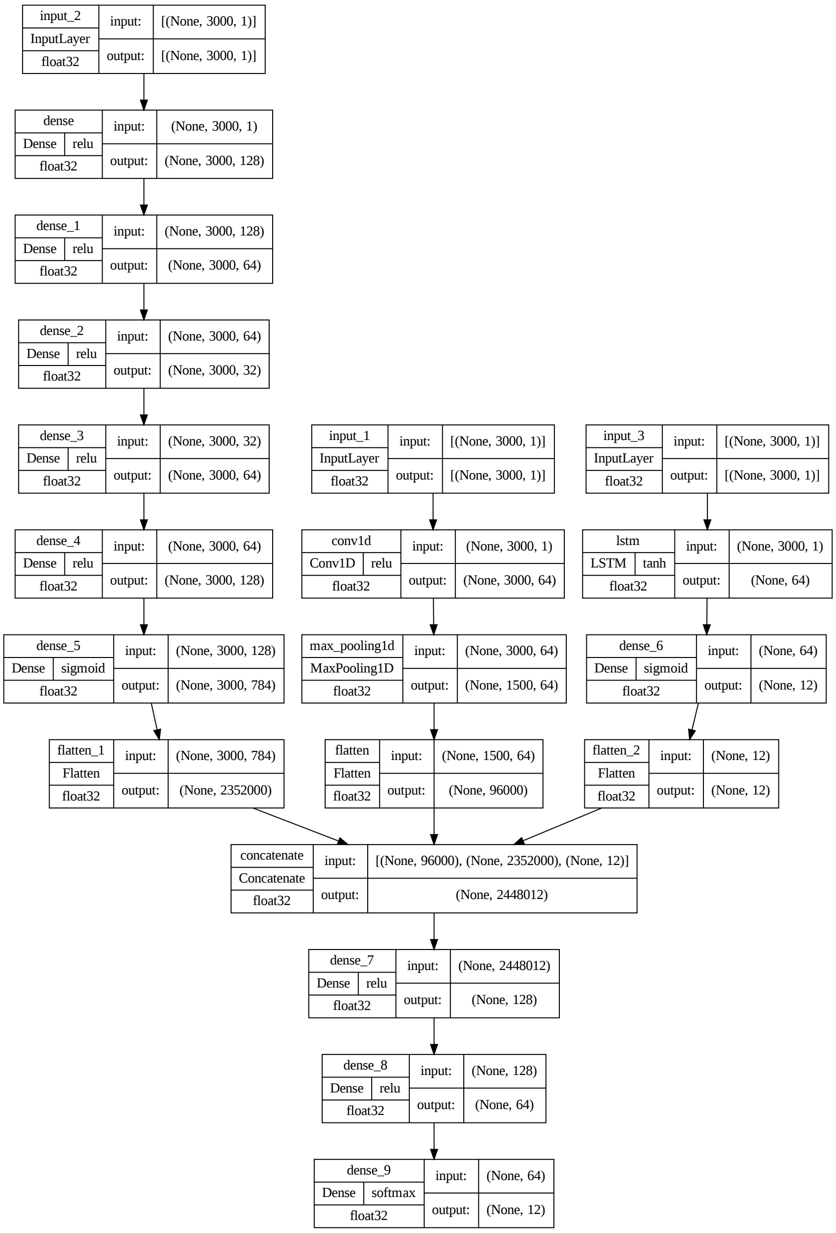

Figure 3.9: Model architecture of Stacked Deep Neural Network

The architecture used for the implementation of a hybrid functional model from its model plot is shown in Figure 3.9 in which the resulting features from CNN, LSTM, and auto-encoder were flattened and then concatenated before passing through the output dense layer for the final predictions. The architecture for our proposed hybrid functional model is given as follows from its model summary:

```
__________________________________________________________________________________________________
 Layer (type)                   Output Shape         Param #     Connected to
==================================================================================================
 input_5 (InputLayer)           [(None, 3000, 1)]    0           []

 dense_8 (Dense)                (None, 3000, 128)    256         ['input_5[0][0]']

 dense_9 (Dense)                (None, 3000, 64)     8256        ['dense_8[0][0]']

 dense_10 (Dense)               (None, 3000, 32)     2080        ['dense_9[0][0]']

 input_4 (InputLayer)           [(None, 3000, 1)]    0           []

 dense_11 (Dense)               (None, 3000, 64)     2112        ['dense_10[0][0]']

 input_6 (InputLayer)           [(None, 3000, 1)]    0           []

 conv1d_1 (Conv1D)              (None, 3000, 64)     128         ['input_4[0][0]']

 dense_12 (Dense)               (None, 3000, 128)    8320        ['dense_11[0][0]']

 lstm_1 (LSTM)                  (None, 64)           16896       ['input_6[0][0]']

 max_pooling1d_1 (MaxPooling1D)  (None, 1500, 64)    0           ['conv1d_1[0][0]']

 dense_13 (Dense)               (None, 3000, 784)    101136      ['dense_12[0][0]']

 dense_14 (Dense)               (None, 1)            65          ['lstm_1[0][0]']

 flatten_3 (Flatten)            (None, 96000)        0           ['max_pooling1d_1[0][0]']

 flatten_4 (Flatten)            (None, 2352000)      0           ['dense_13[0][0]']

 flatten_5 (Flatten)            (None, 1)            0           ['dense_14[0][0]']

 concatenate_1 (Concatenate)    (None, 2448001)      0           ['flatten_3[0][0]',
                                                                  'flatten_4[0][0]',
                                                                  'flatten_5[0][0]']

 dense_15 (Dense)               (None, 12)           29376024    ['concatenate_1[0][0]']

==================================================================================================
Total params: 29,515,273
Trainable params: 29,515,273
Non-trainable params: 0
__________________________________________________________________________________________________
```

Figure 3.10: Model Summary of Proposed Stacked Deep Neural Network

The architecture used for the implementation of a hybrid functional model from its summary is shown in Figure 3.10 in which the resulting features from CNN, LSTM, and auto-encoder were flattened and then concatenated before passing through the output dense layer for the final predictions.

Chapter 4

RESULTS AND DISCUSSIONS

This chapter presents a thorough analysis and interpretation of the acquired data in order to meet the research objectives and answer the research questions. This section will describe the results of experiments, surveys, observations, or other methods used throughout the study process. Furthermore, we will conduct a thorough discussion and analysis of these data, comparing them to current literature and ideas in order to detect patterns, trends, connections, and any notable findings. This chapter will contribute to a deeper knowledge of the study topic and give significant insights that can inspire future research endeavors and practical applications in the area by evaluating the results and engaging in critical dialogue.

4.1 ARCHITECTURE 1: LIGHT CONVNET

4.1.1 USING WAVELET TRANSFORM

In Figure 4.1 we can see the system's accuracy increased after just a few epochs, and on testing data, the accuracy was good from the start.

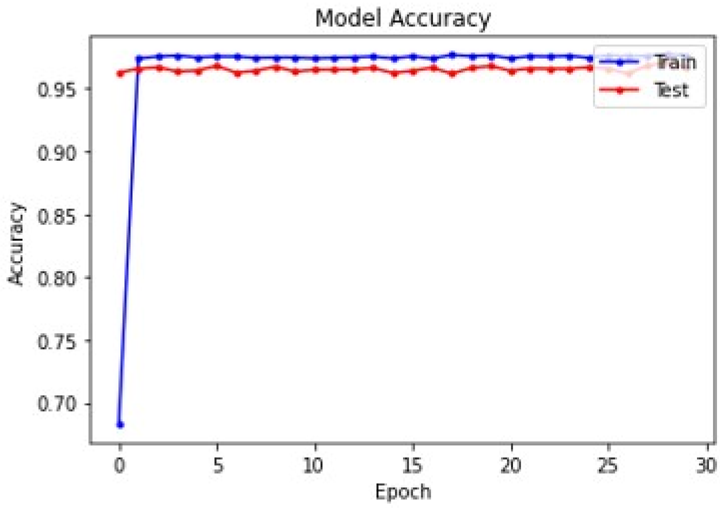

Figure 4.1: 1D CNN model accuracy using wavelet transform

In Figure 4.2 below we can see the model loss is decreasing over time which is precisely what we want.

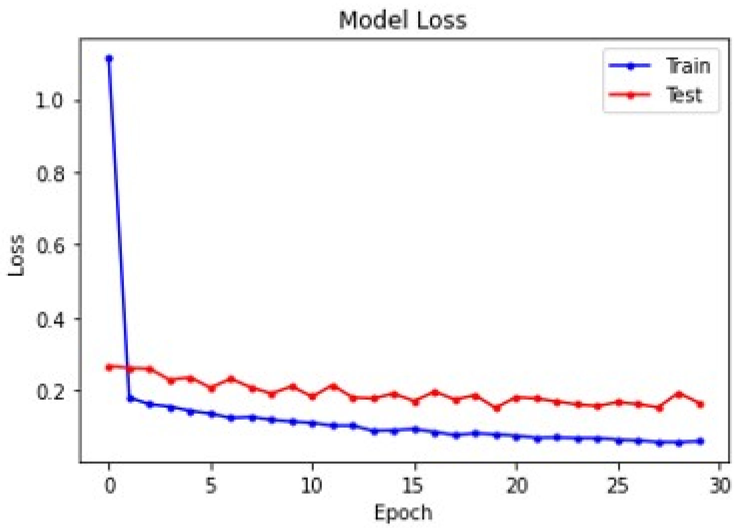

Figure 4.2: Loss Graph of 1D CNN model using wavelet transform

Figure 4.3 below shows the loss vs accuracy of how loss is decreasing and vice versa our model accuracy is increasing.

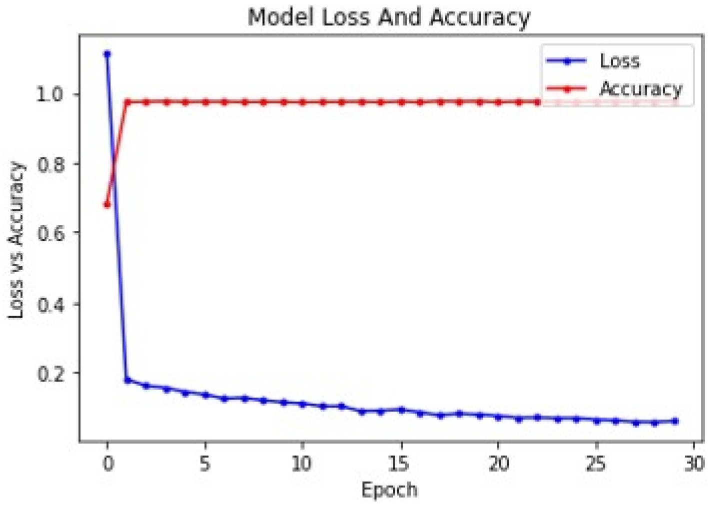

Figure 4.3: Model Loss and accuracy vs epoch using wavelet transform

Figure 4.4 shows the confusion matrix for 12 modulation schemes at 5db SNR.

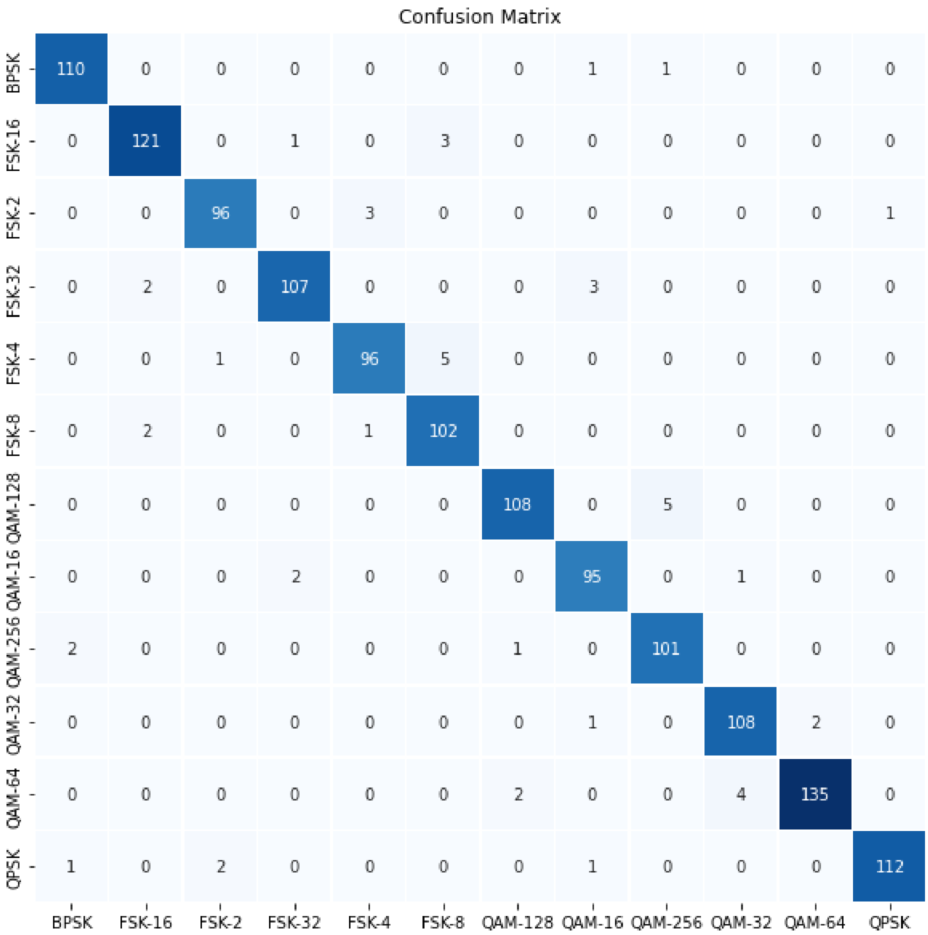

Figure 4.4: 12 Modulations Schemes at 5db SNR

Figure 4.5 shows the confusion matrix for 5 modulation schemes at 5db SNR.

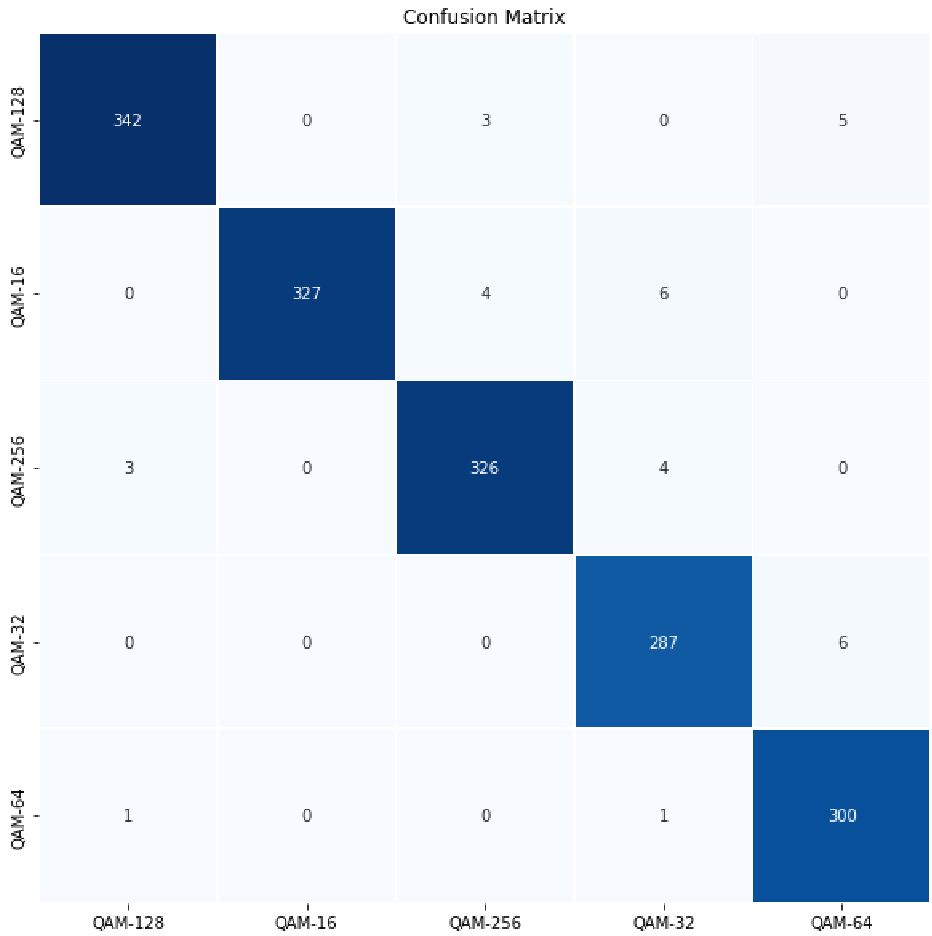

Figure 4.5: 5 modulation Schemes at 5db SNR

4.1.2 ACCURACY VS SNR

Figure 4.12 shows the SNR vs Accuracy plot on the basis of which we can say our model performs very well and model accuracy stays constant even at various SNRs.

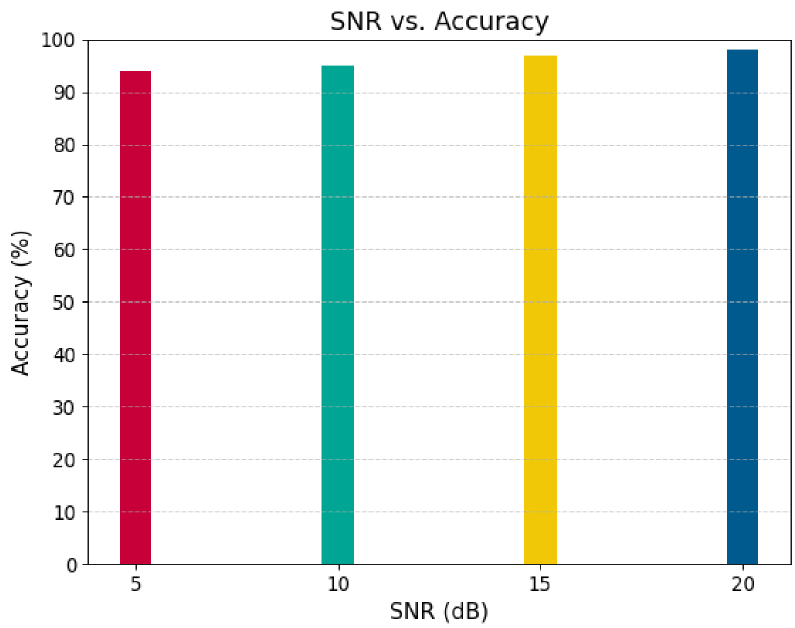

Figure 4.6: Architecture 1 SNR vs Accuracy Plot

4.1.3 USING SAVITZKY GOLAY FILTER

In Figure 4.7 we can see the system's accuracy increased after just a few epochs, and on testing data, the accuracy was good from the start.

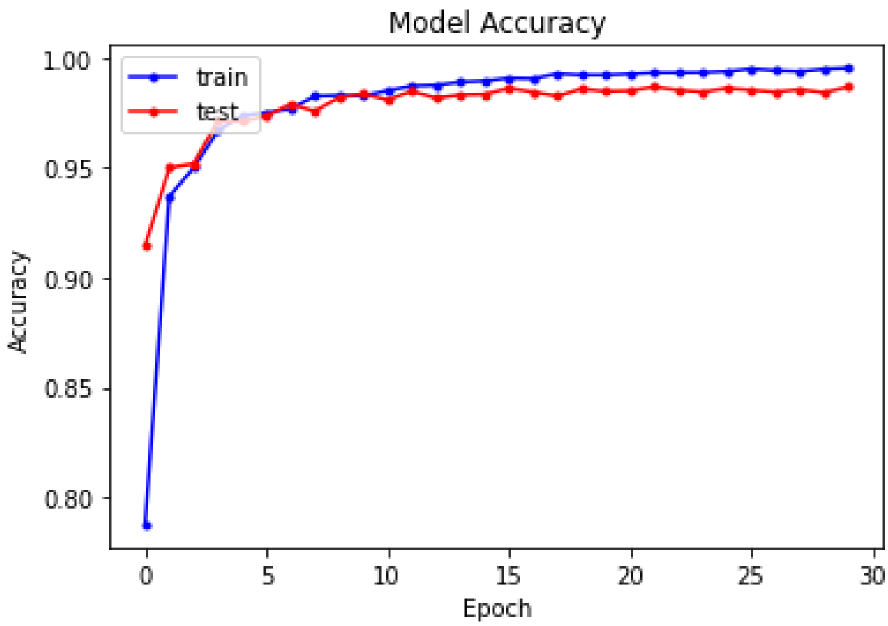

Figure 4.7: 1D CNN model accuracy using Golay filter

In Figure 4.8 below we can see the model loss is decreasing over time which is precisely what we want.

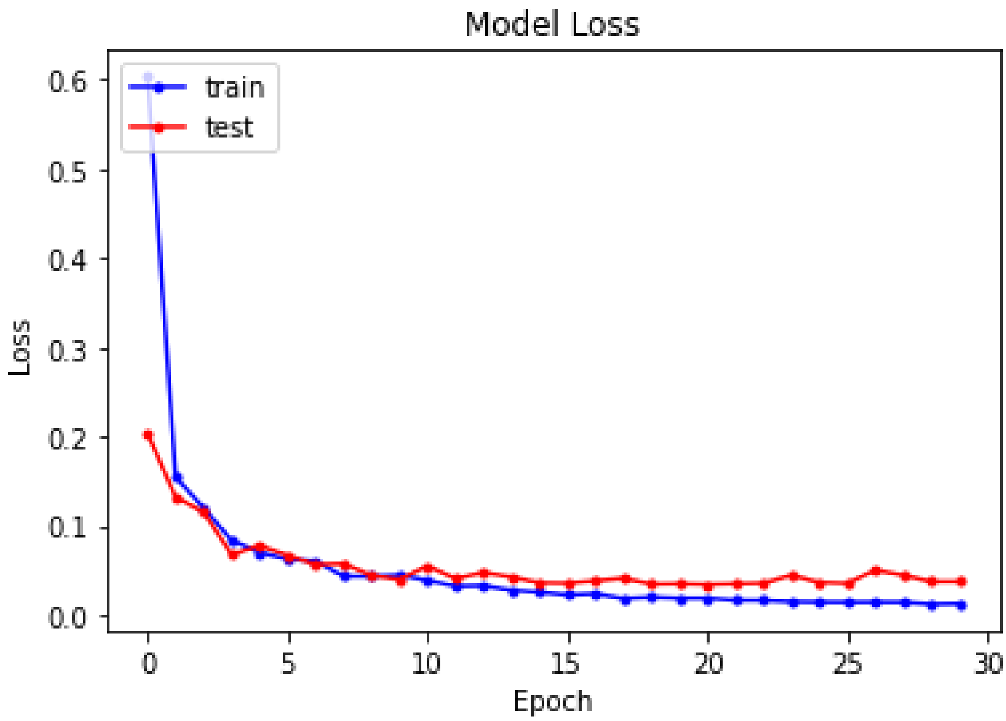

Figure 4.8: Loss Graph of 1D CNN model using filter

The graph in Figure 4.9 below shows the loss is decreasing relative to accuracy.

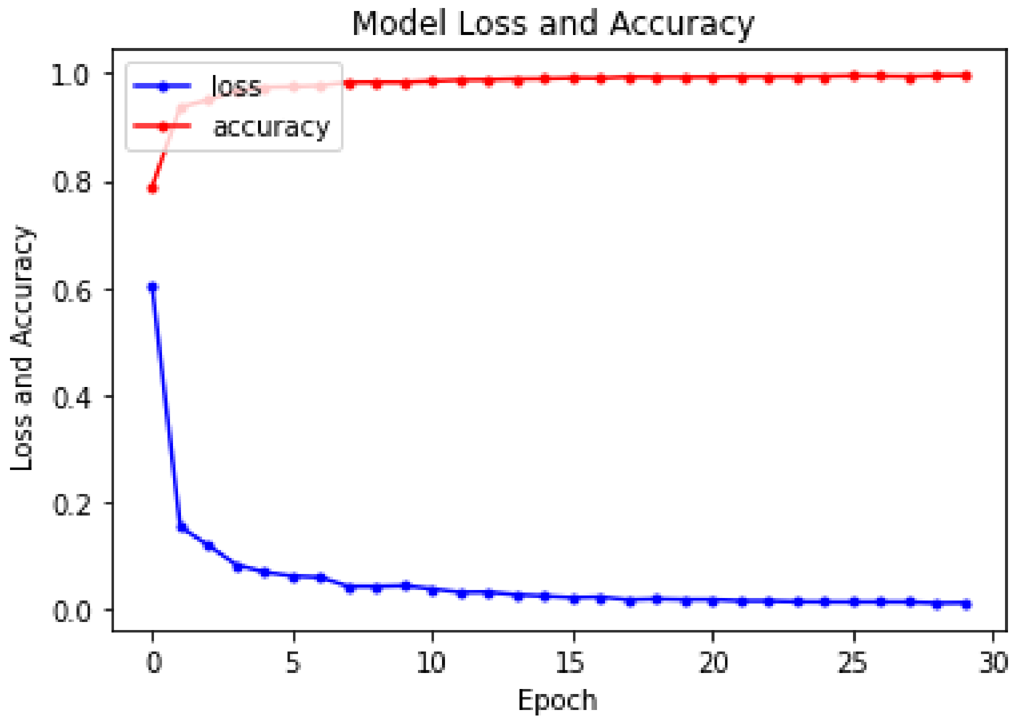

Figure 4.9: Model Loss and accuracy vs epoch using Golay filter

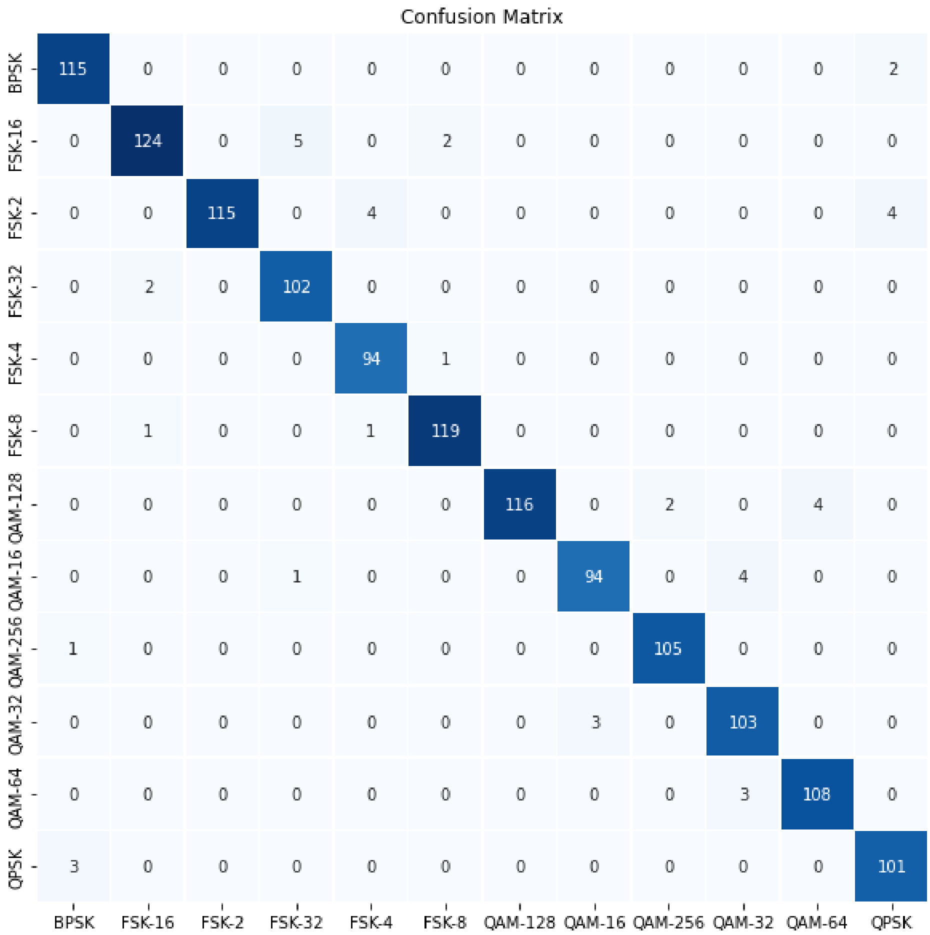

Figure 4.10: 12 Modulation Schemes at 5dB SNR

As shown in Figure 4.10, the total labels for BPSK were 113 and this model has done correct predictions for 110 labels and only 3 predictions went wrong. Similarly, for FSK-16 there was a total of 125 labels and this model has given 121 correct predictions and only 4 wrong predictions. For FSK-2 this model has given 96 correct predictions out of a total of 99 labels. This architecture has an accuracy of **98%**.

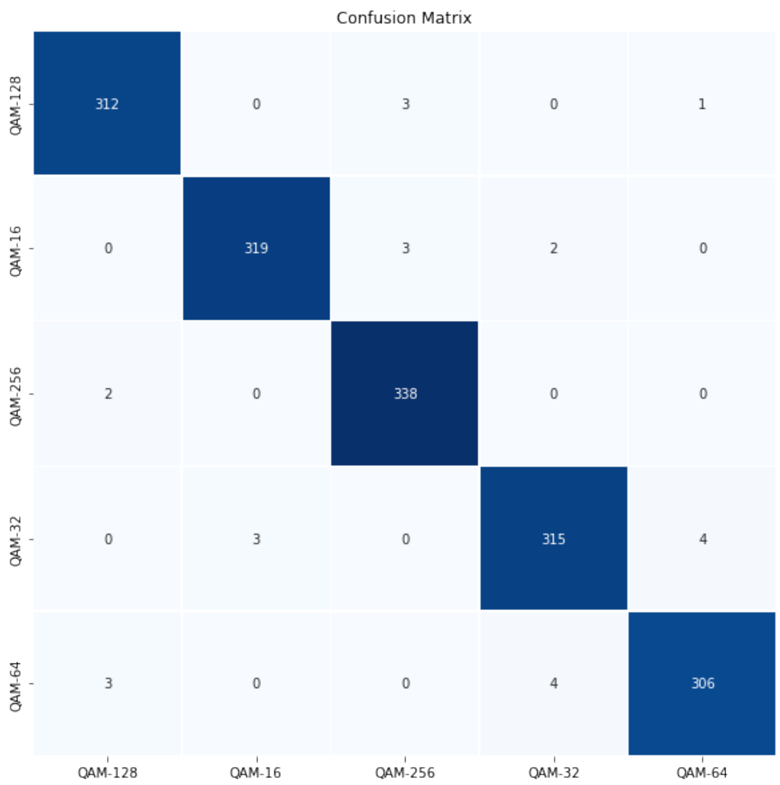

Figure 4.11: 5 Modulation Schemes at 5dB SNR

Figure 4.10 and Figure 4.11 show the confusion matrices for two different datasets containing 5 and 12 modulation schemes, respectively. This confusion matrix is proof of our successful model performance that is giving about **98%** accuracy.

4.1.4 ACCURACY VS SNR

Figure 4.12 shows the SNR vs Accuracy plot on the basis of which we can say our model performs very well and model accuracy stays constant even at various SNRs.

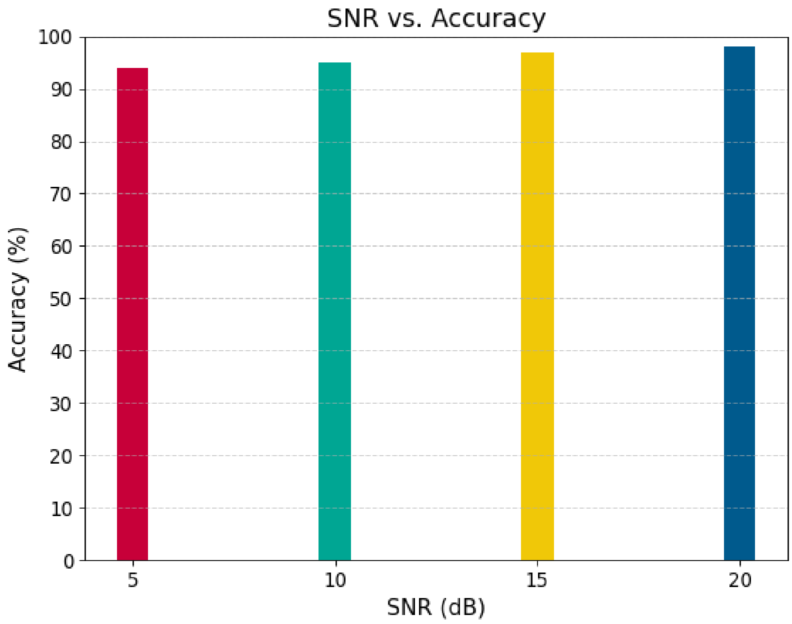

Figure 4.12: Architecture 1 SNR vs Accuracy Plot

4.2 ARCHITECTURE 2: STACKED DEEP NEURAL NETWORK

Following our proposed model, we achieved a hybrid functional API model of CNN, LSTM, and Auto-Encoders, with the following accuracy and loss for predictions shown in figure 4.13 and 4.14.

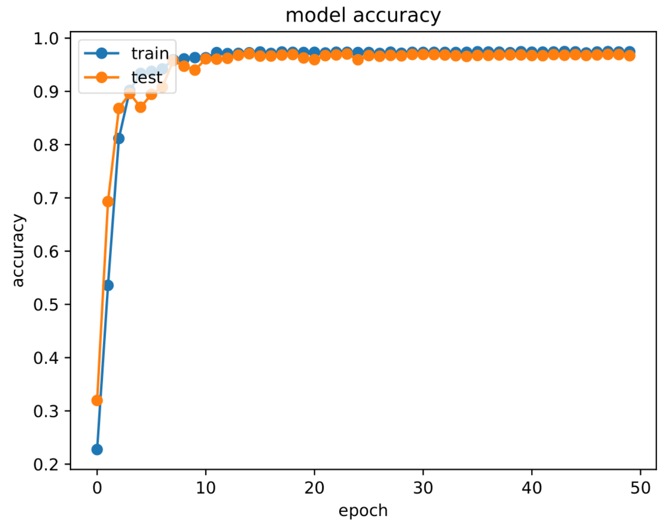

Figure 4.13: Functional Model Accuracy Graph

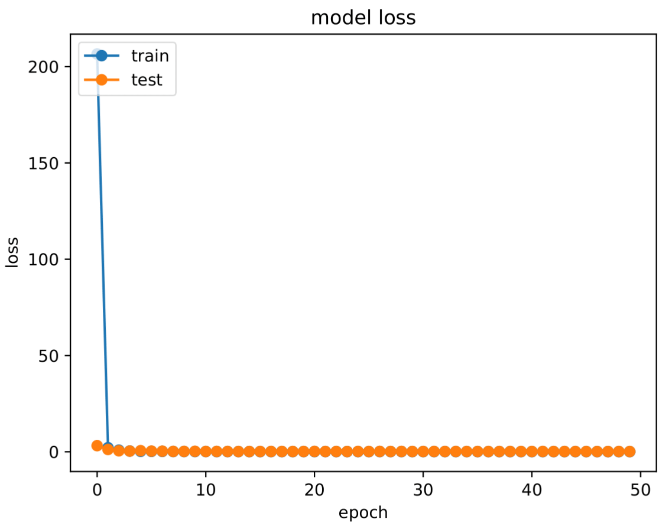

Figure 4.14: Functional Model Loss Graph

The confusion matrix is shown in figure 4.15. As shown in 4.15, the total labels for BPSK were 219 and this model has done correct predictions for 214 labels and only 5 predictions went wrong. Similarly, for FSK-16 there was a total of 216 labels and this model has given 206 correct predictions and only 8 wrong predictions. For FSK-2 this model has given 229 correct predictions out of a total of 235 labels. This architecture has an accuracy of **98%**.

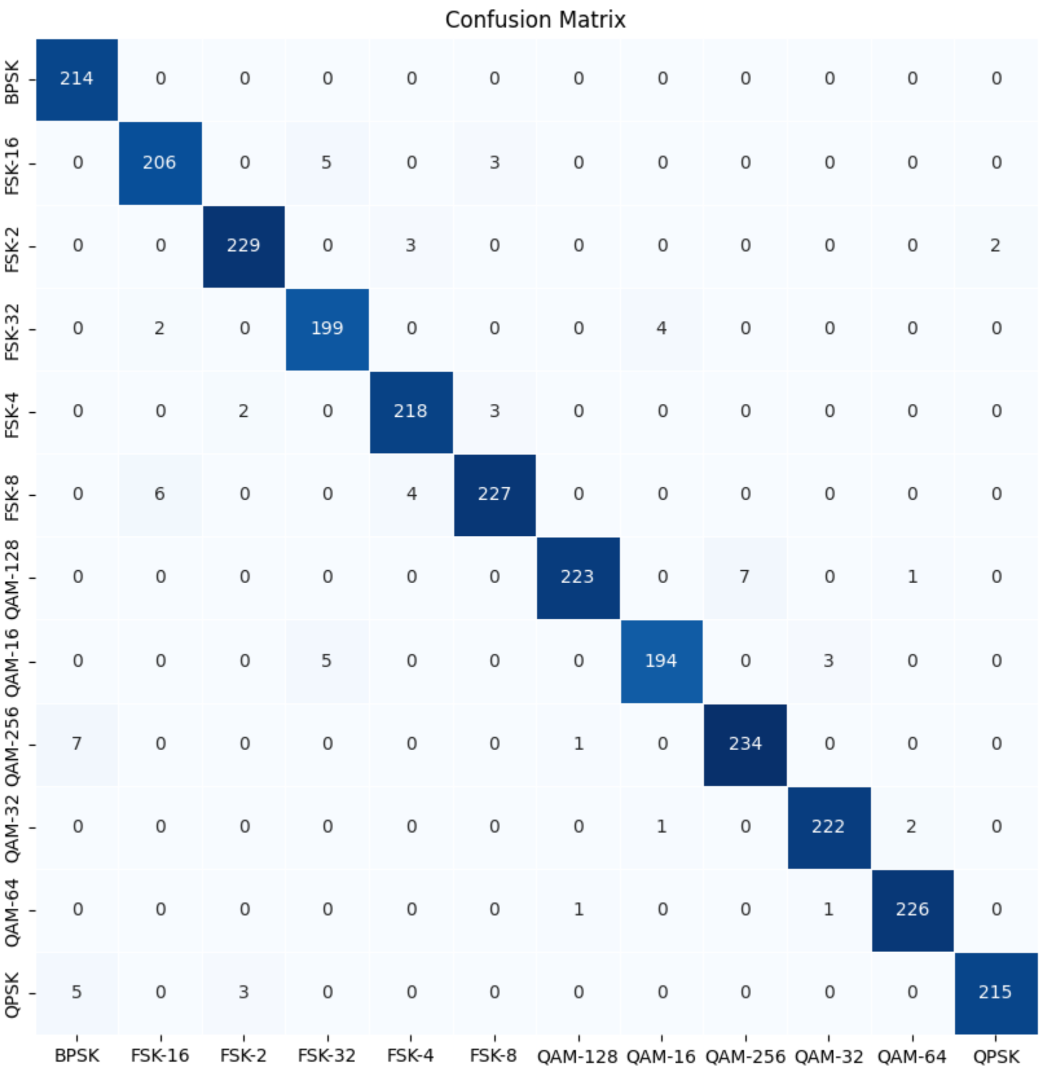

Figure 4.15: Confusion Matrix for Functional API model with 98% Accuracy

4.2.1 ACCURACY VS SNR

To show some more results we have some graphs to show the working of our model. Figure 4.16 shows the SNR vs Accuracy plot on the basis of which we can that our model performs very well even at various SNRs.The model accuracy stays constant even at various SNRs.

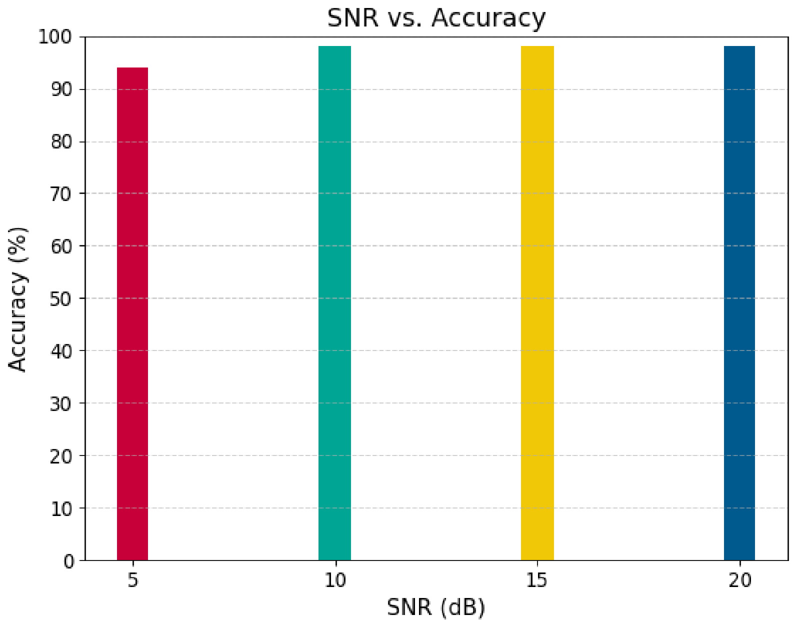

Figure 4.16: Architecture 2 SNR vs Accuracy Plot

Finally, we created 2 successful classifiers that have been discussed above. Both the models are giving up to **98%** accuracy. Model 2 is preferable due to its hybrid nature and less complex time.

4.3 COMPARISON WITH PREVIOUS MODELS

4.3.1 LIGHT CONVNET ANALYSIS

Figure 4.17 shows the proposed model comparison with previous models. We tested all the models on the same data and got the accuracy as shown in the graph. The proposed

Light ConvNet achieved significantly higher accuracy in comparison to the previously tested models, as shown in the graph. The accuracy values of our model, indicated by the purple color, are consistently higher than the values of the other models, represented by orange, green, red, and blue lines. For instance, at SNR level 20 dB, our model achieved an accuracy of 0.98, which is much higher than the accuracy values of all other models. Therefore, our proposed architecture can be considered a significant improvement over the previous models in terms of accuracy.

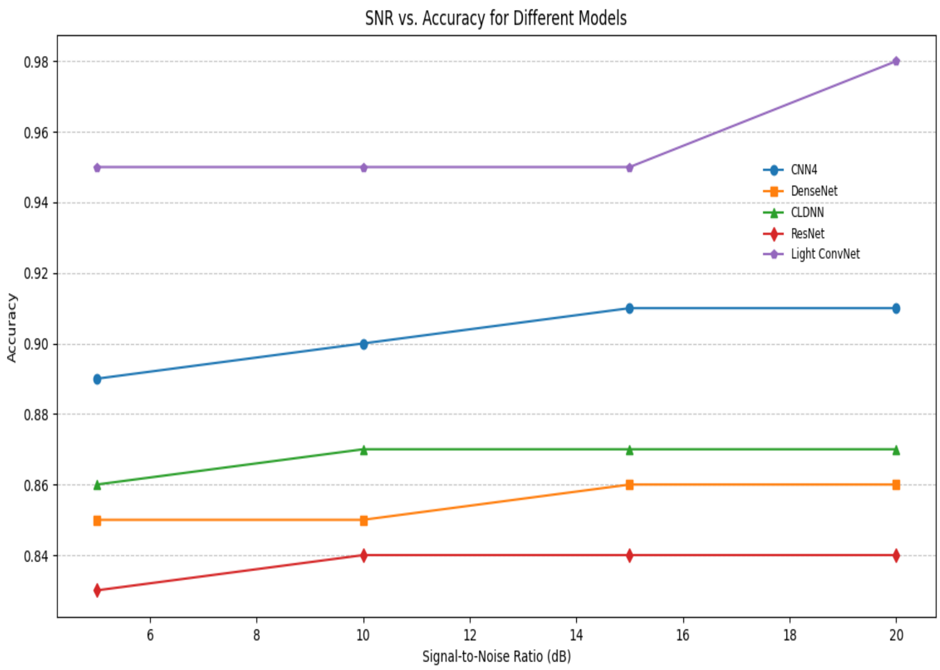

Figure 4.17: Architecture 1 Comparison with previous Models

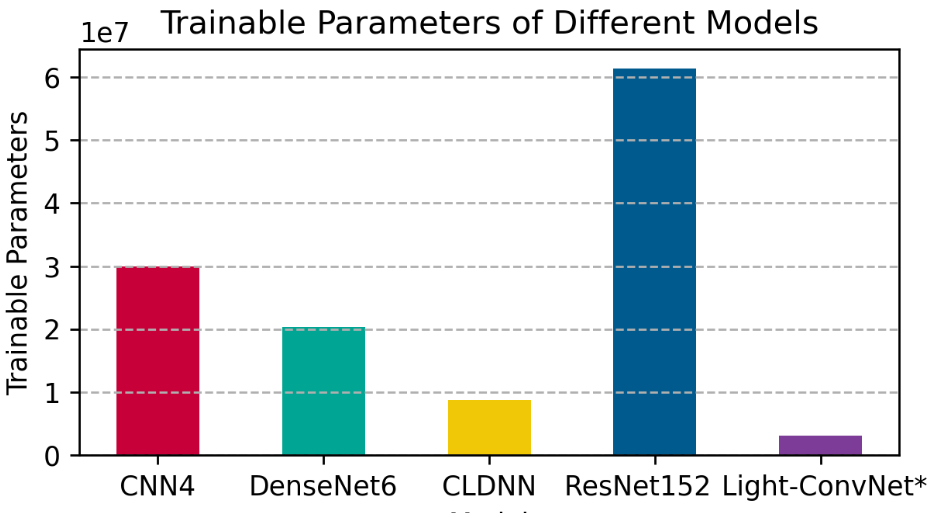

Figure 4.18: Architecture 1 Complexity Analysis

Figure 4.18 shows the total number of trainable parameters used by each model. We can judge by this graph that the proposed light convnet has very less parameters as compared to other models we have tested. The ResNet 152 has the greatest number of trainable parameters (over 60 million), whereas the light convnet has the fewest (about 3 million). In terms of the number of trainable parameters, our suggested light convnet is the best model, using a smaller number of parameters compared to the other four.

4.3.2 STACKED DEEP NEURAL NETWORK ANALYSIS

Figure 4.19 shows the proposed model comparison with previous models. We tested all the models on the same data and got the accuracies as shown in the graph. Our proposed architecture 2 achieved significantly higher accuracy in comparison to the previously tested models, as shown in the graph. The accuracy values of our model, indicated by the purple color, are consistently higher than the values of the other models, represented by orange, green, red, and blue lines. For instance, at SNR level 20 dB, our model achieved an accuracy of 0.98, which is much higher than the accuracy values of all other models. Therefore, our proposed architecture can be considered a significant improvement over the previous models in terms of accuracy.

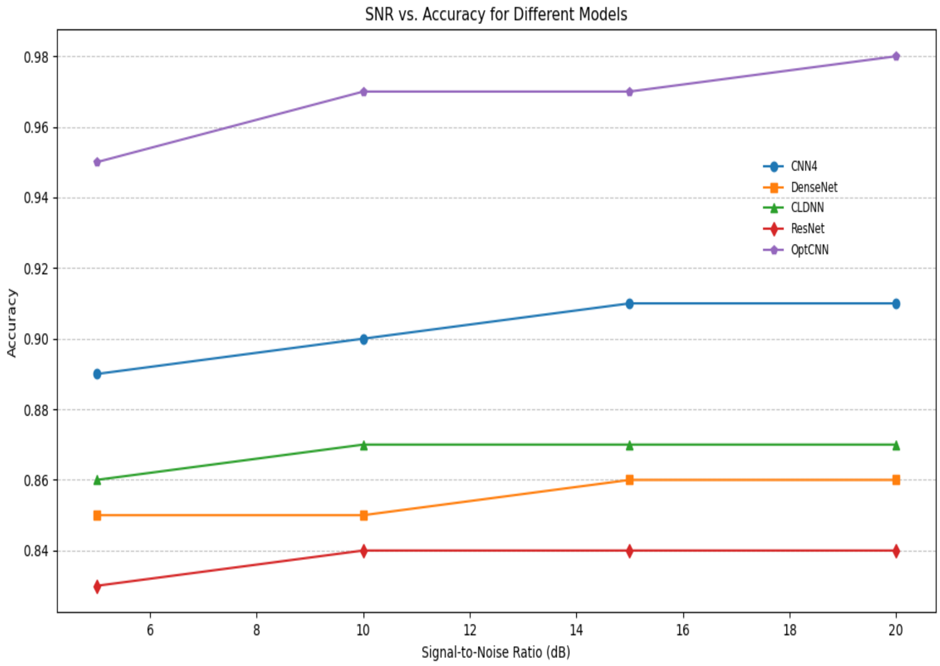

Figure 4.19: Architecture 2 Comparison with previous Models

The total number of trainable parameters utilized by each model is shown in Figure 4.20. The graph reveals that DenseNet6 and CLDNN have the lowest parameters among the five models, and our proposed stacking DNN is third, however, the penalty of low parameters in this model is accuracy. Both Desnenet6 and CLDNN do not achieve accuracy beyond 87 percent, their performance is also very low. However our model, while having more parameters in terms of computational complexity, produces considerably greater results.

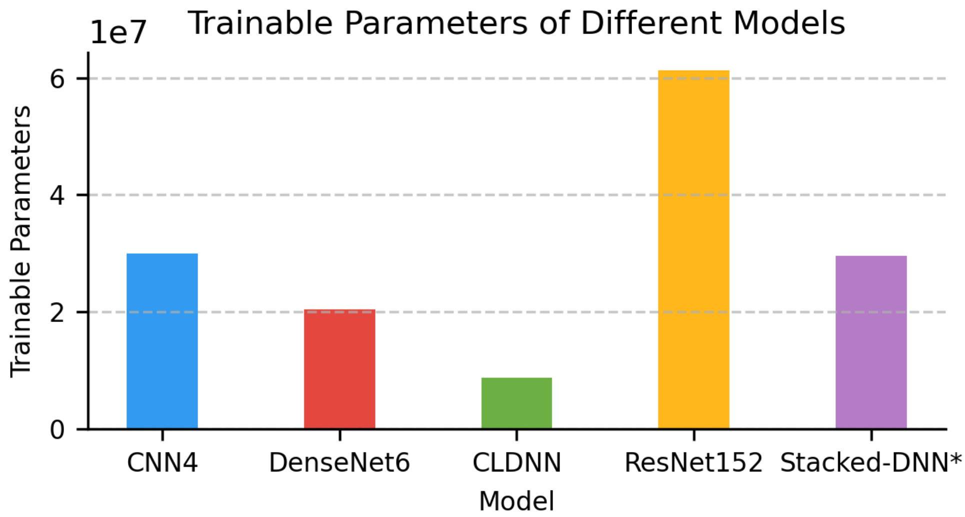

Figure 4.20: Architecture 2 Complexity Analysis

4.4 USER INTERFACE

After the completion of the architectures, a graphical user interface or GUI was created to showcase the working of our model. For this purpose, we used the pyqt5 library in Python. PyQt5 is a Python framework for the Qt toolkit, a well-known cross-platform application framework. PyQt5 allows Python programmers to leverage Qt's sophisticated frameworks [26] to construct graphical user interfaces (GUIs) for desktop applications. PyQt5 allows Python programmers to create graphical user interfaces (GUIs) for desktop applications using Qt's powerful libraries. PyQt5 enables developers to construct applications with a contemporary and aesthetically attractive interface that includes elements such as buttons, menus, text fields, and more. PyQt5 also gives you access to many sophisticated Qt capabilities, such as multimedia, network programming, and 2D/3D graphics [27]. Because of its simplicity of use and robust features, PyQt5 is a common option among Python developers. Its documentation is robust, with various examples and tutorials to help newcomers get started using [28]. PyQt5 is open-source and distributed under the GNU GPL v3 license, giving it a versatile and cost-effective option for programmers of all skill levels. It also has a strong and active developer community that contributes to its continuing development and provides support for [29].

Overall, PyQt5 is a flexible and powerful tool for constructing Python desktop apps, and its popularity with programmers is growing [30].

4.4.1 MAIN MENU

Let's take a look at how the app looks like once it is loaded.

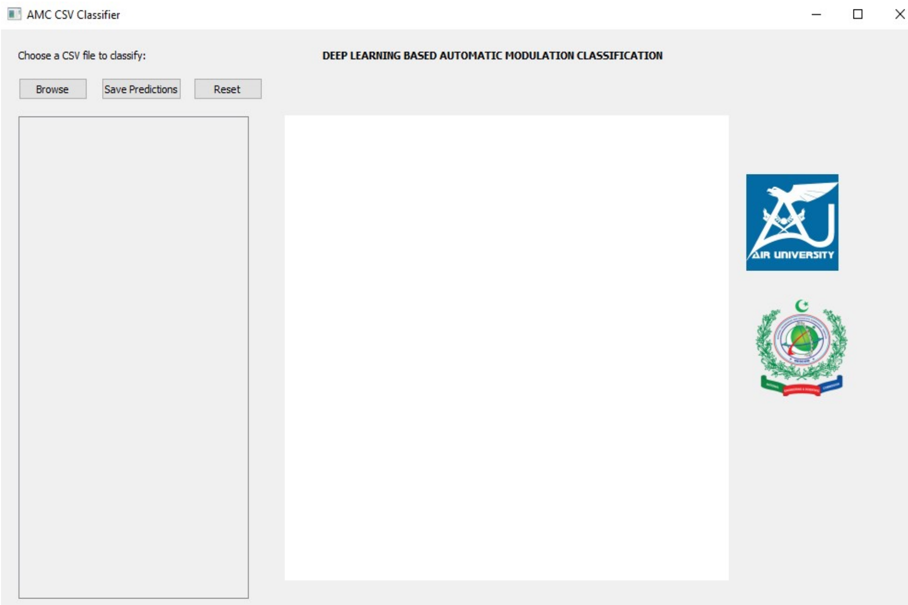

Figure 4.21: Project GUI Using PyQt5

Figure 4.21 shows the main menu of our classifier once it is started. It has three buttons

- Browse
- Save Predictions
- Reset

4.4.2 BUTTONS

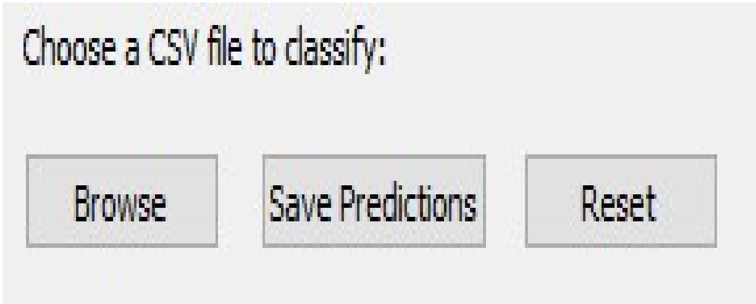

Figure 4.22: Button included in GUI

Figure 4.22 shows the buttons of our classifier.

- **Browse:** The browse button opens a menu that is used to load the test data from the disk which is present in a CSV file.
- **Save Predictions:** The save button is used the save the predicted result for further use if required.
- **Reset:** The reset button is used to reset the classification and start over again if there are multiple files.

4.4.3 RESULT DISPLAY

After the file has been loaded into the GUI the GUI takes a few seconds and after that gets the following result on the GUI.

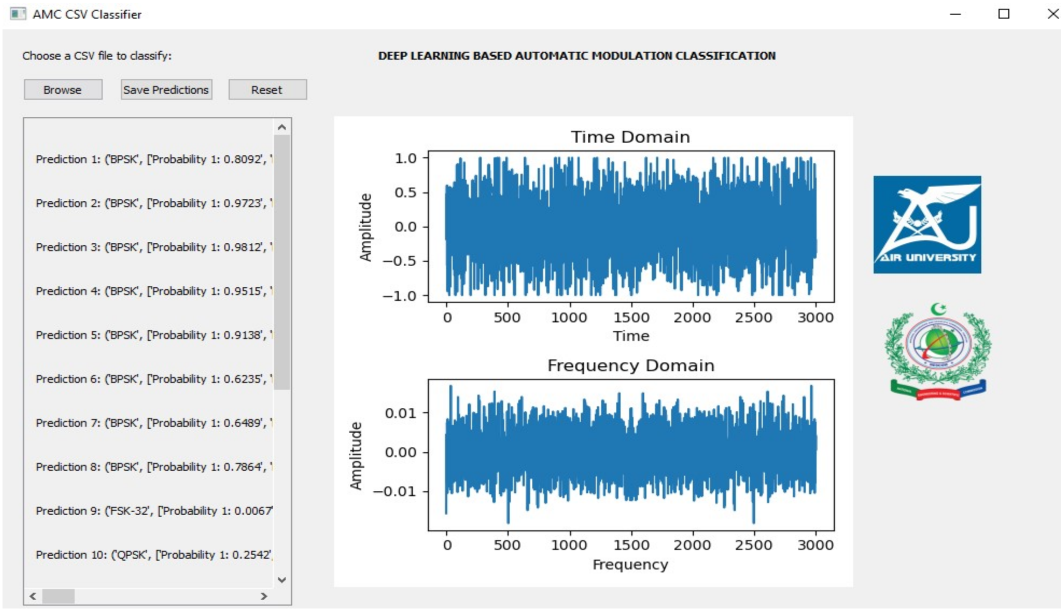

Figure 4.23: Classification Result on Display

Figure 4.23 shows what happens after the classification has been completed. We have two boxes one shows the predicted classes and the probabilities and the other shows the result of the first sample. Right now we have only shown the time and frequency domain plots of the first sample. We also have scrollable bars because the predictions were very large and we tried to make it look like an original Software defined radio software. In our test file we had 39 samples and after classifying all the samples it have given the predicted class and the probability values of all the other classes.

Figure 4.24: Predicted labels

Figure 4.24 shows the prediction of the test file. Right now we only used QAM 16 sample which is why all the labels are listed as QAM 16.

This was all about the results and the classifier. We have created a classifier that is computationally less expensive than all the previous models and also is more accurate with the greatest accuracy value reaching 98%. To make things easier we also created a GUI for anyone to use and make it simple for them to test their data using CSV files.

Chapter 5
IMPACT ON ENVIRONMENT AND SUSTAINABILITY

5.1 INTRODUCTION

In this section, we will discuss the impact of AMC on the environment and its sustainability.

5.2 APPLICATIONS OF AMC

AMC is becoming popular quickly nowadays and is used in many different ways in today's world. Each new model tries to become more accurate and efficient than the previous one, like an improvement. This is used for both regular and military talking.

5.3 AMC FOR CIVIL APPLICATIONS

AMC (Automatic Modulation Classification) is a critical technology in modern communication systems, particularly in civil applications. It is frequently used in signal identification, disturbance detection, and security of communication. AMC applications in civil communications are constantly evolving, and various research investigations are being conducted in this field.

The paper [31] did research on the development of AMC systems in wired communication networks employing high-order modulations. For high-order modulations, the authors offer a maximum probability (ML) classifier, a distributional test classification algorithm, and higher-order emulation features. The study finds that the ML classifier has higher performance in wired communication systems with both high-order modulations and restricted signal length.

In this [32] authors analyze the use of deep learning in AMC for wireless communication systems in another study. The researchers suggest a deep learning-based AMC technique that employs convolutional neural networks (CNN) and long short-term memory (LSTM) networks. According to the findings, the suggested technique outperforms standard AMC methods in terms of accuracy and resilience.
Wang in [33] also examines the use of AMC in cognitive radio networks (CRN). The authors present a WPT and SVM-based AMC technique. The study indicates that the suggested technique outperforms others in CRN contexts in terms of accuracy, computational complexity, and resilience.

Finally, AMC is an important approach in civil communication systems, and its applications are constantly evolving. The preceding experiments highlight the relevance of AMC in many communication systems, including wired and wireless communication systems, as well as cognitive radio networks. These studies also give useful information for the design and development of AMC systems for various applications.

5.4 AMC FOR MILITARY APPLICATIONS

Automatic Modulation Classification (AMC) is critical in military applications where signals must be categorized in real-time to aid decision-making. In the last few years, researchers have done important studies about using AMC in the military. According to [34], one research presented a hybrid method to AMC that incorporated two machine learning algorithms, Random Forest and Convolutional Neural Network (CNN), to categorize signals with improved accuracy and lower computing complexity. Another work employed a deep learning-based system to categorize signals in a low SNR environment, obtaining over 95 % accuracy [35].
Furthermore, significant research has concentrated on the durability of AMC algorithms in military settings. In a wartime environment, there is a considerable likelihood that a signal would be jammed, affecting the performance of an AMC algorithm. To solve this issue, [23] suggested an approach based on a deep neural network that can categorize

signals even in the presence of a jamming signal. Another research offered a multi-resolution time-frequency representation-based strategy to combating signal fading and interference in a military setting [36].

AMC is also being used in drone communication systems, which are becoming more relevant in military applications. An AMC algorithm was employed to categorize drone signals in one research, with a high accuracy rate of over 95 % [35].
Finally, AMC is critical in military applications, and various research has offered unique techniques for addressing signal categorization problems in such settings. These studies show that AMC has the ability to enhance the way decisions are made in military operations.

5.5 SUSTAINABILITY OF DLB AMC

Automatic modulation categorization (AMC) is critical in wireless communication systems such as military use, cognitive radio, and the Internet of Things (IoT). Traditional AMC approaches, on the other hand, rely primarily on hand-engineered characteristics and lack flexibility in the constantly shifting wireless communication environment, resulting in limited sustainability. Deep neural networks, which have recently been developed, have been frequently used for AMC problems and have obtained considerable performance increases when compared to older approaches. However, deep learning applications encounter sustainability issues such as high computational demands, consumption of energy, and model size. Researchers have also investigated the use of acceleration hardware and compression methods to minimize the deep learning-based AMC system's computational complexity and storage utilization. Kim [37] suggested a hardware acceleration for deep learning-based AMC that delivers great performance while consuming less power. In conclusion, deep learning-based AMC's durability is crucial for its practical implementations. To address the sustainability concerns, researchers have proposed a variety of ways, including optimizing the deep learning framework's architecture and instruction strategy, utilizing transfer learning as well as domain adaption techniques, and investigating hardware-based acceleration and compression strategies. These methodologies offer useful insights into the long-term evolution of deep learning-based AMC systems.

5.6 SUSTAINABLE DEVELOPMENT GOALS

The UN General Assembly established the Sustainable Development Goals, also known as the SDGs, in 2015 as a worldwide call to action to eradicate poverty, safeguard the planet, and guarantee that all people experience peace and prosperity by 2030. The SDGs address a wide variety of concerns, including poverty, hunger, and health, as well as the promotion of gender equality, education, and long-term development.
Our study "Automatic Modulation Classification Using Deep Neural Network" has the potential to contribute to a number of SDGs. Several of the SDGs to which this initiative may contribute are:

Figure 5.1: SDG 9

Figure 5.2: SDG 11

1. Industry, Innovation, and Infrastructure (SDG 9) – By enhancing the reliability and effectiveness of wireless communication networks through improved modulation categorization, this project may contribute to the advancement of technological advancement in communications and technology for the communication sector.
2. Sustainable Cities and Communities (SDG 11) - By allowing more efficient and dependable wireless connectivity, this initiative may assist to enhance city and community infrastructure.

Overall, this initiative has the potential to contribute to a number of SDGs, notably those concerning innovation, infrastructure, and sustainable development.

Chapter 6

CONCLUSION

To conclude we presented a deep learning-based approach for automatic modulation classification using our own collected dataset, starting by denoising the dataset using Savitzky Golay Filter which helped us remove unwanted noise and improve the quality of our signal. We propose two deep learning architectures, which proved to be effective at low SNR levels. This method has the potential to be applied in other signal-processing tasks where both time and frequency features are important. Experimental results reveal that by approach 1, classification accuracy reaches **97%** and with approach 2 **98%** at 5dB and 20dB SNR levels.

6.1 FUTURE WORKS

Currently, our system uses a limited dataset to train and test the model. Future work includes a plan to incorporate more data from different sources to improve the model's accuracy and robustness. We have used a sequential neural network consisting of CNNs, LSTMs, and Autoencoders. However, there are several other architectures that we can explore, such as Recurrent Neural Networks (RNNs), Transformer Networks, and Attention Mechanisms. These architectures may have different strengths and weaknesses, and we need to evaluate their performance on our dataset. Deep learning models have several hyperparameters that need to be tuned to achieve optimal performance. In the future, we plan to explore different hyperparameter settings and perform a comprehensive grid search to find the best hyperparameter configuration. Data pre-processing is crucial for deep learning models. In the future, we plan to explore different data pre-processing techniques such as normalization, data augmentation, and feature extraction to improve the model's accuracy. Transfer learning is a technique that allows us to use pre-trained models for a new task with minimal training data. In the future, we plan to investigate the potential of transfer learning for AMC and explore the use of pre-trained models for feature extraction. Deep learning models are often considered black boxes, which makes it challenging to understand how the model makes decisions. In the future, we plan to investigate different techniques for enhancing interpretability, such as GAN, and transfer learning to make a better model. We would also like to implement our work on hardware devices such as RTL-SDR in the future if available. Automatic modulation classification using deep learning is an exciting area of research that has the potential to revolutionize wireless communication systems. By exploring the areas mentioned above, we hope to develop an accurate, robust, and interpretable AMC system that can be used in various applications.

References

[1] O. A. Dobre, A. Abdi, Y. Bar-Ness, and W. Su, "Survey of automatic modulation classification techniques: classical approaches and new trends," *IET communications*, vol. 1, no. 2, pp. 137–156, 2007.

[2] Y. Xu, D. Li, Z. Wang, Q. Guo, and W. Xiang, "A deep learning method based on convolutional neural network for automatic modulation classification of wireless signals," *Wireless Networks*, vol. 25, pp. 3735–3746, 2019.

[3] S. Chen, B. Shen, X. Wang, and S.-J. Yoo, "A strong machine learning classifier and decision stumps based hybrid adaboost classification algorithm for cognitive radios," *Sensors*, vol. 19, no. 23, 2019. [Online]. Available: https://www.mdpi.com/1424-8220/19/23/5077

[4] Z. Zhu and A. K. Nandi, *Automatic modulation classification: principles, algorithms and applications*. John Wiley & Sons, 2015.

[5] Z. Zhang, C. Wang, C. Gan, S. Sun, and M. Wang, "Automatic modulation classification using convolutional neural network with features fusion of spwvd and bjd," *IEEE Transactions on Signal and Information Processing over Networks*, vol. 5, no. 3, pp. 469–478, 2019.

[6] R. Sahay, C. G. Brinton, and D. J. Love, "Frequency-based automated modulation classification in the presence of adversaries," in *ICC 2021 - IEEE International Conference on Communications*, 2021, pp. 1–6.

[7] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[8] Z. Liu, J. Li, Z. Shen, G. Huang, S. Yan, and C. Zhang, "Learning efficient convolutional networks through network slimming," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2736–2744.

[9] T. J. O'Shea, T. Roy, and T. C. Clancy, "Over-the-air deep learning based radio signal classification," *IEEE Journal of Selected Topics in Signal Processing*, vol. 12, no. 1, pp. 168–179, 2018.

[10] W.-N. Hsu, Y. Zhang, A. Lee, J. Glass *et al.*, "Exploiting depth and highway connections in convolutional recurrent deep neural networks for speech recognition," *cell*, vol. 50, no. 1, 2016.

[11] N. E. West and T. O'shea, "Deep architectures for modulation recognition," in *2017 IEEE International Symposium on Dynamic Spectrum Access Networks (DySPAN)*. IEEE, 2017, pp. 1–6.

[12] T. J. O'Shea, J. Corgan, and T. C. Clancy, "Convolutional radio modulation recognition networks," in *Engineering Applications of Neural Networks: 17th International Conference, EANN 2016, Aberdeen, UK, September 2-5, 2016, Proceedings 17*. Springer, 2016, pp. 213–226.

[13] M. R. Bahloul, M. Z. Yusoff, A.-H. Abdel-Aty, M. N. M. Saad, and M. Al-Jemeli, "Modulation classification for mimo systems: State of the art and research directions," *Chaos, Solitons & Fractals*, vol. 89, pp. 497–505, 2016.

[14] D. Hong, Z. Zhang, and X. Xu, "Automatic modulation classification using recurrent neural networks," in *2017 3rd IEEE International Conference on Computer and Communications (ICCC)*, 2017, pp. 695–700.

[15] F. Meng, P. Chen, L. Wu, and X. Wang, "Automatic modulation classification: A deep learning enabled approach," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 11, pp. 10 760–10 772, 2018.

[16] S. Ramjee, S. Ju, D. Yang, X. Liu, A. E. Gamal, and Y. C. Eldar, “Fast deep learning for automatic modulation classification,” *arXiv preprint arXiv:1901.05850*, 2019.

[17] H. Zhang, Y. Wang, L. Xu, T. Aaron Gulliver, and C. Cao, “Automatic modulation classification using a deep multi-stream neural network,” *IEEE Access*, vol. 8, pp. 43 888–43 897, 2020.

[18] Y. Kumar, M. Sheoran, G. Jajoo, and S. K. Yadav, “Automatic modulation classification based on constellation density using deep learning,” *IEEE Communications Letters*, vol. 24, no. 6, pp. 1275–1278, 2020.

[19] T. Huynh-The, Q.-V. Pham, T.-V. Nguyen, T. T. Nguyen, R. Ruby, M. Zeng, and D.-S. Kim, “Automatic modulation classification: A deep architecture survey,” *IEEE Access*, vol. 9, pp. 142 950–142 971, 2021.

[20] D. Zhang, “Wavelet transform,” in *Fundamentals of image data mining*. Springer, 2019, pp. 35–44.

[21] L. Chun-Lin, “A tutorial of the wavelet transform,” *NTUEE, Taiwan*, vol. 21, p. 22, 2010.

[22] R. W. Schafer, “What is a savitzky-golay filter?[lecture notes],” *IEEE Signal processing magazine*, vol. 28, no. 4, pp. 111–117, 2011.

[23] Z. Zhang, H. Luo, C. Wang, C. Gan, and Y. Xiang, “Automatic modulation classification using cnn-lstm based dual-stream structure,” *IEEE Transactions on Vehicular Technology*, vol. 69, no. 11, pp. 13 521–13 531, 2020.

[24] W. Tang, G. Long, L. Liu, T. Zhou, J. Jiang, and M. Blumenstein, “Rethinking 1d-cnn for time series classification: A stronger baseline,” *arXiv preprint arXiv:2002.10061*, pp. 1–7, 2020.

[25] M. Azizjon, A. Jumabek, and W. Kim, “1d cnn based network intrusion detection with normalization on imbalanced data,” in *2020 International Conference on Artificial Intelligence in Information and Communication (ICAIIC)*. IEEE, 2020, pp. 218–224.

[26] Riverbank Computing Limited, “Pyqt5 reference guide,” n.d. [Online]. Available: https://www.riverbankcomputing.com/static/Docs/PyQt5/

[27] ZetCode, “Pyqt5 tutorial,” 2021. [Online]. Available: https://zetcode.com/gui/pyqt5/

[28] Python.org, “Pyqt5,” n.d. [Online]. Available: https://wiki.python.org/moin/PyQt5

[29] baoboa, “Pyqt5 examples,” n.d. [Online]. Available: https://github.com/baoboa/pyqt5/tree/master/examples

[30] M. Summerfield, *Rapid GUI Programming with Python and Qt: The Definitive Guide to PyQt Programming*. Prentice Hall, 2008.

[31] S. Wei, Z. Wang, Z. Sun, F. Liao, Z. Li, L. Zou, and H. Mi, “A family of automatic modulation classification models based on domain knowledge for various platforms,” *Electronics*, vol. 12, no. 8, 2023. [Online]. Available: https://www.mdpi.com/2079-9292/12/8/1820

[32] B. Jdid, K. Hassan, I. Dayoub, W. H. Lim, and M. Mokayef, “Machine learning based automatic modulation recognition for wireless communications: A comprehensive survey,” *IEEE Access*, vol. 9, pp. 57 851–57 873, 2021.

[33] X. Feng, J. Yang, F. Luo, J. Chen, and X. Zhong, “Automatic modulation recognition by support vector machines using wavelet kernel,” *Journal of Physics: Conference Series*, vol. 48, p. 1264, 10 2006.

[34] F. Wang, S. Huang, H. Wang, and C. Yang, "Automatic modulation classification exploiting hybrid machine learning network," *mathematical Problems in engineering*, vol. 2018, pp. 1–14, 2018.

[35] F. Wang, S. Huang, and C. Liang, "Automatic jamming modulation classification exploiting convolutional neural network for cognitive radar," *Mathematical Problems in Engineering*, vol. 2020, pp. 1–12, 2020.

[36] W. Xie, S. Hu, C. Yu, P. Zhu, X. Peng, and J. Ouyang, "Deep learning in digital modulation recognition using high order cumulants," *IEEE Access*, vol. 7, pp. 63 760–63 766, 2019.

[37] Y. Kim, J. Lee, and H. Kim, "Hardware-based always-on heap memory safety," in *2020 53rd Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)*, 2020, pp. 1153–1166.