ITC 4/48

Information Technology
and Control
Vol. 48 / No. 4 / 2019
pp. 618-625
DOI 10.5755/j01.itc.48.4.24036

Detection of Network Intrusion Threat Based on the Probabilistic Neural Network Model

Received 2019/08/15

Accepted after revision 2019/10/18

http://dx.doi.org/10.5755/j01.itc.48.4.24036

# Detection of Network Intrusion Threat Based on the Probabilistic Neural Network Model

**Benyou Wang**

School of Electronic and Information Engineering, West Anhui University, Lu'an, Anhui 237012, China

**Li Gu**

School of Information Science and Technology, University of Science and Technology of China, Hefei, Anhui 230026, China

**Corresponding author:** benyouwby@yeah.net

With the popularity of the Internet, people's lives are becoming more and more convenient. However, the network security problems are becoming increasingly serious. This paper, aiming to better protect users' network security from the internal and external malicious attacks, briefly introduces the probabilistic neural network and principal component analysis method, and combines them for detection of network intrusion data. Simulation analysis of Probabilistic Neural Network (PNN) and Principal Component Analysis-Probabilistic Neural Network (PCA-PNN) are carried out in MATLAB software. The results suggest that the Principal Component Analysis (PCA) algorithm greatly reduce the dimension of the original data and the amount of calculation. Compared with PNN, PCA-PNN has higher accuracy and precision rate, lower false alarm rate, and faster detecting speed. Moreover, PCA-PNN has better detecting performance when there are few training samples. In summary, PCA-PNN can be used for the detection of network intrusion threat.

KEYWORDS: Probabilistic Neural Network, network security, Principal Component Analysis, detection algorithm.

## 1. Introduction

Since the advent of computers, the Internet [15] has brought great convenience to our life. However, its advantages also benefit criminals, providing a convenient platform for illegal crimes, such as hackers' illegal approach to confidential corporate data by cyber intrusion [8]. In the Internet information age, software for cyber attacks is so easy to obtain that criminals can maliciously attack the network at a low cost without professional knowledge [12]. The use of networks and computers are seriously affected by these malicious network attacks. Therefore, network intrusion detection is getting more and more attention with the development of network. Related research is as follows. Fossaceca et al. [1] applied multi-core enhancement and multi-class Kelm to network intrusion detection. Simulation tests were carried out and achieved the results of higher detecting rate and lower false alarm rate than support vector machine. Singh et al. [13] proposed an intrusion detection technology based on online sequential extremum learning machine. The simulation results showed that the detecting accuracy was 98.66%, the false alarm rate was 1.74%, and the detecting time was 2.43 seconds. Hoz et al. [6] proposed a network anomaly detection classification method, which combined statistical techniques and SOM together. The simulation results showed that the system can effectively detect network intrusion data, and its detection capability can be modified without retraining. In this study, a probabilistic neural network (PNN) algorithm was used to distinguish the data types of network intrusion. This algorithm made full use of the historical information of data intrusion and made a judgment on the data types of intrusion by using Bayesian decision theory. However, in the actual data detection, most types of data have high feature dimension and contain a lot of information, but the main feature dimension which can be used to judge whether it is the intrusion data is only a part of it, and the other dimensions are either irrelevant or have little impact. Using this data directly will affect the PNN training or detection because of redundant information. The main contribution of of this study is that Principal Component Analysis (PCA) is combined with PCA and PCA is used to reduce the dimension of detection data and remove the redundant information, so as to improve the training and detection effect of PNN. Moreover, the simulation results also show that the PCA-PNN algorithm has higher accuracy and precision and lower false alarm rate than PNN algorithm.

## 2. Probabilistic Neural Network

Artificial neural network is a distributed parallel algorithm that imitates human neural network, and its types include Error Back Propagation (BP), Convolutional Neural Networks (CNN), Time-delay Neural Networks (TDNN) and so on. In this paper, the probabilistic neural network (PNN) with better classification effect was used for network intrusion detection. PNN judges the most probable type of input vector by probability density function and Bayesian optimal decision theory. Bayesian decision theory [9] is:

$$\begin{array}{l} \text{if} \quad p(a|x) > p(b|x), \forall b \neq a \\ \text{then} \quad x \in a \end{array}, \tag{1}$$

where $x$ stands for the input vector, $a,b$ stand for a collection of two different types, and $p(a|x), p(b|x)$ represent the probability that $x$ belongs to $a,b$ respectively.

In short, Bayesian decision theory is to classify the detected vector as the type of which its probability is the highest among all types. The calculation of $p(a|x)$ is based on probability density function $p(x|a)$ , and its calculation formula [7] is as follows:

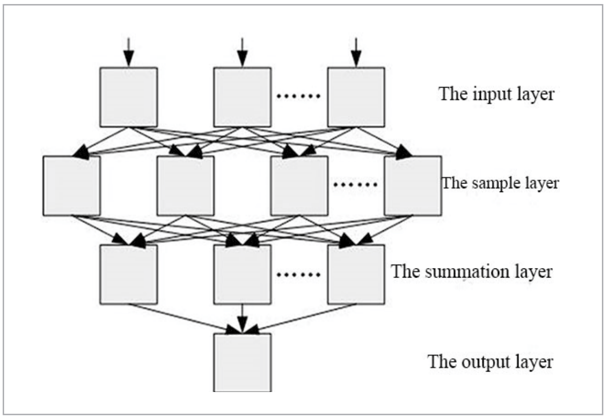
$$\begin{cases} p(a|x) = p(a)p(x|a) \\ p(x|a) = \dfrac{1}{N_a}\displaystyle\sum_{m=1}^{N_a} \dfrac{\exp(-\dfrac{\|x - x_{am}\|^2}{\sigma^2})}{(2\pi)^{l/2}\sigma^l} \end{cases}, \tag{2}$$

where $x_{am}$ is the m-th training sample of type a and $l$ is the sample dimension.

According to the above theory, the basic structure of PNN [5] is divided into an input layer, a sample layer, a summation layer, and an output layer, as shown in Figure 1.

**Figure 1**

The structure of the probabilistic neural network



1 Calculation is not performed in the input layer, and the input vector samples are only assigned to the sample layer by a linear function.

2 Calculation is performed according to Equation (3) after the input vector is received in the sample layer. It is then passed down to the summation layer.

$$y_{am} = \exp(-\frac{\|x - x_{am}\|^2}{\sigma^2}), \tag{3}$$

where $y_{am}$ stands for the value passed by type a to the summation layer and $\sigma$ stands for the smoothing factor.

3 Linear summation is performed after the data is received in the summation layer and its formula [3] is as follows:

$$g_a(x) = \frac{\sum_{m=1}^{N_a} \exp(\frac{-\|x - x_{am}\|^2}{\sigma^2})}{N_a}, \tag{4}$$

where $g_a(x)$ stands for the overall probability of the input vector $x$ belonging to type *a*, and $N_a$ is the total number of training samples in type *a*. It is then passed to the output layer.

4 The output layer is also called the decision layer. The number of output results of this layer is determined by the number of classification types, and there is only one 1 and the rest are 0. The sum probability of the input vector in different types, which is calculated by the summation layer, is received in this layer. Then the type with the highest probability is determined according to the Bayesian decision theory above, that is, definition (1). The output of the type is 1, and the rest of the types are 0 regardless of the probability.

## 3. Principal Component Analysis

PCA is short for principal component analysis [4]. Its basic principle is to transform the original component-related random vector into a component-unrelated random vector by means of orthogonal transformation, and then perform dimensionality reduction on the transformed multi-dimensional variable system. After that, the low-dimensional variable system is transformed into a one-dimensional system through the value function. In this paper, the variance is used to measure the correlation between variables, and the new unrelated principal components are combined according to the variance. After combining this method with PNN, the calculation complexity can be greatly reduced and the calculation efficiency can be improved on the premise of ensuring the accuracy.

The calculation steps of PCA for input variables are as follows:

1 The sample data is standardized to form matrix *X*.

2 The covariance matrix is calculated through matrix *X*:

$$C_X = \frac{\sum_{i=1}^{p} (X_i, X_i^T)}{p-1}, \tag{5}$$

where $X_i$ stands for the *i*-th row matrix, $X_i^T$ stands for the vertical conversion of the *i*-th row matrix, and $p$ is the total number of rows in the matrix.

3 Jacobian matrix [10] is used to extract the eigenvalues of $C_X$ and the corresponding eigenvectors, and then the contribution and cumulative contribution rates of the principal components is calculated:

$$\begin{cases} \eta = \dfrac{\lambda_i}{\sum_{k=1}^{p} \lambda_k} \\ \sum \eta = \dfrac{\sum_{k=1}^{i} \lambda_k}{\sum_{k=1}^{p} \lambda_k} \end{cases}, \tag{6}$$

where $\eta$ refers to the *i*-th principal component contribution rate, $\lambda_i$ refers to the *i*-th eigenvalue, and $\sum \eta$ refers to the cumulative contribution rate of the first *i* principal components.

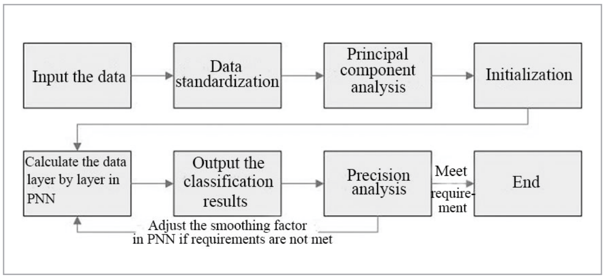
4 The input vector is sorted in the descending order according to the main component contribution rate. Then the eigenvectors corresponding to the appropriate number of eigenvalue are selected according to the cumulative contribution rate. The eigenvectors are used to form matrix *Q* and the data is finally obtained after dimensionality reduction according to

$$Y = X \cdot Q. \tag{7}$$

As shown in Figure 2, in the learning process of PCA-PNN, the training sample data is the first to input. In this paper, the detection object of the PCA-PNN model is network intrusion data. The numerical span of different types of data is so wide that it would hinder training and application if it is not dealt with standardization processing. After data standardization, the above PCA method is adopted to reduce the dimension of the data. Then after data reduction, the relevant parameters of the model are initialized according to the data dimension, including the output layer node, the sample layer node and the smoothing parameters, etc. These data are calculated layer by layer in PNN to get the final classification result, which is compared with the actual result to perform precision analysis. If the result does not conform to the requirements, the smoothing factor in PNN will be adjusted and the data recalculated. The training is repeated and ends until the accuracy met the requirements. After all steps, the data can be directly input for application. In addition, the training and use process of PNN model before improvement is similar to that of Figure 2 if the step of principal component analysis is removed.

**Figure 2**

Learning process of PCA-PNN


Input the data
Data standardization
Principal component analysis
Initialization
Calculate the data layer by layer in PNN
Output the classification results
Precision analysis
Meet require-ment
End
Adjust the smoothing factor in PNN if requirements are not met

## 4. Simulation Experiment

### 4.1. Experimental Environment

The MATLAB software [14] is used to perform the simulation analysis of the PNN and PCA-PNN detection models. The experiments are carried out on a laboratory server. The configuration of the server is Windows 7 system, I7 processor and 16G memory.

### 4.2. Experimental Data

As shown in Table 1, the KDD99 data set [2] is used in this paper. Each data in the data set is 42-dimensional. The first 41 dimensions are the characteristic attributes of the data. The last one is the decision attribute, which indicates whether the data are abnormal or not and is used to detect the performance of algorithm. The data set includes normal data and four intrusion data of Dos, R2L, U2R and Probe, which can simulate the real network environment.

**Table 1**

Experimental data

| Type of data | Normal | Dos | R2L | U2R | Probe |
|---|---|---|---|---|---|
| Number of training samples | 500 | 600 | 100 | 150 | 250 |
| Number of test samples | 5000 | 6000 | 1000 | 1500 | 2500 |

### 4.3. Experimental Steps

1 Data processing: Among the 41-dimensional features of the data in the KDD99 data set, only the 38-dimensional features are numbers and the 3-dimensional features are characters, which cannot be directly recognized by PNN. Therefore, the character features are first converted into digital features and the final 41-dimensional features become 122-dimensional digital features. In order to eliminate excessive numerical span between different data, the maximum and minimum method [11] is used to standardize the data:

$$y = \frac{x - x_{\min}}{x_{\max} - x_{\min}}, \tag{8}$$

where $y$ stands for the standardized data, $x$ stands for the data that need to be standardized, $x_{\min}$ stands for the minimum value among the data, and $x_{\max}$ stands for the maximum value among the data.

2 Data dimension reduction: The dimension of the data after standardization is still 122-dimensional without change. If the PNN model is used for detection, the PNN layer-by-layer calculation is directly performed. The dimension, however, is so large that the data computation will increase. Therefore, in the PCA-PNN model, PCA is used at first to reduce the dimension of the data. Then, the dimensional features to input are selected according to the contribution and cumulative contribution rate of the new dimension. In this paper, the new dimension with the cumulative contribution up to 98% is set as the input vector.

3 Model training: The processed data is input into PNN and the model is repeatedly trained according to the accuracy requirements. It is mainly to adjust the smoothing factor in (0, 1].

4 Model detection: After the training of test model, the test set samples are used for performance test. Then the actual performance of two detection algorithms was tested. A simple website is established using server ① in the laboratory. Then server ① is connected with server ② via the domain name. The PNN and PCA-PNN algorithm was used in server ② to detect the data which are transmitted to the website in server ①. In server ③, data in KDD99 data set are sent to the website in server ① every ten minutes within three hours; the data content is randomly selected, and the amount was between 10 and 100.

### 4.4. Detecting Indices

Accuracy *ACC*, false alarm rate *FAR*, and precision rate *DR* are adopted to evaluate the performance of the algorithm. Its calculation formula is as follows:

$$\begin{cases} ACC = \dfrac{TP+TN}{TP+TN+FP+FN} \\ FAR = \dfrac{FN}{TP+FN} \\ DR = \dfrac{TP}{TP+FP} \end{cases}, \quad (9)$$

where *TP* refers to the number of samples which are classified as attacks and are attacks in fact, *TN* refers to the number of samples which are classified as normal and moreover are normal in fact, *FP* refers to the number of samples which are classified as normal but are attacks in fact, and *FN* refers to the number of samples which are classified as attacks but are normal in fact.

In addition, for the network intrusion detection model, it is not only the high detecting accuracy but also the fast detecting speed that is needed to issue alarms and interception in time. Therefore, the detecting time is also used as the evaluation index in this paper.

### 4.5. Experimental Results

As shown in Table 2, the dimension of KDD99 data after PCA dimensionality reduction reduces from 122 to 8. The dimensions are arranged in descending order of contribution rate. It can been seen from the table that the cumulative contribution is 0.987 for the 6th dimension, which means that the new dimension 1~6 provides 98.7% effective information. According to the present standard of cumulative contribution rate, 98%, the new dimension 1~6 is selected as the input vector of PCA-PNN.

As shown in Figure 3, the detecting accuracy of PNN is 94.77% for normal data, 98.81% for Dos, 62.62% for R2L, 63.54% for U2R, and 63.51% for Probe. The detecting accuracy of PCA-PNN is 97.63% for normal data, 98.86% for Dos, 82.44% for R2L, 82.89%for U2R, and 88.12% for Probe. It can be clearly seen from the figure that PCA-PNN has higher accuracy in detecting both the normal and attack data, and the accuracy of the two models was close except in the detection of Dos attack data.

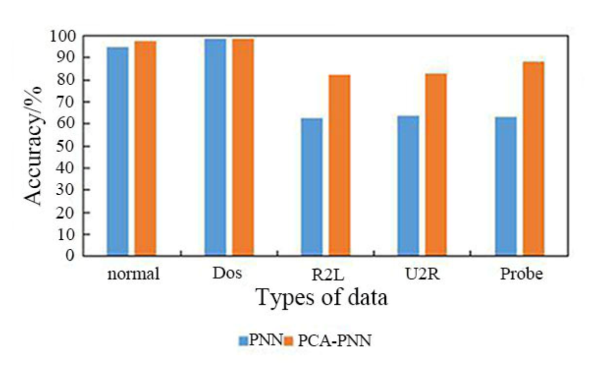
**Table 1**

PCA results

| Number of dimension | Contribution rate | Cumulative contribution rate | Number of dimension | Contribution rate | Cumulative contribution rate |
|---|---|---|---|---|---|
| 1 | 0.478 | 0.478 | 5 | 0.028 | 0.973 |
| 2 | 0.223 | 0.701 | 6 | 0.014 | 0.987 |
| 3 | 0.149 | 0.850 | 7 | 0.007 | 0.994 |
| 4 | 0.095 | 0.945 | 8 | 0.006 | 1.000 |

**Figure 3**

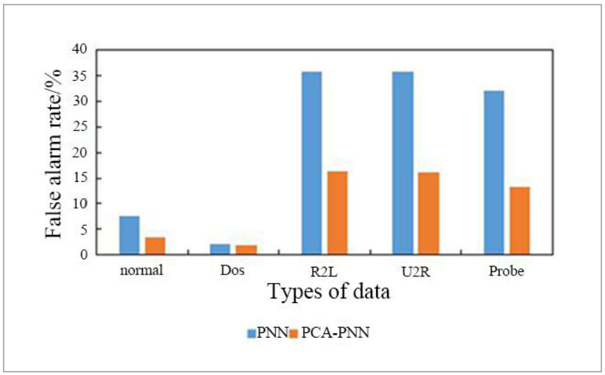
The accuracy of two models in detecting different types of data



As shown in Figure 4, the detecting false alarm rate of PNN is 7.53% for the normal data, 2.07% for Dos, 35.75% for R2L, 35.78% for U2R, and 32.15% for Probe. The detecting false alarm rate of PCA-PNN is 3.36. % for the normal data, 1.87% for Dos, 16.33% for R2L, 16.15% for U2R, and 13.25% for Probe. It can be clearly seen from the figure that PCA-PNN has lower detecting false alarm rate except in the detection of the Dos attack data.

**Figure 4**

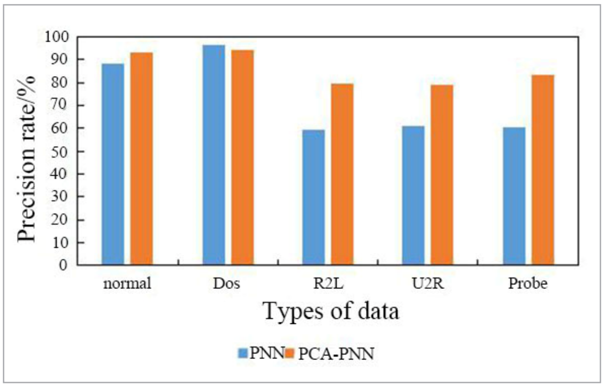
False alarm rate of two models in detecting different types of data



As shown in Figure 5, the detecting precision rate of PNN is 88.36% for the normal data, 96.45% for Dos, 59.35% for R2L, 60.75% for U2R, and 60.27% for Probe. The detecting precision rate of PCA-PNN is 93.34% for the normal data, 94.55% for Dos, 79.66% for R2L, 79.11% for U2R, and 83.35% for Probe. It can be clearly seen from the figure that PCA-PNN has higher detecting precision rate for all data except the Dos attack data. Moreover, comparing the three detecting indices of different types of data, it can be found that no matter which detection model is used to detect the normal data and Dos data, the accuracy and precision rate are higher, and the false alarm rate is lower. The reason is that there are relatively more training and test samples which are normal or belong to Dos type and the training is more. However, PCA-PNN is less affected by the number of small training samples and has stable performance compared with PNN.

**Figure 5**

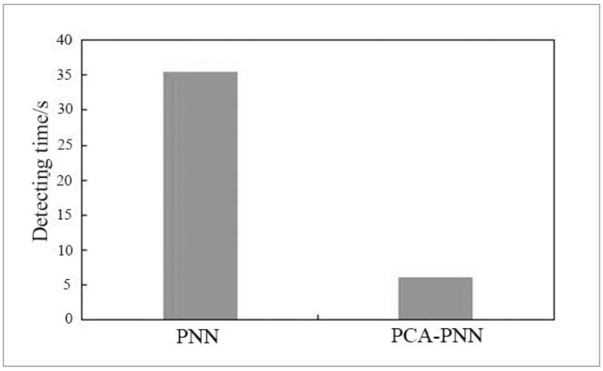
The precision rate of two models in detecting different types of data



As shown in Figure 6, the time of PNN detection is 35.38 s and the time of PCA-PNN detection is 6.13 s. It can be clearly seen from the figure that the detecting time of PCA-PNN is much shorter than that of PNN.

**Figure 6**

Detecting time of two detection models

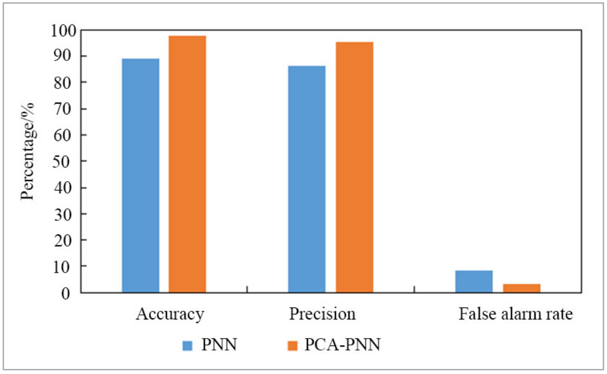


The reason is that the PCA-PNN model reduces the data from the original 122-dimension to the 6-dimension with only main features by the PCA algorithm. The computation of data is greatly reduced so that the calculation speed is improved and the detecting time is reduced.

As shown in Figure 7, in the simulation web page protection network constructed in the laboratory, the detection accuracy, precision and false alarm rate of the PNN model is 89.2%, 86.2% and 8.33% for network intrusion data; the detection accuracy, precision and false alarm rate of th PCA-PNN model is 97.6%, 95.2% and 3.33% for network intrusion data. It can be seen from Figure 7 that even in the complex web protection network, the PCA-PNN model has higher accuracy and precision and lower false alarm rate than the traditional PNN model.

**Figure 7**

The detection performance of the two detection models in website protection



## 5. Discussion

Although the rapid development of network technology greatly facilitates people's life, it also increases risks of network intrusion. The diversity and complexities of network intrusion data make the traditional passive firewall technology gradually difficult to effectively intercept dangerous data. Therefore, the more active network intrusion detection technology has been gradually widely used. The PCA-PNN model in this study mainly analyzed the probability of the type of intrusion data using the Bayesian decision theory in PNN network, so as to judge the type of intrusion data, and the PCA is used to reduce the dimension of the detected data, delete the redundant data, and improve the detection efficiency. The experimental results demonstrated that the detection accuracy and precision of the PCA-PNN model are higher than that of the traditional PNN model, and the false alarm rate is lower than that of PNN model when KDD99 data set is taken as the training set and testing set in a simple simulation environment; then in the simulation web page network environment set up in the laboratory server, the accuracy and precision of the PCA-PNN model are still higher than that of the PNN model, and the false alarm rate is lower. The reason is that the amount of network intrusion data is large, and the dimension is high. For example, the KDD99 data set used in the simulation experiment in this study has 122 dimensions. However, the feature information reflecting whether the data belongs to the intrusion data and the type of intrusion data usually is only a part, and the other dimensions have little impact. The traditional PNN model only performs simple preprocessing on the detected data  before detection; the high data dimension not only increases the calculation amount, but also interferes with the calculation of the model and increases the error. In this study, the PCA-PNN model makes PCA on the preprocessed data to obtain data dimension containing recognition features and deletes redundant data. On the one hand, it reduces the amount of computation; on the other hand, it also reduces the error of the detection model. In conclusion, the PCA-PNN model not only has less detection time, but also has higher detection accuracy and precision and lower false alarm rate.

For the PCA-PNN model, the training focuses on adjusting the parameters in the model to reduce the error as much as possible, so as to improve the performance of the detection model. PSO evolutionary algorithm can improve the performance of detection model. By imitating the foraging of birds in nature, PSO algorithm regards the parameters to be optimized as population particles and then iteratively evolves them to find the optimal parameters. One of the future research directions of this study is to optimize the parameters of PNN with PSO evolutionary algorithm, so as to improve the performance of detection model.

## 6. Conclusion

This paper briefly introduces a probabilistic neural network and PCA and combines them for network intrusion data detection. Then, the simulation analysis of PNN and PCA-PNN is performed in MATLAB software. The results are as follows: (1) the contribution rate and cumulative contribution rate of the new dimension is obtained through PCA algorithm, and the 122-dimensional original data is reduced to the 6-dimensional data; (2) different types of data have different accuracy, precision rate and false alarm rate; compared with PNN, PCA-PNN have higher accuracy and precision rate, and lower false alarm rate; moreover, PCA-PNN maintains better detecting performance than PNN when there are few training samples; (3) The detecting time of PNN is 35.38 s and that of PCA-PNN is 6.13 s, indicating that PCA-PNN can identify intrusion data more quickly; (4) in the simulated web page protection network, the accuracy and presion of the PCA-PNN model is also higher, and the false alarm rate is lower.

## References

1. Fossaceca, J. M., Mazzuchi, T. A., Sarkani, S. MARK-ELM: Application of a Novel Multiple Kernel Learning Framework for Improving the Robustness of Network Intrusion Detection. Expert Systems with Applications, 2015, 42(8), 4062-4080. https://doi.org/10.1016/j.eswa.2014.12.040
2. Gautam, S. K., Om, H. Computational Neural Network Regression Model for Host Based Intrusion Detection System. Perspectives in Science, 2016, 8(C), 93-95. https://doi.org/10.1016/j.pisc.2016.04.005
3. Hamid, Y., Shah, F. A., Sugumaran, M. Wavelet Neural Network Model for Network Intrusion Detection System. International Journal of Information Technology, 2018, 1-13. https://doi.org/10.1007/s41870-018-0225-x
4. Hayashi, A., Tokusashi, Y., Matsutani, H. A Line Rate Outlier Filtering FPGA NIC using 10GbE Interface. Acm Sigarch Computer Architecture News, 2016, 43(4), 22-27. https://doi.org/10.1145/2927964.2927969
5. Hodo, E., Bellekens, X., Hamilton, A., Dubouilh, P. L., Iorkyase, E., Tachtatzis, C., Atkinson, R. C. Threat Analysis of IoT Networks using Artificial Neural Network Intrusion Detection System. Tetrahedron Letters, 2017, 42(39), 6865-6867. https://doi.org/10.1109/ISNCC.2016.7746067
6. Hoz, E. D. L., Hoz, E. D. L., Ortiz, A., Ortega, J., Prieto, B. PCA Filtering and Probabilistic SOM for Network Intrusion Detection. Neurocomputing, 2015, 164, 71-81. https://doi.org/10.1016/j.neucom.2014.09.083
7. Kang, M. J., Kang, J. W. Intrusion Detection System using Deep Neural Network for In-vehicle Network Security. Plos One, 2016, 11(6), e0155781. https://doi.org/10.1371/journal.pone.0155781
8. Kevric. J., Jukic, S., Subasi, A. An Effective Combining Classifier Approach using Tree Algorithms for Network Intrusion Detection. Neural Computing & Applications, 2016, 1-8. https://doi.org/10.1007/s00521-016-2418-1
9. Raman, M. R. G., Somu, N., Kirthivasan, K., Sriram, V. S. S. A Hypergraph and Arithmetic Residue-based Probabilistic Neural Network for Classification in Intrusion Detection Systems. Neural Networks, 2017, 92, S0893608017300333. https://doi.org/10.1016/j.neunet. 2017.01.012
10. Ronao, C. A., Cho, S. B. Anomalous Query Access Detection in RBAC-administered Databases with Random Forest and PCA. Information Sciences, 2016, 369, 238-250. https://doi.org/10.1016/j.ins.2016.06.038
11. Shenfield, A., Day, D., Ayesh, A. Intelligent Intrusion Detection Systems using Artificial Neural Networks. ICT Express, 2018, S2405959518300493. https://doi.org/10.1016/j.icte.2018.04.003
12. Singh, R., Kumar, H., Singla, R. K. An Intrusion Detection System Using Network Traffic Profiling and Online Sequential Extreme Learning Machine. Expert Systems with Applications, 2015, 42(22), 8609-8624. https://doi.org/10.1016/j.eswa.2015.07.015
13. Singh, R., Kumar, H., Singla, R. K. An Intrusion Detection System using Network Traffic Profiling and Online Sequential Extreme Learning Machine. Expert Systems with Applications, 2015, 42(22), 8609-8624. https://doi.org/10.1016/j.eswa.2015.07.015
14. Staudemeyer, R. C. Applying Long Short-Term Memory Recurrent Neural Networks to Intrusion Detection. South African Computer Journal, 2015, 56(1). https://doi.org/10.18489/sacj.v56i1.248
15. Weller-Fahy, D. J., Borghetti, B. J., Sodemann, A. A. A Survey of Distance and Similarity Measures Used Within Network Intrusion Anomaly Detection. IEEE Communications Surveys & Tutorials, 2015, 17(1), 70-91. https://doi.org/10.1109/COMST.2014.2336610