Automatic Modulation Classification Using Reinforcement Learning

Timothy Calvert*, Celyn Jacobs†, Emily Huynh‡ and Cameron Flores§
*Department of Electrical and Computer Engineering, California State Polytechnic University, Pomona, California 91768 Email: tjcalvert@cpp.edu
†Department of Electrical and Computer Engineering, University of Arizona, Tucson, Arizona 85721 Email: celynjacobs@email.arizona.edu
‡Department of Electrical and Computer Engineering, University of Nevada, Las Vegas, Las Vegas, Nevada 89154 Email: emilyhuynh2018@gmail.com
§Department of Computer Science, Vassar College, Poughkeepsie, New York 12604 Email: mflores@vassar.edu

***Abstract*—Automatic modulation classification (AMC) is the method of determining the modulation of a signal by trying to learn different features/properties of the received signal. AMC is useful in various civilian and military applications; such as recognizing allies and reducing overhead requirements. Many recent works have shown that different neural network architectures can be an effective tool for performing AMC. There have also been works exploring the use of AMC with equalization (i.e. filtering) to remove distortion caused by the channel. However, depending on the channel conditions, one combination of a neural network and equalizer could perform better than another. In this paper, we investigate applying reinforcement learning (RL) to a system using a blind equalizer followed by a bank of neural networks for the purpose of modulation classification. We use RL to learn/select the optimal equalizer structure and neural network for AMC.**

I. Introduction

Digital modulation is the process of encoding digital information into the amplitude, phase, or frequency of a transmitted signal. Different digital modulation schemes (i.e. PSK, QAM, FSK, PAM) are used to achieve different bit rates, where higher-order modulations allow transmitters to send more bits per symbol. In noisy channels, however, signals modulated using high-order schemes have a higher probability of being incorrectly decoded by receivers [1]. This has motivated the exploration of various adaptive modulation systems, which allow transmitters to determine how best to modulate their data for a particular channel. In military applications, these systems can be used to improve communication security. In civilian applications, these systems can be used to improve data throughput. These adaptive systems, however, require some additional signal overhead to inform receivers of the forthcoming modulation type ahead of every frame. Automatic Modulation Classification (AMC) is useful in civillian settings to reduce such overhead, and is useful in military settings to facilitate electronic warfare efforts [2]. AMC is the process of automatically identifying the modulation used to send a signal based on its physical features alone. In the past two decades, neural networks have become a popular tool for accomplishing AMC [3], [4], [5], [6]. In recent years, neural networks (NNs) have become extremely effective at AMC, achieving high classification accuracy even in complex channels. However, some of the highest performing networks require substantial knowledge of the channel in which they are deployed and the most relevant signal features. This requires additional effort on the part of users to determine those features ahead of time and program them into the network [7], [8], [9]. In this paper, we investigate an AMC system that does not require knowledge of the channel but still achieves high classification accuracy by applying a Reinforcement Learning (RL) algorithm to a bank of several different neural networks, which can determine in real time which network is most effective in the current channel conditions. To improve classification accuracy further, we implement a blind equalizer using the Multi-Modulus algorithm (MMA) ahead of the bank of neural networks. The parameters of the equalizer are then selected by the RL algorithm in real time to optimize its effectiveness.

We focus on two types of channels: High Frequency (HF) channels and fixed Finite Impulse Response (FIR) channels. HF channels characterize the HF radio band, ranging from 3-30MHz, which is often used for long-range communications because signals can be bounced off of the ionosphere before reaching a receiver. Unfortunately, this makes the HF channel particularly noisy because ionospheric conditions fluctuate frequently, causing HF signals to be subject to unpredictable multi-path distortion and making modulation classification more challenging. FIR channels represent an easier scenario where the channel is time-invariant and causes constant distortion.

This paper is organized as follows. Section II will provide a brief overview of current state of the art approaches in AMC. Section III provides a discussion of our blind equalization method. Section IV will report on the bank of neural networks and the training data used. The application of reinforcement learning will be explained in Section V. Results are shown in Section VI, followed by Conclusions in Section VII.

II. Modulation Classification

A. Past Methods

In recent literature, many different methods have been researched and experimented on for AMC. More recently, the novel approach to AMC includes the use of deep learning. Recurrent and convolutional neural networks along with various hybrid networks have been frequently implemented for the task of modulation classification as seen in [3], [5]. In previous research, non deep learning approaches have also been examined such as likelihood-based classification and zero crossing [10], [11]. These past approaches display the myriad of ways to approach this problem, but neural networks have shown to produce the best results. In [3], the best neural network was a convolutional network with a dropout of 0.6 which produced an overall accuracy of 87.4%. This proved that networks were able to match and even outperform some of the expert feature-based detection methods previously used for AMC. From here, recurrent neural networks such as the Long-Short Term Memory (LSTM) and Gated Recurrent Unit (GRU) began being experimented with. In [12] and [13], the classification accuracies were becoming closer to the 90th percentile with the LSTM model reaching close to 90% and the GRU model reaching 91%. However, the use of deep learning is only able to classify well enough to a certain extent. One would assume that deeper, more complex networks would increase classification accuracy, but that is not exactly the case. There are suggestions in previous work that efforts should be put towards improving equalization and synchronization instead [14].

B. Proposed Method and Model

The proposed method we experiment with in this paper is implemented in two different experiments. The first experiment consists of using a bank of neural networks accessed by a reinforcement learning algorithm to classify modulation types in three different HF channels as shown in Figure 1.

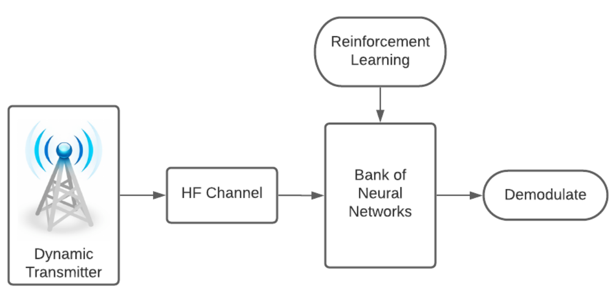

Fig. 1. AMC using Reinforcement Learning in HF Channels

The second experiment is done performed assuming a fixed channel in which blind equalization is implemented before the bank of neural networks. Reinforcement learning is applied to the blind equalizer to determine optimal parameters for equalization and to the bank of neural networks to determine which network provides the highest classification accuracy for the given conditions as shown in figure 2.

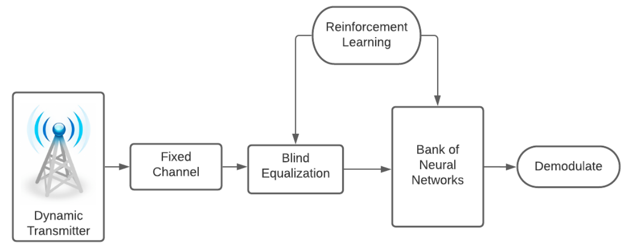

Fig. 2. AMC using Reinforcement Learning and Blind Equalization in a Fixed Channel

III. Blind Equalization

In order to improve modulation recognition, we implement an equalizer ahead of the neural network that uses a blind equalization algorithm to reduce the effects of noise on the signal. The main appeal of blind equalizers is that they do not require a training sequence. The most well-known blind equalization algorithm is the Constant Modulus Algorithm (CMA), originally proposed in [15]. However, this algorithm relies on the assumption that the constellation associated with the received signal has a constant modulus, which is not the case for multiple digital modulation schemes. To address this issue, an extension of the CMA was proposed in [16]. This algorithm, referred to in the literature as the Multi-Modulus Algorithm (MMA), penalizes deviation of the real and imaginary parts of the received I/Q data separately.

The algorithm, adopted from [16] and [17], operates as follows. Given a received noisy signal vector $\mathbf{X}_n = [x_n, x_{n-1}, ..., x_{n-L+1}]$ where x_n is the n^{th} symbol/sample being equalized and L is the length of the equalizer, the MMA produces the equalizer output y_n as follows [17]:

$$y_n = \mathbf{w}_n^T \mathbf{X}_n, \quad (1)$$

where $\mathbf{w}_n$ is the vector of equalizer tap weights at the n^{th} iteration. $\mathbf{w}_n$ is center-tap initialized in the form [0, ..., 0, 1, 0, ..., 0] and updated according to

$$\mathbf{w}_{n+1} = \mathbf{w}_n - \mu e_n \mathbf{X}_n^*, \quad (2)$$

where μ is the step size and e_n is the error value determined by [17]

$$e_n = (y_{n,R})(R_R - y_{n,R}^2) + (y_{n,I})(R_I - y_{n,I}^2)j, \quad (3)$$

where $y_{n,R}$ and $y_{n,I}$ are the real and imaginary components of y_n respectively, while R_I and R_R are the dispersion constants given by [17]

$$R_I = \frac{E\{S_I^4\}}{E\{S_I^2\}} \quad (4)$$

$$R_R = \frac{E\{S_R^4\}}{E\{S_R^2\}}, \quad (5)$$

where S represents the set of all constellations corresponding to the modulation of the signal being equalized; S_I and S_R

correspond to the real and imaginary components of that set respectively.

It is important to note that in the context of AMC, S is unknown when the signal is being equalized because it is determined by the modulation of the received signal. The neural network then classifies based on the *output* of the blind equalizer. In [6], a workaround for this problem was proposed involving a bank of equalizers, each configured with dispersion constants (R) for a different modulation, followed by a “null block” that removes the null output that may be generated by equalizers that are configured for the wrong modulation. In this paper, we propose an alternative workaround by calculating R_I and R_R directly from the received signal rather than its underlying set of constellations. Figure 3 below demonstrates the effectiveness of this algorithm, which we refer to as the “Fully Blind MMA”, for a particular channel when compared to the original MMA with full knowledge of the modulation of the received signal. The channel used is a fixed FIR channel taken from [18] with the following complex-valued coefficients: [-0.005-0.004i 0.009+0.03i -0.024-0.104i 0.854+0.520i -0.218+0.273i 0.049-0.074i -0.016+0.02i]. The equalizer step size used is 0.0001 and the tap length is 7. The number of 16-QAM frames sent is 1,000 and the number of symbols per frame is 10,000.

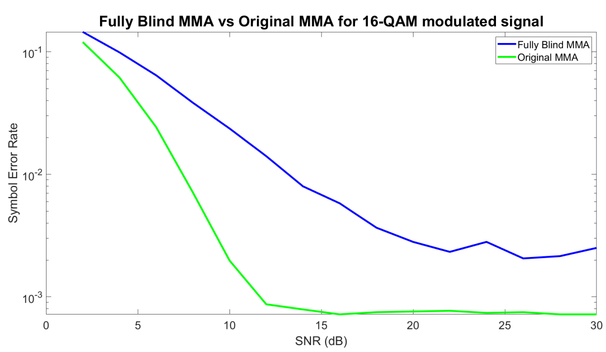

Fig. 3. Plots the performance of the Fully Blind MMA versus the original semi-blind MMA using the Symbol Error Rate (SER) vs Signal to Noise Ratio (SNR) metric, which is common in communication theory.

Note that, although the performance of the fully blind MMA is not equal to that of the original MMA, it performs fairly well considering that the constellation is unknown and follows the same trend as the SNR increases. We have observed this to be the case in different channels and determined that the fully blind MMA is suitable for use in our system.

IV. Neural Networks

With the intuition that different neural networks will excel at classifying different modulation types and produce various accuracies in different channel effects we allow a reinforcement learning algorithm to chose from a bank of neural networks in order to learn the optimal network for the given conditions.

A. Baseline models

Our bank of neural networks are based on different solutions proposed in related literature. We have modified the structure and hyper-parmeters of each network from their original sources for optimal performance with the HF channel and our synthetic dataset.

*1) **CNN**:* [19] The convolutional neural network (CNN) is constructed with 29 layers consisting of convolutional, batch normalization, ReLU activation, pooling, dropout, dense, and softmax activation layers. This network is trained using the stochastic gradient descent with momentum (SGDM) algorithm.

*2) **LSTM**:* [12] The Long Short Term Memory (LSTM) Neural Network is constructed with only 6 layers consisting of LSTM, dense, and softmax activation layers. This network is trained using the adaptive learning rate method (ADAM).

*3) **GRU**:* [13] The Gated Recurrent Unit (GRU) Neural Network is constructed with only 10 layers consisting of GRU, ReLU activation, dropout, dense, and softmax activation layers. This network is trained using the adaptive learning rate method (ADAM).

*4) **LSTM-GRU**:* [20] The LSTM-GRU Hybrid network is constructed with only 8 layers consisting of LSTM, GRU, ReLU activation, dense, and softmax activation layers. This network is trained using the adaptive learning rate method (ADAM).

B. Dataset Generation

We generate the dataset for our AMC system in the HF channel by passing symbols through the three HF channel models given in Table I [21]. Each frame of data is a 2x128 I/Q vector with the in-phase (I) and quadrature (Q) components split into two row vectors. The whole set consists of 10,000 frames per each modulation type fed through a random channel chosen from Table I and a random SNR value ranging from -10 dB to 25 dB (in increments of 5 dB).

MATLAB HF Channel Models		
Channel	**Time Delay**	**Frequency Spread**
Mid-Latitude Quiet	0.5 ms	0.1 Hz
Mid-Latitude Moderate	1 ms	0.5 Hz
Mid-Latitude Disturbed	2 ms	1 Hz

TABLE I
HF Channel Models [21]

The I/Q data [22] is also normalized with unit power over a window length equal to the samples per frame. Table II shows parameters of the data such as the samples per frame and samples per symbol as well as how the data is split for training, validation, and testing. We modulate the data using 8 common schemes given in Table II below.

Neural Network HF Dataset Format	
Modulation Types	8PSK, BPSK, CPFSK, GFSK, PAM4, QAM16, QAM64, QPSK
Samples per Symbol	8
Samples per Frame	128
Sampling Rate	6400
Number of Frames	10,000
SNR Range (dB)	-10 to 25
Number of Training Frames	8,000
Number of Validation Frames	1,000
Number of Testing Frames	1,000

TABLE II
DATASET PARAMETERS PER MODULATION

We generate the dataset for our fixed-channel AMC system using slightly different parameters given in Table III. Each frame of data is passed through one of six different FIR channels taken from [6], given in Table IV, and a range of SNR from 0dB to 20dB chosen randomly.

Neural Network Fixed Channel Dataset Format	
Modulation Types	8PSK, BPSK, PAM4, QAM16, QAM64, QPSK
Samples per Symbol	8
Samples per Frame	128
Sampling Rate	200,000
Number of Frames	10,000
SNR Range (dB)	0 to 20
Number of Training Frames	8,000
Number of Validation Frames	1,000
Number of Testing Frames	1,000

TABLE III
DATASET PARAMETERS PER MODULATION

FIR Channels	
Channel 1	[0.067+0.106i -0.226-0.966i]
Channel 2	[0.006+0.019i 0.203-0.963i -0.040+0.169i]
Channel 3	[0.235+0.146i -0.087-0.033i 1.036+1.588i -0.224-0.160i]
Channel 4	[-0.078+0.276i -0.596-0.344i 0.640-1.734i -0.335+0.009i -0.334+0.062i]
Channel 5	[0.275-0.516i -0.309+0.603i 1.577+0.948i 0.182+0.386i 0.239-0.198i -0.160+0.203i]
Channel 6	[-0.154-0.061i 0.533+0.377i 0.657+0.465i 1.856-1.168i -0.114+1.100i -0.132+1.055i -0.637+0.245i]

TABLE IV
CHANNELS CHOSEN RANDOMLY FOR TRAINING PURPOSES FROM [6]

V. REINFORCEMENT LEARNING

Reinforcement learning (RL) refers to the automated process of repeatedly selecting the best option from a set to solve a problem in real time with no prior knowledge of the effectiveness of each option. RL is characterized by the choice between “exploring” new options or “exploiting” options that have already shown promising results. For the purpose of AMC, we use RL to select the optimal parameters for the blind equalizer and the best neural network to maximize modulation classification accuracy in a given channel. Our chosen mechanism for RL is known as the Upper Confidence Bound (UCB) algorithm. This algorithm approaches the “exploitation versus exploration” dilemma by being optimistic about the actions that it has explored the least. The UCB algorithm is given below. [23], [24]

$$A_t = argmax[Q_t(a) + c\sqrt{log(t)/N_t(a)}] \tag{6}$$

Where A_t represents the action the agent will choose at time t. $Q_t(a)$ is the action value estimate of action a. c is the confidence value used to balance exploration and exploitation and $N_t(a)$ is the number of times action a has been chosen. Given a finite number of actions available to the agent, this equation will select the action with the highest perceived average. The highest perceived reward is dependent on $Q_t(a)$, the exploit term, which is analogous to a history of past rewards obtained by choosing action a. The second term, $c\sqrt{log(t)/N_t(a)}$, is the explore term that decreases as more actions are performed; hence, the agent becomes more confident about the action that will yield the highest reward as it gains more experience [23], [24].

Reinforcement Learning Parameters	
Algorithm	Upper Confidence Bound
C value	0.5
Number of Frames	88,000 (11,000 per modulation)
Reward	Minimum Average Distance

TABLE V
PARAMETERS FOR RL ON BANK OF NEURAL NETWORKS

In our system models, the actions available to the RL agent consist of different parameters to be used by the equalizer and/or different neural network architectures to be used for classification. The agent selects a new action for each new frame of data. Our chosen reward scheme is the average distance between symbols in the equalized signal and the true constellations to which they are demodulated. This scheme is based upon our intuitive assumption that there exists a positive correlation between the accuracy of the modulation classification and the closeness of the classified points to the corresponding constellations. The agent aims to select the action that will return the lowest reward value in order to minimize the average distance, ideally maximizing classification accuracy.

VI. Results

A. RL applied to Blind Equalizer and Bank of Neural Networks

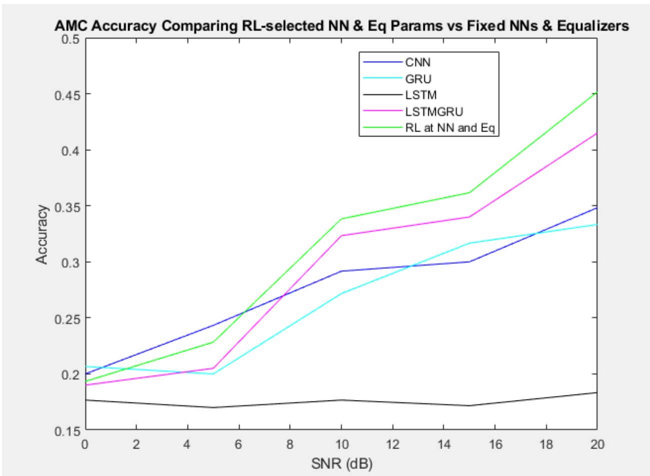

Fig. 4. Reinforcement learning performance on the SNR range of 0 to 20dB. Reinforcement learning selection outperformed other neural networks in accuracy from 8dB to 20dB.

As seen in the above graph, the optimal neural network and equalizer parameters selected by the UCB reinforcement learning algorithm outperforms other neural networks from signal to noise ranges of 8dB to 20dB. When RL selects the step size, equalizer length, and choice of neural network, classification accuracy increases. The channel is sufficiently challenging that classification accuracy does not exceed 50%. To improve upon this, individual networks could be better tuned to fit the channel or an "easier" channel could be used.

B. Reinforcement Learning on a Bank of Neural Networks

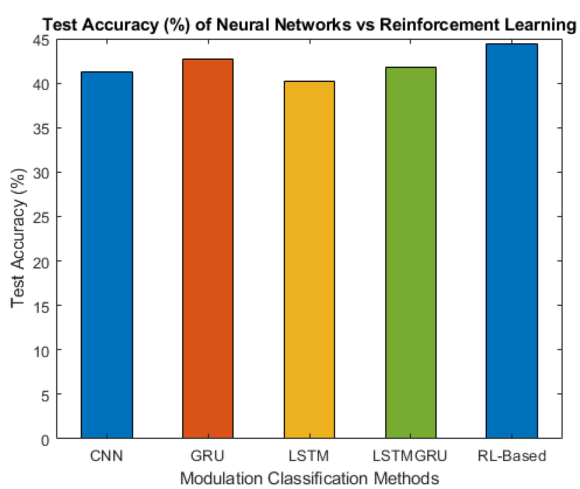

Fig. 5. Reinforcement learning performance on the SNR range -10 to 0 dB. Reinforcement learning performed the best with a 44.4125% accuracy rate with the second best performance from the GRU being 42.7625%.

We found that the reinforcement learning model consistently receives better accuracy rates than neural networks at low SNRs, namely -10, -5, and 0 dB. This discovery is important as classifying modulations at low SNR values is a difficult task most classification techniques struggles with. Figure 4 depicts a bar graph of the accuracy of the reinforcement learning model in comparison to the neural networks from our bank of neural networks when the reinforcement learning model is trained solely on data with a SNR range of -10 to 0 dB. If the reinforcement learning model was selecting the neural networks at random, its accuracy rate would be the average accuracy of the neural networks, which is 41.521875%. However, our results show that the accuracy of the RL-selected network is 3% increase above the average and 2% above the accuracy of the best performing individual network. This demonstrates that reinforcement learning can consistently outperform individual neural networks in the given SNR range proves that the network that is most effective at AMC changes as channel effects and noise vary. Reinforcement learning enables us to identify and exploit the most effective network in real time in order to achieve very high accuracy classification without any prior knowledge of which network will perform the best.

The reinforcement learning model is also able to receive higher accuracy rates at higher SNR values, namely 10, 15, 20, and 25 dB, and comparable accuracy rates at 5 dB. These results were obtained by running the reinforcement learning model on 11,000 frames per modulation and calculating the accuracy based on the results of the last 1,000 frames.

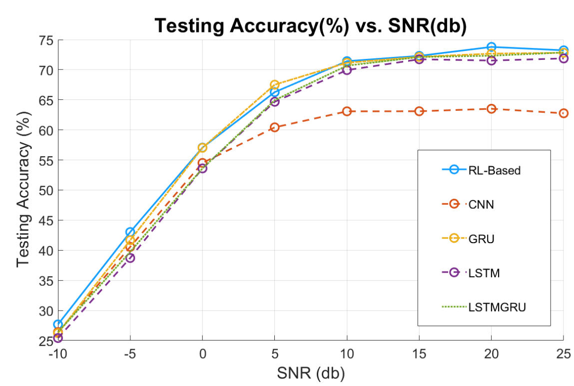

Fig. 6. Performance of RL-selected neural network versus individual networks, each trained on the same data.

The generated frames, however, must be trained solely on a fixed SNR in order for the reinforcement learning model to be able to outperform the neural networks at the respective SNR. Additionally, the ability of the reinforcement learning model to outperform the neural networks at higher SNR values is inconsistent and was obtained through repetitive testing, i.e. the reinforcement learning model was repeatedly tested on a newly generated 11,000 frames per modulation and if it did not receive a higher accuracy than the neural networks then all variables were reset, new frames were generated, and the model was run again.

VII. Conclusion

Using reinforcement learning (RL) to select from a bank of pre-trained neural networks, we are able to demonstrate that the RL-assisted model outperforms the individual networks that it selects. It appears that different neural networks excel at classifying modulations given different channel effects and noise levels; reinforcement learning is able to recognize and exploit this. The worst case for RL occurs when one neural network outperforms all the other neural networks for every type of channel effect and noise level. In such a situation, the RL-assisted model cannot perform as well as the best individual network because it explores some of the lesser-performing networks, which necessarily cause it to lose some accuracy. However, RL is valuable even in that case because it is able to closely track the accuracy of the best-performing network without any prior knowledge of which network would be most effective. This eliminates the need for extensive offline testing to determine the most effective individual network because RL can be trusted to determine that on its own. Our work in this paper demonstrates the potential of RL to combine the best parts of multiple classification methods to optimize testing accuracy.

A. Future Work

The primary avenues for future work are in training more specialized networks, investigating other reward schemes, and varying the c-value of the UCB algorithm.

In the experiment shown in Figure 6, we were able to identify a set of parameters in which the RL agent was able to outperform the individual networks except at 5dB SNR, but this was a rare case. We observed that one type of neural network (most often the GRU) outperformed the others for most channels and noise levels. One example of this is shown in Figure 7. This represents the worst case for RL, as described in the Conclusion above. To remedy this, specialized networks should be trained so that different networks are better for different channel conditions. This would allow the RL agent to learn the best network for each set of conditions and select a new network when conditions change, thereby achieving higher accuracy than any individual network could across a wide range of channel effects.

The effectiveness of the UCB RL agent is highly dependent on its parameters, including its associated c-value and reward scheme. We have chosen a reward scheme that attempts to minimize the average distance between the equalized symbols and the set of constellations associated with the classified modulation. While intuitive, this may not be the best way to reward the RL agent. Further research should focus on identifying a superior reward scheme.

We also note that the weight of the "explore" term increases with SNR if the c-value remains constant. This is because a higher SNR value corresponds to less distortion and therefore a lower average distance between received points and constellations. Thus, the exploit term grows smaller on average, which may cause the "explore" term to dominate it. Further work should focus on determining the significance of this effect and possibly varying the c-value to counteract it.

Additional work might also focus on testing RL with different networks, different channels, and different modulation types. This paper focuses on a very limited number of each to provide a preliminary demonstration of the potential effectiveness of RL applied to AMC. By increasing the variety of tools available to the RL agent, its effectiveness may change drastically.

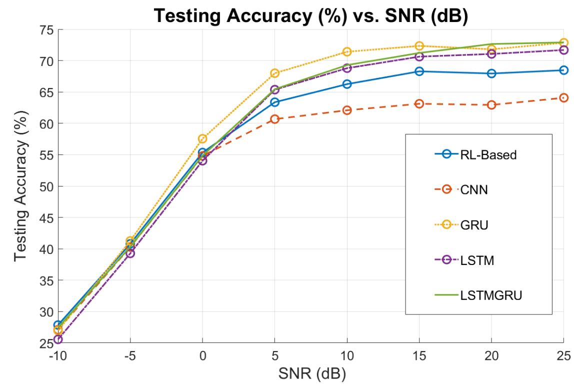

Fig. 7. Performance of reinforcement learning model trained on a range of SNR values

References

[1] J. Walrand and P. Varaiya, "Chapter 7 - wireless networks," in *High-Performance Communication Networks (Second Edition)*, second edition ed., J. Walrand and P. Varaiya, Eds. San Francisco: Morgan Kaufmann, 2000, pp. 305–361. [Online]. Available: https://www.sciencedirect.com/science/article/pii/B9780080508030500125

[2] *Introduction*. John Wiley & Sons, Ltd, 2014, ch. 1, pp. 1–17. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/9781118906507.ch1

[3] T. J. O'Shea, J. Corgan, and T. C. Clancy, "Convolutional radio modulation recognition networks," *Communications in Computer and Information Science*, vol. 629, pp. 213–226, 2016.

[4] B. Ramkumar, T. Bose, and M. S. Radenkovic, "Combined blind equalization and automatic modulation classification for cognitive radios," in *2009 IEEE 13th Digital Signal Processing Workshop and 5th IEEE Signal Processing Education Workshop*, 2009, pp. 172–177.

[5] S. Scholl, "Classification of radio signals and hf transmission modes with deep learning," 2019.

[6] X. Ji, J. Wang, Y. Li, Q. Sun, and C. Xu, "Modulation recognition in maritime multipath channels: A blind equalization-aided deep learning approach," *China Communications*, vol. 17, no. 3, pp. 12–25, 2020.

[7] B. Kim, J. Kim, H. Chae, D. Yoon, and J. W. Choi, "Deep neural network-based automatic modulation classification technique," in *2016 International Conference on Information and Communication Technology Convergence (ICTC)*, 2016, pp. 579–582.

[8] J. H. Lee, J. Kim, B. Kim, D. Yoon, and J. W. Choi, "Robust automatic modulation classification technique for fading channels via deep neural network," *Entropy*, vol. 19, no. 9, 2017. [Online]. Available: https://www.mdpi.com/1099-4300/19/9/454

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

[10] F. Hameed, O. A. Dobre, S. Member, D. C. Popescu, and S. Member, "On the Likelihood-Based Approach to Modulation Classi fi cation," vol. 8, no. 12, pp. 5884–5892, 2009.

[11] S. Z. Hsue and S. S. Soliman, "Automatic modulation classification using zero crossing," *IEE Proceedings, Part F Radar Signal Process.*, vol. 137, no. 6, pp. 459–464, 1990.

[12] S. Rajendran, W. Meert, D. Giustiniano, V. Lenders, and S. Pollin, "Deep learning models for wireless signal classification with distributed low-cost spectrum sensors," *IEEE Transactions on Cognitive Communications and Networking*, vol. 4, no. 3, pp. 433–445, 2018.

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

[14] T. J. O'Shea and N. West, "Radio Machine Learning Dataset Generation with GNU Radio," *6th GNU Radio Conf.*, pp. 1–6, 2016.

[15] D. Godard, "Self-recovering equalization and carrier tracking in two-dimensional data communication systems," *IEEE Transactions on Communications*, vol. 28, no. 11, pp. 1867–1875, 1980.

[16] K. N. Oh and Y. O. Chin, "Modified constant modulus algorithm: blind equalization and carrier phase recovery algorithm," in *Proceedings IEEE International Conference on Communications ICC '95*, vol. 1, 1995, pp. 498–502 vol.1.

[17] A. Moussa, M. Pouliquen, M. Frikel, S. Bedoui, K. Abderrahim, and M. M'Saad, "An overview of blind equalization algorithms for digital communications," in *2019 19th International Conference on Sciences and Techniques of Automatic Control and Computer Engineering (STA)*, 2019, pp. 491–496.

[18] S. Abrar and R. Axford, "Sliced multi-modulus blind equalization algorithm," *Etri Journal*, vol. 27, 06 2005.

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

[20] A. U. Muhammad, X. Li, and J. Feng, "Using lstm gru and hybrid models for streamflow forecasting," in *Machine Learning and Intelligent Communications*, X. B. Zhai, B. Chen, and K. Zhu, Eds. Cham: Springer International Publishing, 2019, pp. 510–524.

[21] I.-R. F.1487, "Testing of hf modems with bandwidths of up to about 12 khz using ionospheric channel simulators," *Group*, vol. 1487, 2000.

[22] "Modulation classification with deep learning." [Online]. Available: https://www.mathworks.com/help/deeplearning/ug/modulation-classification-with-deep-learning.html

[23] K.-Y. Kao and I.-H. Chen, "Maximal expectation as upper confidence bound for multi-armed bandit problems," in *2014 IEEE 7th Joint International Information Technology and Artificial Intelligence Conference*, 2014, pp. 325–329.

[24] S. Roberts, "The Upper Confidence Bound (UCB) Bandit Algorithm," 2020.