Artificial Neural Content Techniques for Enhanced Intrusion Detection and Prevention System

K.Karthikeyan [1], Dr.R.Mala [2]

[1]*Research Scholar, Marudupandiyar College, Vallam, Thanjavur*

[2]*Assistant Professor, Department of Computer Science, Marudupandiyar College, Vallam, Thanjavur*

Abstract- This paper presents a novel approach for detecting network intrusions based on a competitive training neural network. In the paper, the performance of this approach is compared to that of the self-organizing map (SOM), which is a popular unsupervised training algorithm used in intrusion detection. While obtaining a similarly accurate detection rate as the SOM does, the proposed approach uses only one forth of the computation times of the SOM. Furthermore, the clustering result of this method is independent of the number of the initial neurons. This approach also exhibits the ability to detect the known and unknown network attacks. The experimental results obtained by applying this approach to the KDD-99 data set demonstrate that the proposed approach performs exceptionally in terms of both accuracy and computation time.

Keywords--Network Security, Network Intrusion Detection, Data Mining, Artificial Neural Network, Competitive Learning

1. INTRODUCTION

Intrusion detection is a critical process in network security. Traditional methods of network intrusion detection are based on the saved patterns of known attacks. They detect intrusion by comparing the network connection features to the attack patterns that are provided by human experts. The main drawback of the traditional methods is that they cannot detect unknown intrusions. Even if a new pattern of the attacks were discovered, this new pattern would have to be manually updated into the system. On the other hand, as the speed and complexity of networks develop rapidly, especially when these networks are open to the public Web, the number and types of the intrusions increase dramatically. Human analysis becomes insufficient. This leads to the interest in using data mining techniques in network intrusion detection [3, 11].

Data mining-based intrusion detection techniques canbe categorized into misuse detection and anomaly detection [11]. The misuse detection techniques build the patterns of the attacks by learning from the labelled data. The main drawback of the misuse detection techniques is that they cannot detect new attacks that have never occurred in the training data. On the other hand, the anomaly detection techniques establish normal usage patterns. They can detect the unseen intrusions by investigating their deviation from the normal patterns. The artificial neural networks provide a number of advantages in the detection of network intrusions [2]. The application of the neural network techniques has been considered for both the misuse detection model and the anomaly detection model [10, 15]. As an unsupervised neural network, the SOM has beenapplied in anomaly detection. It implicitly prepares itself todetect any aberrant network activity by learning to characterizethe normal behaviours [15]. However, the SOM has asignificant shortage: the number of neurons affects the network'sperformance. Increasing the number of output nodeswill increase the resolution of the map, but the computationtime will dramatically increase. In this paper, we propose anefficient clustering algorithm based on the competitive neuralnetworks. This approach obtains accuracy similar to thatof the self-organizing map while costing much less computation time. The rest of the paper is organized as follows. In the nextsection, we briefly review the background of the applicationof artificial neural networks to network intrusion detection.Section 4 discusses the self-organizing maps and the proposed approach. In Session 5, experiments reveal the speed and accuracy of the proposed approach compared to the SOM. Finally, Section 6 gives a summary and concludes the current study.

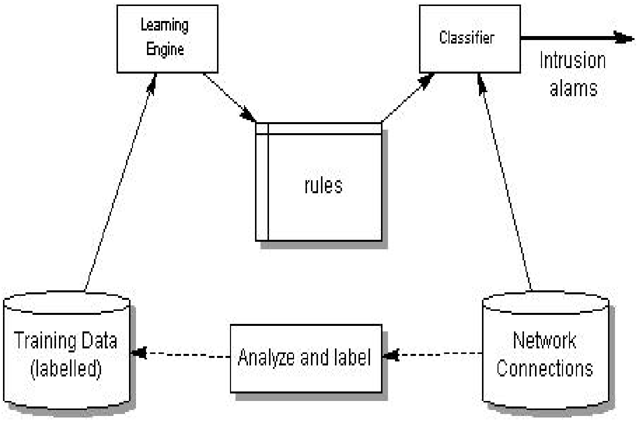


Figure 1. Network intrusion detection using labelled data

2. RELATED WORKS

An increasing amount of research has been conducted on the application of neural networks for detecting network intrusions. The artificial neural networks have the potential to resolve a number of problems encountered by the other current approaches in intrusion detection. The neural networks gain experience by training the system to correctly identify the preselected examples of the problem. A Multilayer Perceptron (MLP) was used in [5] foranomaly detection. The proposed model is a single hiddenlayer neural network. The performance of this model tested on the DARPA 1998 data set was a correct detection rate of 77% with 2.2% false alarms.

The MLP was also applied in [12]. Generic keywords wereselected to detect the attack preparations and actions after the break-in. The back-propagation algorithm was used inthe learning phase to adapt the weights of the neural network. This approach obtained a detection rate of 80%when it was tested on the DARPA 1998 data set. A hybrid model of the SOM and the MLP was proposedin [2]. In that work, the self-organizing map was combinedwith the feed-forward neural network. This model wasdesignedto detect the dispersing and possibly collaborative attacks. The SOM was also applied to perform the clusteringof network traffic and to detect attacks in [6, 14]. In [6],SOM was used to map the network connections onto 2-dimensional surfaces, which were displayed to the networkadministrator. The intrusions were easily detected in thisview. However, the approach needs a visual interpretation by the network administrator. The SOM is trained by using the normal network traffic in [14]. The trained SOM reflects the distribution of the normal network connections. If the minimum distance between a network connection and the neurons of the trained SOM is more than a pre-set threshold, this connection is classified as an intrusion. In addition, artificial neural networks have also been proposed in the detection of the computer viruses. A self-organizing map was selected in [4] for intrusion detection. In that work, the self-organizing map was designed to learn the characteristics of normal activities. The variations from normal activities provided an indication of a virus.

3. THE DETECTION PROCESS

The data source can be labelled or unlabelled based on the learning algorithm used in the data mining-based intrusion detections. Unsupervised algorithms can be applied to unlabelled data while supervised algorithms can only use labelled data. In supervised learning, the training data must be labelled before they are presented to the training algorithm. Figure1 shows the intrusion detection process using supervised learning algorithm. First, the original data must be analyzed and labelled as normal connections or attacks by human experts. After that, the learning algorithms generalize the rules from the training data. Finally, The classifier uses the generated rules to classify the new network connections. A difficulty of the supervised learning is labelling the data. If a large data set is used for training, the labelling duty could be very hard. If choosing a small portion as the training data, the selection of the training examples is crucial to the learning result.

Unlike supervised learning algorithms, which can only use labelled data, unsupervised learning algorithms have the ability to learn from unlabelled data. The algorithm we pro pose in this paper is a clustering method, a typical unsupervised learning algorithm. This approach can be applied on not only labelled data but also unlabelled data. The detection process using the unlabelled data is illustrated in Figure 2. First, the training data are clustered by the clustering algorithm. Second, the clustered weight vectors can be labelled by a labelling process. Various methods can be applied to this process. One approach to label a cluster center is to select a sample group of the data from this cluster randomly and label this cluster with the major type of the sample. Finally, the labelled weight vectors can be used to classify the network connections.

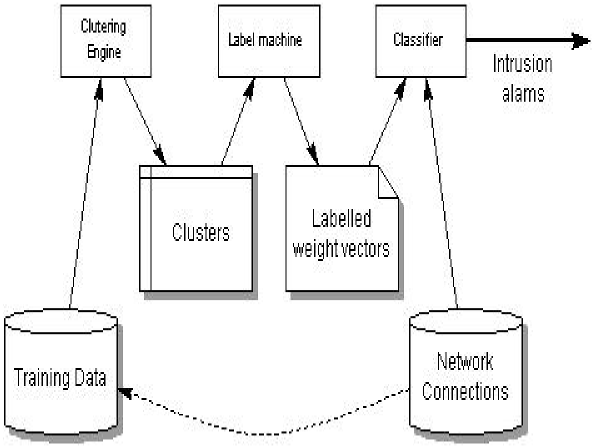
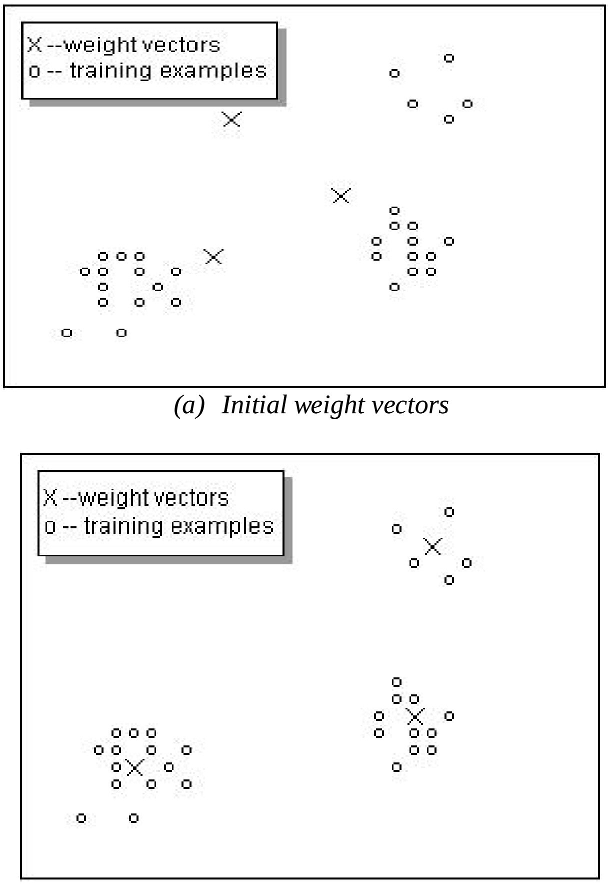


Figure 2. Network intrusion detection using unlabelled data



(a) Initial weight vectors

(b) Clustering result

Figure 3. The principle of SCLN

The main difference between the two processes discussed above is the time and the number of examples for labelling. Unlike the first process, in which the data must be labelled before training, the second process has the ability to organize the unlabelled data and to provide the cluster centers for labelling. Therefore, the second process reduces the risk of selecting improper data as the training set.

4. METHODOLOGY

This section discusses our improved competitive neural network approach for detecting network intrusions. We derive the new approach from the Standard Competitive Learning Network (SCLN). After that, a brief review of the SOM is given since the new approach will be compared with the SOM.

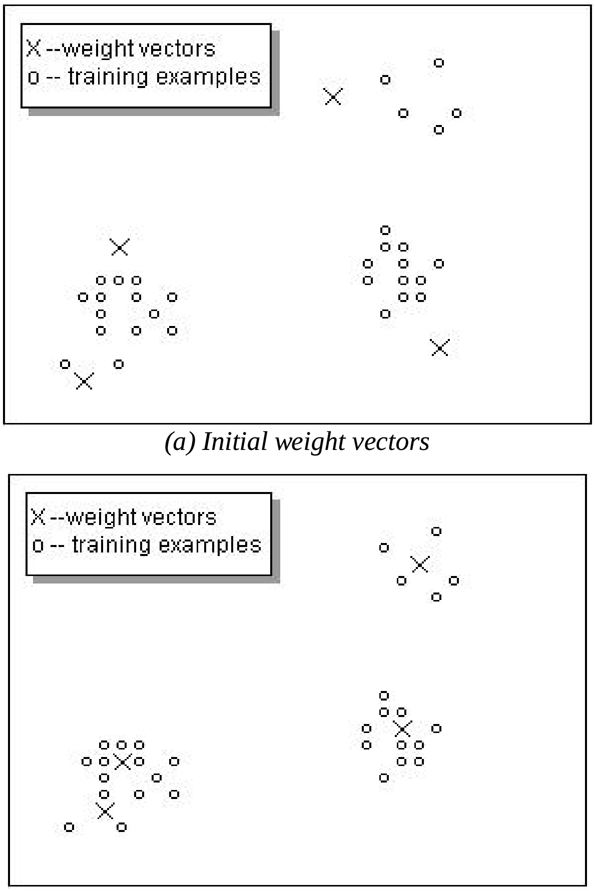
4.1. THE IMPROVED COMPETITIVE LEARNING NETWORK

The improved competitive learning network (ICLN) isbased on the SCLN. The simplest SCLN is a single-layerneural network in which each output neuron is fullyconnectedto the input nodes. In the SCLN, the output neuronsof a neural network compete to become active. Whena training example is presented to the network, the outputneurons compete among themselves. If a neuron won thecompetition, its weight vector would be updated. Accordingto the standard competitive learning rule, the weight updateis calculated by the following update rule:

$$W_j(n+1) = W_j(n) + \eta(n)(x-W_j(n)) \quad (1)$$

Where η is the learning rate, and w_j is the weight vector ofthe winning neuron j. The essence of competitive learningis illustrated in Figure 3. The network initialized a numberof neurons randomly. The initial neurons learn by shiftingtheir synaptic weights towards the input nodes. After training, each output neuron should represent a cluster of the input data set by moving its own synaptic weightvector to the center of that cluster. This process shows that the SCLN has the ability of performing clustering. However, the performance of the SCLN is heavily dependent on the number of the output neurons and the initialization of their weight vectors. Once the number of the output neurons is set, the number of clusters is also determined regardless of the distribution of the data.

On the other hand, different initial weight vectors may lead to different final clusters because the update function in Equation 1 only moves the weight vector of the winning neuron toward its local nearby examples. Figure 4 shows a scenario that reveals the limitations of the SCLN. In this scenario, two neurons are initialized in one cluster. The SCLN will result in four clusters although only three clusters are expected. A critical shortage of the SCLN is that it may split one cluster into many small clusters.



(a) Initial weight vectors

(b) Clustering result

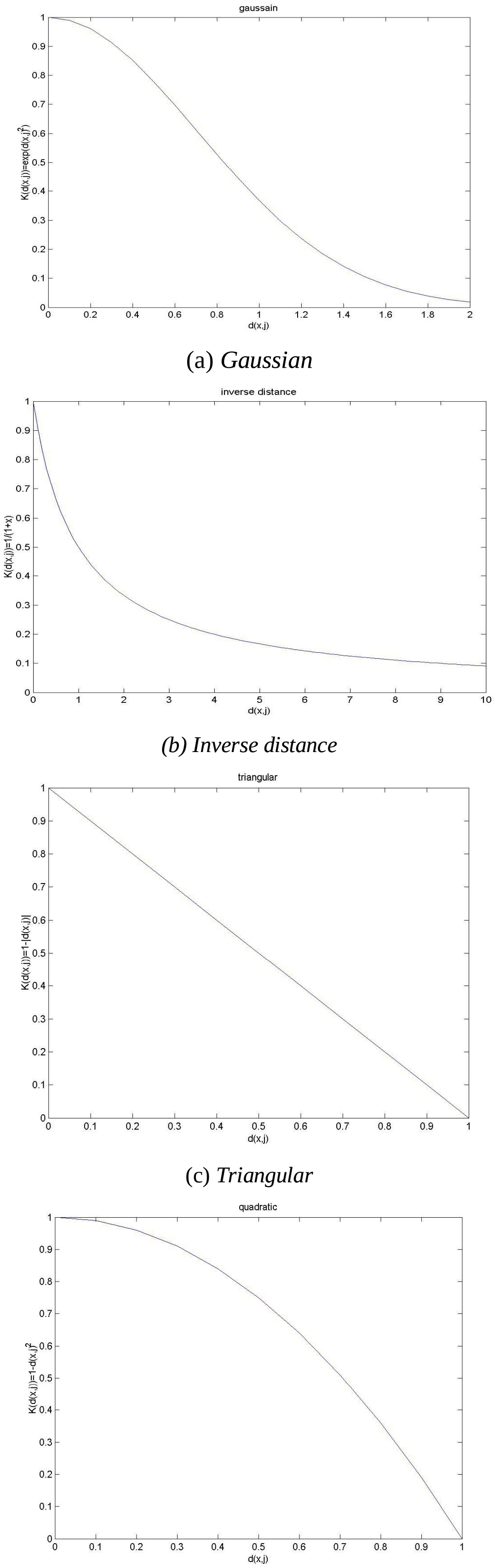
Figure 4. The shortage of SCLN a) and b)

The improved competitive learning network (ICLN) can overcome the shortages of the SCLN. Furthermore, the ICLN obtains a better performance regarding the computation time. In this approach, the winning neuron updates its weight vector by using the same update rule in Equation 1. At the same time, the other neurons also update their weight vectors based on the following equation:

$$W_j(n+1) = W_j(n) -\eta_2(n)k(d(x,j))(x-W_j(n))(2)$$

Where η_2 is the learning rate, and K (d(x, j)) is a kernel function in which d(x, j) is the distance between the neuron j and the input x. There are various choices of the kernel function [1], such as the inverse distance, the triangular kernel, the quadratic kernel, and the Gausian kernel. The *Gaussian Kernel* is a commonly chosen kernel function:

$$K(d(x, j)) = e-d2(x,j)$$



(a) *Gaussian*

(b) Inverse distance

(c) *Triangular*

(d) Quadratic

Figure 5. The curves of some kernel functions

A kernel function obtains the maximum value at zero distance, and the value decays as the distance increases. The kernel functions reflect the influence of the distance to the update rule. Figure 5 shows the curves of some commonly used kernel functions. As a result, the updated value would be smaller if the distance between the neuron and the input were greater. The new update rule applied to the losing neurons moves the weight vectors of these neurons away from the input pattern. The effect of the above update rules is shown in Figure 6. This update process not only avoids the limitation of the SCLN but also makes the clustering much faster. The algorithm of the ICLN is outlined in Figure 7.

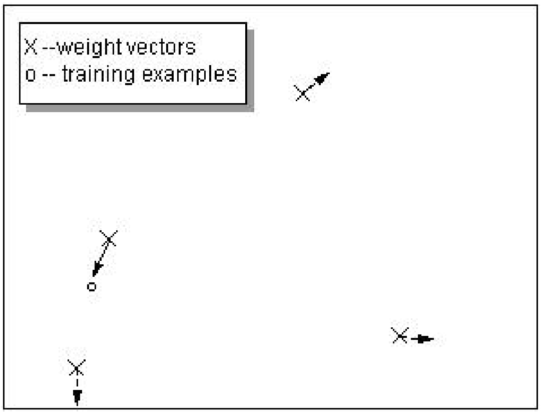


Figure 6. The effect of the ICLN update rules

4.2. The Self-Organizing Maps

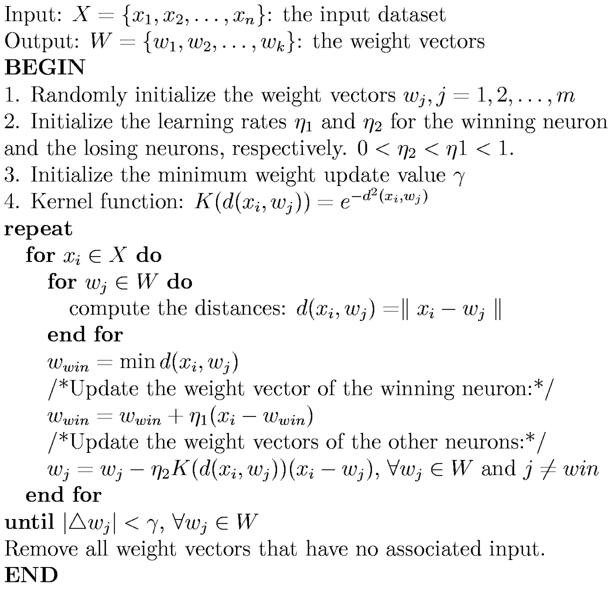
The SOM is one of the most popular neural network models. It is a fully connected, single-layer neural network [8]. It maps a multi-dimensional data set onto a one or two-dimensional space. In the SOMs, data are clustered by using soft competition, which is the term opposite to hard competition [13]. In hard competition, there is only one winner: in each competition, only one node is active, and all of the others are inactive. Soft competition allows not only the winner but also its neighbours to be active. That is, after competing for the presented inputs, the winner and its neighbour nodes have their weights updated by the following rule:

$$W_j(n+1) = W_j(n)\_+\eta(n)h_{j\ i(x)}(n)(x-W_j(n))(3)$$

where $w_j(n)$ denotes the weight vectors of the winning neuron and its neighbours at time n, $\eta(n)$ denotes the learning rate at time n, and $h_{j,i(x)}(n)$ is the neighbourhood function centered at the winning node. The Gaussian function is commonly used as the neighbourhood function:

hj,i(x)(n) = e−d2 j,i/2_2(n)

Where i is the center, η_2 (n) denotes the variance at time n, and $d_{j,i}$ represents the distance between the winning neuron i and its neighbour node j.



```
Input: $X = \{x_1, x_2, \dots, x_n\}$: the input dataset
Output: $W = \{w_1, w_2, \dots, w_k\}$: the weight vectors
BEGIN
1. Randomly initialize the weight vectors $w_j, j = 1, 2, \dots, m$
2. Initialize the learning rates $\eta_1$ and $\eta_2$ for the winning neuron
and the losing neurons, respectively. $0 < \eta_2 < \eta 1 < 1$.
3. Initialize the minimum weight update value $\gamma$
4. Kernel function: $K(d(x_i, w_j)) = e^{-d^2(x_i, w_j)}$
repeat
  for $x_i \in X$ do
    for $w_j \in W$ do
      compute the distances: $d(x_i, w_j) = \| x_i - w_j \|$
    end for
    $w_{win} = \min d(x_i, w_j)$
    /*Update the weight vector of the winning neuron:*/
    $w_{win} = w_{win} + \eta_1(x_i - w_{win})$
    /*Update the weight vectors of the other neurons:*/
    $w_j = w_j - \eta_2 K(d(x_i, w_j))(x_i - w_j), \forall w_j \in W$ and $j \neq win$
  end for
until $|\triangle w_j| < \gamma, \forall w_j \in W$
Remove all weight vectors that have no associated input.
END
```

Figure 7. Algorithm: improved competitive learning network (ICLN)

This update process moves the winning neuron and its neighbours to the input vector. The effect of applying the neighbor hood function is that the closer neighbours obtain the greater updates. After training, the output layer is expected to reflect the topology or density of the input data set. It has excellent capabilities for visualizing high dimensional data onto a 1-or 2-dimensional space. One drawback of the SOMs is that the number of neurons affects the performance of the clustering. To obtain a better clustering result, various numbers of nodes have to be evaluated. Increasing the number of nodes could increase the resolution of the map. However, it could increase the computation time dramatically [7].

5. EXPERIMENTS AND RESULTS

In this section, we experiment with the SOM and the ICLN by using the KDD-99 data [9] and compare the results of these two methods. The KDD-99 dataset was used for the Third International Knowledge Discovery and Data Mining Tools Competition. This dataset was acquired from the 1998 DARPA intrusion detection evaluation program. There were 4,898,431 connection records, of which 3,925,650 were attacks. From this data set, 501,000 records were chosen as our experimental data. The selected connections were further split into the training set and the test set, containing 101,000 and 400,000 connections respectively. There were 21 types of intrusions in the test set, but only 7 of them were chosen in the training set. Therefore, the selected data also challenged the ability to detect the unknown intrusions. The same data sets were used in the experiments to evaluate the performance of the SOMs and the ICLN in the same environment.

5.1. DATA PREPARATION

In the KDD-99 data set, each connection is labelled as “normal” or a particular type of the attacks. A connection is represented by 41 features, which include the basic features of the individual TCP connections, the content features within a connection suggested by the domain knowledge, and the traffic features computed by using a two second time window [9]. The features in columns 2, 3, and 4 in the KDD-99 data set are the protocol type, the service type, and the flag, respectively. The value of the protocol type may be TCP, UDP, or ICMP; the service type could be one of the 70 different network services such as HTTP and SMTP; and the flag has 11 possible values such as SF or S2.These qualitative features are mapped into quantitative valuesin the preparation process for calculating the similaritiesof the connections.

5.2. EXPERIMENT 1: THE SOMS

In this experiment, we investigate both the accuracy and elapsed time of the SOMs. In the training phase, the SOMs were used to cluster the training data. After training, each cluster was labelled according to the majority type of data in this cluster. For instance, if more than 50% of the connections in a cluster were intrusions, the cluster and its centroid weight vector would be labelled as intrusion. In the test phase, each test connection was assigned to its closest neuron, which was the center of a cluster, and this connection was identified by the label of that cluster. Table 1 shows the effect of using various number of initial neurons. The detail of the clustering result of the 3 × 5 SOM was further investigated. After training, 5 of the 15 initial weight vectors were removed because they did not have any associated data to identify the connection types. The remaining 10 weight vectors were labelled as the majority types of the data in their group. The labelled weight vectors were used to detect the connections in the test data set. The detail of clustering result generated by a 3 × 5 SOM on the training data is shown in Table 2.

5.3. EXPERIMENT 2: THE ICLN

The performance of the ICLN was also investigated by using various numbers of the initial neurons. The results of various trials are summarized in Table 3, in terms of the final number of the clusters discovered by the ICLN. In all of the trials, 6 clusters were discovered after training regardless the number of initial neurons. Moreover, the performances of the ICLN in these trials are similar. This result implies that the clustering result is unaffected by the number of the initial neurons in the ICLN algorithm. The detailed clustering result of the training data generated by a 15 neuron ICLN is shown in Table 4. After training, 6 clusters were detected. The weight vectors of these clusters were used to identify the connections in the test data set.

TABLE 1. THE PERFORMANCE OF SOM

NO. OF INITIAL NEURONS	NO. OF CLUSTERS	ELAPSED TIME	ACCURACY	PRECISION	RECALL
9	8	1057s	97.80%	98.31%	98.96%
15	10	2162s	97.89%	98.41%	98.97%
18	12	2507s	97.82%	98.33%	98.97%
20	15	3308s	97.89%	98.41%	98.97%

TABLE 2. THE RESULT OF THE 3 × 5 SOM

NO. OF INITIAL NEURONS	NO. OF CLUSTERS	ELAPSED TIME	ACCURACY	PRECISION	RECALL
9	6	454s	97.89%	98.42%	98.97%
15	6	608s	97.89%	98.42%	98.97%
18	6	682s	97.89%	98.42%	98.97%
20	6	723s	97.89%	98.42%	98.97%

TABLE 3. THE PERFORMANCE OF ICLN

	CLUSTER				
	1	2	3	4	5
NORMAL	69413	26	5776	869	1422
BUFFER OVERFLOW				2	
LOAD MODULE				1	
PERL				1	
NEPTUNE					
SMURF					
IPSWEEP	1		11	20	
BACK	59				

TABLE 4. THE RESULT OF THE ICLN

	CLUSTER				
	6	7	8	9	10
NORMAL	54	70	1	77	180
BUFFER OVERFLOW					
LOAD MODULE					
PERL					
NEPTUNE				30	22093
SMURF					
IPSWEEP	33			859	
BACK				2	

	CLUSTER					
	1	2	3	4	5	6
NORMAL	1422	70	75260	180	878	78
BUFFER OVERFLOW					2	
LOAD MODULE					1	
PERL					1	
NEPTUNE						30
SMURF				22093		
IPSWEEP			12		53	859
BACK			59			2

5.4. DISCUSSION

While obtaining similar accuracy, the ICLN requires less computation time. Figure 8 shows the computation time of the SOM and ICLN algorithms in the training phase. The elapsed time of the SOM increases rapidly when the number of initial neurons increases. The results demonstrate that the ICLN uses much less time than the SOM. Interestingly, the ICLN discovered clusters similar to those discovered by the SOMs. The clusters 1, 2, and 4 in Table 4 are exactly the same as the clusters 5, 7, and 10 in Table 2. The cluster 8 in the SOM contains only one instance. This instance moves to the nearest neighbour, cluster 9, when the weight vector of cluster 8 is removed. The reconstructed cluster becomes the same as the cluster 6 in the ICLN. The above experiments confirmed that the performance of the ICLN is unaffected by the number of the initial output neurons. The results also suggest that this approach has the ability to detect unseen network attacks.

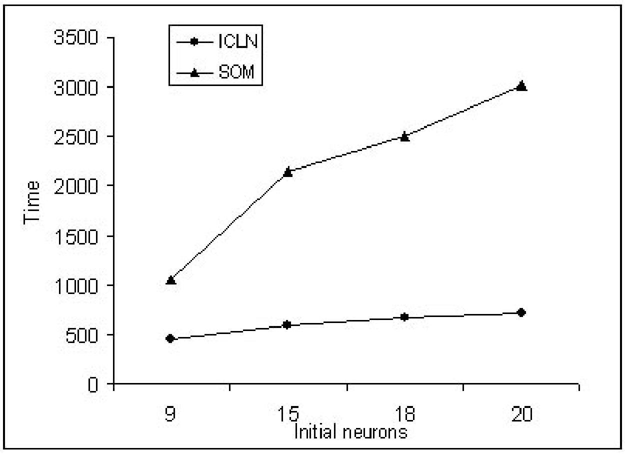


Figure 8. The elapsed time of the SOM and ICLN

6. CONCLUSION

A novel approach for detecting network intrusions is proposed in this paper. The proposed approach obtains a significant improvement in speed. The experiments also show that the proposed approach has the ability to detect the unknown intrusions by clustering the connections based on their similarities. Specifically, we compared the performance of this approach with the SOM. The proposed approach obtains a similar accuracy as the SOM does whereas it only uses one fourth of the computation time of the SOM. In addition, the clustering result of the proposed approach is independent of the number of initial neurons. The experimental results on the KDD-99 data set demonstrates that the developed algorithm is successful in terms of not only accuracy but also efficiency in network intrusion detection.

REFERENCES

[1] C. G. Atkeson, A. W. Moore, and S. Schaal. Locally weighted learning. *Artificial Intelligence Review*, 11(1–5):11 – 73, 1996.

[2] J. Cannady. Artificial neural networks for misuse detection. In *Proceedings of the 1998 National Information Systems Security Conference (NISSC'98) October 5-8 1998. Arlington, VA.*, pages 443–456, 1998.

[3] P. Dokas, L. Ertoz, V. Kumar, A. Lazarevic, J. Srivastava, and P. Tan. Data mining for network intrusion detection. In *Proc.NSF Workshop on Next Generation Data Mining*, 2012.

[4] K. Fox, R. Henning, J. Reed, and R. Simonian. A neural network approach towards intrusion detection. In *In Proceedingsof the 13th National Computer Security Conference*, 1990.

[5] A. K. Ghosh and A. Schwartzbard. A study in using neural networks for anomaly and misuse detection. In *In Proceedings of USENIX Security Symposium*, 1999.

[6] L. Girardin. An eye on network intruder-administrator shootouts. In *Proceedings of the Workshop on Intrusion Detection and Network Monitoring (ID'99)*, pages 19–28, Berkeley, CA, USA, 1999. USENIX Association.

[7] Y. Guan, A. A. Ghorbani, and N. Belacel. Y-means: A clustering method for intrusion detection. In *IEEE Canadian Conference on Electrical and Computer Engineering, proceeding*, page 1083, 2003.

[8] S. Haykin. Neural networks: A comprehensive foundation. *Second Edition, Prentice Hall Inc.*, 1999. [9] S. Hettich and S. D. Bay. The uci kdd archive, 1999.

[10] S. Lee and D. Heinbuch. Training a neural-network basedintrusion detector to recognize novel attacks. *IEEE Transactionson Systems, Man & Cybernetics, Part A (Systems &Humans)*, 31(4):294 – 9, July 2011.

[11] W. Lee and S. Stolfo. Data mining approaches for intrusiondetection. In *Proceedings of the 7th USENIX Security Symposium*,San Antonio, TX, 1998.

[12] R. P. Lippmann and R. K. Cunningham. Improving intrusiondetection performance using keyword selection and neuralnetworks. *Computer Networks (Amsterdam, Netherlands:1999)*, 34(4):597–603, 2008.

[13] J. C. Principle, N. Eulano, and W. C. Lefebvre. Neural andadaptive systems: Fundamentals through simulations. *JohnWiley & Sons, Inc.*, 2011.

[14] M. Ramadas, S. Ostermann, and B. Tjaden. Detectinganomalous network traffic with self-organizing maps. In *RecentAdvances in Intrusion Detection, 6th International Symposium,RAID 2003*, pages 36–54, 2013.

[15] B. C. Rhodes, J. A. Mahaffey, and J. D. Cannady. Multipleself-organizing maps for intrusion detection. In *Proceedingsof the 23rd National Information Systems Security Conference*,2009.