IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received March 17, 2021, accepted April 5, 2021, date of publication April 8, 2021, date of current version April 21, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3071801

Machine Learning Based Automatic Modulation Recognition for Wireless Communications: A Comprehensive Survey

BACHIR JDID[1], KAIS HASSAN[2], IYAD DAYOUB[3,4], (Senior Member, IEEE), WEI HONG LIM[1], (Senior Member, IEEE), AND MASTANEH MOKAYEF[1]

[1]Faculty of Engineering Technology and Built Environment, UCSI University, Kuala Lumpur 56000, Malaysia
[2]Laboratoire d'Acoustique de l'Université du Mans (LAUM) –UMR CNRS 6613, Le Mans University, 72085 Le Mans, France
[3]CNRS, Université Polytechnique Hauts-de-France, Université de Lille, ISEN, Centrale Lille, UMR 8520, Département d'Opto-Acousto-Électronique (DOAE), Institut d'Électronique de Microélectronique et de Nanotechnologie (IEMN), 59313 Valenciennes, France
[4]INSA des Hauts de France, 01100 Bellignat, France

Corresponding author: Iyad Dayoub (iyad.dayoub@uphf.fr)

This work was supported by the Ministry of Higher Education Malaysia through the Fundamental Research Grant Scheme under Grant Proj-FRGS/1/2019/TK04/UCSI/02/1 and Grant Proj-FRGS/1/2020/TK0/UCSI/02/4.

ABSTRACT The rapid development of information and wireless communication technologies together with the large increase in the number of end-users have made the radio spectrum more crowded than ever. Besides, providing a stable and reliable service is challenging, as electromagnetic environments are evolving and becoming more sophisticated. Accordingly, there is an urgent need for more reliable and intelligent communication systems that can improve the spectrum efficiency and the quality of service to provide agile management of network resources, so as to better meet the needs of future wireless users. Specifically, Automatic Modulation Recognition (AMR) plays an essential role in most intelligent communication systems especially with the emergence of Software Defined Radio (SDR). AMR is an indispensable task while performing spectrum sensing in Cognitive Radio (CR). Thanks to the significant advancements in Deep Learning (DL) applications, new and powerful tools have been provided which can tackle problems in this space. Thus, today, integrating DL models into AMR has gained the attention of many researchers. This work aims to provide a comprehensive state-of-the-art review of the most recent Machine Learning (ML) based AMR methods for Single-Input Single-Output (SISO) and Multiple-Input Multiple-Output (MIMO) systems. Furthermore, the architecture of each model will be identified along with a detailed comparison in terms of specifications and performance. Finally, an outline of the open problems, challenges, and potential research directions is provided along with discussion and conclusion.

INDEX TERMS Automatic modulation recognition, deep learning, machine learning, MIMO, SISO, wireless signal classification.

I. INTRODUCTION

AMR was initially motivated by its significance in many military applications such as electronic warfare, intelligence, surveillance, and threat analysis. It is a crucial task in intercepting the communications between adversary units and recovering the intercepted signal [1]. Moreover, from the standpoint of communication security, modulations can serve as another level of encryption. Specifically, an encryption key may be integrated into the received signal so that the AMR process can verify the legitimacy of the received signal to avoid emulation attacks and jam.

Today, the new generation of communication standards (5G & beyond) integrates multiple technologies of communication and information processing. This integration increases the cognition of intelligent communication systems and allows operators to reduce operational and maintenance costs plus attracts more users. An intelligent communication system is a system that can generate and implement new services quickly, easily, flexibly, economically, and efficiently [2].

The associate editor coordinating the review of this manuscript and approving it for publication was Stefan Schwarz.

It is recognized by its autonomy in making decisions based on its external conditions; and its ability of transmission environment cognition. In this direction, AMR can improve the awareness of such systems and provides interference detection and spectrum management. For instance, AMR is considered as a major task in CR which can provide dynamic radio resource management by employing re-configurable software defined transceivers. These transceivers can reconfigure their transmission parameters based on the available communication resources in the electromagnetic environment [3]. However, it is worth mentioning that the term 'automatic' is the opposite of the initial implementation of manual modulation recognition.

In modern communication systems, a pool of modulation types can be used by the transmitter to control both the data rate and the bandwidth usage. While the transmitter selects the modulation type adaptively, the receiving end may or may not know the modulation type. Correspondingly, the modulation information can be included in each signal frame so that the receiver would have the knowledge of the modulation type and react accordingly. However, the frequency spectrum is extremely limited and, thus, this strategy may not be efficient enough in real scenarios since it will affect the spectrum efficiency due to the extra information in each signal frame [4]–[7]. In fact, today's wireless networks are highly heterogeneous and the number of users is increasing significantly. Hence, AMR mechanism can be used to detect the modulation type of the received signals and, thus, eliminate any potential overhead in the network protocol. Ultimately, the signals will be correctly demodulated and the received data will be accurately recovered. Nowadays, the receivers enjoy a high computational power, especially with the advancements in microprocessors. Thus, the signal processing required in AMR becomes more feasible [4].

In literature, most of the existing AMR algorithms are typically implemented via two main approaches: Likelihood-Based (LB) and Feature-based (FB). On one hand, LB approaches calculate the likelihood functions of all candidate modulation schemes of the received signal and select the scheme with maximal likelihood value [3]. Several LB-based AMR techniques have been introduced for MIMO systems to blindly and semi-blindly identify the modulation scheme especially when the channel suffers from severe spatial correlation. Theoretically, LB approaches can provide the optimal solution but they suffer from an extremely high computational complexity and lack of robustness against the model mismatch. These undesirable drawbacks tend to restrict the feasibility of LB-AMR to be implemented in real-time and low-cost applications. On the other hand, FB approaches are considered to be a good alternative for LB approaches because they can produce suboptimal solutions with much lower computational complexity.

Nowadays, ML and DL applications have shown overwhelming advantages in wide-ranging fields such as computer vision, healthcare, robotics, and communication systems. Typically, ML algorithms need some guidance from expert engineers to make the required adjustments when these algorithms fail in the prediction process. On the contrary, DL networks can learn unsupervised from unlabelled or unstructured data, and make decisions without any human supervision. In fact, DL imitates the functionality of the human brain in processing data and creating patterns to be used in decisions making. However, integrating ML and DL in wireless communication networks and systems has great potential for their development since it can make them more intelligent. This suggests that integrating ML and DL approaches in AMR process can achieve significant results and boost its performance.

A. RELATED SURVEYS

In the literature, there are some brief surveys on DL-AMR[1] for SISO communication systems, e.g., [8] and [9]. For instance, the third section in [8] was dedicated to giving a brief overview of modulation recognition approaches based on the category of the extracted features. The authors focused mainly on briefing the AMR methods based on crafted expert features and developed ML classifiers. However, several DL-AMR models were also reviewed in terms of the employed features and classification criteria. Likewise, in [9], a brief review of DL-AMR techniques in terms of modulation pool are provided. The main DL models in the AMR methods were outlined along with a demonstration on the feasibility of using Convolutional Neural Network (CNN) to recognize wireless signals. Table 1 provides a comparison between the existing surveys and our paper.

B. SCOPE AND OBJECTIVE OF THE SURVEY

This survey aims to provide a comprehensive view on state-of-the-art ML and DL practices in AMR domain for SISO and MIMO wireless communication systems. In this survey, a detailed and fair comparison between the latest methods in this area is provided along with emphasizing their limitations and key advantages. Each method is presented in a clear and concise manner along with all of its aspects. For instance, the features extraction and classification criteria along with channel assumption and modulation sets are pointed out for ML-AMR methods. Furthermore, DL-AMR methods are compared in terms of network architecture, network training and testing parameters, used datasets, classification accuracy, and concluding remarks.

However, there are several architectural parameters that must be considered when designing a DL model. These parameters have critical impacts on designing or choosing the right DL model for solving the required AMR problem. For instance, whether the input data has a fixed-length or varying one. Besides, the depth of the network is a key parameter since it may simplify or complicate the task of modulation scheme classification. Equally, using an insufficient or badly constructed dataset can distort the hyperplanes of classifiers

[1]For simplicity, we refer to ML-based AMR and DL-based AMR by ML-AMR and DL-AMR, respectively.

TABLE 1. Summary and comparison of existing surveys related to ML-AMR and DL-AMR. The symbol ✓ indicates that a survey includes publications in the scope of a domain. The numbers indicate the count of reviewed publications in the scope of a domain. The symbol ✓ indicates that a survey compares publications in terms of this aspect. The symbol indicates that a survey did not review the scope of a domain.

Publication	One-sentence summary	ML-AMR	DL-AMR	SISO systems	MIMO systems	DL network structure	Training & testing configurations	Performance & accuracy	Input formats	Modulations pool
Zhou *et al.* [9]	A brief review of the most widely used DL techniques for recognizing a wireless signal in terms of modulation schemes	-	23	✔	✗	✗	✗	✗	✗	✓
Li *et al.* [8] (the 3rd section)	A brief overview of emerging ML and DL approaches for signal recognition	10	25	✔	✗	✗	✗	✗	✗	✗
This survey [8]	A comprehensive state-of-the-art review of the existing ML-AMR and DL-AMR methods for SISO and MIMO systems	19	75	✔	✔	✓	✓	✓	✓	✓

and, hence, affect badly its accuracy. Trainable parameters are very essential since they reflect the computational complexity and the required storage of the model and, thus, reveal the possibilities of implementing the model on real hardware [10]. Similarly, there are many other parameters to be considered during the model training process such as the batch normalization, activation functions, cost functions, and training configurations [10]. The training configurations include the number of batches, number of epochs, dropout, optimization and regularization algorithms applied during the training process. Ultimately, all of these parameters can affect the accuracy of the DL-based classifier and, thus, should be pointed out. To the best of our knowledge, a comprehensive survey of ML-AMR and DL-AMR methods considering these parameters does not exist in the literature. Moreover, few works are dedicated to reviewing the existing methods for both SISO and MIMO systems.

This article fills this gap by carrying a comprehensive up-to-date survey of researches in these domains. Beyond reviewing the most relevant literature, we discuss the feasibility of various DL architectures in view of solving AMR issues. At the end of this paper, we provide potential future research directions along with open challenges that need more investigation. Hence, our ultimate objective is to provide future affiliated researchers, who plan to take advantage of DL models to resolve problems in the field of AMR, with a definite guide that answers the following key questions:

- Which are the most recent ML and DL models in the field of AMR for MIMO and SISO systems?
- Why are ML and DL promising in AMR domain?
- What should be considered when designing a DL model for AMR?
- What are the most important and promising directions worthy of further investigation?

C. CONTRIBUTIONS OF THE SURVEY

The aforementioned surveys provide partial answers to some of these questions. This survey goes beyond the previous works and covers a wide range of DL models that have not been explicitly discussed in earlier surveys, e.g., [9]. Unlike such existing surveys, we also review the network architecture, accuracy against Signal-to-Noise Ratio (SNR) range, training parameters, and key concepts. We also review DL-AMR methods not looked at in other related surveys, including those for MIMO communication systems. While our main scope remains the DL models in AMR domain, for completeness we also discuss the most relevant ML models connected to AMR field. It is worth mentioning that we differentiate between DL-AMR, which refers to the models where the features are automatically extracted from data that has a complex structure and inner correlations, and ML-AMR, where features hand-crafting is required to be done by algorithms designed by expert engineers. **Hence, the main key perspectives which distinguish this paper from earlier surveys can be outlined as follows:**

- We provide an inclusive review of the existing DL models in AMR domain for both SISO and MIMO systems, instead of briefly discussing the main DL models or focusing on a single type of communication system, e.g. SISO systems [8], [9].
- We particularly discuss the most recent DL models from the perspective of AMR, focusing on their applicability to this area and their structures and parameters.

To the best of our knowledge, this is the first time that AMR is inclusively reviewed for both SISO and MIMO systems in wireless communications from a DL angle.

D. ORGANIZATION OF THE SURVEY

This paper is structured in a top-down manner as shown in Fig. 1. The outline of this paper can be summarised as:

- First, we begin, in section II, by presenting a basic background of the traditional process of FB-AMR followed by providing a basic ML and DL background to illustrate the main required concepts for the rest of this article. Thereafter, we discuss the main key factors that encourage the implementation of ML and DL models in AMR domain, aiming to clarify why such models can outperform the traditional AMR methods.
- Later in section III, a discussion on the integration of ML models into AMR methods is provided for SISO and MIMO systems.
- Since DL models require large-scale datasets, we begin section IV by presenting the specifications of some existing radio signals datasets. This provides the required information about the signal parameters and

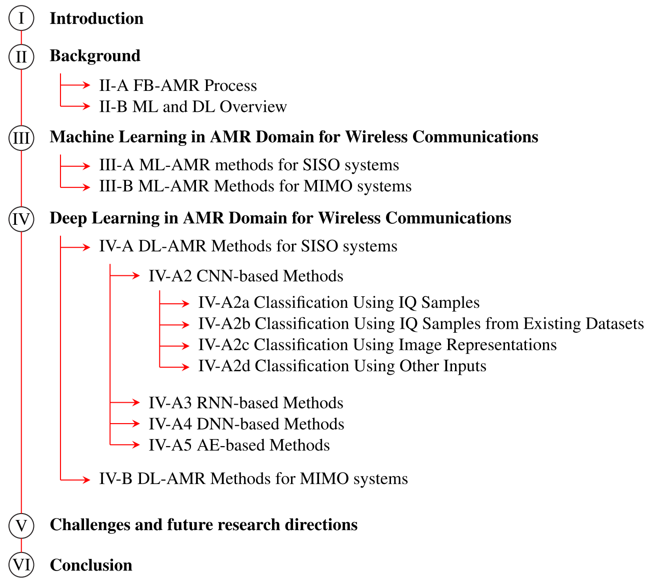

FIGURE 1. The overall organization of the paper.

channel characteristics of these datasets. Therefore, it will help to understand their usage in several of the presented DL-AMR methods. Then, we review the recent DL algorithms for AMR in SISO systems. We then review the existing DL-AMR methods for MIMO systems.

- Finally, we wrap up this article by providing open challenges and potential research directions along with concluding remarks in section V and in section VI, respectively.

In Fig. 1, a chart of the overall organization of the paper is provided. Furthermore, a list of main abbreviations is provided in Table 2.

II. BACKGROUND

In order to further illustrate the integration of ML and DL models into AMR process, we begin this section by providing an overview of a typical AMR process. Afterwards, we present some basic background information on ML and DL, and then we seal up this section by clarifying why such models are needed in AMR domain.

A. FB-AMR PROCESS

A typical AMR system consists of two subsystems: features extraction and classification subsystems. In the former subsystem, there are two phases: the pre-processing phase and the features selection phase. Fig. 2 shows a general block diagram of a features-based recognition AMR method. First, the signal will pass throw the pre-processing phase to determine several variables such as Carrier Frequency Offset (CFO), baud rate, Phase Offset (PO), SNR, and timing offsets. Next, the features will be extracted to

TABLE 2. List of abbreviations in alphabetical order.

Acronym	Explanation
ACGAN	Auxiliary Classification Generative Adversarial Network
AE	Auto-Encoder
AEN	Auto-Encoding Network
AI	Artificial Intelligence
AMR	Automatic Modulation Recognition
ANN	Artificial Neural Network
APoZ	Average Percentage of Zeros
AWGN	Additive White Gaussian Noise
BRNN	Bidirectional RNN
CAE	Convolutional AE
CCF	Cyclic Correntropy Function
CCNN	Cyclic CNN
CFO	Carrier Frequency Offset
CLDNN	Convolutional Long short-term Deep Neural Network
CNN	Convolutional Neural Network
CR	Cognitive Radio
CS	Cyclic Spectra
CSI	Channel State Information
CWD	Choi-Williams time-frequency Distribution
DBN	Deep Belief Network
DL	Deep Learning
DNN	Deep Neural Network
DPN	Dual Path Network
DRCN	Deep Reconstruction Classification Network
ELM	Extreme Learning Machine
FB	Feature-based
FLOCAF	Fractional Lower-Order Cyclic-Autocorrelation Function
FLOCS	Fractional Lower-Order Cyclic-Spectrum
FLOPs	Floating point Operations per Seconds
FPGA	Field-Programmable Gate Arrays
GCN	Graph Convolutional Network
GNU	GNU's Not Unix
HDNN	Hybrid DNN
HOC	Higher Order Cumulant
HOM	Higher Order Moment
IQ	In-Phase and Quadrature
LB	Likelihood-Based
LOS	Line-Of-Sight
LRN	Layered Resnet Network
LRR	Low-Rank Representation
LSTM	Long Short-Term Memory
M-ASK	M-ary Amplitude Shift Keying
MIMO	Multiple-Input Multiple-Output
ML	Machine Learning
MLP	Multilayer Perceptron
M-PSK	M-ary phase-shift keying
M-QAM	M-Ary Quadrature Amplitude Modulation
NiN	Network-in-Network
NLOS	Non-Line-Of-Sight
OFDM	Orthogonal Frequency-Division Multiplexing
OSNR	Optical SNR
PCA	Principle Component Analysis
PO	Phase Offset
PReLU	Parametric ReLU
PSO	Particle Swarm Optimization
RBM	Restricted Boltzmann Machine
RNN	Recurrent Neural Network
SCAE	Stacked Convolutional Auto-Encoder
SCF	Spectral Correlation Function
SELU	Scaled Exponential Linear Unit
SGD	Stochastic Gradient Descent
SISO	Single-Input Single-Output
SNR	Signal-to-Noise Ratio
SSAE	Stacked Sparse Auto-Encoder
SSCCF	Spatial Sign Cyclic Correlation Function
STFT	Short-Time Fourier transform
SVM	Support Vector Machine
SW-SCMA	Sliding Window Simplified Constant Modulus Algorithm
Tanh	hyperbolic tangent
TL	Transfer Learning
ZF	Zero-Forcing

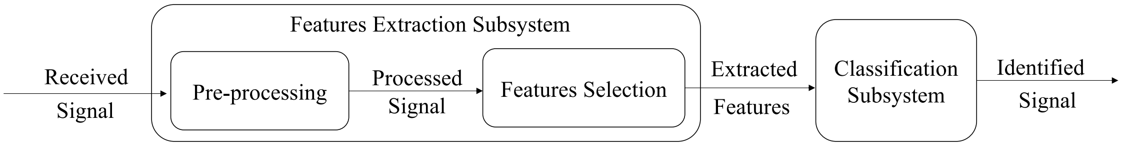

FIGURE 2. Block diagram of the Feature-based AMR method.

be then fed into the classification subsystem. Specifically, the extracted features can be categorized as follows: instantaneous time features [11], wavelet features [12], statistical features [13]–[15]. Ultimately, the decision of the modulation scheme is made by a specific classifier. Classifiers can be categorized into three main categories: (1) traditional classifiers such as classification tree, (2) ML-based classifiers, and (3) DL-based classifiers.

B. ML AND DL OVERVIEW

Fundamentally, ML is an area of Artificial Intelligence (AI) in which the algorithms can parse data, learn from them and apply whatever they learned to make informed decisions. In fact, DL is a sophisticated evolution subset of ML which gives machines the ability to make decisions without the intervention of humans. Technically, DL is ML and functions in a similar way but they have different capabilities and provide a different interpretation of the data they convey. Moreover, DL models can learn knowledge from raw data and decide if a prediction is accurate or not without any guidance. Furthermore, a key difference between traditional ML and DL is in the features extraction mechanism. ML methods apply the learning algorithms on hand-crafted engineering features, while in DL models, the features are learned automatically in multiple levels [10]. However, ML and DL approaches can be categorized as follows: Supervised, semi-supervised and unsupervised approaches [10]. Fig. 3 shows the Venn diagram of the relation between AI, ML, and DL.

1) DEEP NEURAL NETWORK (DNN) OVERVIEW

DL models are designed based on a specific layered structure of Artificial Neural Networks (ANNs). The idea behind the design of ANN is inspired by the biological neural network of the human brain. Fig. 4 shows the structure of a simple ANN. The first layer is called the input layer which receives the input data, while the rightmost one is the output layer which returns the output data. The layers between these two layers perform mathematical computations on the input data and they are called hidden layers since their values aren't observable in the training set. In general, an ANN with several hidden layers is referred to as a DNN and the more hidden layers it has, the deeper it is.

Each layer comprises several nodes called neurons. The neuron is the basic element of a neural network. It receives an input x, processes it, and generates an output. This output is either the final output or it will be sent to other neurons, through weighted connections, for further processing.

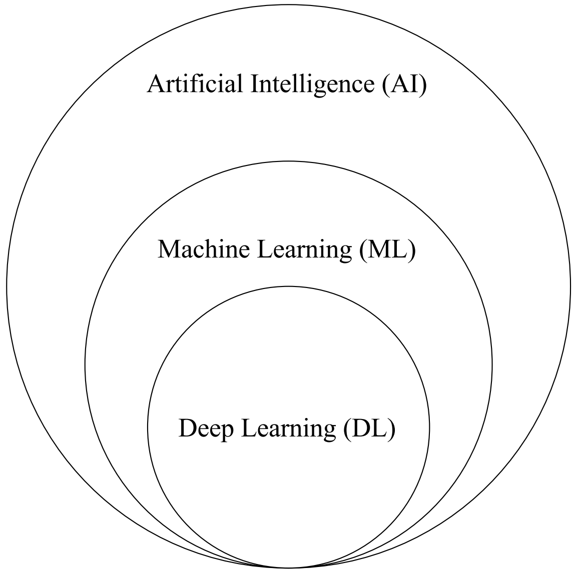

FIGURE 3. The taxonomy of AI as the broader umbrella for both ML and DL, and DL as a sub-branch of ML.

Each connection between these neurons is associated with a specific weight W_i. These weights are randomly initialized and are updated during the model training process. Each weight dictates the importance of the associated input value. Besides, another linear component, called the bias b_i, is applied to the input. The bias role is to change the range of the weight multiplied input. As can be seen, the final linear component is:

$$L_i = x_i \times W_i + b_i \tag{1}$$

Subsequently, a non-linear function, called activation function f, is applied to the combination of all linear components of a specific neuron. Thereafter, the output y_i will be look like:

$$y_i = f\left(\sum_i L_i\right) = f\left(\sum_i x_i W_i + b_i\right) \tag{2}$$

As shown in Table 3, there are several activation functions such as Sigmoid, hyperbolic tangent (*Tanh*), Rectified Linear Unit (*ReLU*) [16], Parametric ReLU (*PReLU*), Scaled Exponential Linear Unit (*SELU*) [17] and *Softmax* [10], [18].

However, the main aim when training a DNN is to optimize the prediction accuracy and, thus to minimize the prediction error. Therefore, a cost or loss function should be used for measuring this error and react accordingly. Besides, the neural network should be trained on random equal-sized

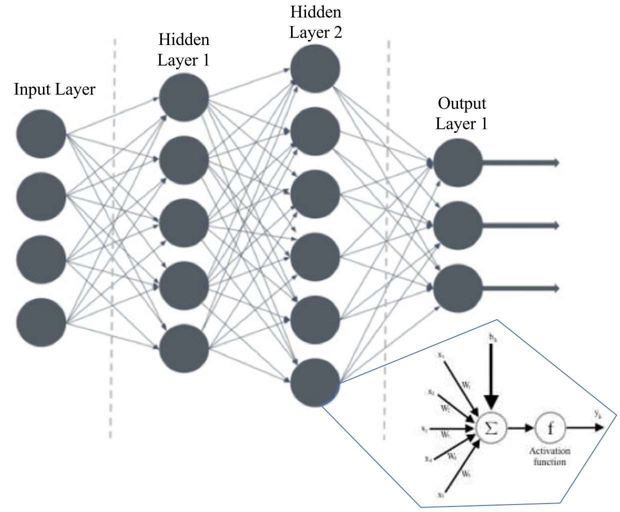

FIGURE 4. A simple Artificial Neural Network (ANN).

TABLE 3. The mathematical representations of activation functions.

Name	Equation
$sigmoid$	$\text{sigmoid}(\mathbf{x}) = \frac{1}{1+e^{-\mathbf{x}}}$ (3)
$ReLU$	$\text{ReLu}(x) = \begin{cases} x & x > 0 \\ 0 & x \leq 0 \end{cases}$ (4)
$PReLU$	$\text{PReLu}\,(x) = \begin{cases} x & x > 0 \\ ax & x \leq 0 \end{cases}$ (5)
$Tanh$	$\tanh(\mathbf{x}) = \frac{e^{\mathbf{x}} - e^{-\mathbf{x}}}{e^{\mathbf{x}} + e^{-\mathbf{x}}}$ (6)
$SELU$	$\text{SELU}(\mathbf{x}) = \lambda \begin{cases} \mathbf{x}, & \text{if } \mathbf{x} > 0 \\ \alpha e^{\mathbf{x}} - \alpha, & \text{if } \mathbf{x} \leq 0 \end{cases}$ (7) where α and λ are pre-defined constants
$Softmax$	$\text{softmax}\,(\mathbf{x}_i) = \frac{e^{\mathbf{x}_i}}{\sum_{j=0}^{k} e^{\mathbf{x}_k}}$ (8)

batches which makes the model more generalized than sending the whole input in one go. Moreover, optimization and regularization techniques can be adopted such as dropout regularization and gradient descent optimization algorithms. It is worth mentioning that Multi-Layer Perceptron (MLP) is always feed-forward, while DNN can have loops.

2) CNN OVERVIEW

CNN is a DL algorithm that comprises several types of layers, and has several advantages over DNNs. CNN is much more like the human visual processing system, especially when it comes to multi-dimensional inputs (e.g. 2D and 3D images). Most of all, the parameters of CNNs are significantly fewer than a fully connected network of similar size, and suffer less from the diminishing gradient problem. Fig. 5 shows an example of CNN structure. Mainly, the architecture of CNN consists of a combination of several types of layers. These layers can be summarized as follows [10], [18]:

- Convolutional Layer: This layer is responsible for convolving feature maps from previous layers with learnable kernels. Thereafter, a linear or non-linear activation function is applied to the output of these kernels to generate an activation map as its output.
- Rectification Layer: This layer performs element-wise absolute value operation on the input value.
- Pooling Layer: This layer takes the activation map generated by the non-linearity layer, and performs a down sampling operation in order to reduce its size.
- Fully Connected or Dense Layer: This layer is a MLP. It is responsible for mapping the activation maps from previous layers into a class probability distribution.
- Dropout Layer: This layer is practically used to prevent over-fitting in the training process by using Dropout along with other techniques such as L2 Regularization.
- Softmax Layer: This layer uses Softmax function in order to computes the score of each class over all possible classes. This function is usually used in the classification layer.

However, the total number of learnable parameters is an important metric to measure the complexity and the required memory of a CNN model.

3) RECURRENT NEURAL NETWORK (RNN) OVERVIEW

RNN is simply a generalized feed-forward ANN that has a memory to store the output it has already learned from the previous input. As shown in Fig. 6, RNN performs the same function for every input with a consideration of the last computed value. However, traditional ANNs approaches, DNNs, and CNNs deal only with fixed-length input. On the contrary, RNNs can handle a sequence of vectors over time. Moreover, a modified version of RNN, called Long Short-Term Memory (LSTM), is able to not only solve the problem of vanishing gradient, but also remember past data easily. LSTM is trained using back-propagation and can classify a time series given time lags of unknown duration. It is worth mentioning that there are several modified versions of RNN in the literature such as convolutional LSTM [18].

4) AUTO-ENCODER (AE) OVERVIEW

AE is practically a DNN approach which can learn features unsupervised based on efficient data encoding and decoding. As shown in Fig. 7, the inputs are encoded and mapped in the lower dimensional features space with a constructive feature representation. Thereafter, the same process is repeated till the desired feature dimensional space is reached. In the

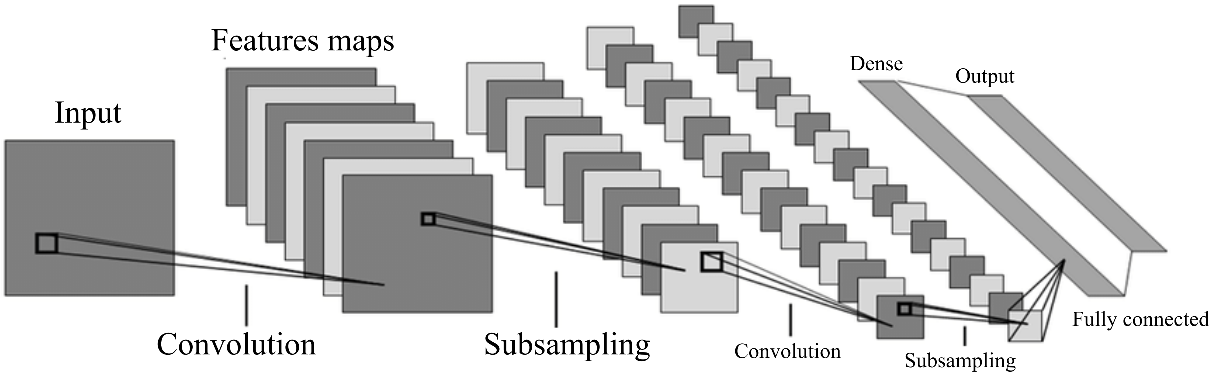

FIGURE 5. Structure example of CNN.

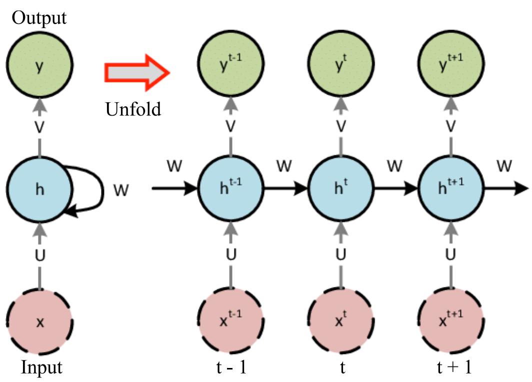

FIGURE 6. Structure example of RNN.

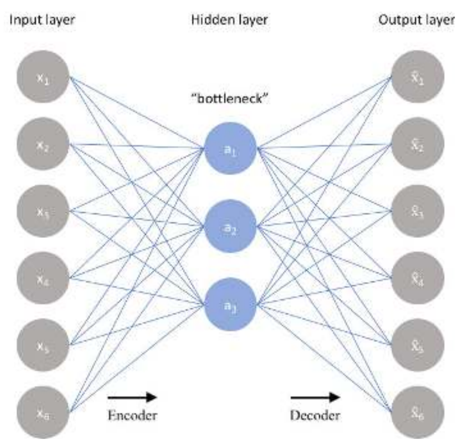

FIGURE 7. Structure example of AE.

decoding phase, the actual features are obtained from lower dimensional ones with reverse processing. Several kinds of AE exist in the literature such as variational AEs and split-brain AEs [10], [18].

5) RESTRICTED BOLTZMANN MACHINE (RBM) OVERVIEW

A RBM is also trained in an unsupervised manner. RBM consists of a visible layer, a hidden layer, and weighted connections between them. Each neuron in RBM is a stochastic binary unit. The neurons in the visible layer are mutually independent given the hidden ones, and *vice versa* [18]. As in a simple feed forward ANN, the visible layer receives the input data, and the neurons of the hidden layer are fully connected to all neurons in the visible layer. However, the state of the neurons in the hidden layer can affect those in the visible one, and *vice versa*.

The architecture of a Deep Belief Network (DBN) [18] can be formed by stacking RBMs or AEs. The process in a DBN consists of two main stages: (1) pre-training stage and (2) fine-tuning stage. The former stage is performed in each layer of RBM in order to initialize the parameters of the DBN. In the latter stage, regression algorithms such as softmax regression can be applied to provide error prediction and, hence, optimize the parameters by back-propagation algorithm. Fig. 8 shows the general structure of a DBN.

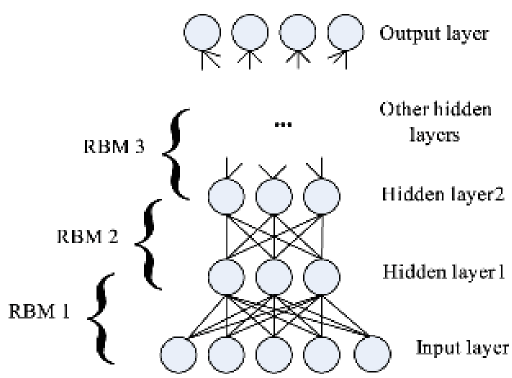

FIGURE 8. General structure of DBN.

a: THE NEED OF ML AND DL IN FB-AMR

The performance of the conventional FB-AMR methods heavily depends on the extracted features from the received signals and requires the supervision of expert engineers. Moreover, a set of different features has to be built manually in order to recognize different modulation schemes. Besides, conventional FB methods require building a classification tree with manual thresholds in order to decide on the modulation scheme. In such methods, the AMR process may need the knowledge of SNR value especially in the case of wide range SNR applications, then SNR estimation is also required. However, the hand-crafted features may have different values under varying noise conditions and, thus, different classification trees should be built or feature selection algorithms should be designed to build a robust set of hand-crafted features [19].

That is why ML and DL approaches have been developed to reduce tedious threshold operations and to improve the overall performance. DL approaches can eliminate the SNR estimation requirement since they can learn automatically optimal features, under varying noise regimes, without any need for precisely designed features. Moreover, DL models can be extended to several applications by adopting the transfer learning approach for example. Recently, many researchers proposed incorporating ML and DL models into FB-AMR methods to explore different types of features. This integration allows an automated features extraction process by employing different architectures of DNNs. However, self-learned features can be more relevant and, hence, achieve better performance.

III. MACHINE LEARNING IN AMR DOMAIN FOR WIRELESS COMMUNICATIONS

ML is a subset of AI that exploits statistical learning algorithms for designing and implementing intelligent systems. Many existing works employed ML classifiers in order to improve the classification accuracy of AMR for both SISO and MIMO systems. Numerous researches were proposed based on ANNs [15], [20]–[22], Extreme Learning Machine (ELM) [23]–[25] as well as Support Vector Machine (SVM) [26], [27]. In this section, the presented ML classifiers are considered to be different from those in the DL subset. First, we review the recent ML-AMR methods in SISO systems. Then, we discuss the most relevant ML models in AMR for MIMO systems. Finally, we conclude this section with a brief discussion and learned lessons.

A. ML-AMR METHODS FOR SISO SYSTEMS

In [24], an ELM-based AMR method was proposed by using one dimension local binary pattern [28] for features extraction. The classification accuracy was investigated under known channel states where the PO and CFO were recovered prior to modulation classification. Moreover, the overall performance was studied under several receiver conditions such as the symbols number, timing offset, PO, impulsive noise, and Doppler shift. In [25], Gabor filter and Cuckoo search algorithm were employed for features extraction and optimization, respectively. However, the decision on the modulation scheme was made by ELM classifier. In [26], a low-complexity method for AMR and SNR estimation was introduced over multipath fading channels. Principal Component Analysis (PCA) was adopted for features extraction, while two different types of SVMs were used for classification and SNR estimation. In particular, three models of support vector classifiers were trained to differentiate a modulation type from the others. K-means clustering was used in [27] to extract the required features to feed the SVM classifier. The authors in [29] presented a compressive sensing AMR method by employing the higher-order cyclic cumulants and SVM classifier. SNR variations were considered in [19] in order to design a robust SVM-based AMR method. Hence, the normalized-centered variance was used to select the noise-insensitive features from a large set of extracted features. Besides, attribute reduction based on rough set theory was applied to remove all possible redundant features.

Wireless communication systems may be subject to non-Gaussian interference and impulsive noise. Therefore, the communication channel can be better modeled by heavy-tailed distributions, such as the non-Gaussian α-stable one. However, there are several promising solutions for AMR in the presence of impulsive non-Gaussian noise such as the Fractional Lower-Order Cyclic-Spectrum (FLOCS), the Spatial Sign Cyclic Correlation Function (SSCCF), the Fractional Lower-Order Cyclic Autocorrelation Function (FLOCAF), and the Cyclic Correntropy Function (CCF). The authors in [30] employed the graph-based FLOCS analysis in the α-stable impulsive noise environment for AMR. Firstly, the FLOCS of the received signal was obtained by the transformation of its fractional lower-order moments. Then, the modulation scheme was identified using the graph-based AMR mechanism by employing the extracted features from the adjacency matrices corresponding to the graph representation of FLOCS. In comparison, FLOCAF and CCF were adopted in [31], and a mathematical expression demonstrating that SSCCF is a particular case of the FLOCAF was derived. Besides, a detailed analysis of the cyclic spectrum of BPSK, QPSK, 8QAM, 16QAM, and 32QAM signals obtained by the FLOCAF and CCF was provided. Moreover, both FLOCAF and CCF were found to allow the symbol rate parameter estimation, where the latter was found to be more efficient.

B. ML-AMR METHODS FOR MIMO SYSTEMS

The ANN-based AMR algorithm in [20] was proposed for MIMO systems, and it was then modified in [21] by considering the time variation in the channel over an observation interval. A sliding window simplified constant modulus algorithm (SW-SCMA) was used during the blind channel estimation phase. The work reported in [21] considered a more realistic scenario because the channel matrix is unknown and blindly estimated by SCMA without any channel compensation. In [22], three AMR algorithms were presented. In the first one, the modulation schemes were directly recognized without any channel equalization. The second algorithm assumes that the channel matrix is perfectly known at the receiver side and it uses the Zero-Forcing (ZF) equalization to recover the transmitted signal symbols before features extraction. In the last one, the receiver has no prior knowledge of the channel matrix and it is blindly estimated by SCMA likewise in [20]. It is worth mentioning that the final decision was made using a majority rule by fusing all individual decisions.

In [23], two ELM algorithms were introduced for multipath fading channel without any prior knowledge of the channel and signal parameters. In spite of the good performance and the low complexity of ELM algorithm, it has some disadvantages. For instance, the number of the hidden nodes is fixed and their parameters are randomly generated to remain unaltered during the training phase. These disadvantages were handled by an algorithm called self-adaptive evolutionary ELM. The authors in [32] proposed a novel cepstrum-based pre-processing algorithm and a logarithmic functional fitting method. The main aim was to eliminate the multipath fading effects in order to improve the performance of AMR. An impartiality comparison study in [33] was introduced between four classifiers (classification tree, K-nearest neighbours, ANN, and SVM) in FB-AMR in terms of classification accuracy and computational complexity over MIMO channels. This study considered Higher Order Cumulants (HOCs) up to 6th-order, and results revealed that the ANN classifiers have the best performance/complexity trade-off for the M-ary phase-shift keying (M-PSK), M-ary Amplitude Shift Keying (M-ASK), and M-Ary Quadrature Amplitude Modulation (M-QAM) modulations.

The work presented in [34] was devoted to resolving the issue of lower recognition rate of HOC-based AMR in space division multiplexing MIMO system. Auto-Encoding Network (AEN) and ANN were used for data dimension reduction and classification, respectively. Tian *et al.* [35] proposed an ML approach for AMR based on a modulation-constrained clustering with unknown channel matrix and noise variance. The problem of modulation classification was converted into clustering one without direct channel estimation. However, the maximum likelihood criterion was used in order to obtain the final decision on the modulation scheme. The main core was called centroid reconstruction, in which the cluster centroids were reconstructed with fewer parameters. The model in [36] was introduced for signal detection in the Orthogonal Frequency-Division Multiplexing (OFDM) system. AEN and ELM were, respectively, exploited for features extraction and classification. Table 4 summarizes the extracted features, decision-maker, antennas number, noise nature, channel, required estimation, and the recognized modulation schemes of the previously discussed ML-AMR.

1) DISCUSSION AND LEARNED LESSONS

Although ML-AMR algorithms outperform the conventional methods [22], yet they still need a robust complex set of hand-crafted features to achieve a good performance. Further, these algorithms work well under relatively ideal conditions. In such methods, some pre-processing techniques have to be conducted on the received signals in order to perform some parameters estimation before AMR to enhance the classification accuracy [24]. However, ML algorithms enjoy low implementation cost with high performance on small data [26]. Nevertheless, they are time demanding and can be affected badly by the curse of dimensionality.

IV. DEEP LEARNING IN AMR DOMAIN FOR WIRELESS COMMUNICATIONS

DL is a powerful technique that can be integrated into FB-AMR to provide a high classification accuracy along with high efficiency and robustness. It is worth noting that DNNs are deeper than simple ANNs since they consist of several layers between the input and output layers. Therefore, the more layers they have, the deeper they are. In this section, we begin by reviewing the most recent DL-AMR methods for SISO systems. Then, we discuss the existing DL models for AMR in MIMO systems. Finally, we wrap up this section with a brief discussion and learned lessons.

A. DL-AMR METHODS FOR SISO SYSTEMS

We categorize the DL-AMR methods in this subsection based on the employed deep network models, which simplify the comparison among them. First, we outline some of the exiting signal datasets in the literature. Second, the methods that exploit CNN are discussed. Then, those using RNN are explored. Next, DNN-based AMR methods are summarized. Thereafter, we review the methods proposed for AMR based on AE models.

1) RELATED SIGNALS DATASETS

Signals dataset are particularly required for training, validating, and testing the networks in ML-AMR and DL-AMR models. Some researchers used their own simulated datasets, and others prefer to employ those introduced in the literature. Some of the datasets are presented as follows.

a: RadioML 2016.10A DATASET

In [37], a survey on the emerging applications of ML in radio signal processing domain was proposed along with a synthetic dataset. It was generated with GNU's Not Unix (GNU) Radio and it includes eight digital modulations and three analog ones. 220k signals for 20 different SNRs were generated and divided as 1k signals per modulation per SNR. In fact, it represents a cleaner and more normalized version of the RadioML 2016.04C dataset [38]. However, there is a larger version of this dataset, called RadioML 2016.10B.

b: RadioML 2018.01A DATASET

This dataset is one of the most challenging datasets of modulation classification which was presented in [39].[2] It includes over-the-air measurements of 24 digital and analog modulation schemes spreading in a wide range of SNR values. Moreover, it contains more than 2.5M signals with synthetic simulated channel effects.

c: HisarMod2019.1 DATASET

In [40], 5 modulation families passing through 5 different wireless channels (ideal, static, Rayleigh, Rician, and Nakagami–m) were considered in the generated dataset. This dataset was generated using MATLAB2017. It includes 780k

[2] A discussion on [39] will be provided later in this paper, in section IV.

TABLE 4. Summary of ML-AMR method for SISO and MIMO systems.

Paper	Features	Classifier	Antennas number	Channel	Modulation Pool	Remarks
[20]	HOMs up to eighth-order and HOCs up to sixth-order	ANNs	$N_t \times N_r \in$ {2×6, 3×6}	Frequency-flat time-varying MIMO	{BPSK, QPSK, 8PSK} and {BPSK, QPSK, 8PSK, 4ASK, 8ASK, 16QAM}	•Required estimation: Channel estimation. •SNR range: [-10, 10] dB. •Recognition accuracy with mobility: >97% for SNR>-5dB.
[21]	HOMs up to eighth-order and HOCs up to sixth-order	ANNs	$N_t \times N_r \in$ {2×6, 3×6}	Frequency-flat time-selective MIMO	{BPSK, QPSK, 8PSK} and {BPSK, QPSK, 8PSK, 4ASK, 8ASK, 16QAM, 64QAM}	•Required estimation: Channel estimation: Only for SCMA-DMI. •SNR range: [-5, 15] dB.
[22]	HOMs and HOCs up to sixth-order	ANNs	N_t = 2, N_r = 4	Frequency-flat, spatially correlated block fading MIMO	{BPSK, QPSK, 8PSK} and {BPSK, QPSK, 8PSK, 4ASK, 8ASK, 16QAM, 64QAM}	•Required estimation: Channel estimation: Only with ZF-ELM and ZF-Sa-ELM. •SNR range: [-5, 15] dB.
[23]	HOMs and HOCs up to sixth-order	Two ELMs	$N_t \in$ {2,4}, N_r = 4	Time invariant and frequency flat MIMO	{BPSK, QPSK, 8PSK} and {16QAM, 64QAM}	•Required estimation: CFO and PO in the first case of known channel scenarios . •SNR range: [0, 20] dB. •Recognition accuracy for 2x4 antennas: ≈95% for SNR>4dB.
[24]	LBP histogram features	ELM	N_t = 1, N_r = 1	Additive White Gaussian Noise (AWGN)	{BPSK, QPSK, 8-PSK, 16- QAM, 64-QAM and 4-ASK}	•Required estimation: CFO and PO in the first case of known channel scenarios . •SNR range: [-10, 10] dB. •Recognition accuracy for 2048 symbols: ≈95% for SNR>-2dB.
[25]	shift c , scale σ , modulation parameters f and Weight w	ELM	N_t = 1, N_r = 1	AWGN and Rayleigh fading	{QPSK, 16PSK, 64PSK, BFSK, 4FSK, 16FSK, QAM,16QAM, 64QAM}	•SNR range: [-10, 5] dB. •Recognition accuracy for SNR=0dB: ≈99.7% and ≈100% at 512 samples and 1024 samples, respectively.
[26]	Histogram features	SVM	N_t = 1, N_r = 1	Multipath fading	{2ASK, QPSK and 16QAM}	•Required estimation: SNR Estimation. •SNR range: [0, 30] dB. •Recognition accuracy: ≈99.83% by employing all features.
[27]	characteristic parameters: T2, T4, T8, T16, T32	SVM	N_t = 1, N_r = 1	AWGN	{2PSK, 4PSK, 8PSK and 16QAM, 32QAM and 64QAM}	•SNR range: [-2, 10] dB. •Recognition accuracy: ≈59% for SNR>-2dB.
[29]	Higher-order cyclic cumulants (CCs)	SVM	N_t = 1, N_r = 1	Multipath fading channels	{4QAM,16QAM,64QAM} and {8ASK,16QAM, 64QAM, BPSK}	•SNR range: [5, 15] dB. •Recognition accuracy: >90% for SNR>10dB.
[19]	8 instantaneous feature, 7 HOCs, 5 wavelet features, and 5 cyclostationary features	SVM	N_t = 1, N_r = 1	AWGN Channel	{2ASK, 4ASK, 8ASK, 16QAM, 2FSK, 4FSK, 8FSK, 2PSK, 4PSK, 8PSK}	•SNR range: [0, 20] dB. •Recognition accuracy: ≈95% for SNR>5dB and ≈97% for SNR>10dB.

of In-Phase and Quadrature (IQ) samples spreading in the same SNR range of the aforementioned RadioML.2016.10A.

Table 5 summarizes the modulation schemes, covered SNR range, dataset's size, and the synthetic simulated channel effects of the datasets.

2) CNN-BASED METHODS

For a better understanding of the existing CNN-based AMR methods, they are assorted based on the adopted signal representation as follows.

a: CLASSIFICATION USING IQ SAMPLES

IQ samples convey the changes in magnitude and phase of a wireless signal, and they are often used in RF applications. Many researchers simply used their own simulated IQ signals as the input of CNN. Shi *et al.* [41] conducted some pre-processing techniques and SNR estimation in order to design a CNN-based AMR method. This method aims at eliminating the bad effects of PO in uncooperative OFDM systems and, therefore, achieving a high classification accuracy. The algorithm in [6] can learn to extract features automatically from the long symbol-rate signals but still requires high SNR to achieve high classification accuracy. A unit classifier was applied to deal with the changes in input dimensions. While two-step training method was adopted to overcome the problem of direct training, transfer learning was applied to improve the retraining efficiency. A multi-stream structure was used in [42] to increase the network width and obtain more valuable features. Several superposition convolutional units were used in each stream to overcome the problem of network training and overfitting. Moreover, a non-linear function approximator called Network-in-Network (NiN) was exploited to deal with the problem of the non-linearity in the extracted feature. Gu *et al.* [43] studied both blind channel identification and AMR. Two CNNs were used to remedy the flaws of

TABLE 5. Summary of dataset used in the literature.

Dataset	Modulation Pool	SNR range	Size	Frame length	Included effects
[39]	Normal Classes: OOK, 4ASK, BPSK, QPSK, 8PSK, 16QAM, AM-SSB-SC, AM-DSB-SC, FM, GMSK, OQPSK Difficult Classes: OOK, 4ASK, 8ASK, BPSK, QPSK, 8PSK, 16PSK, 32PSK, 16APSK, 32APSK, 64APSK, 128APSK, 16QAM, 32QAM, 64QAM, 128QAM, 256QAM, AM-SSB-WC, AM-SSB-SC, AM-DSB-WC, AM-DSB-SC, FM, GMSK, OQPSK	[-20,+30]	2.5M	1024	CFO, symbol rate offset, multipath fading, and thermal noise
[37]	BPSK, QPSK, 8PSK, 16QAM, 64QAM, GFSK, CPFSK, 4PAM, WB-FM, AM-SSB, and AM-DSB	[-20,+18]	220k	128	CFO, sample rate offset, AWGN, and fading
[40]	BPSK, QPSK, 8PSK, 16PSK, 32PSK, 64PSK, 4QAM, 8QAM, 16QAM, 32QAM, 64QAM, 128QAM, 256QAM, 4PAM, 8PAM, 16PAM, 2FSK, 4FSK, 8FSK, 16FSK, FM, PM, AM-LSB, AM–USB, AM–SC, and AM–DSB	[-20,+18]	780k	1024	multipath fading with different number of channel taps

classification on the same channel. The first CNN was responsible for identifying the channel whether it is a Line-Of-Sight (LOS) or Non-Line-Of-Sight (NLOS) channel. The second one was used to classify the modulation schemes under the identified channel. The work in [44] was devoted to making a comparison study between CNNs, RNNs, inception modules, and Convolutional Long short-term DNNs (CLDNN). Results revealed that CLDNN outperforms other networks for SNRs above -8dB.

Wang *et al.* [45], applied two CNNs and dropout instead of pooling operation in order to achieve high classification accuracy. On one hand, the first CNN, named DrCNN, was responsible for classifying the adopted modulation schemes (BPSK, QPSK, 8PSK, GFSK, CPFSK, PAM4) in addition to separate 16QAM and 64QAM schemes from others. On the other hand, the second CNN aimed at classifying 16QAM and 64QAM schemes. The authors created two datasets to be used in AMR. The first dataset was created by IQ samples in order to train the former CNN, while the second one adopted constellation diagrams with a density window to train the latter CNN. Simulation results demonstrated a considerable improvement when compared to the benchmark methods.

CNN and LSTM were employed in [46] for modulation scheme classification. The method in [47] was proved to have higher robustness against SNR variation and less memory consuming than the benchmark methods. Similarly, the algorithm in [48] was proposed for varying noise regimes with less computational complexity and likewise smaller model sizes. In particular, the redundant neurons were pruned through a compressive sensing based neuron pruning technology. Simulations showed that the proposed method not only reduces the computational time but also diminishes the required device memories under an acceptable performance loss. In their work, Zheng *et al.* [49] aimed to solve the problem of variable CNN's input size and to make full use of the complete signal burst. Thus, three fusion methods were explored: voting-based fusion, confidence-based fusion, and feature-based fusion. First, the signal is divided into multiple segments of the same length. Then, each segment is sent to the CNN to make a decision by the corresponding fusion method. Finally, all results will be fused based on classification results, confidence, or intermediate features of these segmented signals. Simulations revealed that all of the proposed fusion methods overcome the non-fusion method. In [50], a graph mapping CNN was employed to convert the received signals into graphs. While CNN was used for feature extraction, Graph Convolutional Network (GCN) was employed for the classification part. The dataset is divided into multiple subsets for the sake of converting modulated signals into graphs. Each subset consists of 40 labeled signals and one unlabelled one.

b: CLASSIFICATION USING IQ SAMPLES FROM EXISTING DATASETS

Many researchers chose to use one of the existing IQ signals datasets[3] in order to form the input of CNN. In [38], the adaptation of CNN to the dataset in [37] was studied along with a performance comparison between the proposed CNN, and those of the expert HOM-based methods. Then in [39], the authors extended their prior work on using deep CNNs for AMR [38], [44], [51]. They investigated the performance of DL-AMR considering a rigorous baseline method using HOMs and strongly boosted gradient tree classification. Besides, the effects of a wide range of design parameters, channel impairment conditions, and training dataset parameters were studied. They conducted several simulations including over-the-air measurement of AMR performance. Moreover, they built an improved version of the tool described in [37] to be used in dataset generation[3].

In [52], a signal distortion correction module was proposed to improve the accuracy of CNN-based AMR. The proposed module employed an ANN in order to estimate both CFO and PO of received signals. In contrast to [41], this module eliminates the effects of both CFO and PO by shifting the signal frequency and phase prior to modulation recognition. The work reported in [53] exploited the attention mechanism for the fusion of the extracted multi-scale features. The SNR was used for optimizing the categorical cross-entropy loss with correct weight to improve the performance and the convergence time of the network. The network structure in [54] consists of six convolutional blocks (denoted M-block). Each one of these M-blocks includes three convolutional layers of different kernels. The output of every single block is the concatenation of all feature maps in the depth dimension. In order to improve the AMR accuracy, a Cyclic CNN CCNN) and a Bidirectional RNN (BRNN) were employed in [7].

[3]A discussion on these datasets was provided in section IV-A1.

The former network was devoted to extracting spatial features, while the latter meant to obtain temporal ones. Besides, the designed framework used an attention mechanism in order to improve the efficiency of the received signal features. In addition, an Auxiliary Classification Generative Adversarial Network (ACGAN) was proposed to expand the training data set. PCA procedure was used in [55] to create a set of uncorrelated features and, therefore, reduce the features vector length required for AMR.

In [40], HisarMod2019.1 dataset[3] was used to evaluate the proposed CNN-based AMR method. The modulation type of each signal was firstly identified, then the signals were classified based on modulation order. The model in [56] was dedicated for beyond 5th-generation communication systems. Therefore, the main objective was to reduce the computing time below 0.01 millisecond to comply with the future communications standards. In [57], the signals were identified whether they are wideband frequency modulation or not by using two expert features: (1) the maximum value of the power spectral density of the normalized-centered instantaneous amplitude, and (2) the kurtosis of the normalized instantaneous frequency. Then, the signals were fed into a CNN-LSTM classifier where QAM16 and QAM64 schemes were considered as the same class. Finally, QAM16 and QAM64 were classified by using Haar-wavelet transform crest searching.

In [58], an adversarial transfer learning architecture was introduced using the RadioML2016.10A dataset. In order to improve the AMR overall performance, both adversarial training and knowledge transfer were exploited. The computational complexity of this architecture was found to be slightly larger than CNN and DNN. However, it was acceptable since the performance improvement was significant. Table 6 summarizes the network configuration, modulation pool, maximum achieved accuracy per SNR, and remarks of the previously discussed CNN-based AMR based on IQ samples.

c: CLASSIFICATION USING IMAGE REPRESENTATIONS

DL technologies in the last decades have achieved outstanding achievements in the field of image processing. Therefore, many researchers in the field of wireless communication are very keen on the idea of converting the signal recognition problem into image recognition one. For example, in [68], a CNN-based AMR was presented in which various signal spectrograms were converted into an image dataset using Short-Time Fourier Transform (STFT). The authors introduced two image classification approaches in order to examine the AMR accuracy. The first one is by optimizing activation functions, while the second one is by using optimization functions. Similarly, Sun *et al.* [64] and Zhang *et al.* [69], adopted the same methodology. However, the authors in [64] implemented VGG-16 [70], a widely known CNN model, for recognizing 10 different modulation schemes. Similarly, the algorithm in [69] exploited both smooth pseudo Wigner-Ville and Born-Jordan distributions to obtain a different kind of images. Then, these images were fused with hand-crafted features of the received signals. In the same context, images were used in [71] as the signal representation, while Wigner–Ville map pictures were used in [72]. In [73], a spectrum CNN based AMR framework and a Gaussian filter for noise elimination were presented.

The authors in [74] used constellation diagram and AlexNet CNN model [75] for network training and classification tasks. Furthermore, Caffe framework [76] was adopted as well in the whole modulation classification process. Data augmentation was applied in [77] based on ACGAN. Particularly, the data conversion algorithm converts constellation diagram to contour stellar image. Therefore, more color features were obtained compared to the constellation diagram in [74]. Moreover, this made the CNN significantly outperform the same CNN model mentioned in [74]. Traditional Generative Adversarial Nets [78] suffers from several training issues such as generator disconverge, discriminator overfitting, and mode collapse. Hence, the authors presented several measures to overcome these issues. Results showed a 0.1-6% increase in the classification accuracy using the extended dataset when comparing to the original one.

Waveform identification is considered a major part of CR technology. Therefore, Zhang *et al.* [79], proposed a blind method for CR waveforms recognition using Choi-Williams time-frequency Distribution (CWD). The received signals were also transformed to 2D images through CWD. Then, they were converted into a binary image by using image binarization and image denoising algorithm. In [80], an AMR method for PSK/QAM schemes of different orders was proposed based on constellation structure and k-medoids clustering in the slow and flat fading channel. Neuron pruning techniques were used in [81] to reduce not only the convolution parameters but also the number of Floating Point Operations per Second (FLOPs). This work was dedicated for AMR implementation in the edge equipment. Anyhow, the average percentage of zeros criterion was adopted for convolution layers. In addition, contour stellar image was used as the signal representation. In [82], a CNN-based AMR was proposed for the 5G signal modulation. AlexNet CNN was used for features extraction and classification. $\pi/2$-BPSK, QPSK, 16QAM, 64QAM, and 256QAM were employed as the modulation pool and the constellation was selected as the input feature of the AlexNet network. In [83], constellation diagram and transfer adaptation was adopted in features extraction subsystem, while SVM was used for classification.

Zha *et al.* [84], introduced a DL framework for multi-signals detection and modulation recognition. The proposed scheme can obtain the signal modulation format, center frequency, and start-stop time. Both single shot multi-box detector networks and multi-inputs CNNs were built for signal detection and modulation recognition, respectively. The time-frequency spectrum was employed as the signal characteristic expression. Hence, MFSK modulation schemes were identified during the signal detection phase while MPSK, MAPSK, and MQAM schemes were identified in

TABLE 6. Summary of network configuration, modulation pool, achieved accuracy, and remarks of CNN-based AMR Methods.

Paper	Configuration: Dense	Conv	Flatten/BN	Pooling	Dropout	Training dataset size	Testing/validating dataset size	Modulation Pool	Accuracy per SNR	Remark	
[39]	3	7	✗	7	✗	(*)	(*)	RadioML 2018.01A	(**)	• **Input:** IQ samples.• Epochs: [0-50] • **SNR Range:** [-20, +20] • **Benchmarks:** Statistical Features in XGBoost[59]. • Transfer learning was investigated. • Refer to Table 8 for RNN configuration. • (*) 80% training data and 20% for testing where dataset size∈{240k,1M,1.4M,2M}.	• (**) >98.3%, > 60%, and >67% at high SNR for the normal classes with 1M training samples, the difficult classes with 240k training samples, and with unsynchronized LOs and Doppler frequency offset with 240k training samples, respectively.
[38]	2	2	✗	-	✔	96k	64k	RadioML 2016.1A	>80% for SNR>-2dB	• **Input:** IQ samples. • **SNR Range:** [-20, +18] • **Benchmarks:** DNN, DT, NaiveByes, KNN & SVM. • Epochs: 92 • Batch size: 1024	• Adam solver was used. • Regularization was used to prevent overfitting in CNN. • Epochs take roughly 15 sec.
[41]	3	2	✔	-	✔	20k[ts]	20k[ts]	D1={BPSK, QPSK, 8PSK} and D2={BPSK, QPSK, 8PSK, 16QAM}	≈100% for SNR>0dB (D1) and for SNR>+5dB (D2).	• **Input:** IQ samples. • **Benchmarks:** DT, RF, and DNN. • **SNR Range:** [-10, +20] with 5dB step.	• Stochastic Gradient Descent (SGD) was used to avoid overfitting. • Epochs: 500 • Batch size: 500
[6]	2	5	✔	4	✗	79k, 228k	8.8k, 25k	{2PSK, 4PSK, 8PSK, 16QAM, 16APSK, 32APSK, 64QAM}	86% for SNR>+6dB	• **Input:** IQ samples. • **Benchmarks:** FB-AMR[60] & ML-AMR. • **SNR Range:** [-6, +10]	• SGD was used for minimising loss function. • Adopted transfer learning and two-step training. • Epochs: 20-240 • Batch size: 500-1000
[42]	2	3	✔	2	✗	0.5k[t]	0.5k[t]	{BPSK, QPSK, 8PSK, 16QAM, 64QAM, 4FSK, 8FSK, PAM, DSB, FM}	>80% for SNR>-2dB	• **Input:** IQ samples. • **SNR Range:** [-20, +20] • **Benchmarks:**Resnet[61],Densenet[62], ResNeXT[63]. • Parameters: 590848 without 1×1 convolution kernel & 153600 with it.	• (NiN) multi stream network was adopted. • parameter size: 164618 (R1=1, R2=1), 535754(R1=2, R2=2), 700842(R1=3, R2=3) & 906890(R1=4, R2=4). • Epochs: 0-55 • Batch size: 128
[43]	4[a] 3[b]	2	✔	-	✔	2k[t]	2k[t]	{2FSK, DQPSK, 16QAM, 4PAM, MSK, and GMSK}	>83% for SNR>0dB, and >98% for SNR>6dB	• **Benchmarks:** CNN,CNN without BN,RNN,HOC-DNN & HOC-RF.	• **SNR Range:** [0, +12] • **Input:** IQ samples. • Epochs: [0-100]
[46]	2	3	✔	2	✔	1.4k[t]	0.6k[t]	{BFSK, DQPSK, 16QAM, 4PAM, MSK, GMSK}	≈100%, and >85% for SNR>0dB under AWGN and Rayleigh, respectively.	• **Input:** IQ samples. • **Benchmarks:** HOC & SVM.	• **SNR Range:** [0, +12] • Refer to Table 8 for RNN configuration.
[47]	3	2	✗	-	✔	6k[ts]	6k[ts]	{FSK, PSK, QAM}	AWGN:≈100% for SNR>-3dB Rayleigh:≈98% for SNR>-1dB	• **Input:** IQ samples. • **Benchmarks:** Traditional CNN. • **Simulation SNR Range:** [-5, +5] with an interval of 5 dB.	•Epoch times: 40 for AWGN Channel and 55 for Rayleigh channel. • Early-stopping is adopted to avoid overfitting.
[48]	3	2	✔	-	✔	6k[ts]	6k[ts]	G1={BPSK, QPSK, 8PSK}and G2={BPSK, QPSK, 8PSK, 16QAM}	>80% for SNR>0dB in G1 and >70% for SNR>0dB in G2	• **Input:** IQ samples. • **Benchmarks:** Traditional CNN-based AMR. • **SNR Range:** [-10, +10] with 2 dB as an interval.	• Neuron pruning via regularization was adopted. • Fine-tuning and retrain the trimmed CNN in the short epochs was applied.
[49]	2[e]-2[f]	2[e]-46[f]	✔	2	✔	1k[ts]	1k[ts]	{BPSK, QPSK, 8PSK, OQPSK, 2FSK, 4FSK, 8FSK, 16QAM, 32QAM, 64QAM, 4PAM}	≈88% at SNR = 10dB[e] - ≈ 94.5% at SNR = 10dB[f]	• **Input:** IQ samples. • **Benchmarks:** CNN1 and CNN2. • **SNR Range:** [-20, +30] with an interval of 2 dB. • Validation is performed every 1500 iterations.	• SGD was used to prevent overfitting. • The performance of confidence-based and feature-based fusions are almost the same and they are both better than voting-based fusion. • The performance gains under CNN1 is much more obvious.
[50]	1[g]-2[h]	2[g]-4[h]	✔	-	✗	-	-	{2ASK, 4ASK, 2FSK, 4FSK, BPSK, QPSK, 16QAM and 64QAM}	≈ 69% at SNR = 0dB	• **Input:** IQ samples. • **SNR Range:** [-14, +10] • **Benchmarks:** CNN and KNN • Parameters: 437071	• SGD was used • Semi-supervised training. • Inference time: 0.78ms • Batch size: 10
[52]	4	4	✗	2	✗	-	-	RadioML 2016.01A	80% for SNR>0dB	• **Input:** IQ samples. • **Benchmarks:** CNN & CLDNN [44].	• **SNR Range:** [-20, +18] • Adam optimizer was used.
[53]	25	3	✗	-	✗	110k	110k	RadioML2016.10A [64]	>80% for SNR>-2dB	• **Input:** IQ samples. • **Benchmarks:** CNN, GRU, focal loss & cross entropy. • **SNR Range:** [-20, +18]	• Adam optimizer was used. • Epochs: 10 • BN layer & Global average pooling were adopted.
[54]	1	21	✗	7	✗	80% 1.6M	20% 0.4M	RadioML 2018.01A	85% for SNR>10dB	• **Input:** IQ samples. • **Benchmarks:**(ML-XGBoost,VGG,ResNet)[39], CNN-AMR[6]. • **SNR Range:** [-20, +30] • Trainable parameters: 141,880	• M-block comprises 3 convolutional layers. • Skip connections technique was applied. • SGD was used to avoid overfitting. • Accuracy increased as M-block increased. • Epochs: 60 • Batch size: 128
[55]	1	3	✔	3	✗	70% = 1.4M	30% = 0.6M	RadioML 2018.01A	>70% for SNR>7dB	• **Input:** IQ samples. • Epochs: 15 • **Benchmarks:** CNN[39].	• **SNR Range:** [-20, +30] • PCA was adopted.
[40]	2	4	✔	4	✔	416k - 600k	364k - 600k	HisarMod2019[4] - RML2016.10a2	>90% at SNR>14dB - ≈90% at SNR>4dB	• **Input:** IQ samples. • **Benchmarks:** CLDNN[65], LSTM[66]. • **SNR Range:** [-20, +18]	• Parameters = 1576453 for HisarMod2019 and 659594 for RML2016.10a. • Adam optimizer was used.
[56]	2	4	✔	2	✔	720k	480k	D1= RML2016.10b, and D2 = RML2016.10a	91.70% for SNR≈ 10dB for D1, and 83.40% at SNR = 18dB for D2	• **Input:** IQ samples. • **Benchmarks:** CNN2 [51], ANN [15], CNN & CNN1 [67].	• **SNR Range:** [-20, +18] • Gaussian noise layer was applied for regularization. • Computing time = 0.0048ms
[57]	2	3	✔	-	✗	80% = 960k	20% = 240k	RML2016.10a	=92.3% at SNR = 10dB	• **Input:** IQ samples an 2 expert features. • **Benchmarks:** CNN-LSTM [39].	• **Simulation SNR Range:** [-20, +18] • One LSTM layer was added in the network structure.
[45]	4[c] 5[d]	2[c] 3[d]	✔	1[c] 3[d]	✔	4.2k[s]	1.8k[s]	{BPSK, QPSK, 8PSK, GFSK, CPFSK, 4PAM, 16QAM, and 64QAM}	95% for SNR>+2dB	• **Input:** IQ samples in DrCNN & constellation diagrams in CNN. • **Benchmarks:** MaxCNN[51],RNN,DNN & Inception.	• **SNR Range:** [-8, +18] with an interval of 2 dB. • Distinguish QAMs with high accuracy at SNR>-6dB. • Netwrok parameters: 1,101,448

[ts]per type per SNR, [t]per type, [s]per SNR, [a]Former CNN, [b]Latter CNN, [c]DrCNN, [d]LatterCNN, [e]CNN1, [f]CNN2, [g]FECNN, [h]GMCNN

the same categories to be then recognized using multi-inputs CNNs. In [85], a comparative analysis was provided in order to clarify the effects of choosing the features, classifiers, and datasets. Also, the effects of the presence of noises on the classification performance were investigated. Feature discrimination analysis was provided through SVM, DNN, and CNN. In [86], a two-stage hybrid digital AMR method was proposed based on STFT and CNN. Firstly, STFT was used to convert the signals into 2D images in order to extract the time-frequency features and feed the input of the CNN classifier. In [87], a CNN-based intelligent eye-diagram analyzer was proposed for modulation format recognition and optical SNR OSNR) estimation. Received signals were converted into eye diagram images that have a grid-like topology and then processed by CNN for modulation classification.

d: CLASSIFICATION USING OTHER INPUTS

Anti-noise processing and deep sparse-filtering CNN were used in [88] to design an AMR method for very high frequencies. The Cyclic Spectra (CS) of modulated signals were calculated first. Then, Low-Rank Representation (LRR) algorithm was applied in the de-noising process of the obtained cyclic spectrum images. Zhang *et al.* [89] proposed a combination of the IQ samples and the 4th-order cumulants of the modulated signals. The combined vector is then fed into both CNN and LSTM classifiers, namely CNN-IQFOC and LSTM-IQFOC, respectively. In [90], the impacts of features selection for AMR were investigated. Moreover, a novel method for selecting the most effective and distinctive features from a larger set of features was proposed. This method was called Bhattacharrya distance-based feature selection algorithm. In particular, the Bhattacharyya distance metric was applied to evaluate the similarity between two different probability distributions, and calculate the highest distance for all candidate modulation schemes. This research studied the case of SISO system with perfect timing synchronization, AWGN, and frequency-selective fading channels. However, many simulations showed that the proposed algorithm achieves a substantial reduction in the computational complexity and outperforms the traditional PCA.

Teng *et al.* [5] extend their previous work in [91] and proposed an accumulated polar feature-based DL with a channel compensation mechanism. In [91], they have shown that learning features from polar coordinates, which can be obtained from Cartesian coordinates, can achieve higher recognition accuracy. Then in [5], they add a new temporal axis to accumulate historical information of symbols in such dimension. In the proposed method, the polar coordinates were projected to grid-like images. After that, the grid-like images were converted into colorful images to feed the CNN classifier. Moreover, an ANN-based channel estimator was presented in order to cope with the problem of fading channels. This was accomplished by finding the inverse channel response and reduce the impact of power scaling and phase shift on the performance. Two mechanisms for online retraining were introduced to deal with the time-varying fading channel while having lower transmission and retraining overhead. The model reported in [91] was used as a benchmark. Results revealed that the proposed method can reduce the offline training overhead by about 190 times compared to [91] and, therefore, provide better efficiency and accuracy. Table 7 summarizes the network configuration, modulation pool, maximum achieved accuracy per SNR, and remarks of the previously discussed CNN-based AMR when images and other signal representations are employed.

3) RNN-BASED METHODS

Data augmentation techniques are considered as a proper solution for the drawbacks of insufficient training data. In [102], for example, a state-of-the-art LSTM-based AMR was presented to evaluate Gaussian noise, rotation, and flip methods of radio signals augmentation in both training and inference phases. Similarly in [103], RNN and LSTM were investigated for AMR. First, the received signals were converted into two normalized matrices to be then fed into RNN. After that, sequence-correlated features of I/Q signal components and amplitude/phase signal components were extracted using 4-layer dual-channel LSTM. In [104], a class activation vector was introduced to visualize the extracted features by using different DL-based radio modulation classifiers including CNN and LSTM classifiers. On one hand, the CNN classifier was found to be insensitive to the input formats and capture similar radio features. On the other hand, the LSTM classifier discriminates different modulation schemes in a similar manner to the knowledge of human experts. Moreover, the LeNet-based and the ResNet-based classifiers tend to capture the transitions between and around modulation reference points, respectively.

Authors in [105] outlined a Dual Path Network (DPN) for joint blind modulation classification and symbol recovery. This architecture combines both DL and linear signal processing in order to estimate signal parameters and correct signal distortions like CFO and multipath fading. It consists of a combination of residual blocks, LSTM and RNNs. Briefly, the received samples were used to generate five different outputs. Each output has a corresponding loss function different from others. Hence, the training loss was a combination of these different loss functions. Simulation results revealed that the proposed DPN not only provides good accuracy in signal distortion estimation but also outperforms many DL methods in terms of classification accuracy. In [106], a blind modulation scheme detection algorithm of non-orthogonal multiple access [107] systems was designed. ResNet [61] was used as a classifier, and joint constellation density diagrams were used as the discriminating representation. Furthermore, the wavelet de-noising method was adopted in order to reduce the bad effects of high-intensity noise and losses, thereby improve the quality of constellations.

TABLE 7. Summary of network configuration, modulation pool, achieved accuracy, and remarks of CNN-based AMR Methods.

Paper	Configuration: Dense	Conv	Flatten/BN	Pooling	Dropout	Training dataset size	Testing/validating dataset size	Modulation Pool	Accuracy per SNR	Remark	
[68]	3	2	✔	2	✗	36k^{t}	16.8k^{t}	{AM, BPSK, FSK, FM}	>80% for SNR>-6dB	• **Input:** STFT spectrogram images. • **Benchmarks:** SGD[65], Adam[92] & RMSProp[93]. • **SNR Range:** [-20, +20]	• Iterations < 80k • Adam solver was used. • The recognition accuracy under different activation functions (ReLU, Sigmoid, softplus, tanh) was studied.
[72]	1	2	✗	2	✗	4k^{t}	0.2k^{t}	{Const, LFM, SFM, PFM, and FSK}	>93% when SNR>0dB	• **Input:** time-frequency images using Wigner-Ville formula. • **Benchmarks:** Methods in [4]. • **SNR Range:** [-4, +15]	• Mini-batch gradient descent was used to improve efficiency. • The error rate decreases with increasing iteration number. • Iterations: [0-40]
[73]	1	4	✗	3	✗	0.7k^{ts}	0.3k^{ts}	RadioML2016.10A [64]	>80% for SNR>0dB	• **Input:** Spectrogram images • **SNR Range:** [-20, +18] • **Benchmarks:** CNNR- IQFOC [89] & CNNR-IQ [38].	• Training time: 7.14 ms • Memory: 941k • iterations: 15 • Epochs: 0-120 • Network parameters: 199k
[74], [77]	3	5	✗	3	✔	10k^{t}	1k^{t}	{BPSK, 4ASK, QPSK, OQPSK, 8PSK, 16QAM, 32QAM, 64QAM} [77] & {QPSK, 8PSK, 16QAM, 64QAM} [74]	> 95% for SNR > -2 dB [77] and for SNR>8 dB [74]	• **Input:**contour stellar images[77] & constellation diagram[74]. • **Benchmarks:** GoogleNet, SVM & Cumulant. • **Simulation SNR Range:** [-6, +14][77], [-4, +14][74].	• Data Augmentation was applied using ACGAN [77]. • Iterations < 50k [77], 100k [74] • Batch size: 64 • AlexNet [75] was adopted.
[81]	5	2	✗	-	✔	8k^{s}	8k^{s}	{4ASK, BSPK, QPSK, OQPSK, 8PSK, 16QAM, 32QAM, 64QAM}	>90% for SNR>-4dB	• **Input:** contour stellar images. • **Benchmarks:** (AlexNet,SVM-5,AVM-7,Cumulant)[94], Orginal AlexNet. • **SNR Range:** [-6, +6] with an interval of 2 dB.	• AlexNet was dopted with 60M parameters & 727M FLOPs. • This method could use only 1.5%-5% parameter and 33%-35% time without losing accuracy more than 1.2% • This method was implemented on NVIDIA Jetson TX2.
[84]	1	3	✔	3	✗	-	-	{BPSK, QPSK, OQPSK, 8PSK, 16QAM, 16APSK, 32APSK, 64QAM}	>80% for SNR>0dB	• **Input:** vector and eye diagrams. • **SNR Range:** [0, +10]	• **Benchmarks:** Cumulant[95],SVM-7[4],CNNR-IQ[38] & CNN[94].
[86]	3	5	✗	3	✔	1071^{t}	459^{t}	{ASK, FSK, PSK, QASK, QFSK, QPSK}	>99% for SNR>0dB	• **Input:** Spectrograms images using STFT. • **Benchmarks:** VGG-16[96], VGG-19[70], GoogLeNet[97], and ResNet-101[61].	• **SNR Range:** [0, +25] • transfer learning approach was adopted. • Batch size: 64 • Epoch number: 32
[88]	1	2	✗	2	✗	10k^{t}	1k^{t}	{BPSK, QPSK, 2FSK, 4FSK, MSK, AM, FM}	95% for SNR>+2dB	• **Input:** CS-LRR samples. • **SNR Range:** [0, +20] • **Benchmarks:**CS-Graph[4] & RawWave-CNN[98]. • Fine-tune the entire CNN with supervised learning.	• Unsupervised sparse filters to pre-train the CNN. • Netwrok Parameters: 3312 • Epochs: 5 • Batch size: 50
[89]	2	2	✗	2	✔	0.7k^{ts}	0.3k^{ts}	RadioML2016.10aA [64]	>80% for SNR>0dB	• **Input:** IQFOC samples. • **SNR Range:** [-20, +18] • **Benchmarks:** LSTM-IQ , CNN-IQ , and CNNR-IQ [38].	• Refer to Table 8 for LSTM configuration.
[90]	3	2	✗	2	✗	64k[1] 96k[2]	64k[1] 96k[2]	{BPSK, QPSK,8-PSK,16QAM}	• >90% for SNR>2dB for AWGN channel. • >80% for SNR>6dB in FS-fading channels.	• **Input:** feature sets including HOC. • **Benchmarks:** CNN, Sparse AE DNN (SAEDNN), Radial Basis Function Network (RBFN), and PCA. • Loss function: MSE for SAEDNN and cross-entropy for CNN. • Epochs: 400[1] -10[2] • Batch size: 256[2]	• **SNR Range:** [0, +15] • Iterations: 500 • Optimization Algo: L-BFGS for SAEDNN & SGDM for CNN. • The complexity can be coarsely approximated to O(NDm), where "N", "D" and "m" are - dataset size, dimension of input data and number of targets respectively.
[5]*	3	2	✔	2	✔	5k^{t}	1k^{t}	{QPSK, 8PSK, 16QAM, 64QAM}	(*)	• **Input:** Polar features. • **SNR Range:** [-4, +12] • **Benchmarks:** CNN[74], CNN[91], ML[99], HLRT[100], HOC[101]. • Classification accuracy average: 97.9% at high SNR. • Parameters: 2 / 1,032 • Time per epoch: 1.08 s	• (*) ≈ 98% for model size = 1032 and SNR = 8dB,≈ 98% for training data size = 10k and SNR = 8dB, >80% for SNR>2 dB under AWGN channel, >80 % for SNR>3 dB under fading channel.
[5]**	3	2	✔	2	✔	259k	317k	RadioML 2018.01A	74.2% for SNR = 20dB	• **Input:** IQ samples. • **SNR Range:** [-10, +20] • **Benchmarks:**LSTM[66],CNN[74],ResNet[39] & CNN[4].	• Time per epoch: 6s • Training time: 198s • Epochs: 33

^{ts}per type per SNR, ^{s}per SNR, ^{t}per type, [1]SAEDNN, [2]CNN

4) DNN-BASED METHODS

Several papers [108], [109] proposed DL-AMR where the decision on the modulation scheme is made by DNN. For instance, a novel Particle Swarm Optimization (PSO) scheme was introduced in [108] to improve the structure of DNN for optimizing the number of hidden layer nodes. Similarly, the authors in [110] employed DNN for features extraction in digital coherent receivers. Amplitude histograms of the received signals were obtained after constant modulus algorithm equalization. Results revealed that the proposed method was non-data-aided and, therefore, it does not have any bad effects on the spectral efficiency of the system. In [111], twenty-one statistical features were extracted from the received signals in order to feed the fully connected DNN classifier. The proposed technique has a higher dimension of the feature space than the conventional AMR methods, it has nonetheless a good classification performance. The extracted features were carefully selected based on the analysis of the empirical distribution for each modulation class. Simulations demonstrated good performance for both AWGN and Rican channels at different Doppler frequencies and SNRs.

In [112], a DL-AMR framework was proposed based on the IQ constellation polluted by AWGN. The noisy received samples were first normalized using the K-means algorithm. Thereafter, the signals were converted into images to be then fed into the DNN classifier. It is worth mentioning that 20% dropout was applied on hidden layers to avoid overfitting. The method in [113] consists of a large Hybrid DNN (HDNN) and a smaller Layered Resnet Network (LRN). The HDNN is composed of 1-D Resnet and LSTM-RNN layers. Cross model DL was proposed in order to reduce the consumption of both storage and time. Therefore, the low-complexity requirement can be fulfilled for real-time applications. Besides, a knowledge distillation method was also proposed to drive the learning process of LRN.

DBN classifier was exploited in [114] for AMR. First, the Spectral Correlation Function (SCF) generates 2-D images. After that, the generated images will be pre-processed and fed into the DBN classifier. To evaluate the performance, MAXNET classifier [115] and a feed-forward backpropagation ANN with a continuous output [116] were used. However, a common application of DBN is feature extraction which can be used in different concepts with different granularity. The authors of [117] propose a combination of DBN and SVM classifier. Specifically, the stacked RBM networks were used to form a DBN structure in order to extract relevant features of the received signals.

5) AE-BASED METHODS

Li *et al.* [118], introduced a Stacked Sparse AE (SSAE) and softmax regression-based DL network for AMR. They employed a cyclic spectrum in order to pre-process the received signals. Also, an unsupervised SSAE-DNN-based AMR method was proposed in [119]. Its main aim was to cope with much-neglected frequency selective fading scenarios with Doppler shift. It was particularly trained by several low complexity features such as spectral and cumulant features. Furthermore, the network was trained on a range of SNR values. Simulation results showed that the proposed method can be feasible for all channel conditions. Moreover, it can achieve a robust classification performance under several impairments such as phase-frequency impairments and Doppler shift for frequency selective fading scenarios.

Zhang *et al.* [120], proposed a hybrid AMR algorithm for multipath fading channels. In particular, HOCs up to sixth-order and Stacked Convolutional AEs (SCAEs) were employed. The overall process starts when the received multipath signal was down-converted to the baseband. Then, HOCs were calculated and a series of cumulant patches were generated. Subsequently, zero-phase component analysis whitening was applied to each cumulant patch to reduce its dimension. Finally, these patches and their real modulation types were combined into pairs to feed the SCAE classifier. Dai *et al.* [71] proposed a novel inter-class DL-AMR method by using two SSAE for features extraction. Received signals were first converted into AF images to be then fed into SSAE. The decision on the corresponding modulation scheme was made by a softmax regression classifier. Table 8 summarizes the network configuration, modulation pool, training and testing parameters, and remarks of the previously discussed RNN, DNN and AE based AMR methods.

B. DL-AMR METHODS FOR MIMO SYSTEMS

MIMO and massive MIMO are among the smartest antenna technologies and they are considered as key enablers in the current and future communication systems. In such systems, there are multiple antennas at both ends, the transmitter and the receiver. The increasing number of used antennas will result in a significant improvement in link range and data throughput. A survey on the emerging research on DL-based models for MIMO and massive MIMO systems can be found in the literature [121]. An investigation on the MIMO detection problem, in the presence of correlated interference over time or frequency, by employing a deep CNN was proposed in [122].

Wang *et al.* [123], proposed a CNN-based cooperative AMR method for the MIMO systems. In order to make the final decision on the modulation scheme, two kinds of cooperative decision rules were introduced in the decision-maker. The first kind is the voting method, while the second kind is the averaging method. In the former method, all votes on the decided modulation schemes will be gathered from each antenna, and the scheme with maximum votes will be the final decision. While in the latter method, the average of probability distribution functions achieved by all of the receiving antennas will be calculated and the scheme with the highest average probability will be the final decision. However, the decision at each antenna is equally important in the direct method, whereas each antenna has a different weight in the weighty method. Results revealed that the averaging method is better than the voting one and the weighty method is better than the direct one. Besides, they proposed a CNN/ZF-AMR method [124] for MIMO systems. ZF equalization was employed since it can increase the SNR of the received signal. The conducted simulations revealed that the performance was much better in the case of perfect Channel State Information (CSI). Furthermore, the classification performance was proved to be influenced by the imperfect CSI; and also with the number of the transmit and receive antennas. They also proposed a Transfer Learning (TL)-based semi-supervised AMR (TL-AMR) in a ZF-MIMO system [125]. The proposed method includes a novel Deep Reconstruction and Classification Network (DRCN) structure. This DRCN consists of two pipelines: Convolutional AE (CAE) and CNN. The former pipeline was used for labeled samples classification, while the former was used for the massive unlabelled samples reconstruction. Consequently, the knowledge will be transferred

TABLE 8. Summary of (RNN, DNN & AE) based AMR methods.

Algorithm	Network configuration	Modulation pool	Remarks	
OTA-DL-based radio signal classification [39]	Input layer, 6 residual stacks, 2 fully connected layers, and softmax output layer	RadioML 2018.01A	• ResNet performance under channel impairments & versus depth was studied.	• Training parameters: 236,344
CNN-RNN-Based AMR [46]	Input layer, 3 LSTM layers, 3 fully connected layers and softmax output layer	{BFSK, DQPSK, 16QAM, 4PAM, MSK and GMSK}	Refer to Table 7	
SSAE-AF [71]	Input layer, hidden layer and output layer	{ASK, PSK, QAM, FSK, MSK, LFM, and OFDM}	• Training data: (6*7)k • Testing data: (1*7)k • Batch size: 100 • Epochs: 100	• Unsupervised pre-training and supervised fine-tuning.
LSTM-IQFOC [89]	Input layer, 3 convolution layers, 1 LSTM layer, 1 fully connected layer and softmax output layer	Refer to Table 7	• LSTM-IQFOC performance was around higher 2% than CNN-IQFOC (Table 7) for SNR>-5dB.	• Tanh & RELU activation functions were used.
LSTM-based AMR [102]	Input layer, 2 LSTM layers, one fully connected layer and softmax output layer	RadioML2016.10a	• Training data: 110-440k • Testing data: 110k • Batch size: 128 • Epochs: 80	• Combine both rotation and flip methods can provide further improvement.
DC-LSTM AMR [103]	Input layer, 2 dual-channel LSTM layers and softmax output layer	RadioML2016.10a	• Training data: 110k • Testing data: 110k • Batch size: 256 • Epochs: 70	• Center loss and weibull distribution.
DL-based Radio Modulation Classification [104]	Input layer, 2 LSTM layers, 2 fully connected layers and softmax output layer	RadioML2016.10a	• Training data: 88k • Testing Data: 11k • Batch size: 128 • Epochs: 150	• Validation Data: 11k • Adam optimizer was employed.
PSO-DNN [108]	Input layer, 2 hidden layers and softmax output layer	{BPSK, QPSK, 8PSK} and {16QAM, 64QAM, 256QAM}	• No. PS: (10-20-30-50) • Iteration number of training/optimizing: 300/5	• SGD method used for training the hidden layers and MSE function to calculate the output error.
DNN [109]	Input layer, 2 hidden layers and softmax output layer	{2FSK, 4FSK, 2PSK, 4PSK, 2ASK, 4ASK}	• Epochs: 278	• Loss function: cross-entropy
DNN-based MFI [110]	Input layer, 2 hidden layers and softmax output layer	{QPSK, 16QAM, 64QAM}	• Training data: 135*3 • Testing data: 60*3	• Unsupervised pre-training and supervised fine-tuning.
DNN-based AMR [111]	Input layer, 3 hidden layers and softmax output layer	{BPSK, QPSK, 8PSK, 16QAM, 64QAM}	• Batch size: 50 • Epochs: 15k • Training data: (15*5)k • Validation Data: (5*5)k	• Testing Data: (5*5)k • Training method: SGD. • Cost function: negative-log- likelihood.
DNNG [112]	Input layer, 3 hidden layers and softmax output layer [48]	{16QAM, 4QAM, 8PSK, 16PSK, BPSK, 8QAM, 32QAM}	• The best performance results was found to be at epoch equal to 2M.	
DBN-SCF-based AMR [114]	5 hidden layers and an output layer	{4FSK, 16QAM, BPSK, QPSK, and OFDM}	• Training data: 6k • Testing data: 500	• Verification data: 2k • Semi-supervised learning
SSAE [118]	Input layer, hidden layer and output layer	{FSK, PSK, ASK, QAM and MSK}	• Training data: (2.5*5)k • Testing data: (1*5)k • Batch size: 100 • Epochs: 100	• Unsupervised learning.
SCAE-HOC [120]	Input layer, six hidden layers (3 convolutional layers, 2 pooling layers and one fully connected layer)	{BPSK, QPSK, 16-QAM, and 64-QAM}	• Training data: 120k • Testing data: 60k	• Unsupervised pre-training and fine-tuning were performed. • SGD was performed to reduce the training error.

from CAE to the feature layer of CNN in order to cope with ineffective training and over-fitting, which is caused by using limited labeled samples. However, it is worth mentioning that the modulation schemes used in all of these researches were PSK, QPSK, 8PSK, and 16QAM. Besides, the CNN structure was also the same including 2 convolutional layers, 3 dense layers, and a softmax output layer. Furthermore, the CAE in [125] had the same structure as in CNN with three deconvolution layers.

1) DISCUSSION AND LEARNED LESSONS

DL models play an important role in AMR and the performance demonstrated by recent prototypes is remarkable. This is because DL can prove advantageous with regard to feature learning, performance, complexity, and generalization capabilities [2]. On one hand, training DL models on a wide range of SNR can eliminate the requirement of SNR estimation and generalize the classification process in the different environmental conditions [39]. However, maintaining low computational complexity, in such models, remains challenging as well as constructing and gathering a large dataset of signals in order to train the model and to guarantee the performance under varying noise regimes. For example, the CNN model reported in [42] enjoys a high classification accuracy but suffers from a large number of training parameters and, hence,

TABLE 9. Summary of DL-AMR method for MIMO systems.

Paper	Training & Testing parameters	Remarks
[123]	• Training data: 14k^{ts} • Validation data: 6k^{ts} • Testing data: 10k^{ts}	• ReLU & dropout were used. • Antennas number: N_r×N_t∈{4×1, 4×2, 4x4}. • Benchmark: HOC and ANN-based AMR. • Classification accuracy: ≈100%,>91%,>87% for SNR>0dB when N_r N_t∈{4×1,4×2,4x4}, respectively.
[124]	• Training data: 20k^{ts} • Testing data: 10k^{ts} • Maximum epoch, early-stopping epoch, batch size were set as 100, 20 and 500, respectively	• ReLU and Dropout were used. • Antennas number: $N_r \times N_t \in$ {4×1, 4×2, 4x4}. • Benchmark: HOC and ANN-based AMR. • Classification accuracy: >90%,>83%,>81% for SNR>-5dB when N_r×N_t.∈{4×1,4×2,4x4}, respectively.
[125]	• Training data: 20k^{ts} • Testing data: 10k^{ts} • Maximum epoch, and batch size were set as 500, and 500, respectively.	• ReLU used for all layers, while softmax and PReLU were used for the output layer in CNN and the last deconvolution layer in CAE. • Dropout were used. • Antennas number: $N_r \times N_t \in$ {4×1, 4×2, 4x4}. • Benchmark: CNN/ZF-AMR [125]. • Classification accuracy: >90%, >82%, and >80% for SNR>-5dB when $N_r \times N_t \in$ {4×1,4×2,4x4}, respectively.

^{ts}per type per SNR

high complexity. On the other hand, training a DL model on a narrow range of SNR [46] can obtain high accuracy. However, to target a wide SNR range, it is needed to use multiple DL models for each SNR value, which will result in high storage requirements. Fundamentally, the main difference between RNN and CNN is the nature of their input data. While CNN is specialized in modeling spatial data, RNN is very effective in modeling temporal data. In fact, weight sharing is a key advantage in CNN as well as the auto-learning of deep spatial features. Alternatively, RNN can remember each information through time and, hence, very appropriate for sequenced data applications such as speech-to-text applications and video processing applications. However, both CNN and RNN suffer from vanishing and exploding gradient problems and require large amounts of data. It is worth mentioning that training RNN is a complex process and requires additional effort. AEs can be used effectively for unsupervised hand-crafted features learning [118], [120], or even for features extraction and learning [71].

However, RNN and CNN models can be used together to form a hybrid DL model and extend the effective pixel neighbourhood [57], [89]. Due to the advancements in the field of image recognition, image processing techniques are being extended to AMR domain. For this purpose, some kind of transformation algorithms have to be employed in order to adapt the techniques previously used for imaging to AMR [68], [72], [74]. However, this kind of data manipulation requires additional time which can complicate the AMR problem in time-sensitive applications. Likewise, several imaging techniques are being extended to AMR domain such as data augmentation [77] transfer learning [39], [86], and PSO algorithm [108], yet additional efforts are required to achieve the required performance and enable confident decision making.

V. CHALLENGES AND FUTURE RESEARCH DIRECTIONS

Although the advancement of DL-AMR models has been witnessed in the past few years, there are many issues that need to be addressed in future researches. Generally speaking, the design of an ideal and robust classifier remains an open-ended challenge, despite the significant improvements and blooming results of the previous works. For example, some algorithms need prior information of the signal (e.g. carrier frequency, baud rate, offset timing, etc.) [52], while others are limited to a small number of modulation schemes [47], [48]. Besides, some algorithms have considerable computational complexity [42] and cannot be used in real-time applications. Others are based on ideal cases with ideal assumptions and cannot be used in practical applications. Some classifiers need high SNR values [6] which is different from realistic scenarios. Consequently, if the current researches on DL-AMR are to be more successful, they need to focus on the key feature extraction [69] and the selection of classification criteria at low SNRs. Specifically, artificially choosing features, however, is a complicated and difficult process. Nowadays, there is an increasing need to enhance the flexibility of AMR in the current and future wireless communications to provide reliable low latency services. The rest of this section is devoted to point out some potential research directions and open challenges.

A. ELECTROMAGNETIC ENVIRONMENT

As found in the present literature, the use of DL technologies can simplify the signal processing steps and improve the AMR performance. Moreover, it can provide more practical, robust, and efficient AMR methods with high classification accuracy. Although DL-AMR methods outperform the traditional modulation recognition ones [119], but there is still a long way for signal recognition in real electromagnetic environments and practical applications. The performance of FB-based AMR methods tends to degrade badly with channel impairment effects such as the high-speed mobility and impulsive nature of noise [20], multipath fading effects [32], PO [41], CFO [52], and heavy noises and interferences [88]. However, the quality of service in the complex communication environment under low SNRs is hard to be guaranteed and, therefore, further investigation is required for the robustness of DL classifiers in a larger range of SNR [43].

B. SIGNALS DE-NOISING BEFORE AMR PROCESS

AMR at low SNR becomes more challenging due to the complex channel environment and the interference of multiple noises. Therefore, eliminating such noises and enhancing the SNR can improve the performance of AMR methods. Therefore, signals de-noising algorithms can be applied before AMR methods. For example, LRR algorithm was applied in [88] for de-noising the cyclic spectrum images before the AMR process. Similarly, image de-noising algorithms can be adopted in order to remove the different kind of noises [79], [106]. Moreover, DL models can be employed directly on the received raw signals in order to perform de-noising process before AMR method [126].

C. VARIABLE LENGTH SIGNALS

A large amount of researches found in the literature consider the input of the proposed DL model as a fixed-length vector, which is not the real scenario where the signal frame length is diverse [42], [89]. For instance, inclusive investigation and enhanced models based on fully convolutional networks [127], NiN [128], RNN [129], LSTM [102], [130] and signal segmentation algorithms [49] can obtain DL-AMR methods that can deal with variable-length inputs. Moreover, several techniques in the field of computer vision can be considered to resolve such issues. For instance, the following techniques can be used for this target:

- Resizing the input images to the same input size [79],
- Crop out certain regions of the input image when the images are large,
- Pad the images with a suitable background.

However, DL-AMR methods with variable-length input can contribute to the problem of short-time classification.

This kind of classification is required in many systems in real-world scenarios. For instance, scanning a receiver need to have a swift decision without consuming much time to acquire more data to increase certainty. Besides, it could be unavoidable when short signal bursts in the environment. Moreover, the classification on short observations can be very challenging but yet very needed in several systems where the delay is a very crucial parameter. Therefore, there is an urgent need to provide effective methods to deal with the variable-length input.

D. SIGNALS DATASETS

Even though DL-AMR can provide significant performance, it requires a large amount of training data. In practice, collecting a sufficient amount of reliable training samples is usually costly and difficult. Therefore, the used large-scale training dataset should be constructed adequately and carefully with more realistic signals under a wide range of SNR [39]. Furthermore, data augmentation methods can make the classification of radio modulation schemes more successful by using shorter radio samples [77], [102], which will provide a simplified DL model with a smaller classification response time.

E. ENHANCED DL MODELS

As mentioned before, the DL-based methods can operate under a wide range of SNR. Hence, large-scale datasets will be needed in order to achieve the expected results. This may result in a high number of training parameters and, hence, high computational complexity along with high storage requirements. Besides, the training process may consume a long time and suffers from over-fitting issues. Therefore, optimized and hybrid DL methods [46] can be studied in order to avoid over-fitting and reduce the computational complexity as well as to improve the classification accuracy [25], [108]. Moreover, the samples-labeling process in the training phase in supervised learning methods is very difficult to accomplish in addition to be time-consuming. Hence, the idea of semi-supervised [114] and unsupervised learning methods is being adopted [119]. These methods can meet the rapid growth of AMR applications and enhance the overall performance.

F. UNKNOWN MODULATION TYPES

In several military applications such as intelligence and surveillance systems, the AMR process should be able to detect the modulation type of unknown signals. For example, jamming devices are deployed in the communication channel between adversary units to prevent communication among them. More specifically, they transmit noise or dummy signals modulated by the matching modulation type. Therefore, the modulation scheme of the adversary communication signals should be detected in order to allow the jamming devices to transmit jamming signals modulated using the corresponding modulation scheme.

However, as stated in the existing literature, the type of modulation schemes to be classified is assumed to be within a pool of known ones [110], [111]. More specifically, seldom works are capable of coping with unknown signals. Therefore, a comprehensive set of modulation types or a more reasonable model design is needed to overcome the lack of universality for the unknown signal recognition.

G. REAL HARDWARE IMPLEMENTATION

In order to design an AMR method feasible to be implemented on real hardware, further investigation is required to achieve high classification accuracy along with low computational complexity. For example, the methods presented in [40], [50] suffer from a large number of learnable parameters. Conversely, the method proposed in [56] achieve low complexity with relatively low classification accuracy. Besides, the proposed DL methods found in the literature are still in the simulation phase and seldom works are devoted to implementing the proposed AMR method on real hardware. Hence, deploying DL-AMR algorithm to real powerful computing hardware (e.g. field-programmable gate arrays (FPGA) and beyond) can be very challenging [81], especially when it comes to timing and validation issues.

H. DL-AMR IN MIMO SYSTEMS

DL-AMR models are scarcely explored for MIMO and massive MIMO communication systems. In this direction, further investigation can be done in order to design new methods or even extend prior works which were dedicated for SISO systems. Although some signals datasets already exist in the literature for SISO systems, but yet there are seldom datasets for MIMO and massive MIMO systems. Hence, it is very important for future researches to build robust and sufficient signals datasets for such systems.

VI. CONCLUSION

Machine learning and deep learning are becoming increasingly popular in AMR domain. Research in this area is still incipient but it has however shown outstanding results. In this article, we provided a comprehensive survey of recent work lying at the junction of AMR, machine learning and deep learning. We reviewed both ML and DL models in AMR area for both SISO ans MIMO communication systems. We discussed the architecture of various deep learning models and pointed out the key advantages of each model. We warped up this paper by presenting potential research directions and open challenges, which may result in significant future research results.

REFERENCES

[1] Z. Zhu and A. K. Nandi, *Automatic Modulation Classification: Principles, Algorithms and Applications*, 1st ed. Hoboken, NJ, USA: Wiley, 2015.

[2] C. Zhang, P. Patras, and H. Haddadi, "Deep learning in mobile and wireless networking: A survey," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 3, pp. 2224–2287, 3rd Quart., 2019, doi: 10.1109/COMST.2019.2904897.

[3] Y. A. Eldemerdash, O. A. Dobre, and M. Oner, "Signal identification for multiple-antenna wireless systems: Achievements and challenges," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 3, pp. 1524–1551, 3rd Quart., 2016, doi: 10.1109/COMST.2016.2519148.

[4] Z. Zhu and A. Nandi, "Automatic modulation classification: Principles, algorithms and applications," in *Automatic Modulation Classification: Principles, Algorithms and Applications*. Hoboken, NJ, USA: Wiley, Jan. 2015, pp. 1–163, doi: 10.1002/9781118906507.
[5] C.-F. Teng, C.-Y. Chou, C.-H. Chen, and A.-Y. Wu, "Accumulated polar feature-based deep learning for efficient and lightweight automatic modulation classification with channel compensation mechanism," 2020, *arXiv:2001.01395*. [Online]. Available: http://arxiv.org/abs/2001.01395
[6] F. Meng, P. Chen, L. Wu, and X. Wang, "Automatic modulation classification: A deep learning enabled approach," *IEEE Trans. Veh. Technol.*, vol. 67, no. 11, pp. 10760–10772, Nov. 2018, doi: 10.1109/TVT.2018.2868698.
[7] S. Chen, Y. Zhang, Z. He, J. Nie, and W. Zhang, "A novel attention cooperative framework for automatic modulation recognition," *IEEE Access*, vol. 8, pp. 15673–15686, 2020, doi: 10.1109/ACCESS.2020.2966777.
[8] X. Li, F. Dong, S. Zhang, and W. Guo, "A survey on deep learning techniques in wireless signal recognition," *Wireless Commun. Mobile Comput.*, vol. 2019, pp. 1–12, Feb. 2019, doi: 10.1155/2019/5629572.
[9] R. Zhou, F. Liu, and C. W. Gravelle, "Deep learning for modulation recognition: A survey with a demonstration," *IEEE Access*, vol. 8, pp. 67366–67376, 2020, doi: 10.1109/ACCESS.2020.2986330.
[10] M. Z. Alom, T. M. Taha, C. Yakopcic, S. Westberg, P. Sidike, M. S. Nasrin, M. Hasan, B. C. Van Essen, A. A. S. Awwal, and V. K. Asari, "A state-of-the-art survey on deep learning theory and architectures," *Electronics*, vol. 8, no. 3, p. 292, Mar. 2019, doi: 10.3390/electronics8030292.
[11] E. E. Azzouz and A. K. Nandi, "Automatic identification of digital modulation types," *Signal Process.*, vol. 47, no. 1, pp. 55–69, Nov. 1995. [Online]. Available: http://www.sciencedirect.com/science/article/pii/0165168495000992, doi: 10.1016/0165-1684(95)00099-2.
[12] S. Mallat and S. Zhong, "Characterization of signals from multiscale edges," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 14, no. 7, pp. 710–732, Jul. 1992, doi: 10.1109/34.142909.
[13] R. S. Roberts, W. A. Brown, and H. H. Loomis, "Computationally efficient algorithms for cyclic spectral analysis," *IEEE Signal Process. Mag.*, vol. 8, no. 2, pp. 38–49, Apr. 1991, doi: 10.1109/79.81008.
[14] U. Satija, M. S. Manikandan, and B. Ramkumar, "Performance study of cyclostationary based digital modulation classification schemes," in *Proc. 9th Int. Conf. Ind. Inf. Syst. (ICIIS)*, Dec. 2014, pp. 1–5, doi: 10.1109/ICIINFS.2014.7036609.
[15] J. Jagannath, N. Polosky, D. O'Connor, L. N. Theagarajan, B. Sheaffer, S. Foulke, and P. K. Varshney, "Artificial neural network based automatic modulation classification over a software defined radio testbed," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2018, pp. 1–6, doi: 10.1109/ICC.2018.8422346.
[16] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks," in *Proc. 14th Int. Conf. Artif. Intell. Statist.*, G. Gordon, D. Dunson, and M. Dudík, Eds., vol. 15. Fort Lauderdale, FL, USA: JMLR, Apr. 2011, pp. 315–323. [Online]. Available: http://proceedings.mlr.press/v15/glorot11a.html
[17] G. Klambauer, T. Unterthiner, A. Mayr, and S. Hochreiter, "Self-normalizing neural networks," 2017, *arXiv:1706.02515*. [Online]. Available: http://arxiv.org/abs/1706.02515
[18] R. Collobert and S. Bengio, "Links between perceptrons, MLPs and SVMs," in *Proc. 21st Int. Conf. Mach. Learn. (ICML)*, 2004, p. 23, doi: 10.1145/1015330.1015415.
[19] Z. Wu, S. Zhou, Z. Yin, B. Ma, and Z. Yang, "Robust automatic modulation classification under varying noise conditions," *IEEE Access*, vol. 5, pp. 19733–19741, 2017, doi: 10.1109/ACCESS.2017.2746140.
[20] S. Kharbech, I. Dayoub, M. Zwingelstein-Colin, and E. P. Simon, "Blind digital modulation identification for MIMO systems in railway environments with high-speed channels and impulsive noise," *IEEE Trans. Veh. Technol.*, vol. 67, no. 8, pp. 7370–7379, Aug. 2018, doi: 10.1109/TVT.2018.2834869.
[21] S. Kharbech, I. Dayoub, M. Zwingelstein-Colin, E. P. Simon, and K. Hassan, "Blind digital modulation identification for time-selective MIMO channels," *IEEE Wireless Commun. Lett.*, vol. 3, no. 4, pp. 373–376, Aug. 2014, doi: 10.1109/LWC.2014.2318041.
[22] K. Hassan, I. Dayoub, W. Hamouda, C. N. Nzeza, and M. Berbineau, "Blind digital modulation identification for spatially-correlated MIMO systems," *IEEE Trans. Wireless Commun.*, vol. 11, no. 2, pp. 683–693, Feb. 2012, doi: 10.1109/TWC.2011.122211.110236.
[23] X. Liu, C. Zhao, P. Wang, Y. Zhang, and T. Yang, "Blind modulation classification algorithm based on machine learning for spatially correlated MIMO system," *IET Commun.*, vol. 11, no. 7, pp. 1000–1007, May 2017, doi: 10.1049/iet-com.2015.1222.
[24] A. Güner, Ö. F. Alçin, and A. engür, "Automatic digital modulation classification using extreme learning machine with local binary pattern histogram features," *Measurement*, vol. 145, pp. 214–225, Oct. 2019, doi: 10.1016/j.measurement.2019.05.061.
[25] S. I. H. Shah, S. Alam, S. A. Ghauri, A. Hussain, and F. Ahmed Ansari, "A novel hybrid cuckoo search- extreme learning machine approach for modulation classification," *IEEE Access*, vol. 7, pp. 90525–90537, 2019, doi: 10.1109/ACCESS.2019.2926615.
[26] T. A. Almohamad, M. F. M. Salleh, M. N. Mahmud, and A. H. Y. Sa'D, "Simultaneous determination of modulation types and Signal-to-Noise ratios using feature-based approach," *IEEE Access*, vol. 6, pp. 9262–9271, 2018, doi: 10.1109/ACCESS.2018.2809448.
[27] F. Yang, L. Yang, D. Wang, P. Qi, and H. Wang, "Method of modulation recognition based on combination algorithm of K-means clustering and grading training SVM," *China Commun.*, vol. 15, no. 12, pp. 55–63, Dec. 2018.
[28] D. Huang, C. Shan, M. Ardabilian, Y. Wang, and L. Chen, "Local binary patterns and its application to facial image analysis: A survey," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 41, no. 6, pp. 765–781, Nov. 2011, doi: 10.1109/TSMCC.2011.2118750.
[29] L. Xie and Q. Wan, "Cyclic feature-based modulation recognition using compressive sensing," *IEEE Wireless Commun. Lett.*, vol. 6, no. 3, pp. 402–405, Jun. 2017, doi: 10.1109/LWC.2017.2697853.
[30] X. Yan, G. Liu, H. Wu, G. Zhang, Q. Wang, and Y. Wu, "Robust modulation classification over α-stable noise using graph-based fractional lower-order cyclic spectrum analysis," *IEEE Trans. Veh. Technol.*, vol. 69, no. 3, pp. 2836–2849, Mar. 2020, doi: 10.1109/TVT.2020.2965137.
[31] T. V. R. O. Câmara, A. D. L. Lima, B. M. M. Lima, A. I. R. Fontes, A. D. M. Martins, and L. F. Q. Silveira, "Automatic modulation classification architectures based on cyclostationary features in impulsive environments," *IEEE Access*, vol. 7, pp. 138512–138527, 2019, doi: 10.1109/ACCESS.2019.2943300.
[32] Z. Xing and Y. Gao, "A modulation classification algorithm for multipath signals based on cepstrum," *IEEE Trans. Instrum. Meas.*, vol. 69, no. 7, pp. 4742–4752, Jul. 2020, doi: 10.1109/TIM.2019.2955535.
[33] S. Kharbech, I. Dayoub, M. Zwingelstein-Colin, and E. P. Simon, "On classifiers for blind feature-based automatic modulation classification over multiple-input–multiple-output channels," *IET Commun.*, vol. 10, no. 7, pp. 790–795, May 2016, doi: 10.1049/iet-com.2015.1124.
[34] M. Wei, Z. Wei, J. Yang, and L. Sang, "Automatic modulation recognition of digital signal based on auto-encoding network in MIMO system," in *Proc. IEEE 18th Int. Conf. Commun. Technol. (ICCT)*, Oct. 2018, pp. 1017–1021, doi: 10.1109/ICCT.2018.8600148.
[35] J. Tian, Y. Pei, Y.-D. Huang, and Y.-C. Liang, "A machine learning approach to blind modulation classification for MIMO systems," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2018, pp. 1–6, doi: 10.1109/ICC.2018.8422500.
[36] X. Yan, F. Long, J. Wang, N. Fu, W. Ou, and B. Liu, "Signal detection of MIMO-OFDM system based on auto encoder and extreme learning machine," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, May 2017, pp. 1602–1606, doi: 10.1109/IJCNN.2017.7966042.
[37] T. O'Shea and N. West, "Radio machine learning dataset generation with gnu radio," in *Proc. GNU Radio Conf.*, vol. 1, no. 1, 2016. [Online]. Available: https://pubs.gnuradio.org/index.php/grcon/article/view/11
[38] T. J O'Shea, J. Corgan, and T. C. Clancy, "Convolutional radio modulation recognition networks," 2016, *arXiv:1602.04105*. [Online]. Available: http://arxiv.org/abs/1602.04105
[39] T. J. O'Shea, T. Roy, and T. C. Clancy, "Over-the-air deep learning based radio signal classification," *IEEE J. Sel. Topics Signal Process.*, vol. 12, no. 1, pp. 168–179, Feb. 2018, doi: 10.1109/JSTSP.2018.2797022.
[40] K. Tekbıyık, A. R. Ekti, A. Görçin, G. K. Kurt, and C. Keçeci, "Robust and fast automatic modulation classification with CNN under multipath fading channels," in *Proc. IEEE 91st Veh. Technol. Conf. (VTC-Spring)*, May 2020, pp. 1–6, doi: 10.1109/VTC2020-Spring48590.2020.9128408.
[41] J. Shi, S. Hong, C. Cai, Y. Wang, H. Huang, and G. Gui, "Deep learning-based automatic modulation recognition method in the presence of phase offset," *IEEE Access*, vol. 8, pp. 42841–42847, 2020, doi: 10.1109/ACCESS.2020.2978094.

[42] H. Zhang, Y. Wang, L. Xu, T. A. Gulliver, and C. Cao, "Automatic modulation classification using a deep multi-stream neural network," *IEEE Access*, vol. 8, pp. 43888–43897, 2020, doi: 10.1109/ACCESS.2020.2971698.
[43] H. Gu, Y. Wang, S. Hong, and G. Gui, "Blind channel identification aided generalized automatic modulation recognition based on deep learning," *IEEE Access*, vol. 7, pp. 110722–110729, 2019, doi: 10.1109/ACCESS.2019.2934354.
[44] N. E West and T. J. O'Shea, "Deep architectures for modulation recognition," 2017, *arXiv:1703.09197*. [Online]. Available: http://arxiv.org/abs/1703.09197
[45] Y. Wang, M. Liu, J. Yang, and G. Gui, "Data-driven deep learning for automatic modulation recognition in cognitive radios," *IEEE Trans. Veh. Technol.*, vol. 68, no. 4, pp. 4074–4077, Apr. 2019, doi: 10.1109/TVT.2019.2900460.
[46] C. Yang, Z. He, Y. Peng, Y. Wang, and J. Yang, "Deep learning aided method for automatic modulation recognition," *IEEE Access*, vol. 7, pp. 109063–109068, 2019, doi: 10.1109/ACCESS.2019.2933448.
[47] T. Zhang, C. Shuai, and Y. Zhou, "Deep learning for robust automatic modulation recognition method for IoT applications," *IEEE Access*, vol. 8, pp. 117689–117697, 2020, doi: 10.1109/ACCESS.2020.2981130.
[48] Y. Wang, J. Yang, M. Liu, and G. Gui, "LightAMC: Lightweight automatic modulation classification via deep learning and compressive sensing," *IEEE Trans. Veh. Technol.*, vol. 69, no. 3, pp. 3491–3495, Mar. 2020, doi: 10.1109/TVT.2020.2971001.
[49] S. Zheng, P. Qi, S. Chen, and X. Yang, "Fusion methods for CNN-based automatic modulation classification," *IEEE Access*, vol. 7, pp. 66496–66504, 2019, doi: 10.1109/ACCESS.2019.2918136.
[50] Y. Liu, Y. Liu, and C. Yang, "Modulation recognition with graph convolutional network," *IEEE Wireless Commun. Lett.*, vol. 9, no. 5, pp. 624–627, May 2020, doi: 10.1109/LWC.2019.2963828.
[51] T. O'Shea and J. Hoydis, "An introduction to deep learning for the physical layer," *IEEE Trans. Cognit. Commun. Netw.*, vol. 3, no. 4, pp. 563–575, Dec. 2017, doi: 10.1109/TCCN.2017.2758370.
[52] K. Yashashwi, A. Sethi, and P. Chaporkar, "A learnable distortion correction module for modulation recognition," *IEEE Wireless Commun. Lett.*, vol. 8, no. 1, pp. 77–80, Feb. 2019, doi: 10.1109/LWC.2018.2855749.
[53] J. Nie, Y. Zhang, Z. He, S. Chen, S. Gong, and W. Zhang, "Deep hierarchical network for automatic modulation classification," *IEEE Access*, vol. 7, pp. 94604–94613, 2019, doi: 10.1109/ACCESS.2019.2928463.
[54] T. Huynh-The, C.-H. Hua, Q.-V. Pham, and D.-S. Kim, "MCNet: An efficient CNN architecture for robust automatic modulation classification," *IEEE Commun. Lett.*, vol. 24, no. 4, pp. 811–815, Apr. 2020, doi: 10.1109/LCOMM.2020.2968030.
[55] D. Figueiredo, A. Furtado, and R. Oliveiray, "Modulation classification using joint time and frequency-domain data," in *Proc. IEEE 91st Veh. Technol. Conf. (VTC-Spring)*, May 2020, pp. 1–5, doi: 10.1109/VTC2020-Spring48590.2020.9128493.
[56] A. P. Hermawan, R. R. Ginanjar, D.-S. Kim, and J.-M. Lee, "CNN-based automatic modulation classification for beyond 5G communications," *IEEE Commun. Lett.*, vol. 24, no. 5, pp. 1038–1041, May 2020, doi: 10.1109/LCOMM.2020.2970922.
[57] T. Yao, Y. Chai, S. Wang, X. Miao, and X. Bu, "Radio signal automatic modulation classification based on deep learning and expert features," in *Proc. IEEE 4th Inf. Technol., Netw., Electron. Autom. Control Conf. (ITNEC)*, Jun. 2020, pp. 1225–1230, doi: 10.1109/ITNEC48623.2020.9085077.
[58] K. Bu, Y. He, X. Jing, and J. Han, "Adversarial transfer learning for deep learning based automatic modulation classification," *IEEE Signal Process. Lett.*, vol. 27, pp. 880–884, 2020, doi: 10.1109/LSP.2020.2991875.
[59] T. Chen and C. Guestrin, "XGBoost," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2016, pp. 785–794, doi: 10.1145/2939672.2939785.
[60] M. W. Aslam, Z. Zhu, and A. K. Nandi, "Automatic modulation classification using combination of genetic programming and KNN," *IEEE Trans. Wireless Commun.*, vol. 11, no. 8, pp. 2742–2750, Aug. 2012, doi: 10.1109/TWC.2012.060412.110460.
[61] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778, doi: 10.1109/CVPR.2016.90.
[62] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2261–2269, doi: 10.1109/CVPR.2017.243.
[63] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 5987–5995, doi: 10.1109/CVPR.2017.634.
[64] Y. Sun, J. Li, F. Lin, and G. Pan, "Automatic signal modulation recognition based on deep convolutional neural network," in *Proc. 3rd Int. Conf. Comput. Eng., Inf. Sci. Appl. Technol. (ICCIA)*, 2019, pp. 550–554, doi: 10.2991/iccia-19.2019.86.
[65] N. B. Karayiannis, "Reformulated radial basis neural networks trained by gradient descent," *IEEE Trans. Neural Netw.*, vol. 10, no. 3, pp. 657–671, May 1999, doi: 10.1109/72.761725.
[66] S. Rajendran, W. Meert, D. Giustiniano, V. Lenders, and S. Pollin, "Deep learning models for wireless signal classification with distributed low-cost spectrum sensors," *IEEE Trans. Cognit. Commun. Netw.*, vol. 4, no. 3, pp. 433–445, Sep. 2018, doi: 10.1109/TCCN.2018.2835460.
[67] X. Liu, D. Yang, and A. E. Gamal, "Deep neural network architectures for modulation classification," in *Proc. 51st Asilomar Conf. Signals, Syst., Comput.*, Oct. 2017, pp. 915–919, doi: 10.1109/ACSSC.2017.8335483.
[68] Q. Zhang, Z. Xu, and P. Zhang, "Modulation scheme recognition using convolutional neural network," *J. Eng.*, vol. 2019, no. 23, pp. 9075–9078, Dec. 2019, doi: 10.1049/joe.2018.9188.
[69] Z. Zhang, C. Wang, C. Gan, S. Sun, and M. Wang, "Automatic modulation classification using convolutional neural network with features fusion of SPWVD and BJD," *IEEE Trans. Signal Inf. Process. over Netw.*, vol. 5, no. 3, pp. 469–478, Sep. 2019, doi: 10.1109/TSIPN.2019.2900201.
[70] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*. [Online]. Available: http://arxiv.org/abs/1409.1556
[71] A. Dai, H. Zhang, and H. Sun, "Automatic modulation classification using stacked sparse auto-encoders," in *Proc. IEEE 13th Int. Conf. Signal Process. (ICSP)*, Nov. 2016, pp. 248–252, doi: 10.1109/ICSP.2016.7877834.
[72] J. Zhang, Y. Li, and J. Yin, "A novel modulation classification method for fm signals based on the time-frequency distribution and CNN," *IET Radar, Sonar Navigat.*, vol. 12, no. 12, pp. 244–249, 2017, doi: 10.1049/iet-rsn.2017.0265.
[73] Y. Zeng, M. Zhang, F. Han, Y. Gong, and J. Zhang, "Spectrum analysis and convolutional neural network for automatic modulation recognition," *IEEE Wireless Commun. Lett.*, vol. 8, no. 3, pp. 929–932, Jun. 2019, doi: 10.1109/LWC.2019.2900247.
[74] S. Peng, H. Jiang, H. Wang, H. Alwageed, and Y.-D. Yao, "Modulation classification using convolutional neural network based deep learning model," in *Proc. 26th Wireless Opt. Commun. Conf. (WOCC)*, Apr. 2017, pp. 1–5, doi: 10.1109/WOCC.2017.7929000.
[75] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Commun. ACM*, vol. 60, no. 6, pp. 84–90, May 2017, doi: 10.1145/3065386.
[76] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," in *Proc. 22nd ACM Int. Conf. Multimedia*, Nov. 2014, pp. 675–678, doi: 10.1145/2647868.2654889.
[77] B. Tang, Y. Tu, Z. Zhang, and Y. Lin, "Digital signal modulation classification with data augmentation using generative adversarial nets in cognitive radio networks," *IEEE Access*, vol. 6, pp. 15713–15722, 2018, doi: 10.1109/ACCESS.2018.2815741.
[78] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," 2014, *arXiv:1406.2661*. [Online]. Available: http://arxiv.org/abs/1406.2661
[79] M. Zhang, M. Diao, and L. Guo, "Convolutional neural networks for automatic cognitive radio waveform recognition," *IEEE Access*, vol. 5, pp. 11074–11082, 2017, doi: 10.1109/ACCESS.2017.2716191.
[80] G. Jajoo, Y. Kumar, and S. K. Yadav, "Blind signal PSK/QAM recognition using clustering analysis of constellation signature in flat fading channel," *IEEE Commun. Lett.*, vol. 23, no. 10, pp. 1853–1856, Oct. 2019, doi: 10.1109/LCOMM.2019.2929127.
[81] Y. Tu and Y. Lin, "Deep neural network compression technique towards efficient digital signal modulation recognition in edge device," *IEEE Access*, vol. 7, pp. 58113–58119, 2019, doi: 10.1109/ACCESS.2019.2913945.
[82] Q. Zhang, G. Hu, P. Zhao, and L. Yang, "Modulation recognition of 5G signals based on AlexNet convolutional neural network," *J. Phys., Conf. Ser.*, vol. 1453, Jan. 2020, Art. no. 012118, doi: 10.1088/1742-6596/1453/1/012118.

[83] K. Jiang, J. Zhang, H. Wu, A. Wang, and Y. Iwahori, "A novel digital modulation recognition algorithm based on deep convolutional neural network," *Appl. Sci.*, vol. 10, no. 3, p. 1166, Feb. 2020, doi: 10.3390/app10031166.

[84] X. Zha, H. Peng, X. Qin, G. Li, and S. Yang, "A deep learning framework for signal detection and modulation classification," *Sensors*, vol. 19, no. 18, p. 4042, Sep. 2019, doi: 10.3390/s19184042.

[85] A. Ivanov, K. Tonchev, V. Poulkov, and A. Manolova, "Deep learning for modulation classification: Signal features in performance analysis," in *Proc. Joint Int. Conf. Digit. Arts, Media Technol. ECTI Northern Sect. Conf. Electr., Electron., Comput. Telecommun. Eng. (ECTI DAMT NCON)*, Mar. 2020, pp. 162–167, doi: 10.1109/ECTIDAMT-NCON48261.2020.9090764.

[86] N. Daldal, Z. Cömert, and K. Polat, "Automatic determination of digital modulation types with different noises using convolutional neural network based on time–frequency information," *Appl. Soft Comput.*, vol. 86, Jan. 2020, Art. no. 105834, doi: 10.1016/j.asoc.2019.105834.

[87] D. Wang, M. Zhang, Z. Li, J. Li, M. Fu, Y. Cui, and X. Chen, "Modulation format recognition and OSNR estimation using CNN-based deep learning," *IEEE Photon. Technol. Lett.*, vol. 29, no. 19, pp. 1667–1670, Oct. 1, 2017, doi: 10.1109/LPT.2017.2742553.

[88] R. Li, L. Li, S. Yang, and S. Li, "Robust automated VHF modulation recognition based on deep convolutional neural networks," *IEEE Commun. Lett.*, vol. 22, no. 5, pp. 946–949, May 2018, doi: 10.1109/LCOMM.2018.2809732.

[89] M. Zhang, Y. Zeng, Z. Han, and Y. Gong, "Automatic modulation recognition using deep learning architectures," in *Proc. IEEE 19th Int. Workshop Signal Process. Adv. Wireless Commun. (SPAWC)*, Jun. 2018, pp. 1–5, doi: 10.1109/SPAWC.2018.8446021.

[90] M. H. Shah and X. Dang, "Novel feature selection method using Bhattacharyya distance for neural networks based automatic modulation classification," *IEEE Signal Process. Lett.*, vol. 27, pp. 106–110, 2020, doi: 10.1109/LSP.2019.2957924.

[91] C. Teng, C. Liao, C. Chen, and A. A. Wu, "Polar feature based deep architectures for automatic modulation classification considering channel fading," in *Proc. IEEE Global Conf. Signal Inf. Process. (GlobalSIP)*, Nov. 2018, pp. 554–558, doi: 10.1109/GlobalSIP.2018.8646375.

[92] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*. [Online]. Available: http://arxiv.org/abs/1412.6980

[93] M. C. Mukkamala and M. Hein, "Variants of RMSProp and adagrad with logarithmic regret bounds," 2017, *arXiv:1706.05507*. [Online]. Available: http://arxiv.org/abs/1706.05507

[94] S. Peng, H. Jiang, H. Wang, H. Alwageed, Y. Zhou, M. M. Sebdani, and Y.-D. Yao, "Modulation classification based on signal constellation diagrams and deep learning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 3, pp. 718–727, Mar. 2019, doi: 10.1109/TNNLS.2018.2850703.

[95] A. Swami and B. M. Sadler, "Hierarchical digital modulation classification using cumulants," *IEEE Trans. Commun.*, vol. 48, no. 3, pp. 416–429, Mar. 2000, doi: 10.1109/26.837045.

[96] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet large scale visual recognition challenge," 2014, *arXiv:1409.0575*. [Online]. Available: http://arxiv.org/abs/1409.0575

[97] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 1–9, doi: 10.1109/CVPR.2015.7298594.

[98] X. Yan, G. Feng, H.-C. Wu, W. Xiang, and Q. Wang, "Innovative robust modulation classification using graph-based cyclic-spectrum analysis," *IEEE Commun. Lett.*, vol. 21, no. 1, pp. 16–19, Jan. 2017, doi: 10.1109/LCOMM.2016.2618868.

[99] J. A. Sills, "Maximum-likelihood modulation classification for PSK/QAM," in *Proc. MILCOM IEEE Mil. Commun., Conf.*, Oct. 1999, pp. 217–220, doi: 10.1109/MILCOM.1999.822675.

[100] P. Panagiotou, A. Anastasopoulos, and A. Polydoros, "Likelihood ratio tests for modulation classification," in *Proc. MILCOM 21st Century Mil. Commun., Archit. Technol. Inf. Superiority*, Oct. 2000, pp. 670–674, doi: 10.1109/MILCOM.2000.904013.

[101] V. Orlic and M. Dukic, "Automatic modulation classification algorithm using higher-order cumulants under real-world channel conditions," *IEEE Commun. Lett.*, vol. 13, no. 12, pp. 917–919, Dec. 2009, doi: 10.1109/LCOMM.2009.12.091711.

[102] L. Huang, W. Pan, Y. Zhang, L. Qian, N. Gao, and Y. Wu, "Data augmentation for deep learning-based radio modulation classification," 2019, *arXiv:1912.03026*. [Online]. Available: http://arxiv.org/abs/1912.03026

[103] Y. Guo, H. Jiang, J. Wu, and J. Zhou, "Open set modulation recognition based on dual-channel LSTM model," 2020, *arXiv:2002.12037*. [Online]. Available: http://arxiv.org/abs/2002.12037

[104] L. Huang, Y. Zhang, W. Pan, J. Chen, L. Ping Qian, and Y. Wu, "Visualizing deep learning-based radio modulation classifier," 2020, *arXiv:2005.02175*. [Online]. Available: http://arxiv.org/abs/2005.02175

[105] S. Hanna, C. Dick, and D. Cabric, "Combining deep learning and linear processing for modulation classification and symbol decoding," 2020, *arXiv:2006.00729*. [Online]. Available: http://arxiv.org/abs/2006.00729

[106] W. Xie, J. Xiao, J. Yang, X. Peng, C. Yu, and P. Zhu, "Deep learning-based modulation detection for NOMA systems," 2020, *arXiv:2005.11684*. [Online]. Available: http://arxiv.org/abs/2005.11684

[107] Z. Ding, Z. Yang, P. Fan, and H. V. Poor, "On the performance of non-orthogonal multiple access in 5G systems with randomly deployed users," *IEEE Signal Process. Lett.*, vol. 21, no. 12, pp. 1501–1505, Dec. 2014, doi: 10.1109/LSP.2014.2343971.

[108] W. Shi, D. Liu, X. Cheng, Y. Li, and Y. Zhao, "Particle swarm optimization-based deep neural network for digital modulation recognition," *IEEE Access*, vol. 7, pp. 104591–104600, 2019, doi: 10.1109/ACCESS.2019.2932266.

[109] W. Xie, S. Hu, C. Yu, P. Zhu, X. Peng, and J. Ouyang, "Deep learning in digital modulation recognition using high order cumulants," *IEEE Access*, vol. 7, pp. 63760–63766, 2019, doi: 10.1109/ACCESS.2019.2916833.

[110] F. N. Khan, K. Zhong, W. H. Al-Arashi, C. Yu, C. Lu, and A. P. T. Lau, "Modulation format identification in coherent receivers using deep machine learning," *IEEE Photon. Technol. Lett.*, vol. 28, no. 17, pp. 1886–1889, Sep. 1, 2016, doi: 10.1109/LPT.2016.2574800.

[111] B. Kim, J. Kim, H. Chae, D. Yoon, and J. W. Choi, "Deep neural network-based automatic modulation classification technique," in *Proc. Int. Conf. Inf. Commun. Technol. Converg. (ICTC)*, Oct. 2016, pp. 579–582, doi: 10.1109/ICTC.2016.7763537.

[112] J. B. B. Gouho, "Automatic modulation classification based on in-phase quadrature diagram constellation combined with a deep learning model," *Indian J. Sci. Technol.*, vol. 13, no. 2, pp. 200–212, Mar. 2020, doi: 10.17485/ijst/2020/v13i02/148648.

[113] H. Ma, G. Xu, H. Meng, M. Wang, S. Yang, R. Wu, and W. Wang, "Cross model deep learning scheme for automatic modulation classification," *IEEE Access*, vol. 8, pp. 78923–78931, 2020, doi: 10.1109/ACCESS.2020.2988727.

[114] G. J. Mendis, J. Wei, and A. Madanayake, "Deep learning-based automated modulation classification for cognitive radio," in *Proc. IEEE Int. Conf. Commun. Syst. (ICCS)*, Dec. 2016, pp. 1–6, doi: 10.1109/ICCS.2016.7833571.

[115] A. Fehske, J. Gaeddert, and J. H. Reed, "A new approach to signal classification using spectral correlation and neural networks," in *Proc. 1st IEEE Int. Symp. New Frontiers Dyn. Spectr. Access Netw. (DySPAN)*, Nov. 2005, pp. 144–150, doi: 10.1109/DYSPAN.2005.1542629.

[116] W. C. Headley, J. D. Reed, and C. R. C. M. da Silva, "Distributed cyclic spectrum feature-based modulation classification," in *Proc. IEEE Wireless Commun. Netw. Conf.*, Mar. 2008, pp. 1200–1204, doi: 10.1109/WCNC.2008.216.

[117] H. Wang, Z. Wu, S. Ma, S. Lu, H. Zhang, G. Ding, and S. Li, "Deep learning for signal demodulation in physical layer wireless communications: Prototype platform, open dataset, and analytics," *IEEE Access*, vol. 7, pp. 30792–30801, 2019, doi: 10.1109/ACCESS.2019.2903130.

[118] J. Li, L. Qi, and Y. Lin, "Research on modulation identification of digital signals based on deep learning," in *Proc. IEEE Int. Conf. Electron. Inf. Commun. Technol. (ICEICT)*, Aug. 2016, pp. 402–405, doi: 10.1109/ICEICT.2016.7879726.

[119] M. Shah and X. Dang, "A robust approach for AMC in frequency selective fading scenarios using unsupervised sparse autoencoder based deep neural network (SAE-DNN)," *IET Commun.*, vol. 13, pp. 423–432, Nov. 2018, doi: 10.1049/iet-com.2018.5688.

[120] Z. Zhang, Z. Hua, and Y. Liu, "Modulation classification in multipath fading channels using sixth-order cumulants and stacked convolutional auto-encoders," *IET Commun.*, vol. 11, no. 6, pp. 910–915, Apr. 2017, doi: 10.1049/iet-com.2016.0533.

[121] V. Bhatia, M. R. Tripathy, and P. Ranjan, "Deep learning for massive MIMO: Challenges and future prospects," in *Proc. IEEE 9th Int. Conf. Commun. Syst. Netw. Technol. (CSNT)*, Apr. 2020, pp. 26–31, doi: 10.1109/CSNT48778.2020.9115783.

[122] J. Xia, K. He, W. Xu, S. Zhang, L. Fan, and G. K. Karagiannidis, "A MIMO detector with deep learning in the presence of correlated interference," *IEEE Trans. Veh. Technol.*, vol. 69, no. 4, pp. 4492–4497, Apr. 2020, doi: 10.1109/TVT.2020.2972806.

[123] Y. Wang, J. Wang, W. Zhang, J. Yang, and G. Gui, "Deep learning-based cooperative automatic modulation classification method for MIMO systems," *IEEE Trans. Veh. Technol.*, vol. 69, no. 4, pp. 4575–4579, Apr. 2020, doi: 10.1109/TVT.2020.2976942.

[124] Y. Wang, J. Gui, Y. Yin, J. Wang, J. Sun, G. Gui, H. Gacanin, H. Sari, and F. Adachi, "Automatic modulation classification for MIMO systems via deep learning and zero-forcing equalization," *IEEE Trans. Veh. Technol.*, vol. 69, no. 5, pp. 5688–5692, May 2020, doi: 10.1109/TVT.2020.2981995.

[125] Y. Wang, G. Gui, H. Gacanin, T. Ohtsuki, H. Sari, and F. Adachi, "Transfer learning for semi-supervised automatic modulation classification in ZF-MIMO systems," *IEEE J. Emerg. Sel. Topics Circuits Syst.*, vol. 10, no. 2, pp. 231–239, Jun. 2020, doi: 10.1109/JETCAS.2020.2992128.

[126] Y. Liu and Y. Liu, "Modulation recognition with pre-denoising convolutional neural network," *Electron. Lett.*, vol. 56, no. 5, pp. 255–257, Mar. 2020, doi: 10.1049/el.2019.3586.

[127] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 3431–3440, doi: 10.1109/CVPR.2015.729896.

[128] M. Lin, Q. Chen, and S. Yan, "Network in network," 2013, *arXiv:1312.4400*. [Online]. Available: http://arxiv.org/abs/1312.4400

[129] A. Murad and J.-Y. Pyun, "Deep recurrent neural networks for human activity recognition," *Sensors*, vol. 17, no. 11, p. 2556, Nov. 2017, doi: 10.3390/s17112556.

[130] F. Karim, S. Majumdar, H. Darabi, and S. Chen, "LSTM fully convolutional networks for time series classification," *IEEE Access*, vol. 6, pp. 1662–1669, 2018, doi: 10.1109/ACCESS.2017.2779939.

BACHIR JDID received the B.Eng.Sc. degree in communications engineering from the Higher Institute for Applied Science and Technology, Damascus, Syria, in 2013. He is currently pursuing the M.Phil. degree with the Faculty of Engineering, Technology and Built Environment, UCSI University, Kuala Lumpur, Malaysia. His research interests include wireless communications, signal processing, image processing, embedded systems, and artificial intelligence and its application in wireless communications.

KAIS HASSAN was born in 1975. He graduated in telecommunications engineering from the Higher Institute for Applied Science and Technology (HIAST), Syria, in 2000. He received the M.S. degree in telecommunications systems and the Ph.D. degree in telecommunications from the University of Valenciennes, France, in 2009 and 2012, respectively. From 2000 to 2008, he was a Research Engineer with HIAST. From 2013 to 2014, he was an Assistant Professor with the University of Western Brittany, France. Since September 2014, he has been an Associate Professor with Le Mans University, France. He is currently with the Acoustics Laboratory of Le Mans University (LAUM). His research interests include wireless communications, signal processing, and cognitive radio.

IYAD DAYOUB (Senior Member, IEEE) received the B.Eng. degree in telecommunications and electronics from Syria, in 1993, the M.A.Sc. degree in electrical engineering from the National Polytechnic Institute of Lorraine (INPL), and the Ph.D. degree from the Institute of Electronics, Microelectronics and Nanotechnology (IEMN), University of Valenciennes, in 2001. He has worked as a System Engineer with Siemens (Middle East) and as a Researcher with Alcatel Business Systems Alcatel, Colombes, Paris. From 2007 to 2014, he was a member of the National Council of Universities (CNU), France, in the area of electrical engineering, electronics, photonics and systems. From 2010 to 2014, he was an Adjunct Professor with Concordia University, Montreal. He is currently a Professor of communications engineering. His current research activities at the IEMN of Université Polytechnique Hauts de France (UPHF) & INSA H-d-F are focused on wireless communications, high-speed communications, cognitive radio, and hybrid radio- optic technologies. He is a member of several International Conference Advisory Committees, Technical Program Committees and Organization Committees, such as VTC, GLOBECOM, ICC, PIMRC, WWC, and so on.

WEI HONG LIM (Senior Member, IEEE) received the B.Eng. degree (Hons.) in mechatronic engineering and the Ph.D. degree in computational intelligence from the Universiti Sains Malaysia, Penang, Malaysia, in 2011 and 2014, respectively. From 2015 to 2017, he was a Postdoctoral Researcher with the Intelligent Control Laboratory, National Taipei University of Technology, Taiwan, where he was a Visiting Researcher, in 2019. He is currently an Assistant Professor with the Faculty of Engineering, Technology and Built Environment, UCSI University. He is also working with three national research grants awarded by the Ministry of Education, Malaysia, and five internal grant projects supported by UCSI University. He has published more than 50 research articles in research areas related to computational intelligence, optimization algorithms, energy management, and digital image processing. He has been actively involved in various professional bodies and registered as the European Engineer (EUR ING) with the Fédération Européenne d'Associations Nationales d'Ingénieurs (FEANI), a Chartered Engineer (CEng) with the Engineering Council, U.K., (ECUK) and a Professional Technologist (PTech) with the Malaysia Board of Technologist (MBOT). He is also an Active Reviewer for various reputable journals, such as IEEE ACCESS, *Complexity*, *Mathematical Problem in Engineering*, and *Computational Intelligence and Neuroscience*.

MASTANEH MOKAYEF received the master's degree from the Faculty of Engineering, University Technology Malaysia, in 2009, and the Ph.D. degree from the Wireless Communication Centre, Faculty of Electrical Engineering, University Technology Malaysia (UTM), in 2014. Since 2015, she has been working with UCSI University, Malaysia, where she is currently works as an Assistant Professor with the Faculty of Engineering and Built Environment (FETBE). Her research interests include wireless communications, spectrum sharing method, spectrum management, cellular communication systems and antenna design. Since 2017, she has been a member of Board of Engineers Malaysia (BEM).