IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received April 10, 2020, accepted April 19, 2020, date of publication April 22, 2020, date of current version May 8, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2989498

Intelligent Processing of Intrusion Detection Data

TAO DUAN[1,3,4,5,6], YOUHUI TIAN[2], HANRUI ZHANG[1,3,5,6], YAOZONG LIU[3], QIANMU LI[1], JIAN JIANG[4], AND ZONGSHENG SHI[4]

[1]School of Cyber Science and Engineering, Nanjing University of Science and Technology, Nanjing 210094, China
[2]Jiangsu Vocational Institute of Commerce, Nanjing 211168, China
[3]Intelligent Manufacturing Department, Wuyi University, Jiangmen 529020, China
[4]Jiangsu Zhongtian Internet Technology Company, Ltd., Nantong 226463, China
[5]Jiangsu Graduate Workstation, Nanjing University of Science and Technology, Nanjing 210094, China
[6]Nanjing Liancheng Technology Development Company, Ltd., Nanjing 210008, China

Corresponding authors: Hanrui Zhang (jessica_9533@njust.edu.cn) and Qianmu Li (qianmu@njust.edu.cn)

This work was supported in part by the 2019 Industrial Internet Innovation and Development Project from Ministry of Industry and Information Technology of China, in part by the 2018 Jiangsu Province Major Technical Research Project ''Information Security Simulation System'', and in part by the Fundamental Research Funds for the Central Universities under Grant 30918012204.

ABSTRACT Intrusion detection technology, as an active and effective dynamic network defense technology, has rapidly become a hot research topic in the field of network security since it was proposed. However, current intrusion detection still faces some problems and challenges that affect its detection performance. Especially with the rapid development of the current network, the volume and dimension of network data are increasing day by day, and the network is full of a large number of unlabeled data, which brings great pressure on the data processing methods of IDS. In view of the tremendous pressure of intrusion detection brought by the current complex and high-dimensional network environment, this paper provides a feasible solution. Firstly, this paper briefly outlines the necessity of feature learning, the shortcomings of traditional feature learning methods and the new breakthroughs brought by deep belief network in feature learning, and focuses on the principle and working mechanism of deep belief network and Principal Component Analysis (PCA). Then, it constructs the intrusion detection model based on PCA-BP and DBN respectively. And through the experimental evaluation of the two detection models, a comparative experiment between deep belief network and principal component analysis is constructed. The experimental results show that deep belief network has unique advantages and good performance in feature learning. Therefore, deep belief network can be applied in the field of intrusion detection to extract effective features from the current high-dimensional and redundant network data, thereby improving the detection performance of IDS and its adaptability to the current complex and high-dimensional network environment.

INDEX TERMS Intrusion detection, data mining, deep belief network.

I. INTRODUCTION

In the current stage of intrusion detection research, integrating intelligent technologies such as expert systems, statistical analysis, and data mining into intrusion detection has become a hot topic in the field of intrusion detection [1]–[5]. The mainstream intelligent processing algorithms include: (1) Genetic algorithms [6], [7]. This is a global optimization algorithm. This type of algorithm introduces the idea of natural selection and survival of the fittest in evolution theory to optimize IDS. IDS based on this technology has better detection capabilities for different types of attacks, but its real-time performance is low. (2) Statistical analysis [8]–[10]. Statistical analysis-based intrusion detection assumes that the user's normal operations are followed regularly. Therefore, some statistical variables can be used to describe user behavior. The intrusion detection system has higher detection rate and availability, but also has disadvantages such as poor real-time performance. (3) Knowledge map analysis [11]–[12]. This kind of method first constructs a knowledge graph describing attack behavior and constitutes the corresponding reasoning rules. Then, an inference algorithm is used to automatically analyze and determine whether there is an intrusion behavior. This type of intrusion detection system can achieve high detection efficiency. But it can only detect known attacks and it is more difficult to maintain. (4) Data mining technology [13]–[15]. Intrusion detection based on data mining technology transforms the detection process

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

into a data analysis process. Analyze and process audit data to discover hidden illegal behavior. It has better effectiveness and adaptability.

The above intrusion detection technologies generally have some problems that need to be solved urgently:

(1) Low detection efficiency

The current network speed is constantly increasing, and data traffic is also expanding. IDS need to spend more system resources and time to capture network packets and analyze to match whether there is a certain attack feature [16]. If the transmission speed of IDS cannot be compared with the data traffic, the performance indicators of most intrusion detection-related products are still far from the actual requirements. Therefore, how to improve the processing speed of IDS to adapt to the current high-bandwidth, high-speed network environment has become a major challenge [17].

(2) Poor adaptability

The network topology is constantly changing due to changes in network topology. However, the development of traditional IDS did not consider the needs of a specific network environment. This will inevitably cause the system to fail to take corresponding measures according to changes in the environment, and eventually lead to poor performance [18]–[20]. Therefore, how to enable IDS to automatically adjust the system through changes in the network platform, and to ensure the efficiency of the system for a long time, are all urgent problems to be solved.

How to further improve the detection rate and reduce the false detection rate is the challenge of this type of method research.

In the intelligent processing of intrusion detection data, a classifier is usually used to detect and identify the current network connection. And the quality of the classifier's classification effect has a lot to do with the accuracy of the input data to represent the original valid information. However, as the development of the network continues to complicate, the feature dimensions of network data instances continue to expand [21]. This high-dimensional network data contains a lot of redundant information. The existence of this information not only cannot help intrusion detection, but even affects the detection performance of the final classifier. Therefore, before the intrusion detection model performs feature learning, it becomes a necessary process to obtain better low-dimensional feature information of the original data [22].

There are two traditional feature learning methods: feature selection [23]–[25] and feature extraction [26], [27]. Feature selection uses the evaluation function to select feature sets that are relevant or important from the original feature set to form a feature subset. So as to reduce the feature dimension. In other words, feature selection ignores redundant, independent features that do not contribute to class discrimination, or contribute little. Feature extraction uses transformation methods to transform high-dimensional features to low-dimensional features.

Common feature selection methods include information gain IG, information gain ratio, distance measurement, and so on [28]–[30]. Feature extraction methods include PCA, PCA, ICA, and so on [31], [32]. Although they can obtain the effective feature information of the original data to a certain extent, they can reduce the size of the classifier by reducing the feature dimension. However, in the process of feature learning, there is often a large loss of information, which leads to the final classification accuracy is not too ideal.

The proposal of deep learning just makes up for the deficiency of traditional feature learning. The study found that deep learning has unique advantages and good performance in feature learning. The deep network structure on which deep learning is based is similar to the structure in which the human cerebral cortex processes information. Therefore, the original data features can be abstracted layer by layer from bottom to top to reduce the feature dimension. The loss of feature information is reduced. Therefore, the feature information that is most suitable for classifier training and recognition can finally be obtained.

Deep belief network belongs to the deep network structure. The network structure contains multiple hidden layers. The multi-layer hidden layer extracts the optimal low-dimensional representation of the original features by performing layer-by-layer non-linear feature transformation on the original data.

This paper constructs an intrusion detection model based on PCA-BP and an intrusion detection model based on DBN. Through the experimental evaluation of the detection model, the comparative demonstration demonstrates the advantages and good performance of deep belief networks in feature learning over principal component analysis.

II. INTRUSION DETECTION DATA PROCESSING MODEL BASED ON PRINCIPAL COMPONENT ANALYSIS AND BP NEURAL NETWORK

Principal component analysis is a method to evaluate and analyze multi-indicator problems into fewer indicators. It is a transformation method commonly used in feature learning. Because the top layer of the deep belief network is composed of BP neural network, when applying DBN to the field of intrusion detection data processing, this layer of BP neural network can be used as a classifier for detection and identification. Therefore, for the purpose of comparison, the feature learning performance of PCA and DBN in the field of intrusion detection data processing is compared. This paper uses principal component analysis algorithm to perform feature learning on the collected network data, and combines BP neural network as a classifier to construct an intrusion detection data processing model based on principal component analysis and BP neural network.

A. MODEL FRAMEWORK

The framework of intrusion detection data processing model based on PCA-BP neural network is shown in the figure below.

Here, the BP neural network is used as a classifier, and the output of the PCA feature learning module is used as an input

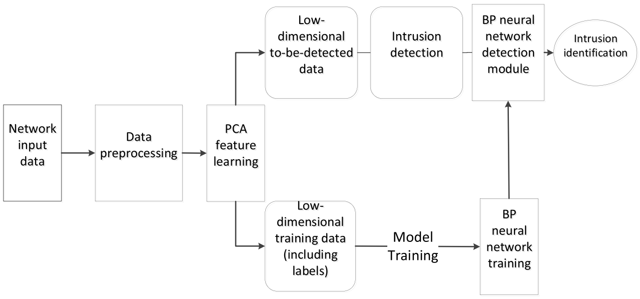
Network input data
Data preprocessing
PCA feature learning
Low-dimensional to-be-detected data
Intrusion detection
BP neural network detection module
Intrusion identification
Low-dimensional training data (including labels)
Model Training
BP neural network training

FIGURE 1. Intrusion detection data processing model based on PCA-BP neural network.

to train the classifier and classify and identify the intrusion behavior. The construction of the BP neural network structure includes setting the number of network layers, the number of neurons in the input and output layers, the number of neurons in the hidden layer, the initial value of the connection weight between each layer, and the initial value of the bias between the hidden layer and the output layer., And other parameter information. The value of each parameter can be set according to the amount of input and output data of the network and related research. Here:

Number of network layers: BP neural network contains one input, one output layer and N hidden layers. The larger the number of hidden layers, the longer the network learning time. And the network is more likely to fall into local minima during the learning process. Therefore, based on a comprehensive consideration of efficiency and complexity, this paper uses a 3-layer BP network with a hidden layer.

Number of hidden layer neurons: The choice of the number of hidden layer nodes plays a key role in whether the BP neural network can achieve the desired effect. There is no theoretical guidance for the selection of the optimal number of hidden nodes. There are only some empirical formulas. The following two are more commonly used here:

$$N = \sqrt{n \times m}$$
$$N = \sqrt{n + m} + a$$

Here N is the number of nodes in the hidden layer, m is the number of nodes in the output layer, n is the number of nodes in the input layer, and a is a positive integer of [1, 10].

B. ALGORITHMIC IDEAS

The basic idea of intrusion detection data processing based on PCA-BP neural network is:

First, the high-dimensional standard to-be-detected data generated by the preprocessing module is subjected to feature learning through the PCA feature learning module to eliminate redundant information. Thus, an effective low-dimensional representation of high-dimensional data is obtained. Then, the low-dimensional to-be-detected data is used as the input of the BP neural network to perform intrusion recognition through the BP network. Here, the BP neural network classifier uses a multi-output competitive classification method. The classification principle is: each output node represents a classification category. When input data is given, after network learning processing, each output node gets a value between 0 and 1. This value represents the probability that the input data belongs to this category, and the output with the highest probability represents the classification result of the current data.

C. WORKING STEPS

The working steps of the intrusion detection data processing model based on PCA-BP neural network are as follows:

Model training phase:

Step 1: Pre-process the labeled training data to obtain high-dimensional labeled training with standardized format data;

Step 2: Use PCA to perform feature learning on the pre-processed training data to eliminate redundant and useless information.

Get its effective low-dimensional representation;

Step 3: Use the low-dimensional labeled training data obtained by the PCA feature learning module as the BP neural network

Input to train the classifier. The error back-propagation algorithm is used to iteratively train the network weights until the total network error meets the accuracy requirements or reaches the maximum number of trainings.

Intrusion detection data processing steps:

Step 1: Pre-process the acquired data to be tested to obtain high-dimensional samples to be tested with standardized format data;

Step 2: Use PCA to perform feature learning on the pre-processed data to be detected, eliminating redundant and useless information to get its effective low-dimensional representation;

Step 3: Based on the effective detection module obtained during the training phase, and use the low-dimensional to-be-detected data obtained by the PCA feature learning module as the input of the detection module for intrusion recognition. Perform non-linear mapping on the low-dimensional to-be-detected data, and determine whether an intrusion has occurred based on the output results.

III. INTRUSION DETECTION DATA PROCESSING MODEL BASED ON DEEP BELIEF NETWORK

Deep Belief Network (DBN) is a deep learning structure. It stacks multiple unsupervised restricted Boltzmann machines (RBM), and at the top layer, it sets up a supervised reverse Back Propagation (BP) network. The model framework is shown in Figure 2. DBN extracts the original input layer by layer, from concrete to abstract. The deep structure composed of multi-layer RBM can weaken the redundancy or error information in the feature extraction process layer by layer. Therefore, the top-level BP network can obtain feature vectors that are easier to be classified.

Deep belief networks have unique advantages and good performance in feature learning. Its deep structure composed of multiple layers of RBM can perform a non-linear

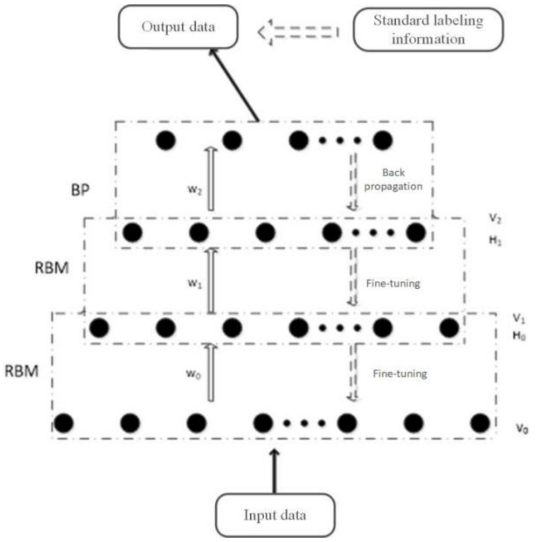

FIGURE 2. Deep belief network model.

transformationP from the bottom to the top of the original data features. This reduces the feature dimension and reduces information loss. At the same time, the structure can also weaken the error or redundant information in the feature learning process layer by layer. Finally, low-dimensional feature information suitable for classifier training and recognition is obtained. Because the top layer of DBN is BP neural network, which has better classification and recognition ability, based on this, this paper constructs an intrusion detection data processing model based on deep belief network.

A. MODEL FRAMEWORK

The framework of a DBN-based intrusion detection data processing model is shown below.

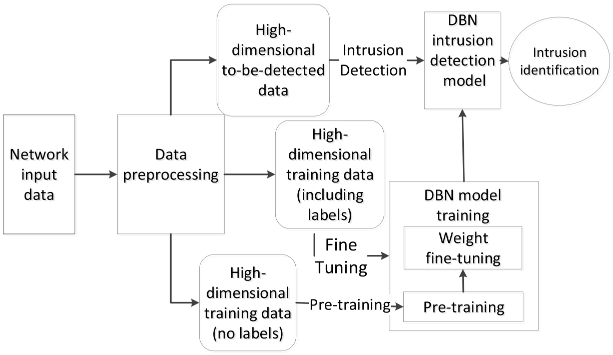
Network input data
Data preprocessing
High-dimensional to-be-detected data
Intrusion Detection
DBN intrusion detection model
Intrusion identification
High-dimensional training data (including labels)
Fine Tuning
DBN model training
Weight fine-tuning
High-dimensional training data (no labels)
Pre-training
Pre-training

FIGURE 3. DBN-based intrusion detection data processing model.

Here DBN is used as a feature learning and classification module, and receives the output of the data preprocessing module as input, which is used to train a classifier and classify and identify intrusion behaviors. The structure of the DBN model structure can be determined according to the amount of input and output data and experimental analysis or related research. This includes setting the number of RBMs, related parameters of each layer of RBM, the number of nodes in the DBN input layer (the first layer of RBM's explicit layer), the number of nodes in each hidden layer of DBN, and related parameters of the top BP neural network.

The number of RBMs and the number of nodes in each hidden layer of DBN: The choice of these two parameters has a great impact on the overall performance of the DBN algorithm: if the number of RBMs is too large, the overall network training process will be too much complex. The learning time is long and the error is difficult to adjust to the global minimum, which seriously affects the learning effect of the overall network. This leads to poor algorithm performance. If the number is too small, the deep network structure will be degraded to a shallow structure, and complex sample data cannot be processed. If the number of nodes in each hidden layer is too large, not only will the overall network learning time be long, but it will also easily lead to overfitting. If the number of nodes is too small, a good convergence effect may not be achieved. There is no theoretical guidance for the optimal selection of these two parameters, and it is highly subjective in practical applications. For the selection of these parameters, the method adopted in this paper is to obtain better solutions based on a large number of comparative experiments.

The number of nodes in the DBN input layer: It can be determined according to the data feature dimension generated by the data preprocessing module.

Relevant parameters of each layer of RBM: This part needs to set the initial value of the connection weight of each layer of RBM and the initial value of the explicit layer and the hidden layer offset. The specific parameter values are given in this paper based on related research and actual conditions.

Related parameters of the top BP neural network: In the DBN model structure, the top BP neural network is a 2-layer network structure without a hidden layer, and it uses the output layer of the last layer of the RBM of the DBN as the input layer. Therefore, this part only needs to set the number of output layer nodes, the initial value of the output layer offset, and the initial value of the connection weight. In this paper, the value of the number of nodes in the output layer is the same as that of the BP network in the PCA-BP detection model.

B. ALGORITHMIC IDEAS

The basic idea of DBN-based intrusion detection data processing is to use the high-dimensional standard to-be-detected sample data generated by the preprocessing module as the DBN input. Firstly, through layer-by-layer feature learning of multi-layer RBM, redundant and useless information is eliminated to obtain its effective low-dimensional representation. Then take the output of the last layer of RBM (that is, the effective low-dimensional data to be detected) as the input of the top BP neural network. Thus, intrusion identification is performed through the BP network. Here, the principle of BP neural network classification is the same as the BP network classifier in the PCA-BP-based detection model.

C. WORKING STEPS

The working steps of the DBN-based intrusion detection data processing model are as follows:

Model training phase:

1) PRE-TRAINING PHASE

Step 1: Preprocess a large amount of unlabeled training data to obtain high-dimensional unlabeled training data with standardized format.

Step 2: Use the preprocessed high-dimensional unlabeled training data as the input to the initial DBN model. An unsupervised, greedy algorithm is used to pre-train the initial model. Thus, a better model parameter is obtained. The RBM parameters of each layer are sequentially trained from the bottom up using the CD algorithm until all RBM training is completed. Here, after the training of a certain layer of RBM, the parameters of this layer need to be fixed. And the output of this layer is used as the input of the lower layer RBM in order to continue training the lower layer RBM.

2) FINE ADJUSTMENT OF WEIGHTS

Step 1: Pre-process the labeled training data to obtain high-dimensional labeled training data with standardized format.

Step 2: Use the pre-processed high-dimensional labeled training data as the input of the better model obtained in the pre-training stage. And adopt the error back propagation algorithm to optimize the model parameters of the whole network globally. Thus, the optimal parameter weight is obtained. Here, the initialization and training of the top-level BP neural network is included in this stage.

Intrusion detection data processing stage:

Step 1: Preprocess the acquired network data to be detected. In this way, high-dimensional to-be-tested sample data with standardized format is obtained.

Step 2: Based on the effective DBN model obtained during the model training phase. Feature learning and output calculation are performed on the preprocessed valid high-dimensional to-be-detected data. And judge whether the data behavior is abnormal according to the output result. The judgment principle here is the same as the BP network classifier in intrusion detection data processing based on PCA-BP.

IV. FEATURE LEARNING COMPARISON EXPERIMENT

In order to analyze the advantages of deep belief networks in feature learning, this paper uses the NSL-KDD dataset to evaluate the PCA-BP-based intrusion detection data processing model and the DBN-based intrusion detection data processing model to form a comparison between DBN and PCA. experiment.

In the field of intrusion detection data processing, the traditional KDD99 data set has always been the benchmark data set for intrusion detection data processing evaluation. However, the research on the data set found that the data set has greater redundancy. This large amount of redundant information makes it impossible to evaluate the intrusion detection data processing algorithms effectively. Therefore, in order to provide more practical and effective evaluation results, this paper uses the optimized version of the KDD99 dataset-the NSL-KDD dataset to evaluate the detection model.

The NSL-KDD data set optimizes the redundancy of the KDD99 data set and the ratio of the training set to the test set. The new data set contains 125,973 training data and 22,544 test data. And each piece of data belongs to one of five types of attacks: Normal, DOS, Probing, R2L, and U2R. The following table lists the attack category distributions included in the training and test sets:

TABLE 1. NSL-KDD training set attack category distribution.

Attack type	Number of data	Rate
Normal	67,343	53.46%
DOS	45,927	36.46%
Probing	11,656	9.25%
R2L	995	0.79%
U2R	52	0.04%

TABLE 2. NSL-KDD test set attack category distribution.

Attack type	Number of data	Rate
Normal	9711	43.08%
DOS	7167	31.79%
Probing	2421	10.74%
R2L	3178	14.1%
U2R	67	0.3%

Here DOS, Probing, R2L, U2R four types of attacks can be divided into 39 specific types of attacks. The training set contains 22 kinds, and the other 17 kinds only appear in the test set.

The record format of the NSL-KDD dataset is the same as that of the KDD99 dataset. That is, each piece of data is composed of 41 features and an attack category identifier. Here, 41 features are composed of 38 numeric features and 3 character features. Specific examples of data records are as follows. (Each eigenvalue is separated by a comma.):

0,tcp,http,SF,54540,8314,0,0,0,2,0,1,1,0,0,0,0,0,0,0,0,0,
4,4,0.00,0.00,0.00,0.00,1.00,0.00,0.00,139,139,1.00,0.00,
0.01,0.00,0.00,0.00,0.05,0.05, back

In the above example, the first feature represents the connection time of the network. The second character data character indicates the type of protocol used for the connection. There are three values: TCP, UDP, and ICMP. The third character data character indicates the service type of the destination host, with a total of 70 values. The fourth character-type data feature indicates the status flag of the connection, with a total of 11 values. The 5th to 41st features indicate the packet parameters at the time of connection. The 42nd feature indicates the attack type identification.

A. EVALUATION CRITERIA

This article mainly uses the accuracy rate, detection rate, and false alarm rate as evaluation indicators to measure the performance of intrusion detection data processing models. Their definitions are given below:

Accuracy: The ratio of the number of correctly identified samples to the total number of test data samples. The higher the value, the better the model performance. Calculated as follows:

Accuracy = (number of correctly identified normal samples + number of correctly identified abnormal samples) / (total number of samples in the test data)

False alarm rate: It indicates the ratio of the number of normal samples that are misclassified to the total number of normal samples. The lower the value, the better the model performance. Calculated as follows:

False Positive Rate = (Number of Normal Samples Mis detected as Intrusion) / (Total Number of Normal Samples in Test Data)

Detection rate: indicates the ratio of the number of samples that are correctly identified for an anomaly to the total number of corresponding anomaly samples. The higher the value, the better the model's adaptability to the anomaly. Calculated as follows:

Detection rate = (number of samples for which an anomaly was correctly identified) / (total number of corresponding anomaly samples in the test data)

B. PRETREATMENT

The connection records of the NSL-KDD dataset are all composed of 38 numerical features and 3-character features. And through research on the specific connection records, it is found that there are large differences in the magnitude of the values of multiple numerical features in the data records. For example, in a record, the value of the dst_bytes feature is 14421, while the value of the dst_host_rerror_rate feature is 0.02. Therefore, the data records of the NSL-KDD dataset have differences in the feature format and the magnitude of the values. The existence of these differences will hinder the processing of the data by the detection model, and then affect the classification accuracy of the model. Therefore, the data set needs to be pre-processed in advance to standardize it.

The preprocessing of the NSL-KDD data set in this paper includes three steps: the digitization of character features, the normalization of numeric features, and the digitization of category identifiers, as follows:

1) NUMERICAL PROCESSING OF CHARACTER FEATURES

From the introduction of the NSL-KDD data set, it can be known that in each data record, the second column feature protocol_type, the third column feature service, and the fourth column feature flag are all character-type features. Therefore, the three columns of features need to be pre-processed separately. This article adopts the feature mapping method to realize the digitization of character-type features. The specific processing method is:

① Count all the values contained in the corresponding character features.

② Map the corresponding character data with the numeric vector one by one.

For example, the protocol type protocol_type contains three values: TCP, UDP, and ICMP. Therefore, the mapping result with the numeric vector is shown in the following table:

TABLE 3. Example of protocol_type mapping.

Character type	Numeric vector
TCP	1,0,0
UDP	0,1,0
ICMP	0,0,1

In NSL-KDD data, protocol type contains a total of 3-character values, service contains a total of 70-character values, and flag contains a total of 11-character values. Therefore, after these character data are pre-processed using a mapping method similar to the above table, the 41-dimensional data in the original NSL-KDD data set is converted into 122-dimensional data.

2) NORMALIZATION OF NUMERICAL FEATURES

This paper uses the maximum and minimum method to normalize the numerical features in the NSL-KDD data set processed in step 1). The features are compressed to the range of [0,1] to eliminate the order of magnitude difference between the numerical features.

3) NUMERICAL PROCESSING OF CATEGORY IDENTIFICATION

The five categories of identifiers are mapped in vector form, and the corresponding codes are as follows:

TABLE 4. Code mapping for category identification.

Category identification	Mapping encoding
Normal	1,0,0,0,0
DOS	0,1,0,0,0
Probing	0,0,1,0,0
R2L	0,0,0,1,0
U2R	0,0,0,0,1

C. EXPERIMENT AND ANALYSIS

The experimental evaluation of the intrusion detection data processing model mainly includes two parts: model parameter setting and simulation experiment analysis. Here, the model parameter setting part aims to determine the optimal structure and weight coefficient of the model through repeated experiments. The simulation experiment analysis aims to build a detection model based on the better model parameters determined by the model parameter setting section. And a lot of experiments are performed to accurately evaluate the performance of the model. Both of the above experiments were performed on the NSL-KDD data set. A detailed description of the corresponding experiments is given below.

1) MODEL PARAMETER SETTING

a: PARAMETER SETTING OF PCA-BP DETECTION MODEL

In order to obtain the optimal structure and weight coefficient of the PCA-BP model, the following parameters need to be determined or provided:

① Number of principal components.

② Structural parameters of BP neural network:

Number of network layers.

Number of neurons in the input layer.

Number of neurons in the output layer.

Number of hidden layer neurons.

The initial value of the connection weight between layers.

The offset initial value of the hidden layer and the output layer.

③ Parameters of BP neural network training phase:

Learning rate during training.

Number of training iterations.

Here, the structural parameters of the BP neural network and the parameters of the training phase of the BP neural network can be given according to related research and actual conditions. However, the parameter information of the number of principal components has a greater impact on the overall performance of the PCA-BP model, and its value lacks better theoretical support. Therefore, comparative experiments are needed to obtain its better parameter values. The specific values of each parameter are given below:

1) Number of network layers of BP neural network: This paper uses a three-layer BP network with one hidden layer.

2) Number of neurons in the input layer of the BP neural network: The value of this parameter is the same as the feature dimension after the PCA feature learning. That is, it is the same as the number of principal components. Therefore, after determining the number of better principal components, this parameter can be determined.

3) Number of neurons in the output layer of the BP neural network: The value of this parameter can be determined according to the number of types of classification results. Because the experiment is based on the NSL-KDD data set, and this data set contains a total of 5 types of network data, the value of this parameter is 5.

4) Number of hidden layer neurons in BP neural network: This article uses a formula to determine the number of hidden layer neurons that are better in BP network. Here, N is the number of nodes in the hidden layer, n is the number of nodes in the input layer, and m is the number of nodes in the output layer.

5) The initial values of the connection weights between the layers of the BP neural network and the initial offset values of the hidden layer and the output layer. This article uses the following methods to initialize it:

rnd = np.random.RandomState (4444)

self.weights = [rnd.uniform (−1,1, (y, x)) for x, y in zip (sizes [:−1], sizes [1:])]

self.biases = [rnd.uniform (−0.4,0.4, (y, 1)) for y in sizes [1:]]

Here, sizes stores the number of neurons in each layer.

For example: sizes = [2,3,2] means there are 2 neurons in the input layer, 3 neurons in the hidden layer, and 2 neurons in the output layer.

The learning rate of BP neural network during training is 3.

The number of iterations during training of the BP neural network is 200.

① In the comparison experiment to determine the number of principal components, the maximum accuracy rate in 200 trainings is used as the optimal value of the current model.

② During the simulation experiment analysis, the model was constructed using the training times corresponding to the maximum accuracy of the 200 trainings determined based on the overall training set and the overall test set in the NSL-KDD data set. For the number of principal components, this paper uses the entire training data and all test data in the NSL-KDD data set to perform the following comparative experiments to determine its better parameter values. In the experiment, the cumulative contribution rate of the principal components of the training data and the accuracy rate of the PCA-BP detection model were used as evaluation indicators. By limiting other parameters to be the same, and only changing the number of principal components, a comparative experiment was performed. The experimental data are as follows:

As can be seen from the above table, the value of the number of principal components is very difficult to predict. Moreover, the number of principal components and the accuracy of PCA-BP are not purely linear. From the analysis of the above 22 experimental data, it can be known that when the number of principal components is 55, the cumulative contribution rate reaches 0.98075, and the accuracy rate of PCA-BP reaches 63.759759%, both of which are at a better level. Therefore, considering the cumulative contribution rate of the principal components and the accuracy of the model, this paper sets the number of principal components to 55.

So far, this section has determined all the parameter information required for the implementation of the PCA-BP model. Then, a detection model can be constructed based on the parameter information. And analyze it by simulation experiment.

b: PARAMETER SETTING OF DBN DETECTION MODEL

In order to obtain the better structure and weight coefficient of the DBN model, the following parameters need to be determined or provided:

TABLE 5. Comparative experiments on the number of principal components.

Number of principal components	Cumulative contribution rate	PCA-BP accuracy	Number of principal components	Cumulative contribution rate	PCA-BP accuracy
5	0.770211	59.7%	60	0.984993	63.4%
10	0.869683	62.9%	65	0.988841	60.9%
15	0.910381	58.6%	70	0.992154	61.9%
20	0.931663	61.4%	75	0.994894	61. 1%
25	0.944659	61.4%	80	0.996999	59.0%
30	0.952887	59.9%	85	0.998352	63.0%
35	0.959526	61.0%	90	0.999107	61.3%
40	0.965609	57.7%	95	0.999641	62.0%
45	0.971117	63.1%	100	0.999859	62.8%
50	0.976153	61.9%	105	0.999954	57.7%
55	0.98075	63.8%	110	0.999983	62.7%

① Structural parameters of DBN:

The number of RBM.

Relevant parameters of each layer of RBM:

The initial value of the bias between the visible and hidden layers.

Connection weight initial value.

The number of nodes in the DBN input layer (the first layer of the RBM).

Number of DBN hidden nodes.

Related parameters of the top BP neural network:

Number of output layer nodes.

Connection weight initial value.

Output layer offset initial value.

② Parameters in the pre-training phase:

Number of iterations of each layer of RBM.

The learning rate of each layer of RBM and the value of k in the CD algorithm.

③ Fine-tuning parameters:

Learning rate and number of iterations during fine-tuning.

Here, the number of nodes in the DBN input layer, the relevant parameters of each layer of RBM, the relevant parameters of the top BP neural network, the learning rate of each layer of RBM during pre-training, the value of k in the CD algorithm, the learning rate and iteration times during fine tuning Other parameters can be given according to relevant research and actual conditions. The parameters such as the number of RBMs, the number of nodes in each hidden layer of the DBN, and the number of iterations of the RBM in each layer in the pre-training have a great impact on the overall performance of the DBN model, and the value lacks theoretical support. Therefore, it is necessary to obtain their better parameter values through comparative experiments. The specific values of each parameter are given below:

Number of nodes in the DBN input layer: The pre-processed NSL-KDD dataset contains a total of 122-dimensional feature information. Therefore, the DBN input layer is set to contain 122 nodes, and each node corresponds to a feature information.

For the relevant parameters of each layer of RBM:

The explicit and hidden layer offsets are initialized to 0.

The connection weight is initialized to:

Uniform sampling between $-4*\text{sqrt}\ (6\ /\ (\text{n\_visible} + \text{n\_hidden}))$ and $4 * \text{sqrt}\ (6\ /\ (\text{n\_hidden} + \text{n\_visible}))$.

Here n_visible represents the number of nodes in the display layer. n_hidden represents the number of hidden layer nodes.

Related parameters of the top BP neural network:

Number of output layer nodes is 5.

Output layer offset is initialized to 0.

The initialization of connection weights is the same as the initialization of connection weights in RBM.

RBM of each layer during pre-training:

Learning rate is 3.

The value of k in the CD algorithm is 1.

The value of the learning rate during fine-tuning is 10.

For the number of iterations during fine-tuning, the value is 1000:

① In the comparison experiment of parameter setting, the maximum accuracy rate in 1000 fine adjustments is used as the optimal value of the current model.

② In the simulation experiment, model is constructed based on the number of trimmings corresponding to the maximum accuracy among the 1000 trimmings determined by the overall training set and the overall test set.

Next, consider parameters such as the number of RBMs, the number of nodes in each hidden layer of the DBN, and the

TABLE 6. Comparative experiments to determine the number of RBMs.

Limit parameters for each round of experiments	
Number of iterations of each layer of RBM in pre-training	Number of DBN nodes in hidden layers
7	110

Comparative experiment of constant number of RBMs					
number of RBMs	Accuracy	number of RBMs	Accuracy	number of RBMs	Accuracy
1	78.5 %	5	81.3 %	9	68.8 %
2	77.9 %	6	78.9 %	10	64.5 %
3	79.1%	7	75.0 %		
4	79.9%	8	71.7 %		

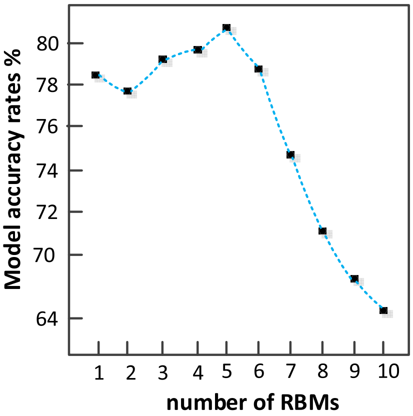

FIGURE 4. Model accuracy rates for different numbers of RBMs.

number of iterations of the RBM in each layer in pre-training. By using all the training data and all the test data in the NSL-KDD data set, the following comparative experiments were carried out to determine the relatively better parameter values. (Several sets of representative data that can explain the influence trend of the parameters are listed below, indicating the correctness and scientific of the better parameter selection):

1) Comparative experiment on determining the number of RBMs

In this part, the accuracy of the DBN detection model is used as the evaluation index. A comparative experiment is performed by limiting the way that other parameters remain unchanged and only the number of RBMs is changed. The influence of the number of RBMs on the performance of the model was analyzed. Finally, the number of better RBMs is determined. Considering that the number of RBMs is too large, the model will be too complicated and the model performance will be affected. Therefore, the number of RBMs is limited to 10 (DBN depth is 11) for comparison experiments. The experimental data is as follows:

As can be seen from the above figure, when the number of RBMs is increased to 5, the accuracy of the detection model has been greatly improved. But as the number continues to increase, the accuracy rate gradually decreases. Therefore, it can be shown that the influence of the number of RBMs on the accuracy has a certain range limit. Within the range, increasing the number will improve the feature extraction capability of the model and optimize the model performance. Beyond the range, the model structure is too complicated, resulting in poor model learning and training results. Then the overall performance of the algorithm is low. Therefore, based on the comprehensive experimental results and the complexity of the model structure, five RBMs are selected to form the DBN model.

2) Comparative experiment on determining the number of nodes in each hidden layer of DBN

This part is based on the number of RBMs determined by the above experiments, and uses the accuracy of the detection model as an evaluation index. By limiting other parameters to be unchanged and only changing the number of nodes in each hidden layer, a comparative experiment is performed. The optimal number of hidden nodes is determined. The value scheme for the number of nodes in each hidden layer is as follows:

① The number of nodes in each hidden layer is taken from the array: [110, 100, 90, 80, 70, 60, 50, 40, 30].

② The number of hidden nodes 1> the number of hidden nodes 2> the number of hidden nodes 3> the number of hidden nodes 4> the number of hidden nodes 5.

③ The number of nodes in the first hidden layer must be >= 50.

④ The number of nodes in the fifth hidden layer should be <= 60.

Some representative experimental data are given below

A total of 125 experimental data were generated according to the above experimental scheme. Due to space limitations, this section only lists all cases when the number of nodes in the first hidden layer is 110. As can be seen from the above table, the value of the number of nodes in each hidden layer of the DBN is very difficult to predict. And through analysis of all experimental data, it can be obtained that when the DBN model structure value is [122,110,100,90,70,30,5], the model accuracy can reach a relatively high value. Therefore, combining the experimental results and the complexity of the model structure, the DBN structure selected in this paper is: [122,110,100,90,70,30,5].

TABLE 7. Comparative experiments to determine the number of nodes in each hidden layer.

Limit parameters for each round of experiments					
Number of iterations of each layer of RBM in pre-training			Number of RBMs		
7			5		
Comparative experiment to determine the number of nodes in each hidden layer					
DBN structure	Accuracy	DBN structure	Accuracy	DBN structure	Accuracy
122,110,100,90,80,60,5	78.7%	122,110,100,80,40,30,5	82.1%	122,110,90,70,60,30,5	81.7%
122,110,100,90,80,50,5	78.7%	122,110,100,70,60,50,5	80.7%	122,110,90,70,50,40,5	80.1%
122,110,100,90,80,40,5	79.3%	122,110,100,70,60,40,5	77.6%	122,110,90,70,50,30,5	79.2%
122,110,100,90,80,30,5	77.3%	122,110,100,70,60,30,5	80.0%	122,110,90,70,40,30,5	80.3%
122,110,100,90,70,60,5	78.2%	122,110,100,70,50,40,5	79.2%	122,110,90,60,50,40,5	81.1%
122,110,100,90,70,50,5	79.1%	122,110,100,70,50,30,5	80.4%	122,110,90,60,50,30,5	78.0%
122,110,100,90,70,40,5	79.1%	122,110,100,70,40,30,5	78.6%	122,110,90,60,40,30,5	77.7%
122,110,100,90,70,30,5	82.3%	122,110,100,60,50,40,5	78.8%	122,110,90,50,40,30,5	78.5%
122,110,100,90,60,50,5	79.3%	122,110,100,60,50,30,5	79.3%	122,110,80,70,60,50,5	80.6%
122,110,100,90,60,40,5	79.3%	122,110,100,60,40,30,5	80.8%	122,110,80,70,60,40,5	79.1%
122,110,100,90,60,30,5	79.4%	122,110,100,50,40,30,5	80.6%	122,110,80,70,60,30,5	78.6%
122,110,100,90,50,40,5	78.8 %	122,110,90,80,70,60,5	80.7%	122,110,80,70,50,40,5	77.4 %
122,110,100,90,50,30,5	79.9%	122,110,90,80,70,50,5	79.5%	122,110,80,70,50,30,5	80.8%
122,110,100,90,40,30,5	79.8%	122,110,90,80,70,40,5	79.5%	122,110,80,70,40,30,5	80.2%
122,110,100,80,70,60,5	79.0%	122,110,90,80,70,30,5	81.0%	122,110,80,60,50,40,5	78.4%
122,110,100,80,70,50,5	78.6%	122,110,90,80,60,50,5	79.6%	122,110,80,60,50,30,5	79.6%
122,110,100,80,70,40,5	78.0%	122,110,90,80,60,40,5	78.6%	122,110,80,60,40,30,5	77.7%
122,110,100,80,70,30,5	78.5%	122,110,90,80,60,30,5	78.6%	122,110,80,50,40,30,5	79.7%
122,110,100,80,60,50,5	79.5%	122,110,90,80,50,40,5	80.6%	122,110,70,60,50,40,5	80.0%
122,110,100,80,60,40,5	78.9%	122,110,90,80,50,30,5	79.3%	122,110,70,60,50,30,5	80.5%
122,110,100,80,60,30,5	81.0%	122,110,90,80,40,30,5	80.8%	122,110,70,60,40,30,5	80.8%
122,110,100,80,50,40,5	81.4%	122,110,90,70,60,50,5	80.7%	122,110,70,50,40,30,5	81.1%
122,110,100,80,50,30,5	78.0%	122,110,90,70,60,40,5	81.8%	122,110,60,50,40,30,5	78.4%

TABLE 8. Comparative experiments to determine the number of RBM iterations of each layer in pre-training.

Limit parameters for each round of experiments					
Number of RBMs			Number of DBN hidden nodes		
5			110,100,90,70,30		
Comparative experiment to determine the number of RBM iterations of each layer in pre-training					
Number of iterations	Accuracy	Number of iterations	Accuracy	Number of iterations	Accuracy
1	77.6%	11	81.2%	21	77.7%
2	78.2%	12	79.8%	22	78.3%
3	79.8%	13	78.9%	23	78.0%
4	78.9%	14	81.4%	24	79.6%
5	80.3%	15	78.2%	25	78.6%
6	81.2%	16	76.8%	26	80.4%
7	82.3%	17	80.5%	27	78.5%
8	81.5%	18	78.7%	28	79.8%
9	79.0%	19	79.6%	29	78.9%
10	79.3%	20	81.1%	30	77.6%

3) Comparative experiment on determining the number of iterations of each layer of RBM in pre-training

This part is based on the number of RBMs and the number of hidden nodes in each DBN based on the above experiments. And the accuracy of the detection model is used as the evaluation index. By limiting the way that other parameters are unchanged and only changing the number of RBM iterations, a comparative experiment is performed to determine the better number of iterations. In the experiment, in the range [1,120], an integer is sequentially selected as the number of iterations of the RBM for comparison experiments. Finally, a total of 120 sets of experimental data were generated. Due to

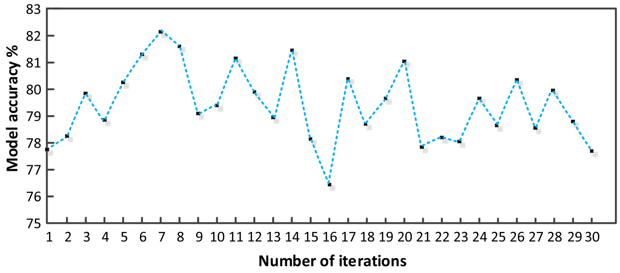

FIGURE 5. Accuracy of RBM iterations in pre-training.

space limitations, only the first 30 sets of experimental data are given below:

As can be seen from the figure above, the value of the number of RBM iterations also has great unpredictability. The number of RBM iterations has a great impact on the accuracy of the model. And through analysis of all 120 sets of experimental data, it can be obtained that when the number of RBM iterations is 7, the accuracy of the detection model can reach a relatively high value. Therefore, based on the experimental results and the consideration of the model running time, the number of iterations of each layer of RBM in the pre-training is set to 7.

Comprehensive comparative analysis of the above experiments, and consider factors such as model accuracy, structural complexity, and running time. In this paper, the parameters of the number of RBMs, the number of hidden nodes in the DBN, and the number of iterations of the RBM in the pre-training are selected as follows:

TABLE 9. Optimal values of key parameters.

RBM number	Number of DBN hidden nodes	Number of iterations of each layer of RBM in pre-training
5	110,100,90,70,30	7

So far, this section has determined all the parameter information required for the implementation of the DBN model. Then, a detection model can be constructed based on the parameter information. And analyze it by simulation experiment.

2) Simulation experiment analysis

Based on model parameter setting experiments, this section determines the model structure and weight coefficients to build the corresponding intrusion detection data processing model. A comparative analysis experiment was constructed through the NSL-KDD data set.

From the introduction of the data set, the four types of attacks in DOS, Probing, R2L, and U2R in NSL-KDD can be divided into 39 specific attack subclasses (the training set contains 22 types, and the other 17 types only appear in the test set). That is, the test set contains 17 unknown intrusions. Based on this, this part trains each intrusion detection data processing model with the entire training set. The false positive rate of each model, the accuracy rate of the entire test set, and the accuracy rate of unknown intrusions in the test set are used as evaluation indicators to achieve the purpose of evaluating model performance. The specific experimental data are as follows:

TABLE 10. Model simulation experiments.

Model structure	DBN	PCA-BP
False alarm rate	4.7%	30.6%
Accuracy for the entire test set	82.3%	63.8%
Accuracy against unknown intrusions in the test set	35.4%	6.8%

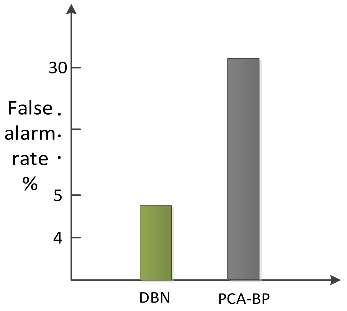

FIGURE 6. False alarm rate of each model.

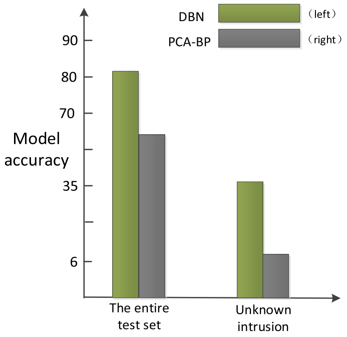

FIGURE 7. Accuracy of each model.

As can be seen from the above figure, compared with the PCA-BP-based intrusion detection data processing model, the DBN-based intrusion detection data processing model has obvious advantages both in terms of false positives and accuracy. Especially in terms of the false positive rate and the accuracy rate of unknown intrusions in the test set, the DBN detection model showed better detection performance, which increased by about 25.9% and 28.6% respectively.

Therefore, deep belief networks have unique advantages in feature learning. DBN can be applied to the field of intrusion detection data processing. Thus, better feature extraction is performed on the current complex network data. This further improves the performance of the intrusion detection data processing model.

V. CONCLUSION

Due to the high dimensionality and redundancy of current network data, feature learning has become a necessary process for current intrusion detection data processing models. However, the traditional feature learning methods have certain shortcomings. And the advent of deep learning has brought new directions to feature learning. In order to verify the unique advantages of deep learning-related technologies in feature learning, this paper constructs a PCA-BP-based intrusion detection data processing model and a DBN-based intrusion detection data processing model. By comparing the PCA-BP detection model and the DBN detection model with the NSL-KDD data set, a comparative experiment of DBN and PCA was constructed. This paper validates the unique advantages and good performance of DBN in feature learning.

REFERENCES

[1] S. Wan, Y. Xia, L. Qi, Y.-H. Yang, and M. Atiquzzaman, ''Automated colorization of a grayscale image with seed points propagation,'' *IEEE Trans. Multimedia*, early access, Feb. 28, 2020, doi: 10.1109/TMM.2020.2976573.

[2] Q. Li, Y. Tian, Q. Wu, Q. Cao, H. Shen, and H. Long, ''A Cloud-Fog-Edge closed-loop feedback security risk prediction method,'' *IEEE Access*, vol. 8, pp. 29004–29020, 2020.

[3] Q. Li, Y. Wang, Z. Pu, S. Wang, and W. Zhang, ''Time series association state analysis method for attacks on the smart Internet of electric vehicle charging network,'' *Transp. Res. Rec., J. Transp. Res. Board*, vol. 2673, no. 4, pp. 217–228, Apr. 2019.

[4] S. Wan and S. Goudos, ''Faster R-CNN for multi-class fruit detection using a robotic vision system,'' *Comput. Netw.*, vol. 168, Feb. 2020, Art. no. 107036.

[5] R. Zhang, P. Xie, C. Wang, G. Liu, and S. Wan, ''Classifying transportation mode and speed from trajectory data via deep multi-scale learning,'' *Comput. Netw.*, vol. 162, Oct. 2019, Art. no. 106861.

[6] Z. Gao, Y. Li, and S. Wan, ''Exploring deep learning for view-based 3D model retrieval,'' *ACM Trans. Multimedia Comput., Commun., Appl.*, vol. 16, no. 1, pp. 1–21, Apr. 2020.

[7] Q. Li, S. Meng, S. Zhang, M. Wu, J. Zhang, M. Taleby Ahvanooey, and M. S. Aslam, ''Safety risk monitoring of cyber-physical power systems based on ensemble learning algorithm,'' *IEEE Access*, vol. 7, pp. 24788–24805, 2019.

[8] S. Ding, S. Qu, Y. Xi, and S. Wan, ''Stimulus-driven and concept-driven analysis for image caption generation,'' *Neurocomputing*, Jul. 2019, doi: 10.1016/j.neucom.2019.04.095.

[9] J. Hou, Q. Li, R. Tan, S. Meng, H. Zhang, and S. Zhang, ''An intrusion tracking watermarking scheme,'' *IEEE Access*, vol. 7, pp. 141438–141455, 2019, doi: 10.1109/ACCESS.2019.2943493.

[10] J. Hou, Q. Li, and S. Cui, ''Low-cohesion differential privacy protection for industrial Internet,'' *The J. Supercomputing.*, vol. 7, pp. 1–23, Jan. 2020, doi: 10.1007/s11227-019-03122-y.

[11] Z. Gao, H.-Z. Xuan, H. Zhang, S. Wan, and K.-K.-R. Choo, ''Adaptive fusion and category-level dictionary learning model for multiview human action recognition,'' *IEEE Internet Things J.*, vol. 6, no. 6, pp. 9280–9293, Dec. 2019.

[12] S. Wan, Z. Gu, and Q. Ni, ''Cognitive computing and wireless communications on the edge for healthcare service robots,'' *Comput. Commun.*, vol. 149, pp. 99–106, Jan. 2020.

[13] Q. Li, S. Meng, S. Wang, J. Zhang, and J. Hou, ''CAD: Command-level anomaly detection for vehicle-road collaborative charging network,'' *IEEE Access*, vol. 7, pp. 34910–34924, 2019.

[14] Z. Gao, H. Xue, and S. Wan, ''Multiple discrimination and pairwise CNN for view-based 3D object retrieval,'' *Neural Netw.*, vol. 125, pp. 290–302, May 2020.

[15] Y. Zhao, H. Li, S. Wan, A. Sekuboyina, X. Hu, G. Tetteh, M. Piraud, and B. Menze, ''Knowledge-aided convolutional neural network for small organ segmentation,'' *IEEE J. Biomed. Health Inform.*, vol. 23, no. 4, pp. 1363–1373, Jul. 2019.

[16] Q. Li, S. Meng, S. Zhang, J. Hou, and L. Qi, ''Complex attack linkage decision-making in edge computing networks,'' *IEEE Access*, vol. 7, pp. 12058–12072, 2019.

[17] J. Hou, Q. Li, S. Meng, Z. Ni, Y. Chen, and Y. Liu, ''DPRF: A differential privacy protection random forest,'' *IEEE Access*, vol. 7, pp. 130707–130720, 2019, doi: 10.1109/ACCESS.2019.2939891.

[18] S. Wang, Q. Li, J. Hou, S. Meng, B. Zhang, and C. Zhou, ''Active defense by mimic association transmission in edge computing,'' *Mobile Netw. Appl.*, vol. 25, no. 2, pp. 725–742, Apr. 2020, doi: 10.1007/s11036-019-01446-w.

[19] A. R. Rajput, Q. Li, M. Taleby Ahvanooey, and I. Masood, ''EACMS: Emergency access control management system for personal health record based on blockchain,'' *IEEE Access*, vol. 7, pp. 84304–84317, 2019.

[20] Q. Li, Y. Song, J. Zhang, and V. S. Sheng, ''Multiclass imbalanced learning with one-versus-one decomposition and spectral clustering,'' *Expert Syst. Appl.*, vol. 147, Jun. 2020, Art. no. 113152, doi: 10.1016/j.eswa.2019.113152.

[21] Q. Li, J. Hou, S. Meng, and H. Long, ''GLIDE: A game theory and data-driven mimicking linkage intrusion detection for edge computing networks,'' *Complexity*, vol. 2020, pp. 1–18, Mar. 2020, doi: 10.1155/2020/7136160.

[22] H. Gao, C. Liu, Y. Li, and X. Yang, ''V2 VR: Reliable hybrid-network-oriented V2 V data transmission and routing considering RSUs and connectivity probability,'' *IEEE Trans. Intell. Transp. Syst.*, early access, Apr. 13, 2020, doi: 10.1109/TITS.2020.2983835.

[23] J. Yu, C. Zhu, J. Zhang, Q. Huang, and D. Tao, ''Spatial pyramid-enhanced NetVLAD with weighted triplet loss for place recognition,'' *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 2, pp. 661–674, Feb. 2020, doi: 10.1109/TNNLS.2019.2908982.

[24] S. Deng, Z. Xiang, P. Zhao, J. Taheri, H. Gao, J. Yin, and A. Y. Zomaya, ''Dynamical resource allocation in edge for trustable IoT systems: A reinforcement learning method,'' *IEEE Trans. Ind. Informat.*, early access, Feb. 18, 2020, doi: 10.1109/TII.2020.2974875.

[25] J. Yu, M. Tan, H. Zhang, D. Tao, and Y. Rui, ''Hierarchical deep click feature prediction for fine-grained image recognition,'' *IEEE Trans. Pattern Anal. Mach. Intell.*, early access, Jul. 30, 2019, doi: 10.1109/TPAMI.2019.2932058.

[26] Y. Yin, J. Xia, Y. Li, Y. Xu, W. Xu, and L. Yu, ''Group-wise itinerary planning in temporary mobile social network,'' *IEEE Access*, vol. 7, pp. 83682–83693, 2019.

[27] H. Gao, Y. Duan, L. Shao, and X. Sun, ''Transformation-based processing of typed resources for multimedia sources in the IoT environment,'' *Wireless Netw.*, vol. 2019, pp. 1–7, Nov. 2019, doi: 10.1007/s11276-019-02200-6.

[28] L. Kuang, T. Gong, S. OuYang, H. Gao, and S. Deng, ''Offloading decision methods for multiple users with structured tasks in edge computing for smart cities,'' *Future Gener. Comput. Syst.*, vol. 105, pp. 717–729, Apr. 2020, doi: 10.1016/j.future.2019.12.039.

[29] H. Gao, Y. Xu, Y. Yin, W. Zhang, R. Li, and X. Wang, ''Context-aware QoS prediction with neural collaborative filtering for Internet-of-Things services,'' *IEEE Internet Things J.*, early access, Dec. 2, 2019, doi: 10.1109/JIOT.2019.2956827.

[30] Y. Zhu, W. Zhang, Y. Chen, and H. Gao, ''A novel approach to workload prediction using attention-based LSTM encoder-decoder network in cloud environment,'' *EURASIP J. Wireless Commun. Netw.*, vol. 2019, no. 1, Dec. 2019, Art. no. 274, doi: 10.1186/s13638-019-1605-z.

[31] J. Yu, J. Li, Z. Yu, and Q. Huang, ''Multimodal transformer with multi-view visual representation for image captioning,'' *IEEE Trans. Circuits Syst. Video Technol.*, early access, Oct. 15, 2019, doi: 10.1109/TCSVT.2019.2947482.

[32] Y. Yin, L. Chen, Y. Xu, J. Wan, H. Zhang, and Z. Mai, ''QoS prediction for service recommendation with deep feature learning in edge computing environment,'' *Mobile Netw. Appl.*, vol. 25, no. 2, pp. 391–401, Apr. 2020, doi: 10.1007/s11036-019-01241-7.

[33] J. Yu, Z. Kuang, B. Zhang, W. Zhang, D. Lin, and J. Fan, ''Leveraging content sensitiveness and user trustworthiness to recommend fine-grained privacy settings for social image sharing,'' *IEEE Trans. Inf. Forensics Security*, vol. 13, no. 5, pp. 1317–1332, May 2018.

TAO DUAN received the master's degree from the Beijing Institute of Technology, China, in 2010. He is currently the CIO of RT Tourism Development Group Company, Ltd. His research interests include the industrial Internet, information security, and artificial intelligence.

YOUHUI TIAN received the master's degree from the Heilongjiang University of Science and Technology. He is currently a Senior Engineer with the Jiangsu Vocational Institute of Commerce. His research area is security communication and information systems.

HANRUI ZHANG was born in Nanjing, Jiangsu, China, in 1995. She is currently pursuing the master's degree with the School of Computer Science and Engineering, Nanjing University of Science and Technology, China. Her major is computer applications technology. She has received several school scholarships during her school years. She has published articles in international journals and international conferences. Her research interests include data mining, information security, and cloud computing.

YAOZONG LIU received the Ph.D. degree from the Nanjing University of Science and Technology, China, in 2016. He is currently a Lecturer with the Department of Intelligent Manufacturing, Wuyi University. His research interests include information security and big data mining.

QIANMU LI received the B.Sc. and Ph.D. degrees from the Nanjing University of Science and Technology, China, in 2001 and 2005, respectively. He is currently a Full Professor with the School of Cyber Science and Engineering, Nanjing University of Science and Technology. His research interests include information security and data mining. He received the China Network and Information Security Outstanding Talent Award, in 2016, and the Education Ministry Science and Technology Awards, in 2012.

JIAN JIANG received the master's degree from Fudan University, China, in 2016. He is currently the Technical Director of Jiangsu Zhongtian Internet Technology Company, Ltd. His research interests include the industrial Internet, big data, cloud computing, and intelligent manufacturing.

ZONGSHENG SHI received the master's degree from the Nanjing University of Aeronautics and Astronautics, China, in 2013. He is currently the CIO of Zhongtian Technology Group Company, Ltd., and a General Manager of Jiangsu Zhongtian Internet Technology Company, Ltd. His research interests include the the industrial Internet, big data, cloud computing, and intelligent manufacturing.