Automatic Modulation Classification with ANN and CNN

Pavithran Palanichamy, *Electrical and Computer Engineering, UBC,*

Abstract

In recent years, neural networks have received much attention due to its superior performance in classifying data with complex structure. In this paper, the feasibility and effectiveness of employing deep learning algorithms for automatic recognition of the modulation type of received wireless communication signals is investigated. Several artificial neural network and convolutional neural network architectures were developed and compared for efficacy of radio modulation classification. The test results show that the statistical features from the input signal data brings huge performance improvement for artificial neural networks. On the other hand, convolutional neural networks achieve performance that exceeds that of expert-based approaches. Results show that blind temporal learning using deep convolutional neural networks is a strong candidate approach for automatic modulation classification especially at low signal to noise ratio whereas artificial neural networks with carefully selected features provides comparable performance at a very lower runtime.

Index Terms

Machine learning, automatic modulation classification, artificial neural network, convolutional networks

I. Introduction

AUTOMATIC modulation classification plays an important role in modern wireless communication. Automatic modulation classification detects the modulation type of received signals to guarantee that the signals can be correctly demodulated and that the transmitted message can be accurately recovered. It has found significant roles in military, civil, intelligence, and security applications [**5**]. Analogue Modulations (e.g., AM and FM) and Digital Modulations (e.g., PSK and QAM) transform baseband message signals (of lower frequency) into modulated bandpass signals (of higher frequency) using a carrier signal for the purpose of enhancing the signal's immunity against noise and extending the transmission range. Different modulations require different hardware configurations and bandwidth allocations. Meanwhile, they provide different levels of noise immunity, data rate, and robustness in various transmission channels. In order to demodulate the modulated signals and to recover the transmitted message, the receiving end of the system must be equipped with the knowledge of the modulation type. In modern applications multiple modulation types can be employed by a signal transmitter to control the data rate, to control the bandwidth usage, and to guarantee the integrity of the message. Though the pool of modulation types is known both to transmitting and receiving ends, the selection of the modulation type is adaptive and may not be known at the receiving end. Therefore, an automatic modulation classification mechanism is required for the receiving end to select the correct demodulation approach in order to guarantee that the message can be successfully recovered [**5**].

There are four major requirements for an automatic modulation classifier. First, the modulation classifier needs to be accurate. Accuracy of detection is crucial, as inaccurate classification leads to misinterpretation of transmitted signals. Second, the modulation classifier needs to be robust. The wireless communication systems are often hindered by many physical phenomena like noise, signal fading, scattering etc., A good classifier must be immune to noises and other factors that affect the signal and provide reliable results. Third, the classifier needs to be computationally efficient since most AMC are used in real time environment. Fourth, the classifier needs to be versatile enough to detect various types of modulations.

Given the importance of AMC in various military and civilian communication applications, there has been a large amount of research work dedicated to the problem of AMC in a wide variety of settings. The nature of the problem creates multiple dimensions in its solutions and inspires continuous contribution from generations of researchers. Various feature-based AMC schemes have been proposed in the past such as decision tree method which uses the cumulants and cyclic moments of the time domain signal waveforms and support vector machines (SVM) [**1**]. In recent years, deep neural network (DNN) [**1**, **2**], a convolutional neural network (CNN) [**3**, **4**] and a recurrent neural network (RNN) [**6**] for the AMC that have been studied and resulted in improved AMC performance. However, since the AMC process should be operated in real time, the computational complexity must be considered low enough. On the other hand, the DNN algorithm can learn complex structures from various data and shows good performance for various machine-learning problems in recent years [**2**].

In this paper, a quantitative comparison is carried out between artificial neural networks (ANN) and convolutional neural network (CNN). While both architectures result in good performance, factors such as run-time, accuracy and performance in low SNR values are compared to decide the best classifier for this task. The selected features [**1**] extracted from the high dimensional inputs are used for training the ANN whereas CNN uses temporal learning using deep convolutional neural networks. Using the softmax layer at the output layer, both classifiers produces the probability of each modulation class at each node. Several models are trained and compared for accuracy and performance and the best model for ANN and CNN is reported.

II. Dataset

RADIOML 2016.10A dataset from by Deep-Sig is used for training and testing of both ANN and CNN[**4**]. Training with real data is important and valuable especially to achieve a reliable wireless communication. The dataset consists of synthetically generated radio signals in a real system, with well characterized transmit parameters identical to a real world signal. The dataset consists of 11 modulations which contains 8 digital and 3 analog modulation, which are widely used in wireless communications systems. These consist of BPSK, QPSK, 8PSK, 16QAM, 64QAM, BFSK, CPFSK, and PAM4 for digital modulations, and WB-FM, AM-SSB, and AM-DSB for analog modulations. The dataset has the size 220,000 x 2 x 128, consisting of 220,000 entries, each with an array of size 2 x 128. Each array represents the samples of about 128 μs of a received waveform. The samples of the signal waveforms are complex-valued, and hence they have been stored as real and imaginary parts, and therefore we have arrays of size 2 x 128 in the data set. The labels of the downloaded dataset contain two parameters: the modulation technique used one of the 11 possible modulation techniques, and the signal-to-noise ratio (SNR) value (one of [-20;-18;-16;-14;-12;-10;-8;-6;-4;-2; 0; 2; 4; 6; 8; 10; 12; 14; 16; 18], so 20 possible SNR values).

III. Artificial Neural Networks

The dataset was trained against several candidate neural networks. Three different models (ANN1, ANN2, ANN3) of dense neural network were designed and experimented. The best model for modulation classification was chosen after comparing the performance of the three models. Comparison is made in terms of overall accuracy at different SNR levels. 110,000 samples are used for both training and validation, and another 110,000 samples are used for the testing the performance of the trained machine, as it has been done in the implementation provided by DeepSig. Early stopping and dropout has used to prevent over-fitting. Training is conducted using a categorical cross entropy loss function and using the famous Adam optimiser. The network was implemented and trained in Keras running on top of TensorFlow on an Intel Core i5-8250U CPU @1.60Ghz and 8 GB RAM without any GPUs.

A. Features for ANN

This subsection summarises the features used to train ANN2 and ANN3. The features were previously implemented in [**1**] and resulted in good performance for 5 different modulation types. These features exhibit sufficiently nice separation between the empirical distributions of each modulation class. In this paper we use the same set of features to classify eleven different modulated signals.The modulated baseband signal is represented as a complex number $\mathrm{a}[i] = a_I[n] + \mathrm{j}a_Q[n]$. The first feature is the ratio of in-phase component and quadrature component signal power

$$F1 = \frac{\sum_n a_Q^2[n]}{\sum_n a_1^2[n]}$$

where $a_Q[n]$ and $a_I[n]$ are the in-phase and quadrature samples for the complex baseband signal. The second feature is the standard deviation of the direct instantaneous phase

$$F2 = \sqrt{\frac{1}{C}\left(\sum_{a_n[i]>a_t} \varphi_{NL}^2[i]\right) - \left(\frac{1}{C}\sum_{a_n[i]>a_t} \varphi_{NL}[i]\right)^2}$$

where $a_n[i] = \frac{a[i]}{mean(a[i])}$, $\varphi_{NL}[i]$ is the instantaneous phase at time instant t and C is the total number of samples and at is the threshold. The third feature is the standard deviation of the absolute value of the non-linear component of the instantaneous phase

$$F3 = \sqrt{\frac{1}{C}\left(\sum_{a_n[i]>a_t} \varphi_{NL}^2[i]\right) - \left(\frac{1}{C}\sum_{a_n[i]>a_t} |\varphi_{NL}[i]|\right)^2}$$

The next feature selected is the standard deviation of the absolute value of the normalized instantaneous amplitude of the simulated signal

$$F4 = \sqrt{\frac{1}{N}\left(\sum_{i=1}^{N} a_{cn}^2[i]\right) - \left(\frac{1}{N}\sum_{i=1}^{N} |a_{cn}[i]|\right)^2}$$

where $\alpha_{cn}[i] = \frac{a[i]}{mean(a[i])} - 1$ and N is the number of the samples for $\alpha_{cn}[i]$. The fifth feature is the standard deviation of the absolute normalized centered instantaneous frequency for the signal segment

$$F5 = \sqrt{\frac{1}{N_s}\left(\sum_{i=1}^{N} f_N^2(i)\right) - \left(\frac{1}{N}\sum_{i=1}^{N}|f_N(i)|\right)^2}$$

where $f_N(i)$ is the centered instantaneous frequency. The sixth feature is

$$\sigma_v = \sqrt{\frac{1}{N}\left(\sum_{i=1}^{N} a_v[i]^2\right) - \frac{1}{N}\left(\sum_{i=1}^{N}|a_v[i]|\right)^2}$$

where v is the normalized signal amplitude, i.e. $\sqrt{\frac{a[i]}{var(a[i])}} - 1$ the mixed order moments $v_{20}[14]$ is chosen as the seventh feature and the mean value of the signal magnitude is the eighth feature

$$F7 = \frac{M_{4,2}(y)}{M_{2,1}^2(y)} = \frac{E\left(|a[i]|^4\right)}{E\left(|a[i]|^2\right)}$$

$$F8 = \frac{1}{N}\sum_{n=1}^{N}|a[i]|$$

where $|a[i]|$ is the instantaneous amplitude. The ninth feature is the normalized square root value of sum of amplitude of signal samples and the tenth feature is the maximum value of power spectral density(PSD) of the normalized signal samples

$$F9 = \frac{\sqrt{\sum_{n=1}^{N}|a[i]|}}{N}$$

$$F10 = \frac{1}{N}\max\left|\mathrm{DFT}\left(a_{cn}[i]\right)\right|^2$$

Features eleven to seventeen are cumulants calculated with the equations below;

$$F11 = E\left[a^2[n]\right]$$

$$F12 = E\left[|a[n]|^2\right]$$

$$F13 = M_{41} - 3M_{20}^2$$

$$F14 = M_{41} - 3M_{20}M_{21}$$

$$F15 = M_{42} - |M_{20}|^2 - 2M_{21}^2$$

$$F16 = M_{63} - 6M_{20}M_{40} - 9M_{42}M_{21} + 18M_{20}^2M_{21} + 12M_{21}^3$$

$$F17 = M_{80} - 35M_{40}^2 - 28M_{60}M_{20} + 420M_{40} - 630M_{20}^4$$

where

$$M_{p+q,p} = E\left[a[n]^p a[n]^{*q}\right]$$

The eighteenth feature measures the tailedness of the distribution while the nineteenth feature represents the skewness, quantifying the asymmetry of the distribution

$$F18 = \left|\frac{E\left[(a - E[a])^4\right]}{E\left[(a - E[a])^2\right]^2}\right|$$

$$F19 = \left|\frac{E\left[(a - E[a])^3\right]}{E\left[(a - E[a])^2\right]^{3/2}}\right|$$

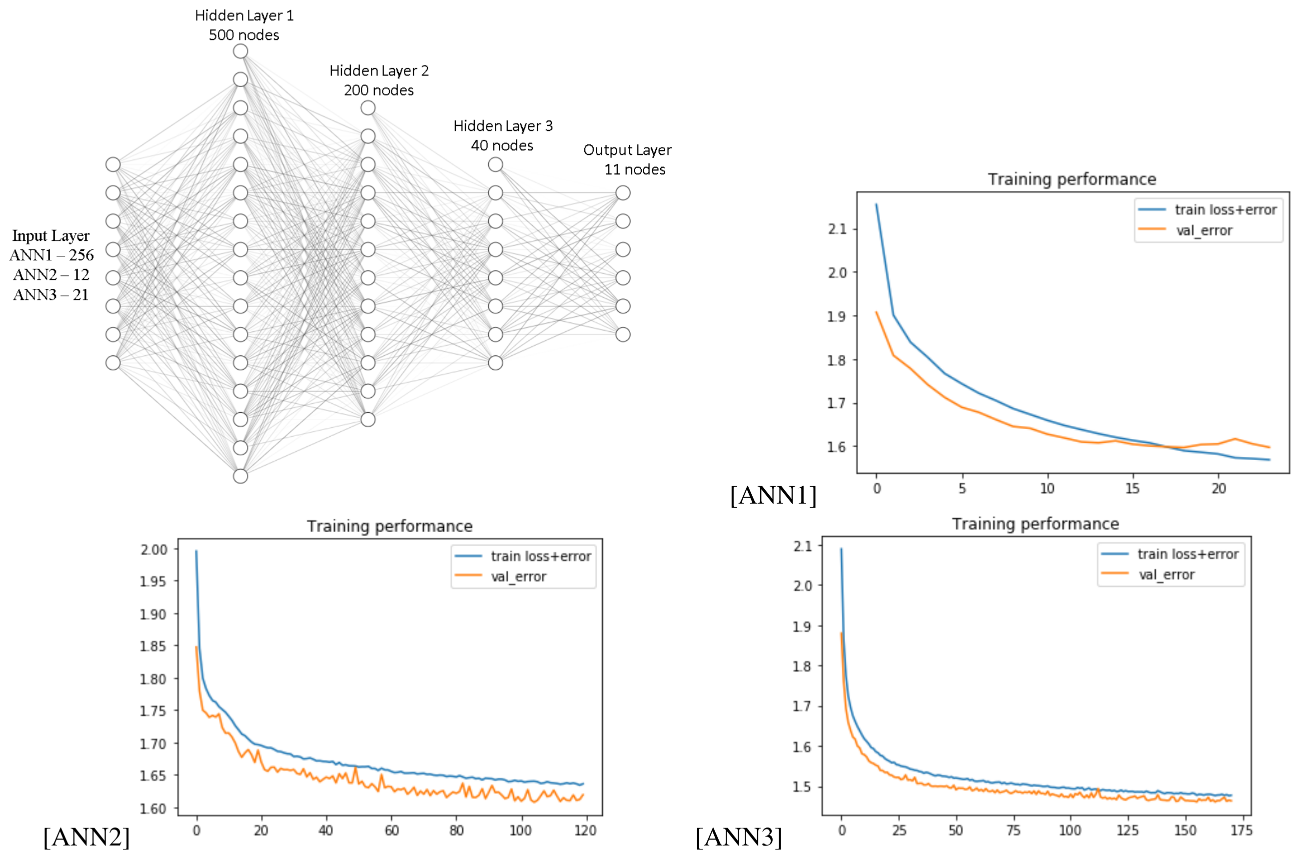


Fig. 1: Structure of artificial neural networks and training losses.

The twentieth feature is the peak-to-rms ratio and the final feature selected is the peak-to-average ratio

$$F20 = \frac{\max |a|^2}{\frac{1}{N}\sum_{n=1}^{N}(a[i])^2}$$

$$F21 = \frac{\max |a|}{\frac{1}{N}\sum_{n=1}^{N} a[i]}$$

B. Structure and configurations of ANN

Figure 1 depicts the general structure of the artificial neural network used in this paper. The only difference between the three networks lies in the input layer. All three models contain 3 hidden layers each of size 500, 200 and 40 with ReLu as the activation function. Bias is used in both input anf hidden layers. All the labels are one-hot encoded and since the number of classes is 11, the output layer has the same size with SoftMax activation function. The class with highest score in soft metric is chosen as a final decision. Early stopping is used to stop training when the model starts overfitting the data.

The first architecture ANN1 is the simplest among all the three models in terms of implementation. The input layer has 256 neurons which correspond to flattened vector of each input data. The in-phase and quadrature components are merged into a single vector. The second architecture ANN2 is similar to ANN1 except for the input layer. The input layer has a size of 12 which corresponds to the first 12 features described in the previous section. The all twenty one features are fed into the input layer after l2 normalization for each feature. The third architecture ANN3 uses all the 21 features with same number of neurons in the input layer.

	Metric	ANN1	ANN2	ANN3
Training data	Loss	1.28	1.62	1.45
	Accuracy	0.48	0.39	0.44
Validation data	Loss	1.56	1.59	1.45
	Accuracy	0.42	0.38	0.44
Test data	Loss	1.52	1.59	1.46
	Accuracy	0.42	0.39	0.44

TABLE I: Summary of performance metrics for ANN models

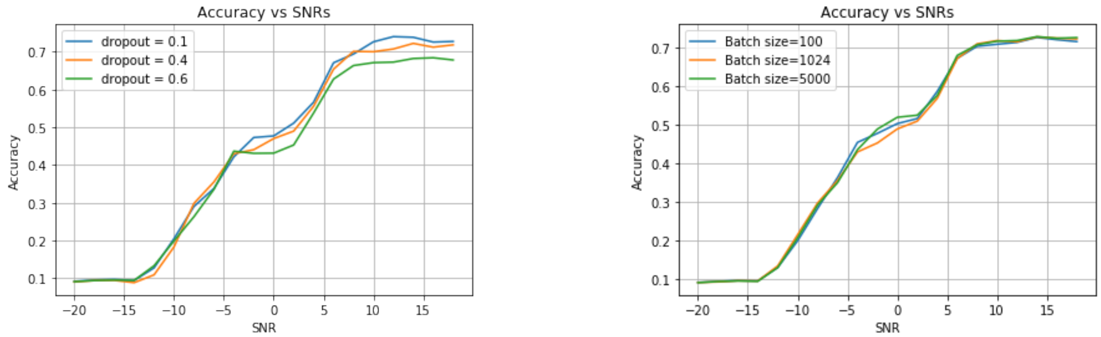


Fig. 2: Effect of dropout rate and batch size on performance of ANN3

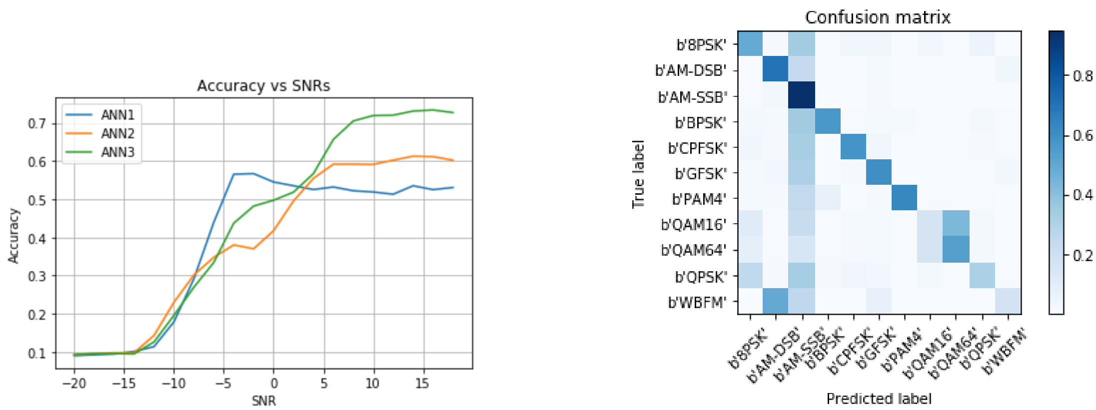


Fig. 3: Classification accuracy vs. SNR and confusion matrix of ANN3

We observe from Table. I that the ANN3 achieves higher classification accuracy and least validation error than the other networks (ANN1 and ANN2) over all scenarios considered. It has an overall validation accuracy of 44% while ANN1 and ANN2 have 42% and 39% accuracy. In terms of categorical cross entropy loss, ANN3 achieves the lowest value at 1.45 whereas ANN1 and ANN2 have much higher loss values of 1.56 and 1.59 respectively. Hence ANN3 is selected as the candidate network for modulation classification. Furthermore, training parameters such as batch size and dropout rate were experimented on ANN3 to further improve the performance.

C. Effect of Dropout and Batch size

Dropout forces a neural network to learn more robust features that are useful in conjunction with many different random subsets of the other neurons. Change in dropout rate has a significant effect on the performance of the dense neural network as shown in Figure 2.The network with dropout rates 0.1 and 0.4 have almost same performance except the high SNR regions. The final validation loss the networks with dropout rates 0.1,0.4 and 0.6 are 1.46, 1.49 and 1.52 respectively. Overall, the network with 0.1 dropout rate has best performance and least validation loss than the other dropout rates and hence is chosen as the final dropout value for the network ANN3.

The network ANN3 was experimented with batch sizes 100, 1024 and 5000. The validation losses for the networks were 1.462, 1.460 and 1.452 respectively. To better interpret the results the test accuracies of the three models with different batch sizes were compared in Figure. 2. In contrast to dropout rate effects, change in batch sizes has minimal effect on performance. But the time taken for training varies drastically. The network with batch size 1024 takes the least amount of time (14 min) to train, whereas the networks with batch sizes 100 and 5000 takes 109.1 and 26.7 mins. Clearly, the best choice of batch size is the one with least training time and hence 1024 is decided as the final batch size for the network ANN3.

D. Results for ANN

After carefully selecting the training parameters, ANN3 was trained with a batch size of 5000 and dropout rate of 0.2. The final test loss of ANN3 is 1.45 and the test accuracy is 44% . The difference is performance are predominantly in higher SNR values as shown in Figure. 3. The overall accuracy for the network at higher SNR values were over 70% for ANN3 whereas ANN1 and ANN2 has accuracies at the same level at 50% and 60%. Figure 3 shows the overall accuracy of the network in the

SNR/Label	8PSK	AM-DSB	AM-SSB	BPSK	CPFSK	GFSK	PAM4	QAM16	QAM64	QPSK	WBFM
-20.0	0.01	0.02	0.96	0.01	0.00	0.00	0.01	0.00	0.00	0.00	0.00
-18.0	0.00	0.02	0.97	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.00
-16.0	0.00	0.04	0.97	0.00	0.00	0.01	0.00	0.00	0.01	0.00	0.01
-14.0	0.02	0.05	0.95	0.00	0.00	0.00	0.00	0.00	0.06	0.01	0.01
-12.0	0.01	0.16	0.95	0.00	0.00	0.01	0.03	0.00	0.28	0.00	0.03
-10.0	0.01	0.37	0.95	0.00	0.00	0.03	0.11	0.00	0.64	0.01	0.11
-8.0	0.04	0.61	0.96	0.01	0.01	0.06	0.46	0.00	0.81	0.01	0.28
-6.0	0.13	0.57	0.96	0.08	0.04	0.20	0.63	0.00	0.86	0.04	0.40
-4.0	0.20	0.64	0.96	0.34	0.14	0.44	0.75	0.01	0.87	0.10	0.38
-2.0	0.14	0.70	0.96	0.56	0.17	0.63	0.90	0.08	0.82	0.10	0.34
0.0	0.02	0.74	0.95	0.75	0.12	0.74	0.95	0.14	0.78	0.08	0.45
2.0	0.02	0.94	0.96	0.80	0.12	0.86	0.93	0.16	0.74	0.11	0.16
4.0	0.28	0.96	0.96	0.82	0.23	0.89	0.92	0.20	0.72	0.24	0.18
6.0	0.53	0.88	0.96	0.89	0.61	0.92	0.91	0.25	0.74	0.32	0.43
8.0	0.67	0.93	0.95	0.96	0.84	0.94	0.94	0.24	0.70	0.34	0.29
10.0	0.74	0.99	0.93	0.98	0.88	0.93	0.92	0.20	0.75	0.33	0.29
12.0	0.76	1.00	0.96	0.98	0.94	0.90	0.91	0.22	0.71	0.30	0.28
14.0	0.74	1.00	0.97	0.98	0.91	0.93	0.93	0.19	0.73	0.32	0.29
16.0	0.71	1.00	0.97	0.98	0.93	0.92	0.91	0.22	0.71	0.32	0.29
18.0	0.75	1.00	0.95	0.97	0.91	0.92	0.92	0.25	0.72	0.30	0.30

TABLE II: Classification accuracy of ANN3 at different SNR levels for each class

form of a confusion matrix. At higher SNR levels certain classes namely AM-DSB, AM-SSB, BPSK, CPSK, GSFK, PAM4 were classified with accuracy of over 95%.

The table II shows the result of test data when applied to ANN3 network. It contains accuracy of each modulation class at different SNR levels. This table gives us an interesting insight of the classification accuracies of each individual modulation class. It can be noted that AM-SSB gets classified with high accuracy even at very low SNR values. Whereas QAM16, QPSK and WBFM have less than 30% accuracy even at very high SNR values. When SNR value is 18 dB 8PSK and QAM64 has relatively high accuracy of around 70% while AM-DSM, AM-SSB, BPSK, CPFSK, GFSK and PAM4 have over 90% accuracy. The overall accuracy is brought down due to very low accuracies of QAM16, QPSK and WBFM. The performance can be improved drastically, by selecting features that distinguish QAM16, QPSK and WBFM modulations well.

IV. Convolutional Neural Networks

Unlike artificial neural networks which relies on the features for learning, convolutional neural networks works on the basis of temporal learning to find the essential features on it's own. It treat the complex valued signal data as an input dimension of 2 real valued inputs and use r(t) as a set of 2xN vectors into a narrow 2D Convolutional Network where the orthogonal synchronously sampled In-Phase and Quadrature (I Q) samples make up this 2-wide dimension. In order to select the best model for classification, several candidate convolutional neural networks were trained and tested (CNN1,CNN2,CNN3). All the CNNs have a similar pattern, each network contains convolutional layers and dense fully connected layers at the end. Regularization is used to prevent over-fitting. The convolutional layers of the CNN use Dropout and early stopping is used to prevent over-fitting and to save time. Training is conducted using a categorical cross entropy loss function and using the famous Adam optimiser with batch size 1024. The network was implemented and trained in Keras running on top of TensorFlow on an Intel Core i5-8250U CPU @1.60 GHz and 8 GB RAM without any GPUs.

The best model for modulation classification was chosen after comparing the performance of the three models at different settings. 110,000 samples are used for both training and validation, and another 110,000 samples are used for the testing the performance of the trained machine, as it has been done in the implementation provided by DeepSig. These samples are uniformly distributed in SNR from -20 dB to +18 dB and tagged so that we can evaluate performance on specific subsets. Classification accuracy across different SNR values of the different training examples and classification accuracy of each modulation types were inspected to evaluate the performance of the different CNN models.

A. *Structure and configurations of CNN*

The architecture of CNN1 is similar to the one used in [**4**], which contains 2 convolutional layers and 2 dense layers as depicted in the Figure 4. The first parameter below each convolutional layer in the figure 4 represents the number of filters in that layer, while the second and third numbers show the size of each filter.CNN2 [**4**] identical but larger, containing 256 and 80 filters in first two convolutional layer and 256 neurons in the first dense layer. CNN3 is the largest among all the models with 4 convolutional layers and 2 dense layers. CNN3[**3**] as shown in the figure has two extra convoluional layers each with 256 and 80 filters respectively. The dense layers of CNN3 has 128 and 11 neurons.

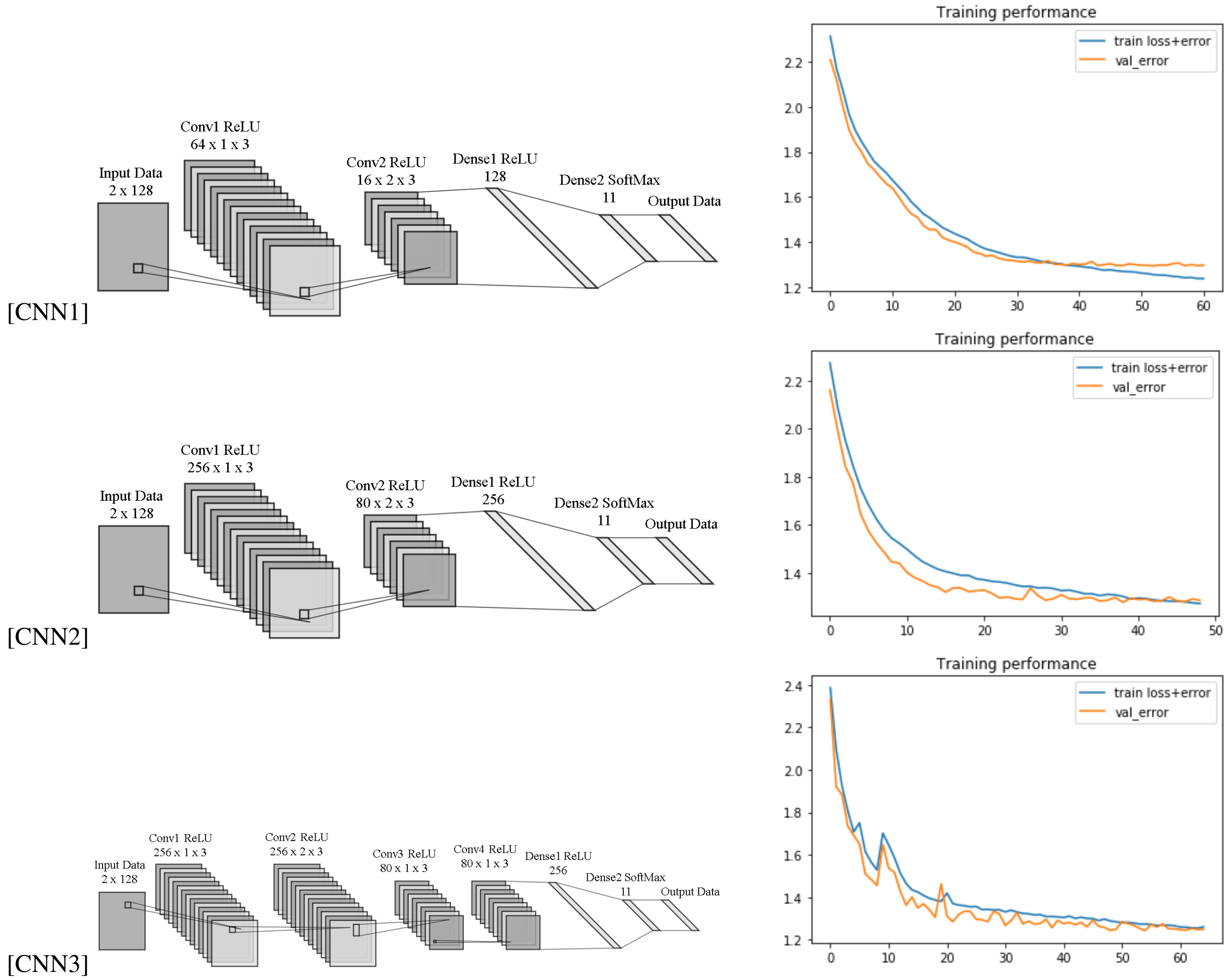


Fig. 4: Structure of convolutional neural networks and training losses.

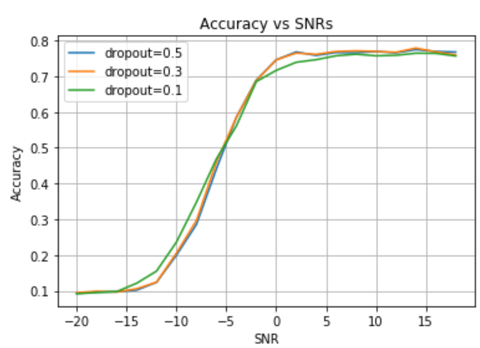


Fig. 5: Effect of dropout on CNN3

B. *Results for CNN*

The results obtained after training the three networks show how model complexity influences the performance as shown in Figure 4. The validation loss at the is lowest for CNN3 at 1.26 it has the highest validation accuracy of 52%. Table. III shows that CNN3 outperforms the other models in terms of accuracy and loss. The effect of change in dropout rate was experimented on CNN3. Interestingly, varying dropout has different effects at different SNR levels as shown in Figure 5. The network trained with low dropout rate of 0.1 has better performance at low SNR levels, whereas the network trained with 0.5 dropout has better performance at high SNR values. In order to select the optimum dropout rate, the validation losses were compared. The final loss the networks with dropout rates 0.1,0.3 and 0.5 are 1.2811, 1.2759 and 1.2485 respectively. Overall, the network

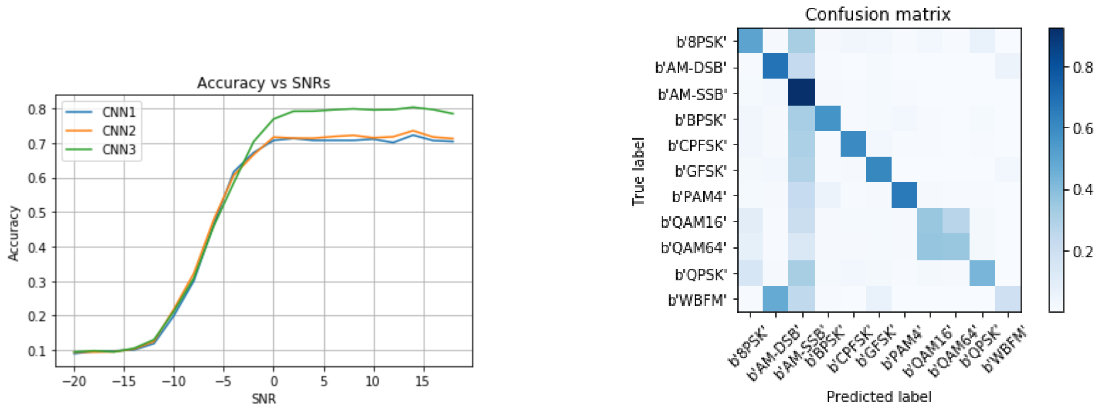


Fig. 6: Classification accuracy vs. SNR and confusion matrix for CNN3

with 0.5 dropout rate has better performance and less validation loss than the other dropout rates and hence is chosen as the final dropout value for the network CNN3. The most complex model CNN3 had the best performance with a test accuracy of 53% while the other models achieved a maximum accuracy of 51%. The difference is more predominant in terms overall accuracy at high SNR values which reached a maximum of 80% whereas the performance of CNN1 and CNN2 were almost identical with maximum accuracy around 71% for high SNR values.

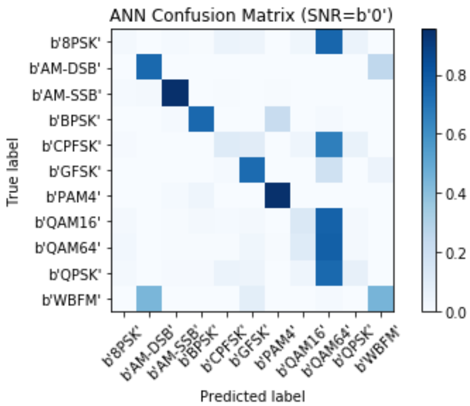
	Metric	CNN1	CNN2	CNN3
Training data	Loss	1.42	1.24	1.25
	Accuracy	0.45	0.51	0.52
Validation data	Loss	1.33	1.29	1.26
	Accuracy	0.48	0.50	0.52
Test data	Loss	1.33	1.27	1.23
	Accuracy	0.49	0.51	0.53

TABLE III: Summary of performance metrics for CNN models

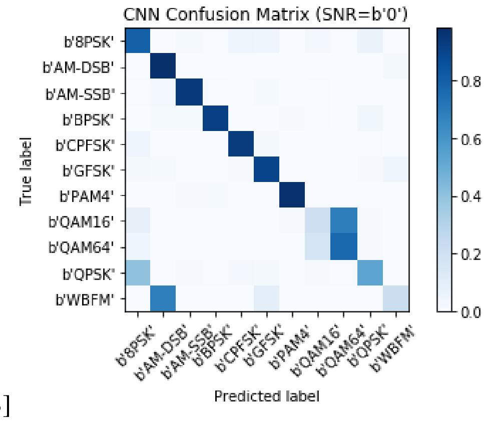
Table IV shows the result of test data when applied to CNN3 network with the selected hyperparameters. It contains accuracy of each modulation class at different SNR levels and gives us an interesting insight of the classification accuracies of different modulations. It can be noted that AM-SSB gets classified with high accuracy even at very low SNR values consistently over 90%. Whereas QAM16, QPSK and WBFM have less than 30% accuracy even at very high SNR values similar to ANN3. When SNR value is 18 dB 8PSK and QAM64 has relatively high accuracy of around 75% while AM-DSM, AM-SSB, BPSK, CPFSK, GFSK and PAM4 have over 95% accuracy. The overall accuracy is brought down to 80% due to misclassifications of QAM16, QPSK and WBFM. It can be noticed that even the more complex CNN networks find it hard to distinguish between QAM16, QPSK signal constellations.

SNR/Label	8PSK	AM-DSB	AM-SSB	BPSK	CPFSK	GFSK	PAM4	QAM16	QAM64	QPSK	WBFM
-20.0	0.01	0.03	0.94	0.01	0.01	0.01	0.00	0.00	0.00	0.00	0.01
-18.0	0.01	0.05	0.96	0.01	0.00	0.01	0.00	0.00	0.00	0.00	0.00
-16.0	0.00	0.07	0.94	0.01	0.00	0.01	0.00	0.00	0.01	0.00	0.01
-14.0	0.01	0.14	0.95	0.00	0.00	0.02	0.01	0.01	0.02	0.00	0.02
-12.0	0.01	0.26	0.92	0.00	0.00	0.02	0.02	0.02	0.11	0.00	0.03
-10.0	0.02	0.51	0.95	0.01	0.01	0.06	0.08	0.11	0.34	0.00	0.06
-8.0	0.06	0.59	0.96	0.07	0.07	0.23	0.25	0.22	0.56	0.02	0.14
-6.0	0.29	0.69	0.96	0.32	0.30	0.56	0.53	0.28	0.68	0.05	0.23
-4.0	0.46	0.83	0.94	0.61	0.61	0.72	0.86	0.29	0.72	0.17	0.24
-2.0	0.62	0.91	0.95	0.82	0.82	0.86	0.94	0.25	0.73	0.42	0.23
0.0	0.80	0.98	0.95	0.92	0.93	0.91	0.97	0.21	0.77	0.53	0.22
2.0	0.84	0.99	0.96	0.97	0.97	0.96	0.97	0.24	0.76	0.51	0.25
4.0	0.83	1.00	0.94	0.97	0.97	0.96	0.98	0.20	0.72	0.57	0.25
6.0	0.82	1.00	0.95	0.97	0.97	0.97	0.97	0.21	0.77	0.52	0.25
8.0	0.81	1.00	0.94	0.96	0.97	0.98	0.98	0.23	0.76	0.58	0.21
10.0	0.84	1.00	0.95	0.96	0.99	0.96	0.98	0.21	0.76	0.57	0.27
12.0	0.84	1.00	0.94	0.96	0.98	0.96	0.97	0.19	0.74	0.63	0.25
14.0	0.81	1.00	0.95	0.97	0.97	0.96	0.99	0.22	0.76	0.58	0.29
16.0	0.82	1.00	0.95	0.94	0.99	0.97	0.98	0.23	0.75	0.56	0.26
18.0	0.83	1.00	0.95	0.97	0.98	0.96	0.98	0.20	0.76	0.56	0.26

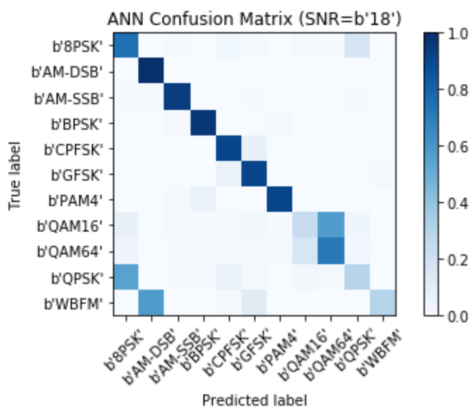
TABLE IV: Classification accuracy of CNN3 at different SNR levels for each class



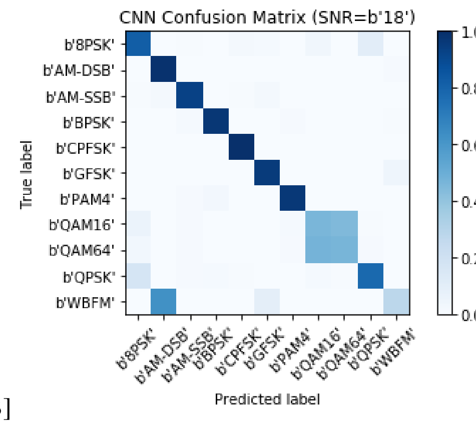
[ANN3 at 0dB]



[CNN3 at 0dB]



[ANN3 at 18dB]



[CNN3 at 0dB]

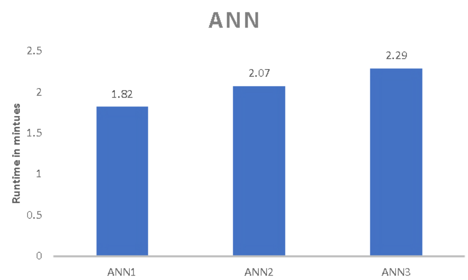
Fig. 7: 0dB and 18dB performance of ANN3 and CNN3

V. Comparison

In general, CNN performs better than ANN in terms of test accuracy and test losses. The maximum test accuracy obtained with an ANN was 44% whereas even the simplest CNN has about 50% accuracy. However, it is interesting to note that ANN3 with 21 features at high SNR values outperformed CNN1 and CNN2 in terms of overall accuracy. This indicates that carefully selected features result in good performance with ANN which has comparatively very less model complexity. We obtain significantly better low-SNR classification accuracy performance from convolutional neural networks than the feature based ANNs. Inspecting the confusion matrices for classifiers at lower SNR levels gives a better interpretation. At very low SNR 0 dB from Figure. 7, it can be noted that the CNN has a cleaner diagonal and is significantly more pronounced than the ANN, in this region learned features have a significant performance advantage. However, there are few misclassifications especially for QAM16, QPSK and WBFM modulation classes and the reason for this is explained in the following section. This is crucial as performance at low SNR impacts range and coverage area over which it can be effectively used as a classifier. This is a significant performance improvement, as this could double effective coverage area of a sensing system[**4**].

Figure 7 shows a confusion matrix of classification for highest SNR case. At +18 dB SNR, we have a clean diagonal in the confusion matrix and can see our remaining discrepancies are that of QAM16 misclassified as QAM64, QPSK misclassified as 8PSK and WBFM misclassified as AM-DSB. These are explainable in the underlying dataset. The QAM64 signal constellation contains the same points that QAM16 has in its constellation, hence the classifier finds it difficult to differentiate between these two. An 8PSK symbol containing the specific bits is indiscernible from QPSK since the QPSK constellation points are spanned by 8PSK points. In the case of WBFM/AM-DSB the analog voice signal has periods of silence where only a carrier tone is present making these examples indiscernible. Therefore, it is unlikely that and accuracy of 100% can be obtained even at high SNR on this data set and making the remaining confusion reasonably tolerated.

An important consideration in real time wireless communication system is the training and classification run time due to computational complexity. Figure 8 shows that convolutional neural networks needs longer times for training and classification than ANNs. CNN3 almost takes 538 minutes to train in a standard i5 processor. High accuracy is obtained only with complex networks as shown previously. In sum, the trade-off between the computation time and accuracy is a crucial while deciding the model for the classifier.



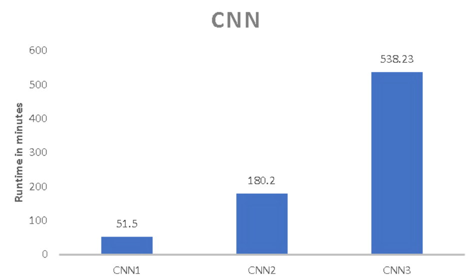


Fig. 8: Training runtime for ANN and CNN

VI. Conclusion

In this work, fast deep learning algorithms for distinguishing between 11 different modulation types, with high classification accuracy over a wide range of SNR values was presented. Artificial neural network and convolutional neural network architectures were examined in several settings. The results demonstrate that compared to a well expert feature bases ANN, Convolutional Networks are work quite well in low levels. Hence CNNs are a preferred choice in applications which rely on robust low SNR classification. One major challenge on using CNN is the long training time. For example, for the problem at hand, even the simple CNN1 architecture would take approximately 51 minutes to train. This creates a serious obstacle towards the feasibility of applying such algorithms in real-time, where online training is needed to adapt the network architecture to changing environmental conditions. Thus it can be concluded that both ANN and CNN can be implemented and trained for providing good performance, the choice however depends on the requirements of the classifier and other constraints.

References

[1] B. Kim, J. Kim, H. Chae, D. Yoon and J. W. Choi, ”Deep neural network-based automatic modulation classification technique,” 2016 International Conference on Information and Communication Technology Convergence (ICTC), Jeju, 2016, pp. 579-582.
[2] Lee, Sang & Kim, Kwang-Yul & Shin, Yoan. (2020). Effective Feature Selection Method for Deep Learning-Based Automatic Modulation Classification Scheme Using Higher-Order Statistics. Applied Sciences. 10. 588. 10.3390/app10020588.
[3] Ramjee, Sharan & Ju, Shengtai & Yang, Diyu & Liu, Xiaoyu & El Gamal, Aly & Eldar, Yonina. (2019). Fast Deep Learning for Automatic Modulation Classification.
[4] O’Shea, Tim & Corgan, Johnathan & Clancy, T.. (2016). Convolutional Radio Modulation Recognition Networks. 10.1007/978-3-319-44188-7_16.
[5] Z. Zhu, A. K. Nandi (2015). “Automatic Modultaion Classification: Principles, Algorithms and Applications,” London : Wiley
[6] Hong, D.; Zhang, Z.; Xu, X. Automatic Modulation Classification Using Recurrent Neural Networks.In Proceedings of the 3rd IEEE International Conference on Computer and Communications (ICCC), Chengdu, China, 13–16 December 2017; pp. 695–700.
[7] X. Liu, D. Yang, and A. El Gamal, “Deep neural network architectures for modulation classification,” in Proc. IEEE Asilomar Conference on Signals, Systems and Computers, 2017.
[8] E. E. Azzouz and A. K. Nandi, “Modulation recognition using artificial neural networks,” Signal Processing, vol. 56, no. 2, pp. 165–175, 1997.
[9] T. O’Shea, J. Corgan, and T. Clancy, “Convolutional radio modulation recognition networks,” in Proc. International conference on engineering applications of neural networks, 2016.
[10] F. Chollet, Keras, https://github.com/fchollet/keras, 2015
[11] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” CoRR, vol. abs/1412.6980, 2014 [Online]. Available: http://arxiv.org/abs/1412.6980.
[12] M. Abadi, A. Agarwal, et al., “Tensorflow: large-scale machine learning on heterogeneous systems, 2015” Software available from tensorflow.org,