A Project Report On

Hybrid CNN-SVM architecture for automatic modulation classification for beyond 5Gcommunication

Submitted in partial fulfilment of the requirements for the award of the degree
Of

BACHELOR OF TECHNOLOGY

In

ELECTRONICS AND COMMUNICATION ENGINEERING

Submitted by:

Divyansh Jain (16116019)

Shivanshu Raj Shrivastava (16116063)

Under the guidance of:

Dr. Meenakshi Rawat

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY ROORKEE
ROORKEE – 247667 (INDIA)

June, 2020

CANDIDATE'S DECLARATION

I hereby certify that the work presented in this report entitled "Hybrid CNN-SVM architecture for automatic modulation classification for beyond 5Gcommunication" in partial fulfilment of the requirements for the award of the degree of Bachelor of Technology is carried out by me during the period of July 2019 to June 2020 under the supervision of **Dr. Meenakshi Rawat**.

This work has not been submitted elsewhere for the award of a degree/diploma/certificate.

Date: 05 June 2020

Place: Roorkee

Divyansh Jain

16116019

Divyansh Jain

……………………..

Shivanshu Raj Shrivastava

16116063

……………………..

Hybrid CNN-SVM architecture for automatic modulation classification for beyond 5G communication

Divyansh Jain
Enrollment Number 16116019
Department of Electronics
and Communication Engineering
Indian Institute of Technology Roorkee
Roorkee, Uttarakhand 247667
Email: djain@ec.iitr.ac.in

Shivanshu Raj Shrivastava
Enrollment Number 16116063
Department of Electronics
and Communication Engineering
Indian Institute of Technology Roorkee
Roorkee, Uttarakhand 247667
Email: sshrivastava@ec.iitr.ac.in

***Abstract*—A novel CNN-SVM hybrid architecture is proposed, which automatically detects the modulation scheme of a captured radio signal. It does so by extracting various discriminating features of the RF signals. It is demonstrated that the proposed method gives an accuracy, which is above 90 percent at -5 dB SNR for 24 types of modulation schemes. The proposed CNN feature extractor utilizes Lightweight, one-dimensional sparse filters to learn the hierarchical features of the raw signal. These extracted features are then fed into a Support Vector Machine (SVM) functioning as a classifier. The use of simple 1-D CNN layers and SVM results in a lower number of parameters and lower complexity, which makes our proposed architecture fit for real-time analysis of 5G signals.**

Keywords :Automatic Modulation Classification, Cognitive radio, Deep Learning, Neural networks, Signal Processing

I. Introduction

Automatic modulation Classification (AMC) is an integral part of non-cooperative communication systems. The recognition of the channel type coding and the modulation scheme is necessary for the receiver to demodulate and recover the transmitted information from the received signal. In a system with AMC at the receiver's end, the transmitter has the ability to freely choose the modulation type of the signals depending on various channels or data related variables in order to transmit the data in the best way possible.

This makes AMC attract the attention of many researchers in the field. Extensive work has been done in the past 20years. The traditional AMC algorithms can be broadly classified into two categories:

1) : Feature Based (FB) [1]

2) : Likelihood Based (LB) [2]

Though Feature-based AMC algorithms obtain sub-optimal results as compared to Likelihood-based AMC, they are generally preferred due to their smaller computational complexity and do not require any prior knowledge of channel or the transmitter.

Hence, a great deal is done on feature extraction of the radio signals. FB methods mostly consist of two main stages, Feature extracting and classifying. Numerous feature extractors have been proposed, such as Instantaneous features [3], which are extracted from instantaneous amplitude, phase, or frequency in the time domain.

Another set of transformation-based features, which are calculated by taking Fourier and wavelet transformation, provide more distinctive, noise immune features [4]. More such examples of conventional feature extractors use algorithms such as High-order cumulant features [5] or spectral correlation function [6].

The problem with most of these methods is that most of them are handcrafted and require a high degree of domain knowledge and empirical trial to fine-tune. These handcrafted methods also give a degraded performance on varied types of signals, which is further degraded by the introduction of noise and heavy channel distortion.

Considering these factors, we propose the use of deep learning (CNN architecture) based feature extraction, which can automatically extract features and achieve superior results. An AMC method based on Deep Belief Networks was proposed in [6].

Most of the conventional methods are either too complex to give real-time classification or require estimation of the SNR value to give results. Since the SNR estimation is oftentimes an inaccurate process or the SNR might change rapidly due to variations in the channel.

Current methods lack the ability to generalize; to overcome this, a CNN-SVM hybrid model is proposed, which exploits the powerful feature learning abilities of the CNN network to feed the distinct and noise-tolerant features into the SVM across the SNR range. The salient features of the proposed method are as follows:

- 24 different types of modulation schemes are used to train the model with SNR range of [-20dB,20dB]
- To reduce the reliance on manual fine tuning the raw signal is directly fed into the DL network for feature extraction This method provides better classification

accuracy at lower SNR values as compared to other traditional methods. 1-D convolution layers are used to process the data so as to reduce the complexity without compromising the accuracy.

- SVM is used as the classifier as it is considered to have some advantages over the traditional softmax layer especially when the number of samples is limited. Major advantage being its ability to provide superior generalization ability and at the same time SVM also reduces the complexity as compared to a softmax layer.

II. Proposed Model

We propose a hybrid model of two different types of superior classifiers: Convolution Neural Network and Support Vector Machine. This type of hybrid method has proven accuracy in different types of pattern recognition tasks.

The proposed architecture utilizes 1-D convolution and thus obtaining a lightweight and easy to train the model with fewer parameters. It also employs an attention layer to weight and assembles feature maps to select the most discriminating features, and this makes the extracted features more easily distinguishable. Thus, it further reduces the computational burden of the classifiers by making the features more distinct and representative.

By replacing the last soft-max layer, which traditionally acted as the classifier, we propose the use of an SVM based classifier.

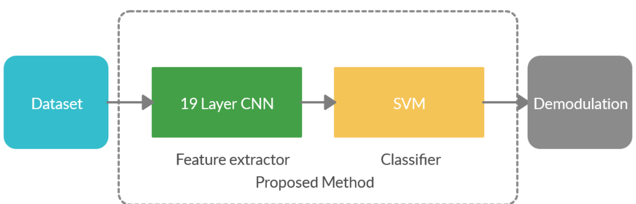

Fig. 1. Block diagram of the proposed Hybrid AMC

A. CNN based feature extraction

The CNN that we propose consists of 16 convolution layers, 4 pooling layers, 1 global pooling instead of a fully connected layer, and an attention layer with the structure loosely based upon VGG-19.The detailed structure is detailed in the following table:

Layer Number	Layer Type	Parameters	Layer Number	Layer type	Parameters
1	Conv1D	(32,4)	12	Conv1D	(50,4)
2	Conv1D	(32,4)	13	Conv1D	(50,4)
3	Pooling	(2,2)	14	Conv1D	(50,4)
4	Conv1D	(64,4)	15	Conv1D	(50,4)
5	Conv1D	(64,4)	16	Pooling	(2,2)
6	Pooling	(2,2)	17	Conv1D	(64,4)
7	Conv1D	(45,4)	18	Conv1D	(64,4)
8	Conv1D	(45,4)	19	Conv1D	(64,4)
9	Conv1D	(45,4)	20	Conv1D	(64,4)
10	Conv1D	(45,4)	21	Global Pooling	None
11	Pooling	(2,2)	22	Attention Layer	None

Fig. 2. Structure of CNN.

The Parameters denote:

- Conv1D : (Filters, Kernel Size)
- Pooling : (Pool size, Strides)

1) 1D Convolution Layer: Conventional CNNs are good at handling and processing 2D data and have given tremendous results in many fields of computer vision.

Thus previously proposed methods [7] usually convert signals into 2D features such as Grid constellation Matrix [7] or rearranged as 2D feature maps [8]. Since raw radio signals are inherently one dimensional, in this model, we propose to use a 1-D sparse convolution layer for superior efficient and distinctive feature learning. We also use relatively small filters in each layer so as to decrease the memory cost and to decrease the complexity and hence computing cost.

Let us say that the l^{th} layer of our Neural Network is a convolutional layer, and also let N_s, N_k^l, L_s^l and L_k^l are Number of inputs, kernal numbers, input length, and kernal length of l^{th} layer, respectively. The convolution operation[9] in the l^{th} layer is described in below equation.

$$h_k^l = f(x^l * W_k^l + b_k^l) \tag{1}$$

$$(x^l * W_k^l)(i) = \sum_{a=-\infty}^{\infty} x(a) W_k^l(i-a) \tag{2}$$

Where $x \in R^{N_s} * L_s^l$ is representing all the set of inputs, $W \in R^{N_k^l} * L_k^l$ is representing all the se of kernels, and $b \in R^{N_s}$ is basically the bias for every output. Representation of the output of the k^{th} (k = 1,2,...,N_k^l) kernal is denoted by equation (2), and $x^l * W_k^l$ represents the convolution between x^l and W_k^l.

Assuming that the output length is L_o^l, the output $h^l \in R^{N_k} * L_o^l$ represents the set of all outputs, which is also called as the feature map. $f(x)$ is the activation function to act as a layer for achieving nonlinear output matching, in general the activation function is chosen as either sigmoid or tanh function.

In our work, we have taken the exponential linear unit (ELU) [10] as the activation function, which is given by

$$f(x) = \begin{cases} \alpha(e^x - 1) & x < 0 \\ x & x \geq 0 \end{cases} \tag{3}$$

The Exponential Linear Unit (ELU) is used as the activation function that is derived from the most commonly used Rectified Linear Unit (ReLU). The ELU accelerates the convergence speed while overcoming the gradient vanishing.

2) Max-Pooling & Global Average Pooling: Another vital layer on CNN is the pooling layer. As mentioned, several convolutions are performed by the convolution layer to produce a set of outputs. The mode of activation for each of which is (ELU) nonlinear.

The pooling function then further modifies the output of the layer by replacing the output of the net at a particular location with a summary statistic of the nearby location. The max-pooling layer used in the proposed model reports the maximum output in the pooling window [6].

Assuming the general output of a convolutional layer h_l is max-pooled, then the output h_{l+1} will be given by

$$h_k^{l+1} = max\{h_k^l[m^{l+1}(i-1)+1], h_k^l[m^{l+1}(i-1)+2], \\ ..., h_k^l[m^{l+1}(i-1)+L_p]\} \tag{4}$$

Where, $i \leq (L_o^l - L_p^{l+1})/m^{l+1} + 1$ and L_p^{l+1} is denoting the length of the each pooling window. Also, m^{l+1} is denoting the margin between any two adjacent pooling windows, this margin is also known as stride.

The global average layer is applied after the last convolution layer and before the attention layer. The output of the layer is the average of each feature map in the input.

Similarly we have taken the assumption that output from the former convolutional layer is h^l, containing the output from N_k^l kernals.

The required output from global average pooling h_k^l is given by

$$h_k^L = 1/L_o^l \sum_{i=1}^{L_o^l} h_k^l(i), \\ k = 1, 2, .., N_k^l \tag{5}$$

3) Attention Layer: After repeatedly performing hit and trial with various different configurations, we found that the overall performance of our 1D convolution network can be significantly improved by adding an Attention Layer, which will also help in the selection of the most discriminative features and to provide filtering maps a better weighing and assembling scheme.

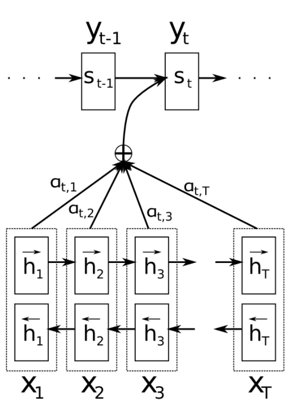

Fig. 3. Attention Layer Model

As shown in the Fig. 3 & Fig. 4, the sequence of annotations represented as h_1, h_2, h_3,..., $H = h_{Ts}$ is generated by Bidirectional LSTM. All the vectors i.e. h_1, h_2, h_3,..., etc used in [11] are nothing but he concatenation of forwarding hidden states and backward hidden states in the encoder.

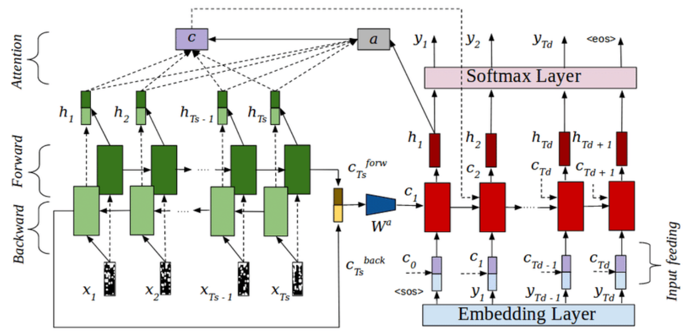

Fig. 4. Detailed Attention Layer Model

In more simple words, all the vectors h_1, h_2, h_3,..., $H = h_{Ts}$ are a vector representation of the T_s number of words in the input sentence. In a much simpler encoder and decoder model, the only useful state is the last state of the encoder LSTM which is used as the context vector.

Keeping this basic model in mind attention mechanism is found useful in many deep neural networks when long-term dependencies are needed to be utilized. Similarly, we explored various dependencies using the saliency mechanism of attention in features. That is with the help of attention mechanism we are able to find some salient features which are strictly more valuable than the other features so far. Those more important features consequently get the larger weight to assemble features which in turn reduce the parameters of the network and also improve the discriminative features.

The synthesis vector C in Fig. 4 is given by

$$c = \sum_{j=1}^{B_f} \alpha_j s_j \tag{6}$$

Th weghts α_j are then evaluated using $e_j = g(s_j w)$, where g(x) is known as Tansig function and $W \in R^{N_f} * 1$ is representing the set of all the learnable parameters. The use of Tansig function is to amplify the difference among neurons as compared to Sigmoid function. To calculate the weights in the attention layer we use the following equation

$$\alpha_j = exp(e_j) / \sum_{k=1}^{B_f} exp(e_k \tag{7}$$

A Schematic representation of the attention layer mechanism is shown in Fig. 3 & Fig. 4. The hidden state sequence vectors are given as inputs to the trainable system producing a probability vector α. The calculation of vector is based upon the weighted average with weighting given by α.

B. SVM based Classifier

The support vector machine(SVM) is a discriminative classifier and has been used extensively with superior results for numerous pattern recognition/classification related tasks. The SVM is used to determine an optimal decision surface or the separating hyper-plane by mapping the sample points into a high-dimensional feature space and then categorizing the data using a non-linear transformation ϕ, even when the data are linearly inseparable.

As visualized later, the extracted features of the different modulation schemes are spatially separable; thus, we apply a transformation of the input data by mapping it into a higher dimensional feature space by the use of a non-linear operator.

In the proposed model, Radial Basis Function(RBF) kernel SVM is used instead of the last layer of conventional CNN models to perform the classification of modulation type of the input RF signals.

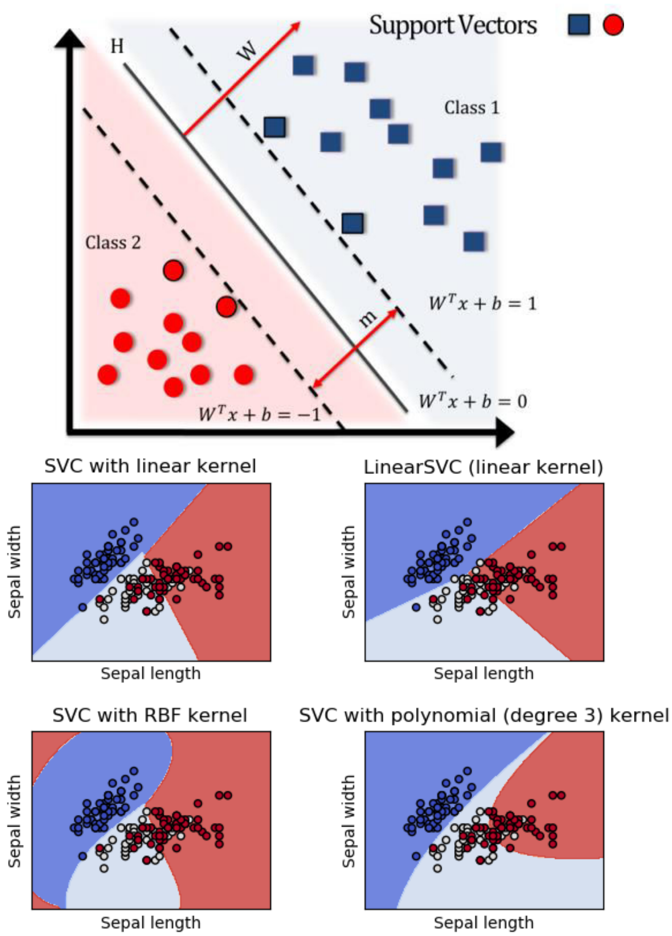

Fig. 5. Functioning of the Support Vector Machine; (a) hyper-plane Classification, (b)Different classification functions of SVM

III. Dataset Generation

In this paper, we generated a new dataset by using the tools provided by GNU Radios, a details description of the tools can be found in [11]. 24 types of different of modulation schemes were generated using both the analog and digital modulators.

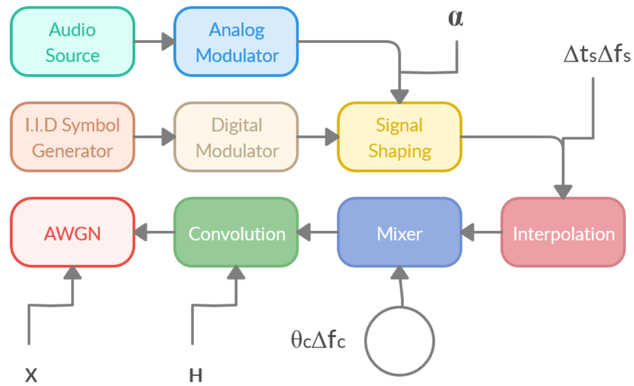

Fig. 6. System for dataset signal generation with synthetic channel impairment modeling.

To simulate the input data that is modulated, we independently pick a value for the variables shown in the table. Thus the resulting signal is uncorrelated and randomly initialized for each 1024 sample long instance.

Several different propagation scenarios were considered while generating the dataset for this project. The wireless signals were generated with different channel parameters, as depicted in Fig. 6. A range of roll-off values α with a Root-raised cosine pulse shaping filter were used to shape the Digital Signals.

Random Variable	Distribution
α	$U(0.1, 0.4)$
Δ_t	$U(0, 16)$
Δf_s	$N(0, \sigma_{clk})$
θ_c	$U(0, 2\pi)$
Δf_c	$N(0, \sigma_{clk})$
H	$\Sigma_i \delta(t - \text{Rayleigh}_i(\tau))$

Fig. 7. Illustrates the range of the values for H, for different delay spreads, the channel impulse response envelope, τ) =[0,0.5,1.0,2.0],corresponding to various levels of multi path fading in Rayleigh Fading environments.

Data was stored in hdf5 format as floating point values, with 2.7 million examples with each one being 1024 samples long. And contains random signal samples for varied (-20dB to 20dB) SNR levels for the model to be trained independent of the SNR levels.

The types of modulation schemes generated for the database are as follows:

OOK, 8PSK, QPSK, 16QAM, OOK, AM(SSB-SC), 8ASK, 4ASK,QPSK, BPSK, 16APSK, 32PSK, 16PSK, 8PSK, 16APSK, AM(DSB-SC), 32APSK, 64APSK, 128APSK, 16QAM, 256QAM, 128QAM, 64QAM, 32QAM, FM, GMSK, OFDM, AM(SSB-WC), AM(SB-SC), AM(SSB-SC), AM(SB-WC),FM, OQPSK, GMSK

The primary impairments that lead to channel loses and distortions are:

- Carrier Frequency Offset (CFO): frequency offset and carrier phase due to Doppler effect and disparate local oscillators.
- Symbol Rate Offset (SRO) Time dilation and symbol clock offset due to motion and out of sync clock sources.
- Delay Spread: Reflection, diffusion and diffraction along the multiple emissions part leads to non-impulsive delay spread
- Thermal Noise (AWGN): Due to the physical device sensitivity in the transmitters and receivers, additive white noise is added to the signal.

Care was taken to simulate the effect of the these channel distortions in the final signals to achieve a close to real world accuracies on the AMC. GNU radio Dynamic Channel Model hierarchical block was used for simulation.

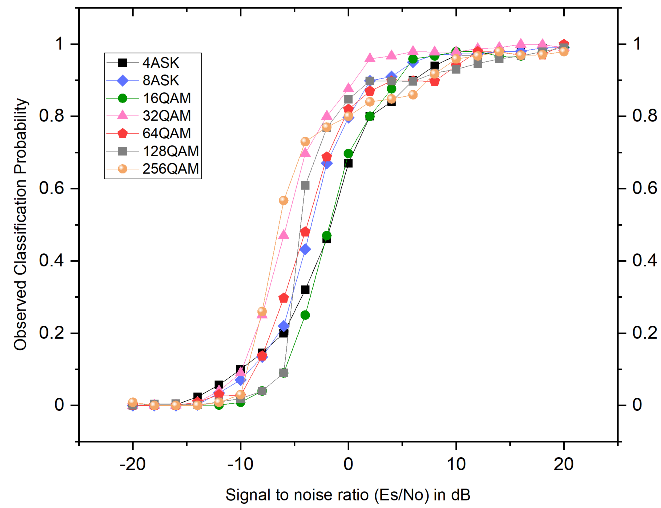

Fig. 8. QAM/ASK-style modulation performance

IV. Training

The training of the prepared CNN model was done on Amazon's AWS Sagemaker, which is a fully managed service that provides the ability to build, train, and deploy machine learning models on the cloud. The Sagemaker provides a preconfigured Jupyter Notebook instance. It comes with ready support for the necessary libraries Tensorflow and Keras with full CUDA capabilities for Graphics Processing Unit (GPU) assisted Traning and classification. The "ml.p3.2xlarge" accelerated instance was selected based on our need for high throughput of 32bit floating-point calculations. It had the following configuration:

- CPU : 8 Core vCPU
  Each vCPU is a thread of an AMD EPYC CPU core.
- GPU: Nvidia V100 Tensor Core
  Equipped with 16GB of Ram, the V100 comes with 640 Tensor Cores and is capable of calculations at a speed of 100+TFLOPS. It provides with the necessary acceleration for large number of floating point calculations in CNN.
- Ram : 61 GB
  Since data access speeds from RAMs is up to 10 times faster as compared to some of the fastest secondary storage device, having enough RAM the dataset was loaded in bigger batches into the RAM to same significant data fetching overheads.

Experimentation was done with the sample size used for training the model to study the effect it has on the accuracy of the model. The Performance vs. training set size is plotted in Fig. 10. Finally, the entire dataset was randomly distributed into two groups for training (80%) and validation (20%). With a batch size of 1024, 80 epochs were trained on SageMaker. Each Epoch took an average of about 45 minutes to train.

We trained the model with a different number of samples to study the effect of the Sample size on the training accuracy of the model; the obtained results are plotted in Fig. 8,9 & 10.

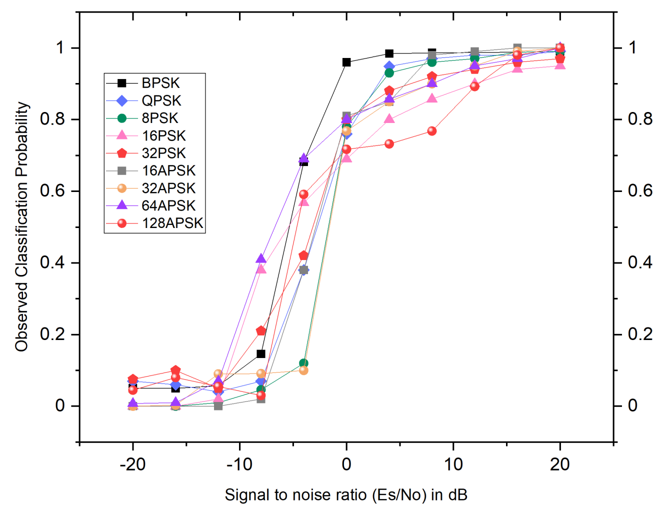

Fig. 9. PSK-style modulation performance

V. Results & Feature learning with CNN

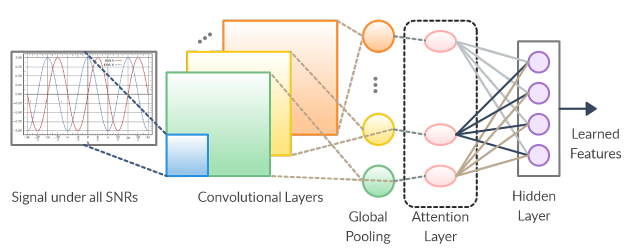

Fig. 11. Feature Learning with CNN

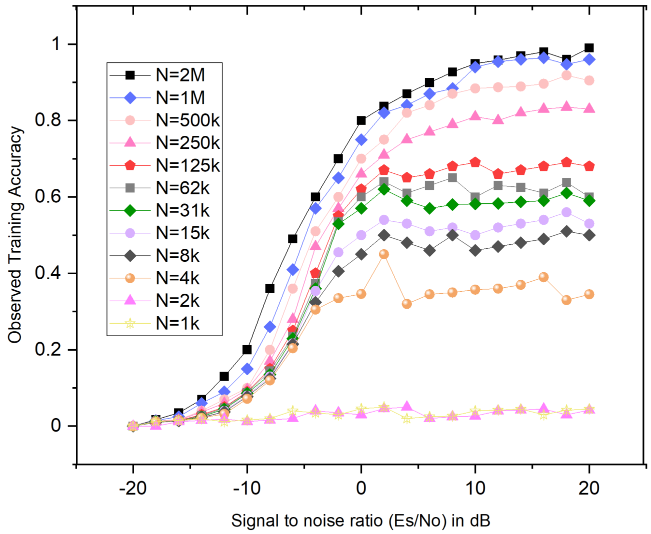

Fig. 10. Performance vs training set size (N) with = 1024.

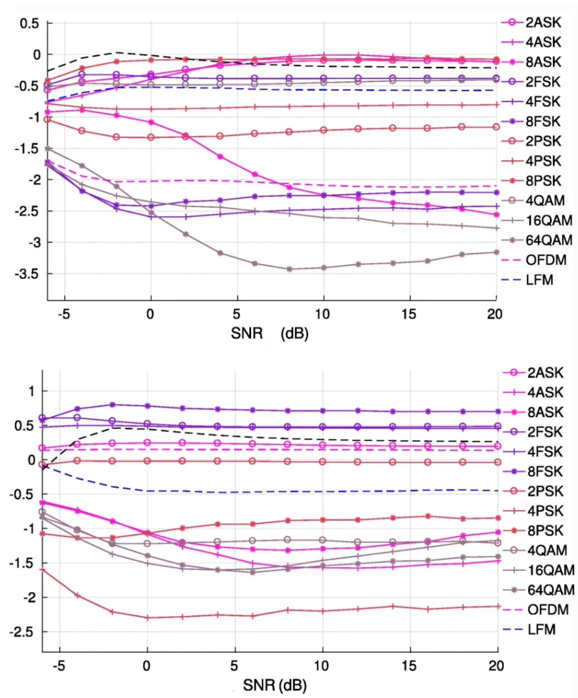

Fig. 12. Learned features as outputs from two neurons of the hidden layer. Distinctive values can be recognised for each individual modulation scheme.

In traditional DL based AMC models, CNN networks are used as classifiers. However, due to a large number of finely tunable parameters, the CNN network can function as a useful feature extractor. Since only the softmax layer acts as the classifier, thus the inputs into the softmax layer can be considered as features.

Finally, the output is reduced to a four-dimensional network with the use of a four neuron hidden layer after the attention layer. We have plotted the output of each of the neurons for a visual representation of how the data is differentiable for the different modulation schemes. Then finally, a linear support vector machine (SVM) classifies the modulation scheme with the use of these 4 extracted features. The results are plotted in Fig. 10. and Fig. 11.

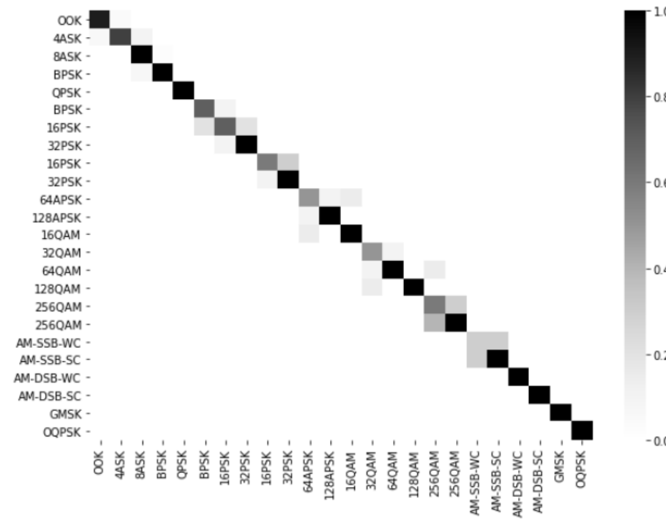

Fig. 13. The confusion matrix for the proposed model at 0dB SNR

VI. Conclusion

We proposed and trained a CNN-SVM based hybrid Automatic Modulation Classifier for raw radio signals. In total, 24 different modulations were classified with an accuracy of greater than 90% for all signals with SNR values as low as -5 dB. The robust feature extraction ability of the Convolution Neural Network was demonstrated, especially under low SNR conditions. The utilization of a 1-D convolution layer in the CNN and attention layer enables the proposed network to process the raw signals without any preprocessing directly and extract efficient and discriminative features for classification. Several intermediate stage data were extracted and visualized to demonstrate its workings and give a sense of its robustness classification accuracy. The overall classification time of the proposed architecture was .1ms, which is comparable to the 5G standard. In the future, this type of lightweight CNN architecture can be used to realize more advanced concepts for better AMC performance like the Siamese networks.

Acknowledgment

This paper and the research behind it would not have been possible without the exceptional support of our supervisor, we would like to thank Dr. Meenakshi Rawat for her constant support.

References

[1] Z. Zhang, R. Han, C. Wang, G. Cui, and W. Wang, “Automatic modulation classification using compressive sensing based on high-order cumulants,” Proc.ICAIT 2017, pp. 120-126, Chengdu, China, Nov. 2017.

[2] V. Gouldieff, J. Palicot, and S. Daumont, “Blind modulation classificationfor cognitive satellite in the spectral coexistence context,”IEEE Trans.Signal Process., vol. 65, no. 12, pp. 3204–3217, Jun. 2017.

[3] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient based learning applied to document recognition," Proc. IEEE, vol. 86, no. 11, pp. 2278-2324, Nov. 1998.

[4] A. Swami and B. M. Sadler, "Hierarchical digital modulation classificationusing cumulants,"IEEE Trans. Commun., vol. 48, no. 3, pp. 416–429,Mar. 2000.

[5] A. K. Nandi and E. E. Azzouz, "Algorithms for automatic modulation-recognition of communication signals,"IEEE Trans. Commun., vol. 46,no. 4, pp. 431–436, Apr. 1998.

[6] O. A. Dobre, Y. Bar-Ness, and W. Su, "Higher-order cyclic cumulants forhigh order modulation classification," inProc. IEEE Mil. Commun. Conf.,vol. 1, Oct. 2003, pp. 112–117.

[7] S. Huang, Y. Jiang, Y. Gao, Z. Feng, and P. Zhang, "Automatic modulationclassification using contrastive fully convolutional network,"IEEE Wire-less Commun. Lett., vol. 8, no. 4, pp. 1044–1047, Aug. 2019.

[8] J. H. Lee, K. Y. Kim, and Y. Shin, "Feature image-based automaticmodulation classification method using CNN algorithm," inProc. Int.Conf. Artif. Intell. Inf. Commun. (ICAIIC), 2019, pp. 1–4

[9] J. Lv, L. Zhang, and X. Teng, "A modulation classification based onSVM," inProc. IEEE Int. Conf. Opt. Commun. Netw., Sep. 2017, pp. 1–3.

[10] T. J. O'Shea, J. Corgan, and T. C. Clancy, "Convolutional radio modula-tion recognition networks," 2016,arXiv:1602.04105. [Online]. Available:https://arxiv.org/abs/1602.04105

[11] Z. Zhang, Y. Li, and X. Zhu, "A method for modulation recognition basedon entropy features and random forest," inProc. IEEE Int. Conf. Softw.Qual., Rel. Secur. Companion (QRS-C), Jul. 2017, pp. 243–246.