# Automatic Modulation Classification Using Parallel Fusion of Convolutional Neural Networks

Erma Perenda, Student Member, IEEE, Sreeraj Rajendran, Student Member, IEEE, Sofie Pollin, Senior Member, IEEE
Department ESAT KU Leuven, Belgium, {name.surname}@esat.kuleuven.be

***Abstract*—Automatic Modulation Classification (AMC) has been the focus of research for more than three decades. Although AMC was first motivated by its applications in military scenarios such as electronic warfare, surveillance or threat analysis, nowadays it can be applied in various civilian applications, especially in the context of dynamic spectrum management. It is very challenging for an AMC classifier to achieve a good trade-off between a classification accuracy and computational cost. In the literature most attention is paid on how to achieve high classification accuracy, while computational complexity is often neglected. In this paper we propose a novel AMC framework based on Parallel Fusion of one-Dimensional (1D) Convolutional Neural Network (CNN), where the amplitude series is fed to one parallel branch, while the phase series is fed to the other parallel branch of the neural network. Through comprehensive performance analysis, we show that the proposed AMC framework achieves a classification accuracy close to $90\%$ over the range of Signal-Noise Ratio (SNR) values from $0$ dB to $18$ dB, but with a computational cost that is $2.5$ times lower than State-of-the-Art (SoA) models with the same achieved accuracy.***

***Index terms* —** Automatic Modulation Classification, Deep Neural Networks, Convolutional Neural Network

## I. Introduction

AMC presents an intermediate step between signal detection and demodulation [1]. The goal of AMC is to maximize the classification accuracy for a large number of modulation formats under different channel conditions, while keeping the computational complexity acceptable. Deep Neural Networks (DNNs) have recently attracted AMC researchers' attention due to their remarkable results in various fields such as computer vision and natural language processing [2]. Moreover, in [3], it is shown that an end-to-end communications system can be modelled using Neural Network (NN) with a performance that is competitive with respect to the SoA modular communication systems. Conventional AMC methods (i.e. Likelihood Based (LB) and Feature Based (FB)) will fail to perform AMC task as those systems have machine made modulations/constellations. In addition, a beauty of Deep Learning (DL) AMCs lies in the no needs for an expert knowledge and in fact that a high classification accuracy can be achieved for a small number of samples, what might make them feasible for a local implementation at the chipsets without need for a cloud-computing. However, SoA DL AMCs mostly neglect the computational cost what results out with very complex proposed NN models [4].

Keeping in mind both, the classification accuracy and computational cost, in this paper we propose an AMC framework which consists of two stages, the signal preprocessing stage and the signal modulation classification. As a classifier is utilized a novel proposed Parallel Fusion of 1D CNN, which reduces the computational cost versus SoA AMC based on Long-Short Term Memory (LSTM) [5] and improves the classification accuracy versus SoA AMCs based on CNN. The low computational cost is achieved by using 1D CNN, which is very effective for time-series and provides a good performance [2]. High classification accuracy is achieved by splitting amplitudes and phase series to different NN branches. Each modulation format is described by amplitude, phase or frequency variations and with learning each of them independently leads to higher accuracy. SoA CNN and other DNN models rely on two channel input data [5]–[7], one channel for amplitude series and other for phase series. Having in mind how the convolution is performed in CNN [2], it is obvious that the values from different channels will be mixed together at deeper layers what will require more complex NN models to learn non-linear mixture of amplitude and phase series.

The remainder of the paper is organized as follows. The overview of SoA AMC methods is presented in II. In Section III the signal model and classifier performance metrics are discussed. Section IV explains the structure of novel AMC framework. The section V introduces the relevant SoA models for AMC that are then discussed next. The conclusions are briefly presented in Section VI.

## II. Related Work

Generally, AMC approaches can be broadly classified to LB, FB and DL.

*1) LB methods:* They are optimal in Bayesian sense, in that they minimize the chance of a wrong classification. LB methods use exact or approximate Likelihood Ratio Tests (LRTs) as a decision criterion. LB methods require a careful design and selection of signal and noise models. The traditional LB methods assume that all channel parameters are known. LB methods are characterized by high computational complexity and are sensitive to noise and interference. A dynamic wireless fading channel introduces a lot of nonlinear imperfections which are hard to describe by maths. Moreover, it is very difficult to obtain all necessary system parameters at receiver side. Having said that, LB AMCs deployed in practical environment might not operate well. A detailed survey about LB methods is given in [8,9].

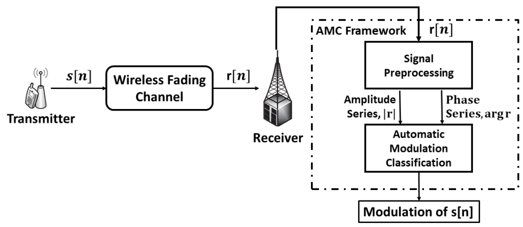



Figure 1. AMC Framework.

*2) FB methods:* They rely on an assumption that any given feature may be similar between different modulations, but no two modulations are similar in all features. FB methods [10]–[13] are sub-optimal in Bayesian sense, but they are popular due to their easy implementation. They may have high computational complexity depending on how many and which type of signal features are used. FB methods require offline phase for feature extraction and substantial system design-side knowledge about modulation properties. FB AMCs require a huge number of samples (theoretically infinite) to ensure required high classification accuracy. However, huge number of samples leads to higher processing delay and for local chipset implementation of AMCs it might be infeasible.

*3) DL methods:* DL relies on Stochastic Gradient Descent (SGD) technique to optimize NN models. Due to their popularity and high performance in various applications, DNNs have been widely considered for AMC task [14]. Multi-Layer Perceptron Network (MLPN), CNN and Recurrent Neural Network (RNN), i.e. LSTM have been investigated for AMC task. Their convergence time depends on the number of considered modulations and the type of environment. CNNs are faster by design, since the computations in CNNs can happen in parallel, while RNNs need to be processed sequentially, since the subsequent steps depend on previous ones. DL methods don't have feature selection phase, since the efficient classification features are selected directly from raw In-phase/Quadrature (I/Q) components of signal [4]–[7]. Thus, there is no need for an expert knowledge, such as it in FB AMCs. Moreover, DL AMCs can achieve high classification accuracy for small number of samples [5,6]. There are also a few proposals which combine two NNs connected in serial or parallel in order to achieve better performance of classifier such is [4] where is shown that serial fusion of LSTM and CNN is better than parallel.

## III. Signal Model and Performance Measure

In this section, we introduced the model of modulated signal as received by the classifier.

Assume that there is one active transmitter which transmits signal and one antenna is utilized at the receiving side. Let $s(n)$ and $r(n)$ denote the transmitted signal and received signal at the $n$-th time slot, respectively (Fig. 1). Their relationship is given by

$$r[n] = \alpha[n] \exp^{j(\omega_0 n + \theta_0)} s[n] + v[n] \tag{1}$$

where $\alpha[n]$ is the complex channel gain, $\omega_0$ is the frequency offset, $\theta_0$ is the phase offset between the transmitter and the receiver, and $v[n]$ is the complex Additive White Gaussian Noise (AWGN) with mean 0 and variance $2\sigma_v^2$.

A task of a classifier is to correctly choose a modulation format of received signal, $r[n]$ from a pool of known $N_{mod}$ candidate modulations. The performance of classifier is determined by a basic measure $P_{cc}$, named as average probability of correct classifications. Let $P_c^{(i\prime|i)}$ denote the classification probability to declare that the $i\prime$-th modulation format has been recognized, when the modulation format of the incoming signal is $i$. Under an assumption that each modulation format has the same probability to be sent, then the average probability of correct classification is given as

$$P_{cc} = \frac{1}{N_{mod}} \sum_{i=1}^{N_{mod}} P_c^{(i\prime|i)} \tag{2}$$

## IV. AMC Framework

In this section, the structure of the AMC framework is introduced. The right-hand side of Fig. 1 illustrates the proposed AMC framework, which consists of two stages, the signal preprocessing stage and the classification stage based on the 1D CNN Parallel Fusion network model.

### A. Signal Preprocessing Stage

In the signal preprocessing stage the rectangular form of received signal is transformed into polar form, i.e. I/Q values are transformed to their corresponding amplitude and phase values. The amplitude and phase components of received signal are separated into two different series. The series of $L_s$ amplitude samples is fed to the first branch of the Parallel Fusion Network, while the series of $L_s$ phase samples is fed to the other branch of the Parallel Fusion Network. If the length of the amplitude/phase series is lower than the required number of samples, $L_s$ the both series are padded with zeros to fill the required length.

### B. Classification Stage using 1D CNN Parallel Fusion network model

As an AMC is proposed 1D CNN Parallel Fusion network which is given in Fig. 2. Each branch has $n_p$ Conv 1D layers with 64 filters with kernel size of 3. To prevent the network from overfitting after each Conv 1D layer, there is a Dropout layer with drop probability of 0.2. Parallel branches are merged by a $Concat$ Merge layer. The output concatenated series of data passes through $n_s$ Conv 1D layers, each with 64 filters and a kernel size of 3. Each serial Conv 1D layers is followed by Max Pooling layer with kernel size of 2 and with Dropout Layer with drop probability of 0.2. The Max Pooling Layer reduces the complexity of the output and prevents overfitting of the data. The data passes further through two Fully Connected Layers (FCLs) with 128 units and Scaled exponential Linear units (SeLu) activation. The dropout layer with keeping probability of 0.5 follows each FCL, as well. Finally, the data are fed to the last FCL with number of the

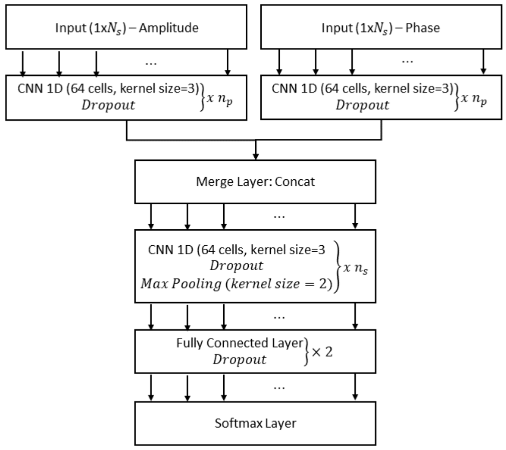



Figure 2. 1D CNN Parallel Fusion Network.

units equal to number of considered modulations and Softmax activation. Model classification accuracies are analysed for different values of $n_p$ and $n_s$ in order to find their optimal values which assures the highest accuracy under a reasonable complexity.

## V. AMC Performance Analysis

The performance and the sensitivity analysis of different system parameters is carried out for the proposed 1D CNN Parallel Fusion network and some SoA single-signal AMC models, what is presented in this section. For our performance analysis, we are mainly interested in classificaiton accuracy and processing cost.

### *A. RadioML dataset*

The single-signal AMC comparison analysis is done by utilizing a baseline modulation dataset, described in [15]. The synthetic dataset is built in GNU radio, by simulating the 11 commercially used modulation formats at varying SNR ratios (-20 dB to 18 dB). The channel is modelled as dynamic and includes a number of imperfections such as center frequency offset, sample rate offset, AWGN, multi-path and fading. Each signal example is normalized and consists $L_s = 128$ complex floating point time I/Q samples.

### *B. SoA Models*

For comparison we utilize the newest SoA FB and DL AMC models. LB AMC models are left out due to their high computational complexity. Among the SoA DL AMC we chose the following DNN models:

- LSTM model with 2 hidden layers as it is given in [5];
- two-Dimensional (2D) CNN model with two 2D convolutional layers and two dense FCLs as it is described in [7];
- 1D CNN model with seven 1D convolutional layers, seven Max Pooling layers and three dense FCLs, such it is given in [6]. Since the input sample length is 128, the first three Max Pooling layers are removed from the model;
- Complex-valued 1D CNN model has the same number and type of hidden layers as model in [7], but the activations are complex-valued and convolutional layers are 1D.

As FB AMCs we consider the following shallow classifiers which are widely used in SoA models:

- Decison Tree Classification (DTC) with the maximal depth equal to number of features;
- Support Vector Machines (SVM) with a radial kernel and a penalty parameter equal to 10;
- K-Nearest Neighbour (KNN) with number of neighbours equal to number of input features.

The features taken into account belong to instantaneous time domain, transform domain, statistical and zero-crossing features. Their definitions and mathematical descriptions can be found in [16]. 30 features given in [16] are calculated for a sample length of 128 I/Q values. The parameters of FB classifiers are set according to the ones that are common in SoA models.

### *C. Implementation Details*

All single-signal AMC models considered in this paper are implemented in TFlearn, a Python's deep learning library built on top of Tensorflow [17]. The training and testing stages for each of them are run on Dell laptop with the following processor Intel i7-8650U CPU @ 1.90GHz 2.11 GHz. The training stage is done through 80 epochs, with the batch size (number of samples per gradient update) of 256. As an optimizer is set Adam (Adaptive Moment Estimation)[18], with a learning rate set to 0.001, while the metric is Categorical cross-entropy.

### *D. Results*

*1) AMC performance by model:* The classification accuracies for the SoA models and our proposed 1D CNN Parallel Fusion network model are summarized in Fig. 3. The highest accuracy is achieved by the LSTM model, while our proposed 1D CNN Parallel Fusion network achieves the same accuracy at high SNR values and a slightly decreased (less than $8\%$) performance at low SNR. A KNN classifier with 30 neighbours has the worst accuracy. The 1D CNN model outperforms the 2D CNN model. The complex-valued 1D CNN model achieves the highest classification accuracy at low SNR values, but due to its high complexity it is not preferable choice. Fig. 3 shows that DNN models outperform the shallow classifiers for a whole range of SNR values. Fig. 4 shows that both LSTM and 1D CNN Parallel Fusion models have a problem to distinguish Wide-Band Frequency Modulation (WBFM) and Amplitude Modulation - Dual-Side Band (AM-DSB), what is a consequence of the way how the AM-DSB dataset is built (there are silence periods of audio signals). A 1D CNN Parallel Fusion model can distinguish higher orders of Quadrature Amplitude Modulation (QAM) modulation with high accuracy of $76-85\%$ at 0 dB SNR. The SoA models

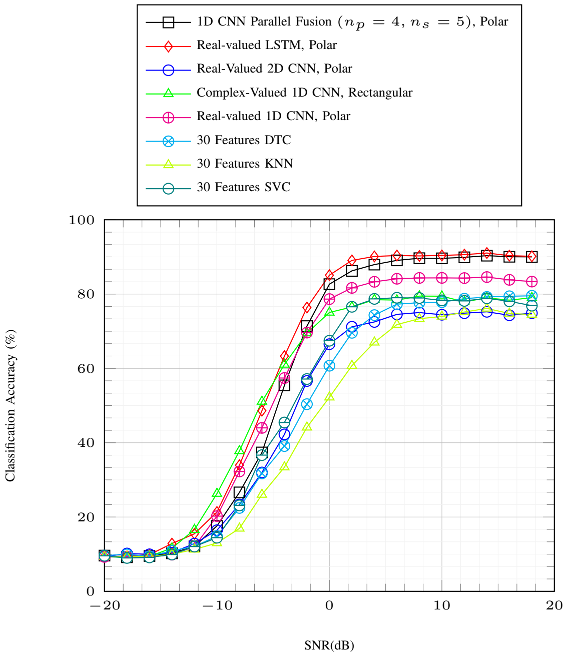



Figure 3. AMC Classification accuracy with input signal in time domain.

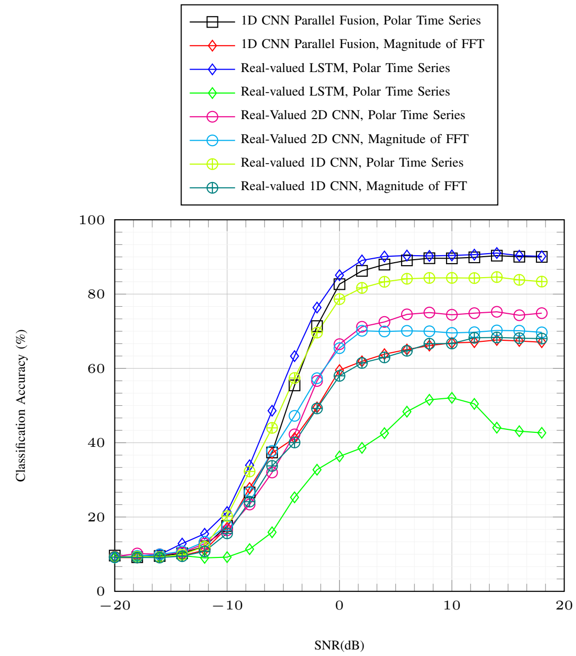
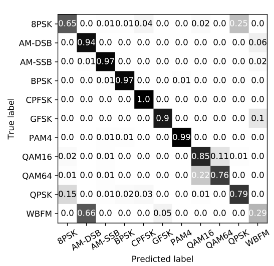
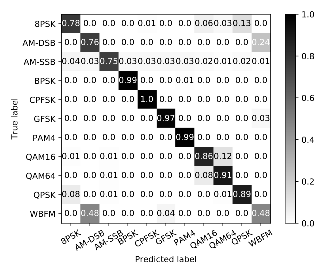


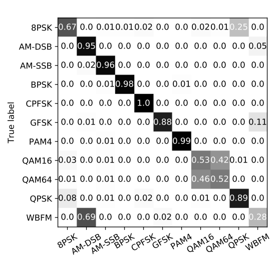

Figure 5. Single-signal AMC performance in time and frequency domain.



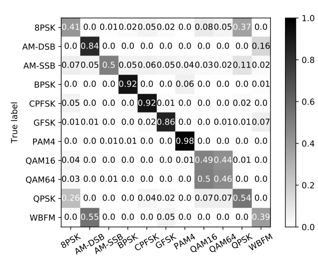
(a) 1D CNN Parallel Fusion



(b) LSTM



(c) 1D CNN



(d) 2D CNN

Figure 4. Confusion matrix at 0 dB of SNR.

of CNN have this accuracy up to 55%. Therefore, by splitting amplitude series and phase series to two CNN branches we achieved 30% increase of classification accuracy for QAM family modulations.

One more interesting thing to compare is the performance of time or frequency domain series. For each model we compare the same set of classifiers for input sequences that present the magnitude of Fast-Fourier Transformation (FFT). Fig. 5 shows that the LSTM performance degrades much when the input signal is the magnitude of FFT. On the other hand, both 1D and 2D CNN, have a lower performance degradation and perform better than LSTM. However, in each case the time series input data provides higher classification accuracy. Thus, AMCs based on spectograms have a bad performance, since the modulation formats within the same family share the same shape of spectrum.

*2) FB AMC performance by number of features:* In order to get insight in the sensitivity of FB AMCs to the feature selection, we explored their performance and cost as a function of feature selection. First, we calculated a correlation coefficient of each feature given in [16] with the modulation output label. Based on obtained correlation coefficients values, we picked 11 features with the highest absolute value and examined the performance of FB AMCs with such reduced number of features. According to the Fig. 6 we can state that FB AMCs are high sensitive on number and type of features that are fed to their inputs. Support-Vector Classifier (SVC) achieves the highest classification accuracy versus DTC and KNN, but it is the most complex. The results shown in Fig. 6 are expected, since the absolute value of correlation factor of each of 30 features with output modulation label is below 0.21, what means that whichever feature is left out it will have an impact on the classification accuracy.

*3) 1D CNN Parallel Fusion AMC performance by depth:* In order to obtain the optimal values for $n_p$ and $n_s$ parameters of 1D CNN Parallel Fusion network we experimented with different combinations. Fig. 7 shows that the classification accuracies are slightly different, but the highest accuracy is obtained for $n_p = 4$ and $n_s = 5$. However, LSTM model has the highest accuracy, but it has the highest computational cost. By 1D CNN Parallel Fusion network we achieved the same accuracy for high SNR values and a slight lower accuracy

Table I
PROCESSING TIME PER SAMPLE [MS]

| Model | 1D CNN Parallel Fusion | | | | | LSTM | SoA 1D CNN |
|---|---|---|---|---|---|---|---|
| | $n_p = 2, n_s = 4$ | $n_p = 3, n_s = 4$ | $n_p = 2, n_s = 5$ | $n_p = 4, n_s = 5$ | $n_p = 5, n_s = 6$ | | |
| Time | 0.6 | 1.0 | 0.87 | 1.0 | 1.5 | 2.5 | 0.5 |

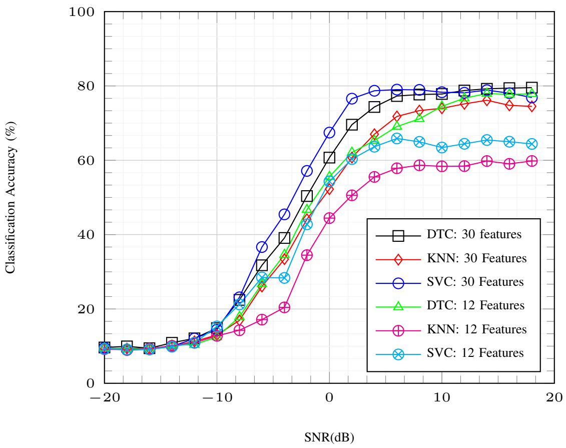



Figure 6. FB AMCs performance by number of features.

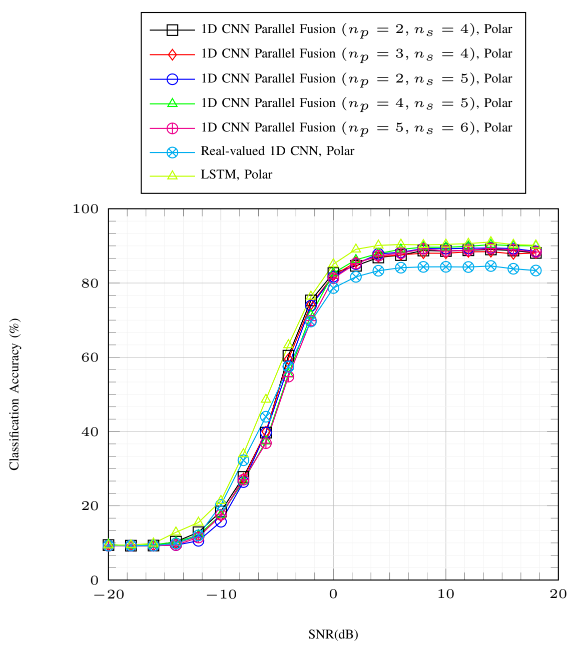



Figure 7. 1D CNN Parallel Fusion Classification Accuracy sensitivity on network depth.

(less than 8%) at low SNR values, but with the significant lower computational cost. Table I shows that for optimal 1D CNN Parallel Fusion network the classification/prediction time consumption is 2.5 times lower than time needed by LSTM.

## VI. CONCLUSIONS

In this paper we propose a Parallel Fusion AMC framework that aims to achieve good feature extraction from raw complex signals, while at the same time minimize computational cost. Through a detailed performance analysis and comparison with various AMC approaches from literature, we show that our method achieves SoA classification performance at high SNR while reducing the computational cost with at least a factor 2. In addition, memory requirements are reduced significantly.

## REFERENCES


[1] J. L. Xu, W. Su, , and M. C. Zhou, "Software-Defined Radio Equipped With Rapid Modulation Recognition," *IEEE Transactions on Vehicular Technology*, vol. 59, pp. 1659–1667, May 2010.

[2] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016.

[3] T. O'Shea and J. Hoydis, "An introduction to deep learning for the physical layer," *IEEE Transactions on Cognitive Communications and Networking*, vol. 3, pp. 563–575, Dec 2017.

[4] D. Zhang, W. Ding, B. Zhang, C. L. C. Xie, J. Han, and H. Li, "Heterogeneous Deep Model Fusion for Automatic Modulation Classification," *Preprints*, 2018. 2018010097 (doi: 10.20944/preprints201801.0097.v1).

[5] 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.

[6] T. O'Shea, T. Roy, and T. C. Clancy, "Over the Air Deep Learning Based Radio Signal Classification," *IEEE Journal of Selected Topics in Signal Processing*, vol. 12, no. 1, 2018.

[7] T. J. O'Shea and J. Corgan, "Convolutional radio modulation recognition network," *International Conference on Engineering Applications of Neural Networks*, pp. 213–226, 2016.

[8] J. L. Xu, W. Su, and M. Zhou, "Likelihood-Ratio Approaches to Automatic Modulation Classification," *IEEE Trans. Systems, Man, and Cybernetics Part C: Applications and Reviews*, vol. 41, pp. 455–469, July 2011.

[9] O. A. Dobre, A. Abdi, Y. Bar-Ness, and W. Su, "Survey of automatic modulation classification techniques: classical approaches and new trends," *IET Communications*, vol. 1, pp. 137–159, April 2007.

[10] A. Ebrahimzadeh and S. A. Seyedin, "Automatic Digital Modulation Identification in Dispersive Channels," *Proceedings of the 5th WSEAS International Conference on Telecommunications and Informatics, Istanbul, Turkey*, pp. 409–414, May 2006.

[11] A. G. Ahmed, "Automatic Modulation Recognition Based On Artificial Neural Networks Over Wireless Fading Channel In Cognitive Radio," *nternational Journal of Advanced Computational Engineering and Networking*, vol. 5, no. 10, 2017.

[12] S. Majhi, R. Gupta, W. Xiang, and S. Glisic, "Hierarchical Hypothesis and Feature-Based Blind Modulation Classification for Linearly Modulated Signal," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 12, 2017.

[13] J. Li, Q. Meng, G. Zhang, Y. Sun, L. Qiu, and W. Ma, "Automatic Modulation Classification Using Support Vector Machines and Error Correcting Output Codes," *2017 IEEE 2nd Information Technology, Networking, Electronic and Automation Control Conference*, 2017.

[14] N. E. West and T. J. O'Shea, "Deep Architectures for Modulation Recognition," *IEEE International Symposium on Dynamic Spectrum Access Networks*, 2017.

[15] T. J. O'Shea and N. West, "Radio machine learning dataset generation with gnu radio," *Proceedings of the GNU Radio Conference*, vol. 1, no. 1, 2016.

[16] iNETS featurized RF signal dataset download, January 2019. [Online]: Availlable: https://www.inets.rwth-aachen.de/registration.html.

[17] T. Documentation, March 2019. [Online]. Available: http://tflearn.org/.

[18] D. P. Kingma and J. L. Ba, "Adam: A Method for Stochastic Optimization," *San Diego: The International Conference on Learning Representations (ICLR)*, vol. 1, no. 1, 2015.