IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

+Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2017.Doi Number

Network Intrusion Detection Combined Hybrid Sampling with Deep Hierarchical Network

Kaiyuan Jiang[1], Wenya Wang[1], Aili Wang[1] and Haibin Wu[1]*

[1]The Higher Educational Key Lab for Measuring & Control Technology and Instrumentations of Heilongjiang, Harbin University of Science and Technology, Harbin, 150080, China

Corresponding author: Haibin Wu (e-mail: woo@hrbust.edu.cn).

This research was funded by National Natural Science Foundation of China (NSFC-61671190).

ABSTRACT Intrusion detection system (IDS) plays an important role in network security by discovering and preventing malicious activities. Due to the complex and time-varying network environment, the network intrusion samples are submerged into a large number of normal samples, which leads to insufficient samples for model training and detection results with a high false detection rate. According to the problem of data imbalance, we propose a network intrusion detection algorithm combined hybrid sampling with deep hierarchical network. Firstly, we use the one-side selection (OSS) to reduce the noise samples in majority category, and then increase the minority samples by Synthetic Minority Over-sampling Technique (SMOTE). In this way, a balanced dataset can be established to make the model fully learn the features of minority samples and greatly reduce the model training time. Secondly, we use convolution neural network (CNN) to extract spatial features and Bi-directional long short-term memory (BiLSTM) to extract temporal features, which forms a deep hierarchical network model. The proposed network intrusion detection algorithm was verified by experiments on the NSL-KDD and UNSW-NB15 dataset, and the classification accuracy can achieve 83.58% and 77.16%, respectively.

INDEX TERMS Network intrusion detection, hybrid sampling, deep hierarchical network, convolution neural network, Bi-directional long short-term memory.

I. INTRODUCTION

With the continuous development of Internet technology, it has almost become an indispensable tool for people's daily life and greatly changed people's lifestyle. But, in today's network environment, various network attack methods are constantly updated, the scale of impact is increasing, the frequency of attacks is increasing, and network security issues are becoming increasingly serious. The task of network intrusion detection is to discover suspicious attacks take corresponding measures to protect the network from sustaining attacks and reduce economic losses. Network intrusion detection has become an important research content in the field of network security [1]-[2].

At present, the commonly used intrusion detection technologies are generally divided into misuse detection and anomaly detection, but these two methods have the disadvantages of low detection rate and high false positive rate [3]-[4]. With the application of artificial intelligence in Intrusion Detection System (IDS) [5], artificial intelligence-based detection methods have become a hotspot in the research of intrusion detection systems. However, there are several major challenges in the design and implementation of the IDS:

(1) When dealing with large-scale, high-dimensional data points, traditional network intrusion detection approaches tend to apply dimension reduction to remove noise in measurements. Consequently, it is likely to remove significant information when extracting features for intrusion behaviors. This may cause high false detection rate.

(2) The imbalanced data is a generic problem for building an network intrusion detection model based on deep learning. The imbalanced data will affect the performance of the model, leading to high false alarm rate and high false miss rate of some minority classes samples.

(3) The characteristics of network traffic data are very complicated, which make it difficult to extract features. If the

data features can't be fully extracted, the classification results will be poor.

To build an efficient intrusion detection system, researchers use machine learning to detect various types of attacks [6]. Ünal Çavuşoğlu proposed that layered architecture is created by determining appropriate machine learning algorithms according to attack type [7]. The most widely studied algorithms are Support Vector Machine (SVM), Naive Bayes, Random Forest (RF) and other clustering algorithms. Zhao proposed the least squares support vector machine (LSSVM) algorithm based on hybrid kernel function, and each parameter of LSSVM is optimized using particle swarm optimization (PSO) algorithm [8]. Thaseen proposed to use Chi-square feature selection and SVM, Modified Naïve Bayes (MNB) and LPBoost integration to build an intrusion detection model. And the prediction of the class label was decided by a majority voting of SVM, MNB and LPBoost [9]. Sumaiya proposed an intrusion detection model using chi-square feature selection and multi class support vector machine which could get a better detection rate and reduced false alarm rate [10]. Tao proposed the feature selection, weight, and parameter optimization of support vector machine based on the genetic algorithm (FWP-SVM-GA) [11]. Compared with other SVM based intrusion detection algorithms, the detection rate is higher and the false positive and false negative rates are lower. Peng proposed a clustering method for IDS based on Mini Batch K-means combined with Principal Component Analysis (PCA), and the clustering method can be used for IDS over big data environment [12]. Farnaaz built a model for intrusion detection system using RF classifier, the model was efficient with low false alarm rate and high detection rate [13].

Traditional machine learning algorithms belong to shallow learning, but with the continuous expansion of network data, a large amount of non-linear network data brings new challenges to intrusion detection. For high-dimensional data, it is often used to extract features efficiently by reducing the data dimension, and got significant result. Zhang proposed an ensemble manifold regularized sparse low-rank approximation (EMR-SLRA) algorithm for multiview feature embedding[14]. By incorporating the discriminant analysis and the local neighborhood relationship of the original samples into the low-rank representation, Liu proposed a novel discriminative low-rank preserving projection (DLRPP) algorithm for dimensionality reduction [15]. A novel structured optimal graph based sparse feature extraction (SOGSFE) method for semi-supervised learning was proposed by Liu et al. [16].

Deep learning (DL) is widely used in various fields and achieved good results [17]. In order to improve the generalization of the neural network, Wang proposed a progressive learning strategy [18]. Alom et al. selected 40% of the training data from the NSL-KDD dataset and use Deep Belief Nets (DBN) and SVM to achieve higher test accuracy [19]. Fiore used the Discriminative Restricted Boltzmann Machine to combine the expressive power of generative models with good classification accuracy capabilities to infer knowledge from incomplete training data [20]. Yin proposed a deep learning approach for intrusion detection using RNN, and studied the performance of the model in binary classification and multiclass classification, and the number of neurons and different learning rate impacts on the performance of the proposed model [21]. Kim applied Long Short-Term Memory (LSTM) architecture to RNN and they confirmed that the deep learning approach was effective for IDS through the performance test [22]. Convolutional neural networks, with their local perception and weight sharing, can greatly reduce the number of parameters, and have been widely used in network intrusion detection research. Reference [23]-[25] used CNN to complete network intrusion detection, which can quickly identify various types of attacks.

These studies prompt us to use deep neural networks to learn the hierarchical features of network traffic (that is, the spatial and temporal features) for classifying network traffic [26]. However, due to the serious imbalance between the network intrusion traffic, the proportion of various traffic data varies greatly, and most detection methods aim to reduce the overall average rate of false positives, which will lead to the increase of the rate of false positives in minority samples. In the real network environment, the attacks of minority attacks are more dangerous than those of majority attacks. To solve these issues, Bamakan developed a precise, sparse and robust methodology for multi-class intrusion detection problem based on the Ramp Loss K-Support Vector Classification-Regression(K-SVCR), which was used to solve the highly imbalanced and skewed attack distribution of the data [27]. Zhang proposed a multiple-layer representation learning model for accurate end-to-end network intrusion detection by combining deep convolutional neural networks with Multi-grained Cascade Forest (gcForest), which achieved accurate detection on imbalanced data and small-scale data with fewer hyperparameters compared to most existing deep learning methods [28]. Wang further improved the resampling strategy inside Oversampling based Online Bagging (OOB) and Undersampling based Online Bagging (UOB), and looked into their performance in both static and dynamic data streams [29]. Yan proposed a new traffic classification method for imbalanced network data, by introducing and improving Synthetic Minority Over-sampling Technique

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.2973730, IEEE Access

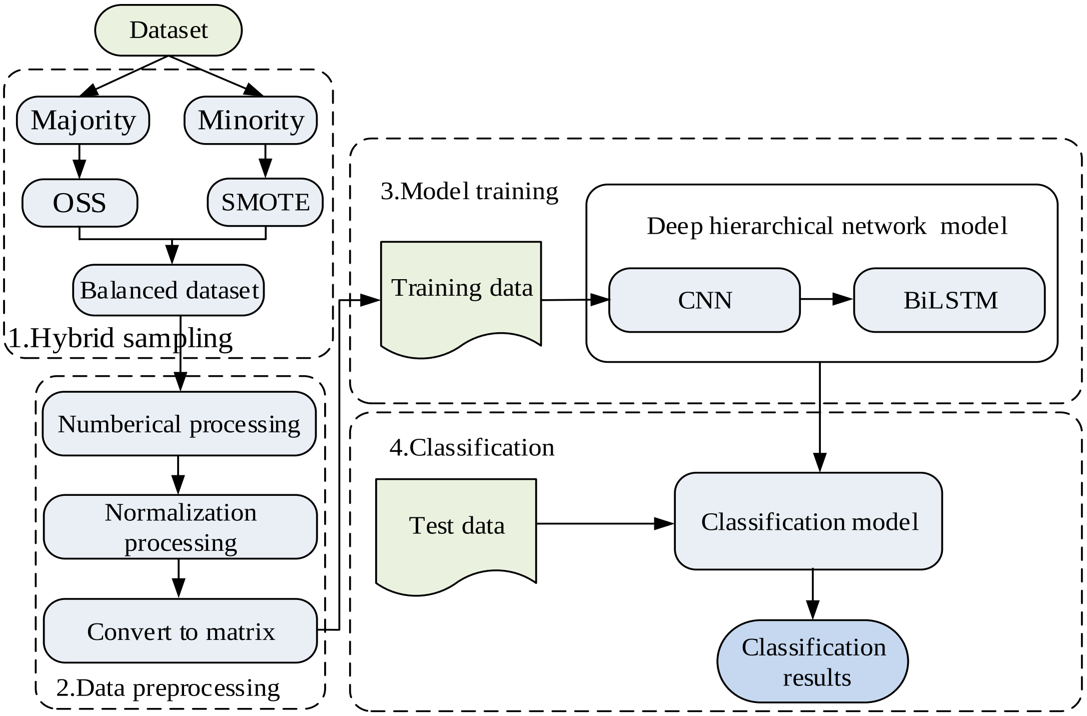


FIGURE 1. Overall framework of network intrusion detection model

(SMOTE) algorithm, a Mean SMOTE (M-SMOTE) algorithm was proposed to realize the balance of traffic data [30].

In this paper, we proposed a novel network intrusion detection algorithm combined hybrid sampling with deep hierarchical network to improve the detection accuracy. The hybrid sampling include two parts. Firstly, the one-side selection (OSS) algorithm is used to remove noise samples in the majority class. Secondly, to alleviate the imbalance of network traffic data, we employ the SMOTE to generate the minority class samples. Thus, imbalanced data can transformed into balanced data for the following classification.

For the complication of data features, we constructed the deep hierarchical network, which integrates the CNN and Bi-directional LSTM (BiLSTM), while learning the spatial and temporal feature of traffic network data. CNN has been gradually applied to network detection because of its excellent performance in automatic feature extraction. Given the fact that the attack samples are usually intra-dependent on time sequence data, we apply BiLSTM to automatically learn the effective features of time sequence measurements.

The two contributions of this paper are summarized as below:

(1) The paper proposed a hybrid sampling method in order to solve the problem of data imbalance. We use the one-side selection (OSS) to reduce the majority samples and use the Synthetic Minority Over-sampling Technique (SMOTE) to increase the minority samples. In this way, a balanced dataset is built for model training. At the same time, the training time of the model is also reduced.

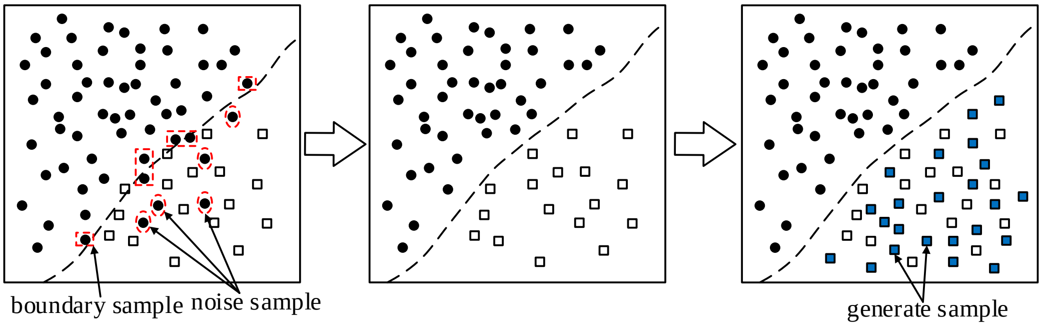
(2) We proposed the deep hierarchical network model which combined CNN with BiLSTM. The features of the data are accurately extracted by this method. After training, we got a model with good classification performance.

The structure of this paper is as follows. Section II describes some of the related work of network intrusion detection. Section III details the classification detection method proposed in this paper. In section IV, we describe the dataset used in this paper and show the experimental results and analysis performed on the dataset. In section V, conclusions of this article are given.

II. THE PROPOSEED INTRUSION DETECTION MODEL

The overall flow of the network intrusion detection model in this paper is shown in Figure 1. Aiming at the imbalance of the network traffic data and the complexity of the features, the original data is firstly subjected to hybrid sampling processing to obtain balanced data, and then the data is subjected to numerical normalization and other preprocessing. At last a deep hierarchical network model is used for classification. The details of the proposed network intrusion detection model include four parts as follows:

(1) The imbalance of network traffic data has an impact on the performance of the classification model. Therefore, this article uses the OSS down sampling method to remove the noise while reducing the majority samples, and then uses the



(a)original data distribution (b)data distribution after OSS sampling (c)data distribution after SMOTE

FIGURE 2. The process of hybrid sampling

SMOTE algorithm to increase the minority samples. A balanced data set is constructed by combining these two methods together.

(2) The symbolic feature attributes in the data set are digitized and normalized to obtain a standardized data set, and then each network data is turned into two-dimensional matrix, so that it can be conformed into the input format of CNN model.

(3) Because of the complexity of the characteristics of network traffic data, the deep hierarchical network model constructed by CNN and BiLSTM are used to extract the spatial and temporal features of the data to improve the accuracy of classification.

(4) After training, a model with good classification performance is obtained, and the model is used to classify the test set to obtain excellent classification results.

A. *CONSTRUCTION OF BALANCED DATA SET BASED ON HYBRID SAMPLING*

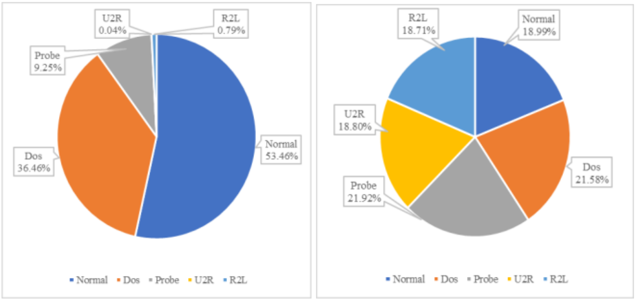
Network traffic data is composed of a large amount of normal traffic and a small amount of abnormal traffic, which is a typical imbalanced data classification problem. In this case, when the overall error is minimized, although the prediction accuracy of some majority classes is improved, the prediction accuracy of minority classes is often very low. Currently, there are two commonly used sampling techniques: random under-sampling (RUS) and random over-sampling (ROS). In the network intrusion detection, the imbalanced ratio (IR) of various traffic data is extremely high. Only using the RUS method may lose the samples with important information. However, only using ROS method will make the classifier unable to learn comprehensive information, which will lead to unstable classification performance.

One-side selection is an under-sampling method resulting from the application of Tomek links followed by the application of KNN [31]. Tomek links are used as an under-sampling method and remove noisy and borderline majority class examples. Borderline examples can be considered "unsafe" since a small amount of noise can make them fall on the wrong side of the decision border. KNN aims to remove samples from the majority class that are distant from the decision border. The remainder samples are "safe" majority class samples and all minority class samples.

SMOTE is an over-sampling method. Its main idea is to form new minority class samples by interpolating between several minority class samples that lie together. Thus, the overfitting problem is avoided. At the same time, it causes the decision boundaries for the minority class to spread further into the majority class space.

So, this paper proposes the hybrid sampling method. The OSS method is used to remove the noise while reducing the number of samples in majority class. The SMOTE method is used to increase the number of samples in the minority class. Finally, a balanced data set with no noise and clear classification boundaries is obtained.

In the hybrid sampling process, we use OSS to remove the noise and the samples on the classification boundary, which expand the decision area of the minority class, and obtain a relatively clear classification surface. Then the new IR is S^-/S^+. S^- is the number of majority samples, and S^+ is the number of minority samples. Since our data contains multiple categories, in order to get a balanced data set, we set the composition ratio to 1: 1. In the end, the proportion of each type of sample is approximately close to 1:1. Figure 2 uses the binary classification problem as an example to show the hybrid sampling process.



(a)Before hybrid sampling (b)After hybrid sampling

FIGURE 3. The proportion of various attack types in the training set

```
Algorithm 1. Hybrid sampling algorithm
Input: Training set $S=\{(x_i,y_i),i=1,2,...N,y_i\in\{+,-\}\}$ ;the number of majority class sample is
       $N^-$, the number of minority class sample is $N^+$, sample rate is $SR$
Output: Hybrid sampled training set $S''$
Step1: Take all minority class samples from training set $S$ and a randomly selected majority class
       sample to form training subset $C$
       For $i=1:N$
           Sample $x_i$ from $S$ ,and judge whether $x_i$ is Tomek links sample
           find all Tomek links sample pairs in $C$, and put the majority samples into subset $L$
       under-sampled training set $S'=C-L$
Step2: The number of majority class sample is $S^-$,and the number of minority class sample is $S^+$
       $IR=S^-/S^+$ ,$S^{New}=\varnothing$
       For $i=1:S^+\times SR$
           For each minority class sample $x_i$
               get its K neighbors
               each paired sample $(x_i,x_n)$ generate a new minority class sample according to the
               flowing formula: $x^{new}=x_i+rand(0,1)\times|x_i-x_n|$
                   $S^{New}=S^{New}\bigcup x^{new}$
       Hybrid sampled set $S''=S'\bigcup S^{New}$
```

Through hybrid sampling, the number of the majority class samples are reduced, thus reducing the training time of the detection model. At the same time, the increased number of minority class samples can make the model fully learn the features. Hybrid sampling algorithm is written as Algorithm 1. Taking NSL-KDD training set as an example, before and after hybrid sampling, the proportion of various attack types in the training set is shown in the Figure 3.

B. SPATIAL FEATURES EXTRACTON BY CNN

Convolutional neural networks have made outstanding research results in many fields such as computer vision, speech recognition, and natural language processing [32]. Compared with traditional methods, it can better extract the features of objects automatically. The features extracted by the convolutional layer are used to train the classification model. Considering that the features have spatial locality in different positions, it is necessary to use pooling layers to aggregate the statistics of the features at different positions to a certain degree in order to reduce the data dimension and avoid overfitting problem. Therefore, CNN is suitable for huge network data to extract the spatial features of network traffic data.

Figure 4 is the structure of a convolutional neural network, which contains two convolutional layers and two pooling layers, and one fully connected layer. The process of extracting spatial features using CNN is as follows:

The traffic data matrix is input of the convolution layer. The convolution layer is the core of the CNN architecture. As the local perception concept is introduced, all neurons can share the same convolution kernel, and the number of convolution kernels determines the number of weights. As a result, the number of weights is reduced greatly, and the computational efficiency is increased. The convolution function can be written as:

$$h_j = f(h_{j-1}\otimes w_j + b_j) \tag{1}$$

Where h_j is the feature map of layer j,(j=1,2,…n). b_j is the bias of layer j and $\otimes$ is the convolution function and $f(x)$ is the activation function. This paper uses rectified linear unit (ReLU) as the activation function.

The pooling layer integrates the feature points in the small neighborhood to obtain new features, and it works after the convolution layer. It can reduce the size of feature map h_j and avoid over-fitting. It can be written as:

$$h_j = pool(h_{j-1}) \tag{2}$$

After several convolution and pooling layers, h_j must be reshaped into a vector. Then, output y_i can be achieved through the fully connected layer. So, we can get the spatial features of the network traffic data extracted by CNN.

The use of CNN can accurately extract the spatial features, but it does not perform well in learning sequence correlation information and can't solve the problem of long-term

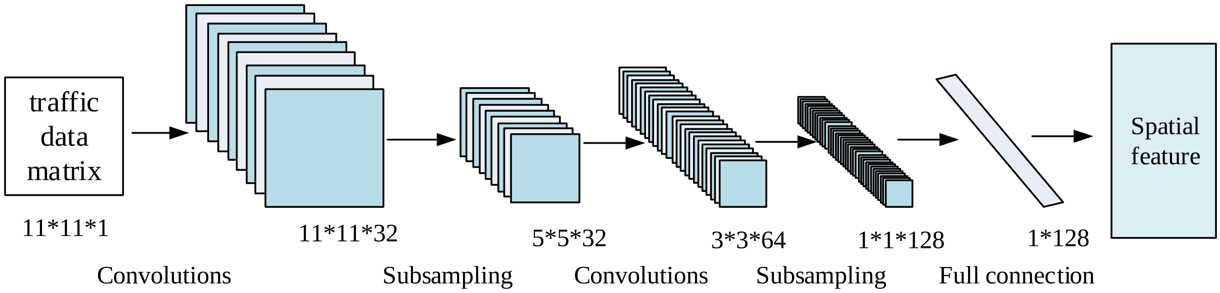


FIGURE 4. CNN network structure model

information dependence. Therefore, the accuracy of network intrusion detection only using CNN needs to be improved.

C TEMPORAL FEATURES EXTRACTON BY BILSTM

RNN is a commonly used model in deep learning [33]. It is widely used in speech processing and has achieved good results in speech recognition and time series processing. In recent sentiment analysis studies, deep neural networks have been used to learn the hierarchical features of natural language and have achieved good results [34]. However, RNN has a gradient disappearing gradient explosion problem, and the LSTM is designed with a memory module that can determine when to keep memory and when to forget certain information. Therefore, LSTM can mine the time series of relatively long intervals and delays in the time series, and can effectively alleviate the problems of gradient disappearance and training difficulties [35]. But LSTM can only read sequence data in one direction, the impact of information after attributes is not fully considered. Therefore, BiLSTM is used instead of LSTM to introduce the following information. The BiLSTM structure is shown in Figure 5.

A typical LSTM network framework consists of an input layer, a hidden loop layer, and an output layer. Different from the traditional recurrent neural network, the cyclic hidden layer is mainly composed of neuron nodes. The basic unit of the LSTM cyclic hidden layer is the memory module. This memory module contains one memory unit and three adaptive multiplication gating units, namely the input gate, output gate and forget gate. The calculation operation of each neuron node in LSTM is as follows:

At time t, he input gate is input according to the output result ht-1 of the cell at the previous moment. The input xt at the current moment determines whether to update the current information into the cell through calculation. W is the weight, and b is the bias of neurons.

$$i_t = sigmoid(W_t \cdot [h_{t-1}, x_t] + b_t) \tag{3}$$

Forget gate based on the last moment hidden layer output h_{t-1} and the current time input is used to decide whether retain or discard the information.

$$f_t = sigmoid(W_f \cdot [h_{t-1}, x_t] + b_f) \tag{4}$$

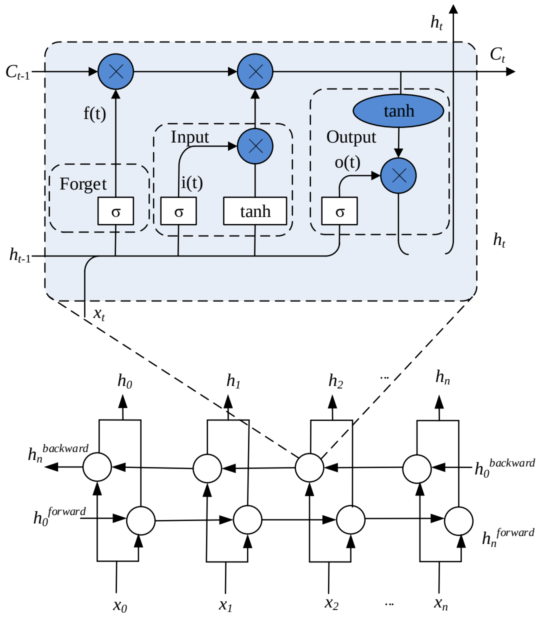


FIGURE 5. BiLSTM network structure model

The current candidate memory cell value is determined by the current input data x_t and the output result h_{t-1} of the LSTM hidden layer cell at the previous moment. In the current moment, the memory cell state value C_t is adjusted by the current candidate cell C_t and its own state C_{t-1}, input gate and forget gate. Character ∗ is the element-wise matrix multiplication.

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \tag{5}$$

$$C_t = f_t * C_{t-1} + i_t * \tilde{C} \tag{6}$$

Calculate the output gate o_t, and the output is used to control the cell status value. The output of last cell is h_t, which can be expressed as (8).

$$o_t = sigmoid(W_o \cdot [h_{t-1}, x_t] + b_o) \tag{7}$$

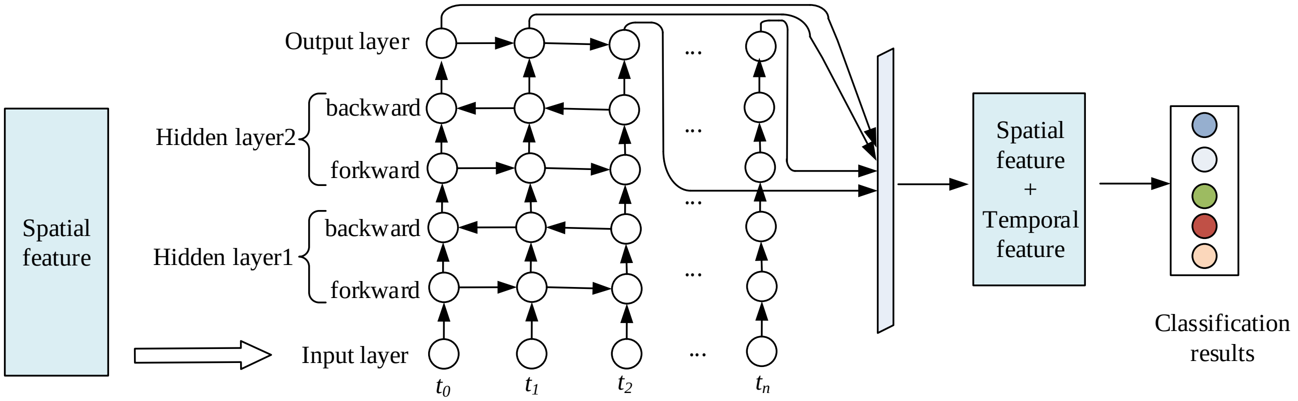


FIGURE 6. Deep hierarchical network structure model

$$h_t = o_t * \tanh(C_t) \tag{8}$$

The BiLSTM is composed of two LSTM networks, one forward LSTM and one backward LSTM. The forward LSTM hidden layer is responsible for the forward feature extraction and the backward one is responsible for backward feature extraction. The use of BiLSTM model can better consider the influence of each attribute before and after in the sequence data. Thus, more comprehensive feature information is obtained. The state of BiLSTM at time *t* includes forward output and backward output.

$$h_t^{forward} = LSTM^{forward}(h_{t-1}, x_t, C_{t-1}) \tag{9}$$

$$h_t^{backward} = LSTM^{backward}(h_{t-1}, x_t, C_{t-1}) \tag{10}$$

$$H_t = [h_t^{forward}, h_t^{backward}] \tag{11}$$

D DEEP HIERARCHICAL NETWORK MODEL

CNN and BiLSTM are both representative in deep learning algorithm. CNN can extract data features in the spatial dimension. BiLSTM has the characteristic of preserving contextual historical information for a long time, and realizes the extraction of data features in the time level. In the feature extraction of network intrusion detection system, it is necessary not only to measure the law of change at time level, but also to consider the feature connection at the spatial level. Therefore, this paper combined CNN with BiLSTM to extract features, and finally constructed a deep hierarchical network model. Its structure is shown in Figure 6.

The spatial and temporal features of traffic data can be extracted simultaneously by building a hierarchical network model, which combined CNN with BiLSTM networks together. Because CNN and BiLSTM network structure inputs have different forms, the extracted spatial features are adjusted at the CNN output to satisfy with the input format of the BiLSTM network. The output of the fully connected layer of the CNN is a 1*128 feature vector. When put to the input layer of the BiLSTM network, the input size is set to 64 and the time step is set to 2. So, the size of the input BiLSTM is consistent with the output size of CNN. Two layers of BiLSTM units perform temporal feature extraction. The first hidden layer uses 128 neurons and the second layer uses 64 neurons. The activation function of each layer uses the Sigmoid function to perform non-linear operations. The result of each recursive operation of the BiLSTM is a fusion of all the previous features and the current features. One fully connected layer is connected after the output layer of the BiLSTM, the previously extracted features are integrated, and the output value of the last fully connected layer is passed to softmax for classification.

III. EXPERIMENT RESULTS AND ANALYSIS

Experiments used the TensorFlow under windows as the backend, encoded with Keras and Python. The simulation system environment is shown in Table1. The learning rate of the network model in this paper was set to 0.001. The weight inactivation rate of Dropout in the regularization method was set to 0.5, and the experiment iteration is 100 times, each batch_size was set to 128.

TABLE 1. Experimental environment

Project	Environment/Version
Operating system	Windows10
CPU	i3-7100U
Memory	12G
Framework	Keras2.2

A. DATA SET DESCRIPTION

We consider the two publicly available intrusion detection datasets that are widely adopted in previous works: NSL-KDD and UNSW-NB15 datasets. In the field of intrusion detection, KDD CUP99 and NSL-KDD are famous data sets [36], and analysis by Revathi shows that NSL-KDD data sets are very suitable for comparing different intrusion detection models [37]. In this data set, each intrusion record has a 42-dimensional feature, which is divided into a 38-dimensional digital feature, a 3-dimensional symbol feature, and a traffic type label. The label mainly contains normal data and 4 types of attack data (Dos, Probe, U2R, R2L). In the experiments of this paper, the training set (KDDTrain +) and test set (KDDTest+) in the NSL-KDD dataset are used as the training set and test set of the model, respectively, and their types and numbers are shown in Table 2.

TABLE 2. Description of NSL-KDD dataset

Category	Train	Test
Normal	67343	9711
Dos	11656	7458
Probe	45927	2421
U2R	52	200
R2L	995	2754
Total	125973	22544

The cyber security research team of Australian Centre for Cyber Security (ACCS) has introduced a new data called as UNSW-NB15 to resolve the issues found in the KDDCup 99 and NSL-KDD datasets. A partition of full connection records which is composed of 175343 train connection records and 82337 test connection records confined with 10 attacks. The partitioned dataset consists of 42 features with their parallel class labels which are normal and nine different attacks. The information regarding simulated attacks category and its detailed statistics are described in Table 3.

TABLE 3. Description of UNSW-NB15 dataset

Category	Train	Test
Normal	56000	37005
Backdoor	1746	583
Analysis	2000	677
Fuzzers	18185	6062
Shellcode	1133	378
Reconnaissance	10492	3496
Exploits	33393	11132
DoS	12264	4089
Worms	130	44
Generic	40000	18871
Total	175343	82337

B. DATA PREPROCESSING

Data preprocessing includes three steps as follows:

1) NUMERICALI PROCESSING

Because the input of the model is a digital matrix, the one-hot encoding method is used to map the data with symbol features in the data set to the digital feature vector. This processing mainly focuses on three features in the data set: protocol_type, service, and flag. They contain 3, 70, and 11 symbol attributes, respectively, and are individually hot-coded. For example, in NSL-KDD dataset, the three attributes of protocol_type, TCP, UDP, and ICMP, are encoded as binary vectors (1,0,0), (0,1,0), and (0,0,1), respectively.

2) NORMALIZATION PROCESSING

In the data set, the value range of continuous feature data is significantly different. For example, in NSL-KDD dataset, the value range of the num_root feature is [0,7468], while the value range of the num_shells feature is only [0,5]. The range of maximum values varies widely. In order to facilitate arithmetic processing and elimination of dimensions, a normalized processing method is adopted to uniformly and linearly map the value range of each feature within the [0,1] interval. The normalized calculation formula is:

$$X_{norm} = \frac{X - X_{\min}}{X_{\max} - X_{\min}} \tag{12}$$

Among them, $X_{\max}$ represents the maximum value of the feature, and $X_{\min}$ represents the minimum value.

3) CONVERTING NORMALIZED DATA INTO MATRIX

Each network record of the read data is dimensionally transformed to conform to the format of the grayscale image. Here, RGB = 1 is the default. In order to be input into the convolutional neural network, the network data is reshaped into matrix, for example the 121 feature vectors are reshaped into a 11 * 11 matrix.

C. EVALUATION METRICS

In this paper, accuracy, precision, recall and F1-Measure are used as the key indicators to evaluate the performance of the model. These indicators are basically derived from the four basic attributes of the confusion matrix.

TABLE 4. Confusion Matrix

Label attribute	Predict attack	Predict normal
Actual attack	TP	FN
Actual normal	FP	TN

1) Ture Positive (TP) -Attack data that is correctly classified as an attack.

2) False Positive (FP) - Normal data that is incorrectly classified as an attack.

3) True Negative (TN) - Normal data that is correctly classified as normal.

4) False Negative (FN) - Attack data that is incorrectly classified as normal.

We will use the follow measures to evaluate the performance of our proposed solution:

$$AC = \frac{TP + TN}{TP + TN + FP + FN} \tag{13}$$

$$\Pr ecision = \frac{TP}{TP + FP} \tag{14}$$

$$\operatorname{Re} call = \frac{TP}{TP + FN} \tag{15}$$

$$F1\text{-Measure} = \frac{2\cdot \text{Precision}\cdot \text{Recall}}{\text{Precision}+\text{Recall}} \qquad (16)$$

D. EXPERIMENTAL RESULTS ON NSL-KDD

The experiment includes two processes of training and testing. Use the hybrid sampling method proposed in this paper to process the training set data, train a deep hierarchical network model, and finally use KDDTest + to test the model. The classification effect of each category is shown in Table 5. The detection accuracy on Dos can reach 96.21%, and the detection accuracy on R2L can also reach 61.32%.

TABLE 5. CNN-BiLSTM classification results of different categories on NSL-KDD

Category	Precision(%)	Recall(%)	F1-Measure(%)
Normal	86.77	**94.11**	90.29
Dos	**96.21**	85.24	90.39
Probe	64.86	**68.56**	66.66
U2R	59.98	**60.45**	60.21
R2L	**61.32**	58.95	60.11

From Figure 7, the distribution of various indicators can be seen more clearly by histogram.

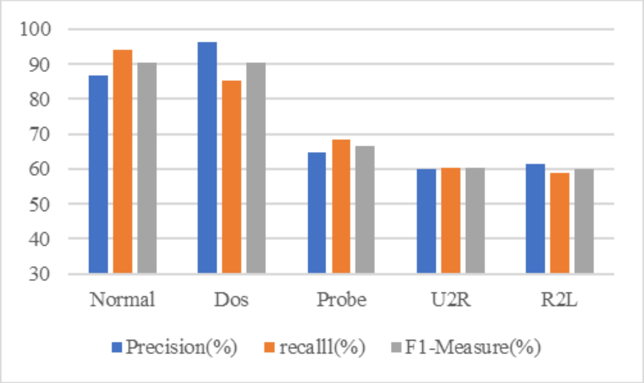


FIGURE 7. CNN-BiLSTM classification results of different categories on NSL-KDD

At present, many machine learning and deep learning algorithms are applied to network intrusion detection. Random forests and classic convolutional neural networks are widely used in network intrusion detection. Therefore, the classic classification algorithms commonly used in intrusion detection are compared with the algorithms in this paper. RF, AlexNet, LeNet-5, and CNN and BiLSTM networks are used in this paper for comparison of classification performance, which are shown in Table 6. It can be seen that compared with other classifiers, the proposed classification algorithm of this paper can reach 83.58% precision and the recall rate reaches 84.49%. Compared with RF algorithm, the accuracy rate and recall rate is increased by 8.87% and 9% respectively. Compared with the AlexNet model, the accuracy rate and F1-Measure are increased by 6.56% and 7.26%, respectively. Figure 8 shows the classification results by different classifiers.

F1-Measure is the balance point between precision and recall. It can be regarded as the harmonic average of precision and recall. Therefore, Table 7 summarizes the F1-Measure of each category by different classifiers. Among them, for the normal score, the F1 Measure reached 92.15%, which was 13.92% higher than that of the RF. It can be seen more intuitively from Figure 9 that for a small amount of U2R data, the F1-Measure is also significantly improved compared to other methods.

TABLE 6. Classification performance comparison by different classifiers on NSL-KDD

Classifier	AC (%)	Precision (%)	Recall (%)	F1-Measure (%)
RF	74.71	81.33	75.49	78.3
AlexNet	77.02	78.54	77.24	77.88
LeNet-5	79.91	82.95	80.01	80.45
CNN	81.75	82.43	82.71	82.57
BiLSTM	79.43	81.14	79.65	80.39
CNN-BiLSTM	**83.58**	**85.82**	**84.49**	**85.14**

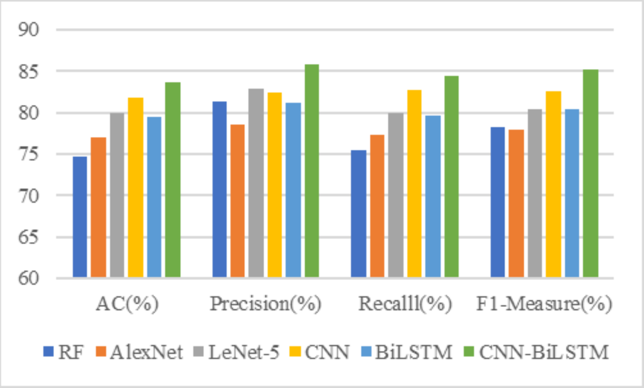


FIGURE 8. Classification results comparison on NSL-KDD

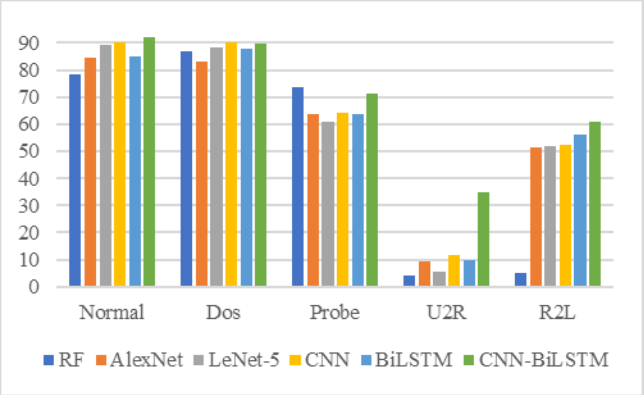


FIGURE 9. Different classifiers comparison for each class on NSL-KDD

The classification performance of all methods is not good on U2R and R2L. The primary reason is that the categories of attacks contain very less number of samples in training sets. During training, the classifiers gives less preference for

TABLE 7. F1-Measure for each class of different classifiers on NSL-KDD

Category	RF	AlexNet	LeNet-5	CNN	LSTM	CNN-BiLSTM
Normal	78.23	84.54	89.48	90.36	84.84	**92.15**
Dos	86.95	83.24	88.14	**90.14**	87.92	89.58
Probe	**73.48**	63.82	60.87	64.28	63.74	71.11
U2R	4.13	9.14	5.34	11.69	9.94	**34.69**
R2L	4.84	51.36	51.87	52.47	56.16	**60.85**

TABLE 8. Training time of different classifiers on NSL-KDD

Data	Classifier	Training time(s)	AC(%)	Precision(%)	Recall(%)	F1-Measure(%)
Original dataset	AlexNet	12900.93	79.73	80.06	**78.26**	**79.15**
	LeNet-5	**713.02**	77.40	81.20	77.61	79.36
	CNN	2929.62	77.01	80.75	77.26	78.97
	BiLSTM	6949.43	76.37	79.64	76.83	78.21
	CNN-BiLSTM	4367.99	**80.05**	**80.83**	76.84	78.78
Hybrid sampling dataset	AlexNet	1017.65	80.54	81.31	80.53	80.92
	LeNet-5	**52.24**	79.78	79.15	78.32	78.73
	CNN	219.33	81.95	84.14	82.56	83.34
	BiLSTM	571.07	75.08	75.63	75.36	75.49
	CNN-BiLSTM	341.69	**82.74**	**84.95**	**83.76**	**84.35**

these attack categories. Although this problem cannot be completely solved, but we can see from the Table 7, our method proposed in the paper, has been greatly improved the detection rate on U2R. This proves that method proposed in this paper effectively improves the problems of low minority detection rate due to data imbalance. Therefore, our proposed method solves the challenges of data imbalance to some extent in the implementation of IDS.

As shown in Table 8, compared with the original dataset, the dataset generated by hybrid sampling greatly reduced training time for classification models when all the experiments ran 50 epochs. At the same time, the CNN-BiLSTM model achieved the best deteation results among all the classification models. When the dataset was hybrid sampled, the training time of AlexNet, LeNet-5, CNN, BiLSTM, and CNN-BiLSTM was reduced by 11883.28s, 660.78s, 2710.29s, 6378.36s and 4026.30s respectively. Although the training time of CNN-BiLSTM was longer than LetNet-5, it was acceptable compared to greatly improved accuracy. CNN-BiLSTM was much faster than AlexNet and BiLSTM.

E. EXPERIMENTAL RESULTS ON UNSW-NB15

To further verify the proposed method in this paper, we also performed experiments on the UNSW-NB15 dataset. The experimental results are shown in Table 9. The classification accuracy of all classifiers is in the range 70% to 78%. In terms of classification accuracy, the method proposed in this paper is 3.7%, 3.27%, 6.05%, 2.55% and 4.92% higher than RF, AlexNet, LetNet, CNN and BiLSTM, respectively. As can be seen from Figure 10, the method proposed in this paper is higher than other methods in Precision, Recall and F1-Measure. From the classification results, we can see that our proposed method is effective. When facing complex data, we still get better results than other methods.

TABLE 9. Classification performance comparison on UNSW-NB15

Classifier	AC (%)	Precision (%)	Recall (%)	F1-Measure (%)
RF	73.46	80.84	73.64	77.07
AlexNet	73.89	79.61	72.73	76.01
LeNet-5	71.11	78.87	71.13	74.80
CNN	74.61	81.01	75.65	78.24
BiLSTM	72.24	79.52	72.43	75.81
CNN-BiLSTM	**77.16**	**82.63**	**79.91**	**81.25**

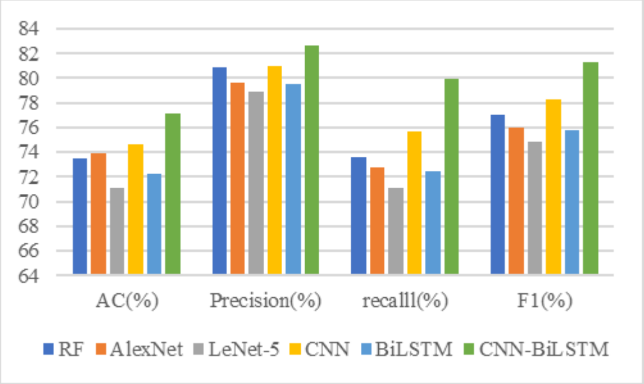


FIGURE 10. Classification results of UNSW-NB15 by different classifiers

TABLE 10. F1-Measure for each class of UNSW-NB15 by different classifiers

Category	RF	AlexNet	LeNet-5	CNN	LSTM	CNN-BiLSTM
Normal	85.86	86.85	82.29	85.20	**88.41**	84.99
Backdoor	4.33	3.62	2.96	2.53	2.90	**8.97**
Analysis	6.92	4.88	6.08	2.09	2.91	**9.69**
Fuzzers	28.13	26.59	28.93	34.68	28.31	**37.47**
Shellcode	**32.59**	23.84	19.99	24.64	18.41	30.95
Reconnaissance	58.11	53.37	63.69	**75.21**	56.65	62.54
Exploits	63.37	66.95	66.44	67.73	58.19	**67.89**
DoS	19.04	24.16	13.56	5.33	6.10	**29.55**
Worms	6.05	3.17	1.96	1.58	1.95	**10.75**
Generic	**98.98**	95.26	96.42	97.31	98.48	98.85

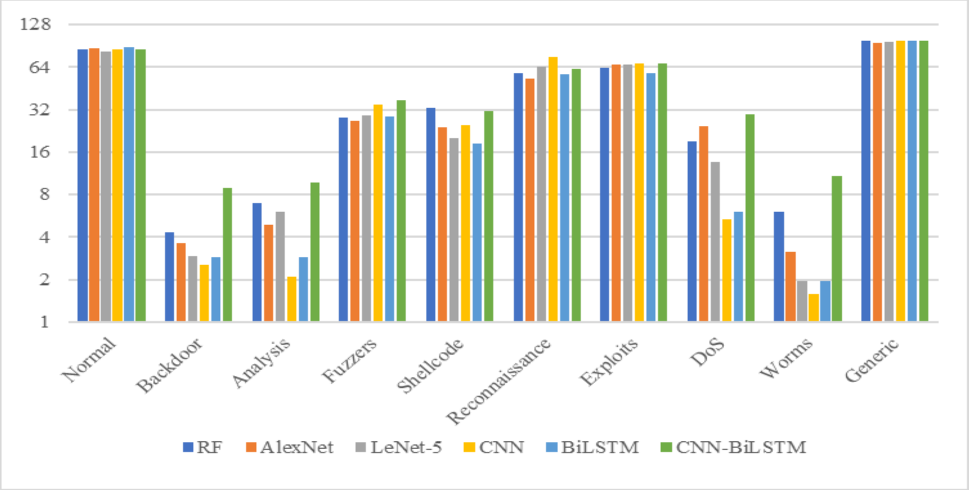


FIGURE 11. Performance comparison of different classifiers on UNSW-NB15

TABLE 11. Training time of different classifiers on UNSW-NB15

Data	Classifier	Training time(s)	AC(%)	Precision(%)	Recall(%)	F1-Measure(%)
Original dataset	AlexNet	24060.35	69.91	73.74	71.74	72.73
	LeNet-5	**1234.76**	71.93	78.83	71.56	75.02
	CNN	6115.78	73.61	**81.86**	75.29	**78.44**
	BiLSTM	7458.32	72.24	79.36	72.53	75.79
	CNN-BiLSTM	3730.68	**75.56**	79.33	**75.64**	77.44
Hybrid sampling dataset	AlexNet	16060.35	71.03	70.83	69.38	70.10
	LeNet-5	**900.74**	72.64	77.95	78.36	78.15
	CNN	4522.56	74.65	80.31	75.54	77.85
	BiLSTM	5603.63	72.19	76.38	**81.37**	78.80
	CNN-BiLSTM	2750.47	**76.82**	**81.48**	78.47	**79.95**

The detailed results for multi-class classification of various classifiers are reported in Table 10. In terms of F1-Measure, the performance of different attacks is varied across from one classifier. From Table 10, we can see that all the classifiers obtained less F1-Measure for Backdoor, Analysis and Worms in compared to the other categories such as Exploits and Generic. In multi-class, we deal with strengthening the classifier in identifying each individual attack. As shown in Figure 11, our proposed method improves detection performance on those attacks which are difficult to classify.

Although the performance on Normal, Reconnaissance, Exploits and Generic are not very outstanding, our proposed method has improved by 5.70%, 5.11%, 8.14%, 15.91%, and

7.81% on Backdoor, Analysis, Fuzzers, DoS and Worms, respectively.

We also compared the training time of different classifiers on the UNSW-NB15 dataset. We trained all models 50 epochs, the statistics time and classification results are shown in Table 11. Compared with other models, the structure of LeNet-5 is simple, so it has the shortest training time. After hybrid sampling, the training time of the model is reduced for all classifiers. The training time of our proposed model was reduced from 3730.68s to 2750.47s, and the accuracy is also improved. Although the training time of this model is not the shortest, it is acceptable in terms of the final classification results. For practical implementation, the time performance is acceptable, since the classification has to be trained only once and can then be used as an off-line anomaly detection tool in the network.

IV. CONCLUSIONS

In this paper, a novel method for intrusion detection system based on the combination of hybrid sampling and deep hierarchical network has been proposed and discussed. Firstly, we combine OSS and SMOTE to construct a balanced dataset for model training. It can reduce the training time of the model and solves the common problems to some extent of inadequate training from unbalanced samples. In addition, a network data preprocessing method is established for complex, multidimensional cyber threats, which is suitable for proposed deep hierarchical network model. Then, classify the input data through the hierarchical network model constructed by CNN and BiLSTM. The model extracts feature automatically through repeated multi-level learning by taking advantage of the outstanding features of deep learning.

Two intrusion datasets (NSL-KDD and UNSW-NB15) have been employed to evaluate the performance of the proposed approach. Based on the statistical significance tests, it could be concluded that the proposed approach outperforms other classifiers, such as Random Forest, LeNet, AlexNet, CNN and BiLSTM. The proposed method yields a superior result in terms of accuracy, precision and recall rate when validated against testing set.

REFERENCES

[1] K. Zheng et al., ''Algorithms to speedup pattern matching for network intrusion detection systems,'' *Comput. Commun*, 62, pp. 47-58, 2015

[2] Papamartzivanos D, Marmol F G, Kambourakis. G. Introducing Deep Learning Self-Adaptive Misuse Network Intrusion Detection Systems[J]. *IEEE Access*, vol. 7, pp. 13546-13560, 2019

[3] Kasongo S M, Sun Y. A Deep Learning Method with Filter Based Feature Engineering for Wireless Intrusion Detection system[J]. *IEEE Access*, vol. 7, pp. 38597-38607, 2019

[4] Ahmad I, Basheri M, Iqbal M J, et al. Performance comparison of support vector machine, random forest, and extreme learning machine for intrusion detection[J]. *IEEE Access*, vol. 6, pp. 33789-33795, 2018.

[5] Y. Xu and H. Zhao, ''Intrusion detection alarm filtering technology based on ant colony clustering algorithm,'' *Proc. 6th Int. Conf. Intell. Syst.Design Eng. Appl*., Washington, DC, USA, pp. 470-473, May 2016

[6] X. Wang, "Design of temporal sequence association rule-based intrusion detection behavior detection system for distributed network," *Modern Electron. Techn*, 41, no. 3, pp. 108-114, 2018.

[7] Ünal Çavuşoğlu. A new hybrid approach for intrusion detection using machine learning methods[J]. *Springer Journal*, vol. 49, no. 7, pp. 2735-2761, 2019.

[8] Z. Fuqun, "Detection method of LSSVM network intrusion based on hybrid kernel function," *Modern Electron. Techn*., 38, no. 21, pp. 96-99, 2015.

[9] Thaseen, I.S., Kumar, C.A. & Ahmad, A. Integrated Intrusion Detection Model Using Cinhi-Square Feature Selection and Ensemble of Classifiers[J]. *Arabian Journal for Science and Engineering*, vol. 44, pp. 3357–3368, 2019.

[10] Sumaiya Thaseen Ikram a, Aswani Kumar Cherukuri. Intrusion detection model using fusion of chi-square feature selection and multi class SVM[J]. *Journal of King Saud University – Computer and Information Sciences*, vol. 29, pp. 462-472, 2017.

[11] Peiying Tao,Zhe Sun,and Zhixin Sun, ''An Improved Intrusion Detection Algorithm Based on GA and SVM,'' *IEEE Access*, vol, 6,pp. 13624-13631, 2018.

[12] Kai Peng, Victor C. M. Leung, Qingjia Huang. Clustering Approach Based on Mini Batch Kmeans for Intrusion Detection System over Big Data[J]. *IEEE Access*, pp. 2169-3536, 2017.

[13] Farnaaz N, Jabbar M A. Random Forest Modeling for Network Intrusion Detection System[J]. *Procedia Computer Science*, vol. 89, pp. 213-217, 2016.

[14] Lefei Zhang, Qian Zhang, Liangpei Zhang, at al. Ensemble manifold regularized sparse low-rank approximation for multiview feature embedding[J]. *Pattern Recognition*, vol, 48, no. 10, pp. 3102-3112, 2015.

[15] Zhonghua Liu, Jingjing Wang, Gang Liu, et al. Discriminative low-rank preserving projection for dimensionality reduction. *Applied Soft Computing* [J], vol. 85, pp. 73-80, 2019.

[16] Zhonghua Liu, Zhihui Lai, Weihua Ou, et al. Structured optimal graph based sparse feature extraction for semi-supervised learning. Signal Processing [J], 2020, doi: 10.1016/j.sigpro.2020.107456.

[17] Y. Lecun, Y. Bengio, and G. Hinton, ''Deep learning,'' *Nature*, vol. 521, no. 7553, pp. 436-444, 2015.

[18] Zengmao Wang, Bo Du, at al. Domain adaptation with neural embedding matching, *IEEE Transactions on Neural Networks and Learning Systems* [J], pp, 1-11, 2019.

[19] Alom Z，Bontupalli V R，Taha T M.Intrusion detection using deep belief network and extreme learning machine[J]. *International Journal of Monitoring and Surveillance Technologies Research*, vol, 3, no. 2, pp. 35-56, 2016.

[20] Fiore U, Palmieri F, Castiglione A, et al. Network anomaly detection with the restricted Boltzmann machine[J]. *Neurocomputing*, Vol, 122, pp. 13-23, 2013.

[21] C. Yin, Y. Zhu, J. Fei, and X. He, ''A deep learning approach for intrusion detection using recurrent neural networks,'' *IEEE Access*, vol. 5, pp. 21594-21961, 2017.

[22] J. Kim, H. L. T. Thu, and H. Kim, ''Long short term memory recurrent neural network classifier for intrusion detection," *Proc. Int. Conf. Platform Technol. Service (PlatCon)*, pp. 1-5, Feb. 2016.

[23] Wu K, Chen Z, Li W. A Novel Intrusion Detection Model for a Massive Network Using Convolutional Neural Networks[J]. *IEEE Access*,1-1, 2018.

[24] M. Wang and J. Li, ''Network intrusion detection system based on convolutional neural network,'' *Netinfo Secur.*, 3, no. 11, pp. 990–994, 2019.

[25] A. Krizhevsky, I. Sutskever, and G. E. Hinton, ''Imagenet classification with deep convolutional neural networks,'' Commun. *ACM*, 60, no. 6, pp. 84-90, 2017.

[26] Zhang Y, Chen X, Jin L, et al. Network Intrusion Detection: Based on Deep Hierarchical Network and Original Flow Data[J]. *IEEE Access*, vol. 7, pp. 37004-37016, 2019.

[27] S. M. H. Bamakan, W. Huadong, and S.Yong, ''Ramp loss K-support vector classification-regression; a robust and sparse multi-class

approach to the intrusion detection problem,'' *Knowl. Based Syst*. 126, no. 10, pp. 113-126, 2017.

[28] Xueqin Zhang, Jiahao Chen, Yue Zhou. A Multiple-Layer Representation Learning Model for Network-Based Attack Detection[J]. *IEEE Access*, vol. 7, pp. 91992-92008, 2019.

[29] Wang S, Minku L, Yao X. Resampling-Based Ensemble Methods for Online Class Imbalance Learning[J]. *IEEE Transactions on Knowledge and Data Engineering*, vol, 27, no. 5, pp. 1356-1368, 2015.

[30] Binghao Yan, Guodong Han, Yajing Huang, et al. New traffic classification method for imbalanced network data[J]. *Journal of Computer Applications*. vol. 38, no. 1, pp, 20-25, 2018

[31] Batista G E A P A, Prati R C, Monard M C. A study of the behavior of several methods for balancing machine learning training data[J]. *ACM SIGKDD Explorations Newsletter*, vol.6, no. 1, pp.20, 2004

[32] A. Krizhevsky, I. Sutskever, and G. E. Hinton, ''Imagenet classification with deep convolutional neural networks,'' Commun. *ACM*, vol.60, no. 6, pp. 84-90, 2017

[33] Chuan-Long Y, Yue-Fei Z, Jin-Long F, et al. A Deep Learning Approach for Intrusion Detection Using Recurrent Neural Networks[J]. *IEEE Access*, vol.5, no.99, pp. 21954-21961, 2017.

[34] Junhao Zhou, Yue Lu, Hong-Ning Dai, et al. Sentiment Analysis of Chinese Microblog Based on Stacked Bidirectional LSTM[J]. *IEEE Access*, vol. 7, pp. 33856-38866, 2019.

[35] Ruben Zazo, Phani Sankar Nidadavolu, Nanxin Chen, et al. Age Estimation in Short Speech Utterances Based on LSTM Recurrent Neural Networks[J]. *IEEE Access*, vol.6, pp. 22524-22530, 2018.

[36] Dhanabal L, Shantharajah S P．A study on NSL-KDD dataset for intrusion detection system based on classification algorithms[J]. *International Journal of Advanced Research in Computer and communication Engineering*, vol.4, no.6, pp.446-452,2015.

[37] S. Revathi and A. Malathi, ''A detailed analysis on NSL-KDD dataset using various machine learning techniques for intrusion detection,'' *Int. J. Eng. Res. Technol*., vol. 2, no. 12, pp. 1848-1853, 2013.

KAIYUAN JIANG was born in Harbin, China in 1982. He received the B.S and M.S degrees from Xidian University, Xi'an, China, in 2006 and 2009, respectively, and the Ph.D. degree in communication and information system from Harbin Institute of Technology, Harbin, China, in 2014.

He joined Harbin University of Science and Technology in 2014.He is currently a postgraduate tutor of College of Measurement and Control Technology and Communication Engineering. His research work mainly focuses on image fusion, class imbalance learning and wireless network simulation.

WENYA WANG was born in Taian, Shandong province. She received the bachelor's degree of communication engineering from physical science and information engineering, Liaocheng University in 2015. She entered Harbin University of Science and Technology for electronics and communication engineering master. Her main research interest includes network information security and artificial intelligence.

AILI WANG was born in Tianjin, China in 1979. She received the B.S., M.S. and Ph.D. degrees in information and signal processing from Harbin Institute of Technology, Harbin, China, in 2002, 2004 and 2008.

She joined Harbin University of Science and Technology as an assistant in 2004 and she became an associate professor and master tutor of the department of communication engineering in 2010. She has been a visiting professor to do the research of 3D polyp reconstruction in Computer Science Lab in Chubu University, Japan in 2014.

She is the author of two books, more than 80 articles which are published on IEEE conferences and journals (EI indexed or SCI indexed). She is the chairman of 11th EAI International on Wireless and Satellites (WISATS) and 7th EAI International Conference on Green Energy and Networking (GreeNets). Her research interests include image super resolution, image fusion, object tracking and so on.

HAIBIN WU was born in Harbin, China in 1977. He received the B.S. and M.S. degrees in Harbin Institute of Technology, Harbin, China, in 2000 and 2002. He received the Ph.D. degree in measuring and testing technologies and instruments from Harbin University of Science and Technology, Harbin, China, in 2008.

From 2009 to 2012, he was a post doctor with the Key Laboratory of Underwater Robot in Harbin Engineering University. From 2014 to 2015, he was a visiting scholar with the Robot Perception and Action Laboratory in University of South Florida. Since 2012, he has been a Professor with the Instrument Science and Technology Discipline, Harbin University of Science and Technology. He is the author of three books, more than 40 articles, and more than 20 inventions. His research interests include robotic vision, visual measuring and image processing, medical virtual reality, and photoelectric testing. He is an Editorial Board Member of the journal of Liquid Crystals and Displays and an Associate Editor in chief of the journal of Harbin University of Science and Technology.

Dr. Wu was a Director of Precision Machinery Branch of China Instrumentation Society, and a Director of Visual Inspection Committee of Chinese Graphic and Image Society.