

Received February 12, 2020, accepted February 28, 2020, date of publication April 21, 2020, date of current version May 28, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2979768

# Communication Protocol Classification Based on LSTM and DBN

WENWEN LI<sup>1</sup>, ZHENG DOU<sup>1</sup>, AND LIN QI<sup>1</sup>

College of Information and Communication Engineering, Harbin Engineering University, Harbin 150001, China

Corresponding author: Lin Qi (qilin@hrbeu.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61671167, and in part by the International Exchange Program of Harbin Engineering University for Innovation-Oriented Talents Cultivation.

**ABSTRACT** In the battlefield, we often don't know the parameters about enemy wireless communication system. Therefore, we need to use electronic reconnaissance equipment to search, intercept, identify and analyze enemy wireless communication signal. However, the exciting electronic reconnaissance methods can only detect signal layer parameters such as signal carrier frequency and bandwidth, and cannot obtain more information. In order to improve the reconnaissance ability, we propose a novel communication protocol classification algorithm based on long short-term memory (LSTM) and deep belief network (DBN). We first introduce the DBN, then simulates communication protocol classification method based on DBN. In order to improve the performance, we optimize the method. We uses the LSTM to pre-process the data to extract the feature firstly. Then we make the feature as the input of DBN to classify the communication protocol. Finally we make simulation to verify the effectiveness of the proposed algorithm. Simulation results show that the proposed algorithm has very good performance to classify the communication protocol.

**INDEX TERMS** LSTM, DBN, wireless channel, protocol classification.

## I. INTRODUCTION

Nowadays, more and more countries have increasingly attached importance to electronic countermeasure system. The electronic countermeasure system is composed of electronic reconnaissance system, electronic jamming system, and electronic defense system [1]. Electronic reconnaissance system can obtain parameters of enemy electronic equipment, and analyze the signal to obtain more information [2]. According to the results of electronic reconnaissance system, electronic interference system can adjust the interference system parameters, interfere the enemy communication system, and block the normal communication [3]. According to the results of electronic reconnaissance system, electronic defense system can improve the anti-jamming capability by adjusting system parameters [4]. We can see that electronic reconnaissance system is the basis for analyzing intelligence, electronic interference system, and electronic defense system. The stronger the electronic reconnaissance ability, we can get more information about enemy equipment. However, the exciting electronic reconnaissance system can only detect signal layer parameters such as signal carrier

frequency and bandwidth, and cannot identify communication protocol [5]. Communication protocols are the ties to operate information networks [6], [7]. If we identify the communication protocol, we can get most of the communication information [8]. In order to improve the electronic reconnaissance ability, we need study the protocol classification algorithm.

The existing protocol classification method mainly includes method based on port number, method based on deep packet inspection, and method based on deep stream inspection [9]. The method based on port number can classify the protocol that has registered the port number in Internet assigned numbers authority (IANA) [10]. The method has the characteristics of simple calculation and low complexity, and does not require complicated processing to obtain classification results. In the early stage of Internet development, the application layer protocol is few and simple, method based on port number to classify protocol is feasible. However, more and more protocols do not register their port numbers in IANA, fewer protocols can be classified by the method based on port number [11], [12]. So the method based on port number is not suitable any more [13]. The method based on deep packet inspection uses the pattern matching algorithm to find out whether data message contains the

The associate editor coordinating the review of this manuscript and approving it for publication was Guan Gui<sup>1</sup>.

protocol feature, so as to classify the protocol. Most data message contains specific protocol feature, which is convenient to distinguish different protocols. And the feature is open and very easy to get in most cases, so the protocol can be classified by searching and detecting the protocol feature. Kazemi and Fanian [14] proposed a method for network tunneling protocols identification. This method is an improvement of deep packet inspection method. It overcomes the problem of high processing cost, low efficiency and requiring high memory and CPU resources. The simulation results show that the proposed method can identify network tunneling protocols with high accuracy and low processing cost. Chen and Liao [15] proposed an optimized algorithm for deep packet inspection. Chen proposed an optimized solution based on regular expressions. In order to reduce memory space requirement of the deep packet inspection engine, Chen used an optimized algorithm to reduce the states of a deterministic finite automaton. Meanwhile, Chen improved the matching performance of the deep packet inspection engine by adopting a hybrid packet matching pattern. The simulation results show that the cost of memory space is reduced and the matching speed is improved. In order to make the protocol identification more accuracy, we should build a good set of application protocol signatures, however, it is a very time consuming task and demands a high expertise. Feng *et al.* [16] proposed an automatic traffic signature generation method based on Smith-waterman algorithm to solve above problem. The simulation results show that the proposed method is a very accurate and quick approach compared to the approach that uses preceding application protocol analysis. Xia and Liu [17] proposed an optimized protocol identification method for high-speed network system. Xia makes a deep system-wide profile and analyze the major hotspots of a typical network system on an Intel X86-64 platform, including a complete TCP/IP stack and a protocol identification engine by deep packet inspection. The simulation results show that the proposed algorithm eliminates the memory bottleneck and computing bottleneck. The protocol classification method based on deep packet detection has high accuracy, but time complexity is high. It is not feasible for the system with high real-time requirement. The method based on deep flow detection uses data flow feature to classify different protocols. When we use different protocols to transmit data, the data flow feature will also be different. Yang *et al.* [18] proposed a P2P network traffic classification method using support vector machine(SVM). The proposed method extracts the traffic feature of Bittorrent, PPLive, Skype and MSN. Yang also introduced the classification framework based on the SVM. The simulation results show that the classification accuracy can reach 91%. Este *et al.* [19] proposed a network traffic classification method based on SVM. The proposed method classifies network traffic according to the size of packets and the direction of data flow. The proposed method classifies correctly only with a few hundred training samples. The simulation results show that the classifiers can be very effective at discriminating network traffic

generated by different applications, and the classification accuracy of Bittorrent can reach 96.8%. Moore and Zuev [20] proposed an optimized network traffic classification method based on Bayesian analysis. The proposed method extracts 249 features of 10 different network traffic. The simulation results show that the method has high classification accuracy and the classification accuracy of Web can reach 96.8%. Because data flow has a large number of features, it is difficult to select the appropriate features to classify protocol.

The existing protocol classification methods are all for Internet protocol, these methods are not suitable for wireless communication protocol classification. The existing communication protocol classification methods first identify the modulation scheme, frequency and other signal parameters, then vectorize the parameters according to the entropy, and finally compare the parameter set with the protocol parameters or use the machine learning method to classify the protocol. However, this method depends on the identification of various signal parameters. When the identification accuracy of the signal parameters is high, the protocol classification can get high accuracy. When the identification accuracy of one or more signal parameters decreases, the protocol classification accuracy decreases seriously due to the error accumulation. In order to improve the communication protocol classification accuracy, we propose a novel communication protocol classification algorithm based on LSTM and DBN.

The remaining parts of this paper are organized as follows. In Section 2, we introduce DBN and LSTM. In Section 3, we propose a protocol classification algorithm based on DBN and make a lot of simulation experiments. In Section 4, we propose a protocol classification algorithm based on DBN and LSTM, and make a lot of simulation experiments. Conclusions are drawn in Section 5.

## II. BACKGROUND KNOWLEDGE

### A. DBN

Because the network structure of Boltzmann machine (BM) is too complex, it is impossible to realize such a complex network in reality [21]. In 1986, Professor Smolensky created restricted Boltzmann machine (RBM) [22]. RBM is a two-layer undirected graphical model, which is composed of visible layer and hidden layer. They are connected by "full connection" [23]. RBM simulates the probability distribution of visible layer nodes by random features [24]. Different from BM, the connection between nodes in RBM needs to satisfy certain limitation [25]. Nodes in the same layer are not allowed to connect, and nodes in different layers must be fully connected [26]. The input and output of standard RBM only have two states of "0" and "1". The "0" represents that the node is inactive and the "1" represents that the node is active [27]. The difference between RBM and BM is that BM allows nodes at the same level to connect. Because of this limitation, RBM has a key feature: its random hidden nodes are conditionally independent under given observation data, and vice versa [28].

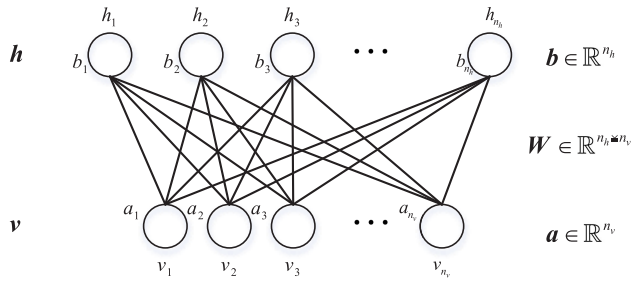


FIGURE 1. RBM structure.

Fig.1 shows the schematic diagram of the RBM structure.  $n_v$  and  $n_h$  represent the number of neurons in the visible and hidden layers respectively, and the subscript  $v, h$  represents the visible layer and the hidden layer respectively.  $v = (v_1, v_2, \dots, v_{n_v})^T$  represents the state of the  $i$ th neuron in the visible layer.  $h = (h_1, h_2, \dots, h_{n_h})^T$  represents the state vector of the hidden layer and  $h_j$  represents the state of the  $j$ th neuron in the hidden layer.  $a = (a_1, a_2, \dots, a_{n_v})^T \in \mathfrak{R}^{n_v}$  represents the offset vector of the visible layer and  $a_i$  represents the offset vector of the  $i$ th neuron in the visible layer.  $b = (b_1, b_2, \dots, b_{n_h})^T \in \mathfrak{R}^{n_h}$  represents the offset vector of the hidden layer and  $b_j$  represents the offset vector of the  $j$ th neuron in the hidden layer.  $W = (w_{i,j}) \in \mathfrak{R}^{n_h \times n_v}$  represents the weight matrix between hidden and visible layer.  $w_{i,j}$  represents the connection weight between the  $i$ th neuron in the hidden layer and the  $j$ th neuron in the visible layer.

DBN is trained greedily by stacking multiple RBMs. A typical DBN structure is shown in Fig.2 [29]. Similar to the RBM structure, nodes between the same layers are not allowed to connect, and nodes between layers are connected by “full connection”. DBN can learn the structure of input data, and extract the features of input data step by step through multi-layer RBM, and use the features to classify signal, image and so on [30].

DBN is a model for learning to extract deep data feature. DBN can simulate the joint probability of the visible layer vector  $x$  and the  $l$  hidden layer. The joint probability is defined as follows:

$$P(x, h^1, \dots, h^l) = \left( \prod_{k=0}^{l-2} P(h^k | h^{k+1}) \right) P(h^{l-1} | h^l) \quad (1)$$

where  $x = h^0$ ,  $P(h^k | h^{k+1})$  is the conditional probability of the visible layer and hidden layer of the  $k$ -layer RBM.  $P(h^{l-1} | h^l)$  is the conditional probability of the visible layer and hidden layer of the top RBM.

In order to classify communication protocol, classifier is usually added to the last layer. Common classifiers include SVM, logistic regression model and softmax regression model. However, SVM and logistic regression model are only suitable for binary classification. So we use softmax regression model. Softmax regression model is an extension of logistic regression model, and their mathematical models

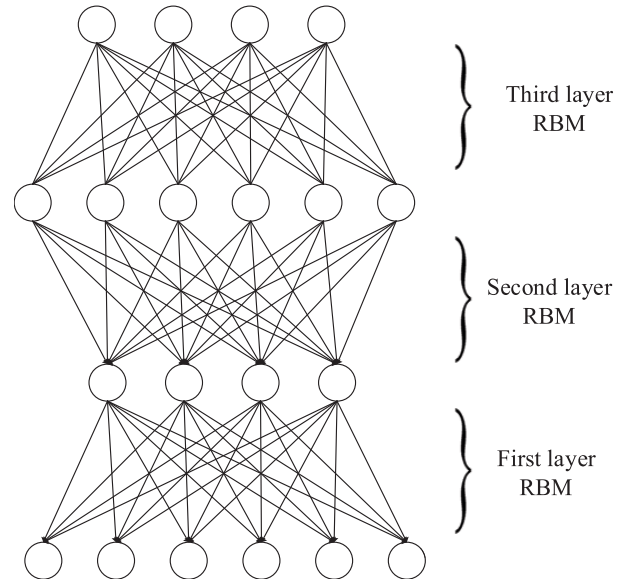


FIGURE 2. DBN structure.

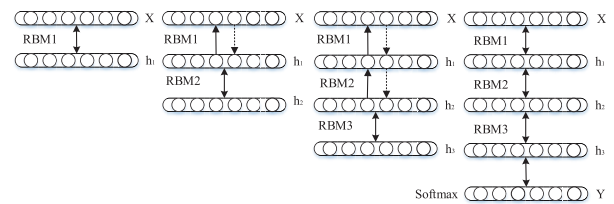


FIGURE 3. The process of training layer by layer.

are very similar. But softmax regression model can support multiple classification.

For a given structure, the training process of DBN is completed in training RBM layer by layer [31]. The process of training layer by layer is shown in Fig.3. First, receive input data and train the first layer RBM. After training, the hidden layer response of all training data are calculated. Then freeze the weight of the first layer and use the response as input to train the next layer, and repeat these steps to train all layers. All training is done in an unsupervised way. The last thing to train DBN is to treat it as a neural network and use tagged data to train all weights acquired during unsupervised training.

First, we use the data  $X$  as input to the first layer RBM. The state of the hidden layer is defined as follows:

$$h_1 = \sigma(W_1^T X + b_1) \quad (2)$$

where  $\sigma$  is sigmoid function. Then, we suppose DBN consisting of  $l$  hidden layers. DBN is initialized layer by layer with greedy algorithm. Then the state of  $i$ th hidden layer is defined as follows:

$$\begin{cases} h_i = 1 / [1 + \exp(-h_{i-1} \cdot W_i^T + b_i)] \\ h_0 = X, \quad \forall i \in \{1, 2, \dots, l\} \end{cases} \quad (3)$$

Finally, we use backward propagation algorithm to optimize the weights. In this way, we can get the global optimal

weight vector. The global optimal weight vector is defined as follows:

$$J(W, b) = \left[ \frac{1}{m} \sum_{i=1}^m \frac{\partial}{\partial W_{ij}^{(l)}} J(W, b; x^{(i)}, y^{(i)}) \right] + \lambda W_{ij}^{(l)} \quad (4)$$

$$\begin{cases} \frac{\partial}{\partial W_{ij}^{(l)}} J(W, b) = \left[ \frac{1}{m} \sum_{i=1}^m \frac{\partial}{\partial W_{ij}^{(l)}} J(W, b; x^{(i)}, y^{(i)}) \right] + \lambda W_{ij}^{(l)} \\ \frac{\partial}{\partial b_i^{(l)}} J(W, b) = \frac{1}{m} \sum_{i=1}^m \frac{\partial}{\partial b_i^{(l)}} J(W, b; x^{(i)}, y^{(i)}) \\ W_{ij}'^{(l)} = W_{ij}^{(l)} - \alpha \frac{\partial}{\partial W_{ij}^{(l)}} J(W, b) \\ b_i'^{(l)} = b_i^{(l)} - \alpha \frac{\partial}{\partial b_i^{(l)}} J(W, b) \end{cases} \quad (5)$$

where  $J(W, b)$  is cost function,  $\alpha$  is the learning rate.

After extracting the features, the extracted features are as input to the softmax classifier for classification training. For training date set  $\{(x^1, y^1), \dots, (x^i, y^i), \dots, (x^m, y^m)\}$ , probability of data  $x$  into category  $j$  in softmax regression model is defined as follows:

$$p(y^{(i)} = j | x^{(i)}; \theta) = \frac{e^{\theta_j^T x^{(i)}}}{\sum_{l=1}^k e^{\theta_l^T x^{(i)}}}, \quad j=1, 2, \dots, k \quad (6)$$

where  $\sum_{l=1}^k e^{\theta_l^T x^{(i)}}$  is to normalize the probability. It can ensure that the sum of the probability that all test samples belong to category  $j$  is 1.

We train the the model parameters  $\theta$  by minimizing the cost function. The cost function is defined as follows:

$$J = -\frac{1}{m} \left[ \sum_{i=1}^m \sum_{j=1}^k \{y^{(i)} = j\} \ln \frac{e^{\theta_j^T x^{(i)}}}{\sum_{l=1}^k e^{\theta_l^T x^{(i)}}} \right] + \frac{\lambda}{2} \sum_{i=1}^k \sum_j \theta_{ij}^2 \quad (7)$$

If the output  $j$  is label  $y^i$ ,  $\{y^i = j\} = 1$ . If the output  $j$  is not label  $y^i$ ,  $\{y^i = j\} = 0$ . Weight attenuation is used to prevent network over-fitting.

**B. LSTM**

Recurrent neural network (RNN) is the network for processing sequence data. RNN is different from other networks obviously [32]. In RNN, the feedback of the hidden layer not only enters the output, but also enters the hidden layer of the next time step, thus affecting the weights of the next time step. The structure of RNN is shown in Fig.4 [33]. In order to derivate and describe conveniently, we make the left structure simplify the right structure.

RNN is different from other multi-layer neural network in that it has the concept of time sequence. The next time step will be affected by current time step. We can better introduce RNN by expanding the network in time sequence [34]. We expand the network through multiple time steps, and

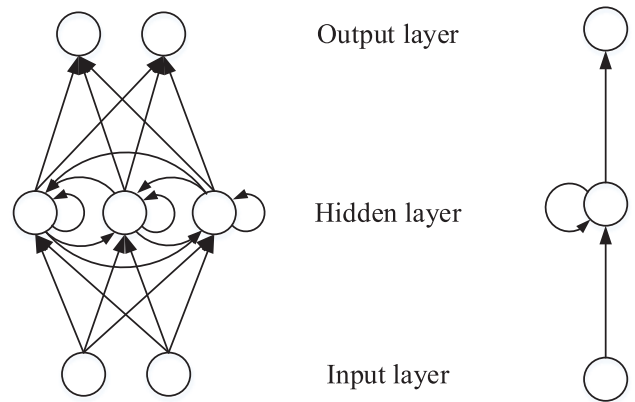


FIGURE 4. The structure of RNN.

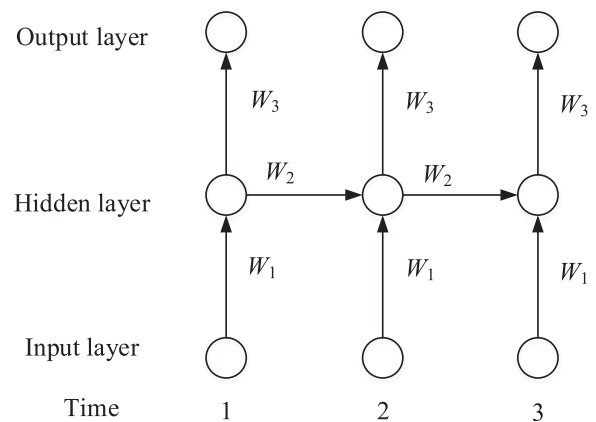


FIGURE 5. The expand structure of RNN.

visualize the connection in the form of acyclic. The expand structure of RNN is shown in Fig.5 [35].

The biggest advantage of RNN is that it introduces the time sequence into neural network, which can make current time step data have a direct impact on the next time step data. We can set the corresponding input and output layer number according to the time step. The hidden layer number is same as that time nodes and the hidden layer is recursive feedback. The first data is as input to the first layer, and then affects the second layer. The second data is as input to the second layer and then affects the third layer. The input impact is from left to right, and the final output feedback is right to left to adjust the weight [36]. The weight adjustment is based on the loss function between the output data and the original data. In order to minimize the loss function, we calculate the multivariate variables partial differential. In the calculation of partial differential, we start from the last layer, and then the weight adjustment of the penultimate layer depends on the last layer, and the rest layers are also like this. Derivative may be zero in the last few layers, which is also called gradient explosion. Therefore, the weights of the previous layers are not adjusted [37]. It is impossible to achieve the complete convergence in each training, only to adjust the last several layers of weights. Because the front layer is not adjusted, the loss is not completely eliminated [38].

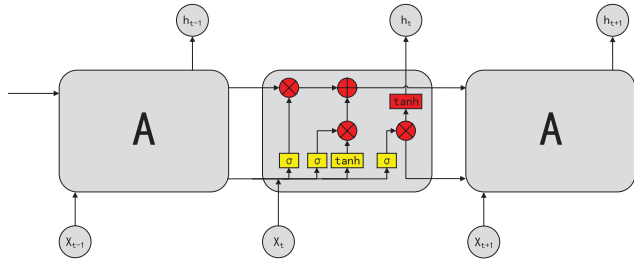


FIGURE 6. The structure of LSTM.

We can also set the learning rate  $\alpha$  which changes with the loss function to prevent the feedback from falling too fast with the gradient and missing the minimum convergence value, or at the same time, we can set the momentum impulse which changes with the  $\alpha$  reverse to prevent falling into the local optimal solution. But these methods can not solve the above problems effectively. This leads to LSTM neural network to solve the above problems [39].

LSTM was proposed by Hochreiter and Schmidhuber in 1997 [40]. LSTM model can overcome the shortcomings of RNN. LSTM consists of recursively connected subnets, which we can call memory cell. We can think of the memory cell as memory in computer. Each memory cell contains one or more self-connected memories and three multipliers. These three multipliers are input gate, output gate and forget gate, which provide read, write and reset operations [41]. As shown in Fig.6, LSTM is not a single layer neural network, but a multilayer structure, which interacts in a very special way. In Fig.6, the yellow module represents the learning neural network layer, and the red module represents the point by point operation, such as vector addition [42].

The key of LSTM is the red part state in Fig.7. It conveys information like conveyor belt, which ensures transfer information not changed. The most important component of LSTM is the gate, which can add or delete information [43]. Gate allows information to pass selectively. As shown in Fig.7, LSTM has three gates that work together. Activation function takes memory state of network as input. We set the specific threshold. If the output is greater than the threshold, we multiply the output of gate and the current layer result, and make the multiplication result as the next layer input. If the output is less than the threshold, the output will be forgot [44].

Fig.8 shows the structure of LSTM memory cell. The structure of LSTM is almost the same as the RNN, but the addition unit of RNN hidden layer is replaced by memory cell. The memory cell of LSTM can store and access information for a long time, so LSTM can solve the problem of gradient disappearance. Memory cell consists of one unit and three gates, that is input gate, output gate and forget gate [45]. The three gates symbolize the information gate. They control the information transmission of neuron and how to distribute information to the current neuron and the next neuron. These three gates have activation functions and control the activation functions by multiplication [46]. The

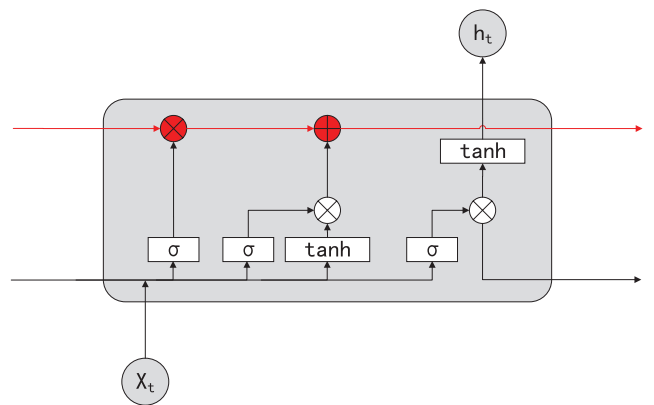


FIGURE 7. The structure of LSTM hidden layer.

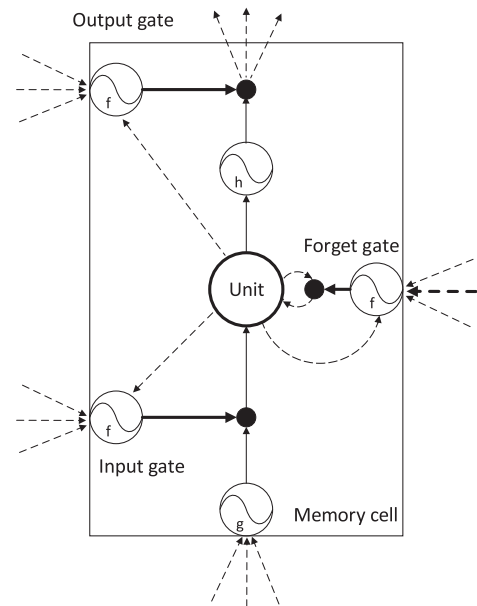


FIGURE 8. The structure of LSTM memory cell.

black dots in Fig.8 represent multiplication.  $g$  is the input function.  $h$  is the output function.  $f$  is the activation function. We usually use sigmoid function or tanh function as activation function. The equations are as follows:

$$\text{sigmoid}(x) = \frac{1}{1 + e^{-x}} \quad (8)$$

$$\text{tanh}(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (9)$$

We assume that the input of LSTM at time moment  $t$  includes input layer  $x_t$ , hidden layer  $h_{t-1}$  which is calculated by the previous unit and memory cell  $c_{t-1}$ . The output of LSTM at time moment  $t$  includes hidden layer  $h_t$  which is calculated by the current unit and memory cell  $c_t$ . The output of LSTM is calculated as follows:

A, First, we calculate the three gates at time moment  $t$ . Here we do not consider the memory cell  $c_{t-1}$ . The detail equations



are as follows:

$$i_t = \sigma(W_{x_i}x_t + W_{h_i}h_{t-1} + b_i) \quad (10)$$

$$o_t = \sigma(W_{x_o}x_t + W_{h_o}h_{t-1} + b_o) \quad (11)$$

$$f_t = \sigma(W_{x_f}x_t + W_{h_f}h_{t-1} + b_f) \quad (12)$$

where  $i_t$  is input gate at time moment  $t$ ,  $o_t$  is output gate at time moment  $t$ ,  $f_t$  is forget gate at time moment  $t$ .

B, Then, we calculate the memory cell  $c_t$ . The detail equations are as follows:

$$\bar{c}_t = \tanh(W_{x_c}x_t + W_{h_c}h_{t-1} + b_c) \quad (13)$$

$$c_t = f_t \cdot c_{t-1} + i_t \cdot \bar{c}_t \quad (14)$$

where  $\bar{c}_t$  is the memory cell without forget gate  $f_t$ .

C, Finally, we calculate the hidden layer  $h_t$  at time moment  $t$ . The detail equation is as follows:

$$h_t = o_t \cdot \tanh(c_t) \quad (15)$$

It can be seen from the above equation that the function of the input gate is to multiply the memory cell without the forget gate, and control the input to enter the memory cell. The function of forget gate is to multiply the memory cell of the last moment and control the attenuation in the memory cell. The function of the output gate is to multiply the memory cell at the current memory to get the output of the hidden layer, control the output of the memory cell to the hidden layer, and further affect the results of each gate at the next moment. LSTM solves the problem of gradient disappearance. LSTM can save the most meaningful information in the memory cell, overcome RNN's difficulty in saving long-distance information.

### III. COMMUNICATION PROTOCOL CLASSIFICATION BASED ON DBN

#### A. DATA SET

With the development of wireless communication, wireless communication protocol has also made great progress. According to the different types of communication, wireless communication network can be divided into wireless wide area network (WWAN), wireless metropolitan area network (WMAN), wireless local area network (WLAN), wireless personal area network (WPAN), low rate wireless personal area network (LR-WPAN) and so on. Wireless communication protocol is the last link to wireless communication. We need to package the information according to the protocol and transmit packaged information according to the protocol transmission parameters, so that the receiver can correctly interpret the transmitted data and get the information. ZigBee, Bluetooth and WiFi are the three most common wireless communication protocols. We takes these three protocols as examples to verify the effectiveness of the algorithm. The I/Q signal data used in this paper is collected between SNR = -5dB to SNR = 10dB. And we remove the gap between signal, only save the data which has signal. Table 1 is an example of the collected data.

TABLE 1. Examples of the collected data.

1	-0.338121055+0.367102853j
2	0.202872645+0.4121855676j
3	-0.869454434+0.44909427j
4	0.563548106+0.178721137j

TABLE 2. Data example after preprocessing.

1	2	3	4	5	6	7	8
-0.3381	0.3671	0.2029	0.4122	-0.8695	0.4491	0.5635	0.1787

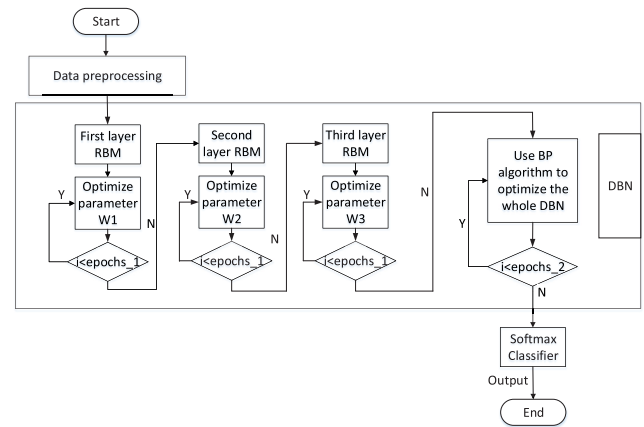


FIGURE 9. Training flow chart.

Because of the uncertainty of wireless channel, the signal data is greatly affected by SNR. And the collected data is complex number, which makes DBN unable to process collected data directly. So we need to preprocess the collected data. We use the following method to preprocess data.

- (1) Cut the data into 256 points and keep 4 decimal.
- (2) According to the real part and the imaginary part, the data processed in step 1 is divided into  $256 \times 2$  array.
- (3) Expand the data processed in step 2 to  $512 \times 1$  array.

Table 2 shows the data example after preprocessing. The training sample and testing sample number are all 48000. Each sample contains 512 data points.

#### B. MODEL FRAMEWORK

Fig.9 shows the training flow chart of communication protocol classification algorithm based on DBN. The algorithm training process detail is as follows:

- (1) After data preprocessing, the data becomes  $512 \times 1$  array.
- (2) The preprocessed data is used as input to train the first layer RBM. After training, we obtain  $512 \times 1$  eigenvector.
- (3) Keep the training parameters of the first layer RBM.  $512 \times 1$  eigenvector is used as input to train the second layer RBM. After training, we obtain  $256 \times 1$  eigenvector.
- (4) Keep the training parameters of the above two layer RBM.  $256 \times 1$  eigenvector is used as input to train the third layer RBM. After training, we obtain  $128 \times 1$  eigenvector.

TABLE 3. Detailed network parameters.

Parameters	Detail
Input nodes	512
Input nodes of first layer RBM	512
Input nodes of second layer RBM	256
Input nodes of third layer RBM	128
Number of softmax classifier output	3
Epochs_1	40
Epochs_2	100
Activation function	Relu
Optimizer	Momentum
Learning rate	0.001

(5) After training above three layer RBM, we use BP algorithm to adjust the whole network parameters. Finally, according to the output of softmax classifier, we can obtain the final classification results.

The detail network parameters are shown in Table 3.

C. EVALUATION INDEX

The algorithm classification performance can be quantified by the prediction accuracy of test sample. If the true label and the predicted label of the classifier are given by  $y_i$  and  $\hat{y}_i$  respectively, then the test error of  $m_{test}$  test samples can be defined as follows:

$$E_{test} = \frac{1}{m_{test}} \sum_{i=1}^{m_{test}} l(\hat{y}_i, y_i) \tag{16}$$

The classification accuracy can be obtained by  $1 - E_{test}$ . The algorithm classification performance can also be measured by other statistics, namely: true positive (TP), false positive (FP), false negative (FN) and true negative (TN). They are defined as follows:

- If the true label is A and the predicted label is A, then the sample is considered as TP.
- If the true label is not A and the predicted label is A, then the sample is considered as FP.
- If the true label is A and the predicted label is not A, then the sample is considered as FN.
- If the true label is not A and the predicted label is not A, then the sample is considered as TN.

Through the above statistics, we can get the performance evaluation index precision(P), recall(R) and  $F_1$  score. They are defined as follows:

$$P = \frac{TP}{TP + FN}, \quad R = \frac{TP}{TP + FP} \tag{17}$$

$$F_1 = 2 \times \frac{P \times R}{P + R} \tag{18}$$

P, R and  $F_1$  score are performance evaluation indicators for each category. We use the average of equal weights ( $P_{avg}$ ,  $R_{avg}$  and  $F_{1_{avg}}$ ) to quantify the overall performance. The closer  $P_{avg}$ ,  $R_{avg}$  and  $F_{1_{avg}}$  is to 1, the algorithm has better performance.

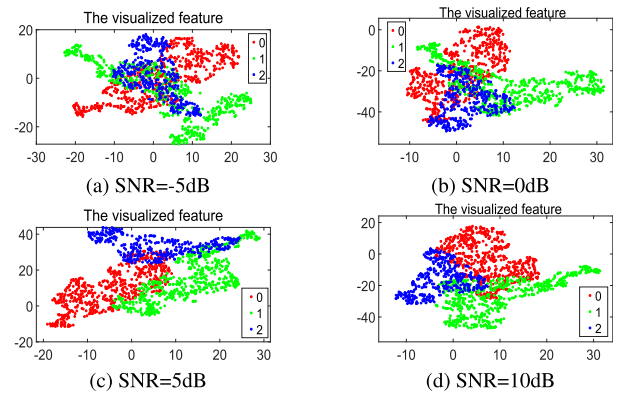


FIGURE 10. The visualized feature.

TABLE 4. Evaluation indicators under different SNR.

SNR	-5dB			0dB		
	$P_{avg}$	$R_{avg}$	$F_{1_{avg}}$	$P_{avg}$	$R_{avg}$	$F_{1_{avg}}$
	0.525	0.618	0.567	0.631	0.738	0.680
SNR	5dB			10dB		
	$P_{avg}$	$R_{avg}$	$F_{1_{avg}}$	$P_{avg}$	$R_{avg}$	$F_{1_{avg}}$
	0.693	0.733	0.712	0.787	0.799	0.793

D. SIMULATION RESULT AND ANALYSIS

In this section, we make a lot of experiments to verify the effectiveness and performance of the proposed algorithm. We use t-SNE algorithm to visualize the feature. Fig.10 shows the visualized feature under different SNR. The red number “0” represents ZigBee. The blue number “1” represents Bluetooth. The green number “2” represents Wifi.

X axis and y axis don’t have real meaning. We focus on the discrimination between extracted features. From Fig.10, we can see that the features extracted by DBN have different discrimination under different SNR. With the decrease of SNR, the discrimination of features gradually decreases. When SNR = 10dB, the feature discrimination is good, but a small number of features still overlap with each other. When SNR = -5dB, most of features are overlap with each other, the feature discrimination is poor. Table 4 shows the evaluation indicators under different SNR.

From Table 4, we can see that with the increase of SNR,  $P_{avg}$ ,  $R_{avg}$  and  $F_{1_{avg}}$  are also increase, the algorithm classification performance is better and better. When SNR = 10dB,  $P_{avg} = 0.787$ ,  $R_{avg} = 0.799$  and  $F_{1_{avg}} = 0.793$ . At this time, the algorithm classification performance is not bad. When SNR = -5dB,  $P_{avg} = 0.525$ ,  $R_{avg} = 0.618$  and  $F_{1_{avg}} = 0.567$ . At this time, the algorithm classification performance is poor. Fig.11 shows the each protocol classification accuracy and average classification accuracy.

From Fig.11, we can see that with the increase of SNR, classification accuracy is also increase. When SNR = 10dB, classification accuracy of ZigBee is 85%, classification accuracy of Bluetooth is 83% and classification accuracy of Wifi is 81%. When SNR = 6dB, the average classification

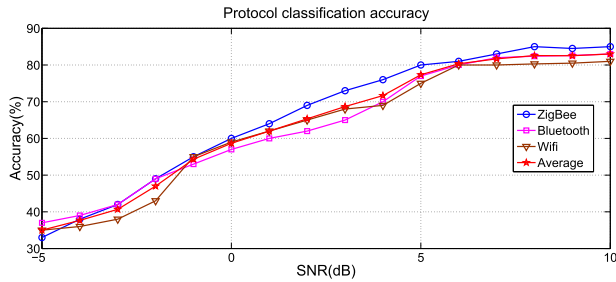


FIGURE 11. Protocol classification accuracy.

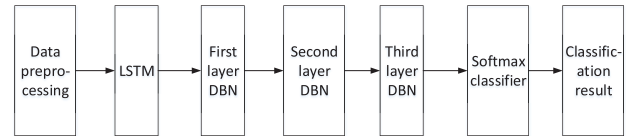


FIGURE 13. The structure of proposed algorithm.

TABLE 5. Detail network parameters.

Parameters	Detail
Input nodes	512
Output nodes of LSTM	512
Input nodes of first layer RBM	512
Input nodes of second layer RBM	256
Input nodes of third layer RBM	128
Number of softmax classifier output	3
Activation function	Relu
Optimizer	Momentum
Learning rate	0.001

requirement about sample data. The structure of the protocol classification algorithm based on LSTM and DBN is shown in Fig.13.

The algorithm process detail is as follows:

- (1) After data preprocessing, the data becomes  $512 \times 1$  array.
- (2) The preprocessed data is used as input to the LSTM. After processing, we obtain  $512 \times 1$  eigenvector.
- (3) After training the LSTM,  $512 \times 1$  extracted eigenvector is used as input to train the first layer RBM. After training, we obtain  $512 \times 1$  eigenvector.
- (4) Keep the training parameters of the first layer RBM.  $512 \times 1$  eigenvector is used as input to train the second layer RBM. After training, we obtain  $256 \times 1$  eigenvector.
- (5) Keep the training parameters of the above two layer RBM.  $256 \times 1$  eigenvector is used as input to train the third layer RBM. After training, we obtain  $128 \times 1$  eigenvector.
- (6) After training above three layer RBM, we use BP algorithm to adjust the whole network parameters. Finally, according to the output of softmax classifier, we can obtain the final classification results.

The detailed network parameters are shown in Table 5.

### B. SIMULATION RESULT AND ANALYSIS

In this section, we make a lot of experiments to verify the effectiveness and performance of the proposed algorithm. We use t-SNE algorithm to visualize the feature. Fig.14 shows the visualized feature under different SNR. The red number “0” represents ZigBee. The blue number “1” represents Bluetooth. The green number “2” represents Wifi..

From Fig.14, we can see that the extracted features have different discrimination under different SNR. With the decrease of SNR, the discrimination of features gradually decreases. When SNR = 10dB, three features don't overlap

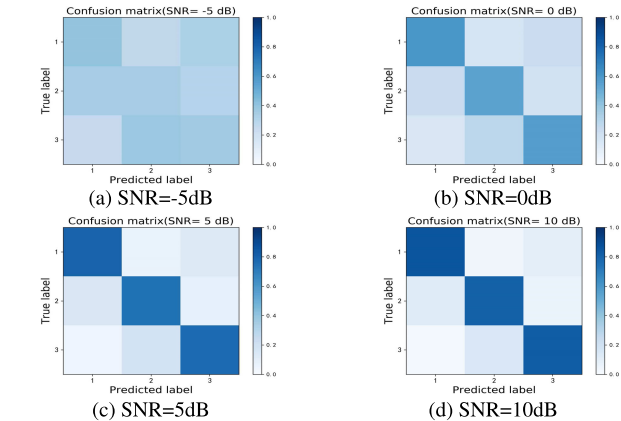


FIGURE 12. Confusion matrix under different SNR.

accuracy has reached 80%. However, with the increase of SNR, the average classification accuracy has not increased too much. When SNR = 10dB, the average classification accuracy doesn't reach 100%. This is because the input of DBN is the original IQ signal data, and the features extracted by DBN don't have good discrimination. Fig.12 shows the confusion matrix under different SNR.

From Fig.12, we can see that with the increase of SNR, the data outside the diagonal is less and less. When SNR = 10dB, most data is on the diagonal, the algorithm classification performance is not bad. When SNR = -5dB, few data is on the diagonal. But even SNR = 10dB, there are still some data outside the diagonal. This shows that even the SNR is high, the algorithm classification performance is still not optimal.

## IV. COMMUNICATION PROTOCOL CLASSIFICATION BASED ON DBN AND LSTM

### A. MODEL FRAME

From the analysis in the previous section, we find that the performance of protocol classification algorithm based on DBN is still not optimal even SNR is high. In order to improve the classification performance, we add LSTM after data preprocessing to further process the data. The advantage of doing so is that the input of DBN will have stronger ability to represent protocol. And the proposed algorithm doesn't need to process data complexly and the sample data doesn't need to obey the certain distribution. It greatly reduces the



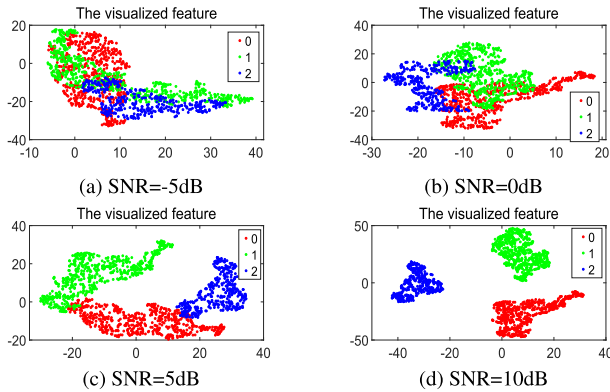


FIGURE 14. The visualized feature.

TABLE 6. Evaluation indicators under different SNR.

SNR	-5dB			0dB		
	$P_{avg}$	$R_{avg}$	$F_{1_{avg}}$	$P_{avg}$	$R_{avg}$	$F_{1_{avg}}$
	0.670	0.716	0.693	0.787	0.799	0.793
SNR	5dB			10dB		
	$P_{avg}$	$R_{avg}$	$F_{1_{avg}}$	$P_{avg}$	$R_{avg}$	$F_{1_{avg}}$
	0.941	0.952	0.946	1	1	1

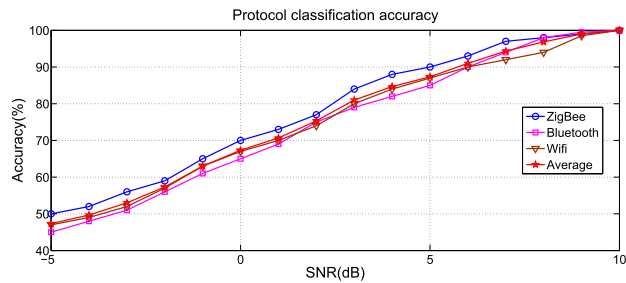


FIGURE 15. Protocol classification accuracy.

with each other, the feature discrimination is very good. When SNR = -5dB, some features are overlap with each other. However, compared with the protocol classification algorithm based on DBN, the feature discrimination has been significantly improved. It shows that by adding LSTM, the feature discrimination is more obvious, which is more conducive to protocol classification. Table 6 shows the evaluation indicators under different SNR.

From Table 6, we can see that with the increase of SNR,  $P_{avg}$ ,  $R_{avg}$  and  $F_{1_{avg}}$  are also increase, the algorithm classification performance is better and better. When SNR = 10dB,  $P_{avg} = 1$ ,  $R_{avg} = 1$  and  $F_{1_{avg}} = 1$ . At this time, the algorithm classification performance is the best. When SNR = -5dB,  $P_{avg} = 0.670$ ,  $R_{avg} = 0.716$  and  $F_{1_{avg}} = 0.693$ . At this time, the algorithm classification performance is poor. However, compared with the protocol classification algorithm based on DBN, evaluation indicators has been significantly increased. Fig.15 shows the each protocol classification accuracy and average classification accuracy.

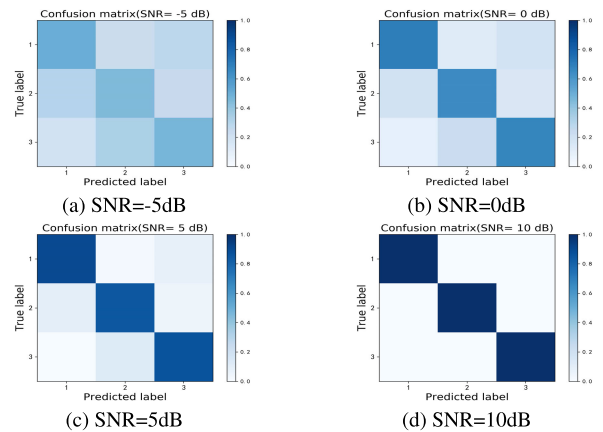


FIGURE 16. Confusion matrix under different SNR.

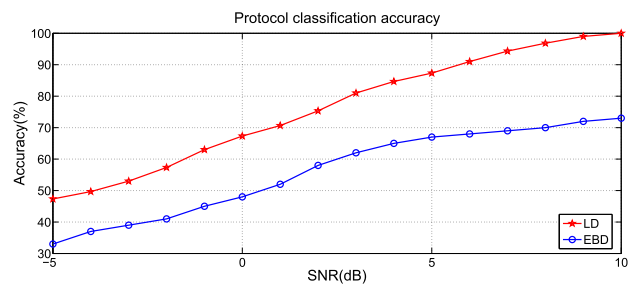


FIGURE 17. Average classification accuracy comparison.

From Fig.15, we can see that with the increase of SNR, classification accuracy is also increase. When SNR = 10dB, classification accuracy of ZigBee, Bluetooth and Wifi are all reach 100%. When SNR = -5dB, the average classification accuracy is only 47%. However, compared with the protocol classification algorithm based on DBN, the average classification accuracy has been significantly increased. Fig.16 shows the confusion matrix under different SNR.

From Fig.16, we can see that with the increase of SNR, the data outside the diagonal is less and less. When SNR = 10dB, all data is on the diagonal, the algorithm classification performance is best. When SNR = -5dB, few data is on the diagonal. However, compared with the protocol classification algorithm based on DBN, the number of data on the diagonal has been significantly increased. Based on the above results, we can see that the performance of protocol classification algorithm based on LSTM and DBN is better than performance of protocol classification algorithm based on DBN.

We compare the performance between the proposed algorithm in this paper and the proposed algorithm in [47]. Fig.17 shows the average classification accuracy comparison. LD represents the algorithm proposed in this paper and EBD represents the algorithm proposed in [47].

From Fig.17, we can see that under the same conditions, the average classification accuracy of LD is higher than EBD. When SNR = -5dB, the average classification accuracy of LD is 87%, the average classification accuracy of EBD is only 67%. Table 7 shows the evaluation indicators comparison.

TABLE 7. Evaluation indicators comparison.

SNR	Indicators	LD	EBD
-5dB	$P_{avg}$	0.670	0.336
	$R_{avg}$	0.716	0.340
	$F_{1_{avg}}$	0.693	0.338
0dB	$P_{avg}$	0.787	0.482
	$R_{avg}$	0.799	0.486
	$F_{1_{avg}}$	0.793	0.484
5dB	$P_{avg}$	0.941	0.635
	$R_{avg}$	0.952	0.688
	$F_{1_{avg}}$	0.946	0.652
10dB	$P_{avg}$	1	0.697
	$R_{avg}$	1	0.759
	$F_{1_{avg}}$	1	0.727

From Table 7, we can see that evaluation indicators of LD is higher than EBD. It shows that the proposed algorithm in this paper has better performance than the algorithm in [47].

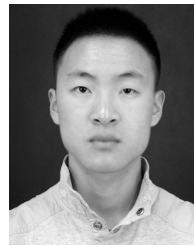
## V. CONCLUSION

In this paper, we study communication protocol classification, and proposed a novel protocol classification algorithm based on LSTM and DBN. The proposed algorithm does not need to process data complexly and the sample data doesn't need to obey the certain distribution. It greatly reduces the requirement about sample data. In this paper, we take three most common communication protocols (ZigBee, Bluetooth and WiFi) as example, and make a lot of simulation experiments to verify the effectiveness of the proposed algorithm. The simulation result shows that when SNR > 6dB, the classification accuracy can reach 90%.

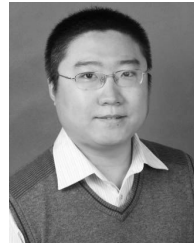
## REFERENCES

- O. Besson, A. Coluccia, E. Chaumette, G. Ricci, and F. Vincent, "Generalized likelihood ratio test for detection of Gaussian rank-one signals in Gaussian noise with unknown statistics," *IEEE Trans. Signal Process.*, vol. 65, no. 4, pp. 1082–1092, Feb. 2017.
- J. Villasenor and M. Tehranipoor, "Chop shop electronics," *IEEE Spectr.*, vol. 50, no. 10, pp. 41–45, Oct. 2013.
- M. Lichtman, J. D. Poston, S. Amuru, C. Shahriar, T. C. Clancy, R. M. Buehrer, and J. H. Reed, "A communications jamming taxonomy," *IEEE Secur. Privacy*, vol. 14, no. 1, pp. 47–54, Jan. 2016.
- F. Neri, *Introduction to Electronic Defense Systems*. Rijeka, Croatia: SciTech, 2006.
- Y. Zhao, C. Shang, and Z. Han, "Design and research of signal-level airborne electronic reconnaissance simulation system," *Int. J. Model., Simul., Sci. Comput.*, vol. 9, no. 4, Aug. 2018, Art. no. 1850030.
- Y. Lin, X. Zhu, Z. Zheng, Z. Dou, and R. Zhou, "The individual identification method of wireless device based on dimensionality reduction and machine learning," *J. Supercomput.*, vol. 75, no. 6, pp. 3010–3027, Jun. 2019.
- Z. Zhang, X. Guo, and Y. Lin, "Trust management method of D2D communication based on RF fingerprint identification," *IEEE Access*, vol. 6, pp. 66082–66087, 2018.
- Y. Tu, Y. Lin, J. Wang, and J.-U. Kim, "Semi-supervised learning with generative adversarial networks on digital signal modulation classification," *Comput. Mater. Continua*, vol. 55, no. 2, pp. 243–254, 2018.
- B. Hullar, S. Laki, and A. Gyorgy, "Efficient methods for early protocol identification," *IEEE J. Sel. Areas Commun.*, vol. 32, no. 10, pp. 1907–1918, Oct. 2014.
- Internet Assigned Numbers Authority (IANA). Accessed: Jan. 7, 2020. [Online]. Available: <http://www.iana.org/assignments/port-numbers>
- M.-S. Kim, Y. J. Won, and J. W.-K. Hong, "Application-level traffic monitoring and an analysis on IP networks," *ETRI J.*, vol. 27, no. 1, pp. 22–42, Feb. 2005.
- M. Roughan, S. Sen, O. Spatscheck, and N. Duffield, "Class-of-service mapping for QoS: A statistical signature-based approach to IP traffic classification," in *Proc. ACM SIGCOMM IMC*, Taormina, Italy, 2004, pp. 135–148.
- A. W. Moore and K. Papagiannaki, "Toward the accurate identification of network applications," in *Proc. PAM*, Boston, MA, USA, 2005, pp. 41–54.
- K. Kazemi and A. Fanian, "Tunneling protocols identification using light packet inspection," in *Proc. 12th Int. Iranian Soc. Cryptol. Conf. Inf. Secur. Cryptol. (ISCISC)*, Rasht, Iran, Sep. 2015, pp. 110–115.
- T. Chen and X. Liao, "An optimized solution of application layer protocol identification based on regular expressions," in *Proc. 18th Asia-Pacific Netw. Oper. Manage. Symp. (APNOMS)*, Kanazawa, Japan, Oct. 2016, pp. 1–4.
- X. Feng, X. Huang, X. Tian, and Y. Ma, "Automatic traffic signature extraction based on smith-waterman algorithm for traffic classification," in *Proc. 3rd IEEE Int. Conf. Broadband Netw. Multimedia Technol. (IC-BNMT)*, Oct. 2010, pp. 154–158.
- G. Xia and B. Liuy, "Accelerating network applications on X86-64 platforms," in *Proc. IEEE Symp. Comput. Commun.*, Riccione, Italy, Jun. 2010, pp. 906–912.
- A.-M. Yang, S.-Y. Jiang, and H. Deng, "A P2P network traffic classification method using SVM," in *Proc. 9th Int. Conf. Young Comput. Scientists*, Hunan, China, Nov. 2008, pp. 398–403.
- A. Este, F. Gringoli, and L. Salgarelli, "Support vector machines for TCP traffic classification," *Comput. Netw.*, vol. 53, no. 14, pp. 2476–2490, Sep. 2009.
- A. W. Moore and D. Zuev, "Internet traffic classification using Bayesian analysis techniques," in *Proc. ICMCCS*, Banff, AB, Canada, 2005, pp. 50–60.
- T. Liu, Y. Guan, and Y. Lin, "Research on modulation recognition with ensemble learning," *EURASIP J. Wireless Commun. Netw.*, vol. 2017, no. 1, p. 179, Dec. 2017.
- P. Smolensky, "Information processing in dynamical systems: Foundations of harmony theory," in *Parallel Distributed Processing*. Cambridge, MA, USA: MIT Press, 1986, pp. 194–281.
- F. Shuang and C. L. P. Chen, "A fuzzy restricted Boltzmann machine: Novel learning algorithms based on the crisp possibilistic mean value of fuzzy numbers," *IEEE Trans. Fuzzy Syst.*, vol. 26, no. 1, pp. 117–130, Feb. 2018.
- M.-A. Côté and H. Larochelle, "An infinite restricted Boltzmann machine," *Neural Comput.*, vol. 28, no. 7, pp. 1265–1288, Jul. 2016.
- G. Gui, H. Huang, Y. Song, and H. Sari, "Deep learning for an effective nonorthogonal multiple access scheme," *IEEE Trans. Veh. Technol.*, vol. 67, no. 9, pp. 8440–8450, Sep. 2018.
- H. Huang, J. Yang, H. Huang, Y. Song, and G. Gui, "Deep learning for super-resolution channel estimation and DOA estimation based massive MIMO system," *IEEE Trans. Veh. Technol.*, vol. 67, no. 9, pp. 8549–8560, Sep. 2018.
- Y. Nomura, A. S. Darmawan, Y. Yamaji, and M. Imada, "Restricted Boltzmann machine learning for solving strongly correlated quantum systems," *Phys. Rev. B, Condens. Matter*, vol. 96, no. 20, Nov. 2017, Art. no. 205152.
- R. D. Hjelm, V. D. Calhoun, R. Salakhutdinov, E. A. Allen, T. Adali, and S. M. Plis, "Restricted Boltzmann machines for neuroimaging: An application in identifying intrinsic networks," *NeuroImage*, vol. 96, pp. 245–260, Aug. 2014.
- L. F. Polania and K. E. Barner, "Exploiting restricted Boltzmann machines and deep belief networks in compressed sensing," *IEEE Trans. Signal Process.*, vol. 65, no. 17, pp. 4538–4550, Sep. 2017.
- T. Brosch and R. Tam, "Efficient training of convolutional deep belief networks in the frequency domain for application to high-resolution 2D and 3D images," *Neural Comput.*, vol. 27, no. 1, pp. 211–227, Jan. 2015.
- F. Rudzicz, A. Frydenlund, S. Robertson, and P. Thaine, "Acoustic-articulatory relationships and inversion in sum-product and deep-belief networks," *Speech Commun.*, vol. 79, pp. 61–73, May 2016.
- M. Liu, T. Song, and G. Gui, "Deep cognitive perspective: Resource allocation for NOMA-based heterogeneous IoT with imperfect SIC," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 2885–2894, Apr. 2019.

- [33] M. Choi and J. Tani, "Predictive coding for dynamic visual processing: Development of functional hierarchy in a multiple spatiotemporal scales RNN model," *Neural Comput.*, vol. 30, no. 1, pp. 237–270, Jan. 2018.
- [34] Y. Lin, C. Wang, J. Wang, and Z. Dou, "A novel dynamic spectrum access framework based on reinforcement learning for cognitive radio sensor networks," *Sensors*, vol. 16, no. 10, p. 1675, 2016.
- [35] J. Song, Y. Guo, L. Gao, X. Li, A. Hanjalic, and H. T. Shen, "From deterministic to generative: Multimodal stochastic RNNs for video captioning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 10, pp. 3047–3058, Oct. 2019.
- [36] Y. Li, Y. Lin, X. Cheng, Z. Xiao, F. Shu, and G. Gui, "Nonconvex penalized regularization for robust sparse recovery in the presence of *sax* noise," *IEEE Access*, vol. 6, pp. 25474–25485, 2018.
- [37] H. Wang, L. Guo, Z. Dou, and Y. Lin, "A new method of cognitive signal recognition based on hybrid information entropy and D-S evidence theory," *Mobile Netw. Appl.*, vol. 23, no. 4, pp. 677–685, Aug. 2018.
- [38] Q. Cui, S. Wu, Q. Liu, W. Zhong, and L. Wang, "MV-RNN: A multi-view recurrent neural network for sequential recommendation," *IEEE Trans. Knowl. Data Eng.*, vol. 32, no. 2, pp. 317–331, Feb. 2020.
- [39] M. Cai and J. Liu, "Maxout neurons for deep convolutional and LSTM neural networks in speech recognition," *Speech Commun.*, vol. 77, pp. 53–64, Mar. 2016.
- [40] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [41] M. Cornia, L. Baraldi, G. Serra, and R. Cucchiara, "Predicting human eye fixations via an LSTM-based saliency attentive model," *IEEE Trans. Image Process.*, vol. 27, no. 10, pp. 5142–5154, Oct. 2018.
- [42] J. Liu, G. Wang, L.-Y. Duan, K. Abdiyeva, and A. C. Kot, "Skeleton-based human action recognition with global context-aware attention LSTM networks," *IEEE Trans. Image Process.*, vol. 27, no. 4, pp. 1586–1599, Apr. 2018.
- [43] J. Zhang, Y. Zhu, X. Zhang, M. Ye, and J. Yang, "Developing a long short-term memory (LSTM) based model for predicting water table depth in agricultural areas," *J. Hydrol.*, vol. 561, pp. 918–929, Jun. 2018.
- [44] Y. Xian and Y. Tian, "Self-guiding multimodal LSTM—when we do not have a perfect training dataset for image captioning," *IEEE Trans. Image Process.*, vol. 28, no. 11, pp. 5241–5252, Nov. 2019.
- [45] S. Song, C. Lan, J. Xing, W. Zeng, and J. Liu, "Spatio-temporal attention-based LSTM networks for 3D action recognition and detection," *IEEE Trans. Image Process.*, vol. 27, no. 7, pp. 3459–3471, Mar. 2018.
- [46] D. Zhang, G. Lindholm, and H. Ratnaweera, "Use long short-term memory to enhance Internet of Things for combined sewer overflow monitoring," *J. Hydrol.*, vol. 556, pp. 409–418, Jan. 2018.
- [47] L. Zhang, W. Xiang, and X. Tang, "An efficient bit-detecting protocol for continuous tag recognition in mobile RFID systems," *IEEE Trans. Mobile Comput.*, vol. 17, no. 3, pp. 503–516, Mar. 2018.



**WENWEN LI** was born in Harbin, China, in 1992. He received the B.E. degree in electrical information from Harbin Engineering University, Harbin, China, in 2014, where he is currently pursuing the Ph.D. degree with the School of Information and Communication Engineering. His current research interests include signal recognition and signal processing.



**ZHENG DOU** was born in Harbin, China, in 1978. He received the B.E. degree in wireless technology and the M.E. and Ph.D. degrees from Harbin Engineering University, Harbin, China, in 2001, 2004, and 2007, respectively. He has been with Harbin Engineering University, since 2007, where he is currently a Professor. His current research interests include cognitive radio networks, intelligent communication systems, intelligent communication, and electronic jamming communication.



**LIN QI** was born in Harbin, China, in 1979. She received the B.S., M.S., and Ph.D. degrees in communication engineering, signal and information processing, and communication and information system from Harbin Engineering University, China, in 2002, 2005, and 2011, respectively. From 2002 to 2005, she was an Assistant Professor with the College of Information and Communication Engineering, Harbin Engineering University, where she has been a Lecturer, since 2007. She also worked as a Visiting Scholar at the Institute of Materials and Systems for Sustainability, Nagoya University, from 2015 to 2016. Her current research interests include wideband digital communication systems, SDR, and communication signal processing.

...