Evaluating Deep Learning Networks for Modulation Recognition

Tina L. Burns*, Richard P. Martin*, Jorge Ortiz*, Ivan Seskar*, Dragoslav Stojadinovic,*, Ryan Davis*, Miguel Camelo†
* WINLAB, Rutgers University, North Brunswick, NJ, USA
† University of Antwerp - imec, IDLab - Department of Computer Science Sint-Pietersvliet 7, 2000 Antwerp, Belgium
tld95@scarletmail.rutgers.edu, jorge.ortiz@rutgers.edu, seskar, dragos, ryan, rmartin@winlab.rutgers.edu,
miguel.camelo@uantwerpen.be

ABSTRACT - As the use of wireless communication expands demand for radio spectrum, so does the need for effective automatic modulation recognition (AMR). Current methods of AMR include feature extractions, maximum likelihood algorithms, and deep learning (DL) networks primarily based on CNN structures. Many methods are limited by the slow training and testing time, the need for massive amounts of training data, and low probability of correct classification in the presence of noise. Our research proposes using a fully connected dense network instead of a convolutional one to mitigate some of these challenges. To test modulation classification accuracy, we used in-phase and quadrature samples from data sets at various signal-to-noise levels to evaluate 5 DL networks and a matched filter approach. Our experiments show that compared to traditional convolutional networks, our fully connected network improves training and testing times by order of magnitude, has an accuracy within 5% of the most accurate convolutional network, and uses a factor of 32 fewer parameters. We also demonstrate that using a bank of matched filters remains challenging, as correctly discriminating amongst several positive matches is not straightforward.

Index Terms: *Deep Learning, Modulation Recognition, Neural Networks, Data Generation, Matched Filter, CNN, LSTM, Autoencoder, Fully Connected Network*

I. INTRODUCTION

The decreasing size of electronics on a chip has resulted in rapid reductions in the size, cost, and power needed to realize sophisticated RF devices as well as the proliferation of communication system standards. The resulting increase in the number of wireless transmitters demands new spectrum management methods in a diverse set of areas including bandwidth allocation, privacy, and security. One way to optimize usage of the radio spectrum is by employing spectrum sensing techniques to identify underutilized bandwidth which can be dynamically allocated. [21] Fortunately, technology developments have provided unique opportunities for spectrum sensing. Advancements in communications have enabled new classes of RF devices to emerge, driven both by new standards and software-only RF designs. Software radios also hold the promise of customized, application-specific protocols. Since transmitter detection is an essential component of spectrum sensing [21], transmitter identification is a vital part of spectrum management in this emerging ecosystem of RF devices.

Signal or transmitter identification has gained popularity because of its widespread applications in military defense, spectrum recognition, cognitive radio, and other areas. [24] In hostile environments, it can help detect unwanted communication activity. In the commercial arena, signal identification aids bandwidth management and allocation [2]. Automatic modulation recognition (AMR) is an important part of this process since it can serve as the intermediary between demodulation and signal identification [1] [24, 20]. Too add to this, AMR is advantageous in blind signal identification which involves being able to effectively classify a signal with a minimal amount of information.

Matched filters are the traditional method to identify transmitters and signals [6]. They are frequently used in various communication applications including radar detection, sonar, blood vessel detection, and fingerprint enhancements.[4, 9, 22] One approach to modulation recognition is to use a bank of matching filters, one for each modulation. However, we show in Section V that selecting the true positive amongst many positive matches remains challenging.

In this work, we present an approach to modulation recognition using neural networks. Our approach is analogous to how speech recognition transitioned from template matching to neural networks [17]. Rather than attempting to scale templates, our approach uses supervised machine learning. We collect scaled raw IQ samples from a number of transmitters using known modulations as input to a variety of neural networks.

Our work has two main contributions. First, we evaluate a variety of NNs and compare them for identification accuracy as the signal is subjected to increasing noise. We also compare NNs on the level of resource use, in terms of the number of parameters and training time, as a function of accuracy. Second, we examine the impact of training data types on accuracy. This includes two methods of synthetic data generation and real radios in both a noiseless and uncontrolled environments. We observed that the convolutional neural network (CNN) had the highest accuracy, but also had a very high resource use. We discovered that a simpler fully connected network uses less resources in terms of the number of parameters and training time, and was within 5% of the accuracy of the CNN. We also found that the training data type used has a wide impact on accuracy.

The remaining sections of this paper are organized as follows: Section II reviews the previous work of AMR and classification schemes. A description of the data collection and simulation techniques is provided in Section III. In Section IV we discuss the design and implementation of our neural networks and matched filter classification network. Next, evaluation results are included in Section V. We provide our conclusion in Section VI. Then future work is discussed in Section VII. The paper ends with the acknowledgements in Section VIII.

II. Background and Related Work

Currently, DL in AMR is an extremely popular area of research. Most of this research focuses on using Convolutional Neural Networks (CNNs), since they have proven to be very reliable in image recognition and feature extraction. In recent work, Shi et. al. used CNNs to evaluate AMR over various SNR ratios. They compared the results with Deep Neural Networks, Random Tree Algorithms, and Random Forests. [20] They discovered that the CNN with IQ dataset without the phase offset provided the best accuracy. [20] Hau Gu et. al. used deep learning algorithms composed of two distinct CNN networks in their research of AMR. [7] Additionally, Fuxin Zhang et. al. created AMR DL learning networks with a CNN and grate recurrent unit (GRU) that achieved 90% accuracy. [3]

Another form of DL used for AMR are autoencoders (AE). These networks have recently gained interest because of their ability to benefit from non-supervised learning [13]. Work by Bouchou et al. showed that a stacked sparse autoencoder provided 100% accurate classification of modulation schemes at an SNR of 5dB. Their results also indicate that the AE had better performance than a Support Vector Machine approach. A concurrent study by Ali et. al used a Sparse Autoencoder (SAE) and AE with non-negative constraints (ANC) in their approach to AMR. Their work showed promising classification accuracy results with PCCs above 95% when SNR levels were 0 dB. [23] Unfortunately, these AE studies did not contain a comparison to CNNs and other network structures.

Some researchers have also used matched filters for modulation detection. Aafreen Shaikh et. al attained a 99% probability of detection accuracy with their matched filters when testing BPSK and AM signals [18]. Matched filters are advantageous because they work well under conditions with AWGN noise. Unfortunately, they require a prior knowledge in order to effectively classify signals. Additionally, as we demonstrate, they are not an optimal solution to distinguish between multiple modulation schemes.

Overall, the studies discussed here provide useful insight into AMR techniques. Unfortunately, the works exclude examinations of a fully connected neural network (FCNN), which is not traditionally studied in AMR. Most of the works mentioned focus on using CNNs and do not compare the CNN performance to all of the networks evaluated here. Our research investigates FCNNs, our novel binary (BIN) network, an LSTM, an Autoencoder Network, a CNN network and matched filter classifier (MFC). Overall, this work provides a more robust view of the CNN performance against other classification approaches. Moreover, previous works mentioned here do not analyze the networks' resources and training times. Such a comparison is vital in real-world applications which must run in energy and space-constrained environments. We also investigate how the networks respond given different dataset sizes, as often large datasets may not be readily available, so developers need to consider the best DL under a data constraint.

III. Training Set Generation

We generated three training sets: (1) The MATLAB set, (2) the GNURadio[5] set, and (3) the Grid set. We begin with two transmit generators: the first based in MATLAB, and the second based on GNURadio. Both generate IQ samples, but in a much different fashion from each other, which we describe below. We added increasing amounts of AWGN to the transmit IQ values. The first two training sets, from MATLAB and GNURadio, use these IQ samples with added AWGN directly. The third training set, Grid, uses the GNURadio IQ values as input to a software radio, which transmits a real signal over the air in the Orbit testbed[16]. To minimize, but not eliminate, the impacts of multipath and noise, the Grid transmitter and receiver were 1 meter apart. We then use the received IQ values from a software radio receiver to form the Grid training set. The GRID data thus is the most representative training set of real data, as it was IQ samples sent over real radios in a challenging multipath environment. The MATLAB and GNURadio sets are useful for understanding how performance degrades as AWGN is added in a controlled fashion.

MATLAB set. The MATLAB dataset generated random numbers between 0 and N-1, where N is the number of constellation points. These are mapped onto the constellation map at their appropriate constellation point locations. AWGN is added to the modulation constellation to achieve the desired SNR. The idea is to select constellation points at random, and then add random noise to each constellation point.

GNURadio set. The GNURadio signal generation can be represented by the following equation 1:

$$y[n] = s[n] + w[n] \tag{1}$$

where w(n) represents the AWGN noise, s[n] is the noiseless signal, and y(n) is the signal combined with the noise. The script outputs binary files containing 32 bits IQ pairs with a 16 bit I value and a 16 bit Q value for BPSK, QPSK, 8PSK, and 16QAM modulation schemes. The SNR was adjusted in 1 dB increments ranging from -30 dB to 30 dB. This technique is qualitatively different from the MATLAB approach because the IQ samples in the GNURadio case include intermediate points between the modulation's constellation points.

The key difference between the MATLAB and GNURadio data sets are the IQ values temporal changes. While both choose random points in the plane, the GNURadio set is more realistic because the IQ pairs are constrained to follow an analog baseband, while the MATLAB IQ points make discrete jumps. This shows up in the intermediate IQ sample points in the plane. The GNURadio IQ samples are more constrained, and thus more likely to remain in between constellation points, where the MATLAB ones are not. Section V shows how the DL networks can use this property of realistic IQ point generation to improve classification accuracy.

Grid set. The generated IQ samples data was collected using National Instrument USRP software defined radios. The GNURadio datasets were fed into the USRP transmitters and the data was collected at the receiver in 32 bit IQ pairs and sent over the air in the Orbit Grid. The Grid data is transmitted over the air from a USRP x310 to a USRP B210 receiver. The Grid data was also collected using receivers located at various distances from the transmitter ranging from 3ft to 70ft. We found that accuracy was not very sensitive to distance; we describe the impact of distance on accuracy further in Section V.

IV. Classification Approaches

In this section, we describe the architectures of our six classification approaches: the convolutional neural network (CNN), the fully connected neural network (FCNN), the binary network (BIN), the long short-term memory (LSTM) neural network, the autoencoder (AE), and the matched filter classifier (MFC). We optimized each neural network by experimentally testing different combinations of layers, filters/nodes, and activation functions. The final network configurations had the best accuracy results with the lowest computation resources. *Space limitation prevents full descriptions, see the references for more details.*

Convolution Neural Network (CNN): Convolutional Neural Networks (CNNs) are one of the most well-known deep learning networks. CNNs have been used in several applications including computer and speech recognition, image classification, SAR image segmentation, image classification, and AMR.[19, 11]. The CNN network of this project contains six convolutional layers, three upsampling layers, three max pooling layers, a dropout layer, a flatten layer, three fully connected dense layers, and one fully connected dense output layer. For more details on this network, refer to the diagram in figure 1. This network uses categorical classification to determine which of the four categories the modulation scheme falls into.

Fully Connected Neural Network (FCNN): Fully connected multi-layer perceptions (MLPS) or fully connected layers are very common in neural networks.[8] In fully connected dense layers, each node is linked to all the nodes in the previous and next layer. FCNNs were traditionally thought to have longer training time than other networks. Our research

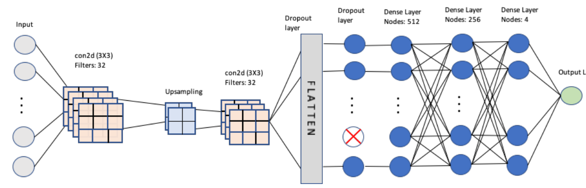


Fig. 1. Layers of the CNN Network

will show that FCNNs can still be effective in AMR. The FCNN here contains: four fully connected hidden layers using a softsign activation function, one dropout layer, and the final fully connected layer with a relu activation function.

Binary Network (BIN): The BIN network is a binary classifier that is one of the non-conventional networks of this project. The hidden layers of this structure are identical to that of the FCNN. However, the data separated into two categories using cross entropy, represented as 1 for the modulation of interest and 0 otherwise. The network is trained and tested independently for each modulation scheme.

Long Short Term Memory (LSTM) Network LSTM's are a time recursive class of machine learning.[12]. The LSTM Network of this project uses a sequential keras model containing two LSTM layers, a dropout layer, and one fully connected dense layer. The network applies a categorical cross entropy loss function and an rmsprop optimizer. As we will discuss in section V-A the LSTM network has very long training and testing times. The shallow design of the network in intended to minimize computational resources and running times.

Autoencoder (AE) The two main components of an AE are the encoder and decoder. Encoders compress the input into fewer bits. The encoder portion of our network consists of a flatten layer and four fully connected dense layers with nodes in descending order in the following pattern 256, 128, 64, and 32. The decoder section has four fully connected dense layers with the following number of nodes 32, 64, 128, and 256. The decoder is followed by a dropout layer and the final fully connected layer. Similar to the FCNN and CNN, the data is set into 4 categories of binary vectors using one-hot encoding.

Matched Filter Classifier In order to compare our DL classification to existing approaches, we also used matched filters. Matched filters convolve a known signal (ideally the shifted and reversed version of the desired signal) with an unknown signal. [15] The filters declare a signal detected if the match function exceeds a certain threshold. They are designed to maximize the signal to noise ratio of a specific signal.[14] Our MFC algorithm uses several matched filters in combination with a classification algorithm. After normalizing the signals' power, the matched filter cross correlates a set of training data from all four modulation schemes against the test signal. The maximum value of each cross-correlation is fed into the classification algorithm.

The classification algorithm operates by using a two-tier approach with Bayesian inference. In the first tier, the most probable modulation scheme is selected based on the correlation output with the highest value. This is supported by the correlation classification equation shown 2 which is a common form of filter matching [10].

$$\int_{-\infty}^{\infty} x(t)s_k^*(t)dt \tag{2}$$

In the second tier of the algorithm, each modulation output is evaluated separately. If filter output is above a certain threshold, it is declared as a part of the modulation class. The threshold for each modulation is determined experimentally and is designed to provide an optimal balance between true positive and false negative values. The prediction array is then updated to reflect the positive classification of thresholding.

As a note, our implementation is qualitatively different than other matched filters since it does not include timing recovery. This is done for simplicity in order to provide a closer comparison to the process of DL networks.

V. Experiment Results and Discussion

Our results can be divided into three categories: (1) accuracy analysis, (2) temporal clustering, and (3) network resources. The Temporal Clustering section discusses a phenomenon that we discovered while observing different data generation methods. Lastly, we review the DL training time and parameter numbers.

A. Accuracy Analysis

Matched Filter Classification The classification results of the Matched Filter Classifier (MFC) are shown in figure 2. The SNR vs accuracy graph shows that the MFC works well for the BPSK and 16QAM modulation schemes and attain values over a 95% true positive rate. Both the QPSK and 8PSK modulation have sub-optimal results. This is a result of misclassification based on high output values from the filter. Ideally, when a signal is matched perfectly against itself, it should produce the highest correlation result. However, if this same filter is used for a modulation scheme with more densely populated values, it may cause high correlation matches in the output.

While each individual matched filter may have a high-true positive rate, resolving false positives from a bank of matched filters remains a challenge, as this is not what they were designed for. As more modulation schemes are added to the test, it becomes more difficult to distinguish the best match from the false positives. This phenomenon is observed in "Filter Output" results in figure 2 which shows that when a QPSK signal is matched against an 8PSK filter, it yields a higher output than when matched with a QPSK filter. Additionally, the Confusion Matrix of the MFC at an SNR of 10dB demonstrates that the QPSK and 8PSK signals often produce false positives for other modulation schemes.

As a note, due to time constraints and computational limitations, the MFC was only tested at 24,440 samples. This sample size is used for all MFC results in figures 2 and 3.

MATLAB Dataset The Probability of Correct Classification, or accuracy, of the neural networks of the MATLAB generated data for 244,000 samples is shown in figure 3. The plot shows that only the CNN network was able to achieve an accuracy level of 100%. The LSTM had the second highest accuracy of 75%. The FCNN has higher levels of accuracy than the AE and BIN networks at some SNR levels but only reached a peak PCC of 70%. The MFC is not able to classify correctly and remains at an accuracy level close to 25%.

When the training samples were increased to 2.44 million samples, the LSTM network vastly improves and follows the CNN network, which converges to 100% accuracy at an SNR of 7dB. The BIN, AE, and FC networks improve slightly do not exceed an accuracy of 75%.

We hypothesize that part of the reason for the networks poor performance was temporal clustering of the constellation points. See section V-B for more details.

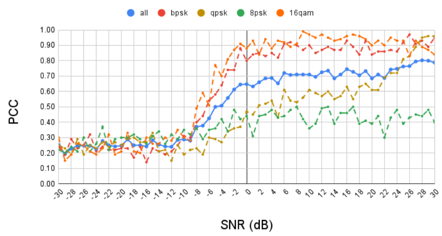
GNURadio Dataset Figure 3 demonstrates the accuracy results of the network with 244,000 samples. The CNN network has the best accuracy performance. The FCNN, AE, BIN networks have very similar accuracy and fall slightly short of the CNN accuracy. The LSTM network lags behind the others. The MFC has the worst performance and achieves a maximum accuracy of 80%. The indicates that CNN network performs better than the other network in conditions with AWGN noise.

Figure 3 demonstrates that, as expected, with the larger training size (2.44 million samples) all networks improved their accuracy. However, the LSTM network still has the worst performance and lags behind the other networks. The CNN performs slightly better than the FC, AE, and BIN networks and has a small gap above their accuracy. The results indicate that the FCNN, AE, and BIN networks provide comparable performance to that of the CNN network for this particular dataset.

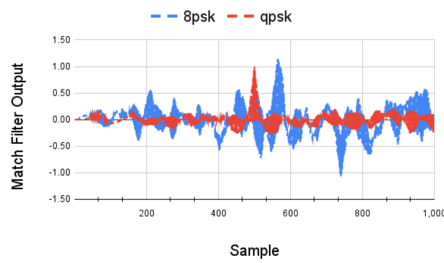
GRID Dataset Figure 3 shows the performance of the neural networks when the GNURadio data is transmitted wirelessly on the software defined radios of the Orbit Grid. Once again, the CNN proves the best response and is the only network that achieves close to 100% accuracy. The FC, AE, and BIN all have similar responses and achieve a maximum accuracy close to 90%. The FC network provides a slight advantage over the other networks in that it has higher accuracy between SNR levels of 0 and 10 dB. The LSTM network performs poorly here and only attains an accuracy of 58%. The MFC has a slightly higher accuracy than the LSTM but only reaches a maximum of 70%.

When we compare this to the results of the synthetic GNURadio, we see that transmitting the same IQ values with the software radios over the air impacts the accuracy of the networks in several ways. All the networks reach their maximum accuracy at higher SNR levels (around 8 dB or higher) for the wirelessly transmitted data. The LSTM seems to be very sensitive to the impact of radio transmission as its accuracy dropped by almost 40%. The FC, AE, and BIN networks are also affected and have lower accuracy curves overall.

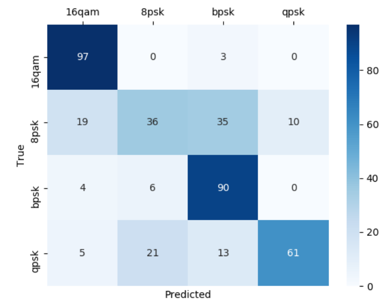
We also used the GRID to evaluate the impact of attenuation and multipath fading on the accuracy results by collecting the data from receivers at various distances. Since farther



SNR vs PCC

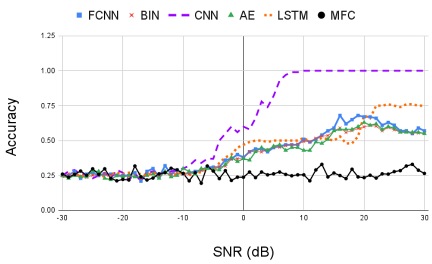


Filter Output of QPSK signal

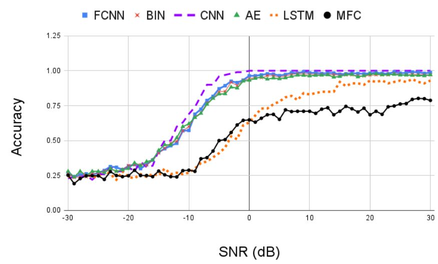


Confusion Matrix at 10 dB SNR

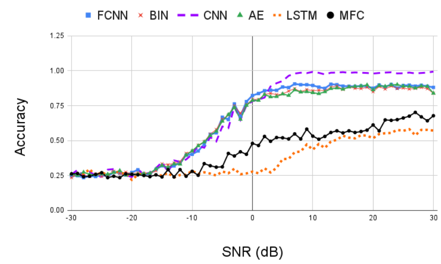
Fig. 2. MATCHED FILTER Results



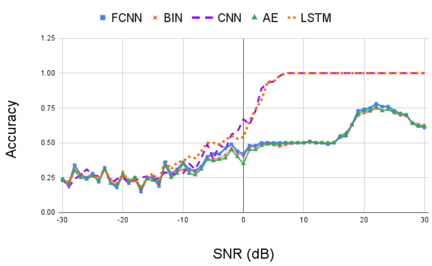
MATLAB(244k samples)



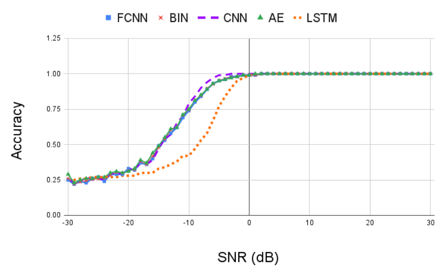
GNURADIO (244k samples)



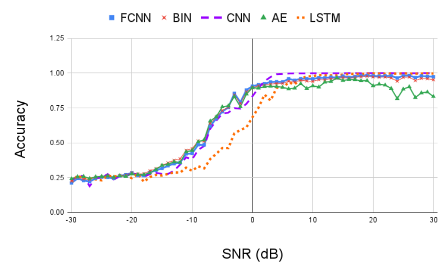
GRID (244k samples)



MATLAB(2.4 Million samples)

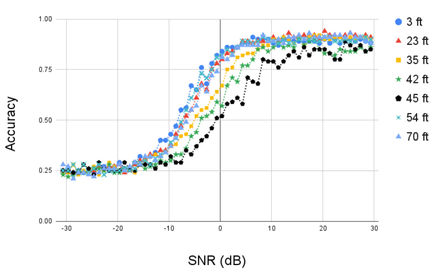


GNURADIO (2.4 Million samples)

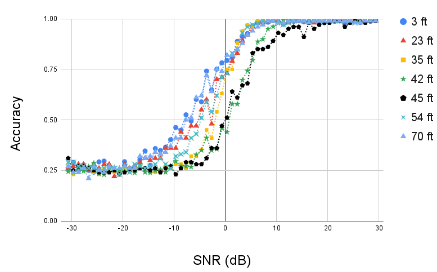


GRID (2.4 Million samples)

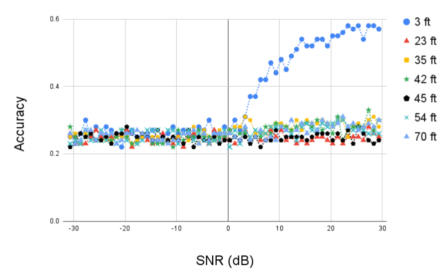
Fig. 3. SNR vs Accuracy for Various Datasets



(FCNN)



(CNN)



(LSTM)

Fig. 4. SNR vs Accuracy for receivers at various distances

distances result in lower power levels at the receiver, increased distance represents greater levels of attenuation. In addition, the Orbit Grid is a high multipath environment, and the distortions caused by multipath components become stronger with distance.

Figures 4 shows the response of the FCNN, CNN, and LSTM neural networks with receivers at various distances from the transmitter. We observed that distance is not the dominating factor in the network's accuracy. In fact, 3 ft and 70 ft are very close for the CNN and FCNN networks. However, the receivers at distances of 42 ft and 45 ft had the worst performance. This is most likely because the specific location of these receivers has more multipath components than at other locations. The CNN network is the only network that reaches 100% accuracy. The FCNN shows comparable results with a maximum accuracy of 94%. Additionally, the FCNN network also has a higher accuracy level at lower SNR levels. This indicates that the FCNN is slightly more robust than the CNN network under noisy conditions when multipath fading is present. The LSTM network only reaches 58% accuracy at 3ft, but collapses at all other distances.

B. Temporal Clustering

While evaluating the simulation results of the emulated data we noticed that there is a slight bias in the temporal allocation of constellation symbols. For example, when plotting 100 IQ constellation points for 8PSK-modulated random data in GNURadio, all the points remained in quadrants I and IV. As the number of samples increases, the constellation points are also distributed to all 4 quadrants. The data generated in MATLAB does not show the same clustering but distributes IQ constellation points to all 4 quadrants even with small sample sizes. The constellation plots of 100 IQ pairs for the GNURadio and MATLAB data are shown in figures 5 and 6. This observation was further quantified by calculating the average Euclidean distance between adjacent IQ points, 100 sampling periods (samples), where each sampling is 500 IQ pairs for each modulation scheme. The results are exhibited in table I. The data shows that on average, the distance between adjacent IQ pairs of the MATLAB data is at three times the distance of GNURadio adjacent temporal points. The shorter distance between the adjacent points of the GNURadio data creates a temporal pattern that is less randomized than that of the MATLAB data. We hypothesize that the DL uses this pattern as a feature, making it easier to classify the GNURadio dataset, which results in higher classification accuracy.

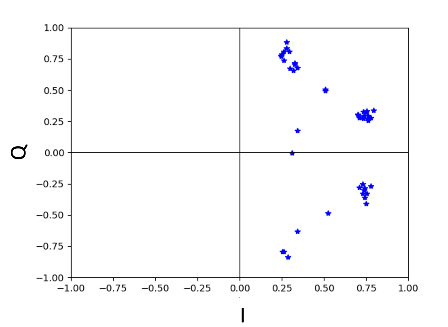


Fig. 5. 8PSK Constellation Plot of GNURadio Data for 100 IQ Pairs

C. Training Times and Resource Load

Timing We evaluated the total training time of each network with 244,000 data samples per file. The following parameters were included in the time assessment: base training time, testing time, and overhead time. Operations such as data collection and labeling are included in the overhead timing. LSTM is the slowest network the FCNN network is the fastest.

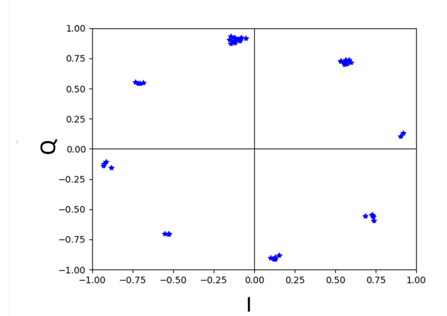


Fig. 6. 8PSK Constellation Plot of MATLAB Data for 100 IQ Pairs

Constellation	GNURADIO	MATLAB
BPSK	0.40	1.31
QPSK	0.21	1.18
8PSK	0.17	1.21
16QAM	0.20	0.95

TABLE I
AVERAGE POINT CONSTELLATION DISTANCE FOR DATA IN GNURADIO AND MATLAB

The CNN is also time-intensive. The results are displayed in figure 7.

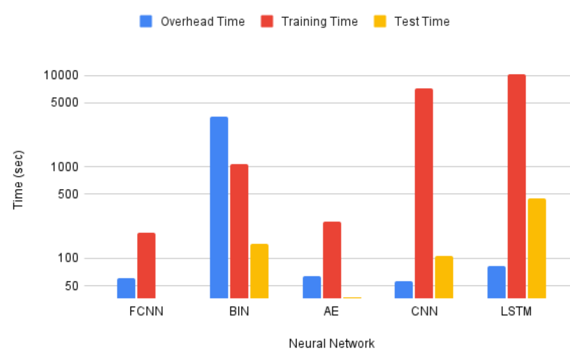


Fig. 7. Network Timing

Resource Analysis One of the important factors in creating the networks is the amount of computational resources that the network requires, as many radios must operate in space and energy constrained environments. The resource allocations, in terms of the number of parameters needed to realize a network, are displayed in table II. The CNN network is the most resource-intensive network and contains 16,529,364 parameters and 17 layers. The FCNN delivers high accuracy with 505,694 parameters, which is a factor of 32 fewer parameters than the CNN.

VI. CONCLUSION

This research bridges the gap between the most common neural networks studied for AMR (LSTM and CNNs) and those that are less observed (FCNN and AE). Another important aspect of this research was comparing the networks performance with simulated data and data transmitted over the

Network	Layers	Parameters
FCN	5	505694
CNN	17	16529364
AE	8	690056
BIN	5	202276
LSTM	4	12804

TABLE II
RESOURCE ALLOCATION FOR EACH NETWORK

air from SDRs. This comparison is essential because it shows the response of the networks with the real world effects multipath fading and signal attenuation are present. Of the networks that were evaluated the CNN provides the best accuracy performance. The FCNN provides comparable accuracy performance for some datasets and is the least resource-intensive. The BIN and AE networks provide comparable accuracy results but have longer training and or overhead times. The LSTM has the worst performance with the longest training time and poor accuracy at lower SNR levels. Although expanding the number of layers in the network would likely enhance it's accuracy, it would also increase the training and testing time. These results indicate that the FCNN network presented here achieves good recognition accuracy with significantly reduced complexity, and should be a serious candidate for AMR recognition implementation.

VII. FUTURE WORK

Due to time and resource constraints, some of the experiments and evaluations of this research were limited. As such, some examinations should be completed in future work. For example, the performance of the networks could be evaluated with signal representations other than IQ samples. The networks should also be tested with more robust datasets containing additional modulations. Moreover, in order to meet page limit requirements for this submission, we excluded components from this submission including some details about related work and results from related works. Future work could include this information.

VIII. ACKNOWLEDGMENTS

We would like to acknowledge all those at WINLAB and Rutgers University who assisted with this project. This work was funded by NSF awards OAC-1925482 and CNS-1836901. Future work involving this project will continue through the support from NSF award RU-NET project 826336.

REFERENCES

[1] OA Dobre, A. Abhi, Y. Bar-ness, and et. al. Survey of automatic modulation classification techniques: classical approaches and new trends. *IET Commun*, pages 137–156, 2007. https://doi.org/10.1049/iet-com:20050176.

[2] Octavia Dobre. Signal identification for emerging intelligent radios: Classical problems and new challenges. *IEEE Instrumentation & Measurement Magazine*, 18:11–18, April 2015. doi:10.1109/MIM.2015.7066677.

[3] Zhang Fuxin, Chunbo Luo, Jialang Xu, and Yang Lou. An efficient deep learning model for automatic modulation recognition based on parameter estimation and transformation. *IEEE Communications Letters*, 10:3287–3290, 2021. https://doi.org/10.1109/LCOMM.2021.3102656.

[4] Ryan M. Gerdes, Thomas E. Daniels, Mani Mina, and Steve Russell. Device identification via analog signal fingerprinting: A matched filter approach. In *NDSS*. The Internet Society, 2006.

[5] GNU Radio Website, accessed September 2021. http://www.gnuradio.org.

[6] Robert Hecht-Nielsen. Neurocomputer nearest matched filter classification of spatiotemporal patterns. *SPIE*, pages 105–15, 1987. doi:10.1117/12.939915.

[7] Hao Gu. Yu Wang. Sheng Hong. and Guan Gui. Blind channel identification aided generalized automatic modulation recognition based on deep learning. *IEEE Access*, page 110722–29, 2019. https://doi.org/10.1109/ACCESS.2019.2934354.

[8] J Janke, M Castelli, and A Popovič. Analysis of the proficiency of fully connected neural networks in the process of classifying digital images. benchmark of different classification algorithms on high-level image features from convolutional layers. *Expert Systems with Applications*, 135:12–38, 2019. doi.org/10.1016/j.eswa.2019.05.058.

[9] Melike Atay Karabalkan and Mehmet Oner. Implementation of match filters as ip core for sonar applications. In *SIU*, pages 887–890. IEEE, 2015.

[10] Aaron Lanterman. Chapter 9: Correlation and matched filtering. https://cpb-us-w2.wpmucdn.com/sites.gatech.edu/dist/e/466/files/2016/11/sp14_notes3084_10_correlation.pdf, 2014.

[11] VJ Lawhern, AJ Solon, NR Waytowich, SM Gordon, CP Hung, and BJ. Lance. Eegnet: a compact convolutional neural network for eeg-based brain–computer interfaces. *Journal of Neural Engineering*, 15:05613, July 2018. doi:10.1088/1741-2552/aace8c.

[12] Shuai Li, Qi Wang, Xiaojie Liu, and Jienan Chen. Low cost lstm implementation based on stochastic computing for channel state information prediction. In *APCCAS*, pages 231–234. IEEE, 2018. https://doi.org/10.1109/APCCAS.2018.8605569.

[13] Mohammed El Hadi Lakhdari. Mohamed Bouchou, Hua Wang. Automatic digital modulation recognition based on stacked sparse autoencoder. *2017 IEEE 17th International Conference on Communication Technology (ICCT)*, pages 28–32, 2017. doi:10.1109/ICCT.2017.8359478.

[14] University of Washington. Lab 3 usrp. http://courses.washington.edu/ee420/projects/lab3_gnuradio.pdf.

[15] radartutorial.eu. Radar basics: Matched filter. https://www.radartutorial.eu/10.processing/Matched%20Filter.en.html.

[16] Dipankar Raychaudhuri, Ivan Seskar, Max Ott, Sachin Ganu, Kishore Ramachandran, Haris Kremo, Robert Siracusa, Hang Liu, and Manpreet Singh. Overview of the

orbit radio grid testbed for evaluation of next-generation wireless network protocols. In *Wireless Communications and Networking Conference, 2005 IEEE*, volume 3, pages 1664–1669. IEEE, 2005.

[17] Suman K Saksamudre, PP Shrishrimal, and RR Deshmukh. A review on different approaches for speech recognition system. *International Journal of Computer Applications*, 115(22), 2015.

[18] Diba Aafreen Shaikh and Tejashree L. Nawale. Improving rf spectrum utilization using matched filter based spectrum sensing for crn. *International Journal of Engineering Research 'I&' Technology (IJERT)*, 5:2278–0181, April 2016.

[19] F Sharifzadeh, Akbarizadeh G, and Y. Seifi Kavian. Ship classification in sar images using a new hybrid cnn–mlp classifier. *Journal of the Indian Society of Remote Sensing.*, 47:551–562, 2019. doi:10.1007/s12524-018-0891-y.

[20] J. Shi, S. Hong, C. Cai, Y. Wang, H. Huang, and G. Gui. Deep learning-based automatic modulation recognition method in the presence of phase offset. *IEEE Access*, 8:42841–42847, 2020.

[21] Mansi Subhedar1 and Gajanan Birajdar. Spectrum sensing techniques in cognitive radio networks: A survey. *International Journal of Next-Generation Networks (IJNGN)*, 3:37–51, 2011. http://www.airccse.org/journal/ijngn/papers/3211ijngn03.pdf.

[22] R. M. Wallingford, E. M. Siwek, and J. N. Gray. Application of two-dimensional matched filters to X-ray radiographic flaw detection and enhancement. In *Review of Progress in Quantitative Nondestructive Evaluation. Vol. 11B*, volume 11, pages 879–886, January 1992. https://ui.adsabs.harvard.edu/abs/1992qnde...11..879W.

[23] Ali A. Fan Yangyu. Automatic modulation classification using deep learning based on sparse autoencoders with nonnegativity constraints. *IEEE signal processing letters*, pages 1626–1630, 2017. doi:10.1109/ICCT.2017.8359478.

[24] Yuan Zeng, Meng Zhang, Fei Han, Yi Gong, and Jin Zhang. Spectrum analysis and convolutional neural network for automatic modulation recognition. *IEEE Wirel. Commun. Lett.*, 8(3):929–932, 2019. http://dblp.uni-trier.de/db/journals/wcl/wcl8.htmlZengZHGZ19.