Research on Intrusion Detection Model Based on improved CPN algorithm

Jin-guang Luo¹, Biao He² and Jinyang Lv²,*

¹ School of Information Engineering, Guanxi Polytechnic of Construction, Nanning, China
² Guilin University of Technology at Nanning, Nanning, China

*Corresponding author e-mail: lvjinyanggolden@163.com

Abstract. In this paper, the clustering analysis method of intrusion detection technology is studied and analysed combined with the algorithm of Counter Propagation Networks. Based on the traditional Counter Propagation Neural Networks (CPN), this paper takes advantage of Gene Express Programming (GEP), optimize the input vector and α of the CPN network model. It overcomes the shortcoming of the network jitter, that because of the influence of the input vector on the connection weight of the Kohonen layer in the traditional CPN network algorithm. improves the convergence time of the network, and obtains a better network clustering effect, and verifies the effectiveness and superiority of the algorithm in the intrusion data analysis from the experiment.

1. Introduction
The audit records, system logs, network packets and monitoring packets in all kinds of network are all the basic factors of intrusion detection. To ensure the security of the system and the security of the network information, the intrusion detection system (IDS) will analyze the behavior data and find the suspicious attack behavior. In recent years, by applying various methods of data mining technology to IDS [1,2,3], the intrusion detection system is more intelligent in information processing and improves the detection accuracy of the system itself.

To quickly analyze and discover unknown knowledge and rules from big data, intrusion detection based on clustering has achieved this goal. The intrusion detection algorithm based on clustering method doesn’t require any data recording in advance. It can also have a high detection rate while maintaining a low error rate. Such as K-means [4,5] clustering algorithm, hierarchical clustering algorithm, SOM algorithm and FCM algorithm[6], in these traditional methods, often make the result fall into the local optimal solution, or the clustering results are unstable, or the quality of the clustering is limited, and even the processing time is long. In the anomaly detection system based on Neural Network [7,8], such as CPN, although its network operation accuracy is high, its computation is too large, which leads to too long time, is not conducive to the real-time requirements of detection. Therefore, a new improvement method is proposed in this paper, which is the introduction of gene expression programming to optimize the input vector of the Kohonen layer and the connection weight value α. The purpose is to overcome this shortcoming, reduce the network running time and improve the network training speed.

The rest of this paper is organized as follows: the second mainly analyzes the CPN network model and points out its shortcomings and improves it by introducing the GEP algorithm; the third applies the new method to intrusion detection, and studies the intrusion detection design model based on the
improved CPN algorithm; in the fourth, the actual intrusion detection model is implemented. It proves the effectiveness of the new algorithm in intrusion detection system. Finally, it puts forward the next research direction by summarizing.

2. Improved CPN Algorithm

2.1. Prior Knowledge

The CPN network model \cite{9,10,11} is a mapping neural network using the approximate function of self-organizing mapping. Through the competition rule of "the strong first, the weak exits", it uses the pre-given similarity rules and the decision function to judge which neuron is closest to the input vector, and chooses the neuron with the maximum output value as "winner", as shown in Figure 1. The most important is how to ensure an input vector($X$) has minimal impact on $W_j$ (CPN network connection weight). And the purpose is to avoid the network jitter in training. Because if the $W_j$ is greatly influenced by one $X$ in later period of training, then $X$ will greatly breach $W_j$ which can express other input vectors. When these vectors are reappeared in the next iteration, it will make a greater adjustment to the $W_j$, the show go on, which will cause the network to fall into jitter. It not only affects the stability of the network, but also affects the loss time caused by the excessive computation of the network itself, making it bigger.

![Figure 1. CPN network model](image)

In the CPN network model, the connection weights of the neurons from the input layer to the competition layer are adjusted, and the adjustment function is:

$$X - W_j^{new} = (1 - \alpha)(X - W_j^{old}) \quad (1)$$

Where the value of $\alpha$ is the neural network learning rate, and $\alpha \in (0,1)$. From the above, we can see that the Kohonen layer in the traditional CPN network, according to equation (1) to update the network connection right $W_j$, can be seen from the formula, if $\alpha$ is 1, it will lead to a wrong class, perhaps get some "abnormal" categories, some too small, some too large.

This experiment shows that if the traditional CPN network is used to analyze the intrusion detection data, when $\alpha = 0.5\sim0.7$ (this data is the empirical value taken in the initial training period of the traditional CPN algorithm, it will become smaller as the training progresses. It shows that it is a linear change.) It will be unsatisfactory to obtain the analysis results, and the network training time is also large, and does not meet the real-time requirements of intrusion detection. For a detailed analysis, see part 4 of this article. There are two reasons for this. One is that the input vector $X$ has a large impact on $W$, and the amount of calculation is too large. Second, the value of $\alpha$ has a large impact on the analysis of intrusion detection data, that network jitter occurs and does not meet intrusion detection. Real-time requirements do not find the best match based on real-time requirements. In this paper, through the introduction of Gene Expression Programming (GEP) \cite{12,13,14} algorithm, the reduced input vector is optimized in advance, so that the optimal $\alpha$ in the intrusion detection model is obtained, and a representative vector sample set is taken out from the sample. Pairs are trained, then the sample set is expanded, and the expanded sample set is used for network training to complete the training of the Kohonen layer, thereby presenting an improved CPN algorithm.
2.2. GEP optimizes the input vector and $\alpha$

The global optimality of the GEP algorithm is used to optimize the input vector and find the optimal initial value for $\alpha$. Its purpose is to avoid network jitter and prevent excessive computation when intrusion detection data is analyzed, thus achieving real-time requirements for intrusion detection.

From the analysis of the previous section, the main idea of the new algorithm is to use the advantages of the GEP algorithm to perform rough classification for the input vector, and at the same time, determine the initial value of $\alpha$ according to the actual situation of the intrusion detection data. Specific implementation steps of: The first step is to preprocess the input data and use the Vector Space Model (VSM) [15,16,17] method to represent the data text structure. Second, the GEP algorithm is used to reduce the text information and get the feature vector set. Because the entire algorithm model is based on the most representative vector composition sample set as the start of training, and then expand the sample set accordingly. Again, using the sample set for network training, and so on, the training of the Kohonen layer in the CPN network is completed. Therefore, this paper uses GEP's global optimization and the advantage of faster convergence rate than the genetic algorithm to find several representative vector samples. As a starting training sample set. The most important step is to find a suitable fitness function, to achieve rapid classification of the entire input vector without affecting the overall algorithm time and performance. In fact, the input vector is optimized by the GEP algorithm to find some intrusion detection data with obvious attack characteristics, so that the data is used for the first training. After the iteration starts, other vectors are gradually added. With the training progresses, the adjustment of $\alpha$ is started. That is, once it is expanded, it is adjusted once. It can be experienced from the experiment that this idea is feasible, and because the input vector has a destination to start training, it can avoid the impact of a single input vector on the connection weight $W$ in the late stage of network training.

This paper will use the method of Literature [18] to determine the fitness function of the new algorithm, which is to improve the fitness function to adapt to the data analysis in the intrusion detection environment by using the nonlinear order selection mechanism. The following constraints is:

\[ C(X_i) = \beta (1 - \beta)^{r_i - 1} \]  

In equation 2, $r_i$ is the sort number of chromosome $X_i$, $\beta$ is the "selection pressure" of the implicit algorithm, between 0 and 1.

To find the best individuals, the fitness function is $1/C(X_i)$, which means that the smaller $C(X_i)$ is, the greater the probability that the individual is selected. The value of $C(X_i)$ is the average of the constraint values of Equation 2. Algorithm 1 implements the functionality of the improved algorithm (as shown in Table 1).

| Table 1. ICPN Algorithm: GEP optimizes the input vector and $\alpha$ |
|---------------------------------------------------------------|
| Input: text vectors                                           |
| Output: Representative input vector sample set                |

(1) Read(); // input text vectors  
(2) Determine the initial value of $\alpha$; // The initial value of $\alpha$ is determined based on the actual situation of the intrusion detection data.  
(3) Call VSM(); // to represent the data text structure  
(4) $< X_1, X_2, X_3, \ldots, X_i, \alpha >$; //Chromosome design  
(5) initial(); //Initialize the population, size is 70;  
(6) he fitness function is $1/C(X_i)$;  
(7) GEP();//Genetic manipulation produces the next generation.  
    Selection(); Replication(); Mutation();  
    Transposition(); // Mainly for the transposition of the $\alpha$ domain, the literature [7] details the transposition method for the domain;  
    Recombination(); Inversion();}  
(8) If the maximum number of iterations is reached, the result is output, otherwise it returns (7) to continue the operation.
3. Achievement and Analysis of Model

The design of the IDM based on the new algorithm (IDM-ICPN) is shown in Figure 2. The design is mainly divided into four modules: data collection, data processing, CPN algorithm clustering and intrusion response module.

(1) data collection module. The data packets obtained by the software are processed, and then different analyzers are called to perform semantic analysis on the data packets to obtain many data sequences, wherein each record contains multiple attributes, including IP information.

(2) data preprocessing module. The collected data is converted into an input vector mode conforming to the CPN neural network, which is used by The VSM method. After completion, it is saved as an input format of the CPN neural network, which facilitates input training.

(3) ICPN Module. After receiving the input vector, the input vector is first optimized by using the GEP algorithm, and the data is trained using the basic idea of the ICPN algorithm. After several trainings, the neural network using the clustering function in the ICPN algorithm can accurately identify the intrusion behavior and realize the identification of intrusion behavior and normal behavior as much as possible. During the detection phase, the determined abnormal data is sent to the intrusion response module.

(4) Intrusion response module. The user detects the current user behavior by outputting the results from the neural network. According to the result of the detection, the corresponding security action is carried out. If one of the records belongs to the intrusion behavior, the warning or termination of the connection is performed accordingly, and the network security administrator is notified to perform the processing.

![Figure 2: IDM-ICPN](image)

4. The results and analysis of the experiment

To realize the effectiveness of the improved CPN neural network algorithm on the intrusion detection system, the performance of the algorithm is measured by two parameters: universal detection rate (UDR) and false positive rate (FPR). Its definition is as follows:

\[ UDR = \frac{\delta}{\emptyset} \quad (3) \]
\[ FPR = \frac{\mu}{\rho} \quad (4) \]

In (3) and (4), \( \delta \) is the number of intrusion detection records detected by the algorithm and \( \emptyset \) is the total number of intrusion records. \( \mu \) is the total number of normal intrusion records misclassified, \( \rho \) is the number of normal records in the dataset.
Since these two parameters can very intuitively represent the detection and misclassification of the algorithm, they can well detect and verify the performance of the algorithm. The data set used in the experiments is the NSL-KDD data set [19,20]. To perform our experiments, we randomly created 5 smaller subsets of the NSL-KDD train set each of which included 18,000 records of information. To facilitate the simulation experiment, the data type is first standardized by VSM method, and the vector set sample is formed.

4.1 Results and analysis

First, set the population operating parameters (as shown in Table 2).

Table 2. the population operating parameters

| Operating Times (OT) | 1000 | Iterative Times (IT) | 70 |
|----------------------|------|----------------------|----|
| Population Size (PS) | 70   | Gene-Head-Length (GHL) | 4 |
| Mutation Rate (MR)   | 0.05 | Single Recombination Rate (SR) | 0.1 |
| Weight-Variation Rate (WV) | 0.002 | Two-point Recombination Rate (TR) | 0.1 |
| IS                   | 0.1  | RIS                  | 0.1 |
| The Input Layer (IL) | 16   | The Output Layer (OL) | 6 |

To evaluate the performance of the ICPN algorithm in UDR, FPR and time, this paper uses the method of comparing ICPN with CPN and the BP neural network algorithm proposed in [5]. The experiment was implemented in C# language and the running platform was Visual Studio 2016. The experimental results are shown in Figure 3 to Figure 6. They respectively show the UDR and the FPR. Where, Figures 3 and 4 show the data comparison of the second subset. Figures 5 and 6 show the data comparison of the experimental subsets. In the figure, ICPN is the improved algorithm of this paper.

![Figure 3. Comparison of UDR (the second subset)](image)

![Figure 4. Comparison of FPR (the second subset)](image)

![Figure 5. Comparison of UDR (5 smaller subsets)](image)

From Figure 3, under the example of 1000 to 18000 in the second subset, the detection rate of the new algorithm is higher, and the number of instances does not affect the impact of the new algorithm on the analysis of intrusion data, which basically tends to a stable range. The other two methods, as the data record increases, the UDR is correspondingly reduced. And it can be seen from the figure that when the data record is increased from 1000 to 18000, the traditional CPN and BP algorithms are reduced by 18.2% and 11.44%, respectively, and the reduction is too large, resulting in poor results.
The reason for these phenomena is that the CPN algorithm does not optimize the input vector and does not use the traditional method to standardize. This will cause the network to calculate too much and easily fall into the network jitter, resulting in the loss of correctness of the training results. The BP algorithm of the literature [8] only increases the momentum term and improves the convergence speed to a certain extent, but it has limited generalization ability. As the amount of data increases, the universal applicability of the network cannot be well reflected. The ICPN algorithm in this paper proposes a solution to these problems and can be confirmed from experiments.

From the comparison of the FPR in Figure 4, although the false rate of the three algorithms increases with the increase of the recorded data. This is because when the data has many ambiguous records, the algorithm itself is not well differentiated, which will lead to an increase in the FPR. This problem needs further study. We can also see from Figure 5 and Figure 6, the ICPN algorithm is superior to the other two algorithms.

In short, the ICPN algorithm proposed in this paper has a lower FPR than the other two algorithms under the same data volume. In addition, from the experiment, by the time of recording, the time of the ICPN algorithm is 48.86% and 57.56% higher than that of CPN and BP respectively. It can be seen that the ICPN algorithm is more in line with the requirements of real-time intrusion detection. This is because the new algorithm uses GEP to optimize the input vector and α, which improves the training ability of the network to a certain extent and achieves convergence faster.

5. Conclusion and Future Works

The paper proposes an intrusion detection algorithm for improved CPN (ICPN). The experimental results also show that the ICPN algorithm has good time saving and effectiveness and compares it with other algorithms from two parameters: detection rate and false positive rate. The ICPN is superior to the other two algorithms. This shows that to a certain extent, it overcomes the shortcomings of the influence of a single input vector on the connection weight $W_j$ in the later stage of the CPN network.

In the experiment, the FPR increases as the amount of data increases, and the increase is too large from a certain range. It also appears in the new algorithm. The next step will be to explore how to overcome this shortcoming and enable it to implement the early warning utility of IDS in complex environments.

Acknowledgments

This research is sponsored by the Institute of Science Computing and Intelligent Information Processing, and supported by the project that is the Project of Technology Planning Project of Guangxi of China (Guangxi Special Foundation “Technology base and talent”, No. AD16380097).

References

[1] Solane Duque, Mohd. Nizam bin Omar. Using Data Mining Algorithms for Developing a Model for Intrusion Detection System (IDS). Procedia Computer Science. vol09(2015)46-51.
[2] Wathiq Laftah Al-Yaseen,Zulaiha Ali Othman,Mohd Zakree Ahmad Nazi. Real-time multi-agent system for an adaptive intrusion detection system[J]. Pattern Recognition Letters,2017,85.
[3] Yanfei Liu,Jieling Wang,Ke Jiang,Yanhui He,Zhilin Wu. Research on regional intrusion prevention and control system based on target tracking[J]. Journal of Physics: Conference Series,2017,887(1).
[4] D. Raja Kishor; N. B. Venkateswarlu. Hybridization of Expectation-Maximization and K-Means Algorithms for Better Clustering Performance. Cybernetics and Information Technologies. vol16 (2016) 16-34.
[5] Shahri B.M.A., Zadeh S.K., Adeyemi I.R., Zainal A. Advances in Computational Science, Engineering and Information Technology. Advances in Intelligent Systems and Computing. Nagamalai D., Kumar A., Annamalai A. (eds). Springer, Heidelberg. vol 225(2013)p 307-316.
[6] Wei Jia Lu,Zhuang Zhi Yan. Improved FCM Algorithm Based on K-Means and Granular Computing[J]. Journal of Intelligent Systems,2015,24(2).
[7] Mohamed M. Abd-Eldayem. A proposed HTTP service based IDS. Egyptian Informatics Journal. vol15(2014) 13-34.

[8] ZHANG Ting-ting, ZHAO Jing-sheng. The research of Intrusion Detection System based on Neural Network. Network & Computer Security. vol08(2010.)25-29.

[9] Anitha J., Jude Hemanth D. Emerging Trends in Computing and Communication. Lecture Notes in Electrical Engineering, Sengupta S., Das K., Khan G. (eds). Springer, New Delhi. vol 298(2014) p63-69.

[10] J. Anitha, D. Jude Hemanth. A Weighted Counter Propagation Neural Network for Abnormal Retinal Image Classification[M]. Springer India:2014-06-15.

[11] Mohadeseh Zarei Ghoebadi, Mohsen Kompany-Zareh. Application of supervised Kohonen map and counter propagation neural network for classification of nucleic acid structures based on their circular dichroism spectra[J]. Spectrochimica Acta Part A: Molecular and Biomolecular Spectroscopy, 2014,132.

[12] Ferreira C. Gene expression programming: a new adaptive algorithm for solving problems. Complex System, vol13(2001) 87-129.

[13] Weimin Shi, Yang Liu, Wei Kong, Qi Shen. Tea Classification by Near Infrared Spectroscopy with Projection Discriminant Analysis and Gene Expression Programming[J]. Analytical Letters, 2015, 48(18).

[14] Zhaohui Jiang, Tingting Li, Wenfang Min, Zhao Qi, Yuan Rao. Fuzzy c-means clustering based on weights and gene expression programming[J]. Pattern Recognition Letters, 2017, 90.

[15] Ch. Aswani Kumar; M. Radvansky; J. Annapurna. Analysis of a Vector Space Model, Latent Semantic Indexing and Formal Concept Analysis for Information Retrieval Cybernetics and Information Technologies, vol12(2012) 34-48.

[16] Hiram Calvo, Oscar Méndez, Marco A. Moreno-Armendáriz. Integrated concept blending with vector space models[J]. Computer Speech & Language, 2016, 40.

[17] K. Rajan, V. Ramalingam, M. Ganesan, S. Palanivel, B. Palaniappan. Automatic classification of Tamil documents using vector space model and artificial neural network[J]. Expert Systems With Applications, 2009, 36(8).

[18] Xiao Qin, Chang’an Yuan, Yuzhong Peng, Chao Ding. A GEP-Based Text Classification Algorithm. Journal of Computational Information Systems. JOURNAL OF INFORMATION & COMPUTATIONAL SCIENCE. Vol 6 (2009) 1303-1309.

[19] Information on http://www.unb.ca/cic/datasets/nsl.html.

[20] Tavallaee, M., Bagheri, E., Lu, W., & Ghorbani, A. A. A detailed analysis of the KDD CUP 99 data set. IEEE International Conference on Computational Intelligence for Security & Defense Applications, (2009) p 1-6.