Detection of DOS Flooding Attacks with an Improved Growing Hierarchical SOM

Xiaofei Qu[1, 2, a], Mu Li[1, 2, b]

[1]Command and Control Engineering College, Army Engineering University of PLA, Beijing, China

[2]Institute of Systems Engineering, Academy of Military Sciences PLA, Beijing, China

[a]crane0106@163.com,[b]2581602139@qq.com

Keywords: DOS flooding attacks; Growing self-organizing map; Growing hierarchical self-organizing map; Improved growing hierarchical SOM

Abstract. In this paper, an improved growing hierarchical self-organizing map (IGHSOM) approach based on growing self-organizing mapping (GSOM) is proposed to detect the DOS flooding attacks. The IGHSOM is a layered architecture, which can be extended from both horizontal and vertical, and utilized to represent the topological relation of data space and the hierarchical relation of data. Compared with the traditional growing hierarchical self-organizing mapping (GHSOM) approach, which has the potential disadvantage of inaccurate mapping of data topological relationships on DOS flooding attack detection, the proposed scheme can accurately represent the topological relationship of data space, increase the DOS detection rate and then reduce the false alarm rate. Through the numerical experiments on KDD data, the results show that the proposed IGHSOM approach can achieve better performance than traditional GHSOM in terms of DOS flooding attack detection, and can further improve the detection rate and reduce the false positive rate.

Introduction

A denial of service attack is an attack that a user occupies a large number of shared resources, leaving the system with no remaining resources for other users. In recent years, DOS flooding attacks are on the rise, and solving DOS flooding attacks becomes a top priority for network security. Many scholars and research institutions have focused on SOM-based DOS flooding attacks detection [1]-[2]. The growing hierarchical self-organizing map (GHSOM) is a typical SOM model, which is used to detect DOS flooding attacks [3]. The GHSOM is a dynamic architecture, which is proposed for the SOM static architecture. The architecture of the GHSOM model is composed of several SOMs arranged in layers, where the whole architecture (number of layers, maps, and neurons) is established during the training process depending on the input data and mirroring their inherent structure [4]. But the GHSOM expands the rules of neurons in the horizontal direction, which generates redundant neurons, bring computational burden, and in return affect the accuracy for data clustering.

This paper describes a modified version of the GHSOM algorithm, which is an improved growth hierarchical self-organizing map (IGHSOM). The IGHSOM is hierarchical architecture, which expands neurons horizontally and vertically. Implementing neuron expansion in the horizontal direction using a special GSOM principle [5]. The main contributions of this paper are worth emphasizing as follows:

- Firstly, we proposed an improved growing hierarchical SOM(IGHSOM) based on a special GSOM to implement DOS flooding attacks detection. The IGHSOM can fully express the topological relationship between data to reduce the false positives rate and reduces the computational burden;
- Secondly, we use the open source datasets to evaluate the performance of the proposed method to show its high accuracy and adaptability.

  DOI: 10.25236/icemit.2019.049

The remaining parts of the paper are organized as follow:

Section II briefly reviews the related work. The details of the special GSOM and IGHSOM algorithms are described in Section III. Section IV reviews the evaluation methodology. Section V presents the experimental results of IGHSOM and traditional GHSOM in network DOS flooding attack detection. Section VI will present the conclusions and possible future aspects of this work.

Related Works

In recent years, there have been many methods to detect DOS flooding attacks, such as machine learning method, neural network method, random forest method, etc. There is a shallow neural network called self-organizing map (SOM) that is widely used in intrusion detection. SOM is capable of clustering, self-organization, self-learning, and visualization. Growing hierarchical self-organizing mapping (GHSOM) based on SOM is an unsupervised learning mechanism and has been used as a tool for intrusion detection [3]. E.J. Palomo proposed to introduce GHSOM into network-based intrusion detection [6], which can deal with the limitation of SOMs related to their static architecture. The architecture of the GHSOM model is composed of several SOMs arranged in layers, where the whole architecture (number of layers, maps, and neurons) is established during the training process depending on the input data and mirroring their inherent structure. However, although GHSOM achieved 97.59% detection rate in DOS flooding attacks detection [7], it still brings some redundant calculation in the growth process of GHSOM, because GHSOM inserted a row of neurons or a column of neurons at a time in the growth process, which brought computing burden and inaccurate topological representation. Sometimes it did not need so many neurons, only one was enough.

The root cause of all these problems is that the imbalance of network traffic data leads to large quantization errors and the insertion strategy of neurons will also introduce errors.

In [5], a dynamic incremental variant of SOM, called growing SOM (GSOM), was proposed to provide a more elastic structure and reduce the training time. The process for learning and adding new neurons in GSOM algorithm is still computationally intensive and time-consuming, which may limit the applicability of GSOM in many time-critical problems. In [9], the authors modified the GSOM algorithm to force the batch learning principle and shorten processing time, while maintaining a high quality of result mapping of large input data sets.

In order to eliminate these problems, GHSOM algorithm and GSOM are modified. An improved GHSOM (IGHSOM) is proposed. The next section will describe IGHSOM algorithm in detail.

Methods

An improved growing hierarchical self-organizing map (IGHSOM) is a hierarchical architecture, and the growth of neurons is from both vertical and horizontal directions, which is the same as GHSOM. Different from GHSOM, IGHSOM adopts DBGSOM for the horizontal growth.

A. Horizontal Growth of IGHSOM:

The growth strategy of GSOM is to insert all the free positions in the boundary neurons, which may lead to redundant neurons and redundant computation. To overcome this problem, a directed batch-growing SOM (DBGSOM) [10] is utilized to enhance the data topology and decrease the redundant, in which only one neuron is inserted at a time in the appropriate place. The DBGSOM has better topology preservation ability and a lower susceptibility for twisting and tangling, which will guide the network growth in an appropriate way within the feature space by considering the cumulative error around the candidate boundary neurons. Moreover, The DBGSOM can also dynamically add neurons to the network during the training procedure, while preserving a grid structure to yield a proper feature map without requiring pre-specification of the network size at the initialization step. This DBGSOM algorithm includes two important stages, i.e., the growth stage of new neurons and the initialization stage of the weight for new inserted neurons.

In Fig. 1, the gray hexagon is the neighbor neurons (NB), and the black is the boundary neurons (BN). CE_p is the cumulative error of NB_p (p = 1,2,3).

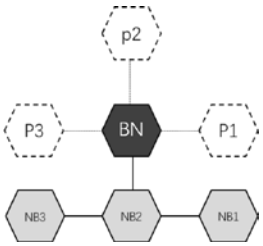

Fig. 1. New node growth in the DBGSOM: (a) If CE2 > CE1 and CE2 > CE3, the new neuron is inserted p_2. (b) If CE1 is greater than CE2, p_2 is an insert position. (c) If CE3 is greater than CE2, p_3 is an insert position.

(a) In the growth stage of new neuron

The winner node p is determined by the formula

$$p = \arg\min(\| x_i - w_p \|) \tag{1}$$

where $\| x_i - w_p \|$ is the distance between the input x_i and the weight vector of node p. The cumulative error (CE_p) of each neuron is calculated as follows:

$$CE_p = \sum_{i=1}^{k} \| x_i - w_p \| \qquad k = 1, 2, ..., n \tag{2}$$

where w_p is the weight vector of the neuron *p*, and *k* is the number of input vectors mapped on the neuron *p*.

As the data is continuously input, the cumulative error CE_p is also increasing. When the error reaches the threshold GT (eq. (3)), new neurons need to be added to generate new clusters. [10] introduced the 3p,2p and 1p principles of neuron insertion. Fig. 1 shows the 3p rule of neuron insertion. GT is defined as follows:

$$GT = - D*\ln(SF) \tag{3}$$

where D represents the dimension of the data vector,

$SF \in (0,1)$ is the spreading factor, which controls the growth of neurons.

(b) Weight initialization of new neuron

DBGSOM provides an appropriate mechanism to find the appropriate growth location while also providing a method for assigning initial weight vectors to new neurons. By selecting the appropriate weight vector for the new neuron, the smoothness of the grid can be achieved. The new weight vector is preferably partially matched to the weight vector of the neighboring neurons, which can reduce the false mapping. [10] introduced the 3w,2w and 1w initialization principles of new neurons. For Fig. 1, the weight initialization rule (3w) can be defined as follow:

$$w_{new} = \begin{cases} [(2w_{bn} - w_{nb2}) + w_{nbi}]/2 & p_i \text{ is } inserted \ p_i \neq 2 \\ 2w_{bn} - w_{nb2} & p_2 \ is \ inserted \end{cases} \tag{4}$$

where w_{bn} is expandable neuron weight, w_{nbi} is weight of neighbor neuron *i* of the selected insertion position. For example, as shown in Fig. 1, if p_2 is selected, then the value of the weight w_{new2} is equal to $2w_{bn} - w_{nb2}$, otherwise, if p_1 or p_3 is selected, then w_{new1} is equal to $[(2w_{bn} - w_{nb2}) + w_{nbi}]/2$.

B. Vertical Growth of IGHSOM:

After the growth of horizontal neurons of IGHSOM, the data sets containing DOS flooding attacks should be preliminarily classified as some clustering. Ideally, each cluster should contain the same type of data, either normal or attack data. However, due to the unbalanced and random distribution of data, it is difficult to accurately distinguish normal data from abnormal data and also difficult to find a general method for accurately dividing data. There are still cases where normal data and abnormal data cannot be separated. Normal data mixed with abnormal data will have a large cumulative error. So, the size of the cumulative error can be used to determine which clusters need further clustering. The cluster with large cumulative error will be clustered again to improve the detection accuracy.

As shown in Fig. 2, it is a 4-layer IGHSOM architecture. After all the data in the first layer is trained by DBGSOM, it is initially divided into four clusters (each black dot represents a cluster), and the cumulative error of each cluster is also obtained (eq. (2)). According to eq. (5), it is judged whether IGHSOM will increase the number of layers; if eq. (5) is not satisfied, the layer will be increased.

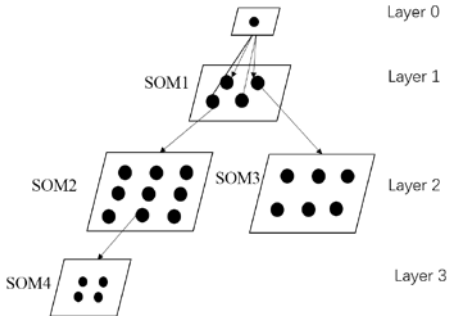

Fig. 2. IGHSOM architecture

$$CE_i < \lambda \cdot CE_0 \tag{5}$$

where $CE_0$is the cumulative error of the top layer neuron calculated from eq. (2) and $\lambda \in [0,1]$ is the parameter which controls the growth of the IGHSOM network layer. If neuron *i* does not satisfy (5), it will automatically expand to establish a sub-layer with 2 × 2 SOM mapping networks and continue DBGSOM on the subnet to achieve adaptive growth. The algorithm of IGHSOM is shown in algorithm 1 below.

Algorithm 1 IGHSOM algorithm

1: Initialization phase

Initialize four neurons in a square topology with random weights; SF = 0.9, λ = 0.5, the number of cycles in the training

is 100;

2: Horizontal growth phase

Calculate the growing threshold GT according to SF (eq.

(3));

3: for i = 1 to 100 do

4: Set the cumulative error (*CE*) of the all neurons to zero;

Enter all sample data X;

Calculate winning neuron using Euclidean distance;

Update the weight vectors of all neurons;

Calculate the cumulative error (*CE*) of the winning neurons;

5: for all non-boundary neurons do

6: If CE_i> GT, distribute the CE_i as follow:

$$CE_{winner}(t+1) = \frac{CE_{winner}(t)}{2}$$

$$CE_{neighbors}(t+1) = CE_{neighbors}(t) + \frac{CE_{winner}(t)}{2N}$$

where N is the number of neighbors of the winner neuron.

7: end for

8: for all boundary neurons which CE_i> GT, do

9: Calculate the number of free positions around the neuron i

10: if number of free positions is 3 then

11: Decide the position of the new neuron by rule 3p, and assign the weight vector of the new neuron by rule 3w

12: end if

13: if number of free positions is 2 then

14: Decide the position of the new neuron by rule 2p, and assign the weight vector of the new neuron by rule 2w

15: end if

16: if number of free positions is 1 then

17: Decide the position of the new neuron by rule 1p , and assign the weight vector of the new neuron by rule 1w

18: end if

19: end for

20: end for

21: return the error set ($CE=CE_0,CE_1,CE_2,...CE_n$) of winning neurons

22: Vertical growth stage

23: for i = 1 to length (*CE*) do

24: if nonconformity eq. (5) then

25: The DBGSOM operation is continued on the input vector set X_i, which belongs to the winning neuron, which is the same as step 3-21.

26: end if

27: end for

28: Repeat till the epoch is satisfied.

Evaluation Methodology

C. Dataset

KDD99 Dataset [12]: KDD is divided into 4 categories and there are 39 attack types, among which 22 attack types appear in the training set and another 17 unknown attack types appear in the test set. KDD99 contains four types of attacks, which are DOS (denial-of-service), R2L

(unauthorized access from a remote machine to a local machine) and U2R (unauthorized access to local super user by a local unprivileged user). PROB (surveillance and probing) and normal data. Fig. 3 is three records written in CSV format in KDD99 dataset, which is 42 items. The last item is a label, which indicates the data is normal or attacked.

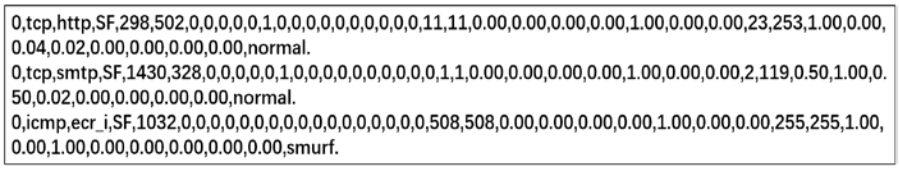

Fig. 3. Basic properties of KDD99.

D. The evaluation criteria

The performance of the classifier is evaluated based on false positives and detection rates, as defined in (7) and (8).

$$FR = \frac{Number\ of\ False\ Positives}{Total\ Number\ of\ Normal\ Connections} \qquad (7)$$

$$DR = 1 - \frac{Number\ of\ False\ Negatives}{Total\ Number\ of\ Attack\ Connections} \qquad (8)$$

The FR is false positive rate, which is the number of normal connections marked as attack, and the DR is detection rate, which is identified as the number of attacks that can be detected [11].

Results And Discussion

This section discusses the experimental results of our proposed IGHSOM method in DOS flooding attacks detection. The method was designed and tested in the MATLAB 2017a environment. The operating system is a Windows 7 Professional personal computer with an Intel Core i7-7700, 3.6GHZ CPU and 8.0 GB RAM. We experimented with the KDD dataset.

For the comprehensive experiment, set1 is 11352 data and set2 is 33,652 data that were extracted from the KDD ’99 data set and mixed with different attacks. Test set is 88950 data.

Fig. 4 is the experimental result of DBGSOM on the KDD99. Each square is a neuron that represents a type of data. The DBGSOM neuron growth rule is to insert only one neuron in the free position [10], so Fig. 4 uses only 38 neurons. The figure shows that similar types of data will be gathered together, the number is 0 for normal data, which is concentrated in one area, the number 15 represents the prob attack, which is concentrated in another area, the number 5 represents the DOS flooding attack convergence in the third Areas, etc. The red part of the figure indicates that single-layer DBGSOM still has misclassified data and cannot distinguish normal data flag 0 from DOS flooding attack data flag 5.

In Fig. 4, DBGSOM uses only 38 neurons, and except for part of DOS flooding attacks that are not distinguished from normal traffic data, the remaining attacks have been successfully clustered. The same data sample, the GHSOM uses 84 neurons. The GHSOM not only does fail to distinguish DOS flooding attacks from normal data, but other types of attacks are not distinguished from normal data. The neurons used by GHSOM are much larger than the neurons used by DBGSOM, which also increases a computational burden. In summary, DBGSOM performs better than GHSOM in DOS flooding attack detection. The quantitative analysis indicators are shown in Tab. 1.

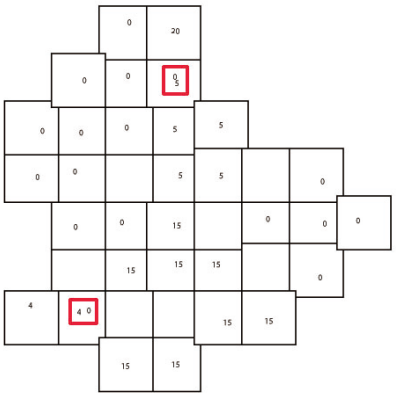

Fig. 4. The growth result of DBGSOM

TABLE I. RESULTS FOR TESTS ON TEST SET FOR DIFFERENCE METHODS TO DETECT DOS FLOODING ATTACK

Methods	***FR***	***DR***	***Train Set***
GHSOM	39.95%	60.91%	set1
DBGSOM	45.13%	100%	
IGHSOM	44.39%	100%	
GHSOM	21.51%	78.50%	set2
HSOM[13]	73.28%	89.34%	
DBGSOM	29.11%	100%	
IGHSOM	17.86%	100%	

Fig. 5 is the experimental result of IGHSOM base on Fig. 4, which shows the first layer of IGHSOM does not distinguish normal data 0 from DOS flooding attacks 4 and 5. In the second layer, DBGSOM has performed again for the corresponding data set of the neuron, and the two types of data are successfully separated.

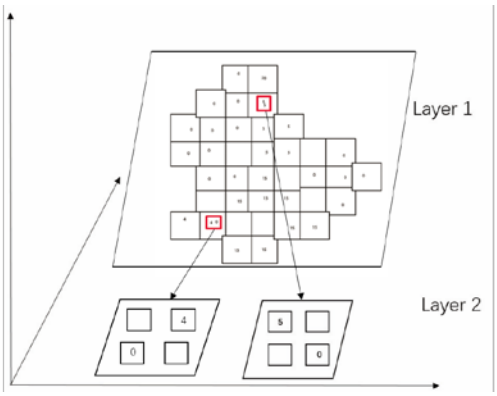

Fig. 5. The growth result of IGHSOM

Fig. 5 illustrates that IGHSOM uses only 46 neurons. Because the cumulative error of the red box in the figure does not satisfy the eq. (5), the neuron growth rule of DBGSOM is re-executed in the second layer. The data that is not distinguished by the upper layer can be distinguished at this level.

Tab. 1 shows that DBGSOM and IGHSOM methods were superior to other methods in detecting DOS flooding attacks. The detection rate of DOS flooding attacks is very high. In Tab. 1, the indicators of detection rate and false alarm rate of [13] were not ideal. However, DBGSOM and IGHSOM are much better than HSOM in both detection rate and false alarm rate.

Conclusion

In this paper, we study the feasibility and effectiveness of DBGSOM and IGHSOM for detecting DOS flood attacks. Specifically, because DBGSOM has good topology protection to accurately represent the topological relationship of unbalanced network data, it is suitable for detecting DOS flood attacks. Therefore, an improved growing hierarchical SOM (IGHSOM) based on DBGSOM is proposed to detect DOS flooding attacks. The proposed architecture implements layered processing, which can accurately represent the topology and reflects the hierarchical relationship of the data. Moreover, it can further improve the DOS flood detection rate and reduce the false positive rate. The results show that IGHSOM can achieve better performance than GHSOM and DBGSOM in detecting DOS flood attacks. The main line of future work is to use a hybrid approach to network intrusion detection, not just a simple SOM approach. For example, use SOM + FCM and SOM + artificial neural networks.

References

[1] Nam T M , Phong P H , Khoa T D , " Self-organizing map-based approaches in DDoS flooding detection using SDN[C]," IEEE 2018 International Conference on Information Networking (ICOIN) – Chiang Mai, Thailand (2018.1.10-2018.1.12)] 2018 International Conference on Information Networking (ICOIN). IEEE Computer Society, pp. 249-254, 2018.

[2] Li D , Guiqiang N , Zhisong P , "DDoS intrusion detection using generalized grey self-organizing maps[C]," IEEE International Conference on Grey Systems & Intelligent Services. IEEE Xplore, 2007.

[3] Huang S Y , Huang Y, "Network forensic analysis using growing hierarchical SOM[C]," IEEE International Conference on Data Mining Workshops. IEEE, 2014.

[4] A. Rauber, D. Merkl, M. Dittenbach, "The growing hierarchical self-organizingmap:exploratory analysis of high-dimensional data," IEEE-Trans.Neural Networks, vol. 13, pp. 1331-1341, 2002.

[5] D. Alahakoon, S.k. Halgamuge, B. Srinivasan, "Dynamic self-organizing maps with controlled growth for knowledge discover, " Neural Networks, IEEE Transactions on , vol. 10, pp. 601-614, 2000.

[6] E. J. Palomo, E. Domlnguez, R. M. Luque, J. Munoz , " Network security using growing hierarchical self-organizing maps, " Proceedings of the 9th International Conference on Adaptive and Natural Computing Algorithms, ICANNGA09, Springer-Verlag, Berlin, Heidelberg, pp. 130-139, 2009.

[7] D. Ippoliti, X. Zhou, "A-GHSOM: an adaptive growing hierarchical self-organizing map for network anomaly detection," J.ParallelDistr.Comput., vol. 72, no. 12, pp. 1576C1590, 2012.

[8] K. Lagus, S. Kaski, T. Kohonen, "Mining massive document collections by the WEBSOM method, " Information Sciences, vol. 163, pp. 135-156, 2004.

[9] Y. Yu, D.Alahakoon, "Batch implementation of growing self-organizing map, " in: Computational Intelligence for Modelling, Control and Automation, 2006 and International Conference on intelligent Agents, Web Technologies and Internet Commerce, International Conference on, IEEE, pp. 162-175, 2006.

[10] Vasighi M , Amini H, "A directed batch growing approach to enhance the topology preservation of self-organizing map, " Applied Soft Computing, vol. 55, pp. 424-435, 2017.

[11] H. G. Kayacik, A. N. Zincir-Heywood, M. I. Heywood, "A hierarchical SOM-based intrusion detection system," Engineering Applications of Artificial Intelligence, vol. 20, no. 4, pp. 439-451, 2007.

[12] http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html

[13] Suseela T. Sarasamma, Qiuming A. Zhu, and Julie Huff, “Hierarchical Kohonenen Net for Anomaly Detection in Network Security,” IEEE Transactions on systems, man, and cybernetics-part B: cybernetics, vol. 35, no. 2, April. 2005.