

Article

Automatic Modulation Recognition Based on a DCN-BiLSTM Network

Kai Liu *, Wanjun Gao  and Qinghua Huang

School of Communication and Information Engineering, Shanghai University, Shanghai 200444, China; WanJun0124@shu.edu.cn (W.G.); qinghua@shu.edu.cn (Q.H.)

* Correspondence: liukai@shu.edu.cn

Abstract: Automatic modulation recognition (AMR) is a significant technology in noncooperative wireless communication systems. This paper proposes a deep complex network that cascades the bidirectional long short-term memory network (DCN-BiLSTM) for AMR. In view of the fact that the convolution operation of the traditional convolutional neural network (CNN) loses the partial phase information of the modulated signal, resulting in low recognition accuracy, we first apply a deep complex network (DCN) to extract the features of the modulated signal containing phase and amplitude information. Then, we cascade bidirectional long short-term memory (BiLSTM) layers to build a bidirectional long short-term memory model according to the extracted features. The BiLSTM layers can extract the contextual information of signals well and address the long-term dependence problems. Next, we feed the features into a fully connected layer. Finally, a softmax classifier is used to perform classification. Simulation experiments show that the performance of our proposed algorithm is better than that of other neural network recognition algorithms. When the signal-to-noise ratio (SNR) exceeds 4 dB, our model's recognition rate for the 11 modulation signals can reach 90%.

Keywords: automatic modulation recognition; deep complex network; convolutional neural network; bidirectional long short-term memory network



Citation: Liu, K.; Gao, W.; Huang, Q. Automatic Modulation Recognition Based on a DCN-BiLSTM Network. *Sensors* **2021**, *21*, 1577. <https://doi.org/10.3390/s21051577>

Received: 17 January 2021

Accepted: 19 February 2021

Published: 24 February 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

As a significant technology for noncooperative wireless communication systems, automatic modulation recognition (AMR) plays an important role in practical civil and military applications, such as cognitive radio, interference recognition and spectrum monitoring [1]. In the absence of prior knowledge, it can identify the modulation type of an intercepted signal, providing parameter information for subsequent demodulation [2].

Traditional AMR algorithms can be divided into two categories. One is based on maximum likelihood (ML) theory [3], and the other is a feature-based (FB) method [4]. The first approach uses probability theory, hypothesis test theory and an appropriate decision strategy to solve the AMR problem. The feature-based approach first extracts the modulated signal characteristics and then completes the recognition using classifiers. In feature-based methods, the number of selected features influences the recognition performance. The main features used for modulation signal identification include instantaneous amplitude, phase, frequency, high-order cumulant [5], cyclic spectrum [6] and wavelet characteristics [7]. Many of the classifiers used are based on machine learning algorithms; these include decision trees, support vector machines (SVMs) [8] and artificial neural networks (ANNs) [9].

In recent years, deep learning (DL), which is a powerful machine learning approach, has achieved great success in diverse fields such as image classification [10] and speech recognition [11]. The concept of DL comes from the research of ANNs. A multilayer perceptron with multiple hidden layers is a DL structure. DL forms a more abstract high-level representation attribute category or feature by combining low-level features to

discover distributed feature representations of data. Its goal is to allow machines to have the ability to analyze and learn similar to humans and recognize data such as text, images and sounds. DL solves nonlinear classification problems by using activation functions and uses regularization to improve the robustness of the model [12]. The DL-based methods cascade multilayer nonlinear processing units to extract features. This approach can automatically optimize the extracted features to minimize later classification errors.

Many scholars have applied DL to the field of AMR in recent years. For example, Lee et al. [13] proposed a new method that calculated 28 statistical characteristics of five modulation signals; these were then sent to a fully connected feedforward network to conduct classification. Wang Yu et al. [14] trained CNN on samples composed of in-phase and quadrature component signals to distinguish modulation patterns that are relatively easy to identify. At the same time, a CNN based on the constellation map was designed to identify modulation modes that were difficult to distinguish in previous CNNs and improved the ability to classify QAM signals under low SNRs. Li et al. [12] studied the AMR method based on the original IQ signal under the parameter estimation error. First, the influence of parameter estimation error on the performance of CNN classifier is analyzed. Then, an AMR method based on spatial transformation network (STN) is proposed, which improves the robustness under parameter estimation errors. Li et al. [15] proposed a deep joint learning algorithm based on CNN and kernel collaborative representation and discriminative projection (KCRDP), including deep learning and kernel dictionary learning, which improves the adaptability of small samples and reduces the computational complexity without prior knowledge and feature enhancement processing.

In 2016, O'Shea et al. [16] generated a public dataset of modulated signals, named RML2016.10a, using GNU Radio software; then, they used a convolutional neural network (CNN) to identify the modulated signals. Subsequent studies have also adopted this dataset for AMR research. That same year, O'Shea [17] proposed a method to optimize the CNN structure. West et al. [18] applied a CNN, a residual network (Resnet), a convolutional long short-term memory deep neural network (CLDNN) and an Inception network to the modulated signal identification task and compared their respective recognition performances. The results show that the modulation signal recognition performance is not solely dependent on the network depth. Zhang et al. [19] proposed a preprocessing signal representation that combined the in-phase, quadrature and fourth-order statistics of the modulated signals. Omar S. Mossad et al. [20] proposed a CNN that used a multitask learning scheme (MTL-CNN) to reduce the confusion between similar classes. Kumar et al. [21] introduced a signal distortion correction module (CM) to estimate the carrier frequency offset and phase noise of the received signal to improve modulation recognition accuracy of deep learning schemes. Liu [22] proposed a group lasso based lightweight DNN for AMR which can learn to prune via automatically removing the neurons in hidden layers.

In 2020, Jakob et al. [23] used a linear combination to enable the DL architecture to perform complex convolutions (CVCs) and learn the characteristics of the real and imaginary parts of modulated signals. Wu et al. [24] constructed a CNN followed by a long short-term memory (LSTM) model as the classifier (CNN-LSTM) to efficiently explore the temporal and spatial correlation. However, to the best of our knowledge, none of the currently available methods fully consider the phase and contextual information of the modulated signals simultaneously.

In view of the current problems, this paper proposes a deep complex network that cascades bidirectional long short-term memory network (DCN-BiLSTM). First, deep complex network (DCN) layers with convolution kernels of different scales are connected; then, the bidirectional long short-term memory (BiLSTM) layers are cascaded. Finally, a softmax classifier is used to classify 11 kinds of digitally modulated signals. The main contributions of this paper are as follows:

- (1) To the best of our knowledge, this study is the first to cascade DCN and BiLSTM models and apply them in the AMR field.

- (2) We demonstrate the effectiveness of the DCN-BiLSTM network through experiments. The experimental results show that the performance of the proposed algorithm is better than that of other neural network recognition algorithms. When the signal-to-noise ratio (SNR) exceeds 4 dB, the recognition rate of our proposed model on the 11 modulation signals can reach 90%.

The remainder of this paper is organized as follows. Section 2 shows the signal model. Section 3 proposes the DCN-BiLSTM network model and introduces its components. In Section 4, we report the results of an experiment conducted to evaluate the proposed method and provide the optimal parameter configuration for the DCN-BiLSTM model. In addition, we use a cross-validation method to evaluate the network. Finally, Section 5 concludes this paper.

2. Signal Model

This paper uses the open dataset named RML2016.10a [16]. Figure 1 illustrates the dataset generation technique used in [16]. For the Rayleigh fading channel, the received signal can be expressed as

$$r(n) = \sum_{l=1}^L h_l(n)s(n - n_l)e^{j2\pi f_{dl}nT_0} + \omega(n) \quad (1)$$

where $s(n)$ is the modulated signal sent by the communication transmitter and L is the number of multipaths; $h_l(n)$ is the Rayleigh fading factor of the l th path; n_l is the delay of the l th path; f_{dl} represents the Doppler frequency; and $\omega(n)$ is additive white Gaussian noise. In addition, to ensure that the channel model is similar to a real channel, the channel model of this dataset includes the sampling rate and carrier rate offsets. The specific modulation types and parameters are shown in Table 1.

Table 1. The specific dataset parameters.

Data Source	RML2016.10a
Modulation types	AM-DSB, AM-SSB, WBFM, BPSK, 8PSK, QPSK, CPFSK, GFSK, PAM4, 16QAM, 64QAM
Data length	128
Data dimension	2×128
Sampling frequency	200 kHz
Sampling rate offset standard deviation	0.01 Hz
Maximum sampling rate offset	50 Hz
Carrier frequency offset standard deviation	0.01 Hz
Maximum carrier rate offset	500 Hz
Number of sinusoids used in frequency selective fading	8
Maximum doppler frequency used in fading	1
Fading model	Rician
Rician K-factor	4
Fractional sample delays for the power delay profile	[0.0, 0.9, 1.7]
Magnitudes corresponding to each delay time	[1, 0.8, 0.3]
Filter length to interpolate the power delay profile	8
Standard deviation of the AWGN process	$\sqrt{10^{-\frac{SNR}{10}}}$
SNR(dB)	−20:2:18

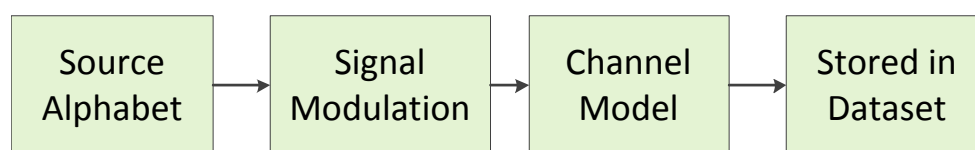


Figure 1. Data generation scheme in [16].

For a received signal, if we know the modulation type range, sampling frequency, sampling rate offset range, carrier rate offset range and the other signal parameters included in Table 1, we can use the recognition system model proposed in Section 3.

3. The Proposed Algorithm

3.1. DCN-BiLSTM Network Model

As shown in Figure 2, we propose the DCN-BiLSTM network model for AMR. First, we preprocess the received signal and divide it into I and Q components. Then, we send I and Q to the DCN-BiLSTM network, which is designed for identification. Finally, we obtain the identified signal type. The network has four parts: an input layer, DCN layers, BiLSTM layers and a fully connected layer.

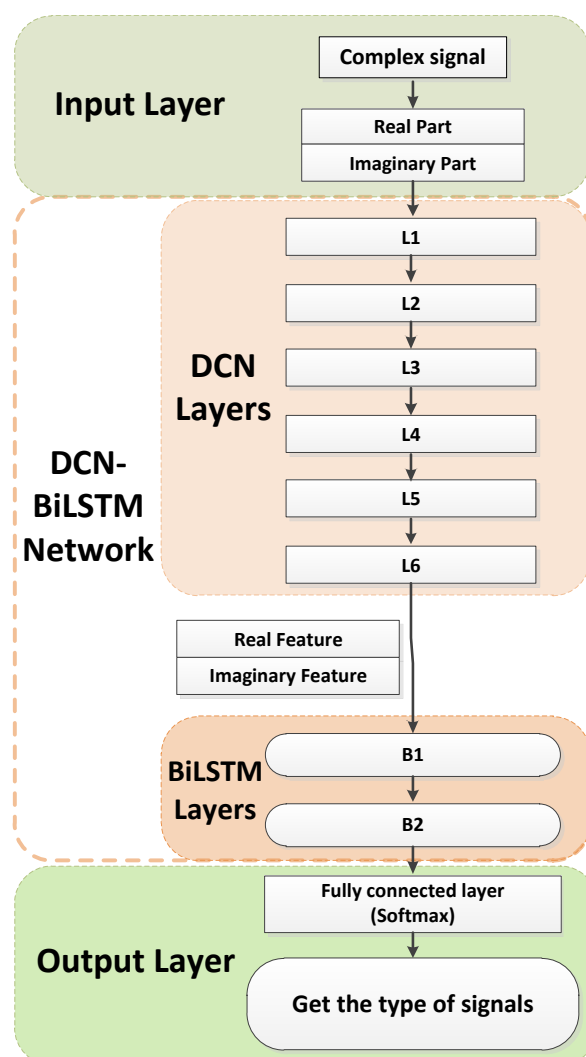


Figure 2. DCN-BiLSTM network identification system model.

In the DCN layers, the I-channel data of the signal are convoluted with the I-channel convolution kernel of the complex-valued convolution kernel, while the Q-channel data are

convoluted with the Q-channel convolution kernel. After convolution, the real features and imaginary features are output. The activation function for the complex-valued convolution is a rectified linear unit (ReLU) function, which is defined as follows:

$$\text{ReLU}(x) = \begin{cases} x, & x > 0 \\ 0, & x \leq 0 \end{cases} \quad (2)$$

where x is the input. When $x > 0$, the activation function has a linear relationship with the input.

The BiLSTM layers connect the contextual information among signals and build a bidirectional long short-term memory model for the extracted features. The fully connected layer uses the softmax activation function to output the predicted probability of the modulation information.

$$y_i = S(z)_i = \frac{e^{z_i}}{\sum_{j=1}^C e^{z_j}}, i = 1, \dots, C, \quad (3)$$

where z is the output of the previous layer and eventually forms the input to the fully connected layer; C is the input dimension and the number of modulation types; and y_i is the probability of an unknown signal being predicted as category i .

The algorithm uses the cross-entropy loss function to calculate the gradient in reverse to update the bias and weight values. The back-propagation update process is as follows

$$\theta_{n+1} = \theta_n - \eta \frac{\partial \varphi(y, f(x, \theta))}{\partial \theta_n}, \quad (4)$$

where θ_n is the bias or weight of the last moment; η is the learning rate; and φ is the loss function.

3.2. Deep Complex-Valued Network Module (DCN)

The DCN [25] layers are composed of many complex-valued convolution kernels. These complex-valued convolution kernels of different scales are stacked together to perform a hierarchical convolution operation on the input signal.

In the complex-valued convolution operation, the real and imaginary parts are convolved separately. In Cartesian notation, the complex input matrix is defined as $M = M_R + iM_I$. Similarly, the complex-valued convolution kernel matrix is defined as $K = K_R + iK_I$. These parameters, including M_R, M_I, K_R, K_I , are all real-valued matrices. The complex-valued convolution expression is

$$M' = M * K = (M_R + iM_I) * (K_R + iK_I). \quad (5)$$

where $*$ is the operation of convolution. The above formula can be expanded to

$$M' = \{M_R * K_R - M_I * K_I\} + i\{M_R * K_I + M_I * K_R\}. \quad (6)$$

Figure 3 shows a schematic diagram of the complex-valued convolution operation. The real and imaginary convolutions of the complex-valued signal are expressed as follows:

$$\begin{bmatrix} \text{Re}\{M * K\} \\ \text{Im}\{M * K\} \end{bmatrix} = \begin{bmatrix} K_R & -K_I \\ K_I & K_R \end{bmatrix} * \begin{bmatrix} M_R \\ M_I \end{bmatrix}, \quad (7)$$

where $\text{Re}\{M * K\}$ is the real part of the signal and $\text{Im}\{M * K\}$ is the imaginary part of the signal.

The outputs of the DCN layers will carry phase information and are used as the input to the next layer.

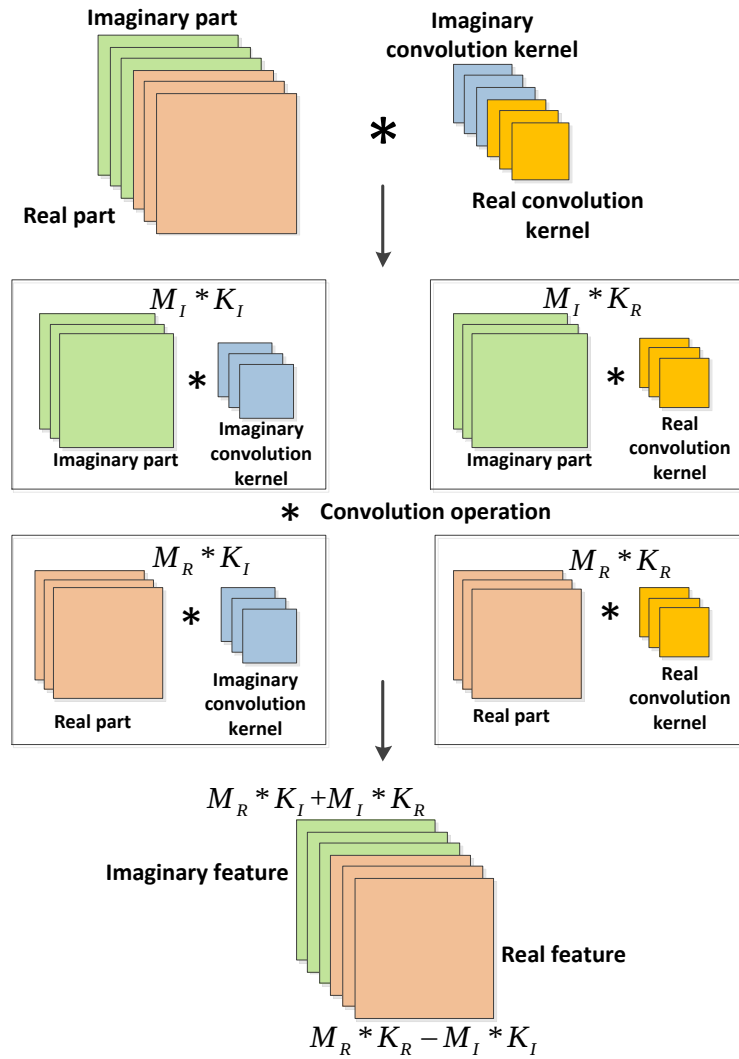


Figure 3. Schematic diagram of the complex-valued convolution operation.

3.3. Bidirectional Long Short-Term Memory Module (BiLSTM)

We cascade a BiLSTM behind the DCN to facilitate the extraction of contextual information of features. The BiLSTM network is composed of both forward and reverse LSTM networks [26]. As shown in Figure 4, the LSTM network contains many LSTM memory cells [27] that each includes three control units, namely, an input gate, a forget gate and an output gate. Figure 5 shows a diagram of the LSTM memory cells [27].

In Figure 5, t represents the current moment. The input feature sequence x_t and the output sequence of the previous time h_{t-1} are input to the memory cell. The forgetting factor f_t is obtained via the forgetting gate and is expressed as follows:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f), \quad (8)$$

where W_f is the connection matrix of x_t, h_{t-1} . b_f is the offset matrix, and σ is the sigmoid activation function, which is used to control the information-passing rate. The expression is

$$\sigma(x) = \frac{1}{1 + e^{-x}}, \quad (9)$$

where the output value of σ is between 0 and 1. The input gate and memory status update information are

$$\begin{aligned} i_t &= \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \\ \tilde{C}_t &= \tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \\ C_t &= f_t * C_{t-1} + i_t * \tilde{C}_t \end{aligned} \quad (10)$$

where i_t is the output of the input gate and $\tanh$ is an activation function that generates candidate values $\tilde{C}_t$. In addition, $\tilde{C}_t$ participates in the calculation to obtain the memory state C_t .

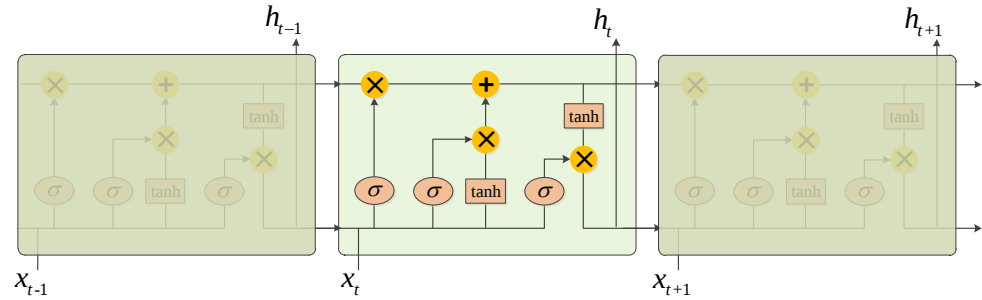


Figure 4. LSTM network.

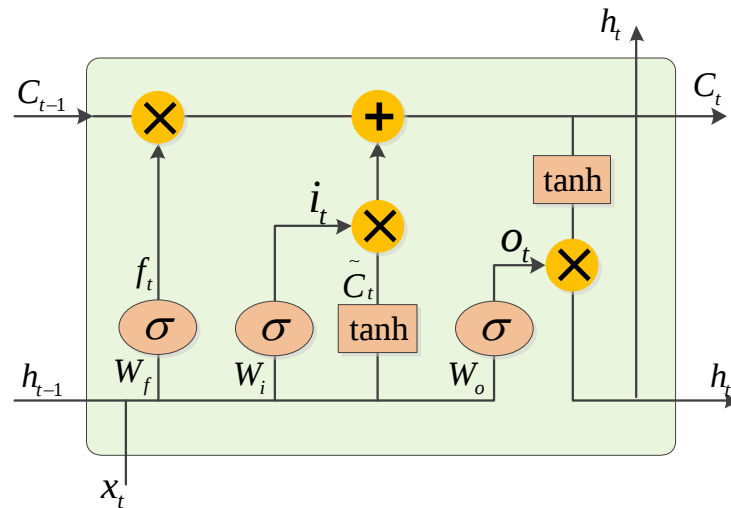


Figure 5. An LSTM memory cell.

Among these various components, the memory state C_t is the most important because it can allow information to flow through the entire link under the condition that it must remain unchanged, ensuring the integrity of the information for a long time. The output gate control factor o_t determines whether to output information h_t and is expressed as follows:

$$\begin{aligned} o_t &= \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \\ h_t &= o_t * \tanh(C_t) \end{aligned} \quad (11)$$

where W_o is the output gate weight matrix and b_o is the offset matrix. Compared with C_t , h_t contains more information about the current moment. Therefore, h_t represents short-term memory, while C_t represents long-term memory.

However, a classical LSTM considers only information from the previous moment. To consider both the former moment and the next moment together, the BiLSTM [26] adds reverse operations based on the LSTM model in [28,29]. Figure 6 shows a structural operation graph of the BiLSTM.

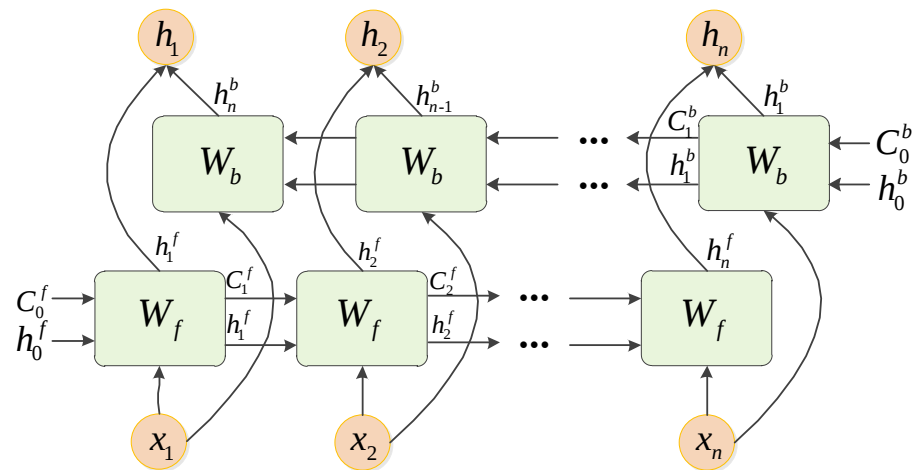


Figure 6. The structural operation graph of BiLSTM.

As shown in Figure 6, the BiLSTM reverses the input sequence and calculates the output again in the same way as an LSTM. The final result is a stack of the forward LSTM and the reverse LSTM, which achieves the goal of considering the contextual information. The final outputs of the BiLSTM are h_t , where $t = 1, 2, \dots, n$, can be expressed as shown in Figure 7. The expression of h_t is as follows:

$$h_t = [h_t^f, h_t^b]. \quad (12)$$

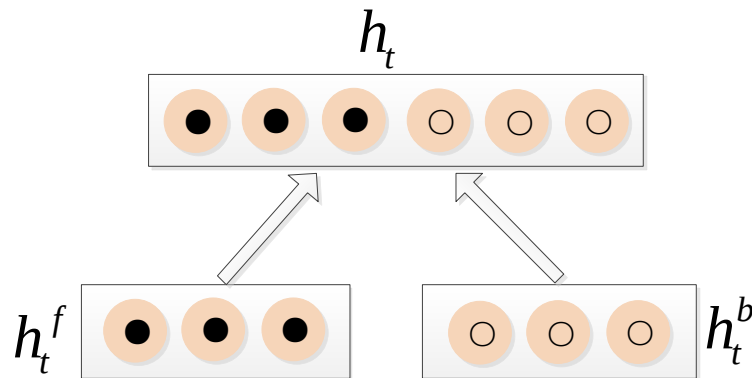


Figure 7. The relationship between h_t and h_t^f, h_t^b .

The output features of the BiLSTM are mapped into a sparse space by the fully connected layer. After the network is trained, the algorithm outputs the classification probability of the corresponding modulation modes.

4. Experiment Results and Discussions

The relevant platform and software settings for this experiment are shown in Table 2. There are 1000 samples of each modulated signal for each SNR comprising a total of samples is 220,000 samples. The ratio of training sets to test sets is 8:2. We use the np.random.choice function to implement the proportional selection of the dataset to obtain the training sets and the test sets. In this experiment, we performed the following steps.

Step1 Initialize the DCN-BiLSTM network randomly, extract a specified number of samples in the training sets and input them into the network for training.

Step2 Compare the classification result obtained in the last layer of the network with the actual type; use the cross-entropy function to calculate the network loss value; and adjust the network weight value through the optimization algorithm.

Step3 Before the next training starts, use the loss value as a standard to measure the network performance. When it does not drop within 10 iterations, the training is stopped.

Step4 Repeat Steps 2–4 until the maximum number of training is reached or the conditions for premature termination of training are met. The maximum number of training in this article is 150. After training, the weights are saved and the classification model is output.

Step5 Input the test sets into the trained model to obtain the recognition result.

Table 2. The relevant experimental platform and software settings.

Signal generation software platform	GNU Radio
Deep learning simulation platform	TensorFlow 1.7.0
Deep learning library	Keras
Hardware acceleration platform	NVIDIA GTX1080Ti

4.1. Algorithm Performance Comparison

We selected the recognition algorithms based on a CNN [17], Resnet [18], Inception [18], CLDNN [18], MTL-CNN [20], CVC [23] and CNN-LSTM [24] as benchmark models. A performance comparison chart for these eight recognition algorithms is shown in Figure 8.

Figure 8 shows that, from -2 to 18 dB, the accuracy of the DCN-BiLSTM network is substantially higher than the accuracy of the other seven recognition algorithms. When the SNR exceeds 4 dB, the recognition accuracy of the DCN-BiLSTM network for the 11 modulation signals can reach 90% .

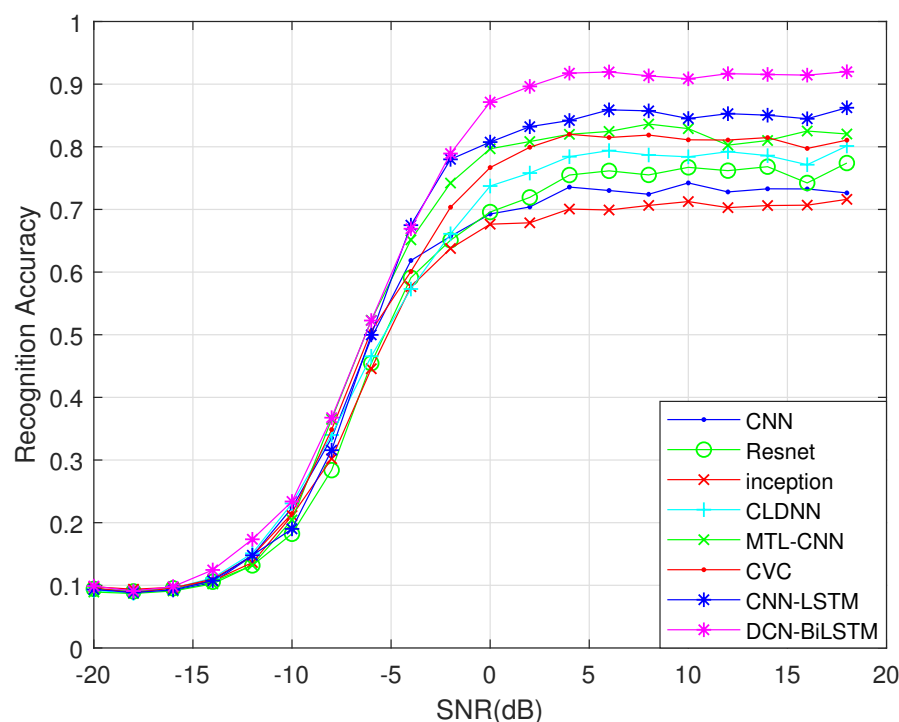


Figure 8. AMR rates of different networks under different signal-to-noise ratios.

Figure 9 shows the values of the cross-entropy loss function during DCN-BiLSTM network model training and testing. As the number of iterations increases, the training and test loss values continue to decrease, indicating that the real label and the predicted label are constantly converging. When the number of iterations reaches 23, the training and testing loss values no longer decrease, and the model obtained at this time is the best.

Figure 10 shows confusion matrices for the CNN, DCN and DCN-BiLSTM models when the SNR is 18 dB. According to Figure 10a, it is difficult for the CNN to distinguish

QPSK (Quadrature Phase Shift Keying) and 8PSK (8 Phase Shift Keying) because the CNN does not fully extract the phase information: the amplitudes of QPSK and 8PSK are the same and the difference between them is the phase. In addition, 16QAM (16 Quadrature Amplitude Modulation) is often misrecognized as 64QAM (64 Quadrature Amplitude Modulation) because the constellation points of 16QAM can be found in the constellation points of 64QAM.

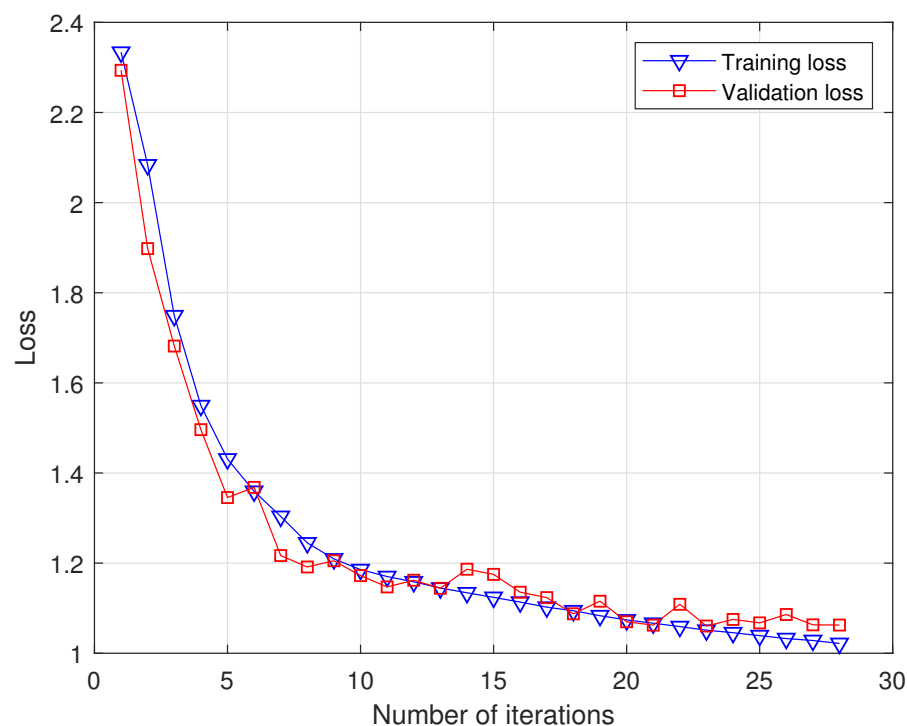


Figure 9. Loss value of DCN-BiLSTM algorithm.

To fully consider the phase information, we replaced the CNN with a DCN. The accuracy confusion matrix is shown in Figure 10b. Clearly, the accuracies on the QPSK and 8PSK modulation types are much better than in Figure 10a, and the accuracy on the 16QAM and 64QAM modulation types has improved as well. However, the accuracy on the 16QAM and 64QAM types is still not good enough for practical applications; therefore, we still need to improve their recognition accuracy.

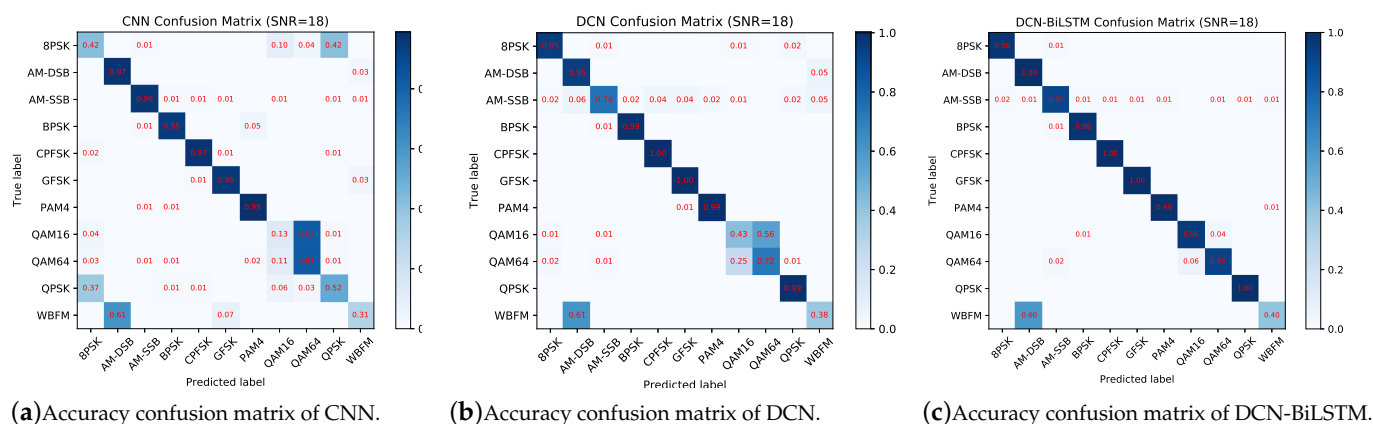


Figure 10. Accuracy confusion matrices of CNN, DCN and DCN-BiLSTM algorithms when SNR = 18 dB.

Considering the connections among data points, we cascaded the BiLSTM after the DCN to extract the contextual information of signals. An accuracy confusion matrix for the DCN-BiLSTM is shown in Figure 10c, showing that the recognition accuracy of 16QAM

and 64QAM is greatly improved compared with the results in Figure 10a,b. This result indicates that the BiLSTM is useful for extracting the contextual features of signals.

As shown in Figure 10a–c, it is quite difficult to recognize wide band frequency modulation (WBFM). The reason is that the dataset uses voice signals to generate analog signals, and people’s voices have silent periods during speaking, leaving only a single carrier during the silent period. Thus, the WBFM signals can easily be misclassified as AM-DSB (amplitude modulation—double side band modulation) signals.

To fully compare the dataset recognition capabilities of the above several networks, we used an online platform that can perform statistical analysis [30]. First, we uploaded the file representing the recognition result in csv format, as shown in Table 3. Table 3 shows the recognition error rate of five types of datasets. The error rates of the first four datasets correspond to the corresponding modulated signals in the fifth dataset. According to Rodríguez-Fdez et al. [30] and the test situation in this paper, we selected Friedman [31] as the test type to be applied. We chose Holm [32], which is widely used, as the post-hoc with control method. At the same time, we set the significance level α to 0.05.

After experiments, the algorithm rankings obtained are shown in Table 4. As we can see, DCN-BiLSTM has the highest ranking and its performance is better than the other seven networks. Moreover, the ranking in Table 4 is consistent with the ranking of the recognition effect in Figure 8, which also shows the correctness of the experiment.

Table 5 summarizes the comparison between DCN-BiLSTM and the other seven algorithms by using post-hoc with control methods. By comparing p-value with α , it can be seen that the DCN-BiLSTM network is significantly different from Inception, CNN and Resnet, indicating that the proposed network has significant progress compared with them. At the same time, there are no significant differences between DCN-BiLSTM and CLDNN, CVC, CNN-LSTM and MTL-CNN, which means that the proposed algorithm inherits the excellent performance of the four networks and can replace them in the field of modulation recognition.

Table 3. The recognition error rate of different networks for different datasets.

Datasets	DCN-BiLSTM	CNN	Resnet	Inception	CLDNN	MTL-CNN	CVC	CNN-LSTM
8PSK	0.442	0.632	0.514	0.574	0.523	0.483	0.461	0.498
16QAM	0.422	0.876	0.758	0.794	0.773	0.621	0.826	0.78
64QAM	0.32	0.381	0.517	0.553	0.474	0.614	0.425	0.506
QPSK	0.429	0.783	0.698	0.722	0.601	0.518	0.583	0.499
RML2016.10a	0.379	0.492	0.476	0.498	0.458	0.445	0.453	0.445

Table 4. Algorithm rankings.

Rank	Algorithm
1.00000	DCN-BiLSTM
3.70000	CNN-LSTM
3.70000	MTL-CNN
4.00000	CVC
4.80000	CLDNN
5.20000	Resnet
6.60000	CNN
7.00000	Inception

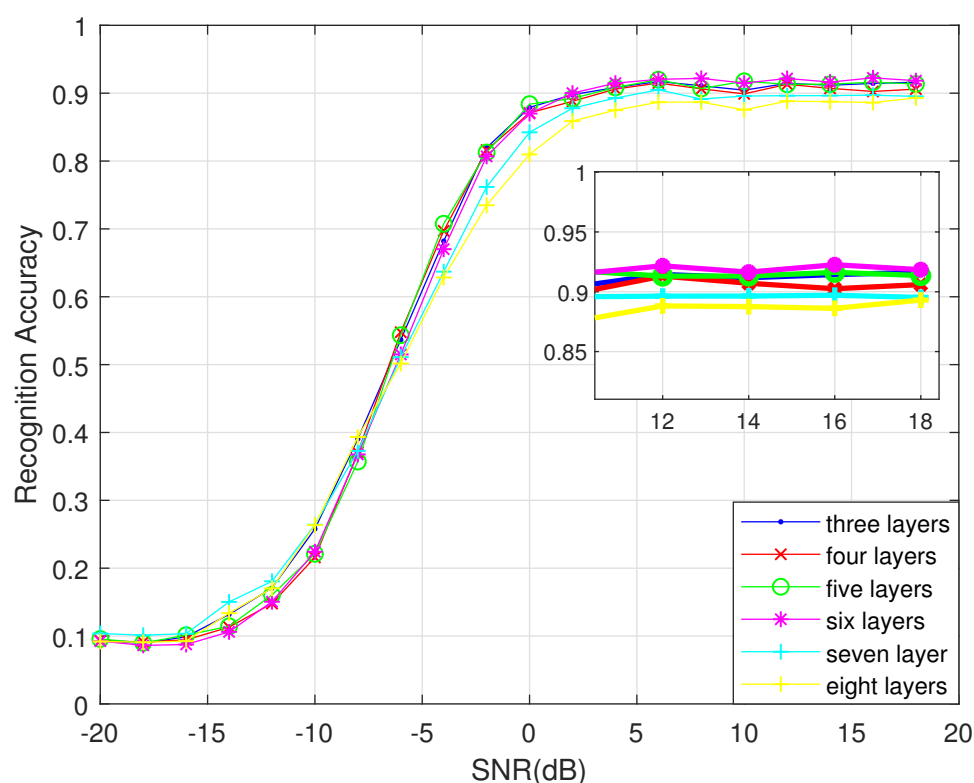
Table 5. The comparison of DCN-BiLSTM with other algorithms using the post-hoc with control methods.

Comparison	Statistic	Adjusted <i>p</i> -Value	Result
DCN-BiLSTM vs Inception	3.87298	0.00075	H0 is rejected
DCN-BiLSTM vs CNN	3.61478	0.00180	H0 is rejected
DCN-BiLSTM vs Resnet	2.71109	0.03353	H0 is rejected
DCN-BiLSTM vs CLDNN	2.45289	0.05669	H0 is accepted
DCN-BiLSTM vs CVC	1.93649	0.15842	H0 is accepted
DCN-BiLSTM vs CNN-LSTM	1.74284	0.16272	H0 is accepted
DCN-BiLSTM vs MTL-CNN	1.74284	0.16272	H0 is accepted

4.2. Performance Comparison When Using Different Parameters for the DCN-BiLSTM Network

To obtain the best parameter configuration for the DCN-BiLSTM network, this section studies the influences of each parameter configuration on the algorithm's performance.

First, we change only the number of DCN layers to find the best number of DCN layers. The different recognition results are shown in Figure 11.

**Figure 11.** Algorithm recognition accuracy under different DCN layers.

As shown in Figure 11, the overall recognition rate is the highest with six DCN layers when the SNR is greater than 2 dB. With fewer than six layers, the network's ability to extract phase features is not strong enough. With more than six layers, the network extracts redundant features, and the recognition rate no longer improves, which wastes memory. Therefore, the best number of DCN layers for this algorithm is six.

Changing the BiLSTM layer number also affects the recognition performance. Figure 12 shows a comparison of the recognition performance under different numbers of BiLSTM layers.

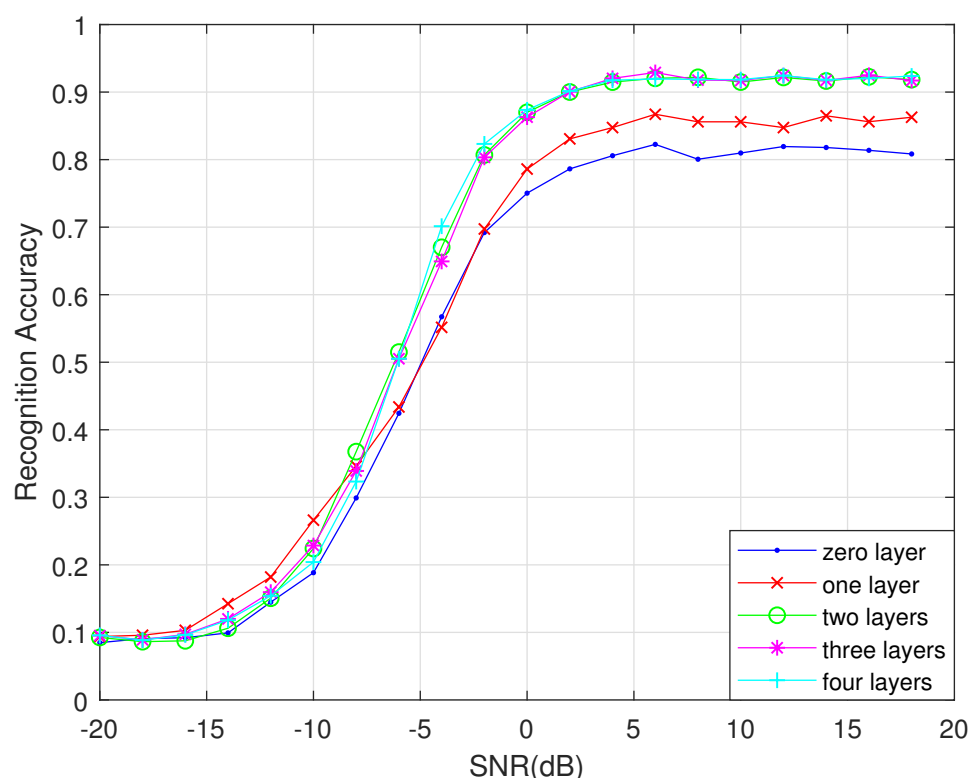


Figure 12. Algorithm recognition accuracy under different BiLSTM layers.

With fewer than two BiLSTM layers, the algorithm's ability to process feature information is poor. With more than two layers, while the accuracy rate is equivalent to that of a two-layer BiLSTM network, the added layers cause a speed reduction and waste memory. Therefore, it is best to set the number of BiLSTM layers to two.

Based on the previous analysis, the final DCN-BiLSTM network parameters are shown in Table 6.

Table 6. The DCN-BiLSTM network parameters.

Layer	Activation Function	Output Dimensions
Input	/	(128, 2)
DCN layer L1	ReLU	(None, 64, 32)
DCN layer L2	ReLU	(None, 64, 128)
DCN layer L3	ReLU	(None, 64, 128)
DCN layer L4	ReLU	(None, 64, 128)
DCN layer L5	ReLU	(None, 64, 128)
DCN layer L6	ReLU	(None, 64, 256)
BiLSTM layer B1	/	(None, 64, 512)
BiLSTM layer B2	/	(None, 256)
Dense	Softmax	(None, 11)

4.3. Five-Fold Cross Validation

To evaluate the performance of the network proposed in this article, we used the five-fold cross-validation method to train and test the network in this part. We divide the dataset into five parts equally, and use one part as the test sets each time and the remaining four parts as the training sets. Finally we obtained five training and test results, as shown

in Figure 13. The average recognition rate curve of these five times is shown in Figure 14. As shown in Figures 13 and 14, when SNR is greater than 4 dB, the recognition accuracy of the proposed network is more than 90% under five trainings, covering the entire dataset, which illustrates the rationality and stability of the designed network.

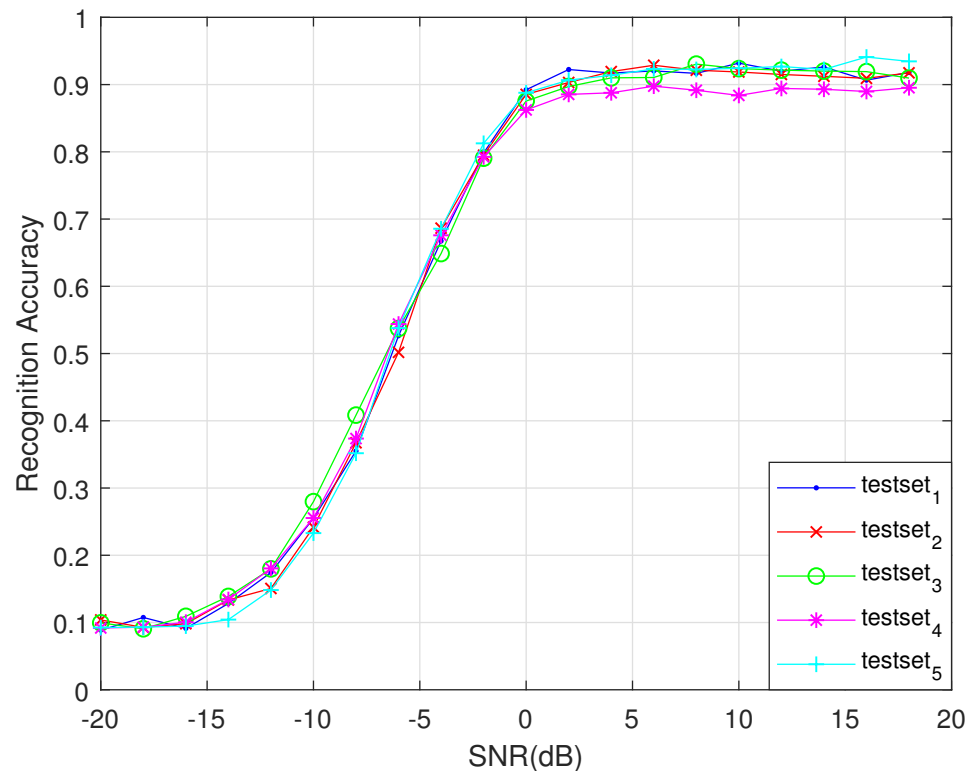


Figure 13. Recognition rate curve for five-fold cross-validation.

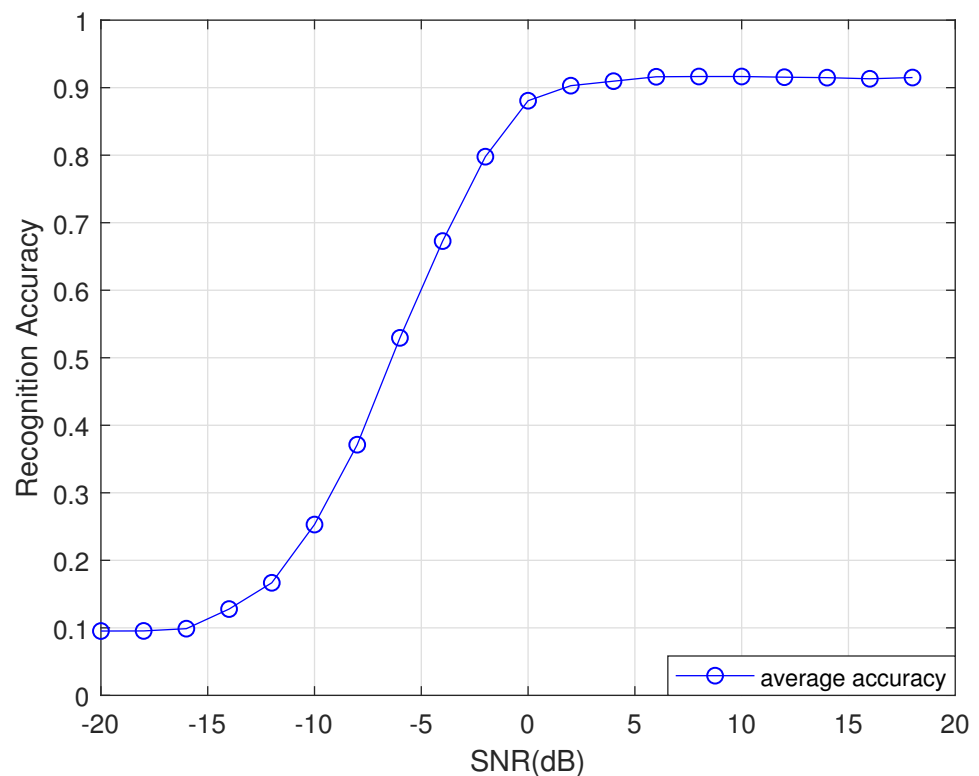


Figure 14. Average recognition rate curve of five-fold cross-validation.

5. Conclusions

In this paper, we propose a classification algorithm based on the DCN-BiLSTM network that achieves direct recognition of 11 different types of modulated signals. First, DCN layers are used to extract the phase features of the modulation signal. Then, BiLSTM layers are used to extract the contextual information and construct a bidirectional long short-term memory model for the features. Compared with previous network recognition algorithms based on CNN, Inception, Resnet, CLDNN, MTL-CNN, CVC and CNN-LSTM, the recognition accuracy of the DCN-BiLSTM network is significantly higher under high SNR. However, even when the SNR is as low as 4 dB, the recognition accuracy rate of the DCN-BiLSTM network can still reach 90%.

However, the DCN-BiLSTM network has a slow training speed, and the method works satisfactorily only for signals whose frequency offset and sampling frequency offset are within a certain range. In addition, the identified signal must belong to one of the 11 specified types. In future work, we plan to modify or optimize these issues. In particular, on the one hand, for the training speed issue, we can use other GPUs with stronger computing capabilities to speed up training. On the other hand, in terms of network structure, each LSTM unit in the BiLSTM layers contains three gate functions, resulting in more parameters, which is the main reason for the slow network training speed. Therefore, we can try to optimize the LSTM unit to reduce the number of parameters and increase the training speed, such as reducing or simplifying the gate function.

In addition to the phase information and contextual information mentioned in this paper, many other characteristics of modulated signals could be considered, such as time-frequency domain characteristics and constellation characteristics. In future work, we plan to use other features in combination with the DCN-BiLSTM network to improve the modulation signal identification performance.

Author Contributions: Conceptualization, K.L. and W.G.; Data curation, W.G.; Formal analysis, K.L.; Funding acquisition, Q.H.; Investigation, K.L. and W.G.; Methodology, K.L.; Software, W.G.; Supervision, K.L.; Validation, W.G.; Writing—original draft, W.G.; and Writing—review editing, K.L. and W.G. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by National Natural Science Foundation of China grant number 61571279.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

AMR	Automatic modulation recognition
DCN-BiLSTM	Deep complex network-bidirectional long short-term memory network
DCN	Deep complex network
BiLSTM	Bidirectional long short-term memory
CNN	Convolutional neural network
SNR	Signal-to-noise ratio
ML	Maximum likelihood
FB	Feature-based
SVMs	Support vector machines
ANNs	Artificial neural networks
DL	Deep learning
QAM	Quadrature Amplitude Modulation
STN	Spatial transformation network

KCRDP	Kernel collaborative representation and discriminative projection
Resnet	Residual network
CLDNN	Long short-term memory deep neural network
MTL-CNN	CNN that uses a multitask learning scheme
CM	Correction module
DNN	Deep neural network
CVCs	Complex convolutions
LSTM	Long short-term memory
ReLU	Rectified linear unit
AM-DSB	Amplitude modulation-Double sideband modulation
AM-SSB	Amplitude modulation-Single sideband modulation
WBFM	Wide band frequency modulation
BPSK	Binary Phase Shift Keying
8PSK	8 Phase Shift Keying
QPSK	Quadrature Phase Shift Keying
CPFSK	Continuous Phase Frequency Shift Keying
GFSK	Gauss Frequency Shift Keying
PAM4	Pulse Amplitude Modulation 4
16QAM	16 Quadrature Amplitude Modulation
64QAM	64 Quadrature Amplitude Modulation
GPU	Graphic Processing Unit

References

- O'Shea, T.J.; Roy, T.; Clancy, T.C. Over-the-air deep learning based radio signal classification. *IEEE J. Sel. Top. Signal Process.* **2018**, *12*, 168–179. [\[CrossRef\]](#)
- Zeng, Y.; Zhang, M.; Han, F.; Gong, Y.; Zhang, J. Spectrum analysis and convolutional neural network for automatic modulation recognition. *IEEE Wirel. Commun. Lett.* **2019**, *8*, 929–932. [\[CrossRef\]](#)
- Xu, J.L.; Su, W.; Zhou, M. Likelihood-ratio approaches to automatic modulation classification. *IEEE Trans. Syst. Man Cybern. Part C* **2011**, *41*, 455–469. [\[CrossRef\]](#)
- Wu, Z.; Zhou, S.; Yin, Z.; Ma, B.; Yang, Z. Robust automatic modulation classification under varying noise conditions. *IEEE Access* **2017**, *5*, 733–741. [\[CrossRef\]](#)
- Wu, H.; Saquib, M.; Yun, Z. Novel automatic modulation classification using cumulant features for communications via multipath channels. *IEEE Trans. Wirel. Commun.* **2008**, *7*, 3098–3105.
- Yan, X.; Liu, G.; Wu, H.; Feng, G. New automatic modulation classifier using cyclic-spectrum graphs with optimal training features. *IEEE Commun. Lett.* **2018**, *22*, 1204–1207. [\[CrossRef\]](#)
- Ta, N.P. A wavelet packet approach to radio signal modulation classification. *Proc. ICCS* **1994**, *1*, 210–214.
- Park, C.; Choi, J.; Nah, S.; Jang, W.; Kim, D.Y. Automatic modulation recognition of digital signals using wavelet features and svm. *Int. Conf. Adv. Commun. Technol.* **2008**, *1*, 387–390.
- Zhao, Y.; Ren, G.; Wan, X.; Wu, Z.; Gu, X. Automatic digital modulation recognition using artificial neural networks. *Int. Conf. Neural Netw. Signal Process.* **2003**, *1*, 257–260.
- Machhour, A.; Mallahi, M.E.; Zouhri, A.; Chenouni, D. Image classification using shifted legendre-fourier moments and deep learning. In Proceedings of the 2019 7th Mediterranean Congress of Telecommunications (CMT), Fes, Morocco, 24–25 October 2019; pp. 1–6.
- Lakkhanawannakun, P.; Noyunsan, C. Speech recognition using deep learning. In Proceedings of the 2019 34th International Technical Conference on Circuits/Systems, Computers and Communications (ITC-CSCC), Jeju, Korea, 23–26 June 2019; pp. 1–4.
- Li, M.; Li, O.; Zhang, C. An automatic modulation recognition method with low parameter estimation dependence based on spatial transformer networks. *Appl. Sci.* **2019**, *9*, 5. [\[CrossRef\]](#)
- Lee, J.H.; Kim, J.; Kim, B.; Yoon, D.; Choi, J.W. Robust automatic modulation classification technique for fading channels via deep neural network. *Entropy* **2017**, *19*, 9.
- Wang, Y.; Liu, M.; Yang, J.; Gui, G. Data-driven deep learning for automatic modulation recognition in cognitive radios. *IEEE Trans. Veh. Technol.* **2019**, *68*, 4074–4077. [\[CrossRef\]](#)
- Li, D.; Yang, R.; Li, X.; Zhu, S. Radar signal modulation recognition based on deep joint learning. *IEEE Access* **2020**, *8*, 515–528. [\[CrossRef\]](#)
- O'Shea, T.; West, N. Radio machine learning dataset generation with gnu radio. *Proc. GNU Radio Conf.* **2016**, *1*, 1–6.
- O'Shea, T.J.; Corgan, J.; Clancy, T.C. Convolutional Radio Modulation Recognition Networks. In *International Conference on Engineering Applications of Neural Networks*; Springer: Berlin/Heidelberg, Germany, 2016; pp. 213–226.

18. West, N.E.; O'Shea, T.J. Deep architectures for modulation recognition. In Proceedings of the 2017 IEEE International Symposium on Dynamic Spectrum Access Networks (DySPAN), Baltimore, MD, USA, 6–9 March 2017.
19. Zhang, M.; Zeng, Y.; Han, Z.; Gong, Y. Automatic modulation recognition using deep learning architectures. In Proceedings of the 2018 IEEE 19th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC), Kalamata, Greece, 25–28 June 2018; pp. 1–5.
20. Mossad, O.S.; ElNainay, M.; Torki, M. Deep convolutional neural network with multi-task learning scheme for modulations recognition. In Proceedings of the 2019 15th International Wireless Communications Mobile Computing Conference (IWCMC), Tangier, Morocco, 24–28 June 2019; pp. 1644–1649.
21. Yashashwi, K.; Sethi, A.; Chaporkar, P. A learnable distortion correction module for modulation recognition. *IEEE Wirel. Commun. Lett.* **2019**, *8*, 77–80. [\[CrossRef\]](#)
22. Liu, X.; Wang, Q.; Wang, H. A two-fold group lasso based lightweight deep neural network for automatic modulation classification. In Proceedings of the 2020 IEEE International Conference on Communications Workshops (ICC Workshops), Dublin, Ireland, 7–11 June 2020; pp. 1–6.
23. Krzyston, J.; Bhattacharjee, R.; Stark, A. Complex-valued convolutions for modulation recognition using deep learning. In Proceedings of the 2020 IEEE International Conference on Communications Workshops (ICC Workshops), Kansas City, MO, USA, 20–24 May 2020; pp. 1–6.
24. Wu, Y.; Li, X.; Fang, J. A deep learning approach for modulation recognition via exploiting temporal correlations. In Proceedings of the 2018 IEEE 19th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC), Kalamata, Greece, 25–28 June 2018; pp. 1–5.
25. Trabelsi, C.; Bilaniuk, O.; Serdyuk, D.; Subramanian, S.; Santos, J.F.; Mehri, S.; Rostamzadeh, N.; Bengio, Y.; Pal, C.J. Deep complex networks. *arXiv* **2017**, arXiv:1705.09792.
26. Huang, Z.; Xu, W.; Yu, K. Bidirectional lstm-crf models for sequence tagging. *arXiv* **2015**, arXiv:1508.01991.
27. Hajiaghayi, M.; Vahedi, E. Code failure prediction and pattern extraction using lstm networks. In Proceedings of the 2019 IEEE Fifth International Conference on Big Data Computing Service and Applications (BigDataService), Newark, CA, USA, 4–9 April 2019; pp. 55–62.
28. Ma, W.; Yu, H.; Zhao, K.; Zhao, D.; Yang, J.; Ma, J. Tibetan location name recognition based on bilstm-crf model. In Proceedings of the 2019 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC), Guilin, China, 17–19 October 2019; pp. 412–416.
29. Dyer, C.; Ballesteros, M.; Ling, W.; Matthews, A.; Smith, N.A. Transition-based dependency parsing with stack long short-term memory. *arXiv* **2015**, arXiv:1505.08075.
30. Rodríguez-Fdez, I.; Canosa, A.; Mucientes, M.; Bugarín, A. STAC: A web platform for the comparison of algorithms using statistical tests. In Proceedings of the 2015 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), Istanbul, Turkey, 2–5 August 2015.
31. Friedman, M. The use of ranks to avoid the assumption of normality implicit in the analysis of variance. *J. Am. Stat. Assoc.* **1937**, *32*, 675–701. [\[CrossRef\]](#)
32. Holm, S. A simple sequentially rejective multiple test procedure. *Scand. J. Stat.* **1979**, *6*, 65–70.