Physical Communication 25 (2017) 75–84

Contents lists available at ScienceDirect

Physical Communication

journal homepage: www.elsevier.com/locate/phycom

Full length article

Unsupervised feature learning and automatic modulation classification using deep learning model

Afan Ali *, Fan Yangyu

School of Electronics and Information, Northwestern Polytechnical University, 127 West Youyi road, Xian, China

ARTICLE INFO

Article history:
Received 2 February 2017
Received in revised form 29 July 2017
Accepted 5 September 2017
Available online 18 September 2017

Keywords:
Deep learning networks
Automatic modulation classification
Digital modulation
Autoencoders

ABSTRACT

Recently, deep learning has received a lot of attention in many machine learning applications for its superior classification performance in speech recognition, natural language understanding and image processing. However, it still lacks attention in automatic modulation classification (AMC) until now. Here, we introduce the application of deep learning in AMC. We propose a fully connected 2 layer feed-forward deep neural network (DNN) with layerwise unsupervised pretraining for the classification of digitally modulated signals in various channel conditions. The system uses independent autoencoders (AEs) for feature learning with multiple hidden nodes. Signal information from the received samples is extracted and preprocessed via I and Q components, and formed into training input to 1st AE layer. A probabilistic based method is employed at the output layer to detect the correct modulation signal. Simulation results show that a significant improvement can be achieved compared to the other conventional machine learning methods in the literature. Moreover, we also show that our proposed method can extract the features from cyclic-stationary data samples. A good classification accuracy was achieved, even when the proposed deep network is trained and tested at different SNRs. This shows the future potential of the deep learning model for application to AMC.

1. Introduction

Automatic modulation classification (AMC) identifies the modulation format of the received signal. It is a challenging task in the sense that the received signal is mostly corrupted by the noise and multipath fading. AMC took prominence mainly due to the applications in military, for example in signal interception of enemy's signal, electronic warfare deployment, and more recently, in the payloads for the unmanned aerial vehicles (UAV). Gradually, it also gained significance in the civilian applications, for example in cognitive radios, software defined radios (SDR) and adaptive modulation classification [1].

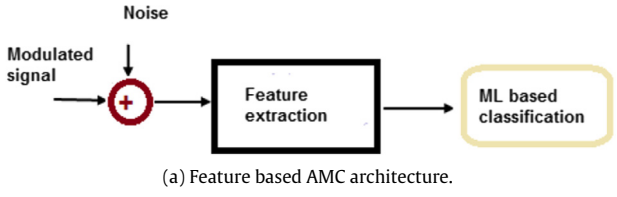
In literature, two main categories of AMC have been defined: likelihood based (LB) and feature based (FB) [2]. The LB AMC is a multiple composite hypothesis testing problem which computes the likelihood ratios of the selected received and known signals. Then, a decision is made by comparing this ratio to a threshold. This usually gives an optimal solution in the Bayesian sense but has a considerable complexity involved. More details on the likelihood based AMC can be found in [3–5]. The FB AMC is normally comprised of two subsystems: (1) feature extraction and (2) classification. Fig. 1(a) shows a common framework used for the AMC with these two subsystems. The FB classification employs certain features of the received modulated signal and a decision is made, by the classifier, based on the observed values of the received signal. The FB solution is usually sub optimal but it has a low complexity as compared to the LB methods and therefore is commonly used in the practical implementations [2,3].

In the FB approach, the feature extraction subsystem computes the distinct features from the received samples which are then translated into the classification parameters. For simple computations, these extracted features are kept as small as possible. Various features have been used in the literature for modulation classification. The two most widely used features are the high-order cumulants [6–11] and the features extracted from the time–frequency distribution which includes the wavelet transform [12–15].

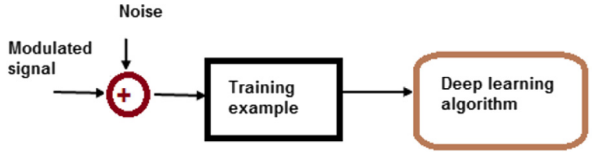
In [6], the cumulants up to sixth-order have been used in the classification of the MPSK and MQAM modulated signals. Likewise, authors used the fourth-order cumulants as the feature vector for the classification of digital modulations in [7,8]. In [9], authors have used the high-order cyclic-cumulants (CCs) of the received signals as the features for the modulation classification. More recently, authors have employed the multi-cumulants based classification to achieve the better results for multiple-input multiple-output (MIMO) communication [10,11]. Multiple cumulants in different orders are combined to form a multi-cumulant vector which is

* Corresponding author.
E-mail address: afanali@mail.nwpu.edu.cn (A. Ali).

http://dx.doi.org/10.1016/j.phycom.2017.09.004
1874-4907/



(a) Feature based AMC architecture.



(b) Proposed deep learning architecture.

Fig. 1. Conventional feature extraction based AMC Vs proposed deep learning based AMC.

then utilized for the classification of the overlapped signals received from more than one transmitter. The authors in these works claim that using multi-cumulant based feature vector yields a performance gain as compared to the single-cumulant based vector. Likewise, in another work, authors proposed a per-layer likelihood based classifier for a MIMO system by first using a subspace decomposition to decouple the transmitted streams [16]. The Stockwell transform (S-transform) based features extraction for the classification of different digitally modulated schemes has been used in [12]. In another work, the Haar Wavelet transform (HWT) of the received signals has been used to compare with a template and similarity between them is employed as a classification feature for the binary modulated signals [12]. A new wavelet cyclic feature (WCF) is proposed in [14] for the classification of the BPSK, QPSK, MSK and 2FSK. On the other hand, the continuous wavelet transform (CWT) has been used in [15] for the template matching and a satisfactory classification rate is achieved at SNR as low as −5 dB.

The classification subsystem of the FB approach identifies the correct target groups based on these distinct input features extracted from the data set. Different methods have been used for the classification subsystems in AMC. The hierarchical tree, distribution test and the machine learning (ML) based classifiers are some of the commonly used classification decision making subsystems [12,15,17–19]. Amongst them, the ML has been a well known choice for researchers over the years due to their superior performance. In [13,17,20] and [18] authors use the support vector machine (SVM) based classifier to identify different modulation schemes. The K-Nearest Neighbor (KNN) imputation technique is applied to deal with the missing data in the classification of radar signals in [21].

In the latest few years, increased demand in high data rate for the communication systems has resulted in extensive use of the high order modulation schemes like 16PSK, 16QAM, 64QAM, 128QAM and 256QAM. The classification of high order modulation schemes is more challenging as compared to the low order schemes due to the smaller distance between the constellation points. Moreover, larger input data set means, greater number of features for the ML based algorithm and hence increased computation complexity and time. To overcome this, we believe it is imperative to use the deep learning models for the ML in future. Deep learning is a branch of the ML which takes a large data as input, then construct this input data into multiple layered distributed network with hidden nodes at each layer. Finally, an output classifier layer provides the desired target classes [22–25]. Various areas like the visual recognition, object detection, tumor segmentation and speech recognition have already focused towards the deep networks [26,27]. In [26], authors have used the stacked auto-encoder (AE) network for the accurate brain tumor segmentation problem. Open source deep learning library, Chainer, has been used in [27] for chemoinformatics and bioinformatics problems. Likewise, other deep learning libraries that have surfaced recently are, Caffe [28,29] and the Torch [30].

However, to the best of our knowledge, very little work has been done on the application of the deep learning networks to the classification of digital modulated signals. In [31], authors have used the three layered deep neural networks (DNN) for the classification of BPSK, QPSK, 8PSK, 16QAM, and 64QAM with 21 statistical features. Encouraging results have been achieved in this work especially for the high Doppler fading channels. However, authors have not employed the constellation points of the modulation signal as the input training examples which would have been computationally more effective. We believe that the deep learning can provide a promising approach to the ML in AMC. In the deep networks, training examples can be provided as an input. This means that no features, such as high-order cumulants or wavelet-based, are required to be calculated for the purpose of AMC. The constellation points in, I and Q dimensions, can be used to collect signal information and directly formed into the input training examples. The suggested framework can be seen in Fig. 1(b).

In this work, our aim is to introduce the application of deep learning network in the area of AMC. With the exponential increase in computational capabilities of the modern intelligent receivers, the two main AMC applications that can utilize this approach are: military intelligence systems e.g. signal interception modules and radio spectrum sensing systems e.g. cognitive radios. Furthermore, the number of layers of the designed deep network based classifier can be customized to accommodate limited processing capabilities of the small devices that employs fast adaptive modulation schemes. We employ AE for the unsupervised feature learning [32–34]. Our study is mainly investigative, wherein, we test the output performance of the overall deep network by varying the design parameters of the AE. The parameters that give the best performance are then taken as the proposed design.

The rest of the paper is arranged as follows: Section 2 introduces the AEs, in general, and reviews the stacked sparse AE, applied in our system, in particular. The system model and modulation schemes used in our system are described in Section 3. Section 4 introduces our proposed stacked AE based deep network. Experiment results are discussed in Section 5, followed by discussion and conclusion.

2. Autoencoders

AE are the deep neural based learning networks [32,33,35]. They are comprised of three components: input, output and hidden layer. Each layer is trained to minimize the reconstruction error based on the cross-entropy [35]. Given the input $x \in \Re^d$, it first maps it into some hidden form, $f(x)$, by using an encoder function f. It then maps $f(x)$ back to its corresponding input, through the decoding function g, where g is called the reconstruction function r and is defined as $r(x) = g(f(x))$. The most common transfer function used for encoder and decoder is a logistic sigmoid function defined by the mathematical equation as $f(x) = \frac{1}{1+e^{-x}}$. Therefore, the first mapping can be defined by the equation $f(x) = sigm(Wx + b)$, where W and b are the weight matrix and the bias vector, respectively. We say that this mapping is parameterized by $\theta = \{W, b\}$. Likewise, this hidden representation is mapped back to reconstructed vector r by another mapping defined by $r(x) = sigm(W'x + b')$ with $\theta' = \{W', b'\}$. The overall aim of the design is

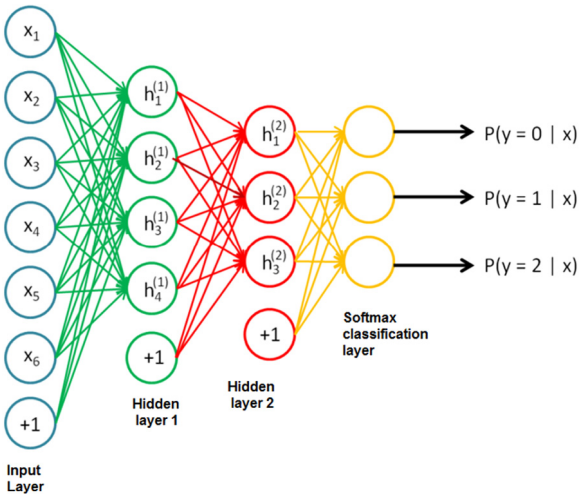


Fig. 2. 6-2-3 AutoEncoder.

to minimize the average reconstruction error by optimizing these parameters as follows [32,36]:

$$\underset{\theta,\theta'}{\operatorname{argmin}} \frac{1}{z} \sum_{i=1}^{z} L(x_i, r_i), \tag{1}$$

where L is a reconstruction cross-entropy and z is number of features. Normally, the stochastic gradient descent (SGD) is used for the purpose of optimizing these parameters.

2.1. Regularized AEs

In regularized AE, the encoding function f is not linear [32]. This means that f can take different directions at different input x, which allows the sparsity of the AE network [32,36], thus the sparse AE. So, the main aim here of the sparse AE is to apply sparsity to the reconstruction function r by minimizing the reconstruction error as follows:

$$L(x_i, r_i) + \beta \sum_{i=1}^{z} KL(\rho, \hat{\rho}_i), \tag{2}$$

where β is weight of sparsity penalty, ρ is average activation of f, $\hat{\rho}_i$ is average activation of ith input vector f_i.

KL is the Kullback–Leibler divergence [37] defined as:

$$KL(\rho \parallel \hat{\rho}_i) = \rho \log \frac{\rho}{\hat{\rho}_i} + (1-\rho) \log \frac{1-\rho}{1-\hat{\rho}_i}. \tag{3}$$

2.2. Stacked AEs

Normally, the deep learning network is formed by stacking two or more AEs together. Each AE undergoes the unsupervised training, where the output of the preceding AE is the input to the succeeding AE. Fig. 2 shows the 3 layered stacked AE network with 6 inputs, 2 hidden layers and 3 output nodes, hence 6-2-3 autoencoder. The last layer is called the softmax classifier layer, which classifies into the final target classes.

3. System model

Our selection of the modulation schemes for the classification in this work is based on the existing technologies. The wireless LAN standard and the OFDM systems uses a variety of different PSKs depending on the data rate required. Moreover, the communication systems designed to achieve very high levels of spectral efficiency usually employ very dense QAM constellations [2]. For example, 64QAM and 256QAM is often used in the digital cable television and cable modem applications. Our original data is a digitally modulated signal categorized in one of the following modulation schemes $I^{(a)}$, $1 < a < A$, where $A = 9$ is the number of modulation schemes: I = [2PAM, 4PAM, 8PAM, 2PSK, 4PSK, 8PSK,16QAM, 64QAM, 256QAM] which is defined in detail in Table 1.

Table 1
Modulation schemes in modulation pool $I^{(a)}$.

$I^{(1)}$ = 2PAM is 2-ary Pulse Amplitude Modulation.
$I^{(2)}$ = 4PAM is 4-ary Pulse Amplitude Modulation.
$I^{(3)}$ = 8PAM is 8-ary Pulse Amplitude Modulation.
$I^{(4)}$ = 2PSK is 2-ary Phase Shift Keying.
$I^{(5)}$ = 4PSK is 4-ary Phase Shift Keying.
$I^{(6)}$ = 8PSK is 8-ary Phase Shift Keying.
$I^{(7)}$ = 16QAM is 4-ary Quadrature Amplitude Modulation.
$I^{(8)}$ = 64QAM is 16-ary Quadrature Amplitude Modulation.
$I^{(9)}$ = 256QAM is 256-ary Quadrature Amplitude Modulation.

The system model assumes that the information is extracted from the received complex baseband signal which is as follows:

$$r(t) = x(t) + g(t) \tag{4}$$

where $r(t)$ is the received modulated baseband signal, $x(t)$ depends on modulation type and $g(t)$ is the additive noise.

The three expressions used for $x(t)$ are as follows [38]:

$$x_{PAM} = ARe \sum_{n} A_n g(t - nT_s), A_n = 2p - P - 1 \tag{5}$$

$$x_{PSK} = ARe \left[\sum_{n} C_n e^{j2\pi f_{ct}} g(t - nT_s) \right], C_n = e^{j2\pi \frac{l}{P}} \tag{6}$$

$$x_{QAM} = ARe \left[\sum_{n} C_n e^{j2\pi f_{ct}} g(t - nT_s) \right], \tag{7}$$
$$C_n = a_n + jb_n, a_n, b_n = 2p - P - 1$$

where x_{PAM} = PAM modulated received sequence, x_{PSK} = PSK modulated received sequence, x_{QAM} = QAM modulated received sequence, $p = 0, 1, \ldots, P-1$, A is the amplitude of the received signal, A_n and C_n map the transmitted symbols, n is the symbol index, T_s is the symbol period, f_c is the carrier frequency, P is the modulation order and $g(t)$ is the finite energy signal with a Ts duration. Although we assumed a single-input-single-output (SISO) system, it is worthy to mention here that this technique can be applied to MIMO systems as well with minor modifications in the system model. Moreover, we assume that the timing of the received symbol is perfectly recovered. Further channel effects like frequency offset, phase offset and flat fading will be covered in the next section.

3.1. Additive white Gaussian noise

First, we consider only the Additive white Gaussian noise (AWGN) with unknown channel gain. The signal model equation for the AWGN channel after matched filtering can be written as follows:

$$r(n) = \alpha \sum_{b=-\infty}^{\infty} x(l) h(nT - lT + \epsilon_T T) + g(n) \tag{8}$$

where α is the attenuation gain and $g(n)$ is the additive Gaussian noise, $h(\cdot)$ are the channel residual effect, ϵ_T is the timing error with symbol timing of T and $x(l)$ are the constellation points on the plane

I–Q. Here we test the case for both known and unknown Signal-to-Noise (SNR) at the receiver. To have a fair comparison with other AMC methods, we assume that the timing error has been recovered at the receiver. The SNR is defined as $SNR = 10log_{10}\frac{\alpha^2}{\sigma^2}$. Moreover, α is presumed to be unknown at the receiver. Therefore, to bring consistency on all signal models, observed signals are first normalized. The normalizing is done separately on I and Q components of the received samples as follows:

$$r_x(n) = \frac{r_x(n) - \mu_i}{\sigma_i} \tag{9}$$

$$r_y(n) = \frac{r_y(n) - \mu_q}{\sigma_q} \tag{10}$$

where $r_x(n)$ and $r_y(n)$ are I and Q components respectively, μ is the estimated signal mean and σ is the estimated standard deviation.

3.2. Flat fading channel

Next, we consider the flat fading channel for our signal model. Therefore, the frequency and phase offsets are added. The equation for the signal model after matched filtering can be written as:

$$r(n) = \alpha e^{j(2\pi f_0 n + \theta_0)} \sum_{b=-\infty}^{\infty} x(l)h(nT - lT + \epsilon_T T) + g(n) \tag{11}$$

where f_0 and θ_0 are the frequency and phase offset, respectively. We consider this channel as the unknown channel scenario. In our work, we create this scenario by generating the data at different SNRs to train our classifier.

4. Proposed deep learning model

4.1. 2-layer sparse AE

In this section, we describe the proposed deep learning network for the problem of AMC. The DNN using two independent AEs is proposed in this work as shown in Fig. 3. Each AE further consists of multiple hidden nodes. The two independent AEs are trained separately to generate the features I and II, respectively, and finally, stacked together to form the DNN, with the softmax classification layer as the final network layer. It is essentially a fully-connected feed-forward neural network with the layer-wise unsupervised pretraining. The motivation for using this approach in AMC has been taken from the work done in [26] where authors have implemented the three independent AE layer based neural network, stacked to form a deep network, for the brain segmentation problems, achieving high classification accuracy. More recently, authors in [31] have also employed independent stacked AE based deep networks in the problem of AMC to achieve very good performance.

4.2. Preprocessing

Signal information from the received complex samples, $r(n)$, is extracted by separating the real, $r_x^{(a)}(n)$ and imaginary, $r_y^{(a)}(n)$ components. $r_x^{(a)}(n)$ and $r_y^{(a)}(n)$ are then concatenated to form a high dimensionality training matrix, X. The formation of a training matrix in this way is shown in Fig. 4.

4.3. Building deep network

Our proposed deep network is built based on two main steps: (1) Learning phase of the independent AEs 1 and 2, respectively, and (2) fine-tuning of the 2-layer stacked AE.

4.3.1. Learning phase of independent AEs

The training example X, is given as the input, where $X \in \Re^d$. The encoding and decoding mapping is represented as follows:

$$v = sigm(WX + b) \tag{12}$$

$$\hat{X} = sigm(W'v + b') \tag{13}$$

where *sigm* is the sigmoid transfer function used for encoder and decoder, W and W' are the encoding and decoding weight matrices, b and b' are the encoding and decoding bias vectors.

Next, we put the output from the 1st AE layer as the input to the 2nd AE layer and train it using Eqs. (12) and (13). Training of the 1st and 2nd AE layer is unsupervised, i.e., it does not require the group labels of the modulation classes. We have used the SGD method for training the AE layers and the cost function used is an adjusted Mean Square Error (MSE) function defined as follows:

$$E = 1/z \sum_{i=1}^{z} \sum_{k=1}^{K} (\hat{X}_{ki} - X_{ki})^2 + \lambda * \Omega_{weights} + \beta * \Omega_{sparsity} \tag{14}$$

where λ is the coefficient for the L_2 regularization term for loss function and β is the coefficient for the sparsity regularization term.

$\Omega_{weights}$ and $\Omega_{sparsity}$ are known as the L_2 regularization terms applied to the weights of the cost function and sparsity term respectively, and are defined as follows:

$$\Omega_{weights} = \frac{1}{2} \sum_{s}^{L} \sum_{j}^{z} \sum_{i}^{m} (w_{ji}^{(s)})^2 \tag{15}$$

where L is the number of hidden layers and m is the number of variables in the training data.

$$\begin{aligned} \Omega_{sparsity} &= \sum_{i=1}^{z} KL(\rho \parallel \hat{\rho}_i) \\ &= \sum_{i=1}^{z} \rho \log \frac{\rho}{\hat{\rho}_i} + (1-\rho) \log \left(\frac{1-\rho}{1-\hat{\rho}_i} \right). \end{aligned} \tag{16}$$

Moreover, we varied the number of hidden nodes from 1 to 80 at this layer to observe the performance based on the reconstruction error. Subsequently, the problem of fixing the optimum number of hidden nodes is investigated based on observing the MSE curve achieved. It is also important to keep in mind that hidden nodes must be selected such that a balance is achieved between the output performance and the computational complexity of the overall system. The same procedure is repeated with the 2nd AE layer for its learning and fixing its number of hidden nodes.

4.3.2. Fine-tuning

After the learning phase of the two independent autoencoder layers, the supervised fine-tuning of the overall deep architecture is performed by stacking the two autoencoders and the softmax classification layers together. The optimum number of hidden nodes achieved during the learning phase, are set for each autoencoder layer.

The softmax classifier is trained with the maximum epochs of 1000 and cross entropy as the loss function. This step is a supervised training step and the group labels are given as the input along with the output from the 2nd AE layer. The goal of the unsupervised pre-training is to set the starting value of hidden nodes weights and biases such that it is better than random initialization, for a subsequent supervised training stage.

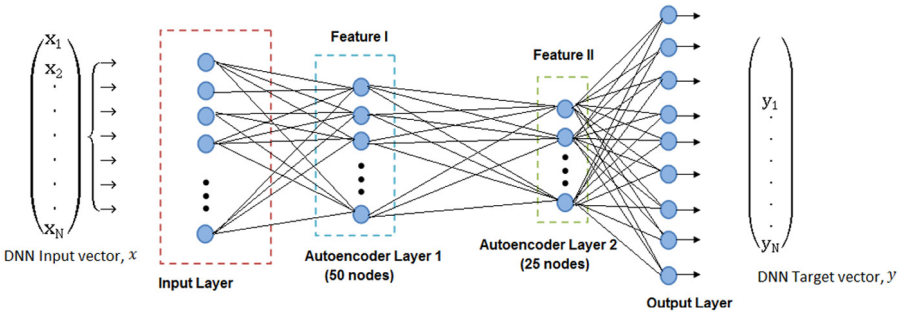


Fig. 3. Proposed deep learning network for AMC with two independent AE layers.

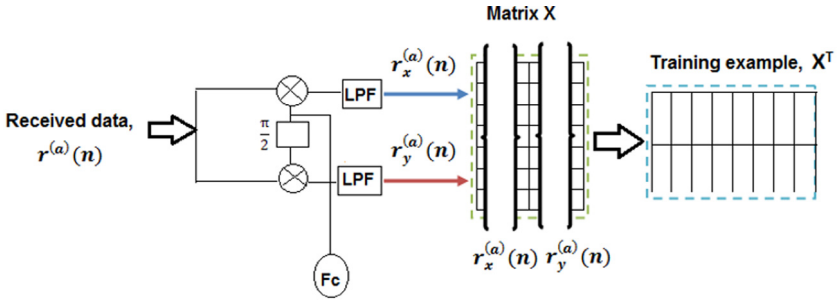


Fig. 4. Formation of training example.

Table 2
Operations needed for different classifiers.

Classifiers	Additions	Multiplications	Exponential	Logarithm
ML	$6IN\sum_{i=1}^{I}M_i$	$5IN\sum_{i=1}^{I}M_i$	$IN\sum_{i=1}^{I}M_i$	IN
Cumulants	$6IN$	$6IN$	0	0
GP-KNN	$6IN + RG + G$	$6IN + RG + G$	$RG + G$	$RG + G$
FNN	$INN_w\sum_{i=1}^{I}M_i$	$INN_w\sum_{i=1}^{I}M_i$	$IN\sum_{i=1}^{I}M_i$	0
DNN	$INN_w\sum_{i=1}^{I}M_i$	$INN_w\sum_{i=1}^{I}M_i$	$IN\sum_{i=1}^{I}M_i$	0

4.4. Complexity analysis

The computational complexity of the modulation classifier plays a significant role in the practical implementation. In some applications, the classification accuracy is the priority and the computational complexity can be compromised. On the other hand, for some time critical applications, classifiers with the low complexity are favored. In this section, we discuss the complexity of the proposed classifier and the classifiers considered in the simulation part. The mathematical operations needed to complete the classification of one piece of signal has been taken as the measure of the complexity for each classifier. This is depicted in Table 2. We make some assumptions to compute the number of operations as follows: (1) Total number of modulation candidates for classification are I, (2) there are M_i number of symbols in the ith modulation candidate, (3) signal samples are N, (4) the number of reference samples for GP-KNN classifier are R, (5) evolved features in GP contains G number of each operator and (6) total number of weights in the proposed DNN are N_w.

For the proposed DNN, there are two complexities involved: at training time and at the test time. At the training time, we must estimate the vectors, weight W and bias b, by solving the sigmoid function which requires an exponential operation, in addition to multiplication and summation. At the test phase, the prediction is linear in the number of features and constant in the size of the training data. Therefore, the complexity order of the proposed DNN can be expressed in the complexity orders of training and test phase as follows:

$$O = O_{train} + O_{test}. \tag{17}$$

4.4.1. Complexity involved in training stage

Our proposed DNN is comprised of 2 layers and 75 neurons (nodes) as described in Table 4. Hence, it has a depth of 2 and size of 75. In our experience, we have observed that taking a signal realization of 100,000 from each modulation scheme with a sample length of 512 results in a realizable training sample for the network.

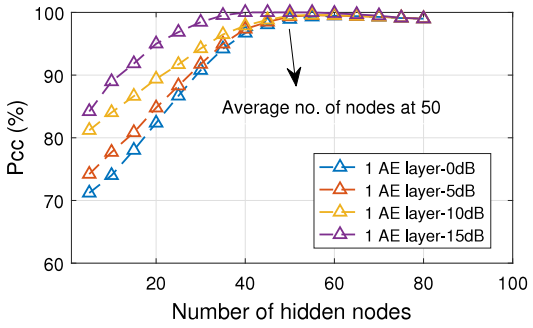
The complexity in the training phase requires $INN_w\sum_{i=1}^{I}M_i$ multiplications needed to compute the weights in the activation of all neurons (vector product) in the ith layer of the network. In addition, it also requires $IN\sum_{i=1}^{I}M_i$ exponential operations in sigmoid function. However, we are usually not worried about these computational complexities as training phase is an offline step.

4.4.2. Complexity involved in testing stage

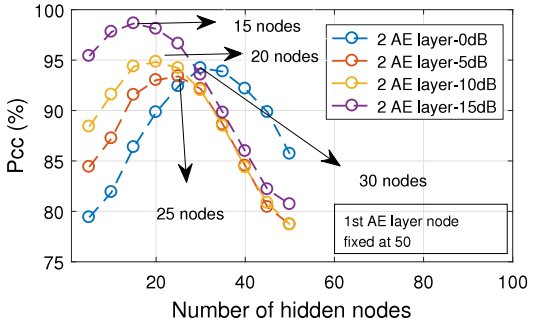
In testing phase, raw I and Q components are applied as an input to the DNN classifier. It is noted that the only complexity in this case is the multiplication of these features with the classifier weight computed in the training phase. Therefore, the network requires INN_w multiplications to compute the output in the testing phase.

4.4.3. Complexity comparison

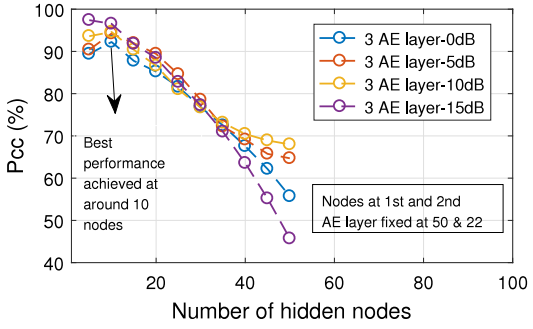
It is obvious that only the maximum likelihood (ML) based classifier requires exponential and logarithm operations while others do not. However, recent work in [16] used max-log-map approximation of the ML, which does not require the exponential operations, and therefore, reduces complexity considerably. The



(a) Classification accuracy vs. hidden nodes for AE layer 1.



(b) Classification accuracy vs. hidden nodes for AE layer 2.



(c) Classification accuracy vs. hidden nodes for AE layer 3.

Fig. 5. Optimum number of hidden nodes in each AE layer for the problem of AMC.

main complexity of Genetic programming-KNN (GP-KNN) is attributed to finding the final super features [39]. However, once the features are selected and combined, the training step does not need to be repeated for each testing signal. For the feed-forward neural network (FNN) classifier, the complexity is essentially the same as the proposed DNN method.

From the above discussion, we can say that cumulants-based classifiers have the lowest complexity, given that number of cumulants employed for classification are small. On the other hand, ML based classifiers have the highest complexities because they employ exponential and logarithmic operations, but their overall classification accuracy is higher. The complexity of our proposed DNN classifier mainly depends on the number of multiplication and exponential operations needed to compute the activation of all the neurons, which is mainly employed during the training phase of the network. Hence, we can say that the complexity is affected by how often training is required. However, it is worth clarifying here that, in our proposed DNN, training of the weights is done offline and is not repeated for classification in the testing phase. Finally, we can say that the more complex computations classifier performs, the higher classification accuracy it can achieve.

5. Experiment results and discussion

In this section, the performance of the proposed DNN is examined and compared to the other methods from the literature. All the simulations have been carried on a computer based environment. In the case where the frequency and phase offset is to be considered, native MATLAB function is used to implement the channel effects.

5.1. Optimum number of hidden nodes

We conducted this experiment to find the optimum number of nodes for each independent AE layer in our scenario. This was done in two steps: First we determined the optimum number of hidden nodes for 1st AE layer by varying the size of the hidden nodes from 1 to 80, and observing the classification performance at SNR value of 0, 5, 10 and 15 dB, respectively. The optimum number of hidden nodes was determined by taking the average value at these four SNRs. Secondly, we fixed the number of hidden nodes for 1st AE layer to the optimum value from step 1, and varied the hidden nodes for the consequent 2nd and 3rd AE layers. We used a signal sample length of 1000 samples for each modulation scheme. In order to populate the training example matrix with a considerable number of the observations from each modulation scheme, 15 simulation runs were carried to get 15 observations of I and Q component for each modulation scheme, respectively. Each I and Q observation from its corresponding simulation run is then saved to a separate row, hence a training example matrix of $\{30 \times 9000\}$, with rows representing a total of 30 observations of I and Q components concatenated together and columns representing 1000 values/modulations scheme of I and Q components. The results can be seen in Fig. 5. It can be seen from Fig. 5(a) that as the SNR is increased from 0 to 15 dB, less number of hidden nodes are required to achieve a better accuracy. For example, using 5 nodes at 0 dB and 15 dB SNR give an accuracy of around 71% and 84%, respectively. This was expected as higher SNR value means less distortion in I and Q values of each modulation scheme, hence better accuracy when using less nodes in the autoencoder layer. Consequently, the average number of optimum nodes at four different SNR values is about 50 nodes, meaning that this is an optimum value for hidden nodes for the 1st AE layer. The performance does not change after 50 nodes and, thus, increasing the nodes gives the same performance at the expense of increased complexity.

Next, Fig. 5(b) illustrates the four different performance curves for the experiment to find optimum number of hidden nodes for the 2nd layer. We fixed the hidden nodes for the 1st AE layer to the optimum value of 50 in this experiment. It can be seen from Fig. 5(b) that 30, 25, 20 and 15 hidden nodes are required to achieve the highest classification accuracy at 0, 5, 10 and 15 dB SNR, respectively. We averaged the number of nodes to find the optimum nodes value for the 2nd AE layer which comes to about 22 nodes. Likewise, similar experiment was repeated for the 3th AE layer. The 1st and 2nd AE layers were fixed to their optimum values of 50 and 22, respectively. The result is illustrated in Fig. 5(c). It is evident from the four curves that the performance achieved at all the four SNR values decreases after around 10 nodes. This is due to the fact that the model becomes excessively complex relative to the number of observations, with the increment in the number of nodes at the 3th AE layer. Therefore, the predictive performance of the DNN decreases due to overreaction to minor fluctuations in the training data.

Table 3
Performance comparison of 1, 2 and 3 independent layer DNN.

	Hidden nodes	$\hat{\rho}$	β	λ
DNN with 1 AE layer	50	0.8	4	0.003
DNN with 2 AE layers	50, 25	0.8, 0.5	4, 4	0.003, 0.004
DNN with 3 AE layers	50, 25, 10	0.8, 0.5, 0.1	4, 4, 4	0.003, 0.004, 0.004

Table 4
Configuration of proposed deep neural network.

Autoencoder	Hidden nodes	$\hat{\rho}$	β	λ	Max epoch
AE layer 1	50	0.8	4	0.003	800
AE layer 2	25	0.5	4	0.004	400
Softmax classifier	–	-	–	-	1000

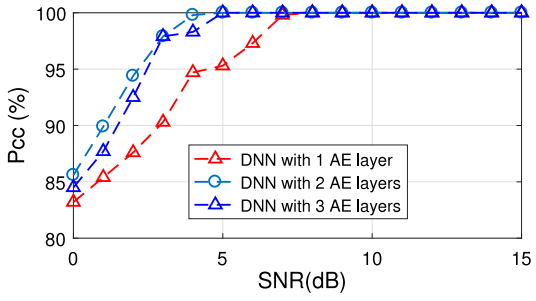


Fig. 6. Performance comparison of DNN with 1, 2 and 3 independent AE layers.

5.2. Performance comparison of 1, 2 and 3 independent AE layer DNN

In this experiment, we investigate the performance of the deep architecture with increasing the number of independent AE layers. We designed the DNN with 1, 2 and 3 independent AEs, thus increasing the deepness of the network from shallow to deep. The configuration parameters for each network is given in Table 3. We used the same signal sample length as in previous section. The result can be seen in Fig. 6. It is evident from the graph that the 2-layer DNN showed the best performance in our scenario, achieving 100% accuracy at an SNR of 4 dB and onwards. This implies that the 2 independent AE layers extract very robust and distinct features for classification in our scenario. As mentioned earlier, there is a relation between the number of input training data samples and the AE layers required for the feature extraction in the DNN. Here we strive to construct our network so that it gives a steady performance up to a wide input data samples. Hence, we choose this as our proposed network and test it under a range of different input data samples which will be discussed in later section. Moreover, as mentioned in previous section, the 3-layer network is an example of over-fitting case in our scenario. Too many parameters are involved with the addition of 3th AE layer relative to the input training example. On the other hand, the 1-layer AE network fits the example of under-fitting, where the model fails to capture the underlying trend of input data. The configuration parameters for our proposed DNN are shown in Table 4.

5.3. Classifiers used as benchmarks

The proposed DNN is compared to the four benchmark classifiers mainly: Maximum likelihood (ML) classifier [40], cumulant based Genetic programming (GP) and KNN classifiers [39] and feed-forward network (FNN) using cumulants and instantaneous power spectral density (PSD) as features [41]. In addition, we also performed an experiment to investigate the performance of our proposed technique using the spectral correlation function (SCF) of the input data samples as the training example. This is done to see if this method can extract good features from the cyclic-stationary data samples. We computed the SCF from the received data samples as was done in [42].

5.3.1. Maximum likelihood classifier

The ML classifier is used to set an upper bound of the classification performance. The decision for classification comes from the hypothesis, H_I, with an underlying constellation for the modulation candidate $I^{(a)}$ whose likelihood function $\bar{l}(H_I|r_N)$ is maximized, where r_N is the set of signal samples received. Therefore, it follows:

$$\hat{H}_I = \underset{H_I}{\operatorname{argmax}}\, \bar{l}(H_I|r_N). \tag{18}$$

We used the likelihood function used in [40] as follows:

$$\bar{l}(H_I|r_N) = \sum_{n=1}^{N}\left\{\frac{1}{I}\sum_{i=1}^{N_C} exp\left(-\frac{1}{2}\|r_n - A_i\|^2\right)\right\} \tag{19}$$

where N is the total samples, N_C are the total number of possible centroids and A_i being the actual values of these centroids. The estimated signal centroids are computed as [43].

$$\hat{A}_i = \sqrt{\frac{10^{\frac{SNR}{10}}}{1+10^{\frac{SNR}{10}}}}A_i. \tag{20}$$

5.3.2. Cumulants based GP-KNN

In this work, we have employed the GP-KNN used in [39] for the comparison. The cumulants used are C_{40}, C_{41}, C_{42}, C_{60}, C_{61}, C_{62} and C_{63}. Each of these cumulants are computed using the general equation as follows:

$$C_{ij} = cum\big(r(n), \ldots, r(n), r(n)^*, \ldots, r(n)^*\big). \tag{21}$$

5.3.3. Power spectral density based FNN

PSD based FNN classifier is the other method used in the experiments for the comparison with the proposed DNN [41]. The two PSD based features are: mean value of the amplitude of the signal samples received r_N and feature γ_{max}. Both these features are defined as follows:

$$X = \frac{1}{N}\sum_{n=1}^{N} A_n \tag{22}$$

where A_n is the instantaneous amplitude.

$$\gamma_{max} = max\frac{|DFT(a_{cn}(i))|^2}{N} \tag{23}$$

where N is the number of samples, a_{cn} is the value of normalized-centered instantaneous amplitude of the received signal.

5.4. Performance testing of the proposed technique

5.4.1. AWGN channel

We conducted two sets of experiments for AWGN channel model. In the first set of experiments, the focus was on the classification accuracy of the proposed DNN with varying SNR. We set the signal length at 512 samples and SNR was varied from 0 dB to 15 dB. Following this, signal realization of 100,000 from each modulation

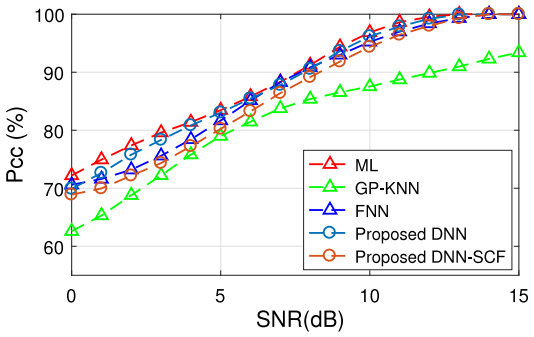


Fig. 7. Classification accuracy vs. SNR with 100,000 signal realizations with each modulation scheme consisting 512 samples.

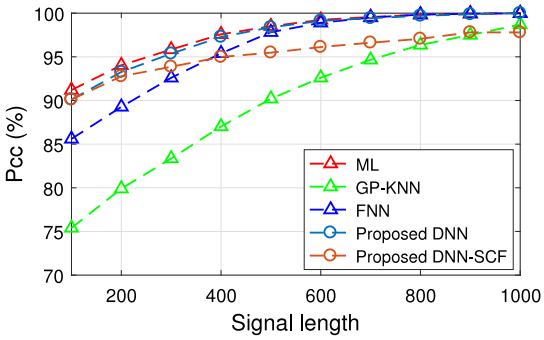
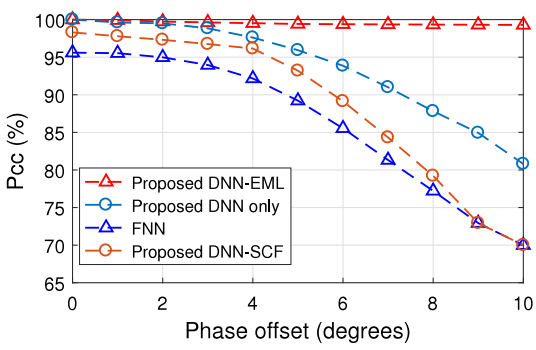
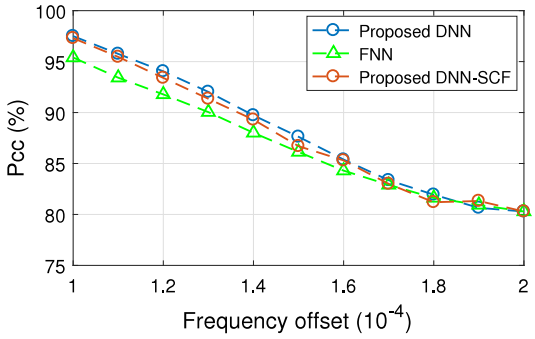


Fig. 8. Classification accuracy vs. signal length with 100,000 signal realizations from each modulation scheme with 512 sample under SNR 10 dB.



(a) Classification accuracy vs. phase offset with 100,000 signal realizations from each modulation scheme with 512 sample and 10 dB SNR.



(b) Classification accuracy vs. frequency offset with 100,000 signal realizations from each modulation scheme with 512 sample and 10 dB SNR.

Fig. 9. Performance of proposed technique under flat-fading channel.

scheme were tested using the ML, GP-KNN, FNN, the proposed DNN classifier with two input data: raw I and Q data samples, SCF based input data. The results are presented in Fig. 7. It is clear that the ML classifier achieves the best accuracy throughout the SNR range. Excluding the ML classifier, it can be seen that the proposed DNN classifier has an advantage over other classifiers at lower SNR. After 5 dB SNR, the proposed DNN almost achieves equal accuracy to that of the ML classifier. This proves the fact that, when trained with optimum parameter settings, the deep architecture can achieve a high accuracy due to its capability to extract distinct features from the incoming raw data. This can be further proved when comparing the performance of proposed DNN to FNN and DNN-SCF classifier. Hence, the proposed DNN takes in raw data samples to achieve a competitive performance to the other methods in literature. This is a promising result as no feature computation is done prior to giving data samples in our proposed technique.

In the second set of experiments, the focus was on the influence of signal sample length to the classification accuracy of the proposed DNN classifier. In this scenario, similar settings were used as before, except that SNR was fixed at 10 dB. Sample size was varied from 100 to 1000. The result can be seen in Fig. 8. Again, the ML classifier outperforms in all signal lengths. Excluding the ML classifier, the proposed DNN classifier and the DNN-SCF classifier achieves almost equal accuracy till a signal length of 200. However, the proposed DNN shows superior robustness when the signal length is in the range from 200 to 600. On the other hand, although less accurate than the DNN classifier, the performance of the FNN and GP-KNN classifiers shows a consistent improvement with increase in the signal length. This result proves the higher ability of the proposed DNN to show robustness in a limited signal length which is an important quality for a good AMC classification.

5.4.2. Flat-fading channel

In the flat-fading channel, we set the sample length to 512 and SNR to 10 dB. The signal realizations of 100,000 were tested for each modulation scheme. The phase and frequency offset are considered separately. For phase offset, we used a range from 0 to 10° with a step of 1°. We also considered Extended Maximum Likelihood (EML) estimator in [44] for preprocessing the signal to recover the phase offset. In the simulation, a relative frequency $\frac{f_0}{f}$ which is calculated from ratio between the actual frequency offset and the symbol sampling frequency is used to indicate different levels of frequency offsets. It is limited to 1×10^{-4} and 2×10^{-4}. As the ML and GP-KNN classifiers are not applicable under the phase and frequency offset channel conditions, only the FNN and DNN-SCF is compared with the proposed technique. The results are presented in Fig. 9. As expected, our proposed technique in combination of EML estimator, gives the best performance when the phase offset is considered (Fig. 9(a)). This is due to the ability of the EML estimator to recover the phase at the receiver, thus eliminating the channel mismatch. The performance of the proposed DNN without EML estimator also shows promising results. For example, the proposed DNN gives an equal accuracy to that of DNN-EML from 0° to 2°. Moreover, the proposed DNN starts with an advantage of about 2% and 3.2% over the DNN-SCF and FNN classifiers, respectively. As more phase offset is introduced, the performance of all the three classifiers start to degrades. However, this degradation in performance is lower for the DNN classifier as compared to the DNN-EML and FNN. One of the reason for this may be attributed to the fact that the FNN and DNN-SCF classifiers rely on an accurate signal model more than the proposed DNN.

On the other hand, in case when the frequency offset is considered, the performance of the proposed DNN and DNN-SCF drops

Table 5
Confusion matrix for 100,000 signal realizations from each modulation scheme of signal sample length 512, Data trained at 15 dB (94.5%).

	2PAM	4PAM	8PAM	2PSK	4PSK	8PSK	16QAM	64QAM	256QAM
2PAM	100	0	0	0	0	0	0	0	0
4PAM	0	100	0	0	0	0	0	0	0
8PAM	0	0	100	0	0	0	0	0	0
2PSK	0	0	0	100	0	0	0	0	0
4PSK	0	0	0	0	100	0	0	0	0
8PSK	0	0	0	0	0	100	0	0	0
16QAM	0	0	0	0	0	0	100	0	0
64QAM	0	0	0	0	0	0	1	95	4
256QAM	0	0	0	0	0	0	0	6	94

Table 6
Confusion matrix for 100,000 signal realizations from each modulation scheme of signal sample length 512 , Data trained at 0 dB (100%).

	2PAM	4PAM	8PAM	2PSK	4PSK	8PSK	16QAM	64QAM	256QAM
2PAM	100	0	0	0	0	0	0	0	0
4PAM	0	100	0	0	0	0	0	0	0
8PAM	0	0	100	0	0	0	0	0	0
2PSK	0	0	0	100	0	0	0	0	0
4PSK	0	0	0	0	100	0	0	0	0
8PSK	0	0	0	0	0	100	0	0	0
16QAM	0	0	0	0	0	0	100	0	0
64QAM	0	0	0	0	0	0	0	100	0
256QAM	0	0	0	0	0	0	0	0	100

significantly as shown in Fig. 9(b). There is a performance advantage of about 2% for the proposed DNN until a frequency offset of 1.6×10^{-4}. The performance becomes almost equal after this and drops to about 80% with a frequency offset of 2×10^{-4}. One of the main reasons for this performance degradation is the presence of the dense signal constellations like 64QAM and 256QAM, which leaves little space for any frequency offset. Some frequency offset estimation approaches have been developed [45] and can be used in the future work to achieve a higher accuracy under the required level of the frequency offset.

5.4.3. Unknown channel SNR

Finally, we test the robustness of our proposed deep network. This is done by training our data samples at different SNRs than the test data. We only consider the AWGN channel in this case for better understanding of the performance with unknown SNR at the receiver. Firstly, the training data samples are generated at two different SNRs of 0 and 15 dB, respectively. Then the test data is generated at a SNR in the range of −5,15 dB. The signal realization of 100,000 are used from each modulation scheme with the signal sample length of 512. The performance curves can be seen in Fig. 10. It can be observed from the figure that the performance curve of the data trained at 0 dB SNR has an advantage of around 20% over curve of the data trained at 15 dB. This can be attributed to the fact that the data generated to train and test the DNN at lower SNR have a close proximity. The gap between the two curves increases till 5 dB SNR, after which the performance becomes equal. It is also worth mentioning here that the classifier trained at lower SNR maintains a high accuracy even when the test data is generated at a high SNR of 15 dB. Confusion matrix of the correct classification for 9 modulation classes can be seen in Tables 5 and 6, for data trained at 15 and 0 dB SNR, respectively.

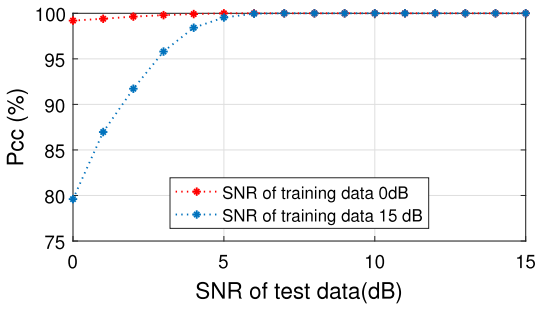


Fig. 10. Classification accuracy vs. unknown SNR conditions with 100,000 signal realizations from each modulation scheme with 512 sample.

6. Conclusion

A novel method for AMC has been introduced using a deep learning approach. As compared to conventional AMC methods used so far, using deep networks have advantages as follows: (1) No distinct features are needed for learning algorithm, complex modulated signals, separated by a filter into real and imaginary parts, can be given as input training examples to 1st AE layer, (2) AE based deep network can be adjusted with number of layers and nodes, depending on more heterogeneous training data sets, (3) Results show that even training the data set at different SNR, than the test data, gives good results.

In this work we proposed a 2 independent layers AE based DNN for the problem of AMC. We strived to find the optimum number of hidden nodes at each AE layer. Moreover, we also investigated the performance of a single and 3 independent layer DNN. It was generally observed that increasing the number of hidden nodes decreases the MSE in reconstruction of input at the decoder. Overall correct classification was measured using the softmax classifier layer at the output. This method can accommodate a range of modulation types, even those with very complex constellation diagrams (as in our case 256QAM).

To the best of our knowledge, there is very little work found in literature for modulation detection, using deep learning networks. This work shows that there is a lot of potential in using the applications of deep networks in modulation identification of digital signals.

References

[1] A.J. Goldsmith, S.G. Chua, Adaptive coded modulation for fading channels, IEEE Trans. Commun. 46 (5) (1998) 595–602.

[2] O.A. Dobre, A. Abdi, Y. Bar-Ness, A survey of automatic modulation classification techniques: Classical approaches and new trends, IET Commun. 1 (2007) 137–156.
[3] A.K. Nandi, E.E. Azzouz, Algorithms for automatic recognition of communication signals, IEEE Trans. Commun. 46 (1998) 431–436.
[4] W. Wei, J.M. Mendel, A new maximum-likelihood method for modulation classification, in: 29th Asilomar Conference on Signals, Systems and Computers, Vol. 2, 1995, pp. 1132–1136.
[5] T. Yucek, H. Arslan, A Novel Sub-optimum Maximum-Likelihood Modulation Classification Algorithm for Adaptive OFDM Systems, Vol. 2, IEEE Communications Society, 2004, pp. 739–744.
[6] A. Abdelmutalab, K. Assaleh, M. El-Tarhuni, Automatic modulation classification based on high order cumulants and hierarchical polynomial classifiers, Phys. Commun. 21 (2016) 10–18.
[7] D. Boiteau, C. Le Martret, Classification of linear modulation by mean of fourth-order cumulant, in: 8th European Signal Processing Conference, 1996, pp. 1–4.
[8] S. Norouzi, A. Jamshidi, A.R. Zolghadrasli, Adaptive modulation recognition based on the evolutionary algorithms, J. Appl. Soft. Comput. 43 (2016) 312–319.
[9] H. Sarieddeen, Z. Dawy, On the blind classification of parametric quadrature amplitude modulations, in: 3rd International Conference on Advances in Computational Tools for Engineering Applications (ACTEA), 2016, pp. 190–194.
[10] S. Huang, Z. Feng, Y. Zhang, K. Zhang, W. Li, Feature based modulation classification using multiple cumulants and antenna array, in: IEEE Wireless Communications and Networking Conference, 2016, pp. 1–6.
[11] S. Huang, Y. Yao, Z. Wei, Z. Feng, P. Zhang, Automatic modulation classification of overlapped sources using multiple cumulants, IEEE Trans. Veh. Technol. PP (99) (2016) 1.
[12] U. Satija, M. Mohanty, B. Ramkumar, Automatic modulation classification using S-transform based features, in: 2nd International Conference on Signal Processing and Integrated Networks (SPIN), 2015, pp. 708–712.
[13] S. Hassanpour, A.M. Pezeshk, F. Behnia, A robust algorithm based on wavelet transform for recognition of binary digital modulations, in: 38th International Conference on Telecommunications and Signal Processing (TSP), 2015, pp. 508–512.
[14] L. Zhou, H. Man, Wavelet Cyclic Feature Based Automatic Modulation Recognition Using Nonuniform Compressive Samples, in: IEEE 78th Vehicular Technology Conference (VTC Fall), 2013, pp. 1–6.
[15] K.M. Ho, C. Vaz, D.G. Daut, Automatic classification of amplitude, frequency, and phase shift keyed signals in the wavelet domain, in: IEEE Sarnoff Symposium, 2010, pp. 1–6.
[16] H. Sarieddeen, M.M. Mansour, A. Chehab, Modulation classification via subspace detection in MIMO systems, IEEE Commun. Lett. 21 (1) (2017) 64–67.
[17] W. Zhang, Automatic modulation classification based on statistical features and support vector machine, in: XXXIth URSI General Assembly and Scientific Symposium, 2014, pp. 1–4.
[18] T.N. AlOtaiby, M. Shoaib, A. Saleh, A. Hazza, Support vector machine based classifier for digital modulations in presence of HF noise, in: Saudi International Electronics, Communications and Photonics Conference, 2013, pp. 1–4.
[19] F. Wang, X. Wang, Fast and robust modulation classification via Kolmogorov–Smirnov test, IEEE Trans. Commun. 58 (8) (2010) 2324–2332.
[20] H. Zhang, G. Bi, S.G. Razul, C.M.S. See, Supervised modulation classification based on ambiguity function image and invariant moments, in: 9th IEEE Conference on Industrial Electronics and Applications, 2014, pp. 1461–1465.
[21] I. Jordanov, N. Petrov, A. Petrozziello, Supervised radar signal classification, in: International Joint Conference on Neural Networks, 2016, pp. 1464–1471.
[22] A. Alhamali, N. Salha, R. Morcel, M. Ezzeddine, O. Hamdan, H. Akkary, H. Hajj, FPGA-accelerated hadoop cluster for deep learning, in: IEEE International Conference on Data Mining Workshop (ICDMW), 2015, pp. 565–574.
[23] Y. Bengio, S. Bengio, Modeling high-dimensional discrete data with multi-layer neural networks, Proc. Adv. Neural Inf. Process. Syst. 12 (2000) 400–406.
[24] Y.M.A. Ranzato, L. Boureau, Y. LeCun, Sparse feature learning for deep belief networks, Proc. Adv. Neural Inf. Process. Syst. 20 (2007) 1185–1192.
[25] G. Hinton, Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups, IEEE Signal Process. 29 (6) (2012) 82–97.
[26] Z. Xiao, R. Huang, Y. Ding, T. Lan, R.F. Dong, Z. Qin, X. Zhang, W. Wang, A deep learning-based segmentation method for brain tumor in MR images, in: IEEE 6th International Conference on Computational Advances in Bio and Medical Sciences, 2016, pp. 1–6.
[27] Y. Kato, S. Hamada, H. Goto, Molecular activity prediction using deep learning software library, in: International Conference on Advanced Informatics: Concepts, Theory and Application, 2016, pp. 1–6.
[28] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, T. Darrell, Caffe: Convolutional architecture for fast feature embedding, 2014. ArXiv preprint arXiv:1408.5093.
[29] Google Brain team, TensorFlow: an open source software library for machine intelligence, Google Brain team within Google's Machine Intelligence Research Organization, 2015. https://www.tensorflow.org/.
[30] J. Murphy, Deep learning frameworks: A survey of tensorflow, torch, theano, caffe, neon, and the ibm machine learning stack, Microway Technol. (2013). https://www.microway.com/hpc-tech-tips/deep-learning-frameworks-survey-tensorflow-torch-theano-caffe-neon-ibm-machine-learning-stack/.
[31] B. Kim, J. Kim, H. Chae, D. Yoon, J.W. Choi, Deep neural network-based automatic modulation classification technique, in: International Conference on Information and Communication Technology Convergence, 2016, pp. 579–582.
[32] G. Alain, Y. Bengio, What regularized auto-encoders learn from the data generating distribution, J. Mach. Learn. Res. 15 (2012) 3743–3773.
[33] L. Deng, M. Seltzer, D. Yu, A. Acero, A.R. Mohamed, G. Hinton, Binary coding of speech spectrograms using a deep auto-encoder, Proc. Interspeech (2010) 1–21.
[34] Y. Bengio, Learning deep architectures for AI, Found. Trends Mach. Learn. 2 (1) (2009) 1–127.
[35] Y. Bengio, A. Courville, P. Vincent, Representation learning: A review and new perspectives. Technical report, 2012. arXiv:1206.5538.
[36] B.A. Olshausen, D.J. Field, Sparse coding with an overcomplete basis set: A strategy employed by V1, Vis. Res. 37 (1997) 3311–3325.
[37] S. Kullback, R.A. Leibler, On information and sufficiency, Ann. Math. Stat. 22 (1951) 79–86.
[38] J.G. Proakis, Digital Communications, McGrow-Hill, 1995.
[39] M.W. Aslam, Z. Zhu, A.K. Nandi, Automatic modulation classification using combination of genetic programming and KNN, IEEE Trans. Wirel. Commun. 11 (8) (2012) 2742–2750.
[40] W. Wei, J.M. Mendel, Maximum-likelihood classification for digital amplitude-phase modulations, IEEE Trans. Commun. 48 (2) (2000) 189–193.
[41] J.J. Popoola, R. van Olst, Automatic classification of combined analog and digital modulation schemes using feedforward neural network, in: AFRICON, 2011, pp. 1–6.
[42] G.J. Mendis, J. Wei, A. Madanayakei, Deep learning-based automated modulation classification for cognitive radio, in: IEEE International Conference on Communication Systems, 2016, pp. 1–6.
[43] Z. Zhu, M.W. Aslam, A.K. Nandi, Genetic algorithm optimized distribution sampling test for M-QAM modulation classification, Signal Process. 94 (2014) 267–277.
[44] V. Zarzoso, A.K. Nandi, Blind separation of independent sources for virtually any source probability density function, IEEE Trans. Signal Process. 47 (9) (1999) 2419–2432.
[45] E. Serpedin, A. Chevreuil, G.B. Giannakis, P. Loubaton, Blind channel and carrier frequency offset estimation using periodic modulation precoders, IEEE Trans. Signal Process. 48 (8) (2000) 2389–2405.

Afan Ali is a Ph.D. student in the School of Electronics and Information, Northwestern Polytechnical University. He received his Master's degree in Electronics and Communication Engineering from Australian National University, Australia in 2011. He completed his B.S. degree in Telecommunication Engineering from National University of Computer and Emerging Sciences, Pakistan in 2008. His research interests include pattern recognition, digital communications and signal processing.

Fan Yangyu is a Professor in Communication and Information Engineering Department at Northwestern Polytechnical University. He is also a Director of the Institute of Signal Processing and Wireless Optical Communications. He completed his PhD degree from North Western Polytechnical University in 1999. He completed his MSc. and BSc. from Shaanxi University of Science & Technology in 1992 and 1982 respectively. He is also Administrative Director of Shaanxi Society of Signal Processing. His research interest include Signal Processing, Wireless Optical Communications, Image Processing and Virtual Reality.