Evaluating Deep Neural Networks to Classify Modulated and Coded Radio Signals

Phui San Cheong[1], Miguel Camelo[2](✉), and Steven Latré[2]

[1] University of Antwerp, Antwerp, Belgium
phuisan.cheong@student.uantwerpen.be
[2] IDLab Research Group, Department of Mathematics and Computer Science, University of Antwerp - imec, Antwerp, Belgium
{miguel.camelo,steven.latre}@uantwerpen.be

Abstract. Cognitive Radio (CR) systems allow optimizing the use of the shared radio spectrum and enhancing the coexistence among different technologies by efficiently changing certain operating parameters of the radios such as transmit-power, carrier frequency, and modulation and coding scheme in real-time. Dynamic Spectrum Access (DSA), which allows radios to dynamically access and use the unused spectrum, is one of the tasks that are fundamental for a better use of the spectrum. In this paper, we extend the previous work on Automatic Modulation Classification (AMC) by using Deep Neural Network (DNNs) and evaluate the performance of these architectures on signals that are not only modulated but are also encoded. We call this the Automatic Modulation and Coding Scheme Classification problem, or AMC^2. In this problem, radio signals are classified according to the Modulation and Coding Scheme (MCS) used during their transmission. Evaluations on a data set of 802.11 radio signals, transmitted with different MCS and Signal to Noise Ratio (SNR), provide important results on the impact of some DNN hyperparameters, e.g. number of layers, batch size, etc., in the classification accuracy.

Keywords: Cognitive Radio · Dynamic Spectrum Access
Deep Neural Network · Convolutional Neural Network
Modulation and Coding Scheme

1 Introduction

Nowadays a large number of technologies are sharing the same radio spectrum and it has become a scarce commodity [1]. However, most of the spectrum is underutilized most of the time. In order to optimize the use of the radio spectrum, Cognitive Radio (CR) systems will play an important role due to their capabilities to communicate efficiently by changing certain operating parameters of the radio, such as transmit-power, carrier frequency, and modulation and coding scheme, in real-time [2]. One of the key features in CR is the Dynamic Spectrum Access (DSA), which allows radios to dynamically access and use the unused spectrum [1]. The decision-making process executed by DSA broadly falls in the fields of cross-layer optimization and is solved by using algorithms from mathematical optimization, e.g. evolutionary algorithms, and Artificial Intelligence, e.g. Machine Learning (ML). In general, DSA involves several tasks that a radio should execute with the aim of improving its performance. One of this task is the Automatic Modulation Classification (AMC), which is used to automatically determine the modulation type of the transmitted signal. One of the main challenges in AMC is to classify a received signal into a modulation type without (or limited) a prior information about the transmitted signal under dynamic channel conditions [3].

This problem has been mainly faced via either acLB and expert Feature-Based (FB) engineering combined with pattern recognition methods [23]. Note that while Likelihood-Based (LB) methods find optimal solutions by minimizing the probability of false classification at the cost of a high computational complexity, FB methods have lower complexity and their performance is (near-)optimal, when they are designed properly. However, these features are usually chosen by an expert and are based upon a certain set of assumptions, which most of the time, are not realistic [4]. In recent years, some of the limitations of such techniques have been partially solved by applying Deep Neural Networks (DNNs). DNNs are able to (1) automatically extract the important features on raw time-series data, and (2) perform the classification task [26] on the extracted features. A DNN consists of a set of interconnected computational nodes, which are called artificial neurons, that are hierarchically organized in different layers, namely input layer, hidden/intermediate layer(s) and an output layer. A set of weighted links connecting the nodes between the layers are constantly adjusted as the training continues, where neurons take the output of the linked neurons in the previous layer and compute a new output based on the activation function of each neuron. DNNs are Neural Networks (NNs) that surpass 3 layers (including input and output layers). Note that the greater the depth of the NN, the greater the ability of it to differentiate complex features on signals for tasks such as regression, classification, and clustering.

In this paper, we will extend the previous work on AMC by using DNNs and evaluating its performance on radio signals that are not only modulated but are also coded. We call this the Automatic Modulation and Coding Classification or AMC^2. In our approach, 802.11ac radio signals are classified using a DNN according to the Modulation and Coding Scheme (MCS) that was used during their transmissions and under different Signal to Noise Ratios (SNRs) of the communication channel. The main contributions of this paper are two-fold. Firstly, a DNN architecture is defined for the radio signal classification task based on raw In-phase and Quadrature (IQ) samples of the transmitted signal. Secondly, several experiments were performed and analyzed in order to determine the impact of some DNN hyperparameters (e.g. the number of layers, batch size, etc.) on the classification accuracy. The remainder of this paper will be structured as follows:

Sect. 2 introduces the related work on AMC using DNNs. In Sect. 3, the DNN architecture used to solve the AMC^2 problem on 802.11 signals is presented. Section 4 discusses the evaluations and the results of the proposed architecture. Finally Sect. 5 contains the conclusion and future extensions of the presented work.

2 Related Works

The task of AMC is to recognize the unknown modulation scheme of a received radio signal automatically. It plays an important role in intelligent communication systems like cognitive radios. In general, this task is hard due to the absence of a prior knowledge of the incoming signal, the effects of multi-path propagation, and the dynamics and uncertainty of the channel, among others. Two traditional methods to solve this problem are the LB and FB [23]. LB methods use a likelihood function of the received signal and try to maximize the likelihood radio given a threshold. These methods find optimal solutions by minimizing the probability of false classification but at the cost of a high computational complexity. FB methods [5] use several features of the received signals, e.g. instantaneous amplitude, phase and frequency, in order to recognize them based on their observed values. These features are usually chosen by an expert and used in combination with classifiers. In comparison to LB methods, FB ones have lower complexity and their performance is (near-)optimal when they are designed properly. Among classifiers, several ML methods for Pattern Recognition (PR) have been used for AMC such as NNs [6], clustering [7], Support Vector Machines (SVM) [8], Decision Trees (DT) [9], etc.

Several works have evaluated and compared the performance of different classifiers under a given set of expert features. NN-based classifiers have shown to overcome the performance of several ML methods for solving the AMC task [11,13,14,22]. Moreover, DNNs [21], NNs with more than one layer between the input and the output layers, have shown the best performance without explicitly requiring a FB method to extract radio signal features [10,12,15,17,18,26]. DNN architectures are capable of performing both the feature extraction and the classification of the radio signals using a unique NN. Authors in [17] showed that a DNN based on a Convolutional Neural Network (CNN), which are simply neural networks that use convolution in the place of general matrix multiplication in at least one of their layers [21], was able to outperform several expert FB methods using different classifiers. The CNN achieved a rough accuracy of 87%, across all the 11 types of modulated radio signals and SNRs on the test dataset, and using only raw IQ samples of the modulated radio signals as input. Similarly, Shengliang et al. [12] proposed a CNN classifier but on images that were generated after a data conversion from raw IQ samples to grey-colored images. Again, improvements and better classification accuracy were achieved over cumulant and ML-based modulation classification algorithms.

Instead of using CNN, authors in [10] proposed a deep learning architecture based on Extensible Neural Networks (ENN) for modulation classification in multi-path fading channel. The proposed DNN is based on energy natural logarithm model. The proposed ENN uses 3 smaller DNN, each one trained to recognize the amplitude, phase, and frequency of the radio signal. Results showed that the proposed ENN has higher Probability of Correct classification than traditional algorithms for modulation classification within the same training sequence and SNR. In [15], a CNN and Long Short-Term Memory (LSTM) are combined to create a Heterogeneous Deep Model Fusion (HDMF). A deep learning-based AMC method that employs Spectral Correlation Function (SCF) is introduced in [20]. In the proposed method, a Deep Belief Network (DBN) is applied for pattern recognition and classification with high accuracy even in the presence of environmental noise. Finally, the use of Sparse Autoencoders (SAE) DNN to solve the AMC problem was evaluated in [19] using different type of inputs. Three training inputs were IQ samples, the centroids of constellation points employing a fuzzy C-means algorithm to IQ diagrams, and the high-order cumulants of received samples. Each autoencoder layer was trained using the unsupervised training followed by the softmax classifier. The results showed that the accuracy of the proposed DNN architecture using an AWGN channel model with varying SNR outperformed AMC methods using Maximum likelihood classifier, cumulant based Genetic programming in combination with KNN classifiers and feed-forward neural network using cumulants and instantaneous power spectral density (PSD) as the features.

Based on the previous works, it is clear that DNN architectures solve the modulation of radio signals with high accuracy, even on pure raw data such as IQ samples and therefore removing the need of FB methods for prepossessing the received signal. However, and to the best of our knowledge, there is not a previous work to solve the problem of automatically classifying modulated radio signals that are also coded, which we have called the AMC^2 problem, using DNN.

3 Deep Architecture for AMC^2

Similar to the AMC task, AMC^2 can be defined as follows:

Definition 1. *The automatic modulation and coding scheme classification (AMC^2) is the task of recognizing the modulation and coding scheme used to transmit a received signal without the need of an a priori knowledge of such info.*

Note that most wireless systems, e.g. IEEE 802.11 and LTE, avoid this problem by including the MCS information in each signal frame so that the receivers are notified about the change in modulation scheme, and therefore, they would be able to react accordingly. However, this strategy affects the spectrum efficiency due to the extra modulation information in each signal frame. By automatically identifying the modulation type of the received signal, the receiver does not need to be notified about the MCS, the demodulation can still be successfully achieved and the spectrum efficiency is improved. For this reason, AMC^2 becomes a fundamental part of CR systems to solve this problem.

As it was shown in Sect. 2, DNN have been proved to be a powerful tool to solve the AMC problem in wireless communication systems even using only raw IQ samples. Traditionally, DNNs are usually built in a modular manner and, most of the cases, are inspired by architectures that have performed well on other problems. In this paper we will use a CNN [16,21] architecture based on the model described in [17]. Typically, a CNN comprises of convolutional layers, each followed by optional sub-sampling/regularization (pooling) layers and ending in fully-connected layers. These are hidden layers that are effectively chained functions that collectively transform the input data to the output.

The intermediate outputs from the hidden layers are not shown and are results of linear and non-linear processing of the input data. The linear processing of the input data involves multiplication of weight matrix and addition of bias vector, while the non-linear processing of the input data is triggered by the activation function. In this paper, the DNN architectures used for the AMC^2 is composed of 3 Convolutional layers, each of them followed by 3 Dense layers. A brief description of the individual layers and their function in the architecture are discussed below. The resulting architecture and size of each layer are shown in Fig. 4. Information about the size of the input layer and the properties of the input data is presented in Sect. 4 (Fig. 1).

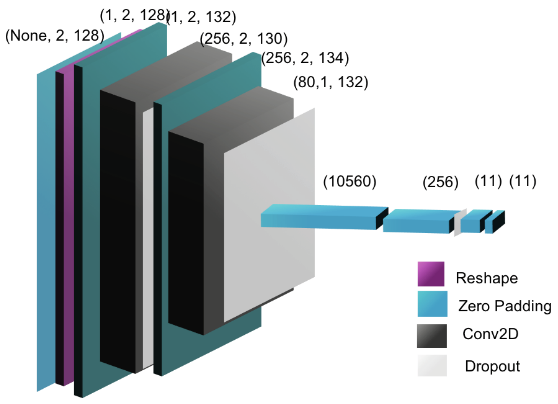


Fig. 1. 2-layered CNN accepting N $\times$ 2 $\times$ 128 inputs, where N is the batch size, and resolving to 11-class softmax output layer

3.1 Reshape and Zero-Padding Layer

A Reshape layer shapes the input data into the desired shape for the subsequent layers in the CNN. On the other hand, the Zero-Padding layer simply pads the border of the input data, to control the dimensionality of the output volumes. Together with input volume size, kernel size and stride zero-padding, this layer will ensure that the input fits nicely to the neurons in the Convolution layer.

3.2 Convolution Layer

A convolutional layer receives a 3D input vector and creates a 3D output vector that measures the filter responses at each input location, calculated as the sum of the element-wise product between filter and kernel size. This convolutional response encodes the input in terms of transformed intermediate outputs to systemically reduce input dimensionality as a part of feature learning.

3.3 Flatten and Dropout Layer

A flatten layer converts output from the previous layer from 3D to 1D vectors. This layer is necessary before Dense Layers, which accept only 1D vectors. The Dropout layer acts as a regularizing method used to avoid overfitting, i.e. the NN is able to classify objects that have not seen before. During training, neurons' weights and biases are adapted to detect specific features from the input dataset. The neuron weights would settle into their context within the network and could be too specific to the training data. When neurons are randomly dropped out of network during training, other neurons would be forced to handle representation required to predict missing neurons. This results in a neural network that would be capable to generalize new unseen data, in contrast to overfitting training data.

3.4 Dense Layer and Activation Function

A dense layer consists of neurons which have full connections to all activations in the previous layer. Their activations can be computed with a matrix multiplication. The activation function used in the Convolution and Fully-Connected/Dense Layers are Rectified Linear Unit (ReLu) and Softmax respectively. ReLu provides non-linear transformation to outputs of previous layers to learn the dataset. It is merely a two piecewise linear transformations that convert all outputs to positive values. This allows easy gradient optimization with good generalization capability. Softmax used in the last Dense Layer to represent a probability distribution over a discrete variable with n possible values, where n is the number of classes.

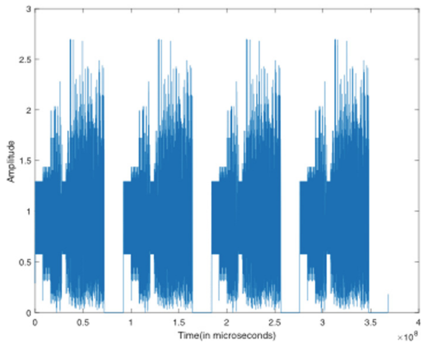
4 Experiments and Results

In this section, we will present the procedure and characteristics of the dataset generated and the main results of the evaluated architectures.

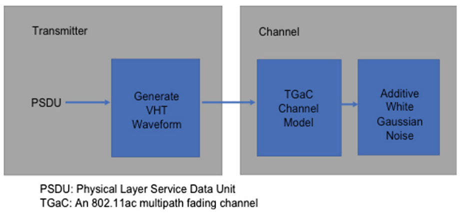
4.1 Dataset Generation

The dataset was created using the Matlab WLAN toolbox [25] and the technology selected for the radio signals with modulation and coding scheme is the IEEE 802.11ac. The radio signal was generated and represented in a 3-D vector with a shape of [N, IQ, P], where N is the number of samples of a signal with a given MCS and SNR, IQ is the real (In-phase) and imaginary (Quadrature) value representing the signal in a specific point in time, and the P is the number of consecutive points of that signal that are used as a single sample for training. For each MCS and SNR, $0 \leq SNR \leq 19$, a complex-valued time-series waveform is generated and a window with a width of 128 sample points is slid across the whole waveform 1000 times. Each IQ sample is normalized and stored separately resulting in [N, 2, 128] shape, where N is a factor (calculated by multiplication of the number of SNR types and number of MCS types) of 1000.

The simulation of the dataset begins with the creation of a Physical Service Data Unit (PSDU), which is then encoded to create a 4-packets waveform. Figure 2 shows an example of the waveform generated after transmitting 4 packets. The encoding is defined by the MCS. Only MCS 0–7 are used in the experiment. A 20-microsecond idle time is fixed between successive packets. The waveform is then passed through an evolving 802.11ac multipath fading channel (TGac). Additive White Gaussian Noise (AWGN) is added to the transmitted waveform to create the desired average SNR per sub-carrier. The 802.11ac transmitter is configured to use the very high throughput (VHT) format physical layer (PHY) packet, with a channel bandwidth of 80 Mhz, and a 1×1 MIMO array for one unique user. The Aggregate MAC Protocol Data Unit length is 1024. The transmission channel impairment is modeled to add channel and receiver noise, to produce a realistic modulated signal represented by $r(t) = s(t) {*} c + n(t)$, where $s(t)$ is time-series signal of a series of discrete bits modulated onto a sinusoid, c is path loss/gain constant on signal and $n(t)$ is additive Gaussian white noise.



(a) Generated Waveform from four packets separated by 20-microsecond idle periods)


Transmitter
Channel
PSDU
Generate VHT Waveform
TGaC Channel Model
Additive White Gaussian Noise
PSDU: Physical Layer Service Data Unit
TGaC: An 802.11ac multipath fading channel

(b) 802.11ac Signal Waveform Modeling

Fig. 2. Data set generation

After generating the data set, a preprocessing step is performed. The dataset generated are matrices of consisting IQ samples, SNR and label information for the different MCS schemes. Dataset is split into training, validation and test sets in ratio 60:20:20. The sample points for these sets are retrieved randomly from the dataset. After the splitting, it is further shuffled in place before being used in the CNN. During training, every training set is shuffled. The validation set that is used during training in order to measure the performance of the trained neural network on previously unseen data. The test data is used during prediction, where probabilities for each class is computed based on the test data's signal values. The maximum of the probabilities will be the 'winner'.

4.2 Prediction Accuracy

The confusion matrix for the proposed model, which is resulting from using the testing dataset for predictions, is presented in Fig. 3. It is possible to see that the average accuracy fluctuates among different SNR. It was found that the lowest obtained accuracy was 0.57, on average, and the highest prediction obtained for a given SNR, e.g. SNR = 19, was 0.8. It is clear that the higher the SNR, the higher the accuracy for predicting the correct MCS of the radio signal.

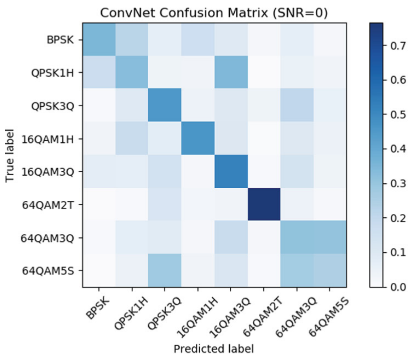
For our highest SNR case classification we show a confusion matrix in Fig. 3. At +19dB SNR, the diagonal in the confusion matrix is almost clear but some remaining discrepancies are that 64QAM with coding scheme 3/4 is misclassified as 64QAM with coding scheme 5/6. These can be explained due to the constellation 64QAM with coding scheme 5/6 is less resilient to noise and requires a higher SNR in order to be correctly identified.

4.3 Hyperparameters Results

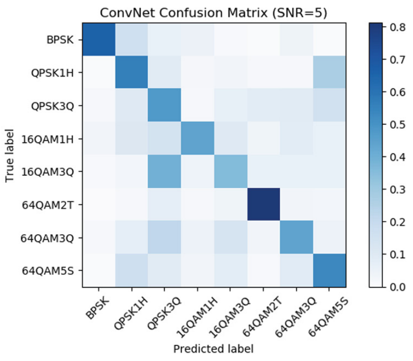
Hyperparameters can be tuned and adjusted by the developer whereas parameters are set/automatically computed by models. While parameters are influenced by the number of filters, kernel size, and bias, the model hyperparameters are defined by the developer. The effects of the hyperparameters on the prediction accuracy of the CNN can be summarized as follow:

Additional Convolutional and Dense Layers. It is observed that the rule-of-thumb of adding convolutional layers with padding and dropout does not necessarily increase the accuracy of the classifications. In fact, the higher the number of neurons, which results from additional layers, increases the training time and the space requirements to store the data. The choice of the number of filters and its size have to be modest in order not to exceed the memory limits. From different model configurations, it can be concluded that 3-layered Convolutional with Padding and Dropout is optimized for the radio signal dataset of size [N, 2, 128] with N equal to 160000 (1000×8 MCS $\times$ 20 SNRs).

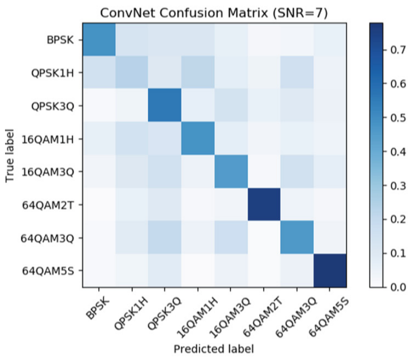
Dropout. The relationship of the dropout to the training/prediction accuracy is non-linear and varies for different CNNs architectures. For Model 1 as an example, it could learn and predict at only drop out rate of 0.0, 0.1 and 0.9, i.e. values close to 0 and 1. For 2-layered CNN, drop out rate 0.3 is the most optimal followed by 0.5 and 0.7.



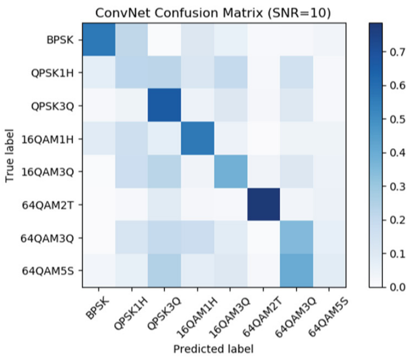
(a) SNR = 0



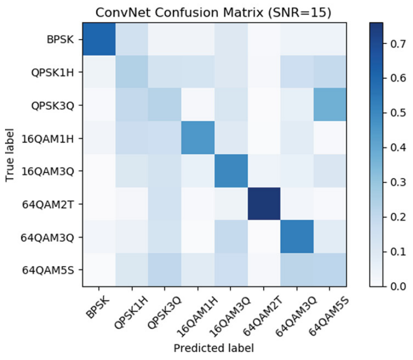
(b) SNR = 5



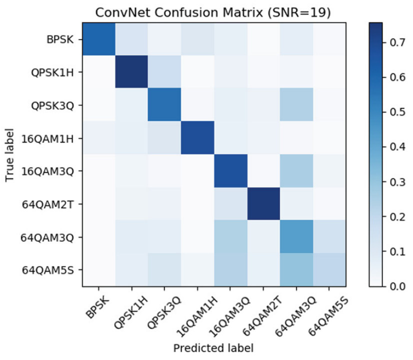
(c) SNR = 7



(d) SNR = 10



(e) SNR = 15



(f) SNR = 19

Fig. 3. Confusion matrix of predicting different MCS at different SNR

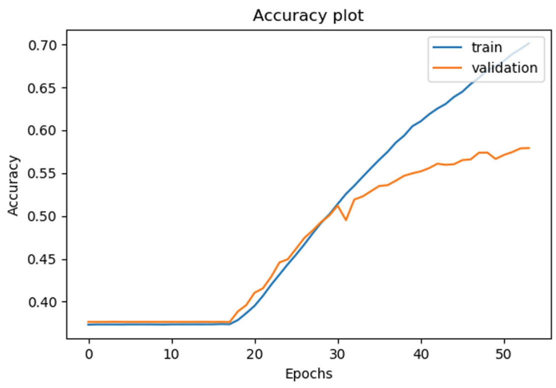


Fig. 4. Accuracy plot of the proposed architecture over the full datasetSNRs

Batch Size and No. of Epochs. Batch size of 2^5 and 2^{10} are chosen with the different models. Batch size defines how often gradient is computed and weights are updated. The effect is immediately seen in the training time per epoch. Smaller batch size will increase the training time per epoch, and it will also take a longer time to reach convergence. From the results, the effects of batch size largely depend on the number of layers of the model. For the 3-layered convolutional model, smaller batch size increases prediction accuracy, whereas, for the 2-layered convolutional model, the reverse is true. A higher number of epochs allow improving accuracy as it was expected.

Adam Optimizer. The default value in [24] is used throughout the evaluation, i.e. adam(lr = 0.001, $beta_1 = 0.9$, $beta_2 = 0.999$, epsilon = 1e−8, decay = 0) as it yields the highest training/prediction accuracy. It was detected that too high or too low learning rates could yield to homogeneous prediction and low accuracy.

5 Conclusion and Future Work

The ability of the CNN to detect different radio modulation types is the first step towards more real-life applications such as dynamic radio spectrum utilization. The paper presents some results that are intended to provide a heuristic guide in CNN to design DNN architectures for cognitive radio classification task using raw IQ data. Factors which influence training and prediction performance are identified and their impact on the CNN results are discussed. This acquired knowledge will be useful to create a new, optimized CNN specifically for radio classification tasks. As future work, we propose to evaluate the proposed DNN architectures on other waveforms, e.g. variations of the IEEE 802.11 standard or other technologies that also apply MCS to the transmitted signal, and exploring the possibility of using transfer learning on multiple radio technologies, each one with different physical layer, in order to reduce the training time.

References

1. Garhwal, A., Bhattacharya, P.P.: A survey on dynamic spectrum access techniques for cognitive radio. arXiv preprint arXiv:1201.1964 (2012)
2. Nolan, K.E., Doyle, L., O'Mahony, D., Mackenzie, P.: Modulation scheme recognition techniques for software radio on a general purpose processor platform. In: Proceedings of the First Joint IEI/IEE Symposium on Telecommunication Systems, Dublin (2001)
3. O'shea, T.J., Clancy, T.C., Ebeid, H.J.: Practical signal detection and classification in gnu radio. In: SDR Forum Technical Conference (SDR) (2007)
4. O'Shea, T., Hoydis, J.: An introduction to deep learning for the physical layer. IEEE Trans. Cogn. Commun. Netw. **3**(4), 563–575 (2017)
5. Hazza, A., Shoaib, M., Alshebeili, S.A., Fahad, A.: An overview of feature-based methods for digital modulation classification. In: 2013 1st International Conference on Communications, Signal Processing, and Their Applications (ICCSPA), pp. 1–6 (2013)
6. Azzouz, E.E., Nandi, A.K.: Modulation recognition using artificial neural networks. In: Azzouz, E.E., Nandi, A.K. (eds.) Automatic Modulation Recognition of Communication Signals, pp. 132–176. Springer, Boston (1996). https://doi.org/10.1007/978-1-4757-2469-1_5. ISBN 978-1-4757-2469-1
7. Swami, A., Sadler, B.: Modulation classification via hierarchical agglomerative cluster analysis. In: First IEEE Signal Processing Workshop on Signal Processing Advances in Wireless Communications. IEEE (1997)
8. Zhao, Z., Zhou, Y., Mei, F., Li, J.: Automatic modulation classification by support vector machines. In: Yin, F.-L., Wang, J., Guo, C. (eds.) ISNN 2004. LNCS, vol. 3173, pp. 654–659. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-28647-9_107. ISBN 978-3-540-28647-9
9. Kim, K., Polyodoros, A.: Digital modulation recognition: the BPSK versus QPSK case. In: MILCOM (1988)
10. Yang, G.Q.: Modulation classification based on extensible neural networks. Math. Prob. Eng. **2017**, 10 (2017). https://doi.org/10.1155/2017/6416019. Article ID 6416019
11. Nandi, A.K., Azzouz, E.E.: Algorithms for automatic modulation recognition of communication signals. IEEE Trans. Commun. **46**(4), 431–436 (1998)
12. Peng, S., et al.: Modulation classification based on signal constellation diagrams and deep learning. IEEE Trans. Neural Netw. Learn. Syst. **99**, 1–10 (2018)
13. Iversen, A.: The use of artificial neural networks for automatic modulation recognition, December 2003. http://www.macs.hw.ac.uk/cs/techreps/docs/files/HW-MACS-TR-0009.pdf
14. Ramakonar, V.S.: Modulation classification of digital communication signals (2002). http://ro.ecu.edu.au/theses/752
15. Zhang, D., et al.: Automatic modulation classification based on deep learning for unmanned aerial vehicles. Sensors **18**(3), 924 (2018)
16. O'Shea, K., Nash, R.: An introduction to convolutional neural networks (2015). https://arxiv.org/abs/1511.08458
17. O'Shea, T.J., Corgan, J., Clancy, T.C.: Convolutional radio modulation recognition networks. https://arxiv.org/abs/1602.04105
18. Kulin, M., et al.: End-to-end learning from spectrum data: a deep learning approach for wireless signal identification in spectrum monitoring applications. IEEE Access **6**, 18484–18501 (2018)

19. Ali, A., Yangyu, F., Liu, S.: Automatic modulation classification of digital modulation signals with stacked autoencoders. Digit. Sig. Process. **71**, 108–116 (2017)
20. Mendis, G.J., Wei, J., Madanayake, A.: Deep: learning-based automated modulation classification for cognitive radio. In: 2016 IEEE International Conference on Communication Systems (ICCS), Shenzhen, pp. 1–6 (2016). https://doi.org/10.1109/ICCS.2016.7833571
21. Goodfellow, I., Bengio, Y., Courville, A.: Deep Learning. MIT Press, Cambridge (2016). ISBN 9780262035613
22. Louis, C., Sehier, P.: Automatic modulation recognition with a hierarchical neural network. In: Proceedings of MILCOM 1994, Fort Monmouth, NJ, USA, vol. 3, pp. 713–717 (1994)
23. Dobre, O.A., Abdi, A., Bar-Ness, Y., Su, W.: Survey of automatic modulation classification techniques: classical approaches and new trends. IET Commun. **1**(2), 137–156 (2007). https://doi.org/10.1049/iet-com:20050176
24. Kingma, D.P., Ba, J.L.: Adam: a method for stochastic optimization. Published as a Conference Paper at ICLR 2015, July 2015. https://arxiv.org/pdf/1412.6980v8.pdf
25. Mathworks Documentation. WLAN System Toolbox. https://nl.mathworks.com/help/wlan/
26. Liu, X., Yang, D., El Gamal, A.: Deep neural network architectures for modulation classification. arXiv preprint arXiv:1712.00443 (2017)