AUTOMATIC MODULATION CLASSIFICATION USING DEEP LEARNING

Payal Pote
19095074, B. Tech
Department of Electronics Engineering
Indian Institute of Technology(BHU)
Varanasi-221005
potepayal.umesh.ece19@itbhu.ac.in

Ajasra Gupta
19095006, B. Tech
Department of Electronics Engineering
Indian Institute of Technology(BHU)
Varanasi-221005
ajasra.gupta.ece19@itbhu.ac.in

Ayush Kumar Shaw
19095023, B. Tech
Department of Electronics Engineering
Indian Institute of Technology(BHU)
Varanasi-221005
ayush.kumarshaw.ece19@itbhu.ac.in

Supervisor: Dr. Amritanshu Pandey
Associate Professor
Department of Electronics Engineering
Indian Institute of Technology(BHU)
Varanasi-221005
apandey.ece@itbhu.ac.in

ABSTRACT

Signal detection and modulation classification are essential for successful transmission of the information. Current methods for AMC use either a feature-based approach or a likelihood based approach. While the likelihood-based approach can be very accurate, it also demands a high computational complexity. Classifiers in the feature-based approach need diverse features to be extracted from the incoming signal. We present an end-to-end deep learning based approach to AMC that takes in the modulated signal as its input and can detect the modulation scheme used in it, without any kind of prior feature extraction. The DL classifier can also include a wide range of modulation schemes, without any significant increase in its complexity.

Keywords- automatic modulation classification, neural networks, communication systems

1 INTRODUCTION

Every communication system consists of three basic elements, a transmitter, a channel, and a receiver. When a source of information produces a signal containing a message, the transmitter will convert it to a suitable form matching the channel properties. The signal is then propagated over the channel to the receiver, which is located at a different place than the transmitter. However, due to the channel imperfections, the received signal is distorted. Fading and attenuation effects appear, and diverse noises are added to the transmitted signal. The receiver reconstructs the signal, so that an end-user gets a recognizable form of the original message.Having an accurate classifier as a part of a communication system is crucial. With incorrect classification, the demodulator will not be able to use a correct demodulation method and the entire transmission fails. The numerous conditions affecting the channel such as fading, Doppler shift, AWGN and many others are challenging for the classifier. It is necessary to consider them when designing the classifier for its practical applications. The knowledge of many modulation types is important as well and it should operate with limited knowledge of the channel scenarios. Another thing to consider is the complexity as it affects the hardware choices as well as the processing time of the computation.

Therefore the key to completing the transmission of the signal and recovering the captured message is a robust, versatile, computationally efficient, and accurate classifier. With deep learning approaches, we can build an end-to-end model for automatic modulation classification which can work efficiently even without any knowledge of the parameters of the signal.

2 BACKGROUND

There are currently two approaches used for classifying radio modulations, namely likelihood-based (LB) and feature-based (FB) approach.

Likelihood based Approach

In this method, a likelihood is evaluated for each modulation hypothesis with observed signal samples. In the next step, a conclusion of the classification decision is made by comparing the likelihood functions of different modulation hypotheses.

This method provides optimal performance for the classification. But as the number of unknown parameters increases, obtaining an exact analytic solution for the decision function becomes difficult.

Feature based Approach

The feature-based approach consists of two main parts, namely feature-extraction and a classification. There are no rules for selecting the right features, but they should be sensitive to the modulations and insensitive to noisy channels and varying SNR. Spectral, statistical, spectrum and constellation shape features are the most commonly used features. After the features are extracted, they are passed to a classifier, which often uses machine learning. Some of the common classifier approaches are decision trees, artificial neural networks, support vector machines, or k-nearest neighbors. The complexity of this approach depends on the feature extraction process and can be computationally expensive depending on the type of features chosen for extraction.

3 DATASET

We have used one of the most popular publicly available dataset for modulation classification, **RadioML dataset(RADIOML 2016.10A)**. The signals in this dataset were created with GNU Radio software and are synthetic. Channel model blocks with sample rate offset, center frequency offset, selective fading, and additive white Gaussian noise(AWGN) were included to represent real-life settings. As a last step, the data was scaled to unity energy in preparation for further use in the machine learning sector.

The data is represented as 2x128 vectors of in-phase and quadrature signals (I/Q) are used to represent the data. They were made for a range of signal-to-noise ratios (SNRs) ranging from -20 dB to 18 dB.

Our dataset consists of a total of 220,000 signal samples modulated using the following digital and analog modulation schemes:

- ➔ BPSK (Digital)
- ➔ QPSK (Digital)
- ➔ 8PSK (Digital)
- ➔ QAM16 (Digital)
- ➔ QAM64 (Digital)
- ➔ CPFSK (Digital)
- ➔ GFSK (Digital)
- ➔ PAM4 (Digital)
- ➔ AM-DSB (Analog)
- ➔ AM-SSB (Analog)
- ➔ WBFM (Analog)

This dataset is available in pickled python format at DeepSig, consisting of time-windowed examples and corresponding modulation class and SNR labels.

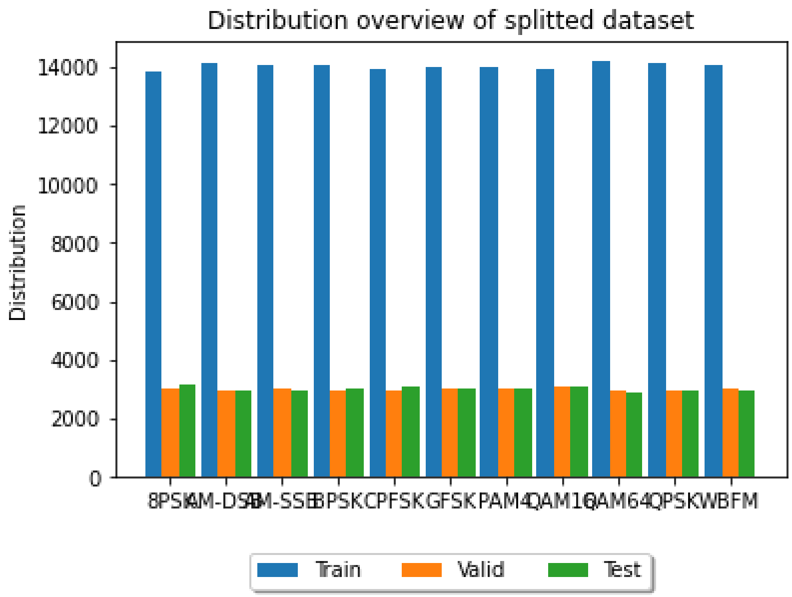


Figure1: Distribution of the various Modulation Schemes in our data

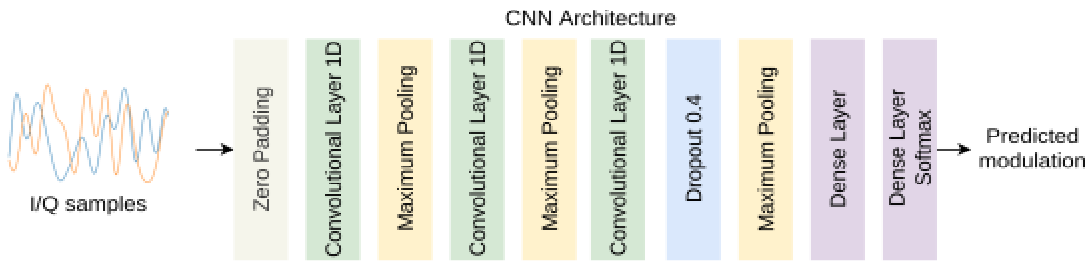
4 METHODS

We have used a deep learning approach to solve the problem of modulation classification. What is different about this approach is that the input signal can be directly passed to a DL model for classification. It can work without any feature extraction, and it does not need to know anything about the parameters of the signal. The classifier can include a huge number of modulations without any significant increase in computational complexity. The most common type of neural networks used for modulation classification are Convolutional or Recurrent neural networks. Here we have a combination of both these networks and compared its performance with a simpler CNN model.

CNN Architecture

CNN Models are generally used for image based tasks. However, due to their amazing performance on other deep learning tasks, we use a simple CNN architecture for modulation classification consisting of 1D Convolutional layers.

Our model architecture consists mainly of three 1D convolutional layers. These layers produce a tensor of outputs by convolving the kernel over a single spatial dimension. The layers are activated by a rectified linear activation function (ReLU) to provide a non-linearity to the layers. The maximum pooling layers down-samples the length of the sequence, by taking a maximum value over a spatial window, which length is given by a chosen pooling size. A dropout layer is added to prevent potential overfitting and the network is finished by adding two fully connected layers. The last dense layer is activated by the Softmax activation function. The Softmax function is commonly used in classification tasks, as it will represent the output of the final layer as probabilities of each class, where all the values add up to 1.

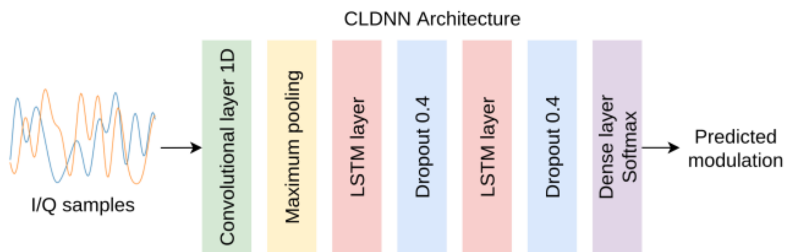



```
 Layer (type)                Output Shape              Param #
=================================================================
 input_1 (InputLayer)        [(None, 128, 2)]          0

 zero_padding1d (ZeroPadding  (None, 136, 2)           0
 1D)

 conv1d (Conv1D)             (None, 129, 50)           850

 max_pooling1d (MaxPooling1D  (None, 64, 50)           0
 )

 conv1d_1 (Conv1D)           (None, 57, 50)            20050

 max_pooling1d_1 (MaxPooling  (None, 28, 50)           0
 1D)

 conv1d_2 (Conv1D)           (None, 25, 50)            10050

 dropout (Dropout)           (None, 25, 50)            0

 max_pooling1d_2 (MaxPooling  (None, 12, 50)           0
 1D)

 flatten (Flatten)           (None, 600)               0

 dense (Dense)               (None, 70)                42070

 dense_1 (Dense)             (None, 11)                781

=================================================================
Total params: 73,801
Trainable params: 73,801
Non-trainable params: 0
```

Figure 2: CNN Architecture

CLDNN Architecture

LSTMs have been long used for sequential tasks and time series data like signals. They have been found to perform very well modeling the relationships in sequential data. Hence, we leverage the power of LSTMs combining it with Convolutional layers to form a CLDNN model. CLDNN architecture used here is composed of a convolutional and a maximum pooling layer, followed by two LSTM layers with dropout layers, which should prevent overfitting. The final layer is once again a dense layer activated with a Softmax function. This is a very simple architecture having both convolutional and recurrent layers.




```
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 input_1 (InputLayer)        [(None, 128, 2)]          0

 conv1d (Conv1D)             (None, 121, 64)           1088

 max_pooling1d (MaxPooling1D  (None, 60, 64)           0
 )

 lstm (LSTM)                 (None, 60, 64)            33024

 dropout (Dropout)           (None, 60, 64)            0

 lstm_1 (LSTM)               (None, 60, 64)            33024

 dropout_1 (Dropout)         (None, 60, 64)            0

 flatten (Flatten)           (None, 3840)              0

 dense (Dense)               (None, 11)                42251

=================================================================
Total params: 109,387
Trainable params: 109,387
Non-trainable params: 0
_________________________________________________________________
```

Figure 3: CLDNN Architecture

5 EXPERIMENTS AND RESULTS

The architectures were written in Python using a deep learning API Keras utilizing the free GPU offered by Google Colab. Both the architectures use the Adam optimizer, with a starting learning rate between 0.002 and 0.0007 (depending on the network). A ReduceLROnPlateau callback from the Keras API was included to reduce the learning rate size if the validation loss does not improve for 3-5 epochs. The training is stopped with an EarlyStopping callback if the learning rate does not improve for 8 epochs, and a ModelCheckpoint callback then saves a model with the smallest validation loss. Using these settings, we trained both our models for 50 epochs using a batch size of 128 on Google Colab.

CNN

Our CNN model attained a maximum accuracy of **86.23% for SNR value of 8dB**. The overall average accuracy of our model on the test dataset is 56.45%. It is evident from the accuracy plot and confusion matrix that most of the mis-classification occurs at lower values of SNR, where the signal contains a lot of noise. Also, we see that the confusion score for QAM16 and QAM64 is higher than others. This is primarily because, looking at the constellation diagrams of the 2 modulation schemes, we can see that they have the same constellation points at the core, and hence, their signals contain similar features. From the confusion plot at SNR=8dB, it is evident that the model performs almost perfectly in classifying signals except for telling the QAMs apart, and there is also some confusion in the analog modulation schemes WBFM and AM-DSB.

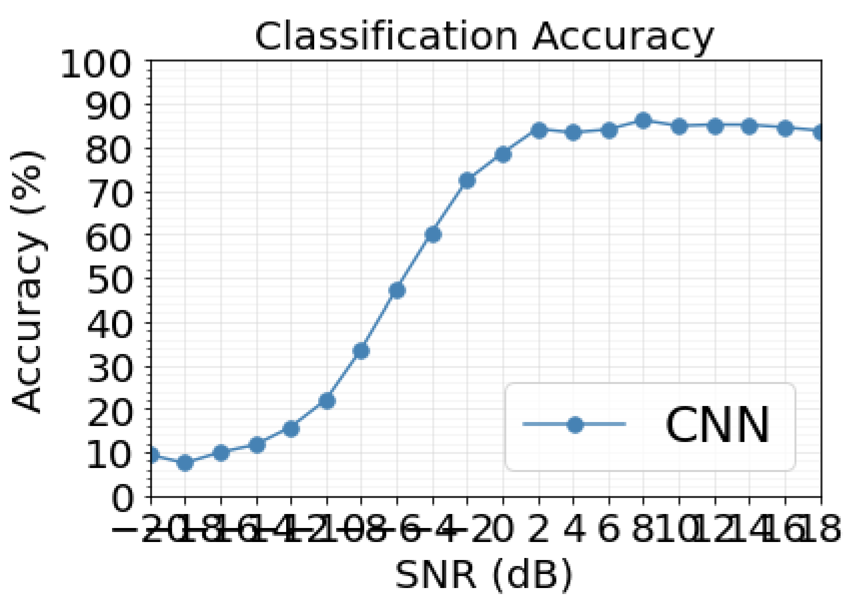
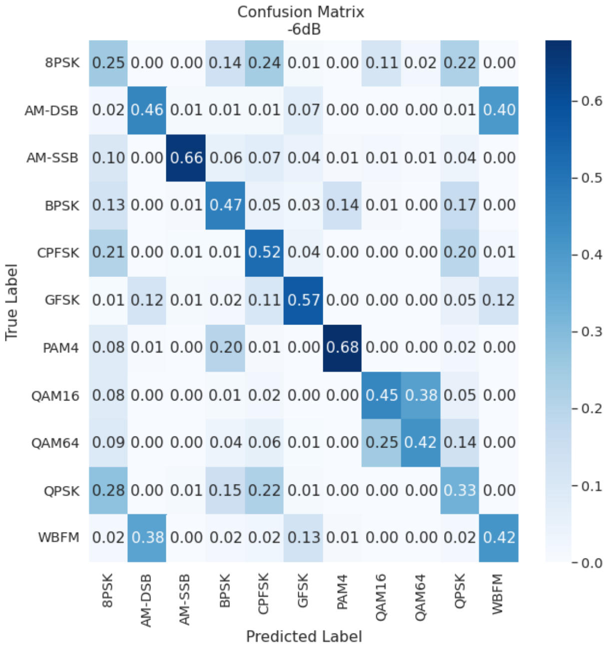
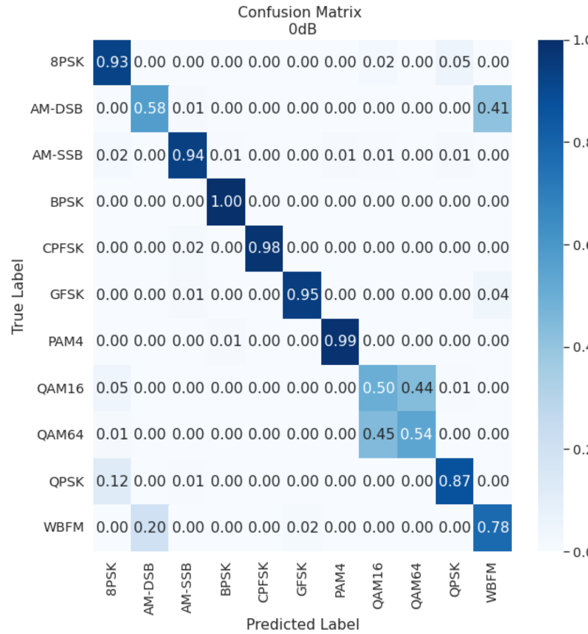


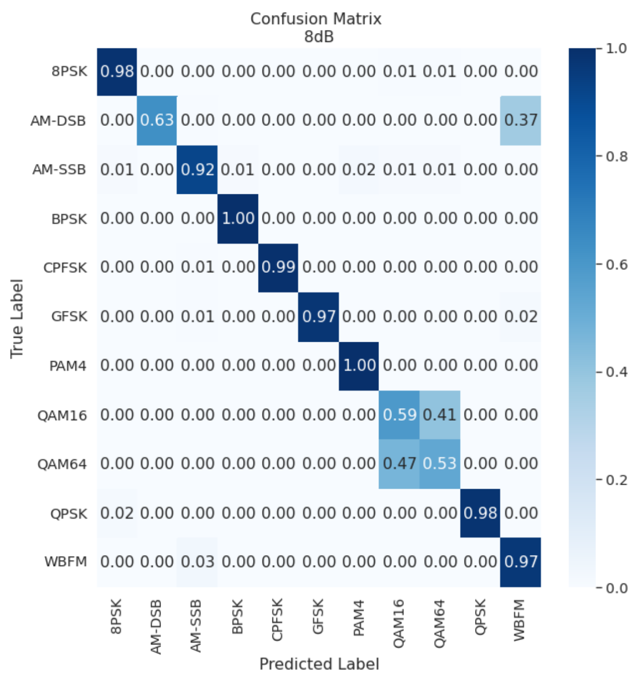
Figure 4: Accuracy vs SNR (CNN)



(a)



(b)



(c)

Figure 5: Confusion Matrix for (a) SNR=-6dB, (b) SNR=0dB, (c) SNR=8dB

CLDNN

The CLDNN model attained a maximum accuracy of **92.05% for SNR value of 8dB**. The overall average accuracy of our model on the test dataset is 61.04%. The overall performance of the CLDNN model is found to be better than the CNN model. The confusion plots for the CLDNN model are almost similar to that of the CNN at lower values of SNR. However, at higher values of SNR, the CLDNN model performs much better in telling the QAMs apart, with an accuracy of 85%.

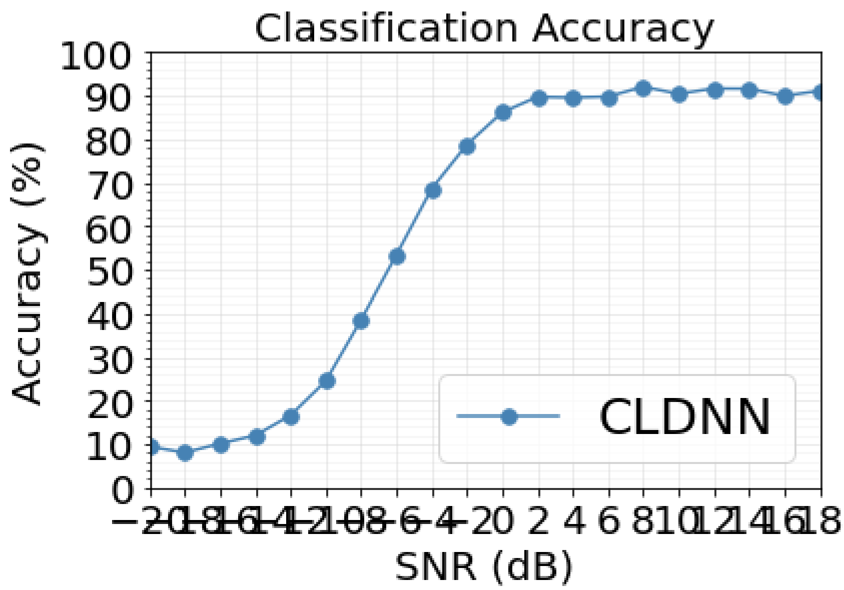
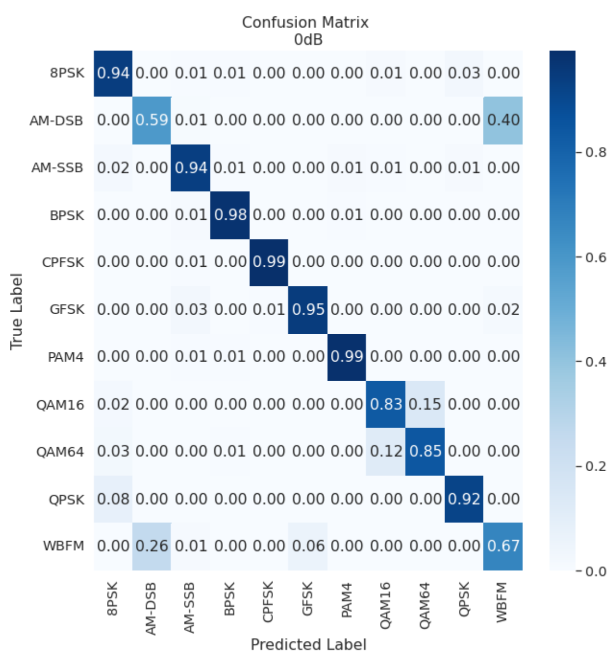
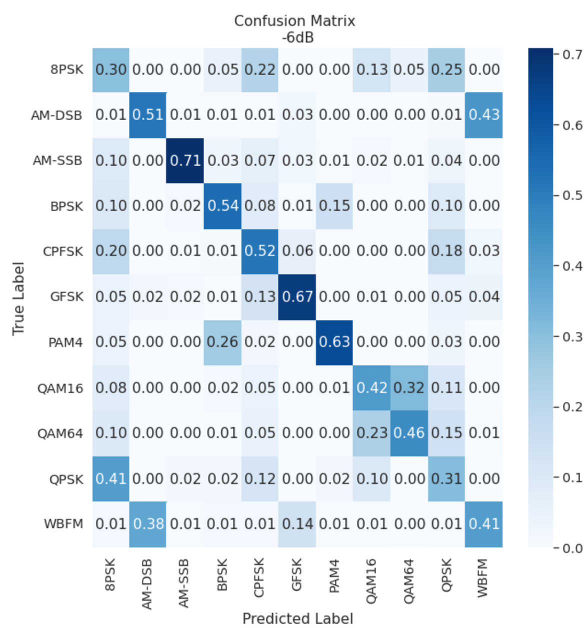


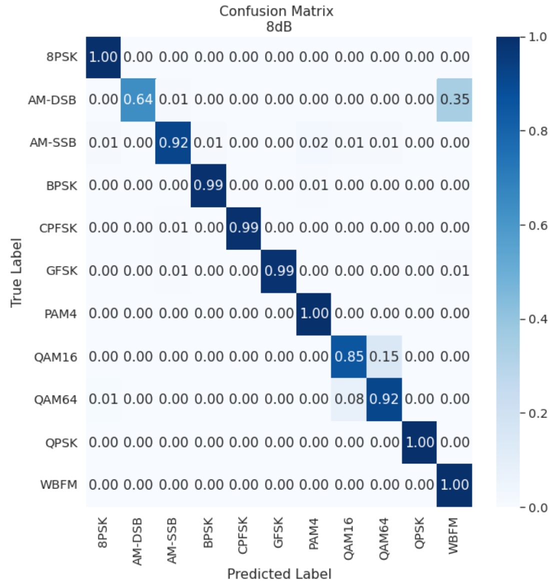
Figure 6: Accuracy vs SNR (CLDNN)



(a)



(b)



(c)

Figure 7: Confusion Matrix for (a) SNR=-6dB, (b) SNR=0dB, (c) SNR=8dB

6 CONCLUSION / FUTURE WORK

With the new 5G technology, there is a focus on deep learning applications in communication systems. The classical approaches to modulation classification like likelihood based approaches and feature based approaches have a big trade-off between the computational complexity, performance, and the number of modulation classes. Deep learning based approaches can work directly with the received data without any previous knowledge or feature extractions. They can be also trained to classify a large number of modulation types without any significant increase in the computational complexity. They can be easily upgraded to hold additional modulation classes or signals from another environment.

Our model architectures use 20 to 50 times less number of parameters than the deep learning models being used for such tasks. Our trained model has a very small size(1.3MB for CLDNN model and 921KB for CNN model) and can be easily deployed on embedded systems. These models can classify signals, at very high speeds and without requiring a lot of computational power.

Thus, we can say that the deep learning approach shows a lot of potential to be used for automatic modulation classification. The DL classifiers can be trained on robust datasets (consisting of real world data along with synthetic data). As future work, we can explore other architectures like transformers which are one of the most recent developments in the deep learning community. We can explore the use of transformer models, which are only used in Natural Language Processing tasks as of now, in signal processing.

7 CODE LINK

Our code can be found at this GitHub repository.

8 REFERENCES

[1] Automatic Modulation Classification Based on Deep Learning for Software-Defined Radio
[2] Machine Learning remakes radio
[3] A Comparative Study between CNN, LSTM, and CLDNN Models in The Context of Radio Modulation Classification
[4] A Survey of Automatic Modulation Classification Techniques: Classical Approaches and New Trends Communications
[5] Fast Deep Learning for Automatic Modulation Classification
[6] Deep Neural Network Architectures for Modulation Classification
[7] Radio Modulation Classification Using Deep Learning Architectures