DEEP FEATURE EXTRACTION VIA SPARSE AUTOENCODER FOR INTRUSION DETECTION SYSTEM

Cao Xiaopeng and Qu Hongyan

School of Computer Science and Technology, Xi'an University of Posts and Telecommunications, Xi'an, China

ABSTRACT

The massive network traffic and high-dimensional features affect detection performance. In order to improve the efficiency and performance of detection, whale optimization sparse autoencoder model (WO-SAE) is proposed. Firstly, sparse autoencoder performs unsupervised training on high-dimensional raw data and extracts low-dimensional features of network traffic. Secondly, the key parameters of sparse autoencoder are optimized automatically by whale optimization algorithm to achieve better feature extraction ability. Finally, gated recurrent unit is used to classify the time series data. The experimental results show that the proposed model is superior to existing detection algorithms in accuracy, precision, and recall. And the accuracy presents 98.69%. WO-SAE model is a novel approach that reduces the user's reliance on deep learning expertise.

KEYWORDS

Traffic anomaly detection, Feature extraction, Sparse autoencoder, Whale optimization algorithm

1. INTRODUCTION

Devices communicate with the internet is increasing rapidly. Information and communication system are exposed to network attacks continuously. Intrusion Detection, as active defense technology, has gradually become a key technology to ensure network system security. The purpose of intrusion detection systems (IDS) is to identify unusual visits or attacks on secure internal networks.

In the process of detecting network attacks, massive network traffic packets need to be obtained and processed. The traffic contains many irrelevant features and redundant features, which affect the performance of the detection system seriously. It is necessary to extract representative features that can improve the performance and efficiency of the detection system. To reduce dimension, the feature selection method [1] selects partial features to represent the raw data. The technique removes some redundant features. It improves the detection efficiency. But it may lose partial information. Generally, traffic features extraction transforms the raw data into a lower-dimensional space through the Principal Component Analysis (PCA)[2] and Linear Discriminant Analysis (LDA) [3]. According to the extracted features, the traffic is classified to identify anomaly traffic in the network [4]. However, when the high-dimensional features present a nonlinear structure, the main disadvantage of the above methods is that they can only learn the low-dimensional structure of the raw data. These methods cannot give a deterministic mapping from a high dimensional space to low dimensional space.

David C. Wyld et al. (Eds): SPPR, NECO, GridCom, ICCSEA, SCAI, UBIC, SEMIT, MLDS - 2020
pp. 61-73, 2020. CS & IT - CSCP 2020 DOI: 10.5121/csit.2020.101906

Recently, autoencoder presented an outstanding performance in deep learning tasks. Autoencoder is an unsupervised learning method. It can reduce the data dimension by minimising the reconstruction layer [5]. It can satisfy the nonlinear learning of bidirectional mapping between high-dimensional data space and low-dimensional data space. Sparse Autoencoder (SAE) was first put forward by Ng [6] in 2011. The sparse network is achieved by adding sparse constraints to the hidden layer neurons of the traditional autoencoder, which is beneficial to reduce dimension. And it can improve the detection efficiency. As an unsupervised learning method, sparse autoencoder can directly deal with data without labels.

However, determining the optimal parameters of autoencoder mainly depends on practical experience. To get the optimal combination of parameters needs to adjust the model structure and parameters repeatedly. The more parameters, the more complex the test situation is. Therefore, it is worth learning parameters automatically by combining the autoencoder with excellent performance optimization algorithms [7].

Deep learning performs well in processing complex and high-dimensional data. It is a promising solution to intrusion detection. So this paper uses sparse autoencoder to reduce dimension by unsupervised learning. The key parameters of sparse autoencoder are optimized by whale optimization algorithm (WOA), which aims to shorten the training time and achieve better feature extraction performance. This model does not require users with an intimate knowledge of parameter tuning. Compared with the existing methods, this model not only effectively reduces the feature dimension of the raw data but also improves the detection accuracy and false positive rate.

The main contributions of this work can be summarized as follows.

(1) Feature extraction using SAE is to increase efficiency and detection accuracy.

(2) The key parameters of the SAE are obtained by WOA algorithm to save time and achieve better performance of the classifier.

This paper is organized as follows. The detailed literature survey is presented in section 2. Section 3 deals with the proposed model related details. Section 4 introduces the experimental results and performance comparison. The general conclusion and the scope for future work are given in the last part.

2. RELATED WORKS

Previous researchers have introduced various deep learning methods in IDS, such as DNN, CNN, LSTM, and so on. These methods have made a breakthrough in the intrusion system. In order to avoid the existence of defects in a single classifier, the ideas of hybrid classifiers [8,9] are applied in IDS. The efficiency of classification is generally better than single classifier models.

Although the above methods achieved excellent results. However, the main purpose of these methods is to improve the detection accuracy and false positive rate. They pay little attention to feature extraction. When it applied to large-scale IDS, IDS usually needs to meet the system requirements for real-time capability and low loss. The essential reason is that the input feature space has high dimensional and nonlinear characteristics. Tang et al. [1] applied DNN to detect anomaly traffic in Software Defined Networking (SDN). This method only selects six basic features from the NSL-KDD dataset. The six basic features selected do not focus on a specific attack. The main advantage of this method is the reduced computation time as the number of features decreases. But the accuracy is lower.

In [8], a new hybrid model has been introduced based on genetic algorithm (GA) and Principal Component Analysis (PCA) along with a support vector machine (SVM) to overcome detection performance issues. The results showed that a hybrid model could effectively detect unknown attacks. Keerthi et al. [10] performed nonlinear dimensional reduction on complex data sets through Principal Component Analysis (PCA). The application of PCA significantly reduced the number of features to be analyzed in the detection system. But it is computationally expensive in terms of training and test time.

Wang et al. [11] proposed a novel intrusion detection system. Deep CNN is used to learn the low-level spatial features of the raw data. And in the second stage, LSTM is used to learning high-level temporal features. They used two stages for feature extraction. This model is computationally expensive in terms of training and test time. Yang et al. [12] combined an improved conditional variational autoencoder (CVAE) and deep neural networks. NSL-KDD and UNSW-NB15 are used to verify this model. The experimental results show that the detection accuracy of 89.08%. Although various neural networks have been developed. Training them requires practical experience to choose the key parameters. Hinton [13] tried to guide users to set up a deep RBM learning network. It is still a very complex process for people who do not have deep learning knowledge.

According to the above literature review, the previous intrusion detection models focus on building the classification model. They pay little attention to pre-processing stages for improving the quality of the dataset. And training deep neural networks is a time-consuming task. To get the optimal combination of parameters needs to adjust the model structure and parameters repeatedly. Based on the analysis, we proposed a feature extraction model based on WOA to adjust the parameters of sparse autoencoder. First, we use WOA to optimize the key parameters of SAE, followed by optimal SAE for feature extraction. Traffic data are time series data. At last, we use gated recurrent unit (GRU) for classification. NSL-KDD dataset is used to evaluate this model.

3. WO-SAE MODEL

3.1. System model

The framework includes three modules: data pre-processing module, feature extraction module, and classification module (see in Figure 1).

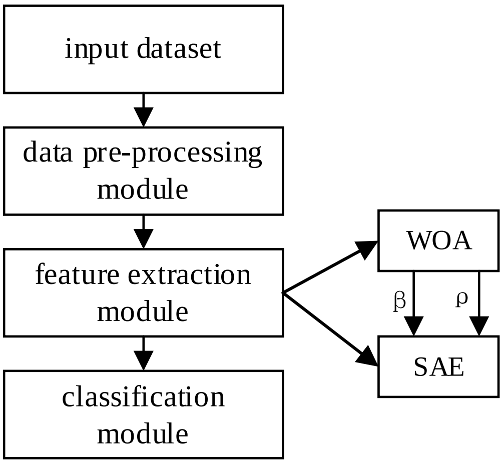

Figure 1. WO-SAE model structure

(1) data pre-processing module: transform the symbolic features into numerical features using one-hot encoding; scale the features in the range [0,1].

(2) feature extraction module: construct a sparse autoencoder with three hidden layers; use WOA algorithm to find the optimal parameters of SAE; extract low-dimensional features using optimized SAE.

(3) classification module: use GRU classification to distinguish between normal and abnormal data.

3.2. Sparse autoencoder

The Autoencoder is an unsupervised neural network, including an input layer, some hidden layers, and an output layer. The goal is to reduce dimension. Autoencoder makes the extracted features represent the raw data, avoids the curse of dimensionality. Autoencoder trained to obtain different output features can be beneficial for the performance of classification. The working process of the autoencoder can be divided into two stages, encoding and decoding. These two stages can be defined as:

The encoding process from the input layer to the hidden layer:

$$h = f(W_1 * h + b_1) \qquad (1)$$

The decoding process from the hidden layer to the output layer:

$$x^{'} = f(W_2 * h + b_2) \qquad (2)$$

where W_1 and b_1 denote the weight matrix and bias matrix of the encoder, W_2 and b_2 denote the weight matrix and bias matrix of the decoder, h is either a linear or nonlinear transfer function.

Sparse autoencoder adds some sparse constraints to the traditional autoencoder. In order to achieve the suppression effect, sparse autoencoder adds regularization terms and sparse constraints to the loss function. It restricts the average activation value of the neurons in the hidden layers. The whole function of SAE is as follows:

$$J_{SAE}(W,b) = J(W,b) + \beta(\textstyle\sum_{j=1}^{h} KL(\rho \,||\, \hat{\rho}))(3)$$

where β is the weightfactor about the strength of the sparse item and h is the number of the hidden units. The Kullback-Leibler (KL) divergence is to measure the difference between the constant ρ and the average activation $\hat{\rho}$. The function of KL is as follows:

$$KL(\rho \,||\, \hat{\rho}_j) = \rho log \frac{\rho}{\hat{\rho}_j} + (1-\rho) log \frac{1-\rho}{1-\hat{\rho}_j} \,(4)$$

However, the feature extraction ability of a single autoencoder is insufficient, and multiple autoencoders connected end to end to form a deep neural network. The stacked structure is beneficial to extract deep features of the data. The structure is shown in Figure 2.

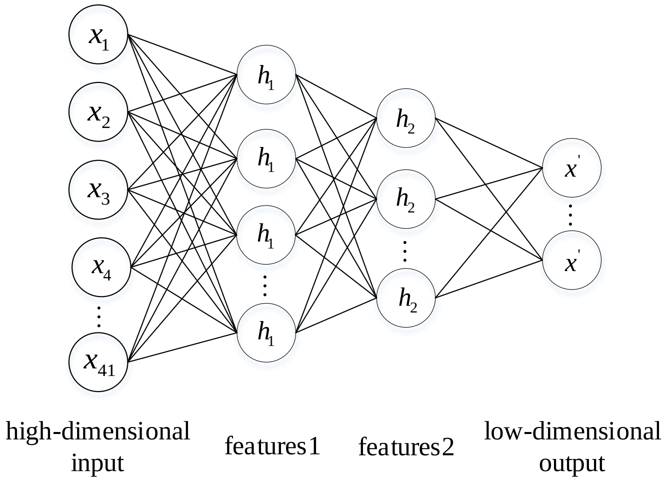

Figure 2. Deep Feature extraction by SAE

The pre-processed data is the input of the previous layer of sparse autoencoder. The output of the first sparse autoencoder is used as the input of the next autoencoder so that higher-level features representations of the raw data can be obtained. The greedy layer-wise pre-training method [14] is used to train each layer of sparse autoencoder to get the optimized connection weights and bias values. Then the error back propagation method is used to fine tune sparse autoencoder until the result of the error function between the input data and the output data satisfies the expected requirements.

3.3. Sparse autoencoder optimized by WOA algorithm

Whale optimization algorithm is a population-based meta-heuristic algorithm that better performance than algorithms such as particle swarm optimization (PSO) and genetic algorithm. WOA has the characteristics of fewer selection parameters, overcoming the local optimum entrapment, and fast convergence to the best solution [15]. In order to prey, the whale creates a spiral structure path and then follows the bubble to determine the position of the prey. The spiral model and the surrounding mechanism are used alternately to simulate this behaviour. The position is updated with a probability of 0.5. This method contains the following three stages: circling hunting, bubble-net attacking, and prey hunting.

For the deep learning systems, the parameters of the model need to be adjusted repeatedly in the experiment. Finding the unknown parameters of the model is an optimization problem that can be solved by a meta-heuristic optimization algorithm [16]. The method to optimize the parameters of SAE using WOA was proposed to ensure that the extracted features are the most representative. It does not need any deep learning specific knowledge. Training a deep neural network is a time-consuming task. The value of β and ρ in (3) affects the classification performance of the constructed model. Therefore, WOA could be used to obtain the optimal parameters of sparse autoencoder.

The process of optimization is shown in Figure 3. The detailed optimization process is as follows:

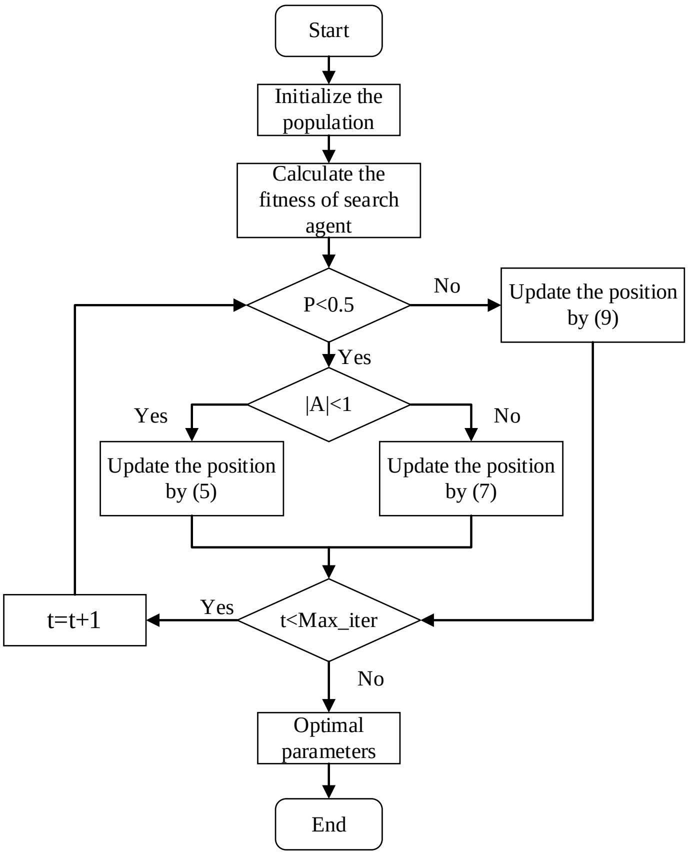
Start
Initialize the population
Calculate the fitness of search agent
P<0.5
No
Update the position by (9)
Yes
|A|<1
Yes
No
Update the position by (5)
Update the position by (7)
t=t+1
Yes
t<Max_iter
No
Optimal parameters
End

Figure 3. use WOA algorithm to optimize SAE

Step 1: Initialize the agent population N , the maximum iteration number Max_iter , and the searching range of optimized parameters $para_i = [\beta_i, \rho_i](i = 1, 2, 3, ...)$.

Step 2: Use the parameters set $para_i$ to train SAE, calculate the fitness of each search agent, and update the position of the current search agent.

Step 3: The process of updating the position of the search agent is as follows:

Generate p in [0,1] randomly, if $p < 0.5$ and $| A |< 1$, then update the position of the current search agent by (5).

$$X(t+1) = X^{*}(t) - A \cdot D \qquad (5)$$

$$D = | C \cdot X^{*}(t) - X(t) | \qquad (6)$$

where t indicates the current iteration, C is a random number evenly distributed in [0,2], $X^{*}(t)$ is the position vector of the best solution obtained so far. Equation (5) allows any search agent to update its position in the neighbourhood of the current best solution and simulates encircling the prey.

If $p < 0.5$ and $|A| \geq 1$, update the position of the current search agent by (7).

$$X(t+1) = X_{rand}(t) - A \cdot D \tag{7}$$

$$D = |C \cdot X_{rand}(t) - X(t)| \tag{8}$$

where X_{rand} is a random position vector selected from the current population.

If $p > 0.5$, update the position of the current search agent by (9).

$$X(t+1) = e^{bl} \cdot \cos(2\pi l) \cdot D^{'} + X^{*}(t) \tag{9}$$

$$D^{'} = | X^{*}(t) - X(t) | \tag{10}$$

where $D^{'}$ is the distance of the search agent from the current best position, b is the constant for defining the shape of the logarithmic spiral, l is a random number in [-1,1].

Step 4: Check if t goes beyond the maximum number of iterations and output the optimal parameters; Otherwise, back to Step 3 to continue to update $X^{*}(t)$.

4. Experiments and Analysis

In this section, the datasets and the evaluation are introduced. Then the experiments are conducted for evaluating the proposed method compared with other intrusion detection methods.

4.1. Dataset and evaluation

This paper selects the NSL-KDD [17] datasets to evaluate the performance of the proposed method. It was improved on KDD 99 dataset and eliminating redundant records from the KDD 99. NSL-KDD contains 41 classification features and the 42nd attribute represents the attack type. The training set contains 21 different attack types, which can be divided into four types: Denial of service attacks (DOS), Probing attacks (Probe), User to root attacks (U2R), and Remote to Local attacks (R2L). In test set, it provides 16 new attack types that do not exist in the training set. The information of the training set and the test set are shown in Table 1.

Table 1. Attacks in the NSL-KDD dataset.

Dataset Type	Instance	Normal	Attack (%)
NSL-KDD Train20	25192	13499	46.6
NSL-KDD Train+	125973	67343	46.5
NSL-KDD Test+	22544	9711	56.9
NSL-KDD Test-	11850	2152	81.8

In this paper, the effect of IDS is evaluated by accuracy, precision, recall, and F1-score. Accuracy measures the percentage of true detection over total records. F1-score is the harmonic mean of the precision and recall to give a better measure of the accuracy.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (11)$$

$$Precision = \frac{TP}{TP+FP} \quad (12)$$

$$Recall = \frac{TP}{TP+FN} \quad (13)$$

$$f1\_score = 2*\frac{precision*recall}{precision+recall} \quad (14)$$

where True Positive (TP) indicates the number of attack records correctly classified. True Negative (TN) indicates the number of normal records correctly classified. False Positive (FP) indicates the number of normal records incorrectly classified. False Negative (FN) indicates the number of attack records incorrectly classified.

4.2. Data pre-processing

The NSL-KDD contains 41 classification features, which include symbolic features,0-1 type features, and percentage-type features. The symbolic features include protocol type, service, and flag. We use one-hot encoding to transform the symbolic features into numerical features. Nonlinear normalization is applied to the features with large data differentiation.

$$X^{'} = \log_{10} X \quad (15)$$

The original feature values are normalized in [0,1] by the maximum-minimum normalization method.

$$x^{'} = \frac{x-min}{max\text{ - }min} \quad (16)$$

where max and min are the maximum and minimum values of the original feature values, $x^{'}$ is the normalized feature value.

4.3. Model parameters

In this paper, the constructed sparse autoencoder network is used to reduce the dimension of the raw data. WOA algorithm is used to optimize the parameters in (3). After dataset pre-processing, the dimensions of features in NSL-KDD is extended to 121 dimensions. Thus, the number of input layer neurons of SAE is 121, and the number of neurons in hidden layers are orderly 80, 50, and 20. The high-dimensional features are extracted to low-dimensional features through the constructed sparse autoencoder. The next stage is to train the GRU classifier using the obtained features. The experimental parameters in Table 2 present optimal performance.

Table 2. The experimental parameters of SAE.

Hyperparameter	**Value**
Sparsity weight β	0.273
Sparsity proportion ρ	0.05
Neurons in input layer	121
Neurons in 1st hidden layer	100
Neurons in 2nd hidden layer	80
Neurons in 3rd hidden layer	50
Neurons in output layer	20
Batch Size	32
Epochs	20
Loss	cross-entropy
optimizer	Adam

4.4. Results and analysis

In order to evaluate the performance of the feature extraction by optimal sparse autoencoder. Firstly, we trained a GRU using the raw data. Secondly using the extracted features, the experimental results are shown in Figure 4.

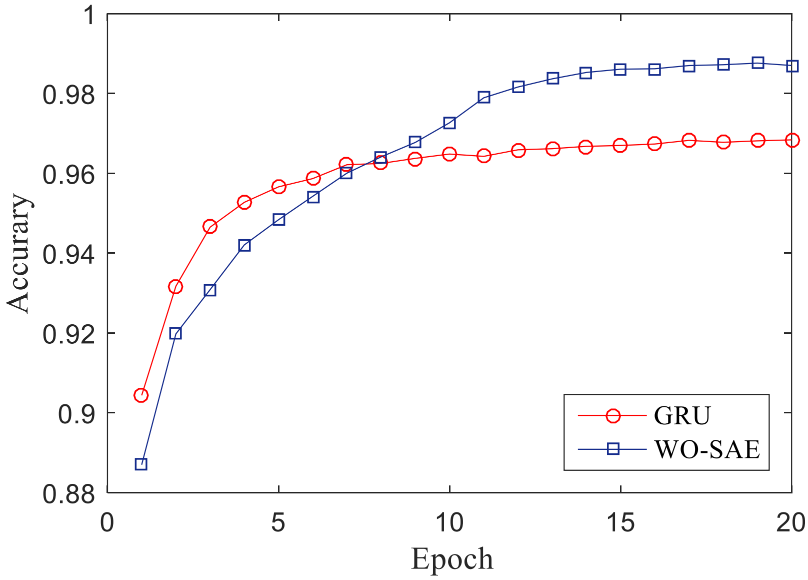

Figure 4. The effect of feature extraction on the performance of the classifier

Deep features extracted by WO-SAE model improve the performance of classifier. The accuracy is 98.69%. The GRU classifier using raw data presents 96.25% accuracy. The training time of WO-SAE model is 8.25s. And the training time of GRU classifier is 9.59s. The proposed method can reduce the training time, improve the efficiency of IDS. The extracted low-dimensional features have no negative effect on the performance of classifier. Sparse autoencoder obtains the low-dimensional features while retaining the information in the input data.

To evaluate the performance of the dimensions of features extracted on classifiers. The parameters of SAE remain unchanged. The dimension of features extracted changes from 5 to 25.

The accuracy in different dimensions is shown in Figure 5. The performance of classification is the best when the dimension of features is reduced to between 20 and 25. The accuracy can reach 98.69% when the dimension of features is 20.

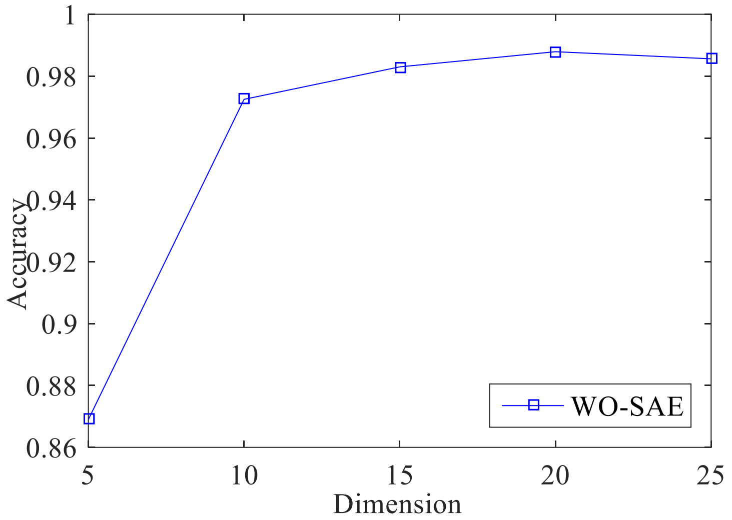

Figure 5. The effect of compression dimension on classifier

The performance of the constructed model depends on the key parameters. Compared with the performance of different learning rates on the classification of the model, Figure 6 shows the experimental accuracy and loss for two-category classification. With the decrease of the learning rate, the accuracy increase, and the loss gradually decrease. The learning rate was 0.001, and the accuracy achieved 98.69%. When the learning rate dropped to 0.0001, the classification accuracy of the training set is the best. But the effect on the test set is not well. The smaller the learning rate, the more accurate the training. The generalization ability of the model cannot express well. The accuracy of the training set decreased.

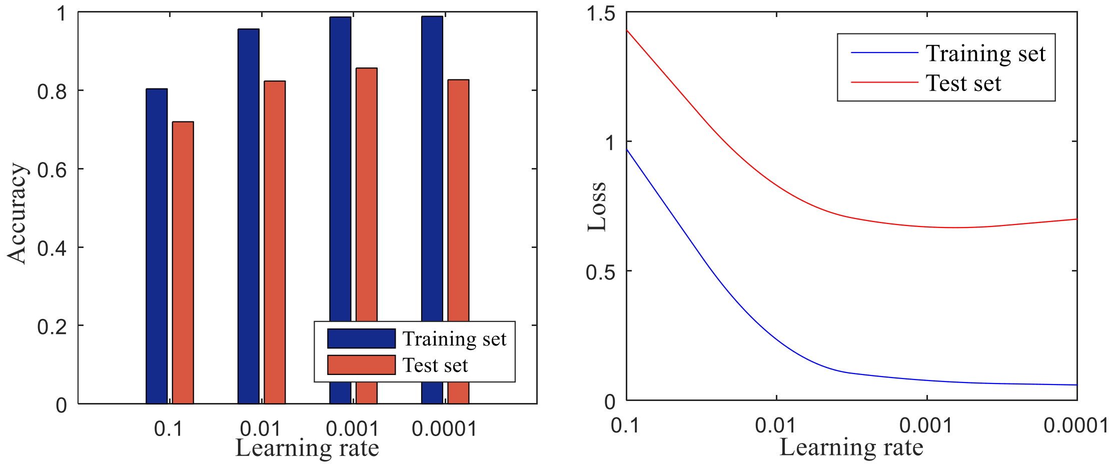

Figure 6. Accuracy and loss of different learning rates

The proposed method is compared with four base classifiers including Decision Tree (DT) algorithm, Random Forest (RF) algorithm, DNN, and LSTM respectively. The input of all algorithms is the low-dimensional features by optimal SAE. The results are shown in Table 3. We can know the proposed method is superior to other algorithms from evaluating the accuracy, precision, recall, and f1-score.

Table 3. Performance comparison of different algorithms.

Method	Accuracy	Precision	Recall	F1-score
DT	0.8503	0.7916	0.7139	0.6997
RF	0.8624	0.7691	0.7933	0.7746
DNN	0.9685	0.9732	0.9585	0.9658
LSTM	0.9459	0.9693	0.9505	0.9525
WO-SAE	0.9869	0.9848	0.9837	0.9843

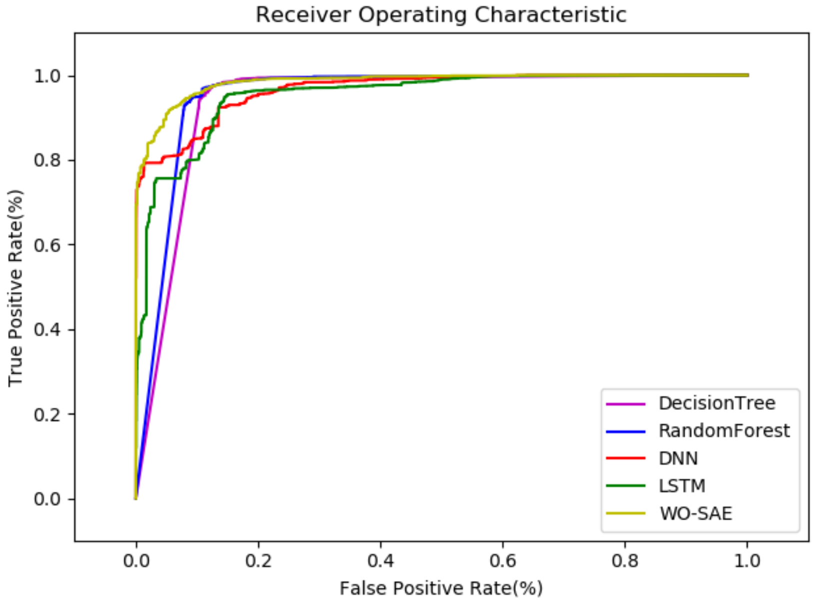

Figure 7. ROC curve comparison for different algorithms

The ROC curve reflects the relationship between true positive rate and false positive rate. The area under the ROC curve is used to evaluate the classifiers. The higher the ROC curve's area, the better the model. From Figure 7, the proposed method performs well among all the algorithms, which verifies that the method proposed in this paper has better detection performance for two-category classification compared with existing algorithms. The method of deep feature extraction by SAE can extract deep features from complex data. Combining the optimal SAE with GRU presents remarkable results.

Table 4. Performance comparison between WO-SAE and other recent scholarly works

.

Method	Accuracy(%)	Precision (%)	Recall (%)	F1-score (%)
Statistical analysis and AE [18]	84.21	87	80.37	81.98
PSO-LSTM [19]	94.07	97.23	92.21	94.65
CBR-CNN [20]	89.41	94.42	-	-
IGAN [21]	84.45	84.85	84.85	84.17
WO-SAE	98.69	98.48	98.37	98.43

Furthermore, the proposed method is compared with some recent scholarly works as shown in Table 4. It can be seen that the proposed method shows significant improvement compared to the other methods in terms of classification performance.

5. CONCLUSIONS

In this paper, an intrusion detection model based on deep feature extraction through sparse autoencoder is proposed. This model does not depend on manual experience. It can automatically obtain the key parameters of sparse autoencoder and extract deep features by optimal SAE. To achieve better classification effect, the accuracy presented 98.69% by using GRU for classification. Compared with the existing IDS methods, the proposed model reduces the complexity of detection and the training time. It can effectively identify the abnormal traffic in the network and provide guarantee for network security. We believe that the WO-SAE model may support in future research. As part of our future work, we would find more realistic network traffic data to verify our model.

ACKNOWLEDGEMENTS

This research was supported by National Natural Science Foundation of China (61136002), Industrial Public Relations Project of Shaanxi Province(2014k06-36), Science and Technology Plan Project of Xi'an (CX12188(7)).

REFERENCES

[1] T. A. Tang, L. Mhamdi, D. McLernon, S. A. R. Zaidi & M. Ghogho, (2016) "Deep learning approach for Network Intrusion Detection in Software Defined Networking", 2016 International Conference on Wireless Networks and Mobile Communications (WINCOM), Fez, Morocco, pp258-263.

[2] U. Demšar, P. Harris, C. Brunsdon, A. S. Fotheringham & S. McLoone, (2013) "Principal Component Analysis on Spatial Data: An Overview", Annals of the Association of American Geographers, Vol. 103, No. 5, pp106-128.

[3] A. Sharma & K. K. Paliwal, (2015) "Linear discriminant analysis for the small sample size problem: an overview", International Journal of Machine Learning and Cybernetics, Vol. 6, No. 3, pp443-454.

[4] Masdari M & Khezri H, (2020) "A survey and taxonomy of the fuzzy signature-based Intrusion Detection Systems", Applied Soft Computing, 106301.

[5] Hinton G E & Salakhutdinov R R, (2006) "Reducing the dimensionality of data with neuralnetworks", Science, Vol. 313, pp504-507.

[6] Ng A, (2011) "Sparse autoencoder", CS294A Lecture Notes, pp1-19.

[7] Yuan, F.-N, Zhang, L. , Shi, J.-T , Xia, X. & Li, G, (2019) "Theories and Applications of Auto-Encoder Neural Networks: A Literature Survey", Chinese Journal of Computers, Vol. 42, pp203-230.

[8] AhmadIftikhar, Abdullah Azween, Alghamdi, Abdullah & Hussain Muhammad, (2011) "Optimized intrusion detection mechanism using soft computing techniques", Telecommunication Systems, Vol. 52, No. 4, pp2187- 2195.

[9] Zhang H , Huang L & Wu C Q, (2020) "An Effective Convolutional Neural Network Based on SMOTE and Gaussian Mixture Model for Intrusion Detection in Imbalanced Dataset", Computer Networks1, Vol. 177, 07315.

[10] Keerthi Vasan. K & Surendiran. B, (2016) "Dimensionality reduction using principal component analysis for network intrusion detection", Perspectives in Science.

[11] WangWei, ShengY, Wang Jinlin, ZengXuewen, YeXiaozhou, HuangYongzhong & ZhuMing, (2018) " HAST-IDS: Learning Hierarchical Spatial-Temporal Features using Deep Neural Networks to Improve Intrusion Detection", IEEE Access, Vol. 6, pp1792-1806.

[12] Yang Yanqing, Zheng Kangfeng, Wu Chunhua & Yang Yixian, (2019) "Improving the Classification Effectiveness of Intrusion Detection by Using Improved Conditional Variational AutoEncoder and Deep Neural Network", Sensors, Vol.19, pp2528.

[13] Hinton, G., (2010) "A practical guide to training restricted boltzmann machines", Momentum, Vol. 9, pp926-947.

[14] T. T. H. Le, J. Kim & H. Kim, (2017) "An effective intrusion detection classifier using long short-term memory with gradient descent optimization", in Proc. IEEE Int. Conf. Plat. Technol. Service (PlatCon), Busan, South Korea, pp1–6.

[15] Mirjalili S & Lewis A, (2016) "The whale optimization algorithm", Advances in Engineering Software, Vol. 95, pp51-67.

[16] N. Sirdeshpande & V. Udupi, (2017) "Fractional lion optimization for cluster head-based routing protocol in wireless sensor network", Journal of the Franklin Institute, Vol. 354, pp4457–4480.

[17] Tavallaee M, Bagheri E & Lu W, (2009) "A detailed analysis of the KDD CUP 99 data set", IEEE International Conference on Computational Intelligence for Security & Defense Applications, Ottawa, pp53-58.

[18] Cosimo Ieracitano, Ahsan Adeel, Francesco Carlo Morabito & Amir Hussain, (2020) "A Novel Statistical Analysis and Autoencoder Driven Intelligent Intrusion Detection Approach", Neurocomputing, Vol 387, pp 51-62.

[19] Wisam Elmasry, Akhan Akbulut & Abdul Halim Zaim, (2020) "Evolving deep learning architectures for network intrusion detection using a double PSO metaheuristic", Computer Networks, Vol. 168, 107042, 10.1016/j.comnet.2019.107042.

[20] Naveed Chouhan, Asifullah Khan & Haroon-ur-Rasheed Khan, (2019) "Network anomaly detection using channel boosted and residual learning based deep convolutional neural network", Applied Soft Computing, Vol 83, 105612, 83. 105612. 10.1016/j.asoc.2019.105612.

[21] HuangShuokang & Lei Kai, (2020) "IGAN-IDS: An Imbalanced Generative Adversarial Network towards Intrusion Detection System in Ad-hoc Networks", Ad Hoc Networks, Vol 105, 102177, 10.1016/j.adhoc.2020.102177.

AUTHORS

Cao Xiaopeng is a Professor in Xi'an University of Posts and Telecommunications. His research interests include natural language processing, swarm intelligence algorithm.

Qu Hongyan is a graduate student in Xi'an University of Posts and Telecommunications. Her main research interests are deep learning and network security.