Deep Learning Based Automatic Modulation Recognition: Models, Datasets, and Challenges
Fuxin Zhang, Chunbo Luo, Jialang Xu, Yang Luo and FuChun Zheng

ARTICLE INFO
Keywords:
Automatic modulation recognition
Deep learning
Neural networks
Modulation

ABSTRACT
Automatic modulation recognition (AMR) detects the modulation scheme of the received signals for further signal processing without needing prior information, and provides the essential function when such information is missing. Recent breakthroughs in deep learning (DL) have laid the foundation for developing high-performance DL-AMR approaches for communications systems. Comparing with traditional modulation detection methods, DL-AMR approaches have achieved promising performance including high recognition accuracy and low false alarms due to the strong feature extraction and classification abilities of deep neural networks. Despite the promising potential, DL-AMR approaches also bring concerns to complexity and explainability, which affect the practical deployment in wireless communications systems. This paper aims to present a review of the current DL-AMR research, with a focus on appropriate DL models and benchmark datasets. We further provide comprehensive experiments to compare the state of the art models for single-input-single-output (SISO) systems from both accuracy and complexity perspectives, and propose to apply DL-AMR in the new multiple-input-multiple-output (MIMO) scenario with precoding. Finally, existing challenges and possible future research directions are discussed.

1. Introduction
Automatic modulation recognition (AMR) provides essential modulation information of the incoming radio signals, especially non-cooperative radio signals, and plays a key role in various scenarios including cognitive radio, spectrum sensing, signal surveillance, interference identification, etc [3, 9, 90]. It aims to detect the modulation scheme of wireless communications signals automatically without prior information, and has attracted significant research interest in recent years [51]. During the transmission process, signals generated by the transmitter are usually reshaped by adversarial factors within the radio frequency chain, such as noise, multi-path fading, shadow fading, center frequency offset, and sample rate offset [51]. The structural characteristics of these signals may also be distorted due to poor hardware design or crystal oscillator drifting, causing difficulty to distinguish different modulation schemes. As an important step between signal detection and demodulation, AMR provides the essential function to detect modulation schemes. With the rapid development of wireless communications technologies, the modulation schemes of signals will become more complex and diverse to meet the needs of increasingly complex communications scenarios, so there is an urgent need to design effective AMR models that are robust in harsh radio environments.

Traditional research on AMR can be grouped into two categories: likelihood theory-based AMR (LB-AMR) [12, 74, 80] and feature-based AMR (FB-AMR) [17]. LB-AMR methods often receive optimal recognition accuracy in the sense of Bayesian estimation, but they have high computational complexity. FB-AMR methods mainly focus on learning representative features from training samples and classifying incoming signals using the trained models. Typical types of features used in FB-AMR include instantaneous time-domain features [4], transform domain features [22], and statistical features [43, 62]. Machine learning models are increasingly employed for classification, including artificial neural network [76, 81], decision tree [83], and support vector machine [53], etc. By contrast, FB-AMR methods usually yield sub-optimal solutions but with low computational complexity and multiple-modulation identification capability.
Deep learning (DL) has achieved breakthroughs in a range of challenging applications where traditional methods struggle to achieve promising performance, such as natural language processing and image processing. The strong feature extraction capability empowered by stacked multi-layers of artificial neurons has inspired extended research in the conquest of modulation detection, where some pioneering DL based methods have been proposed, often out-performing traditional LB- and FB- AMR approaches [51, 56]. Furthermore, existing research on DL-AMR mainly focuses on single-input-single-output (SISO) systems while there is increasing coverage on DL-AMR for multiple-input-multiple-output (MIMO) systems recently. DL-AMR for MIMO would provide new perspectives in this rapidly developing field. We present our taxonomy of existing DL-AMR research in Fig. 1.

![Taxonomy of existing DL-AMR research.](image)

This paper aims to provide a review of current DL-AMR approaches, outline the key challenges of this field, and shed light on future promising opportunities. We first introduce the general framework of DL-AMR and systematically analyze the existing DL-AMR models for SISO systems. The benchmark AMR datasets for SISO systems are summarized with a focus on their key characteristics. We further provide an overview of DL-AMR for MIMO systems, and then propose to apply AMR in the MIMO communications system containing precoding. In addition, extensive experiments with the main state-of-the-art DL-AMR models showed detailed comparisons for practitioners to assess the different models according to the applications (source code can be found at https://github.com/Richardzhangxx after publication). Finally, we discuss the remaining challenges of AMR and future research opportunities.

Note that this paper is conducted in parallel and independent from the recently published survey papers (e.g., [30] and [31]). We analyze the models, datasets, and challenges for AMR from different perspectives including the latest proposed models, and experimentally compare the most representative and state-of-the-art models in AMR field under a unified platform with the same parameter settings, providing a comprehensive reference for future research on AMR. The feasibility of applying DL-AMR in the MIMO system with precoding is also experimentally verified, showing that SISO’s method can be transferred to MIMO systems.
2. DL models for AMR in SISO systems

After the signal passes through the channel and is sampled, the equivalent baseband signal can be expressed by:

\[ y[l] = A[l]e^{j(\omega l + \phi)} x[l] + n[l], l = 1, \ldots, L, \]  

where \( x[l] \) is the signal modulated by the transmitter in a certain modulation scheme, \( n[l] \) denotes the complex Additive Gaussian Noise (AWGN), \( A[l] \) represents the channel gain, \( \omega \) is the frequency offset, \( \phi \) is the phase offset, \( y[l] \) denotes the \( l \)-th value observed by the receiver and \( L \) is the number of symbols in a signal sample. To facilitate data processing and modulation recognition, the received signals can be stored in in-phase/quadrature (I/Q) form, denoted as \( y = [\Re\{y[1]\}, \ldots, \Re\{y[L]\}; \Im\{y[1]\}, \ldots, \Im\{y[L]\}] \). Modulation recognition needs to identify the modulation scheme of the transmitted signal \( x[l] \) from \( y[l] \), while the structural characteristics of the signals may have been distorted.

![Diagram](image)

**Figure 2:** The general structure of DL-AMR.

DL-AMR generally contains three steps, including preprocessing of the received signals, feature extraction, and modulation classification, as shown in Fig. 2. Preprocessing is used to adapt the data format of the signals for subsequent feature extraction and training. Feature extraction and feature classification can be completed by a deep neural network in an end-to-end fashion or divided into two steps, where features are extracted first by traditional methods in the preprocessing part and then used to train a classification model. The recent representative works on SISO-AMR using deep learning are summarized in Table 1, and the representative DL-AMR model for each neural network type is shown in Fig 3.

### 2.1. Unsupervised models

Unsupervised models like deep belief networks (DBN) and autoencoder have been applied for AMR. DBN models consist of multiple layers of restricted Boltzmann machines (RBM), which represent a probabilistic generation method. The authors of [47, 48] integrate spectral correlation function (SCF) and DBN to learn complex patterns effectively. Furthermore, they extend their preliminary work by implementing it and analyzing the reduction of logic utilization on FPGA, further demonstrating the effectiveness of this method. DBN has a greedy layer-by-layer unsupervised training procedure, which is too complex to be effectively applied to large-scale learning problems. Due to the high computational complexity of DBN and the development of other deep learning methods, DBN models are gradually losing popularity in AMR. In [1], a low complexity classifier has been proposed to use the \( k \)-sparse autoencoder to extract features of signals and then classify their modulation schemes with the features extracted. The overall complexity has been reduced by using only \( k \) largest hidden nodes to reconstruct the input rather than using all the hidden nodes, while keeping the classification accuracy to a near-optimal level. The author further proposes a nonnegativity constraint training algorithm in [2] to minimize the reconstruction error, which takes fourth-order cumulants as input instead of in-phase/quadrature (I/Q) data in [1]. Another form of input called ambiguity function (AF) is adopted in [8], which inputs \( 28 \times 28 \) AF images into sparse auto-encoders (SAE) for modulation recognition. However, most unsupervised deep learning methods have only been tested to classify a few modulation schemes, and their classification performance needs to be generalized to more practical applications.
2.2. DNN-based models

The deep neural network (DNN) model has the structure of a fully connected feedforward network with several hidden layers. Due to the capability of DNN to extract the complex feature in high dimensional space, a four-layer DNN model is adopted in [35] to classify the modulation schemes from various statistical features, e.g., high-order cumulants, kurtosis, and skewness. Different modulation recognition schemes are considered in [78], where six high-order cumulants are used as input and a similar DNN structure is used for modulation recognition. To avoid selecting the number of neural network nodes manually, [61] introduces the particle swarm optimization (PSO) scheme to optimize the number of hidden layer nodes, which can improve recognition accuracy compared to traditional DNN.

2.3. CNN-based models

Convolutional neural network (CNN) models have shown exceptional capability in processing data that exhibit spatial characteristics, e.g., image segmentation, object detection and classification. Researches on AMR have introduced CNNs to detect the signal modulation schemes by utilizing their spatial feature extraction power. Depending on the input data types, existing CNN-based AMR approaches can be grouped into two categories: CNN models with raw I/Q inputs and CNN models with pre-processed inputs. Another category is efficient CNN architecture design, which is to meet the latency and complexity requirements in advanced communications systems. These categories are analyzed below.

1) Typical CNN models with raw I/Q data inputs: By using a simple four-layer CNN model and taking I/Q data as inputs, the potential of CNN applied to AMR is firstly explored in [51] which achieves higher recognition accuracy than many traditional methods. Authors of [64] further propose updated CNN models by adapting the number of layers or hyper-parameters. Because these simple CNN model structures may not fully extract representative features from...
the I/Q input signals, a few improved CNN models that utilize a mixture of complex layers or transformation operations have since been applied in AMR. For example, inspired by the winner architecture of the ImageNet 2015 competition, new DL-AMR models based on residual neural network (ResNet) and densely connected network (DenseNet) [44] are proposed to allow features learned from multiple layers to be effectively transmitted to the detection module. A real-time signal modulation classification model is presented in [87], which identifies OFDM modulation signals by a triple-skip residual neural network (TRNN). However, their improvements in recognition accuracy come with the significantly increased cost of computational complexity. Since many CNN-based AMR models directly adopt methods proposed for computer vision tasks, they neglect the intrinsic signal characteristics in communications.

Moreover, constellation diagram is another widely used signal representation method for AMR, and the performance under non-Gaussian noise, and ResNet is then used for classification. In [20], I/Q and the discrete orthonormal Stockwell transform (DOST) of I/Q sequences are combined to represent the received signal for CNN-based modulation recognition. In [87], a real-time signal modulation classification model is presented, which identifies OFDM modulation signals by a triple-skip residual neural network (TRNN). However, their improvements in recognition accuracy come with the significantly increased cost of computational complexity. Since many CNN-based AMR models directly adopt methods proposed for computer vision tasks, they neglect the intrinsic signal characteristics in communications.

By addressing such limitations, the authors of [82] propose to estimate the carrier frequency offsets and phase noise of the received signals before feeding the data into CNN models, by applying a trainable function that calculates the phase and frequency offsets (correction parameters) from the received signals which are corrected by reducing the noise of the received signals before feeding the data into CNN models, by applying a trainable function that calculates the phase and frequency offsets (correction parameters) from the received signals which are corrected by reducing the offsets.

2) Typical CNN models with pre-processed inputs: CNN models taking inputs of raw I/Q signals, which are the results of multiple processing modules through the RF chain coupled with channel effects, if fed into the model directly, would miss important characteristics that could be easily extracted by classical signal processing methods, e.g., high-order cumulants, spectrum images, and constellation diagrams, etc. Combining traditional FB-AMR methods with CNN models has shown significant potential to overcome such limitations. For example, a short-time discrete Fourier transform is used in [85] to transform one-dimensional radio signals into spectrum images, and then a Gaussian filter is added to reduce noise. The proposed SCNN model has achieved better recognition accuracy than the benchmark CNN1 [51] models, demonstrating the effectiveness of this strategy. The Bispectrum-Alexnet method proposed in [38] feeds the CNN with amplitude spectra of bispectrum (ASB) since bispectrum can suppress the effect of white noise. Another spectrum called cyclic correntropy spectrum (CCES) is introduced in [46] to improve the recognition performance under non-Gaussian noise, and ResNet is then used for classification. In [20], I/Q and the discrete orthonormal Stockwell transform (DOST) of I/Q sequences are combined to represent the received signal for CNN-based modulation recognition. Moreover, constellation diagram is another widely used signal representation method for AMR, and the
features could be extracted directly from the constellation diagram to determine the modulation scheme. Constellation diagram is converted into 3-channel image in [54] to leverage the colored images processing ability of CNN models, which is further classified by AlexNet and GoogLeNet models. Other representative CNN models, e.g., ResNet50 and Inception ResNet V2 are also applied in [33], which transfer constellation points to constellation density matrix (CDM) and mask the CDM with a proper filter to remove information corrupted by high noise. Note that the confusion between 16QAM and 64QAM is very severe, [71] uses two CNN models to improve the recognition accuracy, and trains two models by using I/Q data and constellation diagrams respectively. The CNN based on constellation diagrams is mainly used to solve the problem of confusion between 16QAM and 64QAM. The cross-entropy loss function is the most commonly used loss function in AMR, perhaps it is not the most suitable loss function for all DL-AMR models. On this basis, the authors propose to use contrastive loss [27] and compressive loss [26] as the loss function to enhance the discrepancy of the extracted representations among different modulations, and the corresponding models utilize grid constellation matrix (GCM) and multiple constellation images as input, respectively. To leverage the strong noise resistibility of cyclic spectra image and high-order modulation recognition capability brought by constellation diagram, a two-branch CNN is built in [77] to extract multi-features and then fuse them together. Also based on the idea of feature fusion, [93] applied a multimodality fusion model, which fuses different image features extracted by CNNs and eight handcrafted features, to obtain more discriminating features. Some other input signal representations, such as eye diagrams [67], feature point (FP) images [36], and square feature matrix [7] are also used as input to the DL-AMR model.

3) Efficient CNN architecture design: In order to reduce the projected latency in beyond fifth-generation (B5G) communications systems, an improved CNN-based AMR model [19] is developed by adopting a small number of filters in each layer and reducing the overall number of trainable parameters. The processing time of the model can be less than 0.01 ms to comply with the B5G communications requirement. Despite the reduced number of trainable parameters in this model, it can still maintain a high recognition accuracy. To meet the baseline requirements of 5G services in terms of ultra-reliability and low-latency, a cost-efficient and high-performed CNN is proposed in [28] to effectively classify the modulations schemes, which adopts various asymmetric kernels in parallel in each convolutional block and skip connections in a block-wise manner throughout the network. Another more lightweight model is proposed in [60], which introduces the channel attention mechanism by employing the squeeze-and-excitation (SE) block.

2.4. RNN-based models

In addition to the spatial correlation characteristics, wireless communications signals also carry temporal correlation features that can be learned by machine learning models such as recurrent neural network (RNN) for modulation recognition. A few novel model architectures based on RNN have been proposed recently and obtained state-of-the-art performance in AMR. For example, a new AMR method based on RNN is proposed in [21] which uses gated recurrent units (GRU) to obtain better recognition accuracy than a few CNN models. The author of [56] transforms the I/Q signals into amplitude and phase, and inputs them into an LSTM which also achieves high recognition accuracy. The models using only two RNN layers [21, 56] have already demonstrated outstanding accuracy performance in AMR tasks, confirming the distinct temporal features of communications signals and the strong feature extraction ability of RNNs. Another LSTM-based model proposed in [23], which incorporates fully connected layers and a temporal attention mechanism, is proved to be robust to uncertain noise conditions. An LSTM denoising auto-encoder is designed in [32] to automatically infer modulation schemes using a compact RNN architecture readily implemented on a low-cost computational platform while exceeding state-of-the-art accuracy in [56]. Benefitting from the respective advantages of RNN and CNN, researchers have further proposed hybrid models combining RNN and CNN to further improve the AMR performance, as shown below.

2.5. Hybrid models

Since pure CNN or RNN models focus on either the spatial features or temporal features of AMR signals, only using one type of them may not achieve optimal performance. Therefore, researchers have proposed to combine the characteristics of both types of neural network layers to build hybrid models for AMR. A convolutional long-short-term deep neural networks (CLDNN) model has been proposed in [75] consisting of one LSTM and three CNN layers. This model has a skip connection before LSTM that bypasses two CNN layers, which are intended to provide a longer time context for the extracted features. The output of the first convolution layer is concatenated with the output of the third convolution layer before feeding into the LSTM and then the LSTM layer can extract more effective temporal features which can yield better recognition accuracy and more stable gradient descent than other architectures such as CNN1.
Table 2
Main AMR open datasets for SISO systems.

<table>
<thead>
<tr>
<th>Dataset Name</th>
<th>Modulation Schemes</th>
<th>Sample Dimension</th>
<th>Dataset Size</th>
<th>SNR Range(dB)</th>
<th>Characteristics</th>
</tr>
</thead>
<tbody>
<tr>
<td>RML 2016.04c</td>
<td>11 classes (BPSK, BPSK, CPSK, GFSK, PAM4, AM-DSB, AM-SSB, 16QAM, 64QAM, QPSK, WBFM)</td>
<td>2*128</td>
<td>162060</td>
<td>-20:2:18</td>
<td>The original version of the AMR dataset, relatively small.</td>
</tr>
<tr>
<td>RML 2016.10a</td>
<td>11 classes (BPSK, BPSK, CPSK, GFSK, PAM4, 16QAM, AM-DSB, AM-SSB, 64QAM, QPSK, WBFM)</td>
<td>2*128</td>
<td>220000</td>
<td>-20:2:18</td>
<td>A cleaner and more standardized version of RML2016.04c.</td>
</tr>
<tr>
<td>RML 2016.10b</td>
<td>10 classes (BPSK, BPSK, CPSK, GFSK, PAM4, 16QAM, AM-DSB, 64QAM, QPSK, WBFM)</td>
<td>2*128</td>
<td>1200000</td>
<td>-20:2:18</td>
<td>A larger version of RML2016.10a, but not including AM-SSB.</td>
</tr>
<tr>
<td>RML 2018.01a</td>
<td>24 classes (OOK, 4ASK, 8ASK, BPSK, QPSK, 8PSK, 16PSK, 32PSK, 16APSK, 32APSK, 64APSK, 128APSK, 16QAM, 32QAM, 64QAM, 128QAM, 256QAM, AM-SSB-SC, AM-DSB-WC, AM-DSB-SC)</td>
<td>2*1024</td>
<td>2555904</td>
<td>-20:2:30</td>
<td>This dataset is very large and contains more kinds of modulation schemes.</td>
</tr>
<tr>
<td>HisarMod2019.1</td>
<td>26 classes (AM-DSB, AM-SC, AM-USB, AM-LSB, FM, PM, 2FSK, 4FSK, 8FSK, 16FSK, 4PAM, 8PAM, 16PAM, BPSK, QPSK, 8PSK, 16PSK, 32PSK, 64PSK, 4QAM, 8QAM, 16QAM, 256QAM)</td>
<td>2*1024</td>
<td>780000</td>
<td>-20:2:18</td>
<td>Contains signals from five different wireless communications channels.</td>
</tr>
</tbody>
</table>

3. AMR open source datasets for SISO systems

Learning-based AMR approaches heavily depend on datasets. Comprehensive high-quality datasets underpin the key aspects of DL-AMR, including model training, testing, and evaluation.

Table 2 summarizes the main AMR datasets and compares their properties. The RML datasets \(^1\) are generated using GNU radio by O’Shea et al. \([51,52]\), and RML2016.10a has been used in most research as a benchmark dataset. To simulate the data characteristics of modulated signals in real-world situations, the RML datasets take into account the common time-varying random channel effects in most wireless systems, which include center frequency offset, sample rate offset, additive white Gaussian noise, multi-path, and fading. RML2016.10a and RML2016.10b are generated by simulating the propagation properties in a harsh environment while RML2018.01a is produced in a relatively good real laboratory environment. To introduce a more comprehensive dataset, \([64]\) create a new dataset called HisarMod2019.1 \(^2\) using simulation software Matlab with 26 modulation classes of signals passing through the channels. This dataset provides wireless signals under ideal, static, Rayleigh, Rician \((k = 3)\), and Nakagami–m \((m = 2)\) channel conditions with various numbers of channel taps.

At present, most AMR research use RML2016.10a dataset, which has moderate data volume and contains some

\(^1\)http://radioml.com

\(^2\)http://dx.doi.org/10.21227/8k12-2g70
Table 3
Representative deep learning models for AMR in MIMO systems.

<table>
<thead>
<tr>
<th>Model</th>
<th>Author</th>
<th>Input</th>
<th>Main structure</th>
</tr>
</thead>
<tbody>
<tr>
<td>Co-AMC</td>
<td>Wang et al. [72]</td>
<td>I/Q data of multiple antennas</td>
<td>CNN and cooperative decision rules</td>
</tr>
<tr>
<td>CNN/ZF-AMC</td>
<td>Wang et al. [70]</td>
<td>The equalized I/Q sequence</td>
<td>CNN</td>
</tr>
<tr>
<td>STBC-AMC</td>
<td>Shah et al. [59]</td>
<td>Instantaneous and HOC-based features</td>
<td>SAE based DNN and RBFN</td>
</tr>
<tr>
<td>TL-AMC</td>
<td>Wang et al. [69]</td>
<td>The equalized I/Q sequence</td>
<td>CAE and CNN</td>
</tr>
</tbody>
</table>

common analog and digital modulation types like QAM, PSK, FSK, and AM. The later RML2016.10b dataset has a larger data volume and requires more computing resources. The modulation schemes in RML2018.01a dataset increase to 24 and the data length has also increased from 128 to 1024, which requires very high computing resources. However, the rich data of RML2018.01a enable further research to develop more advanced models. Although HisarMod2019.1 has 26 modulation types, the dataset is generated in a better transmission environment and thus the modulation schemes can be more easily identified.

Generally, datasets for deep learning methods are difficult to collect, and demand significant resource investment. Some of the datasets just simulate the channels and signals rather than collect data from real wireless communications environments. Despite the clear advantages such as generating a large number of data samples in a short time, the real-world channel conditions are usually more complex and dynamic. To design a DL-AMR model that can work under real channel conditions would require training or fine-tuning on datasets that are obtained from real-world scenarios.

4. DL models for AMR in MIMO systems

4.1. Introduction of the MIMO DL-AMR systems

The received signal after passing through the MIMO channel can be given as:

$$R(n) = HT(n) + G(n), \quad n = 1, \ldots, N,$$

where $H$ is the MIMO channel matrix, $R(n)$ is the received signal vector at $n$-th sample time, $T(n)$ is the transmitted signal vector and $G(n)$ is the additive white Gaussian noise (AWGN).

Different from SISO systems, MIMO systems can use the signals received by multiple antennas to jointly determine the modulation scheme. The authors of [72] proposed a cooperative DL-AMR method for MIMO systems, which obtains recognition sub-results of the signals received by each antenna through CNN, and then determines the final modulation schemes from the sub-results based on the voting and averaging strategy. To reduce the adverse effects caused by the channel, a CNN based zero-forcing (ZF) equalization AMR method is further proposed for MIMO systems [70]. The input data of the CNN-based model is the signal pre-processed by a ZF equalizer. A more complex system considering space-time block-code (STBC) is proposed in [59], using sparse autoencoders(SAE) based DNN and radial basis function network for modulation recognition. Considering the generally few labeled samples and large unlabeled samples in realistic communication scenarios, a transfer learning-based semi-supervised AMR method for a MIMO system is further proposed in [69]. Knowledge is transferred from the encoder layer of the reconstruction convolutional auto-encoder (CAE) to the feature layer of the AMR CNN by sharing their weights, where the reconstruction convolutional auto-encoder is trained by unlabeled samples and the AMR CNN is trained by labeled samples. Currently, only a few research has paid attention to applying deep neural networks for modulation recognition in MIMO systems, which is a very promising direction for developing novel methods based on the abundant and effective DL-AMR works in SISO systems.

4.2. The adopted MIMO system with precoding

Precoding for MIMO systems has gradually become a research hot spot in recent years with the development of millimeter wave communications technology, and research on the MIMO system incorporating precoding is gaining increased attention. In this paper, we test DL-AMR for the MIMO system containing precoding, and the detailed MIMO system is described as follows.

We consider a typical MIMO system with $N_t$ antennas at the transmitter and $N_r$ antennas at the receiver, as shown in Fig. 4. Transmission is over a flat fading channel $H$ ($H \in \mathbb{C}^{N_r \times N_t}$), and both the transmitter and receiver...
are assumed to have full knowledge of the channel. At the transmitter, the random data $x = [x_1, x_2, ..., x_k]$ is modulated into $s = [s_1, s_2, ..., s_k]$ using one of the candidate modulation schemes, and then normalized to $\tilde{s}$ with unit power, where $k$ is the number of transmitted symbols.

The singular value decomposition (SVD) approach is employed for precoding [34], and the SVD of $H$ can be written as

$$H = UV^H,$$

where $U$ and $V$ are unitary matrices, $V^*$ denotes the transpose conjugate of $V$, and $\Sigma$ is a diagonal matrix of the singular values of $H$ sorted in descending order.

The optimal vectors to be used at the transmitter side and receiver side are the first column of $U$ and $V$ corresponding to the largest singular value of $H$, which are denoted by $U_1$ and $V_1$, respectively. The modulated data $\tilde{s}$ is multiplied by the matrix $V_1$ before being transmitted over the channel. In turn, the receiver multiplies the received data by the matrix $U_1^*$. Then, the vector of the received symbol can be expressed as

$$\tilde{y} = U_1^H(HV_1\tilde{s} + n) = \lambda_1\tilde{s} + \tilde{n},$$

where $n$ is a vector of additive white Gaussian noise (AWGN) at the receiver, $\tilde{n} = U^Hn$ has the same distribution of the noise $n$, $\lambda_1$ is the largest singular value of $H$, and $\tilde{y} = [\tilde{y}_1, \tilde{y}_2, ..., \tilde{y}_k]$ indicates the data to be recognized. Due to the extensive works on blind channel estimation [66, 84], we can assume that the channel state information is perfect at the receiver, thus allowing modulation recognition of the signal after receiving decoding.

5. Experimental comparison for SISO system

This section evaluates the state-of-the-art supervised deep learning models for AMR challenges using extensive experiments focusing on model structures, complexity and recognition accuracy, to provide a complete picture of this field. The experiments will further highlight the key unsolved problems of DL-AMR and the promising future research directions.

5.1. Experimental Setup

We choose the benchmark datasets RML2016.10a, RML2016.10b, RML2018.01a, and HisarMod2019.1 for all experiments. The input data dimension of RML2016.10a and RML2016.10b is $2 \times 128$, while that of RML2018.01a and HisarMod2019.1 is $2 \times 1024$. We only adapt the input dimension and output dimension of the models to fit the input data, and keep the same intermediate parameters across all datasets.

We divide each RML dataset into the training set, validation set and test set at the ratio of 6:2:2, while HisarMod2019.1 dataset is divided into the ratio of 8:2:5 due to its data storage format. The categorical cross-entropy is set as the loss function and the Adam algorithm is used as the optimizer. The initial learning rate starts at 0.001 and the batch size is set as 400 in all experiments. If the validation loss does not decrease in 5 epochs, the learning rate will be halved. The training process stops if the validation loss remains stable in 50 epochs. The experiments are implemented using GeForce GTX 1080Ti GPU and Keras with Tensorflow as the backend.
5.2. Recognition Accuracy

The experimental accuracy of the 14 models is shown in Fig. 5. Under the same experimental parameter settings, the highest recognition accuracy achieved on RML2016.10a, RML2016.10b, RML2018.01a and HisarMod2019.1 datasets are 92.05% (by MCLDNN at 10dB), 94% (by LSTM at 18dB), 98.39% (by LSTM at 22dB), and 100% (by CNN1 and DenseNet at high SNR), respectively.

From the overall experimental results on RML datasets using either I/Q or amplitude/phase as input, several AMR models based on RNN, e.g., LSTM, GRU, PET-CGDNN, and MCLDNN, have clear advantages in recognition accuracy. The experimental results show that the temporal features of the modulated signals provide pivotal support to the recognition accuracy on RML datasets, and the spatial features of signals extracted by CNN models are more effective for HisarMod2019.1. Since HisarMod2019.1 is generated by random bit sequences and RML datasets are generated by specific source data (Serial Episode 1 and Gutenberg works of Shakespeare in ASCII [50]), RML data samples contain more relevant information on the temporal scale, so the RNN layer, which is more capable of extracting temporal features, can not perform as well on HisarMod2019.1 dataset as it does on RML datasets. On the other hand, the channel conditions for generating RML datasets are more complex (e.g., sample rate offset and center frequency offset), so that the RML data symbols sampled at different time have more relevant features on the temporal scale, and the effective extraction of these features by RNN layers can improve the recognition accuracy.

Note that CNN2 has more convolutional layers than CNN1, and the accuracy on RML datasets suggests that a model with more convolutional layers can achieve the same or better performance on certain tasks with fewer parameters.
However, the simplest CNN1 structure achieves better performance on HisarMod2019.1 dataset, implying that the model with more layers or more complex structures (e.g., CLDNN2 and DAE) may fall into local optimality on this relatively more ideal dataset and is prone to degradation problems [18]. The models based on ResNet and DenseNet, which contain an exceptionally large number of internal parameters and achieve superior performance in computer vision tasks, fail to achieve equivalent performance in RML datasets but perform well on HisarMod2019.1 dataset. Multi-channel inputs, as adopted by MCLDNN, result in higher accuracy on RML datasets than the single-channel structure used by several models as shown in the accuracy figures.

The validation loss in the training process shows that all the models converge quickly and reach a steady status beyond 40 epochs, among which MCLDNN, PET-CGDNN, LSTM, and GRU achieve the best convergence trends across three RML datasets. Some other models, e.g., DAE, CLDNN, and CLDNN2, have varied performances and significantly different parameters in different datasets, showing their potential improvements in generalisation. Particularly, RML2018.01a dataset is a larger dataset compared with RML2016.10a and RML2016.10b, and a model’s hyperparameters could be appropriately increased to benefit from the rich data samples. In order to provide a fair comparison, this paper adopts the same hyperparameters. Interested researchers could explore the effect of hyperparameters based on the trends of experimental results here.

### Table 4

<table>
<thead>
<tr>
<th>Model</th>
<th>Learning parameters</th>
<th>Training time (seconds/epoch)</th>
<th>Training epochs</th>
<th>Minimum validation loss</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dataset</td>
<td>A</td>
<td>B</td>
<td>C</td>
<td>D</td>
</tr>
<tr>
<td>CNN1</td>
<td>1592383</td>
<td>1592126</td>
<td>13064524</td>
<td>13065038</td>
</tr>
<tr>
<td>CNN2</td>
<td>858123</td>
<td>857994</td>
<td>1777304</td>
<td>1777562</td>
</tr>
<tr>
<td>MCNET</td>
<td>121511</td>
<td>121226</td>
<td>126616</td>
<td>127386</td>
</tr>
<tr>
<td>IC-AMCNET</td>
<td>126401</td>
<td>1263882</td>
<td>8605720</td>
<td>8605978</td>
</tr>
<tr>
<td>ResNet</td>
<td>3098283</td>
<td>3098154</td>
<td>21450404</td>
<td>21450298</td>
</tr>
<tr>
<td>DenseNet</td>
<td>3282603</td>
<td>3282474</td>
<td>21634360</td>
<td>21634618</td>
</tr>
<tr>
<td>GRU</td>
<td>151179</td>
<td>151050</td>
<td>152856</td>
<td>153114</td>
</tr>
<tr>
<td>LSTM</td>
<td>201099</td>
<td>200970</td>
<td>202776</td>
<td>203034</td>
</tr>
<tr>
<td>DAE</td>
<td>1063599</td>
<td>1063642</td>
<td>67125990</td>
<td>67125994</td>
</tr>
<tr>
<td>MCLDNN</td>
<td>406199</td>
<td>406070</td>
<td>407876</td>
<td>408134</td>
</tr>
<tr>
<td>CLDNN</td>
<td>164433</td>
<td>164176</td>
<td>884374</td>
<td>884888</td>
</tr>
<tr>
<td>CLDNN2</td>
<td>517643</td>
<td>517514</td>
<td>698320</td>
<td>698578</td>
</tr>
<tr>
<td>CGDNet</td>
<td>124933</td>
<td>124676</td>
<td>665874</td>
<td>666386</td>
</tr>
<tr>
<td>PET-CGDNN</td>
<td>71871</td>
<td>71742</td>
<td>75340</td>
<td>75598</td>
</tr>
</tbody>
</table>

5.3. Model complexity and training time

Five key indicators are selected to measure the complexity and evaluate the training process of the 14 models (Table 4), including the learning parameters, training time, training epochs, and minimum validation loss. A model’s complexity is mainly reflected by the learning parameters, the training speed is indicated by the training time and training epoch, while the minimum validation loss demonstrates the convergence trend.

Due to the local connection and weight-sharing characteristics of CNN, it is more computationally efficient. Although some CNN models have more learnable parameters, such as CNN1 and IC-AMCNET, their training time in one epoch is shorter. It is worth noting that MCLDNN has the minimum validation loss on RML2016.10a and RML2016.10b datasets, which contributes to the high recognition accuracy. PET-CGDNN has the least learning parameters and thus requires less memory than others. CNN-based models generally have lower validation loss on HisarMod2019.1 dataset, indicating that CNN models are better suited to this simpler dataset. The high validation loss of MCNET on HisarMod2019.1 dataset suggests that using fewer parameters and a more complex structure on this dataset with a larger input dimension tends to cause poorer model convergence.

5.4. Confusion matrices analysis

The confusion matrices of the 14 models on the three datasets provide important insights on the main modulation types that cause significant classification errors. Fig. 6 selects the typical confusion matrices when SNR is moderate, e.g., at 4dB, across 14 models. The vertical axis on each matrix denotes the true labels while the horizontal axis denotes the predicted labels. We summarize the key observations below: On RML2016.10a (Fig. 6: (A1) - (A14)):
Figure 6: Confusion matrices. A, B and C represent the confusion matrices obtained on the RML2016.10a, RML2016.10b, and RML2018.01a, respectively. The numerical indexes 1 - 14 denote CNN1, CNN2, MCNET, IC-AMCNET, ResNet, DenseNet, GRU, LSTM, DAE, MCLDNN, CLDNN, CLDNN2, CGDNet, PET-CGDNN.
• All models confuse a limited number of samples modulated with 16QAM and 64QAM, caused by the overlapping constellation points between these two modulation schemes. MCLDNN performs best owing partially to the multi-channel structure that differentiates the constellation points that are not completely overlapped.

• A significant number of WBFM modulation data could be classified as AM-DSB by all models. The data of WBFM and AM-DSB both come from a speech signal with some interludes and off-time [51], leading to classification errors.

On RML2016.10b (Fig. 6: (B1) - (B14)):

• Similar results were obtained as RML2016.10a because the two datasets have similar data inputs as described before. For example, the confusion between WBFM and AM-DSB still exists.

• The 16QAM and 64QAM confusion problem on RML2016.10a is slightly alleviated in most models due to the increased number of samples provided by RML2016.10b dataset.

On RML2018.01a (Fig. 6: (C1) - (C14)):

• Because more modulation schemes are introduced, more confusion types are observed in almost all the 14 models, e.g., confusion between 4ASK and 8ASK, 16PSK and 32PSK, 64APSK and 128APSK, QAM schemes (32QAM, 64QAM, 128QAM, and 256QAM), AM-DSB-WC and AM-DSB-SC.

• Data with similar modulation types are more likely to cause misjudgment if they are under strong noise and interference. The increase of SNR could contribute to decreased confusion and improved recognition accuracy.

On HisarMod2019.1 (Fig. 6: (D1) - (D14)):

• The confusion problem is not as severe as the results on the RML datasets, e.g., CNN1 has no recognition error, and the models which have confusion problems mainly exist in the same class of modulation schemes, e.g., PSK, QAM, and PAM.

• DAE model has serious confusion problems on this dataset, with the output modulation schemes distributed in AM-DSB, AM-DSB-SC, FM, and PM, but still maintains some ability to identify FM and PM modulation schemes.

6. Experimental comparison for the adopted MIMO system

Our main objective is to verify the feasibility of applying DL-AMR in the MIMO system with precoding, rather than to propose a new deep learning model for modulation recognition. Therefore, four representative models are compared in the experiments in this section, including a CNN model [72], an RNN-based LSTM model [56], an RNN-based GRU model [21], and a hybrid CLDNN model [44]. This study aims to provide some baseline results for future research to make further improvements and innovations based on our experimental results and open source code.

The dataset is generated by MATLAB and then Keras with Tensorflow as the backend is used to train the modulation recognition model. The data are modulated using different modulation methods, including ‘2PSK’, ‘QPSK’, ‘8PSK’, ‘16QAM’, ‘64QAM’, and ‘128QAM’. The number of transmitted symbols per sample is 128 (\(k=128\)), and we prepare 500 samples per SNR per scheme, which are divided into three parts for training, validation, and testing by the ratio of 6:2:2. The batchsize is set to 128, and the other parameters are set as the same as the SISO model experiments in Section 5.1.

We conducted experiments with three different antenna configurations: 4*2 (\(N_t=4,N_r=2\)), 16*4 (\(N_t=16,N_r=4\)), and 64*16 (\(N_t=64,N_r=16\)). The performance of different deep learning models with different antenna configurations is shown in Fig. 7.

From Fig. 7, the CLDNN model has a higher overall recognition accuracy and the CNN model has a lower overall recognition accuracy. The gain made by the increased number of antennas leads to the improved recognition accuracy of each model. The experimental results exhibit that DL-AMR is feasible for the MIMO system with precoding, and SISO’s DL-AMR method can be applied to such MIMO systems, since their input data contain only one complex data stream which requires similar network structures. The impact of other factors, such as channel estimation error, precoding schemes, etc, is worthy to be further investigated.
The AMR of MIMO systems differs somewhat from that of SISO systems, and the main difference is the number of data streams to be processed. In MIMO systems, one way is to perform modulation recognition directly on the data received from multiple antennas, and the data samples have an extra antenna dimension of $N_r$. The data samples received from multiple antennas can be recognized by cooperative recognition methods [72] or 3D convolution-based approach [29] for modulation recognition. Alternatively, the received data from multiple antennas can be processed at the receiver, e.g., by ZF equalization [70] or precoding, to convert the received data from multiple antennas to a single data stream, ensuring the similar DL-AMR method applied for SISO systems can be applied for MIMO systems.

7. Challenges and future directions

As described in the previous sections, DL shows great potential for AMR. However, there are still several unsolved problems that are worthy of further study. This section discusses the important problems and opportunities in this area from three perspectives, focusing on model innovation, data and datasets, and model optimization and visualisation.

7.1. Designing novel DL-AMR models

**Novel deep learning models specialised for modulation inference:** According to the existing research, CNN-based spatial feature extraction and RNN-based temporal feature extraction have led to the breakthroughs on AMR. Other types of neural network structures such as generative adversarial networks (GAN) [63], attention mechanism [6, 23, 39, 41, 40], and transformer, which have been shown to achieve good performance in specific fields, could be further exploited for AMR. For example, GAN can be used as a generator to expand the training set which can be regarded as a data augmentation method. The attention mechanism can allow the model to focus on relevant features and accelerate the training process. Considering the sequential characteristics of AMR signals, the transformer models, introduced in natural language processing, could further enhance the AMR performance. Joint signal processing and model design would be another promising future direction [82, 86].

**Model compression and simplification:** For AMR challenges, and even the wider deep learning community, achieving higher accuracy is often based on stacking more layers and deepening the model. While such strategies may lead to feasible models in many offline tasks, AMR often requires online processing and will encounter excessive delays if a model is too complex. The high computational complexity also prevents their application in resource-constrained devices such as many internet-of-thing (IoT) devices that have limited memory, computing power, and energy, while efficient DL-AMR models with low complexity and lightweight property can meet the requirement of next generation mobile communication systems, such as 5G systems, which have greater demand for extremely reliable and low-latency [57]. In the machine learning community, model compression and simplification methods such as pruning [42, 73, 65], knowledge distillation [45], and weight quantization have been shown to successfully decrease model complexity significantly while maintaining the same level of performance, hence could be exploited in DL-AMR.

**Model generalisation on different channel conditions:** Most AMR deep learning models are trained with the data generated under specific channel conditions. To apply such models in practical systems would require models to have exceptional generality and robustness. To improve the generality of a model for different channel conditions at the
designing stage is a topic worthy of study in the future. One potential method is transfer learning [69]. For example, a DL-AMR model can be trained on offline datasets such as RML2016.10a, and fine-tuned using online data obtained in real-world by adopting the transfer learning strategy. If no training data is available or only a few training samples are available for some signal classes, then zero-shot [11] and few-shot [94, 37, 91] learning are possible solutions for signal recognition. To improve the robustness of a model, methods that can withstand adversarial attacks need to be further investigated [58].

7.2. Benchmark AMR models, datasets, and data challenges

Open-source datasets and benchmark models: There are only a few open-source datasets available for AMR so far, i.e., RadioML2016.10a, RadioML2016.10b, RadioML2018.01a and HisarMod2019.1. Many research generate their own datasets and train their own models using such datasets, lacking solid common benchmarks for performance comparison and further improvement. More high-quality AMR datasets (similar to ImageNet in computer vision) and a unified benchmark paradigm will be a challenging but rewarding direction, while data cleaning and augmentation may also be one major challenge [24]. Furthermore, with the wide deployment of MIMO systems, it is also necessary to get the corresponding standard datasets which can be used in AMR research for MIMO systems [14].

Multi-channel inputs and parallel processing: Pioneering work on multi-channel inputs and feature fusion, e.g., MCLDNN [79], CNN-LSTM [92], and WSMF [55] has achieved higher accuracy in AMR than the other sophisticated models with single inputs. The rich and often complementary information provided by multiple channels could enable models to learn key distinguishable features, but it also introduces increased computational complexity and potentially poor convergence. The parallel nature of such a method could benefit from high-performance computing architectures based on parallel processing such as graphics processing units (GPUs), which can potentially reduce the computational complexity by many folds. To address the convergence challenge, new loss functions or weight adjustments based on different data types can be exploited.

Semi-supervised methods: Unsupervised learning methods such as DBN [48] and autoencoder [1] have been exploited in AMR, but have not achieved comparatively distinctive results. Most successful state-of-the-art models are supervised models, which require a large amount of labeled data. Building a high-quality labeled dataset often needs significant investment in resources and time. Semi-supervised models [10, 69], which are trained on partially labeled samples or a small number of samples, have achieved breakthroughs in a wide range of applications such as classification and detection, and could therefore be applied in AMR to mitigate the needs for labeled data from different channel conditions.

7.3. Model optimization and visualisation

Fusion of deep learning models with expert knowledge: Deep learning models learn representative features purely from data, while expert knowledge of wireless communications systems has proved to be highly reliable. The fusion of deep learning models with expert knowledge (e.g., high order cumulants (HOC), handcrafted features, spectrum, constellation diagrams, etc.), could lead to a potential breakthrough in AMR [93]. Furthermore, signal processing techniques such as filtering [16, 85] and denoising [88] can also help retain distinctive data features and improve system performance.

AMR model visualisation: The low SNR of the input signal in AMR makes the features extracted by a DL model difficult to assess visually, which affects the design and development of new models. On the one hand, good visualization techniques help design models and optimize parameters thus achieving better performance [25]. On the other hand, visualization can provide the explainability of DL-AMR models, from which we can know what kinds of signal features should be extracted and chosen for classification.

8. Conclusion

This paper reviews the emerging interdisciplinary topic of DL-AMR, which also analyzes the characteristics of benchmark models and discusses the available datasets. We have further identified the unresolved problems and proposed possible solutions through extensive experimental comparisons, which have provided detailed insights of this topic for future research. With the ubiquitous deployment of cognitive communications systems and distributed networked devices in future B5G/6G networks, DL-AMR will have an increasingly important role to play.
Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was supported by the National Natural Science Foundation of China under grant 61871096, and in part by the National Key RD Program of China under grant 2018YFB2101300.

References


[92] Zhang, Z., Li, Y., Zhai, Q., Li, Y., Gao, M., 2022. Few-shot learning for fine-grained signal modulation recognition based on foreground

