IRLNet: A Short-Time and Robust Architecture for Automatic Modulation Recognition

HUOGEN YANG1, LINGZHU ZHAO1,2, GUANGXUE YUE1,2, BOLIN MA2, AND WEI LI2

1College of Science, Jiangxi University of Science and Technology, Ganzhou, Jiangxi 341000, China
2College of Information Science and Engineering, Jiaxing University, Jiaxing, Zhejiang 314001, China

Corresponding author: Guangxue Yue (gxyue@zjxu.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant U19B2015 and Grant 12161043, in part by the Natural Science Foundation of Jiangxi province of China under Grant 20192BAB201007, and in part by the Innovation Fund Designated for Graduate Students of Jiangxi Province of China under Grant YC2020-S449.

ABSTRACT Automatic modulation recognition with deep learning (DL) is challenging in distinguishing high-order modulation modes and balancing complexity against recognition accuracy. In this paper, we propose a novel dual-path modulation recognition framework named IRLNet, which consists of the improved residual stacks (IRS) and long short-term memory (LSTM). The IRS maintains the more initial residual information, learns the signal features in deep and shallow, and achieves various degrees of feature extraction. The model learns from the time domain I/Q, amplitude and phase information presented in the training data. The simulation results on RadioML2016.10B show that IRLNet performs stable in the training stage and has low spatial-temporal complexity. It also achieves a recognition accuracy of more than 93% at high signal-to-noise ratios (SNRs). Transfer learning is introduced to improve the efficiency of retraining, and robustness is proved by transferring the model to RadioML 2018.01A and HisarMod 2019.1. The simulation result shows that the training time is greatly shortened by about 25.6% in RadioML 2018.01A by introducing transfer learning. Moreover, IRLNet improves the confusion of high-order modulations and achieves a recognition accuracy of more than 90% in high SNRs.

INDEX TERMS Automatic modulation recognition, deep learning, feature extraction, improved residual stacks, LSTM.

I. INTRODUCTION

With the remarkable development of wireless communication, the signal modulation modes are becoming more complicated and diversified, and the radio environment is becoming increasingly harsher [1]. As a critical step between signal detection and demodulation, automatic modulation recognition (AMR) can also detect physical threats such as pilot jamming, deceptive jamming, and Sybil attacks [2]. In the civilian field, it is primarily used in spectrum management for signal authentication and interference identification. In the military field, it is one of the important means in the struggle and control electromagnetic power. Besides, it is the previous step to intercept enemy signals, destroy and suppress enemy communication, and implement electromagnetic interference.

Traditional AMR methods are generally divided into two categories: likelihood-based (LB) methods and feature-based (FB) methods [3]. Considering AMR as a multiple composite hypothesis testing problem, LB methods rely on the correct modeling of unknown quantities. The number of hypotheses equals that of classified modulations [4]. The classification result is given according to the compared results of the likelihood ratio of each possible hypothesis and the threshold. LB methods are optimal in the Bayesian sense as they maximize the probability of correct classification with the full knowledge of channel conditions [5]. However, LB methods depend heavily on prior knowledge and parameter estimation, not to mention their high computational complexity. Moreover, LB methods are not robust with respect to model mismatch such as phase and frequency offsets, residual channel effects, timing errors, and non-Gaussian noise distributions [6]. Instead of using the probability distribution function (PDF), FB classifiers calculate certain statistical features of signal samples [7]. The FB methods generally consist of three steps: data preprocessing, feature extraction, and classification. In the process of feature extraction, various statistical features such as high-order cumulants [8], wavelet
transform [9], and cyclostationary features [10] are widely used. Generally, FB methods are sub-optimal in the Bayesian sense, but they are popular due to their easy implementation [11]. However, they need to manually extract expert features from plenty of samples, which leads to a high computational complexity [12]. Furthermore, FB methods are generally designed for a certain modulation mode in a specific environment, making them difficult to extend.

Due to the poor versatility and high complexity of traditional techniques, there is an emerging demand for the quick discrimination of the modulation modes. Recently, machine learning techniques such as deep learning (DL) and deep reinforcement learning (DRL) have achieved great success in the fields of computer vision [13], natural language processing [14], and network resource allocation [15], [16]. To a great degree, they have facilitated the research of signal modulation classification. O’Shea et al. [17] firstly introduced the convolutional neural networks (CNN) for classifying modulation types and proposed a 4-layer CNN2 model. Ramjee et al. [18] repeatedly adjusted the depth and filter settings of the CNN2 in [17] and proposed a 6-layer CNN4 model. The results shown that the neural network not only outperforms the traditional methods in terms of accuracy, but also in terms of flexibility in identifying various modulation types. However, as the network continues to deepen, CNN has the problem of gradient vanishing or exploding, which causes the decline of the recognition accuracy and leads to confusion between higher-order modulations. Subsequently, some other neural networks such as convolutional long short-term deep neural networks (CLDNN), inception architecture, and residual network (ResNet) were applied to modulation recognition. By combining the advantages of CNN and long short-term memory (LSTM) modules, West and O’Shea [19] proposed CLDNN. Besides, an inception architecture was also constructed by connecting three different CNN networks in parallel. And in [20], O’Shea et al. proposed the ResNet by adding the bypass connections which created the identity mapping. Although the recognition accuracy has been improved to a certain extent, the confusion between high-order modulation modes remains unresolved for some models.

With the development of fifth generation (5G) wireless communication technology, the number of communication devices has increased sharply, and spectrum resources have become increasingly scarce. As a practical solution to improve the effectiveness of the spectrum, high-order modulations have been widely used in wireless communication systems. It is known from open protocols and standards [21] that high-order modulations such as 64QAM and 256QAM have been applied to 5G mobile communications. With the prospect of 5G mobile communication for dynamic spectrum access, high-order modulation types possess more practical significance. To solve the issue of confusion between high-order modulations, some experts and scholars have conducted various studies. Data preprocessing is the common method to improve recognition accuracy. Rajendran et al. [22] preprocessed the I/Q signal into amplitude and phase information before using LSTM to feature extraction, which improved the recognition accuracy to a certain extent but increased the execution delay of the classifier. Wang et al. [23] merged two CNNs with different structures to improve the overall average classification accuracy. The second CNN used constellation diagrams as input, which significantly improved the recognition accuracy of high-order modulations (such as 16QAM and 64QAM) while causing long execution times. Teng et al. [24] proposed an AMR method based on accumulated polar features and supplemented by the channel compensation mechanism. The model used the constellation diagrams with the polar feature transformation and temporal accumulation as input. It greatly enhanced network recognition performance and robustness in time-varying fading channels, but at the same time increased the time cost.

The existing modulation recognition methods may have poor performance, especially in identifying high-order modulations and their high complexity. To address these issues, we proposed a new modulation recognition model hybridized CNN and recurrent neural network (RNN) to extract the spatial and temporal characteristics of signals. Besides, to enrich the feature dimension, we carried out feature conversion on the input of LSTM by transforming the baseband I/Q signal into amplitude and phase information. The main contributions of this paper are summarized as follows:

1) An improved residual stack (IRS) is proposed. The IRS maintains the more initial residual information, learns the signal features in deep and shallow, and achieves various degrees of feature extraction.
2) Design an AMR architecture named IRLNet based on the IRS and LSTM, which achieves high recognition accuracy with low complexity. Even the high-order modulations that are easy to be confused can be also properly recognized.
3) Demonstrate robustness of IRLNet by testing its recognition performance for diverse modulation modes and different channel environments.

The remainder of the paper is organized as follows, section II clearly states the problem of modulation recognition. Section III describes the IRLNet used for AMR. Section IV introduces the experimental setting. Section V introduces the classification results in details and discusses the advantages of the proposed model. Finally, we conclude the work in Section VI.

II. PROBLEM STATEMENT

Modulation is an important part of wireless communication which can expand the signal bandwidth and improve the anti-fading and anti-interference capabilities. The modulation process transforms the baseband signals into high-frequency signals, which allow for accurate and low-noise data transmission between distant transmitters and receivers. The modulation recognition module is generally deployed at the receiving side of the communication system to provide
modulation information for the demodulator, as shown in Fig. 1. In a wireless communication system, the received signal is generally expressed as [22]

\[ r(t) = s(t) \ast c(t) + n(t) \] (1)

where \( s(t) \) is the signal transmitted by the transmitter after modulated, \( c(t) \) is the impulse response of wireless channel, \( n(t) \) is additive white gaussian noise (AWGN) with zero mean and variance \( \sigma_n^2 \), and \( r(t) \) is the received signal, which is commonly expressed in I/Q format and sampled \( n \) times by the analog to digital converter at a rate \( f_s = \frac{1}{T} \). The real and imaginary parts of \( r(t) \) represent I and Q components, respectively. Specifically, \( r(t) \) can be modeled as [25]

\[ r(t) = \alpha(t)e^{j(2\pi f_0 + \theta_0(t))}s(t) + n(t) \] (2)

where \( \alpha(t) \) denotes the Rayleigh fading channel, \( f_0 \) and \( \theta_0(t) \) represent the frequency and phase offsets introduced by the disparate local oscillator and Doppler effect, respectively.

The transmitted signal has different mathematical expressions when modulated by different modes. The modulation modes presented in [1] were considered to investigate the differences. If the signal is modulated by amplitude-shift keying (ASK), frequency-shift keying (FSK) or phase-shift keying (PSK), \( s(t) \) is expressed as follows:

\[ s(t) = \left[A_m \sum_n a_n g(t - nT_s) \right] \cos(2\pi (f_c + f_m)t + \phi_0 + \phi_m) \] (3)

where \( A_m \) is the modulated amplitude, \( a_n \) is the symbol sequence, \( g(\cdot) \) is the signal pulse, and \( T_s \) is symbol period. \( f_m \) is the modulated frequency, and \( f_c \) is the carrier frequency. \( \phi_0 \) represents the initial phase, and \( \phi_m \) represents the modulation phase. However, the quadrature amplitude modulation (QAM) is different from the aforementioned modulation modes. Specifically, it has two orthogonal carriers modulated by \( a_n \) and \( b_n \), and \( s(t) \) is expressed as follows:

\[ s(t) = \left[A_m \sum_n a_n g(t - nT_s) \right] \cos(2\pi f_c t + f_0) \]

\[ + \left[A_m \sum_n b_n g(t - nT_s) \right] \sin(2\pi f_c t + f_0) \] (4)

Modulation recognition is generally considered as an N-class classification problem, where \( N \) denotes the number of modulation modes for the transmitted signals. The aim of the modulation recognition task is to classify these modulation categories blindly from the n-sample received symbol vector \( Y = [y^0, y^1, \ldots, y^{n-1}]^T \). As the most common evaluation metric for classification problems, accuracy is used to measure the model performance. Denote \( S = [s^0, s^1, \ldots, s^{N-1}] \) as the pool of known \( N \) candidate modulations. The actual adopted modulation and the predicted modulation for the i-th sample are labeled as \( s^i \) and \( \tilde{s}^i \), respectively, where \( i = [1, 2, \ldots, K] \) and \( K \) is the sample number of the testing set. The accuracy indicator for the i-th recognition is defined as

\[ P_i = \begin{cases} 1, & \tilde{s}^i = s^i \\ 0, & \tilde{s}^i \neq s^i \end{cases} \] (5)

The overall recognition accuracy rate for \( K \) testing samples is then measured as

\[ P_{acc} = \frac{1}{K} \sum_{i=1}^{K} P_i \] (6)

III. MODEL DESCRIPTION

A. SIGNAL PREPROCESSING STAGE

Since the features extracted from the time domain I/Q signals are limited, we preprocess the input data of LSTM by converting the received signals from Cartesian coordinates to polar coordinates to better and more comprehensively extract the features. Learning features from the \( r-\theta \) domain encodes specific communication system information and makes the following network more resilient to fading channels. After that, the I/Q signals are converted to the corresponding amplitude and phase values. Fig. 2 describes the constellation diagrams of five common modulation modes with SNR = 10 dB under AWGN channel before and after the coordinate transformation. For rectangular coordinates and polar coordinates, the horizontal and vertical axes represent I/Q and radius/theta, respectively.

As the images of 16QAM and 64QAM depicted in Fig. 2, the pattern in I-Q plane is more regular, while in \( r-\theta \) plane is more diverse. The 16QAM constellation can be seen as a sub-picture of 64QAM in the I-Q plane, which is referred to as nested modulations. The shared constellation points in I/Q based images cause a lot of misclassification. We transform the I-Q domain data into the \( r-\theta \) domain before sending it to the LSTM in our simulations by leveraging existing expert knowledge in communication. To construct the relation between I and Q components, we associate with polar coordinates which substitute the I-Q axis with the \( r-\theta \) axis. Algorithm 1 summarizes the transformation procedure, where I, Q denote the real and imaginary parts of received complex symbols, \( r, \theta \) represent the transformed polar coordinates of radius and theta, and \( n \) is the symbol length.
H. Yang et al.: IRLNet: Short-Time and Robust Architecture for Automatic Modulation Recognition

**Algorithm 1** Polar Feature Transformation

**Input:** I, Q

```plaintext
for i = 0 : n - 1 do
    r[i] ← √(I[i]² + Q[i]²)
    θ[i] ← arctan(Q[i]/I[i])
end for
```

**Output:** r, θ

After conversion, the constellation diagrams of 16QAM and 64QAM showing a significant difference.

**B. CLASSIFICATION STAGE**

1) IRS

As a variant of CNN, ResNet improves the accuracy of feature extraction by adding skip connections between the input and the output of different convolutional layers, as well as performing feature operations at multiple scales and depths. The typical basic unit of ResNet called residual stack (RS) is depicted in Fig. 3(a). Each RS is made up of one convolutional layer, two residual units, and a max-pooling layer.

To maintain more initial residual information, learn the signal features in deep and shallow, achieve various degrees of feature extraction, and escape the network from complexity, we design the IRS, as shown in Fig. 3(b). The IRS adds an extra skip connection between the input and output of the two residual units. Same with RS, the residual unit in IRS is made up of two convolutional layers and a ξ-times skip connection. A λ-times skip relation is also associated between the input of the former residual unit and the output of the latter residual unit.

The inputs of RS and IRS are both x, and the outputs are y and y’, respectively. Thus [27], [28],

\[
\begin{align*}
u &= f(xW_0 + b_0) \\v &= g((uW_1 + b_1)W_2 + b_2) + \xi u
\end{align*}
\]

Figures 2 and 3 show the data transformation and constellation diagrams for five modulation modes with SNR = 10 dB under AWGN channel.

Figures 4 and 5 illustrate the structure of LSTM cell [26].
where $W_i$ and $b_i$ are the weight and bias of the convolutional layer, $f(\cdot)$, $g(\cdot)$ and $h(\cdot)$ denote the function to be learned, $\xi$ and $\lambda$ are multiplier coefficients of skip connection, and $\text{maxp}(\cdot)$ represents max-pooling operation.

2) LSTM
RNN is a branch of deep learning that has significant advantages in processing time-series data. Different from the feedforward neural networks (e.g., CNN), RNN pays more attention to feedback and uses the internal state to process the sequence data. However, due to the disappearance or explosion of the gradient, the processing sequence data of RNN is limited by the length. As the variant of RNN, LSTM can significantly eliminate the problem of gradient vanishing or explosion for its unique cell structure shows in Fig. 4.

Each LSTM cell contains three types of gates: the input gate ($i$), the forget gate ($f$), and the output gate ($o$) [29]. In the process of forwarding computing, the forget gate ($f_t$) decides whether or not to keep a cell state memory ($c_t$) at time $t$. On the input $x_t$ at time $t$ and output ($h_{t-1}$) at time ($t-1$), the forget gates are constructed according to equation (13).

$$f_t = \sigma(W_f [h_{t-1}, x_t] + b_f)$$

where $\sigma$ represents the sigmoid activation function, $W_f$ and $b_f$ denote the associated weight and bias between the input ($x$) and the forget gate ($f$). The input gates ($i_t$) specify the cell states ($\tilde{C}_t$) to update after $f_t$ decides which memories to forget, as shown in equations (14) and (15).

$$i_t = \sigma(W_i [h_{t-1}, x_t] + b_i)$$
$$\tilde{C}_t = \tanh(W_C [h_{t-1}, x_t] + b_C)$$

where $\tanh$ stands for the tanh activation function, $W_i$ and $b_i$ denote the associated weight and bias between the input ($x$) and the input gate ($i$), and $W_C$ and $b_C$ denote the associated weight and bias between the input ($x$) and the cell memory ($C$). Using forget gates ($f_t$) and input gates ($i_t$), the old cell state ($C_{t-1}$) is changed to the new cell state ($C_t$) as per equation (16).

$$C_t = f_t \cdot C_{t-1} + i_t \cdot \tilde{C}_t$$

where $\cdot$ is the Hadamard product. Finally, based on the cell states ($C_t$), the output ($h_t$) through output gates ($o_t$) according to equations (17) and (18).

$$o_t = \sigma(W_o [h_{t-1}, x_t] + b_o)$$
$$h_t = o_t \cdot \tanh(C_t)$$

where $W_o$ and $b_o$ denote the associated weight and bias between the input ($x$) and the output gate ($o$).

3) IRLNet
Considering the strong representative abilities and the role in alleviating the vanishing gradients problem of IRS, as well as the advantages in processing temporal data of LSTM, we propose the IRLNet to make the network more lightweight while ensuring high recognition accuracy. The architecture of the IRLNet is shown in Fig. 5. To simplify the figure, we omit the batch normalization (BN) and rectified linear unit (ReLU) in the IRS. BN follows each convolutional layer in the IRS. Except for the first convolutional layer which uses the linear activation, the rest convolutional layer all use the ReLU activation. The kernel size of every convolutional layer is set to 5, and the kernel is set to 32 as required. The BN operation is defined as [30]:

$$BN(x) = \sigma_1 \frac{x - \mu_B}{\sqrt{\sigma_B^2 + \varepsilon}} + \mu_1$$

where $\mu_B$ is the mean and $\sigma_B^2$ is the variance of the mini-batch samples, $\mu_1$ and $\sigma_1$ are the offset and scale factors that are
TABLE 1. The detailed structure of IRLNet.

| Layer (type)               | Output Shape | Connected to |
|---------------------------|--------------|--------------|
| input_1 ([InputLayer])    | (None, 2, 128)|              |
| conv00 (Conv1D)           | (None, 32, 128)| input_1[0][0]|
| bn00 (BatchNormalization) | (None, 32, 128)| conv00[0][0]|
| conv100 (Conv1D)          | (None, 32, 128)| bn00[0][0]|
| bn100 (BatchNormalization)| (None, 32, 128)| conv100[0][0]|
| conv200 (Conv1D)          | (None, 32, 128)| bn100[0][0]|
| bn200 (BatchNormalization)| (None, 32, 128)| conv200[0][0]|
| add_1 (Add)               | (None, 32, 128)| bn200[0][0]|
| conv101 (Conv1D)          | (None, 32, 128)| add_1[0][0]|
| bn101 (BatchNormalization)| (None, 32, 128)| conv101[0][0]|
| conv201 (Conv1D)          | (None, 32, 128)| bn101[0][0]|
| bn201 (BatchNormalization)| (None, 32, 128)| conv201[0][0]|
| add_2 (Add)               | (None, 32, 128)| bn201[0][0]|
| add00 (Add)               | (None, 32, 128)| add_2[0][0]|
| maxpool00 (MaxPooling1D)  | (None, 32, 64)| add00[0][0]|
| conv01 (Conv1D)           | (None, 32, 64)| maxpool00[0][0]|
| bn01 (BatchNormalization) | (None, 32, 64)| conv01[0][0]|
| conv110 (Conv1D)          | (None, 32, 64)| bn01[0][0]|
| bn110 (BatchNormalization)| (None, 32, 64)| conv110[0][0]|
| conv210 (Conv1D)          | (None, 32, 64)| bn110[0][0]|
| bn210 (BatchNormalization)| (None, 32, 64)| conv210[0][0]|
| add_3 (Add)               | (None, 32, 64)| bn210[0][0]|
| conv111 (Conv1D)          | (None, 32, 64)| add_3[0][0]|
| bn111 (BatchNormalization)| (None, 32, 64)| conv111[0][0]|
| conv211 (Conv1D)          | (None, 32, 64)| bn111[0][0]|
| bn211 (BatchNormalization)| (None, 32, 64)| conv211[0][0]|
| input_2 ([InputLayer])    | (None, 2, 128)|              |
| lstm1 (LSTM)              | (None, 2, 8)| input_2[0][0]|
| add01 (Add)               | (None, 32, 64)| bn01[0][0]|
| bn_lstm (BatchNormalization)| (None, 2, 8)| lstm1[0][0]|
| maxpool1 (MaxPooling1D)   | (None, 32, 32)| add01[0][0]|
| dr1 (Dropout)             | (None, 2, 8)| bn_lstm[0][0]|
| flatten_1 (Flatten)       | (None, 1024)| maxpool1[0][0]|
| fl1 (Flatten)             | (None, 16)| dr1[0][0]|
| concat (Concatenate)      | (None, 1040)| flatten_1[0][0]|
| dense3 (Dense)            | (None, 256)| fl1[0][0]|
| bn11 (BatchNormalization) | (None, 256)| concat[0][0]|
| dr2 (Dropout)             | (None, 256)| dense3[0][0]|
| dense5 (Dense)            | (None, 10)| bn11[0][0]|
| aci (Activation)          | (None, 10)| dr2[0][0]|
| reshape_1 (Reshape)       | (None, 10)| dense5[0][0]|
|                           |              | aci[0][0]|

The ReLU function is expressed as [24], [31]:

\[
ReLU(x) = \max\{0, x\} = \begin{cases} 
0, & x < 0 \\
 x, & x \geq 0
\end{cases} \quad (20)
\]

One input layer, two IRSs, one LSTM, one fully connected (FC) layer, and one softmax layer make up the network. Thus, there are ten convolutional layers, two maxpooling layers, one LSTM, and three FC layers as a result (softmax layer is also a FC layer). The structure with 2 IRSs and 1 LSTM is selected after detailed analysis the influence of the depths on the recognition accuracy in Section V-A. I/Q samples are sent to IRSs and amplitude-phase samples are sent to LSTM, respectively. Studies have proved: LSTMs are not able to extract any meaningful representations from I/Q samples but perform well in amplitude-phase samples; The amplitude-phase supplied to the CNN model does not show better performance than I/Q samples [22]. Then, a concatenate layer merges the parallel branches. The data further passes a FC layer with 256 units and ReLU activation, which is followed by a dropout layer with the drop probability of 0.6. The final layer is also a FC layer, where the neuron number is consistent with the considered modulation number, and softmax activation. The BN layer is used to minimize training time and prevent gradients from exploding or disappearing [30]. To alleviate overfitting and compel the nodes to be more independent than usual, the regularization strategy of “dropout” is used to prevent updating the weights of part nodes [24]. The detailed structure of IRLNet is shown in Table 1.

IV. EXPERIMENTS BACKGROUND

A. DATASETS

We first serve the well-known DeepSig dataset named RadioML2016.10B to train or test the performance of the IRLNet and make a comparison with the state-of-the-art (SoA) models. More information about this dataset generated can be found in [32]. Then, we consider transfer learning [33] on two most challenging datasets named RadioML2018.01A [20] and HisarMod2019.1 [34]. All considered modulations in the datasets are split equally. The

TABLE 2. RadioML2016.10B dataset parameters.

| Parameter          | Value                          |
|--------------------|--------------------------------|
| Modulations        | Digital: BPSK, QPSK, 8PSK, 16QAM, 64QAM, GFSK, CPFSK, and 4PM |
| Number of dataset  | 1,200,000                      |
| SNR (dB)           | -20.2:18                       |
| Signal dimension of per sample | 2x128                        |
| Channel model      | center frequency offset (CFO), sample rate offset (SRO), AWGN, Rayleigh, Rician |
| Data source        | GNU Radio simulated            |

TABLE 3. RadioML2018.01A dataset parameters.

| Parameter          | Value                          |
|--------------------|--------------------------------|
| Modulations        | Digital: OOK, 4ASK, 8ASK, BPSK, QPSK, 8PSK, 16PSK, 32PSK, 16APS, 32APS, 64APS, 128APS, 16QAM, 32QAM, 64QAM, 128QAM, 256QAM, GMSK, OQPSK, Analog: AM-SSB-WC, AM-SSB-SC, AM-DSB-WC, AM-DSB-SC, FM |
| Number of dataset  | 2,555,904                      |
| SNR (dB)           | -20.2:30                       |
| Signal dimension of per sample | 1024x2                        |
| Channel model      | center frequency offset (CFO), sample rate offset (SRO), AWGN, Rayleigh, Rician |
| Data source        | GNU Radio simulated and Over-the-Air captured |
to other data. Building the training set is the first step on the training set while also attempting to generalize it based on the training dataset to achieve good performance. The training aims to optimize parameters of the network in network training. The training set for the IRLNet is 

\[ D = \{X_{train}, Y_{train}\} \]

where \( X_{train} = \{x^{(1)}, x^{(2)}, \ldots, x^{(m)}\} \), \( Y_{train} = \{y^{(1)}, y^{(2)}, \ldots, y^{(m)}\} \), and the number of samples in the training set is given by \( m \). The training set also includes \( N \) modulation modes.

The loss function is the key to training and cross-entropy is the most commonly used loss function for classification tasks. For a minibatch consisting \( N_B \) samples, we define the loss function as [24]:

\[
\Gamma(y, \hat{y}) = -\frac{1}{N_B} \sum_{i=1}^{N_B} y^{(i)} \log(\hat{y}^{(i)}) + (1 - y^{(i)}) \log(1 - \hat{y}^{(i)})
\]  

(21)

where \( y \) represents the true label and \( \hat{y} \) represents the predict label. Adaptive moment estimation (Adam) [35] has been proved to have a faster convergence rate than the most commonly used stochastic gradient descent (SGD) method [36] in theory and practice, and it is more suitable for large datasets. As an example, in iteration \( t \), the updating process for weight \( w^j \) to be optimized is as follows [37]:

(1) Calculate the loss function \( \Gamma \) for gradient in regard to \( w^j \):

\[
g_t^j = \frac{\partial \Gamma}{\partial w^j}
\]  

(22)
(2) Update the biased first moment estimate $m_{lt}^l$ and biased second raw moment estimate $V_{lt}^l$.

\[
\begin{align*}
  m_{lt}^l &= \beta_1 \cdot m_{lt-1}^l + (1 - \beta_1) \cdot g_{lt}^l \\
  V_{lt}^l &= \beta_2 \cdot V_{lt-1}^l + (1 - \beta_2) \cdot (g_{lt}^l)^2
\end{align*}
\] (23)

(3) Update the parameters.

\[w_{lt+1}^l = w_{lt}^l - lr \cdot m_{lt}^l / \sqrt{V_{lt}^l}\] (24)

where $l$ is the number of the layer located the weight $w_{lt}^l$, $lr = 0.001$ is the learning rate, and $\beta_1 = 0.9$, $\beta_2 = 0.999$ are the exponential decay rates. According to the analysis, the training and inference algorithms for IRLNet can be expressed as Algorithm 2.

D. TRANSFER LEARNING

One DL model is generally trained for a specific task, such as identifying several candidate modulation modes in a certain environment. Once the scenario or modulation set changes, the model needs to be retrained with new training data. The full model training requires considerable time and effort. Transfer learning is adopted in this paper which could reduce computational overhead. The concept of transfer learning is motivated by the fact that people can tackle new issues faster or obtain better solutions by applying learned previously knowledge [33]. For the modulation recognition problem, we perform transfer learning for the following two cases.

1) Diverse modulation modes: Depending on the requirements of different classification tasks, some modulation modes need to be added or removed. In this case, in addition to changing the training data, the model only needs to adjust the output layer of the network according to the number of modulation modes. That is to say, all the network parameter weights except for the ones in the top layer can be loaded as the initialization instead of randomly setting.

2) Different channel environments: The real-world channels vary over time, one-time training to contain all kinds of channel environments seems unrealistic. For the new channel environment, the model needs to be retrained from scratch, but the model structure can be completely retained or only fine-tuned.

E. IMPLEMENTATION DETAILS

The experiment consisted of two components. The first component optimizes the performance of the IRLNet based on the RadioML2016.10B and compares IRLNet with several SoA models in Section IV-B. The second component, the model is transferred to RadioML2018.01A and HisarMod2019.1 to verify the robustness of new modulation modes and transformed channels.

For all of our experiments, we use Keras with TensorFlow as the backend. We have two computers: one is equipped with AMD Ryzen 7 3800X CPU, 16 GB RAM, and NVIDIA GeForce RTX 2070 SUPER for training on RadioML2016.10B; the other is equipped with Intel (R) Xeon(R) Gold 6230 CPU, 256 GB RAM, and NVIDIA Quadro RTX 8000 for robustness testing on the other two datasets.

The training stage is set at 200 epochs, and the gradient update batch size is set at 1024. An early stopping strategy is also adopted. Under the condition that the validation loss does not decrease within five epochs, we multiply the learning rate by a factor of 0.01 to increase training performance. After adjusting, if the validation loss still does not decrease in the next ten epochs, the training will stop. In the testing stage, we split the testing set by SNR. The model calculates the probability that each sample belongs to each modulation mode, and the highest probability category is the predicted label of the corresponding sample. The recognition accuracy is calculated by comparing the real and the predicted labels.
V. RESULTS AND DISCUSSION

A. PERFORMANCE ON RADIOML 2016.10B

1) CLASSIFIER PERFORMANCE BY DEPTH

The ability of large neural network models to accurately reflect complex features is influenced by the model size. Too shallow a model will result in poor performance, while too deep a model will result in too high a complexity but no significant benefit to performance improvement. To achieve the best performance with the lowest complexity, we measure the classifier performance by varying the number of IRSs and LSTMs. Based on the recognition results plotted in Fig. 6, we see the increasing classification accuracy as we introduce more IRSs and LSTMs within the network architecture (i.e., making the network deeper). Surprisingly, when the number of LSTMs increase to four, the performance of the model suddenly decline, which may be overfitting due to the continuous deepening of the network. The recognition accuracy is almost no longer improved (or even decreased) to add more IRS or LSTM, that is to say, the model contains two IRSs and one LSTM is the best among the explored combinations for RML2016.10B.

2) TRAINING PERFORMANCE

The accuracy and loss in the training stage are used to evaluate the training performance of models. All seven models had completed their training in less than 200 epochs. The accuracy and loss in the training stage can be seen in Fig. 7(a) and Fig. 7(b), respectively. The result of training and validation shows, in the early training stage, IRLNet achieves the best training effect at the fastest convergence speed, followed by LSTM. ResNet has a strong early training effect, but there is noticeable jitter during the training process. The training effects of the other four models are not ideal, and even if more epoch times are spent, they cannot achieve the same effect as IRLNet.

3) RECOGNITION PERFORMANCE

The performance of the seven models is measured using accuracy, the most common evaluation metric for classification problems. As shown in Fig. 8, except for the LSTM model, the classification accuracy of IRLNet is significantly better than the other five models. When the SNR is greater than 0 dB, the recognition accuracy of IRLNet all exceeds 90%, and the average recognition accuracy of the SNR in the range of 0-18 dB is higher than the other five models by 3%-13%. However, the recognition accuracies of all the models under the low SNRs (e.g., -20 dB) are approximately identical. This may likely due to the fact that at low SNRs, a significant amount of interference information overwhelms much of the signal characteristics, resulting in low recognition accuracy for most modulation modes and thereby lowering overall recognition accuracy.
Following further exploration, we find that the main reason for the five models which are less effective in modulation recognition is that they are difficult to recognize high-order modulations. As shown in Fig. 9, even at a higher SNR (SNR = 18 dB), models other than IRLNet and LSTM have a certain degree of confusion in recognizing 16QAM and 64QAM, some are even unable to distinguish them at all. There is another area of confusion between WBFM and AM-DSB in the figures. Both belong to continuous modulation make the variations between them are limited on the complex panel. Furthermore, generated from sampling analog audio signals which contain silent intervals, exacerbating the confusion between WBFM and AM-DSB.

4) COMPUTATIONAL COMPLEXITY

We assess the computational complexity of the models in addition to their recognition accuracy. Both spatial and temporal complexity are considered in computational complexity analysis, and the comparison results among seven models is shown in Table 6. Spatial complexity refers to the trainable parameter number. As indicated by the table, except for the CLDNN and ResNet, IRLNet has the smallest number of trainable parameters. IRLNet has about 96.9% fewer parameters than Inception, 90.2% fewer parameters than CNN4, 42.1% fewer parameters than CNN2 and 12.2% fewer parameters than LSTM.

We measure the temporal complexity by the training and inference time. The comparison among seven models is shown in Table 6 and Fig. 10. The IRLNet takes approximately 1589 seconds for model training, longer than CNN2 and ResNet, but shorter than the other four models. The CLDNN costs the longest training time. Since the model is always trained offline and once deployed is not often retrained, the inference time after model deployment is more critical. The IRLNet takes about 4.05 seconds to

![FIGURE 9. Confusion matrices of the IRLNet with others on RadioML 2016.10B at 18 dB of SNR. (a) CNN2. (b) CNN4. (c) CLDNN. (d) Inception. (e) ResNet. (f) LSTM. (g) IRLNet.](image)

![FIGURE 10. The training and inference time of seven models.](image)
infer 360,000 samples in the testing set, which is about 75% faster than LSTM. IRLNet requires the least time in the inference stage except CNN2, but it can achieve the optimal results. On the contrary, LSTM consumes about four times for inference compared to IRLNet. While comparable accuracy can be achieved, it is intolerable for real applications.

B. TRANSFER LEARNING PERFORMANCE

Transfer learning aims at enhancing the performance of target learners on target domains by migrating knowledge from distinct but linked source domains [38]. In this section, we conceive signal classification on a different dataset as a transfer learning issue. As shown in section IV − A, the channel environments of RadioML2018.01 and RadioML2016.10B are the same, but the modulation modes are different. Therefore, for RadioML2018.01, we load the network parameter weights of all layers except for the last fully connected layer as initialization, and the number of neurons in the last fully connected layer is set to 24 according to the number of modulation modes. As shown in Fig. 11 and Fig. 12, the descent speed of the loss curve is faster than retrain from scratch and the training time of the network is significantly shortened. As for HisarMod2019.1, the model needs to be retrained due to the completely different channel environments. In addition, the modulation modes are also different. Therefore, the number of neurons in the last fully connected layer also needs to be adjusted, i.e., adjusted to 26.

To verify the transfer learning ability of the IRLNet model, we consider diverse modulation modes and different channel environments. In addition, we compare it with the LSTM model which demonstrated the similarity recognition accuracy with IRLNet on RadioML 2016.10B in section V-A.

1) CLASSIFIER PERFORMANCE BY SAMPLE LENGTH

Since the sample length of RadioML2016.10B dataset is limited, we discuss the impact of sample length on the model recognition accuracy on two new datasets. Fig. 13 shows
how the model performance varies by sample length. The performance of the network steadily improves as the length of the training samples increases. The variation trend of the accuracy of different model sizes explained as follows: the longer the training sample data is, the more signal features fed back to the network, and the higher model accuracy achieved.

When the frame size decreased from 1024 to 512 at 0 dB, the recognition accuracy on the two datasets decreased by 3.24% and 4.80%, respectively. As the sample length further decreases, the accuracy continues to decrease. Although a longer sample length brings higher recognition accuracy, it comes at the cost of more computation and memory. Thus, the trade-off between the two is critical for real applications. In the following experiments, we all take the sample length of 1024.

2) DIVERSE MODULATION MODES

As communication advances, the modulation modes increase rapidly and multiply, the modulation recognition model requires to capable of reliably recognizing new modulation modes. Some new modulation modes are included in RadioML 2018.01A and HisarMod 2019.1, containing high-order modulations and analog modulations that are easy to confuse. In Fig. 14 and Fig. 15, we show the performance of the classifier for each modulation mode in the RadioML 2018.01A and HisarMod 2019.1. Almost all modulation modes in the two datasets can achieve robust performance around $SNR = 10$ dB. The results of the two datasets are both grouped into three categories for better visualization, where RadioML 2018.01A is divided into PSK + APSK, ASK + QAM, and low-order + analog; HisarMod 2019.1 is divided into PSK + FSK, QAM + PAM, and analog.

In the RadioML 2018.01A, IRLNet has a better recognition performance for all signals with lower information rates and achieves faster convergence to the maximum value compared with high-order modulation modes. High-order modulations are greatly affected by various channel impairments, and their recognition rate continue to decrease as the order increases, especially for high-order QAMs. For instance, when the QAM order increased from 32 to 64, 128, and 256, the recognition accuracy decreased significantly, by 20.05%, 10.10%, and 23.33% at the same $SNR$ (i.e., +10 dB), respectively. Unexpectedly, for AM-SSB-WC, IRLNet has poor recognition performance. Although recognition accuracy can reach 77.66% when the $SNR$ is -2dB, its performance decreases as the $SNR$ increases obviously. This is due to confusion between AM-SSB-WC and AM-SSB-SC, as shown in Fig. 17(a). Both of them are Single-Sideband AM, which means the differences between them on the complicated panel are negligible. Furthermore, the simulated
AM-SSB-WC and AM-SSB-SC data in the dataset were created by sampling analog audio signals, which have silence periods, worsening the issue. By fine-tuning the network, such as deepening the network, the accuracy of high-order modulations and AM-SSB-SC has been improved to varying degrees. As shown in Fig. 16, when the number of IRS increase from 2 to 4, the accuracy of 64QAM, 128QAM, and 256QAM increase by 6.31%, 5.91%, and 12.91% at SNR = 16 dB. As for AM-SSB-WC, it improves the most when the SNR is 4 dB, increasing by 66.41%.

In the HisarMod 2019.1, the recognition accuracy of all modulation modes increases as the SNR increases, and when the SNR is around 10 dB, all modulation modes can reach an accuracy of 100%. At very low SNR (i.e., -20 dB), some modulation modes (2FSK, 4FSK, BPSK, 8PSK, 16PSK, 4PAM, 8PAM, 8QAM, 16QAM, 256QAM, AM-DSB-SC, FM) can achieve greater than 50% recognition accuracy, and even more surprising is that some modulation modes (such as 32PSK, 32QAM) can achieve 100% accuracy.

Fig. 17 and Fig. 18 show the confusion matrices for IRLNet at SNR = 16 dB. From the perspective of modulation recognition, except for the slight confusion among some high-order QAMs and analog modulations, IRLNet has a high recognition accuracy of over 90% for almost all classes in RadioML 2018.01A. IRLNet has the worst recognition effect on AM-SSB-WC and misjudges 50% of AM-SSB-WC as AM-SSB-SC as depicted in Fig. 17(a). It may because that the signal involves a small carrier component, making it more difficult to determine whether the carrier should be suppressed or not. Unsurprisingly, there is confusion among high-order QAMs (64QAM, 128QAM,
3) DIFFERENT CHANNEL ENVIRONMENTS

The real-world channels vary over time, and design a network model explicitly for each channel seems impractical for the high cost of model designing and optimization. The robustness of the model when dealing with time-varying channels is particularly significant. The over-the-air captured data and Nakagami channel simulated data are included in the datasets of RadioML2018.01A and HisarMod2019.1, respectively, which are not included on RadioML 2016.10B. Since LSTM has been proved to have similar recognition accuracy with IRLNet in RadioML 2016.10B, we use it as a comparison to conduct robustness tests in the two datasets. Besides, we also use the ResNet and CNN model proposed by the datasets generators as a comparison in RadioML2018.01A and HisarMod2019, respectively.

As shown in Fig. 19, IRLNet shows better recognition performance on both datasets. In RadioML2018.01A, IRLNet shows obvious advantages over ResNet and LSTM, both in low SNRs and high SNRs. As we can see in Fig. 19(a), compared with ResNet and LSTM, IRLNet improves the recognition accuracy by 20.5% and 29% when SNR = 10 dB. Compared with ResNet and LSTM, the recognition accuracy of IRLNet maximum increases by 22.2% and 29%, respectively. In HisarMod2019, IRLNet shows excellent performance, but LSTM is almost impossible for modulation recognition. Compared with LSTM, the recognition accuracy of IRLNet maximum increases by 69.33%. Compared with CNN, IRLNet improves the recognition accuracy by 22%-60% at the range from -20 dB to 0 dB of SNRs. That is to say, IRLNet is extremely robust in a variety of channel environments.

VI. CONCLUSION

To solve the signal modulation recognition problem, we first add an extra skip connection based on the basic RS and construct the new IRS. The IRS maintains the more initial residual information, learns the signal features in deep and shallow, and achieves various degrees of feature extraction.
Then, we connect several IRSs and LSTM in parallel to form the modulation recognition model IRLNet. Compared with traditional algorithms, IRLNet can automatically extract deep and shallow features from data without manual extraction. And compared with the SoA DL algorithms, the model training process is more stable, and has low spatial-temporal complexity which makes the model much easier to deploy in a variety of real scenarios. In addition, IRLNet shows excellent recognition accuracy. Firstly, on the dataset of RadioML2016.10B, the average recognition accuracy reaches 93% from 0 dB to 18 dB of SNR. Secondly, it also has strong robustness, when transferred to RadioML2018.01A and HisarMod2019.1, the recognition accuracy can reach 95% and 100% at high SNR, respectively. Finally, IRLNet also shows good recognition performance for high-order modulations. The recognition accuracy of the model is more than 90% on both datasets for high-order modulations, such as 12QAM when SNR = 10 dB. Although we have proven the robustness of IRLNet for various channel environments, the real-world channels constantly change over time which may degrade the performance of the model. Therefore, in future work, we plan to design a channel estimator based on neural network to recover the distorted channel.

REFERENCES

[1] 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. Netw., vol. 5, no. 3, pp. 469–478, Sep. 2019.

[2] S. Huang, R. Dai, J. Huang, Y. Yao, Y. Gao, P. Ning, and Z. Feng, “Automatic modulation classification using gated recurrent residual network,” IEEE Internet Things J., vol. 7, no. 8, pp. 7795–7807, Aug. 2020.

[3] 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.

[4] F. Hameed, O. A. Dobre, and D. Popescu, “On the likelihood-based approach to modulation classification,” IEEE Trans. Wireless Commun., vol. 8, no. 12, pp. 5884–5892, Dec. 2009.

[5] O. A. Dobre, Z. Bar-Ness, and W. Su, “Survey of automatic modulation classification techniques: Classical approaches and new trends,” IET Commun., vol. 1, no. 2, pp. 137–156, Apr. 2007.

[6] A. Swami and B. M. Sadler, “Hierarchical digital modulation classification using cumulants,” IEEE Trans. Commun., vol. 48, no. 3, pp. 416–429, Mar. 2000.

[7] S. Huang, Y. Yao, Z. Wei, Z. Feng, and P. Zhang, “Automatic modulation classification of overlapped sources using multiple cumulants,” IEEE Trans. Veh. Technol., vol. 66, no. 7, pp. 6089–6101, Jul. 2016.

[8] D.-C. Chang and P.-K. Shih, “Cumulants-based modulation classification technique in multipath fading channels,” IET Commun., vol. 9, no. 6, pp. 828–835, 2015.

[9] K. K. Ho, P. Prokopiv, and Y. C. T. Chan, “Modulation identification of digital signals by the wavelet transform,” IEEE Proc. Radar Sonar Navigat., vol. 147, no. 4, pp. 169–176, Aug. 2000.

[10] O. A. Dobre, M. Oner, S. Rajan, and R. Inok, “Cyclostationarity-based robust algorithms for QAM signal identification,” IEEE Commun. Lett., vol. 16, no. 1, pp. 12–15, Jan. 2012.

[11] G. B. Tenze, T. Huynh-The, J.-M. Lee, and D.-S. Kim, “Sparsely connected CNN for efficient automatic modulation recognition,” IEEE Trans. Veh. Technol., vol. 69, no. 12, pp. 15557–15568, Dec. 2020.

[12] S. Majhi, R. Gupta, W. Xiang, and S. Glisic, “Hierarchical hypothesis and feature-base blind modulation classification for linearly modulated signals,” IEEE Trans. Veh. Technol., vol. 66, no. 12, pp. 11057–11069, Dec. 2017.

[13] T. Huynh-The, C.-H. Hua, T.-T. Ngo, and D.-S. Kim, “Image representation of pose-transition feature for 3D skeleton-based action recognition,” Inf. Sci., vol. 513, pp. 112–126, Mar. 2020.
HUOGEN YANG received the Ph.D. degree from Zhejiang University, Zhejiang, China, in 2014. From 2016 to 2017, he was a Visiting Researcher with the University of Florida, USA. He is currently a Professor with the College of Science, Jiangxi University of Science and Technology. His research interests include deep learning, intelligent computing, and computer aided geometric design.

LINGZHU ZHAO was born in Shaoxing, Zhejiang, China, in 1995. She received the B.E. degree in network engineering from Jiaxing University, Zhejiang, China, in 2018. She is currently pursuing the master’s degree with the College of Science, Jiangxi University of Science and Technology, China. Her main research interests include deep learning and intelligent communication.

GUANGXUE YUE received the M.Sc. and Ph.D. degrees from Hunan University, in 2004 and 2012, respectively. In 2016, he was a Visiting Researcher at Stony Brook University, NY, USA. He is currently a Professor with the College of Information Science and Engineering, Jiaxing University. His main research interests include cloudy convergence and cooperative services, mobile cloud computing, deep learning, and intelligent communication.

BOLIN MA received the Ph.D. degree from Beijing Normal University, China, in 1995. He is currently a Professor with the College of Data Science, Jiaxing University. His main research interests include wavelet analysis, harmonic analysis, and intelligent communication modeling.

WEI LI received the Ph.D. degree in information and communication engineering from the Huazhong University of Science and Technology, Wuhan, China, in 2020. He is currently a Lecturer with the College of Information Science and Engineering, Jiaxing University. His research interests include vehicular communications and channel modeling.