Deep Learning Based Identification of Wireless Protocols in the PHY layer

Alex Berian, Irmak Aykin, Marwan Krunz, and Tamal Bose
Department of Electrical and Computer Engineering
The University of Arizona
Tucson, Arizona, USA
{berian, aykin, krunz, tbose}@email.ariozona.edu

***Abstract*—The need for Artificial Intelligence algorithms for future Cognitive Radio (CR) systems is unavoidable. For a CR to operate as best as possible it must identify who is present in spectrum of interest, and what they are doing (i.e. jamming, communicating, rogue transmission, etc.). Using this information, a CR can accordingly decide what to do next. Furthermore, being able to determine which wireless protocols are occupying spectrum is an important ability in heterogeneous wireless networks. In this work, we investigate the robustness of various Neural Network (NN) algorithms for classification of wireless protocols when looking at base-band In-phase/Quadrature (IQ) data without needing to decode. We propose a spectrum sensing algorithm based on NNs or other similarly behaved classification algorithms for identifying wireless technologies occupying spectrum. In previous literature, using base-band IQ data, researchers have shown that NN models can classify different modulation formats with promising accuracy. This work explores the potentials, usage, and limitations of using base-band IQ data for classifying various wireless network protocols that employ the same modulation format.**

***Index Terms*—Cognitive Radio, Neural Networks, Signal Classification, Wireless Protocols**

I. Introduction

Signal classification is an essential addition for more effective cognitive radio (CR) technology as spectrum becomes increasingly scarce. It is common practice in modern CRs to use spectrum sensing as means to avoid occupied spectrum [1]. Modern wireless protocols operate on unlicensed shared spectrum, and being able to identify other existing technologies on the spectrum can improve friendly coexistence. To further improve the performance of a CR, it must be able to identify what is present in spectrum of interest, and act accordingly. When identifying what's in the spectrum, it can be complex and often infeasible to decode what's being observed, so a CR will have to settle for simply identifying the signal. A significant number of wireless protocols employ a form of orthogonal frequency division multiplexing (OFDM) which makes it difficult for existing modulation classification algorithms to differentiate between protocols.

This work explores the use of various neural network (NN) architectures for classifying wireless protocols of similar modulation formats. We analyse two main categories of NN: feed forward NN (FNN), and convolutional NN (CNN). We found that a specific kind of CNN known as a residual NN (ResNet) [2] was overall the most effective in classifying wireless protocols.

There are 2 main forms of signal classification: likelihood base, and feature based. Many likelihood based methods make optimal decisions, however they are often too computationally complex for real time implementation. Feature based classification usually involves extraction of features from a signal (i.e. bandwidth, carrier frequency, high order statistics, etc.), and use those features as inputs to some kind of classification algorithm which could be machine learning or something else. This is known as expert feature based classification.

We and others speculate that it may be better for a machine learning model to formulate it's own features to use in the classification process. There exist some works [3], [4], [5], [6] that explore this possibility, and they achieve competitive results with other traditional feature based approaches. However, we have not seen any prior work analyzing the potential or usage of machine learning in classifications of wireless protocols. In section V, we present a technique for applying NNs in real-time systems to identify when wireless protocols are present in the spectrum.

When multiple protocols are utilized in the same portion of spectrum being able to identify who else is out there can be crucial to friendly co-existence. There have been multiple works that describe the challenges in coexistence of multiple wireless technologies in the same bands [7] [8], and it is difficult for devices designed for one protocol to identify signals from another protocol.

In military applications where different protocols are co-existing, but not in a friendly manner, knowing if someone else is in the spectrum at a very low signal to noise ratio (SNR) can be important for radio location of enemy transmitters. A CR that has some prior knowledge of the wireless behavior of the protocols, or their operators, can even formulate and appropriate transmission plan to maximize throughput and secrecy.

Initially, in an effort to explore the robustness of NNs in this task we tested classification accuracy when passing randomly place windows of 1024 samples within frames or between frames to the NNs, as well as testing the performance when only looking at the beginning of packets. Furthermore, we passed the IQ data from MATLAB waveform generations through various channel models such as Rician, Rayleigh, and COST2100 [9]. NNs have been shown to not perform well under such multi-path channel models [4], and our simulations confirm this. As a result, we only analyze additive white Gaussian noise (AWGN) channels in this paper.

The organization of this paper is as follows. In section II we discuss related signal classification algorithms and works that could benefit by employing the algorithm we present. Section III provides an overview of NN algorithms we analyze. The algorithm we present in this paper is described in V. Performance analysis of the algorithm and other discussion is found in section VI. Section VII concludes with a brief recap of the algorithm, and other key takeaways from our results.

II. Related Work

There has been extensive literature in recent years regarding classification of modulation formats using machine learning. [4] analysed the performance of various NN architectures and their hyper-parameters in modulation classification with base-band IQ samples. The authors of [3] show that classification accuracy of modulations can be improved through specialized training strategy, where they train a single NN in stages to differentiate modulation formats in a hierarchical manner. [6] also explored a hierarchical manner of modulation classification, but these authors used multiple NNs each specializing in narrowing down the modulation class which ultimately performed better than [3].

Few existing works have explored protocol classifications as well. [10] explores classifying between slotted ALOHA and Time Division Multiple Access (TDMA) by using power mean and power variance over time as features to be classified with a Support Vector Machine (SVM). Some works such as [5] have explored using IQ data with deep learning algorithms. to classify different applications being used under the same communication protocol.

III. Neural Network Architectures

In this section, we will briefly describe the various NN architectures explored in this work. This section only gives a very brief discussion of what NN techniques are explored in this paper, and we encourage the reader to look in reference [11] if they desire more explanation of these techniques.

Depending on the problem, a user must decide what form the output of their NN model will take on. For a problem where a user wants to classify between two classes, it is common to use a single neuron as the output, with a sigmoid activation function. For example, if there are two classes, 0 and 1, and if an input to an NN with a single sigmoid output neuron outputs .7, the user would classify this input as belonging to class 1. For a two class problem with a sigmoid output the most common loss used is binary cross-entropy, however is is not uncommon to see other loss functions used.

Most NNs are trained using the back-propagation algorithm. Back-propagation is an efficient algorithms to calculate the gradient of multiple variables in a large complex equation. For NNs, it is used to calculate the gradient of weights and biases with respect to the loss.

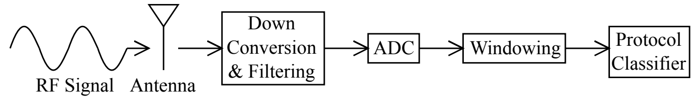

Fig. 1. Receive chain with indication of where to place a wireless protocol classifier.

In this work, we have three classes to choose from, and we only want to pick one. This calls for the softmax output layer. In classification, the softmax output layer is designed for each output neuron to correspond to each of the classes in question, and the sum of all outputs is always equal to 1. To train with softmax as the output layer it is typical to use one-hot encoded vectors as the desired output from the NN. A one-hot encoded vector is vector of all 0s, and one 1 to indicate the desired class to identify. The output of a softmax layer can be viewed as a vector of probabilities. Each element in the vector is the probability that the input belongs to the class corresponding to it's respective output neuron. The largest output from the layer is selected as the class. The loss typically used with softmax output layers is categorical cross-entropy.

A deep FNN consists of many dense layers. A dense layer is multiplying the input to the layer by a matrix of weights, then adding that result to a vector of biases, then passing the results into an element-wise nonlinear activation function (such as ReLU, sigmoid, tanh, ect.). That result is then passed into other layers depending on the network architecture.

A CNN consists of convolutional layers. A convolutional layer is a simple convolution operation where we convolve the input to the layer with one or more filters of trainable weights, and the weights are modified according to their respective gradients during training.

IV. Data Collection

In this paper, to demonstrate the ability of NNs to classify between wireless protocols we classify between LTE, WiFi, and 5G frames. We collect baseband IQ samples of the waveforms for these frames using MATLAB toolboxes. MATLAB has toolboxes that contain waveform generation of all these three target classes: LTE, WiFi, and 5G. Within these toolboxes, various parameters are randomized for the signals. The idea behind doing this is that if we get as much variation within the different classes as possible, we can prove just how effective NNs are in finding the differences anyway. Additionally, in practice a user will not have control over signals that belong to different protocols, so being able to identify a large variety within those protocols is important. Table I displays all of the parameters that were changed and/or randomized within the protocols. All of the protocols were constrained to work for single-input-single-output (SISO) only.

V. System Design

FNNs and CNNs are constricted to a finite view of signals. As a result, it is best if they focus on the beginnings of

Protocol	Parameter	Possible Values
LTE	RC	R.0, R.1, R.2, R.3, R.4, R.5, R.6, R.7, R.8, R.9, R.10, R.11, R.12, R.13, R.14, R.25, R.26, R.27, R.28, R.31-3A, R.31-4, R.43, R.44, R.45, R.45-1, R.48, R.50, R.51, R.6-27RB, R.12-9RB, R.11-45RB
	CellRefP	1
	PDSCH # Layers	1
	CFI	1, 2, 3
	Ng	Sixth, Half, One, Two
	PHICHDuration	Normal, Extended
	SSC	0, 1, 2, 3, 4, 5, 6, 7, 8, 9
5G	Cell ID	0, 1, 2, …, 99, 100
	# of Subframes	1
	SSB Block Pattern	Case A, Case B
	SSB Transmitted	Random Binary Vector of Length 4
	SSB Periodicity	5, 10, 16, 40, 80
	Cyclic Prefix	Normal, Extended
	BWP Size	25, 50
	BWP Separation	10, 50
	PDSCH Modulation	QPSK, 16QAM, 64QAM, 256QAM
	PDSCH RV Sequence	Random Ternary Vector of Length 4
	PDSCH Mapping Type	A, B
	DM-RS First Symbol Position	2, 3
	# Front Loaded DM-RS Symbols	1,2
	Other DMRS Symbol Positions	0, 1, 2, 3
	PDSCH Scrambling Identity	0, 1, 2, …, 65535
	PDSCH Scrambling Initialization	0, 1
Wi-Fi	MCS	0, 1, 2, 3, 4, 5, 6, 7, 8, 9
	APEP Length	2^9, 2^10, 2^11
	Guard Interval	Short, Long
	Group ID	0, 63
	Partial AID	0, 1, 2, …, 511
	Channel Bandwidth	CBW20, CBW40, CBW80, CBW160

TABLE I
PROTOCOL PARAMETER RANDOMIZATION OPTIONS

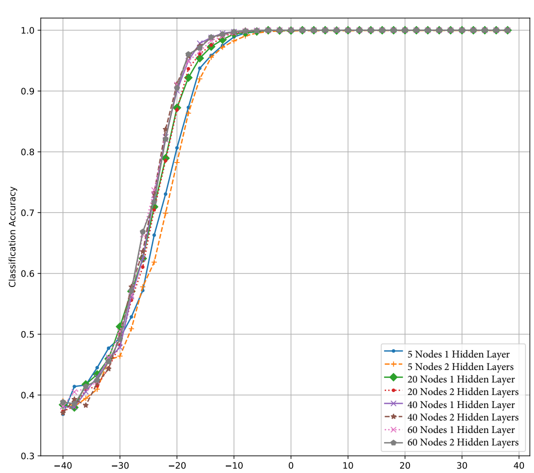

Fig. 2. Variable sizes of FNN for 1024 IQ samples under AWGN

frames as they are the most unique and consistent part of the waveform for most protocols. Figure 1 shows where a user would implement any classification algorithm designed to look at a finite number of samples for the purpose of identifying wireless protocols. After a signal is sampled a window of an application specigic size and type (we explore rectangular) will shift with every received sample, behaving effectively like a queue of IQ samples. In this work, we focus on NNs, however users can employ any classification algorithm that looks at constant input vector sizes as they see fit.

The classification algorithm must be trained offline. Users should collect IQ data at their target sample rates offline, and be able to identify the beginning of frames, and label which protocols they belong to manually. To train the machine learning algorithm, the user will only take a window from the beginnings of the recorded frames and form them as input vectors to the machine learning algorithm for training. Once trained, the user will employ their protocol classifier as depicted in figure 1. The classifier will look at a sliding window of IQ samples directly from an analog to digital converter (ADC), and the classifier will decide whether or not certain protocols are present in the spectrum.

There is significant variation in performance if the samples the network is looking are not at the beginning of a frame. When training and testing CNNs and FNNs on random locations of frame, we only saw a 40% classification accuracy under the best conditions. With 3 classes, 33% accuracy is as good as random guessing. In Figure 3, we see what happens when a window slides across a couple of frames as a FNN processes the window. Incorrect output neurons tend to get very close to 1, however they never surpass the maximum output we observe for the correct class. When implementing a neural network for protocol classification, the user must generate their own plots similar to Figure 3 with their own measurements and decide a threshold for each output neuron. When the threshold is exceeded, that is indication that a particular protocol is present in the spectrum.

VI. PERFORMANCE ANALYSIS

In this section we analyze the testing accuracy of the NNs. Training time never exceeded 1 hour on a relatively slow computer without a GPU. All NN design, training, and testing was done using the Keras library within Tensorflow.

In Figure 2 we simulate probability of correct classification on the beginning of frames as SNR increases in an AWGN channel model. We see that there isn't very much gain in performance when increasing the depth or the width of the FNN. It is worth noting that classification accuracy with a FNN with one hidden layer and a width of 5 neurons performed slightly worse than a width of 20 neurons. The accuracy between width 40 and 60 are the same. We also see that increasing the depth of the network did not have any effects on performance improvement.

NNs can be extremely computationally complex, and reducing the size of a NN reduces the computational complexity. In Figure 4 we show probability of correct classification vs

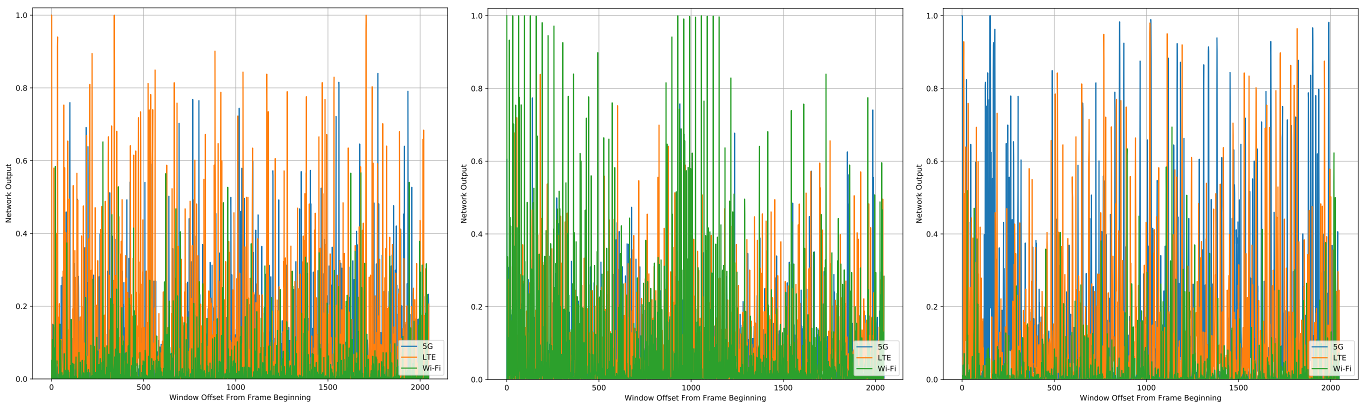

Fig. 3. FNN outputs as the sliding window shifts across a couple of frames. From left to right, LTE, Wi-Fi, ang 5G are analyzed shown respectively.

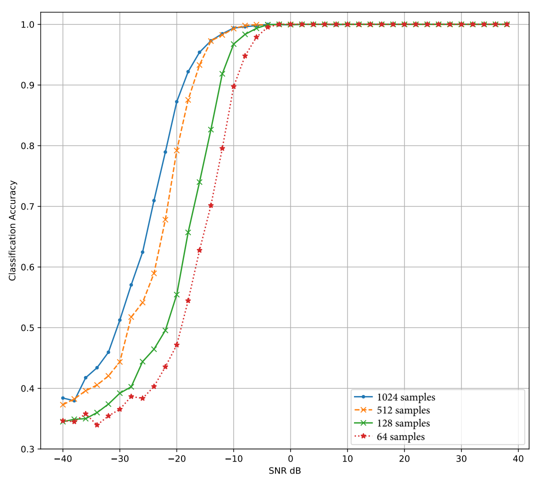

Fig. 4. Comparing classification accuracy under AWGN for different window sizes using a FNN of depth 1 and width 20

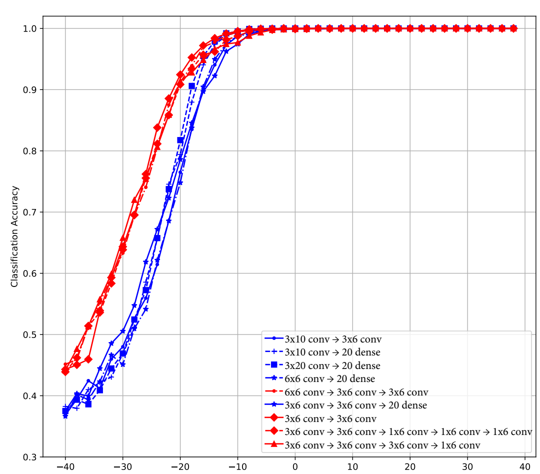

Fig. 5. Comparing varying CNN architectures with and without pooling on 1024 IQ samples under AWGN. Blue curves denote pooling is present in the NN.

SNR on the beginning of frames, as window size decreases. A different FNN of depth 1 and width 20 was trained and tested on their respective window sizes to generate the resulte in Figure 4. As one would expect there is a decrease in performance as the window size decreases, because there is less information for the NN to learn from. The convergence to 100% classification accuracy with respect to SNR only changes by a few dB.

It is common in CNNs to use max pooling. A max pooling layer takes a sliding window across it's input, and throws away all input from each window that are not the maximum in their respective windows. For the case of IQ samples, we choose the max based on magnitudes of the samples. In Figure 5 we show the effects of using max pooling windows of size 2 and 3 after convolutional layers when AWGN is the only channel impairment. As an example to explain the notation in Figure 5, *3x10 conv* means that 3 parallel convolutional filters of 10 taps are used in that layer. Max pooling clearly does not perform as well at low SNR, but CNNs with max pooling do converge to 100% classification at around the same SNR regions as CNNs without max pooling. Though performance is not quite as good at low SNR with pooling, it is important to note that max pooling significantly reduces computational complexity in a NN. We recommend users employ max pooling, because they still converge to 100% classification accuracy at the same SNR as those without, and they significantly reduce computational complexity.

ResNets were first presented in [2] for image classification. Since then they have been one of the most popular NN architectures, because of their great performance across many tasks. The logic behind why ResNets are so effective is because they keep residual information of early layers all the way throughout a deep NN through addition. This allows the back-propagation algorithm to easily calculate the gradients for early layers, while still getting the benefits of having a deep NN. We show the performance of a few ResNet architecture in Figure 7. Since there is virtually no gain in increasing the depth of the network, we do not recommend doing so.

Assuming one protocol will be the only thing present in the spectrum is not practical. LTE, Wi-Fi, and soon 5G will all

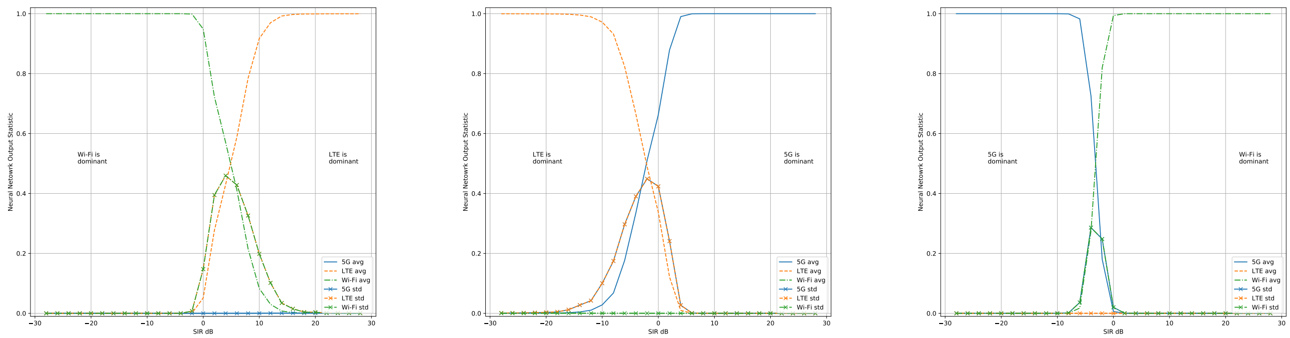

Fig. 6. Average and standard deviation of a FNN's outputs when 2 different protocols are superposed at different power ratios. From left to right, Wi-Fi & LTE, LTE & 5G, 5G & Wi-Fi are shown respectively.

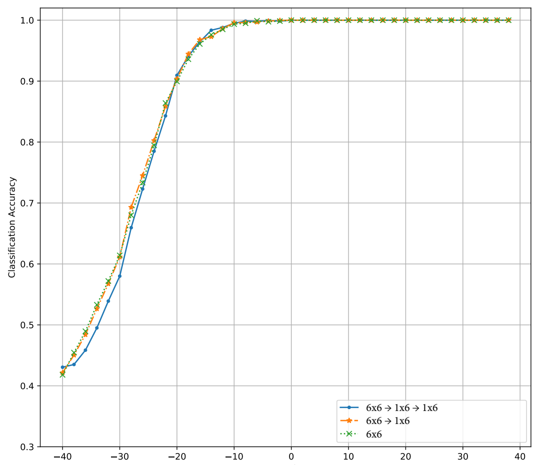

Fig. 7. Comparing classification accuracy under AWGN for different ResNet architectures.

be occupying the same spectrum in certain bands. To explore the effects of interference, we have superposed signals of two different protocols over each other at varying Signal to Interference Ratio (SIR). The phrase signal to *interference* ratio is for lack of a better term, because when one protocols is interference at a very low SIR, then it may as well be the main signal. In Figure 6 we analyze the outputs of a FNN that looks at 512 samples at a time. The FNN sees only the beginning of frames of the two present superposed protocols, which is a worst case scenario that is rare in practice.

For each data point in Figure 6 a FNN processes 1000 different superposed signals consisting of 2 protocols of a constant SIR in batches, and the average (avg) and standard deviation (std) of the values given by the 3 output neurons are plotted according to the 1000 superposed signals. We observe very high std of the outputs when the avg of two outputs are similar. Figure 6 also indicates that when 2 signals are superposed with similar power, both the std and avg are about 0.5. This indicates that the NN is consistently outputting high confidence in one class over the interfering class. This is actually quite common in NNs utilizing softmax output layers, because the NN is only ever taught to output one-hot vectors during training. At first glance, one may think this is disadvantageous. As described in section V we determine if a protocol is present by observing a threshold being passed by an output neuron. The NNs tendency to output high numbers in one class actually helps to pass the threshold, which is desirable because we want a device utilizing this algorithm to determine that both protocols are present in the spectrum.

VII. Conclusion

This paper proposes and analyzes a deep learning based algorithm for identifying if certain protocols are present in spectrum. Users train a NN to identify the beginning of frames belonging to certain protocols offline, then when applied in practice window of baseband IQ samples from an ADC is passed into the NN. When an output neuron passes a user defined threshold, this is indication the the protocol associated with that neuron is present in the spectrum. We showed that depth does not provide any performance improvement. Using max pooling in a CNN is a good idea, because it reduces computational complexity, and converges to 100% classification accuracy at around the same SNR as NNs that don't use max pooling. We showed that reducing window size does not strongly affect the SNR ranges of good classification accuracy. When multiple protocols are present in the spectrum the algorithms is still able to identify which protocols are present.

Acknowledgment

The authors of this paper would like to thank Broadband Wireless Access Center (BWAC) for funding this research. BWAC is an industry sponsored subset of the National Science Foundation (NSF).

References

[1] I. F. Akyildiz, W.-Y. Lee, M. C. Vuran, and S. Mohanty, "Next generation/dynamic spectrum access/cognitive radio wireless networks: A survey," *Computer Networks*, vol. 50, no. 13, pp. 2127 – 2159, 2006. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1389128606001009

[2] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016, pp. 770–778.
[3] G. Vanhoy, N. Thurston, A. Burger, J. Breckenridge, and T. Bose, "Hierarchical modulation classification using deep learning," in *MILCOM 2018 - 2018 IEEE Military Communications Conference (MILCOM)*, Oct 2018, pp. 20–25.
[4] N. E. West and T. O'Shea, "Deep architectures for modulation recognition," in *2017 IEEE International Symposium on Dynamic Spectrum Access Networks (DySPAN)*, March 2017, pp. 1–6.
[5] T. J. O'Shea, S. Hitefield, and J. Corgan, "End-to-end radio traffic sequence recognition with recurrent neural networks," in *2016 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, Dec 2016, pp. 277–281.
[6] K. Karra, S. Kuzdeba, and J. Petersen, "Modulation recognition using hierarchical deep neural networks," in *2017 IEEE International Symposium on Dynamic Spectrum Access Networks (DySPAN)*, March 2017, pp. 1–3.
[7] C. Cano, D. Lopez-Perez, H. Claussen, and D. J. Leith, "Using lte in unlicensed bands: Potential benefits and coexistence issues," *IEEE Communications Magazine*, vol. 54, no. 12, pp. 116–123, December 2016.
[8] M. Hirzallah, W. Afifi, and M. Krunz, "Full-duplex-based rate/mode adaptation strategies for wi-fi/lte-u coexistence: A pomdp approach," *IEEE Journal on Selected Areas in Communications*, vol. 35, no. 1, pp. 20–29, Jan 2017.
[9] L. Liu, C. Oestges, J. Poutanen, K. Haneda, P. Vainikainen, F. Quitin, F. Tufvesson, and P. D. Doncker, "The cost 2100 mimo channel model," *IEEE Wireless Communications*, vol. 19, no. 6, pp. 92–99, December 2012.
[10] and, , and and, "Mac protocol classification in a cognitive radio network," in *The 19th Annual Wireless and Optical Communications Conference (WOCC 2010)*, May 2010, pp. 1–5.
[11] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016, http://www.deeplearningbook.org.