JCSE Journal of Computing Science and Engineering

Regular Paper

*Journal of Computing Science and Engineering,
Vol. 16, No. 1, March 2022, pp. 25-42*

Automatic Modulation Recognition Using Minimum-Phase Reconstruction Coefficients and Feed-Forward Neural Network

Sunday Ajala
Department of Engineering, Norfolk State University, Norfolk, VA, USA
s.a.ajala@spartans.nsu.edu

Emmanuel Adetiba*
Department of Electrical & Information Engineering, Covenant Applied Informatics & Communication Africa Center of Excellence (CApIC-ACE), Covenant University, Ota, Nigeria
Institute for Systems Science, HRA, Durban University of Technology, Durban, South Africa
emmanuel.adetiba@covenantuniversity.edu.ng

Oluwaseun T. Ajayi
Department of Electrical and Computer Engineering, Illinois Institute of Technology, Chicago, IL, USA
oajayi6@hawk.iit.edu

Abdultaofeek Abayomi
Department of Information and Communication Technology, Mangosuthu University of Technology, Umlazi, Durban, South Africa
abayomi.abdultaofeek@mut.ac.za

Anabi Hilary Kelechi and Joke A. Badejo
Department of Electrical & Information Engineering, Covenant Applied Informatics & Communication Africa Center of Excellence (CApIC-ACE), Covenant University, Ota, Nigeria
hilary.anabi@covenantuniversity.edu.ng, joke.badejo@covenantuniversity.edu.ng

Sibusiso Moyo
Office of DVC Research, Innovation & Engagement, Institute for Systems Science, Durban University of Technology, Durban, South Africa
moyos@dut.ac.za

Murimo Bethel Mutanga
Department of Information and Communication Technology, Mangosuthu University of Technology, Umlazi, Durban, South Africa
mutangamb@mut.ac.za

Open Access **http://dx.doi.org/10.5626/JCSE.2022.16.1.25** **http://jcse.kiise.org**

Received 27 July 2021; **Accepted** 07 February 2022

*Corresponding Author

pISSN: 1976-4677 eISSN: 2093-8020

Abstract

Identification of signal waveforms is highly critical in 5G communications and other state-of-the-art radio technologies such as cognitive radios. For instance, to achieve efficient demodulation and spectrum sensing, cognitive radios need to implement automatic modulation recognition (AMR) of detected signals. Although many works have been reported in the literature on the subject, most of them have mainly focused on the additive white Gaussian noise (AWGN) channel. However, addressing the AWGN channel, only, does not sufficiently emulate real-time wireless communications. In this paper, we created datasets of six modulation schemes in GNU Radio. Wireless signal impairment issues such as center frequency offset, sample rate offset, AWGN, and multipath fading effects were applied for the dataset creation. Afterward, we developed AMR models by training different artificial neural network (ANN) architectures using real cepstrum coefficients (RCC), and minimum-phase reconstruction coefficients (MPRC) extracted from the created signals. Between these two features, MPRC features have the best performance, and the ANN architecture with Levenberg-Marquardt learning algorithm, as well as logsig and purelin activation functions in the hidden and output layers, respectively, gave the best performance of 98.7% accuracy, 100% sensitivity, and 99.33% specificity when compared with other algorithms. This model can be leveraged in cognitive radio for spectrum sensing and automatic selection of signal demodulators.

Category: Network and Communications

Keywords: Cognitive radio; Cepstrum analysis; GNU Radio; Modulation schemes; MPRC; RCC

I. INTRODUCTION

Current research in the automatic modulation recognition (AMR) field began with the military due to the need to securely transmit and receive friendly signals and at the same time detect, track, and jam hostile signals. The AMR has recently been applied to civilian applications by exploiting recent developments in cognitive radio systems [1, 2]. To effectively transmit information from a transmitter to a receiver, the receiver must recognize the modulation of the radio signal so that it can effectively demodulate it, thus making the transmission possible. The conventional approach which involves an inclusion of the modulation information in each signal frame suffers a significant drawback on spectral efficiency. Therefore, the spectral inefficiency problem has contributed to the recent shift towards AMR, as it is more bandwidth-efficient because the transmitted signal element does not need to carry information about the types and order of its modulation. The 1 ms end-to-end (E2E) latency of ultra-reliable low latency communication technology amongst other technologies driven by 5G wireless communications will be easily achieved if AMR is deployed [3]. While the AMR algorithms help to control the bandwidth, the receiver requires more computational power during signal processing. Nevertheless, the corresponding increase in computing power and microprocessor size in recent times has substantially addressed this problem.

In general, some literatures classify AMR algorithms into two broad categories, namely likelihood-based (LB) and feature-based (FB) [4, 5]. Although the LB methods work by measuring the probability of a received signal belonging to candidate constellations and selecting the signal with the highest probability, the FB methods work by extracting characteristics from high-order statistics and cyclostationary characteristics. Artificial neural network (ANN) is a commonly used FB machine learning (ML) approach for AMR [4, 6, 7]. ANN has the unique characteristic of an information or data processing system, which is not programmed but can autonomously adapt itself to changing conditions of the information environment. This is usually achieved by performing a nonlinear mapping of the outputs to a set of inputs. Like the human neurological system, they consist primarily of neurons which are network nodes consisting of numerous nonlinear functions, links between the neurons which are weighted and can be adjusted automatically using a training algorithm. As an ML technique, the reason for ANN dominance includes its advantages such as generalization, parallel processing, centralized memory, redundancy, and learning. Several authors have leveraged these attributes for the field of AMR [4-7].

The authors in [8] used a fourth-order cumulant-based approach on multiuser AMR. The shortcomings of their work lie in assumptions such as a specific number of transmitters, transmitting at the same transmission power, and their choice of an additive white Gaussian noise (AWGN) channel. Their method was further limited to the identification of four modulation schemes—binary phase shift keying (BPSK), quadrature phase shift keying (QPSK), 4-level pulse amplitude modulation (4PAM), and 16-state quadrature amplitude modulation (16QAM)—likely present in the frequency band.

As identified above, in [9], the authors addressed the shortcomings of [8] by developing a multiuser AMR algorithm based on fourth-order cumulants for real-multipath fading environments and without the assumption of a specific transmit power. Their algorithm was also effective in determining the precise modulation type. With the advantages of diversity and recent developments in

MIMO schemes, research on multiuser AMR recognition schemes will continue to increase.

It has been established that the signal cepstrum has rich characteristics and can be used to estimate signal waveform parameters that can be utilized to immediately identify and classify signals based on their modulation schemes [10]. The authors extracted the cepstrum of signals with orthogonal frequency-division multiplexing (OFDM) and utilized these distinct features to detect and classify the signals using the Neyman-Pearson detection strategy. The modeled AMR system is robust to noise and outperforms the energy detector instances of low noise uncertainty. However, the fact that it was limited to only one modulation scheme, OFDM, stood out as a drawback.

The authors in [11] presented one of the earliest applications of ANN for recognition of both analog and digital modulation schemes, which included amplitude modulation (AM), frequency modulation (FM), amplitude shift-keying (ASK), QPSK, BPSK, frequency shift-keying 1 (FSK1), frequency shift-keying 2 (FSK2), and continuous wave (CW). They selected a collection of 21 features obtained by finding the standard deviation, kurtosis, and skewness of signal characteristics such as envelope amplitude A, and instantaneous frequency F, and their combinations (i.e., dot products A.F, F.A, A.A, A.A', etc.). Each feature was selected either because it had been successfully used in the past or because statistical modeling suggested that they are good AMR feature descriptors. The 21 characteristics were grouped to create a single 21-dimensional vector of features for each signal. A multilayer perceptron (MLP) ANN architecture with one hidden layer was trained using the backpropagation learning algorithm to classify the extracted feature vectors according to their modulation types. Their results showed that for over 80% of the time, their MLP classifier was able to identify the AM, FM, ASK, QPSK, and BPSK modulation schemes, while it is found difficult to classify the FSK1, FSK2, and CW modulation schemes.

A different approach in [12, 13] proposed a method for AMR by extracting cepstral features referred to as mel-frequency cepstral coefficients (MFCCs) from signals and their transforms from which the modulation type and order of digital modulation schemes such as PSK, MSK, FSK, and QAM were determined. From the results obtained by the authors, it is evident that some digital modulation scheme detection was effective with the cepstral feature extraction approach. However, the fact that the simulation was limited to the (AWGN) channel stood out as a drawback, which could hinder real-world implementation. This is because some other severe channel impairments like center frequency offset (CFO), sample rate offset (SRO), AWGN, multipath, and selective-fading were not considered in their proposed AMR model.

We aim to address some of the previous limitations, where authors were restricting channel impairments to AWGN in their models, leaving out some other severe impairments like CFO, SRO, multipath, and selective fading, which could hinder real-world implementation. In addition, existing works such as [8-13] limited their work to certain modulation types and order, thus presenting a limited view of the problem in the literature. We experimented with six digital modulation schemes, namely BPSK, QPSK, 8PSK, 16QAM, 64QAM, and Gaussian minimum shift keying (GMSK). The real cepstrum coefficients (RCC) and the minimum-phase reconstruction coefficients (MPRC) [14-16] obtained from signals modulated with these six schemes culminated in the feature vectors used to train a single-layered feed-forward neural networks and fine-tuned to produce 18 different AMR models. Thus, the primary contribution of our paper is the MPRCs extracted from signals modulated with BPSK, QPSK, 8PSK, 16QAM, 64QAM, and GMSK signals as an alternative descriptor for modulation recognition using ANN. The MPRC descriptor improved the existing performance of the shallow classifier by enhancing the classification accuracy with reduced computational complexity and memory space requirement needed for efficient AMR systems.

The rest of the paper is organized as follows. Section II is a presentation of the materials and methods. Section III follows this, where we present the results. The discussion is presented in Section IV, while the conclusion of the work is in Section V.

II. MATERIALS AND METHODS

A. Problem Formulation

AMR is an intermediate step between detecting and demodulating a signal [6]. The generic form of a modulated signal $s(t)$ received is given as [4]:

$$s(t) = Re\{\alpha e^{i2\pi\beta} e^{i2\pi\Delta ft} C(t) e^{i2\pi\Delta f_c(t-t_0)}\} + n(t) \qquad (1)$$

where $C(t)$ is the complex envelope of the modulated signal, $n(t)$ is a band-limited noise, f_c is the carrier frequency, α is the channel amplitude, β is the phase offset, Δf_c is the carrier frequency offset, and $Re\{.\}$ denotes the real part. It is possible to categorize the most widely used AMR features into five classes: instantaneous time domain, transform domain, statistical, constellation shape, and zero-crossing features. In this paper, we used transform domain features obtained by converting signals into Fourier domains with various operations for pre-processing and post-processing. Some of these processes include smoothing, normalization, median filtering, and cepstrum analysis. This current study obtained N-dimensional feature vectors (RCCs and MPRCs) from modulated signals through cepstrum analysis.

Unique classification of the transform domain features of each modulation scheme is a supervised learning problem [12, 13]. In supervised learning, the training dataset is $\{(x_k, c_k)\}$ with $k \varepsilon \{1, ..., N\}$, where each x_k is a training sample, N is the total number of samples in the dataset, and the corresponding set of class labels is $c_k \varepsilon \{1, ..., m\}$, where m is the number of classes in the dataset. Therefore, the supervised learning task involves the development of a model with a set of N samples as inputs, and a developed model is used to predict a class label during training (or unknown samples during testing) as the outputs. Thus, in this study, x_k represents the RCCs and MPRCs obtained from created samples of each of the six modulation schemes, where the total sample size $N = 300$ and $m = 6$ classes (i.e., 16QAM, 64QAM, BPSK, QPSK, 8PSK, and GMSK).

ANN is a supervised machine learning technique that can closely approximate linear and nonlinear functions [14, 15]. However, the mapping function represented by a network is not perfect because of the local minima problem, suboptimal network architecture, and finite sample data in neural network training. Typically, the network is designed to minimize a loss function, such as the mean squared error (MSE). Based on the popular statistical least squares estimation theory [14], the mapping function $F : x \rightarrow y$ that minimizes the expected squared error:

$$E[y-F(x)]^2 \tag{2}$$

is the conditional expectation of y given x

$$F(x) = E[y|x] \tag{3}$$

Thus, the least-squares calculated in a classification problem for the mapping function is precisely the posterior probability. Theoretically, to get a reasonable estimate, it can require an extensive network as well as large sample data. For instance, Funahashi [17] reveals that neural networks with at least 2D hidden nodes can estimate the posterior probability with arbitrary accuracy for the two-group d-dimensional Gaussian classification problem when unlimited data is available, and the training continues perfectly. Experimentally, the number of samples is crucial in learning [18, 19].

The output objective of neural networks are the least square the approximation of the Bayesian posteriori probability, of the cost function minimization [19, 20]. This is true for other forms of cost or error functions, such as the cross-entropy function. However, in [21], it was proved that neural networks trained with a generalized MSE cost function could satisfy the optimal Bayes rule and demonstrate that the relationship between neural networks and the optimal Bayes rule is both in theory and experiments a statistical decision problem.

For modulation recognition with ANN, the MSE, which is the loss function (cost function), is often engaged during supervised training to evaluate the difference between the expected and actual predictions [22]. Given N as the number of the training dataset, y is a vector of true labels (y = [target (x_1), target (x_2)…target (x_n)]), and $\hat{y}$ is a vector of the predictions, the loss function L is represented as Eq. (4) and there is a need to find the slope to the loss (or cost) function in order to minimize it.

$$L = \frac{1}{N}\Sigma_{k=1}^{N}(y_k-\hat{y}_k)^2 \tag{4}$$

Fundamentally, in AMR workflow, after signal pre-processing, discriminatory features are extracted as descriptors of the signal modulation, after which the AMR problem is reduced to a pattern classification problem [11, 22].

B. Simulation and Dataset Creation

GNU Radio has many in-built tools to create a reliable and robust dataset in the software-defined and cognitive radio domain of wireless communication. It includes libraries for modulators, signal sources, data visualization, data sinks, encoders, demodulators, and a rich collection of channel simulation modules for the application of simulating channel propagation models to synthetic and real-world signals [11, 23]. As a high-level end-user applications software, the modules in GNU Radio were logically linked together in a flow graph as shown in Fig. 1 to create our dataset for this study.

We selected a random source of GNU Radio as the data source for signals. The BPSK, QPSK, and 8PSK are implemented using the PSK Mod blocks as a transmitter model to achieve each modulation scheme at a sample rate of 32,000 samples per second as recommended by the Audio Engineering Society for transmission-related applications [24]. Also, the sampling rate is at least twice the highest frequency of the signal, consistent with the Nyquist theorem. For 16QAM and 64QAM, QAM Mod blocks were used, and for the GMSK modulation scheme, the GMSK Mod block, as illustrated in Fig. 2, was used to achieve the desired modulation. Several compact stochastic models were used to simulate different propagation effects in modeling a wireless channel [23, 25, 26].

Furthermore, many more real-life propagation effects can be synthetically modeled. Thus, in order to realize the core objective of this study, we used the GNU Radio Dynamic Channel block to simulate some desired effects such as random processes for CFO, SRO, AWGN, multipath propagation, and selective fading in the creation of our dataset. The dynamic channel also implements the random

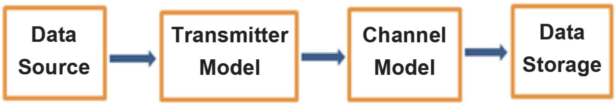
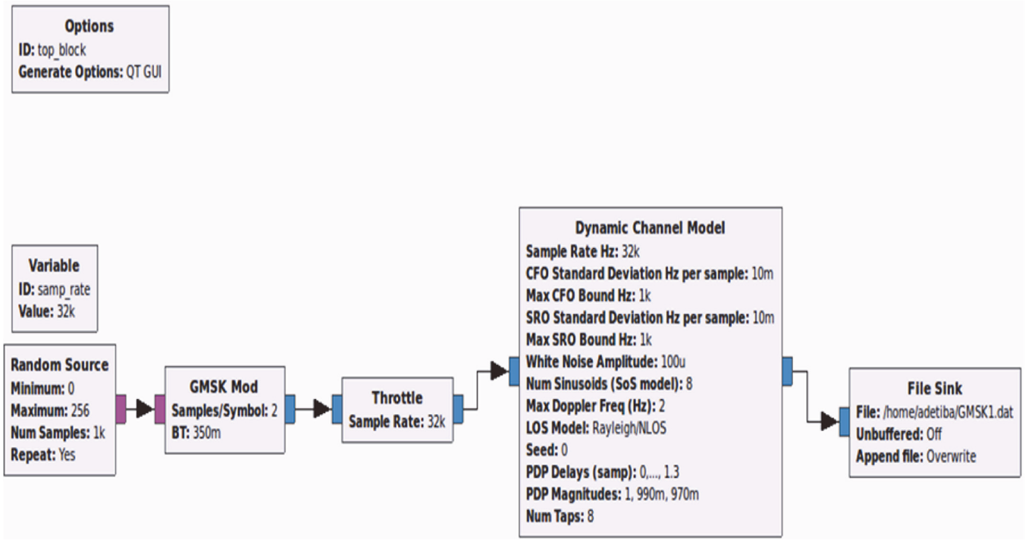


Fig. 1. A high-level block diagram for dataset generation.



Options
ID: top_block
Generate Options: QT GUI

Variable
ID: samp_rate
Value: 32k

Random Source
Minimum: 0
Maximum: 256
Num Samples: 1k
Repeat: Yes

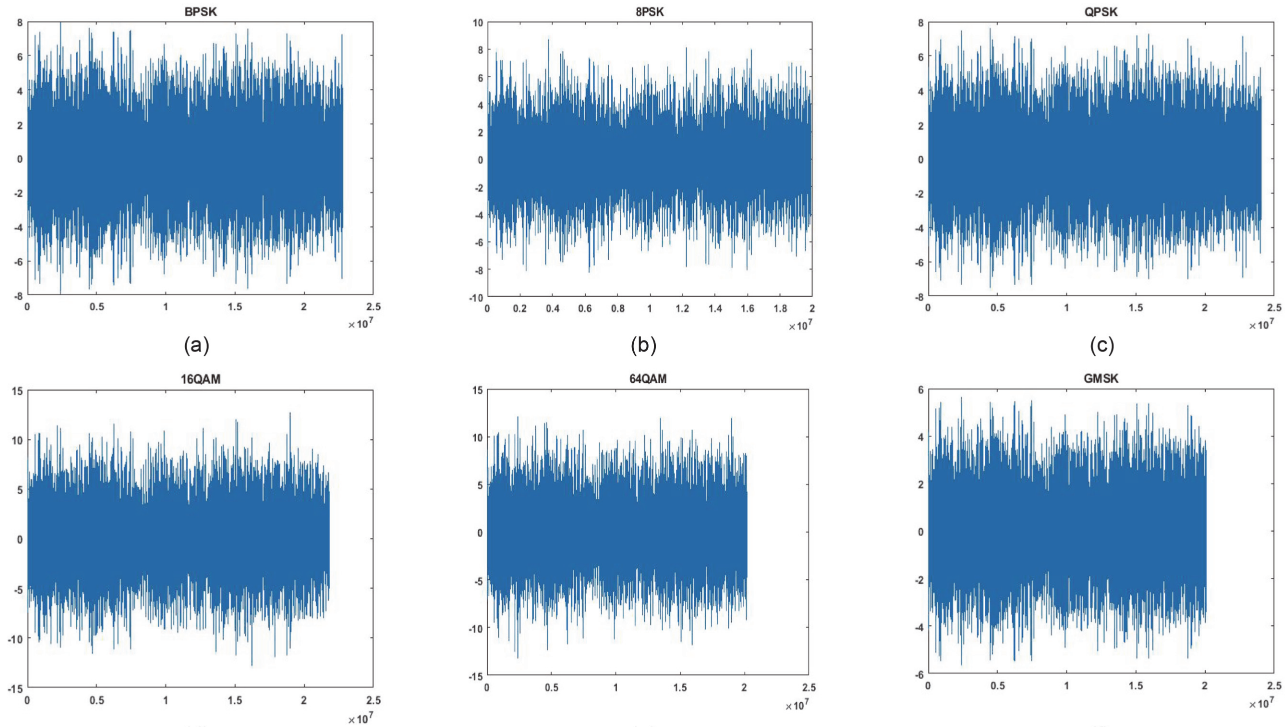
GMSK Mod
Samples/Symbol: 2
BT: 350m

Throttle
Sample Rate: 32k

Dynamic Channel Model
Sample Rate Hz: 32k
CFO Standard Deviation Hz per sample: 10m
Max CFO Bound Hz: 1k
SRO Standard Deviation Hz per sample: 10m
Max SRO Bound Hz: 1k
White Noise Amplitude: 100u
Num Sinusoids (SoS model): 8
Max Doppler Freq (Hz): 2
LOS Model: Rayleigh/NLOS
Seed: 0
PDP Delays (samp): 0,..., 1.3
PDP Magnitudes: 1, 990m, 970m
Num Taps: 8

File Sink
File: /home/adetiba/GMSK1.dat
Unbuffered: Off
Append file: Overwrite

Fig. 2. The GNU Radio GUI showing the GMSK signal generation.



(a) (b) (c)

(d) (e) (f)

Fig. 3. Sample plots of each modulation scheme used in dataset creation: (a) BPSK, (b) 8PSK, (c) QPSK, (d) 16QAM, (e) 64QAM, and (f) GMSK.

phase noise sum-of-sinusoids approach to simulate Rayleigh fading processes on the modulated signal with random time-varying channel response taps [25]. To mimic real-life propagation effects, we set the maximum CFO and SRO at 1 kHz and a standard deviation of 0.01, the maximum Doppler frequency used in fading simulation was set at 2 Hz and an AWGN of 100 μV for a signal of 1 V, thus, setting the signal-to-noise-ratio (SNR) at 80 dB [23-25]. We collected 50 samples for each modulation scheme to obtain a total of 300 samples for all the six schemes considered. Fig. 3 shows sample plots of the signal generated for each modulation scheme.

C. Dataset Processing and Features Extraction

We pre-processed the data obtained from the GNU

Radio file sink to make them usable in ML environments outside of the cognitive radio ecosystem by converting them from data files in .dat to .mat files as *N*-dimensional vectors using MATLAB R2020a [25]. These vectors are the in-phase and quadrature components of the transmitted signal. Operations in the radio domain are largely considered in complex baseband representation, which is not currently suitable for many ML toolbox operations; thus, we performed cepstrum analysis on the datasets to extract the RCC and MPRC features.

D. Cepstrum Analysis for Features Extraction

The term "cepstrum" in the literature was derived by reversing the first four letters of the term spectrum. Quefrency analysis (or quefrency alanysis), liftering, and cepstrum analysis are all terms used to describe operations on cepstra. The terms "quefrency," "alanysis," and "cepstrum" were coined by rearranging the letters in frequency, analysis, and spectrum. These invented terms are defined in this analogy to the older terms [27, 28]. Cepstrum analysis is a nonlinear signal processing technique with vast application in speech and image processing. In AMR, it is used in extracting signal envelopes since these envelopes have some information about the modulation of such a signal [8, 9]. To get the RCCs and MPRCs of the signal, we performed cepstrum analysis, which is the inverse Fourier transform of the real logarithm of the magnitude of the Fourier transform of the modulated signal $s(t)$. RCC is, therefore, a vector resulting from the implementation of Eq. (5) [28].

$$RCC = \frac{1}{2\pi}\int_{-\pi}^{\pi} log\left|S(e^{j\omega})\right|e^{-j\omega}d\omega \tag{5}$$

Appropriate windowing of RCC is carried out in the cepstral domain to form MPRC of the signal, as represented in Eq. (6) [28]:

$$MPRC = \frac{1}{2\pi}\int_{-\pi}^{\pi} exp\left|W(e^{j\omega})\right|e^{-j\omega}d\omega \tag{6}$$

where W is the Fourier transform of windowed RCC of the modulated signal $S(\omega)$. It is implied that for the phase sequence in Eq. (6) to be a minimum phase sequence, its complex cepstrum $\hat{s}(n)$ must be causal, thereby satisfying Eqs. (7) and (8) [29]:

$$\hat{s}(n) = \hat{s}(n) + \hat{s}(-n) \ \ for\ n = 1,2,\ ... \tag{7}$$

and

$$\hat{s}(n) = 0\ for\ n < 0 \tag{8}$$

It has been established that truncating real and complex cepstral coefficients at different frequency scales preserves different amounts of spectral details [30]. Thus, we extracted the first 45 (i.e., N = 45) and first 60 (i.e., N = 60) coefficients of the *N*-dimensional RCC and MPRC vectors obtained from the .mat files created for each of the six modulation schemes, respectively. These coefficients were then utilized as feature vectors (or descriptors) to train different shallow ANN architectures to evolve the appropriate descriptor (between the RCC and MPRC) with the number of coefficients for the most accurate AMR model.

E. ANN Model Design and Development

After feature extraction, creating an AMR model was formalized as a classification problem earlier in this paper with ANN as a viable classification method [8]. ANN is a collection of processing nodes called artificial neurons interlinked to model the human neurological framework in a biological brain using mathematical operations. A neural network is defined by the pattern of links (architecture) among the neurons, the methods, which specify the weights on the connections (learning algorithms), and the mathematical equations that determine the output (activation functions) [31]. This study leveraged the MLP, also known as ANN (a feed-forward architecture) for the classification task.

The real cepstrum coefficients are herein coded as RCCs-45 and RCCs-60, for the first 45 and the first 60 elements, respectively. At the same time, the first 45 and the first 60 elements of the extracted MPRC are coded as MPRCs-45 and MPRCs-60, respectively. Each of RCCs-45, RCCs-60, MPRCs-45, and MPRCs-60 form unique feature vectors that represent the outputs of the cepstrum analysis performed on each of the 50 samples per modulation scheme. This culminates into a total of 300 samples for the six modulation schemes considered. Notably, these feature vectors form the training dataset to the MLP-ANN. Thus, as shown in Fig. 4, the MLP-ANN architectures for RCCs-45 and MPRCs-45 have 45 input neurons (i.e. N = 45) in the input layer while the MLP-ANN architectures for RCCs-60 and MPRCs-60 (i.e. N = 60) contains 60 input layer neurons. A single hidden layer architecture is employed for model simplicity. The output layer of the MLP-ANN consists of six neurons which map the inputs to each class probability, as shown in Table 1. Each class is a unary encoded vector [29].

Extensive experiments were carried out to identify the most accurate network architecture for automated modulation recognition. For the hidden and output layers, pure linear (P), log-sigmoid (L), and tan-sigmoid (T) activation functions were utilized at different times with two learning algorithms: Levenberg-Marquardt (LM) and Scale Conjugate Gradient (SCG). Thus, a total of 18 MLP-ANN architectures were experimentally trained and evaluated. These architectures are coded as PPLM, LPLM, TPLM, PLLM, LLLM, TLLM, PTLM, LTLM, TTLM, PPSCG, LPSCG, TPSCG, PLSCG, LLSCG, TLSCG, PTSCG, LTSCG, and TTSCG [29]. The first two letters in the code represent the transfer functions at the

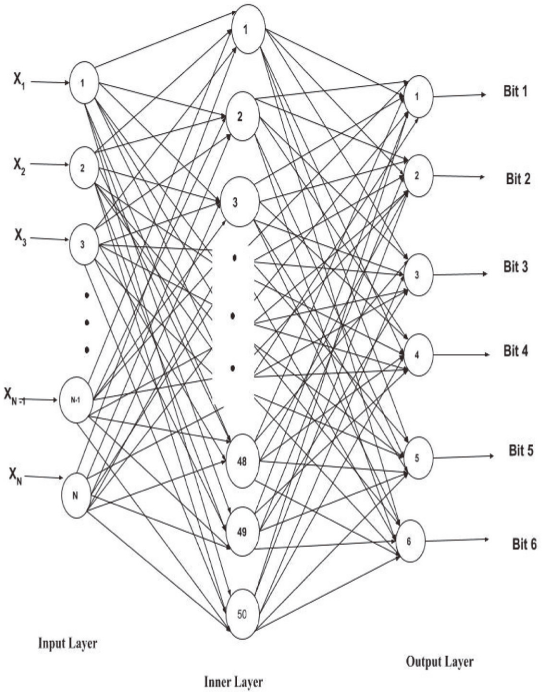
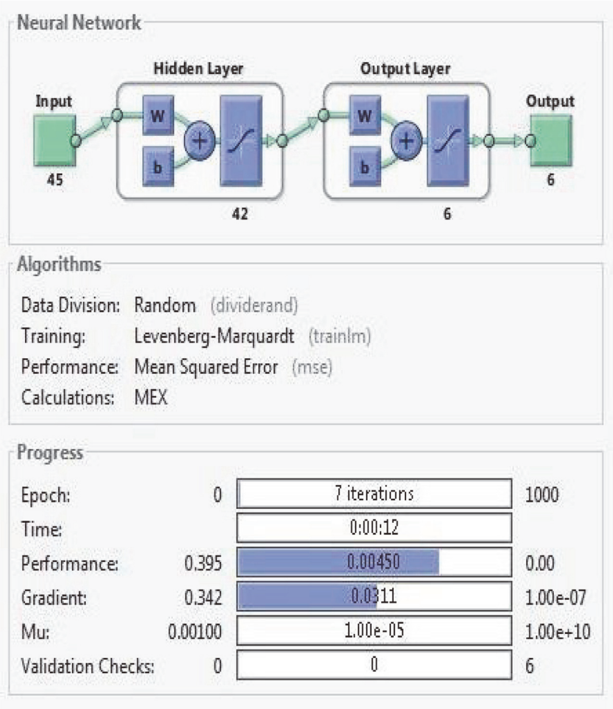


Fig. 4. Network architecture of the ANN model.

Table 1. MLP-ANN target outputs

Class	Bit1	Bit2	Bit3	Bit4	Bit5	Bit6
BPSK	0	0	0	0	0	1
8PSK	0	0	0	0	1	0
QPSK	0	0	0	1	0	0
16QAM	0	0	1	0	0	0
64QAM	0	1	0	0	0	0
GMSK	1	0	0	0	0	0

hidden and the output layers, respectively, while the remaining letters represent the learning algorithm. The number of neurons in the hidden layer was also varied from 1 to 50 for each architecture. The ANN model implementation, training, validation, and testing were performed using the Neural Network Toolbox in MATLAB 2020a running on an Intel Core i5-3210M CPU@2.50 GHz speed with 4 GB RAM and 64-bit Windows 7 operating system. To achieve better generalization, the MATLAB the MATLAB script was programmed to randomize the sample distribution, while dividing the training data into three subsets for training, validation, and testing in the ratio 80:10:10, respectively. Fig. 5 captures a sample of the ANN architecture training visualizations.



Algorithms

Data Division: Random (dividerand)
Training: Levenberg-Marquardt (trainlm)
Performance: Mean Squared Error (mse)
Calculations: MEX

Progress

Epoch:	0	7 iterations	1000
Time:		0:00:12	
Performance:	0.395	0.00450	0.00
Gradient:	0.342	0.0311	1.00e-07
Mu:	0.00100	1.00e-05	1.00e+10
Validation Checks:	0	0	6

Fig. 5. ANN training visualization.

F. ANN Models Performance Evaluation

Adequate steps were taken to ensure that the models built were consistent. To achieve this, we fine-tuned each ANN model and trained for 20 iterations. In essence, we trained, tested, and evaluated 18,000 network configurations. The evaluation of ANN models was based on the following standard Key Performance Indices (KPIs): accuracy, sensitivity, and specificity. The accuracy, sensitivity, and specificity indices were calculated from four parameters, which are true positives (TP), false negatives (FN), true negatives (TN), and false positives (FP). TP represents correct positive prediction, FP represents incorrect positive prediction, TN represents correct negative prediction, and FN incorrect negative prediction of the modulation schemes. The KPIs are mathematically represented as Eqs. (9)-(11):

$$Accuracy = (TP+TN)/(TP+TN+FP+FN) \times 100 \quad (9)$$

$$Sensitivity = TP/(TP+TN) \times 100 \quad (10)$$

$$Specificity = TN/(TP+TN) \times 100 \quad (11)$$

Also, the MSE, mean absolute error (MAE), and standard deviation were computed to measure the errors and to decide which architecture is best suited for the AMR.

III. RESULTS

To illustrate the dissimilarities of the MPRCs and

RCCs feature vectors, their histogram distributions are presented in Figs. 6-9 for MPRCs-45, RCCs-45, MPRCs-60, and RCCs-60, respectively. Through visual inspection, it can be observed that there are considerable differences in the shapes and distributions of the histograms for the two different descriptors. However, the MPRCs-45 and MPRCs-60 exhibit similar shapes or distributions for the same modulation scheme since both are derived from the

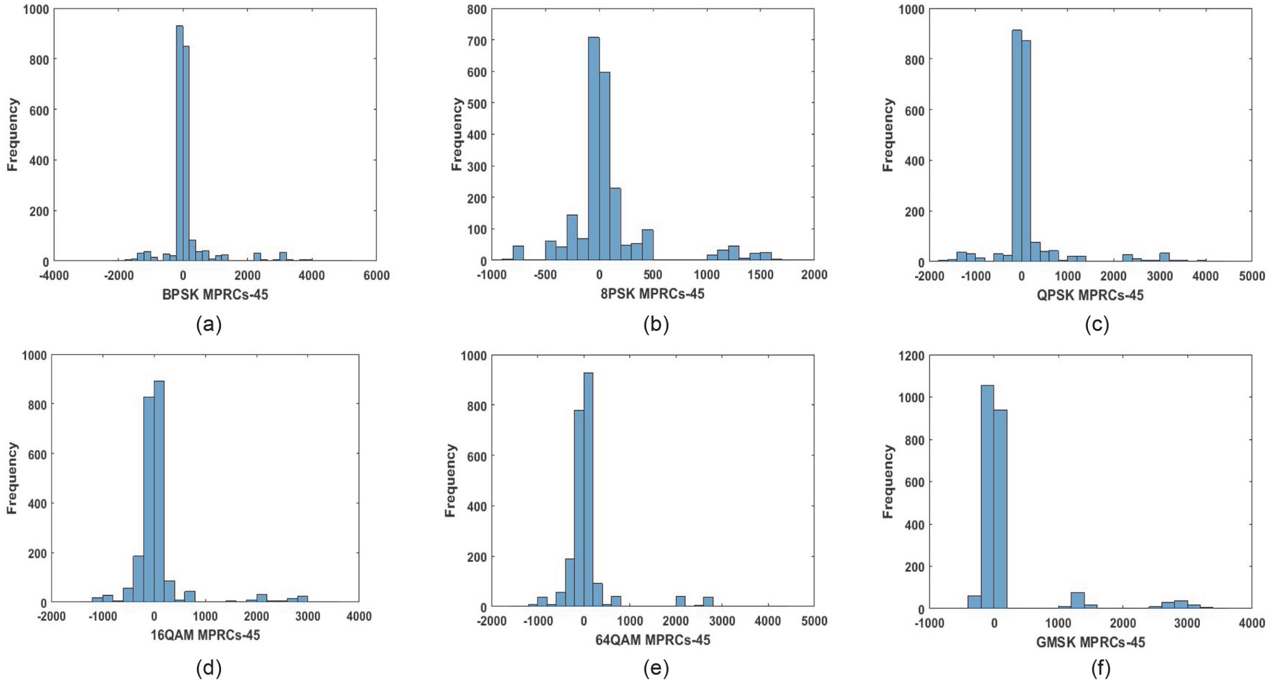


Fig. 6. Histogram plots for MPRCs-45 feature vectors extracted from the modulated signals: (a) BPSK, (b) 8PSK, (c) QPSK, (d) 16QAM, (e) 64QAM, and (f) GMSK.

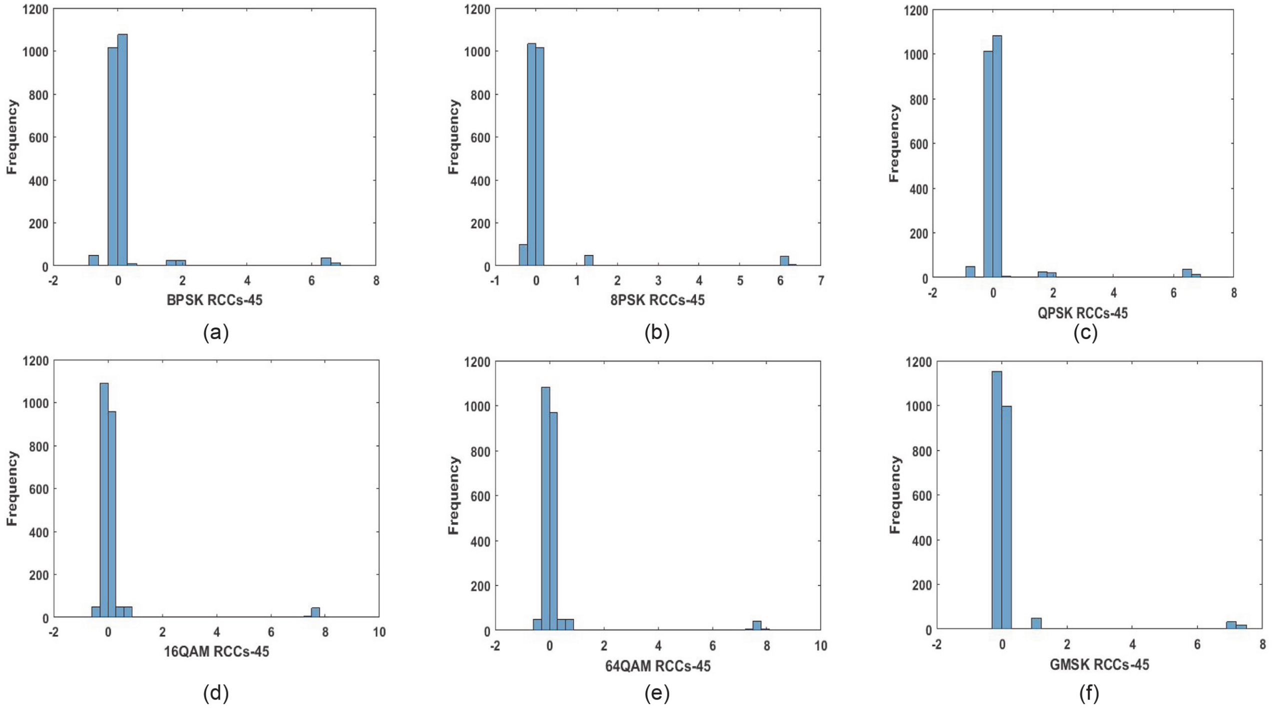
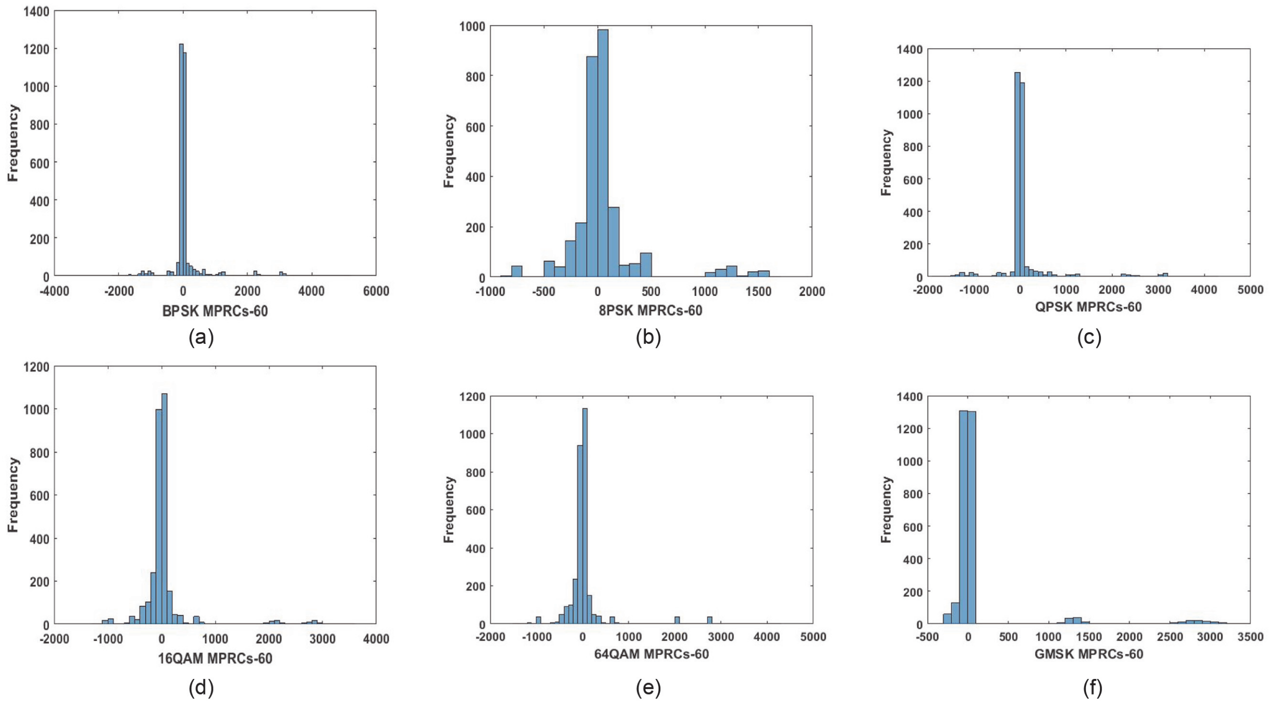


Fig. 7. Histogram plots for RCCs-45 feature vectors extracted from the modulated signals: (a) BPSK, (b) 8PSK, (c) QPSK, (d) 16QAM, (e) 64QAM, and (f) GMSK.

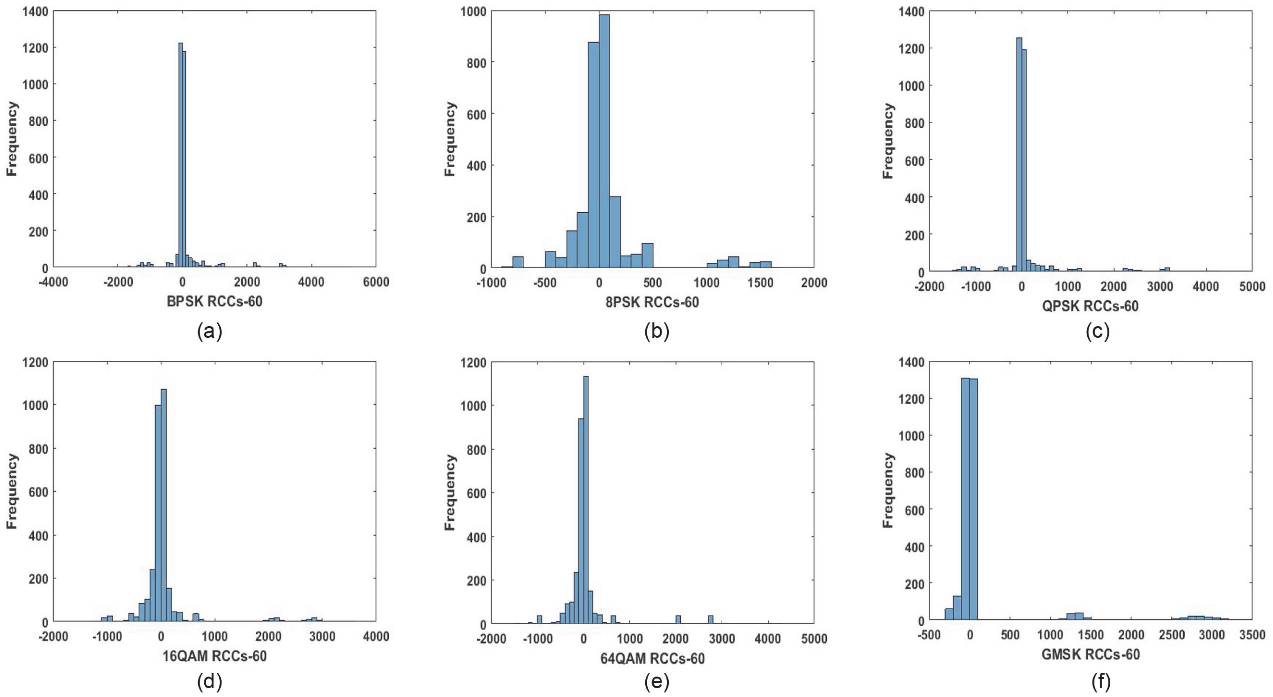
same descriptor. The marginal difference is, however, based on the fact that MPRCs-60 has additional 15 coefficients. The same scenario is observed for RCCs-45 and RCCs-60. Nonetheless, there are remarkable differences between the histograms of MPRCs and RCCs even for the same scheme. From the histograms, we identified



(a) (b) (c)

(d) (e) (f)

Fig. 8. Histogram plots for MPRCs-60 feature vectors extracted from the modulated signals: (a) BPSK, (b) 8PSK, (c) QPSK, (d) 16QAM, (e) 64QAM, and (f) GMSK.



(a) (b) (c)

(d) (e) (f)

Fig. 9. Histogram plots for RCCs-60 feature vectors extracted from the modulated signal: (a) BPSK, (b) 8PSK, (c) QPSK, (d) 16QAM, (e) 64QAM, and (f) GMSK.

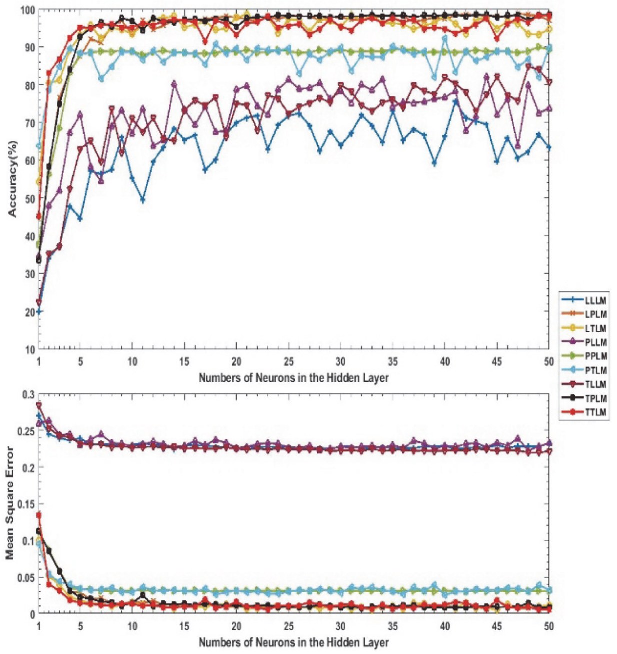


Fig. 10. Performance of MLP-ANN Architectures trained using the LM learning algorithm and MPRCs-45 descriptor.

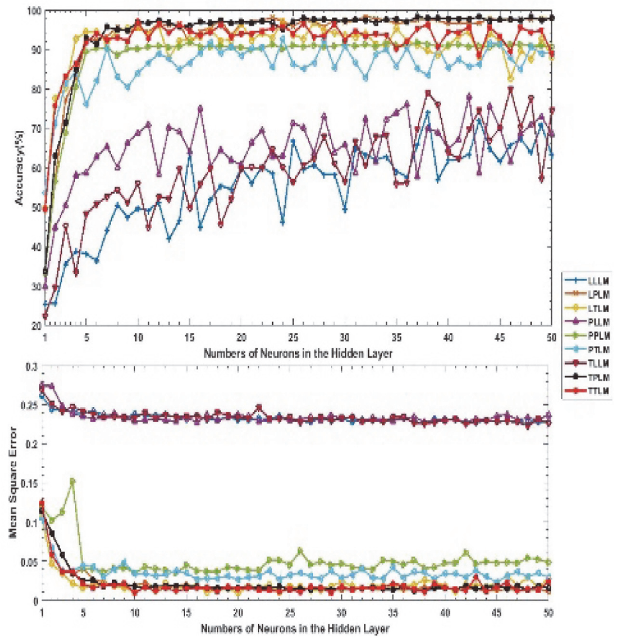


Fig. 11. Performance of MLP-ANN Architectures trained using the LM learning algorithm and MPRCs-60 descriptor.

potential outliers and investigated them. It can be inferred that they are legitimate data points that accurately describe the variation in the distribution of the studied descriptors. In such a case, knowledge is gained from the class distribution, which may explain why MPRCs with fewer outliers and disparities performed better in the final classification task. These clear-cut dissimilarities among the descriptors further provide an empirical basis for the application of MLP-ANN to realize modulation recognition tasks.

Fig. 10 illustrates the performances of the MLP-ANN architectures that were trained using the LM learning algorithm and MPRCs-45 descriptor. It shows the effects of varying the number of neurons in the hidden layer and the impacts of different combinations of activation functions used at the hidden and output layers on the performances of different MLP-ANNs. Generally, the ANNs' classification accuracy improves as the number of neurons increases [29]. While LLLM, PLLM, and TLLM fluctuate throughout, significant accuracies were achieved in LTLM, TTLM, PTLM, PPLM, LPLM, and TPLM. As we increased the number of neurons from 1 to 5 by adding one at a time, the accuracies of LTLM, TTLM, PTLM, PPLM, LPLM, and TPLM swiftly improved to 92.9%, 95.1%, 88.2%, 88.0%, 87.5%, and 92.6%, respectively [26, 28, 29]. Increasing the number of neurons from 5 to 50, adding one at a time, the performances of PPLM did not show any significant improvement. However, the performances of TTLM, LTLM, and PTLM slightly increased with the increasing number of neurons but not as much as observed in LPLM and TPLM. There are gradual but inconsistent improvements in the trends of accuracies in TTLM, LTLM, and LPLM compared to TPLM. Meanwhile, LPLM reached a peak accuracy of 98.7% (and MSE of 0.0075) at 47 neurons, thereby outperforming the others.

Fig. 11 illustrates the performances of the MLP-ANN models that were trained using the LM learning algorithm and MPRCs-60 descriptor. It had a similar pattern as Fig. 7, showing that using the first 60 coefficients as feature vectors performs no better than using the first 45 coefficients. Generally, the MLP-ANNs' classification accuracies improved as the number of neurons increased. While LLLM, PLLM, and TLLM fluctuated throughout, significant accuracies were achieved in LTLM, TTLM, PTLM, PPLM, LPLM, and TPLM. As we increased the number of neurons to 5, adding one at a time, the accuracies of LTLM, TTLM, PTLM, PPLM, LPLM, and TPLM swiftly improved to 94.5%, 91.7%, 76.1%, 89.4%, 92.3%, and 92.9%, respectively. Increasing the number of neurons from 5 to 50, adding one at a time, the performances of PPLM did not show any significant improvement. However, the performances of TTLM, LTLM, and PTLM slightly increased with the increasing number of neurons but not as much as observed in LPLM and TPLM. There are gradual but inconsistent improvements in the trends of accuracies in TTLM, LTLM, and LPLM compared to TPLM. LPLM reached a peak accuracy of 98.3% at 32 neurons outperforming others.

Fig. 12 illustrates the performances of the MLP-ANN models that were trained using the LM learning algorithm and RCCs-45 descriptor. It shows the effects of varying

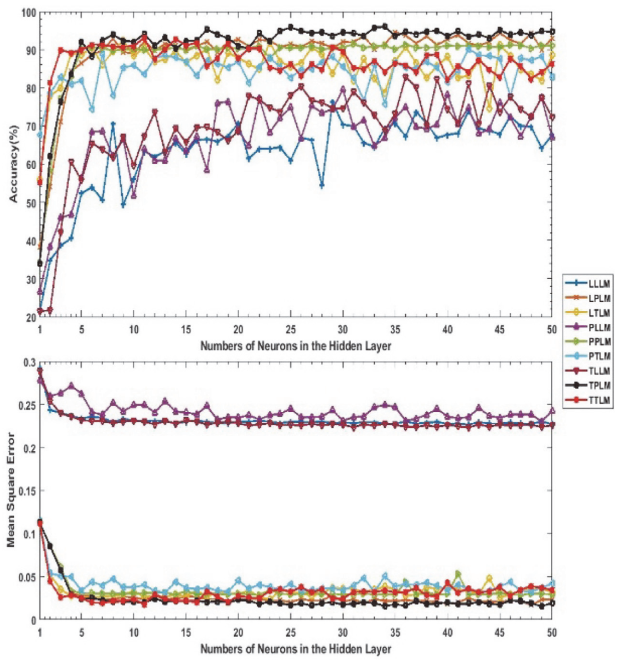


Fig. 12. Performance of MLP-ANN Architectures trained using the LM learning algorithm and RCCs-45 descriptor.

the number of neurons in the hidden layer and the impacts of different combinations of activation functions used at the hidden and output layers on the performance of different MLP-ANNs. Generally, the MLP-ANNs' classification accuracies improve as the number of neurons increases. While LLLM, PLLM, and TLLM fluctuate throughout, significant accuracies were achieved in LTLM, TTLM, PTLM, PPLM, LPLM, and TPLM. As we increased the number of neurons to 5, adding one at a time, the accuracies of LTLM, TTLM, PTLM, PPLM, LPLM, and TPLM swiftly improved to 88.7%, 90.0%, 81.8%, 90.4%, 86.5%, and 92.1%, respectively. Increasing the number of neurons from 5 to 50, adding one at a time, the performances of PPLM did not show any significant improvement. Nevertheless, the performances of TTLM, LTLM, and PTLM slightly increased with the increasing number of neurons but not as much as observed in LPLM and TPLM. There were gradual but inconsistent improvements in the trends of accuracies in TTLM, LTLM, and LPLM compared to TPLM. However, TPLM reached a peak accuracy of 96.1% at 34 neurons, thereby outperforming the others.

Fig. 13 illustrates the performance of the MLP-ANN models that were trained using the LM learning algorithm and RCCs-60 descriptor. It had a similar pattern as Fig. 9, showing that using the first 60 coefficients as feature vectors does not perform better than the first 45 coefficients. Generally, the MLP-ANNs' classification accuracies improved as the number of neurons increased. While LLLM, PLLM, and TLLM fluctuated throughout, significant accuracies were achieved in LTLM, TTLM, PTLM, PPLM, LPLM, and TPLM. As we increased the number of neurons to 5, adding one at a time, the accuracies of LTLM, TTLM, PTLM, PPLM, LPLM, and TPLM swiftly improved to 83.2%, 88.0%, 88.0%, 90.0%, 87.3%, and 94.6%, respectively. Increasing the number of neurons from 5 to 50, adding one at a time, the performances of PPLM did not show any significant improvement despite increasing the number of neurons. However, the performance of TTLM, LTLM, and PTLM slightly increased with the increasing number of neurons but not as much as observed in LPLM and TPLM. There were gradual but inconsistent improvements in the trends of accuracies in TTLM, LTLM, and LPLM compared to TPLM. TPLM reached a peak accuracy of 96.7% at 23 neurons outperforming others.

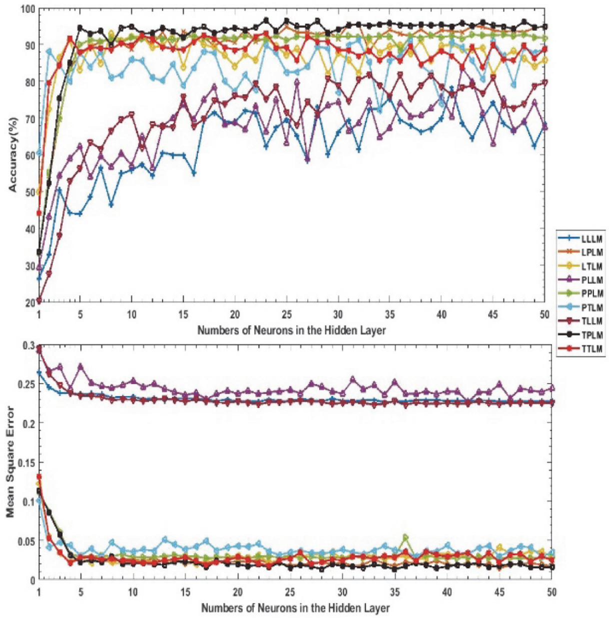


Fig. 13. Performance of MLP-ANN architectures trained using the LM learning algorithm and RCCs-60 descriptor.

For each of the real cepstrum descriptors (i.e., RCCs-45 and RCCs-60) from the confusion matrix, after the training, testing, and validation of the ANN training phase were completed, TPLM gave the best accuracies of 96.1% and 96.7% at 34 and 23 neurons, respectively. On the other hand, for the minimum phase reconstruction MPRCs-45 and MPRCs-60 descriptors, LPLM gave the best accuracies of 98.8% and 98.3% at 47 and 32 neurons, respectively. This clearly shows that the MPRCs are a stronger cepstrum descriptor than the RCCs. It also reveals that choosing between the first 60 coefficients, i.e., RCCs-60 and MPRCs-60, and the first 45 coefficients, i.e., RCCs-45 and MPRCs-45, is a trade-off between achieving accuracy and the number of neurons in the hidden layer.

Fig. 14 illustrates the performances of the MLP-ANN models that were trained using the SCG learning algorithm

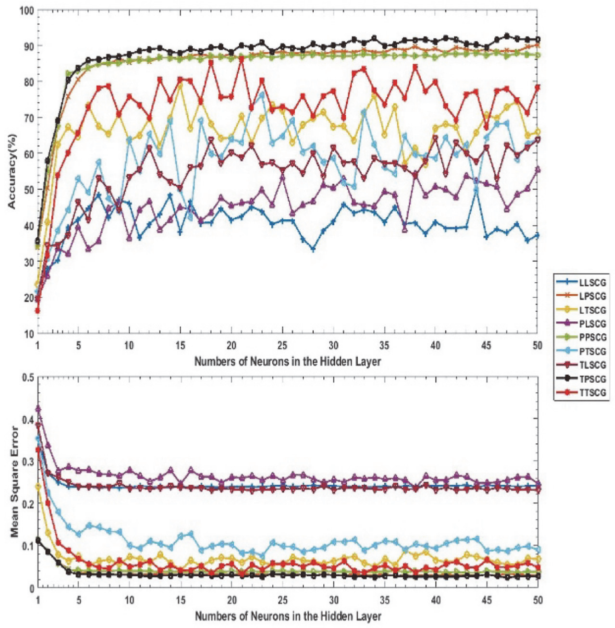


Fig. 14. Performance of MLP-ANN architectures trained using the SCG learning algorithm and MPRCs-45 descriptor.

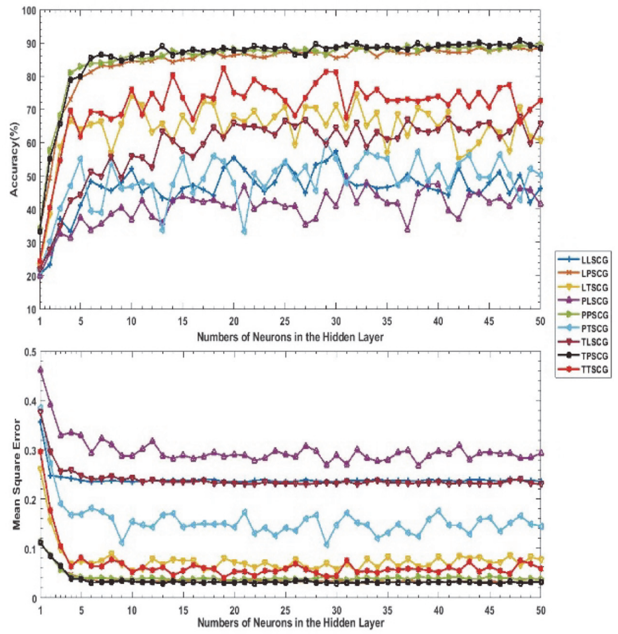


Fig. 15. Performance of MLP-ANN architectures trained using the SCG learning algorithm and RCCs-45 descriptor.

and MPRCs-45 descriptor. It shows the effects of varying the number of neurons in the hidden layer and the impacts of different combinations of activation functions used at the hidden and output layers on the performance of different MLP-ANNs. Generally, the MLP-ANNs classification accuracies improved as the number of neurons increased, similar to what was observed with LM. While LLSCG, PLSCG, TLSCG, PTSCG, LTSCG, and TTSCG fluctuated throughout, significant accuracies were achieved in TTSCG, LPSCG, and TPSCG. As we increased the number of neurons to 5, adding one at a time, the accuracies of TTSCG, LPSCG, and TPSCG swiftly improved to 65.9%, 80.6%, and 83.8%, respectively [28]. Increasing the number of neurons from 5 to 50, adding one at a time, the performance of PPSCG did not show any significant improvement despite increasing the number of neurons. But, the performance of LPSCG and TPSCG slightly increased with the increasing number of neurons, however, not as much as observed in TPSCG. TPSCG reached a peak accuracy of 92.6% at 47 neurons outperforming the others.

Fig. 15 illustrates the performance of the ANN models that were trained using the SCG learning algorithm and MPRCs-60 descriptor. It shows the effects of varying the number of neurons in the hidden layer and the impacts of different combinations of activation functions used at the hidden and output layers on the performance of different MLP-ANNs. Generally, the MLP-ANNs classification accuracies improved as the number of neurons increased, similar to what was observed with LM. While LLSCG, PLSCG, TLSCG, PTLM, LTSCG, and TTSCG fluctuated throughout, significant accuracies were achieved in TTSCG, LPSCG, and TPSCG. As we increased the number of neurons to 5, adding one at a time, the accuracies of TTSCG, LPSCG, and TPSCG improved to 66.5%, 82.6%, and 92.9%, respectively. Increasing the number of neurons from 5 to 50, adding one at a time, the performance of PPSCG did not show any significant improvement despite increasing the number of neurons. But, the performance of LPSCG and TPSCG slightly increased with the increasing number of neurons, however, not as much as observed in TPSCG. TPSCG reached a peak accuracy of 91.9% at 48 neurons outperforming the others.

Fig. 16 illustrates the performance of the ANN models that were trained using the SCG learning algorithm and RCCs-45 descriptor. It shows the effects of varying the number of neurons in the hidden layer and the impacts of different combinations of activation functions used at the hidden and output layers on the performance of different ANNs. Generally, the ANNs classification accuracy improved as the number of neurons increased, similar to what was observed with LM. While LLSCG, PLSCG, TLSCG, PTLM, LTSCG, and TTSCG fluctuated throughout, significant accuracies were achieved in TTSCG, LPSCG, and TPSCG. As we increased the number of neurons to 5, adding one at a time, the accuracies of TTSCG, LPSCG, and TPSCG swiftly improved to 73.5%, 82.5%, and 82.0%, respectively. Increasing the number of neurons from 5 to 50, adding one at a time, the performance of PPSCG did not show any significant improvement despite increasing the number of neurons. But, the performance of LPSCG and TPSCG slightly increased

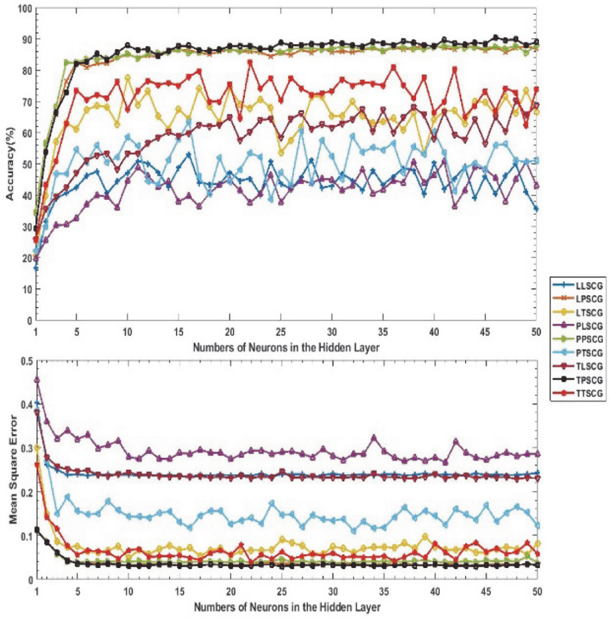


Fig. 16. Performance of MLP-ANN architectures trained using the SCG learning algorithm and RCCs-45 descriptor.

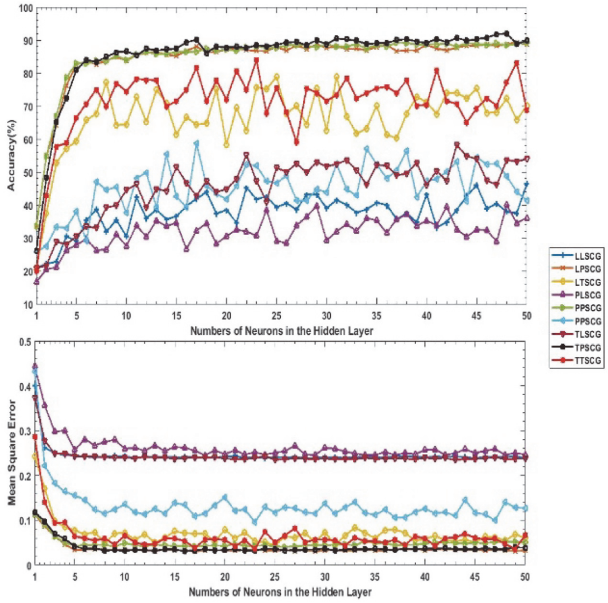


Fig. 17. Performance of MLP-ANN architectures trained using the SCG learning algorithm and RCCs-60 descriptor.

with the increasing number of neurons, however, not as much as observed in TPSCG. TPSCG reached a peak accuracy of 90.4% at 46 neurons outperforming the others.

Fig. 17 illustrates the performances of the MLP-ANN models that were trained using the SCG learning algorithm and RCCs-60 descriptor. It shows the effects of varying the number of neurons in the hidden layer and the impacts of different combinations of activation functions used at the hidden and output layers on the performance of different MLP-ANNs. Generally, the MLP-ANNs classification accuracies improved as the number of neurons increased, similar to what was observed with LM. While LLSCG, PLSCG, TLSCG, PTLM, LTSCG, and TTSCG fluctuated throughout, significant accuracies were achieved in TTSCG, LPSCG, and TPSCG. As we increased the number of neurons to 5, adding one at a time, the accuracies of TTSCG, LPSCG, and TPSCG improved to 62.0%, 79.5%, and 80.0%, respectively. Increasing the number of neurons from 5 to 50, adding one at a time, the performance of PPSCG did not show any significant improvement despite increasing the number of neurons. Meanwhile, the performances of LPSCG and TPSCG slightly increased with the increasing number of neurons but not as much as observed in TPSCG. Thus, TPSCG reached a peak accuracy of 90.9% at 48 neurons, thereby outperforming the other configurations.

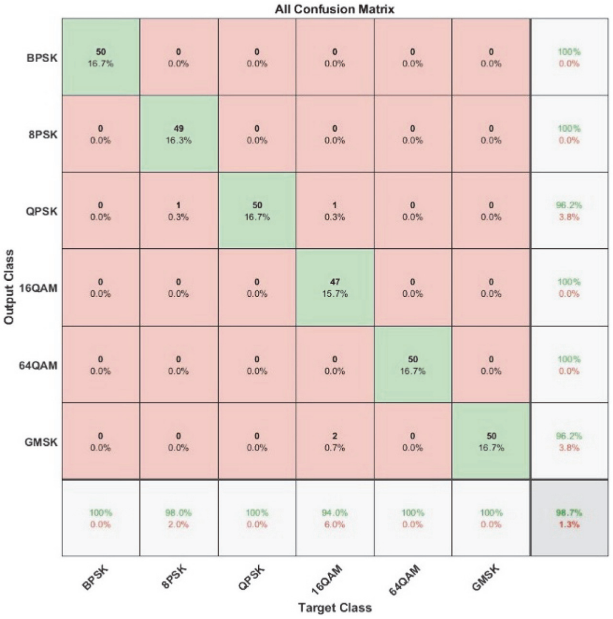
For each of the real cepstrum RCCs-45 and RCCs-60 descriptors, from the confusion matrix, after the training, testing, and validation of the ANN training phase were completed, TPSCG gave the best accuracies of 90.4% and 90.9% at 46 and 48 neurons, respectively. On the other hand, for the minimum phase reconstruction MPRCs-45 and MPRCs-60 descriptors, TPSCG gave the best accuracies of 92.6% and 91.9% at 47 and 48 neurons, respectively. A similar trend is seen in the performance of the MLP-ANN architectures trained using the SCG learning algorithm to what was obtained using LM, but the results show clearly that the LM learning algorithm outperformed the SCG learning algorithm.

It also confirmed that the MPRCs as a cepstrum descriptor consistently performed better than RCCs. Notably, choosing between the first 60 coefficients, i.e. RCCs-60 and MPRCs-60, and the first 45 coefficients, i.e. RCCs-45 and MPRCs-45, did not have a significant difference in terms of accuracy of the architectures. Thus, based on previously reported experimental results, MPRCs are deemed a better descriptor for the recognition of signal modulation schemes than RCCs; hence, new knowledge has been revealed with this study. Comparing the number of coefficients extracted, 60 coefficients of the MPRCs have the highest recognition accuracy of 98.3% with a lower number of neurons in the hidden layer of the MLP-ANNs. Though the 45 coefficients MPRCs have the best recognition accuracy of 98.7%, this was achieved with 47 neurons in the hidden layer, which is higher than the 32 neurons utilized by the 60 coefficients MPRCs (as shown in Table 2). A similar trend is observed between RCCs-45 and RCCs-60.

Table 2 shows the MLP-ANNs models with the best performances in each scenario based on the rigorous experimentations conducted in this study. Overall, LPLM architecture with 47 neurons in the hidden layer trained with MPRCs-45 gave the best result (accuracy = 98.7%,

Table 2. ANN architecture with the overall best performances

Feature vector	ANN architecture	Number of neurons	Accuracy (%)	MSE	SD	MAE
MPRCs-45	LPLM	47	98.7	0.007479	0.623844	0.0390
MPRCs-60	LPLM	32	98.3	0.015988	0.675382	0.0544
RCCs-45	TPLM	34	96.1	0.015896	2.201209	0.0573
RCCs-60	TPLM	23	96.7	0.015991	2.094270	0.0628


All Confusion Matrix

Output Class \ Target Class	BPSK	8PSK	QPSK	16QAM	64QAM	GMSK	
BPSK	50 16.7%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
8PSK	0 0.0%	49 16.3%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
QPSK	0 0.0%	1 0.3%	50 16.7%	1 0.3%	0 0.0%	0 0.0%	96.2% 3.8%
16QAM	0 0.0%	0 0.0%	0 0.0%	47 15.7%	0 0.0%	0 0.0%	100% 0.0%
64QAM	0 0.0%	0 0.0%	0 0.0%	0 0.0%	50 16.7%	0 0.0%	100% 0.0%
GMSK	0 0.0%	0 0.0%	0 0.0%	2 0.7%	0 0.0%	50 16.7%	96.2% 3.8%
	100% 0.0%	98.0% 2.0%	100% 0.0%	94.0% 6.0%	100% 0.0%	100% 0.0%	98.7% 1.3%

Fig. 18. All confusion matrix of the best MLP-ANN architecture trained with MPRCs-45 and LM algorithm.

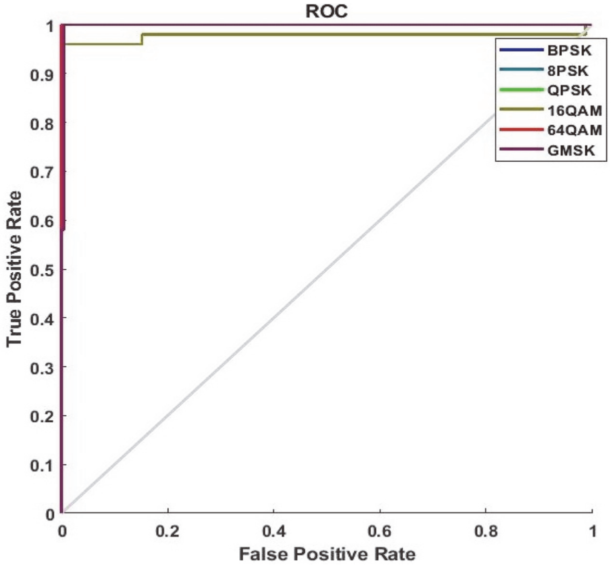


Fig. 19. ROC Plot of the best MLP-ANN architecture trained with MPRCs-45 and LM algorithm.

MSE = 0.007479), as shown in Table 2. The selection of hidden neurons using neural networks is one of the most challenging problems that researchers face. In the ANN training process, there is a problem with overtraining. Overtraining is similar to the problem of data overfitting. The model may have a significant training error due to overfitting if it has a higher number of hidden neurons. The top-ranked models were also evaluated using two loss metrics vis-a-vis MSE and MAE. The number of hidden neurons is fixed in AMR based on minimum error efficiency. It is therefore recommended as the MLP-ANN architecture for accurate AMR.

Furthermore, the confusion matrix and receiver operating characteristic (ROC) curve were plotted as shown in Figs. 18 and 19. With the ROC curves, we were able to compute the sensitivity (100%) and specificity (99.3%) of the AMR model in recognizing and classifying the modulation schemes.

In order to test the AMR model (MPRCs-45) that posted the best performance of 98.7% accuracy and 0.0075 MSE, we randomly selected five samples for each modulation scheme out of the training data to carry out in-sample testing on the AMR model. It accurately recognized them all with an accuracy of 100%. Further tests with out-of-sample over-the-air datasets suggested that the synthetic/simulated signals required further signal pre-processing to fit into the curated real signal perfectly. This provides an open question and a basis for further research studies in this domain.

IV. DISCUSSION

The current study presented the results of a new cepstral method [3, 11] for AMR using MPRC. The experimental results show that modulation schemes can be identified by extracting MPRCs from signals and using these as descriptors for modulation recognition similar to MFCCs [12, 13]. The MLP-ANN with LPLM architecture having 47 hidden neurons trained with MPRCs descriptor posted the best performance of 98.7% accuracy, 0.0075 MSE, 100% sensitivity, and 99.3% specificity, which outperformed the descriptors (e.g., MFCCs and RCCs) previously used in the literature [3, 8-10]. The MPRCs provide a realistic way for all distinguishable band-limited signals to be decoded when the detector is sensitive to the signal field strength and phase but less sensitive to noise. Furthermore, the creation of the dataset for our experiments in this study factored in wireless communication random processes such as CFO, SRO, AWGN, multipath propagation, and selective fading. This is a major improvement in dataset creation for related studies in the literature [12, 13] and an effort to extend the incorporation of cognitive capability in our earlier works [32, 33]. The created dataset in this study and the best MPRC-based AMR model with a .m script for the testing phase are available as open research resources on the Advanced Signal Processing and Machine Intelligence Research (ASPMIR) laboratory GitHub repository at (https://github.com/aspmirlab/ModRecCorpus.git) for interested researchers to explore.

V. CONCLUSION

This paper presents the development and experimentation results on AMR using a dataset created with GNU Radio Companion for six different modulation schemes, MPRC and RCC descriptors, and MLP-ANN architectures of different configurations. The MLP-ANN architecture with logsig activation function in the hidden layer neurons, purelin in the output layer neuron, LM backpropagation algorithm and trained using MPRC descriptor of 45 coefficients gave the best performance results. This compact MPRC descriptor provides an enhanced alternative and vital contribution to the FB AMR literature. Thus, modern wireless communication engineers and researchers can leverage the MPRC-based AMR model to implement dynamic spectrum sensing in cognitive radio systems and for automatic selection of appropriate signal demodulators. In the future, we hope to extend this study for more modulation schemes that capture different SNRs. We also hope to explore state-of-the-art deep learning pipelines for AMR tasks in real-time deployment scenarios.

CONFLICT-OF-INTEREST STATEMENT

The authors declare that there is no conflict of interest.

ACKNOWLEDGMENTS

This research was done at Covenant University's Advanced Signal Processing and Machine Intelligence Research (ASPMIR) group laboratory, IoT Enabled Smart and Connected Community Research Cluster. The High-Performance Computing node of the Covenant Applied Informatics and Communication African Centre of Excellence (CApIC-ACE) FEDGEN Testbed was utilized for experimentations. The Covenant University Centre for Research, Innovation, and Development (CUCRID) at Covenant University in Ota, Nigeria, funded the publication of this paper. We also acknowledge the Office of the DVC Research, Innovation, and Engagement as well as the Institute for Systems Science (ISS), Durban University of Technology, Durban, South Africa, for the collaboration on this study.

REFERENCES

1. E. Adetiba, V. O. Matthews, S. A. Daramola, I. A. Samuel, A. A. Awelewa, and M. E. U. Eleanya, "Nomadic base station (NBS): a software defined radio (SDR) based architecture for capacity enhancement in mobile communications networks," *International Journal of Engineering & Technology*, vol. 11, no. 1, pp. 191-195, 2011.
2. Y. Linn, "New structures for modulation classification and SNR estimation with applications to cognitive radio and software defined radio," in *Proceedings of 2014 IEEE 27th Canadian Conference on Electrical and Computer Engineering (CCECE)*, Toronto, Canada, 2014, pp. 1-8.
3. A. H. Kelechi, M. H. Alsharif, A. M. Ramly, N. F. Abdullah, and R. Nordin, "The four-C framework for high capacity ultra-low latency in 5G networks: a review," *Energies*, vol. 12, no. 18, article no. 3449, 2019. https://doi.org/10.3390/en12183449
4. A. Hazza, M. Shoaib, S. A. Alshebeili, and A. Fahad, "An overview of feature-based methods for digital modulation classification," in *Proceedings of 2013 1st International Conference on Communications, Signal Processing, and their Applications (ICCSPA)*, Sharjah, UAE, 2013, pp. 1-6.
5. B. Ramkumar, "Automatic modulation classification for cognitive radios using cyclic feature detection," *IEEE Circuits and Systems Magazine*, vol. 9, no. 2, pp. 27-45, 2009.
6. A. Papoulis, *Probability, Random Variables, and Stochastic Processes*. New York, NY: McGraw-Hill, 1965.
7. S. Imai, "Cepstral analysis synthesis on the mel frequency scale," in *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, Boston, MA, 1983, pp. 93-96.
8. M. Zaerin and B. Seyfe, "Multiuser modulation classification based on cumulants in additive white Gaussian noise channel," *IET Signal Processing*, vol. 6, no. 9, pp. 815-823, 2012.
9. M. M. Sohul, B. Ramkumar, and T. Bose, "Multiuser automatic modulation classification for cognitive radios using distributed sensing in multipath fading channels," in *Proceedings of 2012 7th international ICST Conference on Cognitive Radio Oriented Wireless Networks and Communications (CROWNCOM)*, Stockholm, Sweden, 2012, pp. 71-76.
10. J. Jantti, S. Chaudhari, and V. Koivunen, "Detection and classification of OFDM waveforms using cepstral analysis," *IEEE Transactions on Signal Processing*, vol. 63, no. 16, pp. 4284-4299, 2015.
11. S. C. Kremer and J. Shiels, "A testbed for automatic modulation recognition using artificial neural networks," in *Proceedings of Canadian Conference on Electrical and Computer Engineering (CCECE)*, St. John's, Canada, 1997, pp. 67-70.

12. R. M. Al-Makhlasawy, M. M. Abd Elnaby, and H. A. El-Khobby, “C24. Automatic modulation recognition in wireless systems using cepstral analysis and neural networks,” in *Proceedings of 2012 29th National Radio Science Conference (NRSC)*, Cairo, Egypt, 2012, pp. 353-362.
13. R. M. Al-Makhlasawy, M. M. A. Elnaby, H. A. El-Khobby, S. El-Rabaie, and F. E. A. El-Samie, “Automatic modulation recognition in OFDM systems using cepstral analysis and support vector machines,” *Journal of Telecommunications System & Management*, vol. 1, no. 3, article no. 1000105, 2012. https://doi.org/10.4172/2167-0919.1000105
14. A. Mecozzi, “A necessary and sufficient condition for minimum phase and implications for phase retrieval,” 2016 [Online]. Available: https://arxiv.org/abs/1606.04861.
15. J. R. Fienup, “Reconstruction of an object from the modulus of its Fourier transform,” *Optics Letters*, vol. 3, no. 1, pp. 27-29, 1978.
16. M. R. Fernandez-Ruiz and A. Carballar, “Minimum-phase reconstruction method for the determination of the specifications of FBGs,” in *Proceedings of 2011 7th International Workshop on Fibre and Optical Passive Components*, Montreal, Canada, 2011, pp. 1-4.
17. K. I. Funahashi, “Multilayer neural networks and Bayes decision theory,” *Neural Networks*, vol. 11, no. 2, pp. 209-213, 1998.
18. M. S. Hung, M. Y. Hu, M. S. Shanker, and B. E. Patuwo, “Estimating posterior probabilities in classification problems with neural networks,” *International Journal of Computational Intelligence and Organizations*, vol. 1, no. 1, pp. 49-60, 1996.
19. M. D. Richard and R. P. Lippmann, “Neural network classifiers estimate Bayesian a posteriori probabilities,” *Neural Computation*, vol. 3, no. 4, pp. 461-483, 1991.
20. D. E. Rumelhart, R. Durbin, R. Golden, and Y. Chauvin, “Backpropagation: the basic theory,” in *Backpropagation: Theory, Architectures and Applications*. Hillsdale, NJ: Lawrence Erlbaum Associates, 1995, pp. 1-34.
21. S. Miyake and F. Kanaya, “A neural network approach to a Bayesian statistical decision problem,” *IEEE Transactions on Neural Networks*, vol. 2, no. 5, pp. 538-540, 1991.
22. G. P. Zhang, “Neural networks for classification: a survey,” *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 30, no. 4, pp. 451-462, 2000.
23. T. J. 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, pp. 168-179, 2018.
24. Audio Engineering Society, “Standards in print,” 2020 [Online]. Available: https://www.aes.org/publications/standards/list.cfm.
25. T. J. O'shea and N. West, “Radio machine learning dataset generation with GNU Radio,” in *Proceedings of the 6th GNU Radio Conference*, Boulder, CO, 2016.
26. S. Ajala, E. Adetiba, M. B. Akanle, O. O. Obiyemi, S. Thakur, and J. Abolarinwa, “Experimentations on the transmit power of a universal software radio peripheral using GNU Radio framework and a handheld RF explorer,” *IOP Conference Series: Earth and Environmental Science*, vol. 655, no. 1, article no. 012006, 2021. https://doi.org/10.1088/1755-1315/655/1/012006
27. A. G. Ravelo-Garcia, J. L. Navarro-Mesa, E. Hernadez-Perez, S. Martin-Gonzalez, P. Quintana-Morales, I. Guerra-Moreno, and G. Julia-Serda, “Cepstrum feature selection for the classification of sleep apnea-hypopnea syndrome based on heart rate variability,” in *Proceedings of Computing in Cardiology Conference*, Zaragoza, Spain, 2013, pp. 959-962.
28. MathWorks, “Signal Processing Toolbox,” 2022 [Online]. Available: https://www.mathworks.com/help/signal/ref/rceps.html1.
29. A. Carballar and M. A. Muriel, “Phase reconstruction from reflectivity in fiber Bragg gratings,” *Journal of Lightwave Technology*, vol. 15, no. 8, pp. 1314-1322, 1997.
30. E. Adetiba, O. O. Olugbara, and T. B. Taiwo, “Identification of pathogenic viruses using genomic cepstral coefficients with radial basis function neural network,” in *Advances in Nature and Biologically Inspired Computing*. Cham, Switzerland: Springer, 2016, pp. 281-291.
31. E. Adetiba, V. C. Iweanya, S. I. Popoola, J. N. Adetiba, and C. Menon, “Automated detection of heart defects in athletes based on electrocardiography and artificial neural network,” *Cogent Engineering*, vol. 4, no. 1, article no. 1411220, 2017. https://doi.org/10.1080/23311916.2017.1411220
32. F. J. Olaloye and E. Adetiba, “Dynamic spectrum sensing with automatic modulation classification for a cognitive radio enabled NomadicBTS,” *Journal of Physics: Conference Series*, vol. 1378, no. 4, article no. 042092, 2019. https://doi.org/10.1088/1742-6596/1378/4/042092
33. E. Adetiba, V. O. Matthews, S. N. John, S. I. Popoola, and A. Abayomi, “NomadicBTS: evolving cellular communication networks with software-defined radio architecture and open-source technologies,” *Cogent Engineering*, vol. 5, no. 1, article no. 1507465, 2018. https://doi.org/10.1080/23311916.2018.1507465

Sunday Ajala https://orcid.org/0000-0002-4212-252X

Sunday Ajala received B.Sc. degree in Electronic and Electrical Engineering from the Obafemi Awolowo University, Ile-Ife, Nigeria in 2017. He is currently a Graduate Research Assistant in the Department of Engineering at Norfolk State University, Norfolk, Virginia, USA where he is also working towards his Master's degree in Electronics Engineering. His research interests are in microelectronics, point-of-care devices, microfluidics, and deep learning applications in CMOS and dielectrophoretic devices.

Emmanuel Adetiba (Member, IEEE) https://orcid.org/0000-0001-9227-7389

Emmanuel Adetiba received the Ph.D. degree in information and communication engineering from Covenant University, Ota, Nigeria. He is a registered engineer (REngr.) with the Council for the Regulation of Engineering in Nigeria (COREN) and a member of the Institute of Information Technology Professional (IITP), South Africa as well as the Institute of Electrical & Electronics Engineering (IEEE), USA. He has received several scholarly grants and research funds from reputable bodies such as World Bank, France Development Agency (AFD), US National Science Foundation, Durban University of Technology in South Africa, Nigeria Communications Commission, Rockefeller Foundation, International Medical Informatics Association (IMIA) and hosts of others. He has authored/co-authored about 100 scholarly publications in journals and conference proceedings some of which are indexed in Scopus/ISI/CPCI. His research interests include Machine Intelligence, Software Defined Radio, Cognitive Radio, Biomedical Signal Processing, Cloud Federation and EduTech. He was the Director at the Center for Systems and Information Services (ICT Center), Covenant University from 2017 to 2019. He is a Deputy Director at the Covenant Applied Informatics and Communication Africa Centre of Excellence (CApIC-ACE) and Co-PI for the FEDGEN cloud infrastructure research project at the center (World Bank funded). He is the founder and Principal Investigator of the Advanced Signal Processing and Machine Intelligence Research (ASPMIR) group. He is a full Professor and the incumbent Head of Department, Electrical & Information Engineering, Covenant University. He is also an honorary research associate at the Institute for System Sciences, Durban University of Technology, Durban, South Africa.

Oluwaseun T. Ajayi https://orcid.org/0000-0001-8659-6199

Oluwaseun T. Ajayi received the B.Sc. Telecommunication Science degree from the University of Ilorin, Ilorin, Nigeria in 2018. He is currently working toward a Ph.D. degree in the Department of Electrical and Computer Engineering, Illinois Institute of Technology, Chicago, USA. His research interests include machine learning in wireless networks, information freshness optimization, and internet of things (IoT) technology.

Abdultaofeek Abayomi https://orcid.org/0000-0003-3129-5246

Abdultaofeek Abayomi received the B.Sc. (Hons) Computer Science from the University of Ilorin, Nigeria. He later obtained Master of Technology in Computer Science from the Federal University of Technology, Akure. Nigeria. He has more than a decade stint with a financial solution service provider and an additional three years with an IT firm in Nigeria. He has lectured computer science courses at the Federal University Oye, Nigeria and obtained a Ph.D. degree in Information Technology at the Durban University of Technology, Durban. South Africa. His research interests are in artificial intelligence, machine learning and intelligence, wearable sensor technology, big data analytics, computer vision and image processing, bioinformatics, affective computing, human computer interaction (HCI), data mining among others. He is a member of the Institute of Information Technology Professionals, South Africa (IITPSA) and the South African Institute of Computer Scientists and Information Technologist (SAICSIT).

Anabi Hilary Kelechi

Anabi Hilary Kelechi received the B.Tech. degree equivalent in electrical, electronic and systems engineering from the Institute of Management Technology, Enugu, Nigeria, in 2003, the M.B.A. (Finance) degree from the University of Ballarat, Australia, in 2009, and a master's degree in wireless communication and the Ph.D. degree in electrical and electronic engineering from the Universiti Kebangsaan Malaysia in 2012 and 2017, respectively. From 2004 to 2007, he was with a private telecommunication firm in Nigeria. His research interest includes cognitive radio, resource allocation, prediction algorithm, fundamental study on TV white spaces, Internet of Things, and artificial intelligence.

Joke A. Badejo

Joke A. Badejo is a Senior Lecturer and the Coordinator of Computer Engineering Program in the Department of Electrical and Information Engineering at Covenant University, Nigeria. She is also a Faculty member of the Covenant Applied Informatics and Communication Africa Centre of Excellence (CApIC-ACE), a World Bank ACE-IMPACT Centre in Covenant University. She received her Ph.D. in Computer Engineering from Covenant University in 2015. Her broad research experiences and interests include biometrics & biomedical image analysis, machine (deep) learning, large-scale data analytics, software engineering, and cloud computing. She has been actively involved in the Covenant University Data Analytics Cluster (CUDAC), which supports data-driven decision making in the University. With more than a decade of computing research, teaching, and leadership experiences, she has published over 40 papers in reputable journals and conference proceedings. Joke is a member of several professional bodies including the Council for the Regulation of Engineering in Nigeria (COREN) and the Institute of Electrical and Electronics Engineers (IEEE). She enjoys being an academic and loves contributing impactful and cost-effective solutions to current societal engineering problems in Africa.

Sibusiso Moyo https://orcid.org/0000-0001-5613-7290

Sibusiso Moyo received her Ph.D. degree in Mathematics from the University of Natal (now University of KwaZulu-Natal), Durban South Africa in 2002. She joined the Durban University of Technology in 2001 and was Head of the Mathematics, Statistics, and Physics Department and then became the Director Research and Postgraduate Support in 2010, and substantive Deputy Vice-Chancellor Research, Innovation and Engagement in mid-2017. Her primary research focus areas are in mathematics, mathematical physics, differential equations, symmetry analysis, and selected inter-disciplinary topics and her secondary research area is in tertiary education management where she works on selected problems.

Murimo Bethel Mutanga

Murimo Bethel Mutanga received his undergraduate degree in 2005 from the Midlands State University in Zimbabwe. In addition, he holds a Master's and Ph.D. degrees in Computer Science from the University of Zululand. His research interest includes wireless communications, ubiquitous computing, and technologies of the 4th Industrial Revolution. He is a Senior Lecturer at Mangosuthu University of Technology in South Africa.