An Integrated Intrusion Detection System by Using Multiple Neural Networks

Guisong Liu and Xiaobin Wang
Computational Intelligence Laboratory
School of Computer Science and Engineering
University of Electronic Science and Technology of China
Chengdu 610054, P. R. China
{lgs,xbwang}@uestc.edu.cn

***Abstract*—Neural networks approach is one of the most promising methodologies for intrusion detection in network security. An integrated intrusion detection system (IIDS) scheme based on multiple neural networks is proposed. The approaches used in IIDS include principal component neural networks, growing neural gas networks and principal component self-organizing map networks. By the abilities of classification and clustering analysis of the above methods, IIDS can be adapted to both anomaly and misuse detections for intrusive outsiders. The training stage is a mixture of supervised manner and unsupervised one. Furthermore, IIDS uses the buffering and spoofing principles of address resolution protocol (ARP) to capture and refuse the insider intruders trying to log on a local area network (LAN). Therefore, IIDS is able to detect the intrusions/attacks both from the outer Internet and an inner LAN. Experiments are carried out to illustrate the performance of the proposed intrusion detection system by using the KDD CUP 1999 Intrusion Detection Evaluation dataset.**

***Index Terms*—Intrusion Detection System, Neural Gas Networks, Principal Component Neural Networks, Self-Organizing Map, Address Resolution Protocol.**

I. Introduction

In recent years, the increase of network intrusions/attacks in number has made the intrusion detection system (IDS) the most important role to the security infrastructure of most organizations. Intrusion Detection can be defined as "software or hardware systems that automate the process of monitoring the events occurring in a computer system or network, analyzing them for signs of security problems"[1]. In general, an important actual computer network contains many security systems, such as a traditional Internet fire-wall, vulnerability scanning system and the IDS. As shown in Fig.1, the Internet fire-wall can filter the network traffic by predetermined rules to prevent some known type of outer intrusions; but the critical limitation is its inability to detect and refuse novel attacks or inner intruders. Access control module can prevent someone from doing what are out of his privileges; but neither can it guarantee who with superior privilege doing harmful things to the protected system, nor prevent who with lower privilege acquiring superior ones. In addition, a number of existing vulnerabilities in a system or a network can be detected by vulnerability scanning system, but the scanning is periodical, not real-time. Due to all the drawbacks of the mentioned traditional security systems, an IDS will play a more and more important role in a common security architecture.

It is well known that a norm profile can be constructed using normal behaviors exhibited by either a user or a system. Any behavior with a certain degree deviation from the norm profile is determined as an intrusion/attack. This methodology in intrusion detection is called anomaly detection. Meanwhile, misuse detection can model a specified attack on a system as a specific pattern; those activities which are similar to (generally there exists a determining threshold) the predefined pattern are considered as an same intrusion in type; therefore, misuse detection is also named signature-based detection [2]. In the area

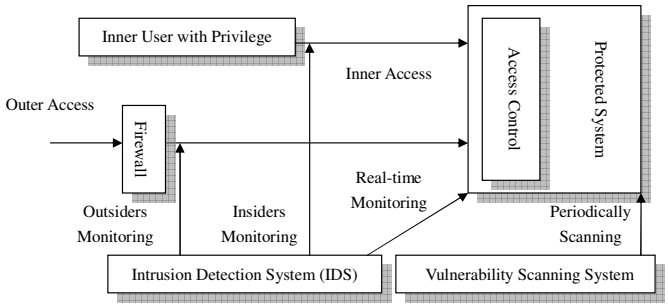

Fig. 1. An IDS plays the most important role in a common security architecture of an actual computer system.

of intrusion detection by using artificial neural networks, the most famous international institutions/organizations includes the University of Georgia, MIT, Research of RST Corp., UBILAB Laboratory, etc. In the pioneer work by J. Cannady [3] and J. Mahaffy [4] for misuse detection, Multi-Layer-Perceptron (MLP) model is performed using backpropagation learning algorithm, and then combined with Self-organizing Map (SOM), an MLP/SOM model is also constructed for better performance. From then on, more and more neural network based models and algorithms are studied and implemented for intrusion detection problems. Recently, a lot of attention has been drawn to the variations of SOM proposed by Kohonen [5]. To separate clusters of high-dimensional feature vectors in a one or two dimensional output space is an SOM's most highlighted advantage. Combined with Radial-Basis-Function networks, T. Horeis proposed the RBFSOM model for intrusion detection [6]; In [7], a hierarchical SOM is built for intrusion detection using different feature sets. To overcome

978–1–4244–1674–5/08/$25.00 © 2008 IEEE

static architecture and limited capabilities in representing the hierarchical relations of the input data, Rauber [8] proposed an self-increasing SOM model. It is well known that principal components analysis (PCA) can be applied to feature extraction or selection. In [9], the authors study the off-line intrusion detection by using PCA and MCA (minor components analysis). Considering the merits of online computing of neural networks, we have proposed a hierarchical PCA neural networks model based on adaptive principal components extraction (APEX) algorithm for both anomaly detection and misuse detection in our previous work [10]. But the crucial drawback of the model is its supervised training manner because it is very difficult to obtain so many labeled training data in practical environment. Therefore, combining PCA with SOM, a novel unsupervised method to cluster intrusions/attacks is presented in [11]. Following the most popular SOM neural networks to clustering analysis, Martinetz et al. proposed the Neural Gas (NG) algorithm in 1993 [12], as a fast neural net-based clustering method, and it has been successfully applied to vector quantization, prediction and topology representation, etc. Following the basic idea of NG for clustering analysis in input space, We propose a growing NG algorithm for pattern construction of normal activities by using network traffic data.

All the mentioned methods based on neural networks can be used to intrusion detection by their classification and clustering analysis abilities, and the training is supervised or unsupervised. Clearly, single method cannot achieve better performance and cannot build an integrated IDS. This motivates us to propose the IIDS model by using multiple neural networks. Furthermore, the inner intruder detection has also been taken into account.

The rest of this paper is organized as follows. The presented methods to construct the IIDS are analyzed in Section II. Section III proposes the IIDS model and a detailed discussion is also given. In Section IV, experiments are carried out on the KDD CUP 1999 intrusion detection datasets. Finally, conclusions are given in Section V.

II. The Methodologies

A. Growing Neural Gas Networks

A traditional neural gas algorithm is Euclidean distance based and the detailed description can be found in [12]. But the network structure is static. In [13], a fixed-width clustering method based on single-linkage clustering was proposed, which was applied to studying intrusion detection by using clustering analysis. Following these basic ideas, a growing neural gas (GNG) algorithm is proposed in this paper.

By a predefined splitting radius, an input space can be represented by split neurons. Each neuron denotes a small cluster using its neural weight vector as the centroid of that cluster. Following competitive and cooperative processes will adjust the topology of the network, so that the boundary of the overall class pattern which can be denoted by all the neurons, will become more accurate. Furthermore, after the training process, we perform a "neuron deleting" operation to the network in order to decrease the influence of "lack-training" or "dead" neurons. This step enables the GNG network to eliminate input noise or outliers in a training dataset. The key point of GNG is that the main purpose is for one-class pattern construction by clustering analysis. Even though anomaly-based approaches are promising, they usually produce a relatively high number of false alarms, due to the difficulties in modelling the normal events. The training algorithm of GNG is given in pseudocode algorithm ALGORITHM-TRAIN-GNG which is stated as follows:

ALGORITHM-TRAIN-GNG:

STEP1: Get necessary input from the user:

- Initial neuron number, M=1
- Splitting radius, θ
- Minimum winning times for neuron deleting operation, *MinWinTime*
- Maximum training epoch, *MaxEPOCH*
- Training dataset $\{\mathbf{x}(t)\}, t = 1, \cdots, N$

STEP2:Repeat: for i=0 to *MaxEPOCH* and j=1 to N, input $\mathbf{x}(j)$, and CASE i:

- 0: goto STEP3
- *MaxEPOCH*: goto STEP4
- else: goto STEP5

STEP3:Auto-increasing:

- for k=1 to M, calculate, $d_k = ||\mathbf{x}(j) - \mathbf{w}_k||$
- and, $d_{min} = \arg\min\{d_k\}$
- if $d_k > \theta$, M=M+1, $\mathbf{w}_m \leftarrow \mathbf{x}(j)$
- Goto STEP2.

STEP4:Neurons deleting:

- Calculate BMU times for every neuron, $Bmu(m), m = 1, \cdots, M$
- if $Bmu(m) < MinWinTime$, delete neuron m
- Goto STEP6.

STEP5:Weights updating:

- Update the weight vectors $\mathbf{w}_i$ as:

$$\mathbf{w}_{i+1} = \mathbf{w}_i + \eta(t) h_\lambda(r_i)(\mathbf{x}(t) - \mathbf{w}_i), \tag{1}$$

- Where r_i is the neighborhood ranking of neurons and the neighborhood function is

$$h_\lambda(r_i) = e^{-\frac{r_i}{\lambda(t)}}, \tag{2}$$

- Learning rate $\eta(t)$ and decay constant $\lambda(t)$ are calculated as

$$\eta(t) = \eta_0 \left(\frac{\eta_{end}}{\eta_0}\right)^{\frac{t}{t_{max}}}, \tag{3}$$

$$\lambda(t) = \lambda_0 \left(\frac{\lambda_{end}}{\lambda_0}\right)^{\frac{t}{t_{max}}}. \tag{4}$$

- Goto STEP2.

STEP6: End of ALGORITHM-TRAIN-GNG

B. Principal Component Analysis Self-Organizing Maps

Combining PCA with SOM, Rubio et al. propose the PCASOM model[14]. In our previous work [11], we have simplified the model and applied it to cluster intrusions. Another approach called "Min-Max Hyperellipsoidal Clustering" [7] seems very similar to the PCASOM. Due to the similarity of normal connections with some specified attacks, the performance of PCASOM is somewhat unsatisfactory. This means that it is very important to separate "normal" connections from other "intrusive" ones before performing clustering analysis using PCASOM.

A PCASOM model stores information of one cluster by its centroid vector like SOM, and a feature subspace with reduced dimensionality is also maintained in a correlation matrix of a local cluster which is similar to PCA manner. Assume that $\{\mathbf{x}(t)\}\ t = 1, 2, \cdots, L$, are n-dimensional stochastic input data, the mean vector $\mathbf{e} = \frac{1}{l}\sum_{t=1}^{l}\mathbf{x}(t)$ and the covariance matrix of $\mathbf{x}(t)$ are defined by

$$\mathbf{R} = \frac{1}{l-1}\sum_{t=1}^{l}\left[(\mathbf{x}(t)-\mathbf{e})\,(\mathbf{x}(t)-\mathbf{e})^T\right]. \tag{5}$$

Hence, the weights updating of a PCASOM can be given by [11],

$$\mathbf{e}_{i+1} = \mathbf{e}_i + \eta(t)h_{i,c}(t)\left[\mathbf{x}(t)-\mathbf{e}_i\right], \tag{6}$$

$$\mathbf{R}_{i+1} = \mathbf{R}_i + \eta(t)h_{i,c}(t)\left[(\mathbf{x}(t)-\mathbf{e}_i)\,(\mathbf{x}(t)-\mathbf{e}_i)^T - \mathbf{R}_i\right]. \tag{7}$$

The basis vectors of a local principal subspace are denoted by $\mathbf{B}_h^i, h = 1, \cdots, K$, where K is the number of principal directions. Correspondingly, the distance calculation between $\mathbf{x}(t)$ and neuron i can be calculated and the Best Matching Unit (BMU) in competition can be figured out by

$$C = \arg\min_i\left\{\left\|\mathbf{x}(t)-\mathbf{e}_i-\sum_{h=1}^{K}\mathbf{B}_h^{iT}(\mathbf{x}(t)-\mathbf{e}_i)\mathbf{B}_h^i\right\|\right\}. \tag{8}$$

C. Principal Component Analysis Neural Networks

By the feature selection/extraction of PCA, a classifier can be designed with a determination threshold. As an adaptive method to implement PCA, neural network is more suitable for both online computing and different number of input samples than many conventional PCA approaches, such as eigen-decomposition and singular value decomposition. We have explored how to design a PCANN based classifier with its application to intrusion detection in [10]. However, the PCANN using APEX has its apparent shortages, i.e., the determination of selected number of principal components and the supervised training manner. Although an estimation method using accumulating variance ratio has been accepted for many applications which is proposed in [14], it does not be appropriate for online applications.

A novel method by modifying the General Hebbian Algorithm, namely adaptive GHA (AGHA) was presented in [15]. AGHA can approach the intrinsic dimension of the input adaptively while coming to its convergent state. The learning algorithm is described as follows:

$$W(t+1) = W(t)+\eta\Pi(t)\left[W(t)C_t - LT\left[W^T(t)C_kW(t)\right]W(t)\right], \tag{9}$$

where η is a learning rate and $C_t = \beta C_{t-1} + (x(t)x^T(t) - \beta C_{t-1})/t$ is an alternative of input $x(t)$ for better robustness of the algorithm. The matrix $\Pi(t)$ is diagonal. The number of its elements equaling to one is the number of extracting principal components at time t (all the others equal to zero). For this purpose, there are three other functions defined to fulfill the iteration of $\Pi(t)$. See [15] for detailed description of AGHA algorithm.

Comparing with other neural networks to implement PCA, AGHA is able to extract a necessary number of principal components under a predefined precision. Therefore, it's more accurate in the training of a PCANN based classifier than using a man-appointed parameter in advance.

D. Inner Intrusive Hosts Detection Using ARP Principles

Generally, the function of an overall intrusion detection system includes two parts, the detection of outer intrusions from the Internet and the detection of illegal activities in a local LAN. For the second part, this paper concentrates on illegal hosts which log on the inner LAN to do harm to legal hosts.

The principle of ARP protocol is used to solve the problem. The function of ARP in network layer is to convert an Internet address (IP address) to a physical one, i.e., a hardware MAC address (also called network card address with a length of 48 bits). A physical MAC address is exclusive to one host in a network in order to guarantee accessing each other precisely. Note that an IP address is not enough to label one's identification. Each host in a network maintains a ARP buffer which can improve accessing speed for a candidate. When a source host wants to access another one with a target IP address, usually the first thing to do is to look for the corresponding MAC address in its ARP buffer. If there does not exist such information, an ARP request will be broadcasted. Only the target host with the same requested IP address will reply to the request. Hence the source host will obtain the target MAC address and refresh its ARP buffer for directly accessing next time. However, all items (IP/MAC) in an ARP buffer are dynamic in default. A degraded mechanism of ARP buffer will delete such items that do not be used for a period of time. This will lead to a shortened ARP buffer for storage and a high speed for querying. In a Windows NT network, the degraded time is set between two and ten minutes.

The detection of inner intruders includes two steps. Firstly, we record the information of all legal hosts including their IP, MAC, CPU code, etc. According to the ARP request of a host, we can determine whether its information is related to the previous registrations or not. Although he has a disguised IP or modified MAC address, we cannot obtain its other legal registration information. So we can deem it an intruder (illegal user). Secondly, a sequence of spoofed message will be sent to the intruder by a APR server. The message contains

wrong IP/MAC pairs information, e.g., all legal IPs with same imaginary MAC "00-A1-B2-C3-D4-E5". This results in a ARP buffer with wrong information to the intrusive host, and he cannot log on the network and cannot access other legal hosts.

III. Integrated Intrusion Detection System

The proposed IIDS model is shown in Fig. 2, where the first part is "Outer Intrusion Detection" using multiple neural networks, and the second part "Inner Intrusion Detection" using ARP principles. Although the second part is very simple and easy to implement, it is very important to provide an overall solution for both outer and inner intrusion detection.

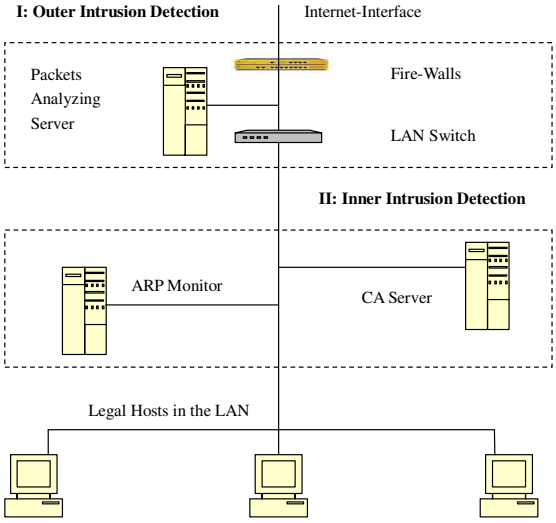

Fig. 2. The integrated intrusion detection system in an actual network environment.

It is clear that the first part for outer intrusion detection is the emphasis of this paper, which is shown in Fig. 3. Because the first part is constructed by multiple neural networks for outer intrusion detection, we name it MNNIDS. The data source is from the Internet interface, which can be developed using WinPcap driver. Training of the neural networks is online or offline, which denoted by solid lines or dashed lines in Fig. 3. Meanwhile, the solid lines also symbolize realtime testing after training. There are many processing steps in MNNIDS, including network data capturing, realtime analyzing, general intrusion alarming, clustering analysis of intrusions, specified intrusion alarming and data types labelling etc. The features of the model are stated as follows:

(1) Anomaly detection by GNG: The training is supervised. A "normal" profile will be built using "normal" type connection data. Due to the resistant ability of GNG to outliers, the purity of "normal" type training data does not have to be 100%. It is very helpful for data preparation. For instance, we can obtain the training data from a closed network for a period of time.

(2)Intrusion clustering by PCASOM: Clearly, the training is unsupervised. The filtered intrusive data by GNG will be grouped by PCASOM sequentially, and these clustered datasets will be the training source of PCANNs for misuse classifiers.

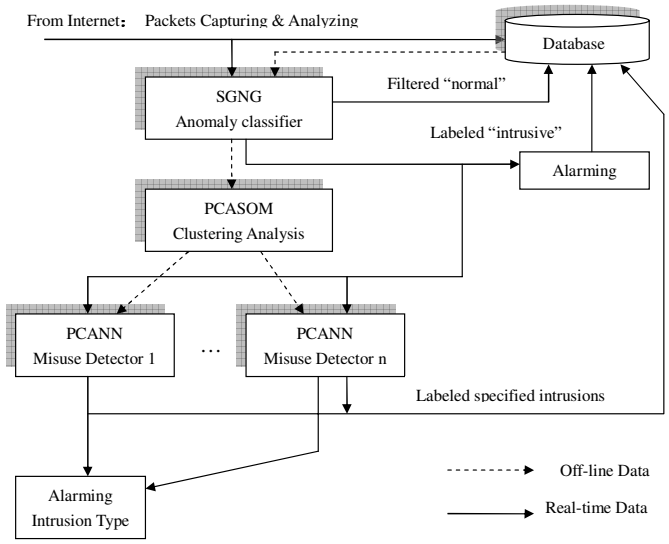

Fig. 3. The multiple neural network based outer intrusion detection scheme.

(3) Misuse detection by PCANNs: For better performance of each misuse classifier, the further training of GHA based PCANNs will be deployed.

(4)MNNIDS is built by multiple neural networks, and this leads to an integration of all their advantages and discarding their disadvantages. Furthermore, MNNIDS is network based IDS, which adapts to many sophisticated network environments.

(5)Portability and extensibility: The misuse detectors can be constructed by other approaches. It is portable for other methodologies to be added to this model. Further, all the labelled intrusive data in a database will be used to train new misuse classifier when the data number is triggered to a certain value. In other words, the signatures of intrusions will be refreshed periodically for novel attacks.

IV. Simulations

A. Experiments Datasets

To improve performances of IDSs with real network traffic, a large-scale realistic Intrusion Detection database has been sponsored by the US Defense Advanced Research Projects Agency (DARPA) in 1998 in order to survey and evaluate research in intrusion detection. The DARPA 1998 and 1999 Intrusion Detection Evaluations consists of comprehensive technical evaluations of research intrusion detection systems [16], which was prepared and managed by MIT Lincoln Labs. The KDD Cup 1999 dataset, which is a subversion of DARPA project, includes "good" normal connections and "bad" intrusion ones. The datasets contain a total of 24 training attack types, with an additional 14 types in the test data only. All the attacks fall in four main categories, such as (1) DoS: denial-of-service, e.g. syn flood; (2) R2L: unauthorized access from a remote machine, e.g. guessing password; (3) U2R: unauthorized access to local superuser (root) privileges, e.g.,

various "buffer overflow" attacks; and (4) Probe: surveillance and other probing, e.g., port scanning [17].

Every network connection is a sequence of TCP packets starting and ending at some well defined times, between which data flows to and from a source IP address to a target IP address under some well defined protocol. Each connection includes forty-one feature values. The attributes in each connection of the KDD datasets has some forms, namely continuous, discrete and symbolic with significantly varying resolution and ranges.

In order to construct "normal" profile using GNG, 11 features of all the 41 items are selected, including src-bytes, dst-bytes, logged-in, count, srv-count, same-srv-rate, dst-host-count, dst-host-srv-count, dst-host-same-srv-rate, dst-host-same-src-port-rate and dst-host-srv-diff-host-rate. Because all the 11 features are *continuous* type, the training and testing data do not require any preprocessing in order to test the adaptive ability of the GNG to raw data.

However, further preprocessing is required for PCASOM to cluster intrusions like most pattern classification methods. For a symbolic type attribute, we first order them with a sequence number from 0 to $n-1$, where n is the specific class number of the attribute. Then we linearly map them to $[0, 1]$. For the discrete type attributes, e.g., land, with value 0 or 1, they do not require any preprocessing. The training data of PCANN classifier uses the same format as PCASOM.

The preparation of training and testing datasets is shown in Table I. where the training datasets are chosen from a 10% subset ($kddcup.data\_10\_percent.gz$) randomly and the testing data are from the labelled dataset ($corrected.gz$). Note that all the individual attack except for "normal" type connections belongs to the four main categories, such as DOS, Probe, U2R and R2L.

TABLE I
DATA PREPARATION FROM KDD CUP 1999 FOR TRAINING AND TESTING OF IIDS

data type	category	training	testing
normal	Normal	16700	51300
back	DOS	890	912
smurf	DOS	6731	1082
neptune	DOS	1620	3487
teardrop	DOS	200	10
ipsweep	Probe	1021	168
portsweep	Probe	586	220
satan	Probe	1072	1128
buffer-overflow	U2R	28	20
loadmodule	U2R	8	2
guess-passwd	R2L	1116	288
imap	R2L	129	304
warezclient	R2L	320	50

B. Anomaly Detection

The most important parameter of GNG network is its splitting radius. A larger radius will lead to a higher detection rate for "normal" connections, i.e., a lower false positive rate; on the other hand, the detection rate for intrusions will accordingly become lower. Hence we conduct the experiments for several times to determine a better choice of the radius. The ROC curve is shown in Fig. 4. The radius values are selected as $2000, 1000, 800, 500, 300, 100$, respectively. The optimal performance for GNG can be achieved with 98.9% detection rate and 1.12% false positive rate; the splitting radius is 500.

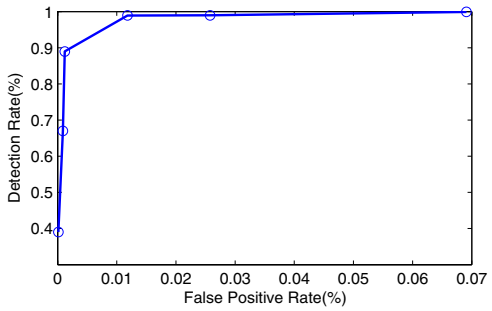

Fig. 4. ROC curve for anomaly detection by using different radius in GNG.

C. Clustering Analysis and Misuse Detection

Following simulations include PCASOM clustering analysis of intrusive connections filtered by GNG, and misuse detectors training using PCANNs. Clearly, PCASOM is able to cluster the datasets with "normal" connections because each cluster in PCASOM can also be regarded as a misuse detector. Therefore, we design several comparative experiments to demonstrate our standpoints by using multiple neural networks based IIDS.

Firstly, we carry out the simulation by PCASOM on the whole training datasets. It means all the training data are not filtered by GNG network in advance. Because of static architecture of PCASOM and for performance comparison, we set the units number as $5, 15, 20, 30$, respectively. A cluster labelling method is also used to calculate the performance indicators as same as Ref.[7]. The results only by PCASOM are shown in Table II.

TABLE II
CLUSTERING RESULTS ONLY BY USING PCASOM.

data type	Detection Rate (%)			
	5 units	15 units	20 units	30 units
normal	92.7	93.5	91.5	91.3
back	94.5	96.0	95.7	80.5
smurf	90.6	93.8	92.0	88.0
neptune	100	100	99.5	99.5
teardrop	20	0	0	0
ipsweep	68.5	78.0	75.4	50.2
portsweep	87.8	91.0	91.0	85.5
satan	78.1	83.5	80.6	60.0
buf-overflow	0	40.0	40.0	40.0
loadmodule	50	0	0	0
guess-passwd	56.5	78.5	60.4	65.7
imap	88.3	91.5	90.0	89.6
warezclient	40	80.0	70.2	70.0

From Table II, we conclude that the best clustering result can be obtained when the units number is 15 in a PCASOM. Meanwhile, the number of training data will influence the performance. For instance, the number of training data for "teardrop" is too small to form a certain cluster; the connections of " loadmodule" is too similar to be separated from

other clusters. Generally, a larger number of training data for one cluster is more helpful for pattern construction.

Secondly, the other simulations follow the basic idea of IIDS. After anomaly detection of GNG, all filtered data will be used to train the PCASOM; then, we input the labeled intrusive data with specified types to train further misuse classifiers. All the results are compared and shown in Table III.

TABLE III
SIMULATION RESULTS FOR INTRUSION DETECTION USING IIDS

data type	GNG+PCASOM	GNG+PCASOM+PCANNs
normal	98.9	-
back	97.5	98.3
smurf	95.0	97.5
neptune	100	100
teardrop	0	-
ipsweep	82.4	88.4
portsweep	96.8	97.9
satan	85.0	94.5
buffer-overflow	40.0	-
loadmodule	0	-
guess-passwd	82.2	91.0
imap	95.0	97.0
warezclient	85.5	98.0

From the comparison between Table II and Table III, some conclusions can be drawn readily: (1). Because of pattern construction for normal activities before intrusions clustering, IIDS scheme obtains higher performance for anomaly detection by GNG than only by PCASOM, i.e., 98.9% vs 93.5%. Furthermore, the detection rate 93.5% for PCASOM means a false positive rate, 6.5%, which is very higher than 1.12% achieved by GNG. It is known that a lower false positive rate is the most important indicator for a practical intrusion detection system. (2). In Table III, PCASOM processed the filtered training data only with intrusive connections (certainly 1.12% of misreported "normal" data is included); the detection rates are higher for almost all the other attacks than that only by PCASOM. The main cause is GNG can give a more accurate pattern for normal data by supervised learning while PCASOM cannot separate those intrusions which is similar to normal data. (3). PCANNs acquired better detection rates than PCASOM clustering for misuse detection. By using adaptive GHA networks, the feature extraction for a specified type intrusion is more adaptive and proper (see comparison in Table III). Certainly, in training stage of a PCANN classifier, the number of training data from a PCASOM cluster should be accumulated to a relative high value; it is very important to built an accurate class pattern.

D. Inner intruders detection

The testing for inner intruders detection is carried out in a LAN. Each time by APR monitoring, IIDS is able to find out the illegal host the moment he loges on the LAN. We check its ARP buffer where there have already existed spoofed MAC/IP information of all legal hosts. Simulations demonstrate the feasibility of inner hosts detection and prevention using ARP principles; the detection rate is 100%. Therefore, not only can the IIDS detect the inner illegal hosts, but also it protects the LAN effectively.

V. CONCLUSIONS

In this paper, we proposed an integrated scheme for intrusion detection (IIDS). Its function can be divided into two parts, outer intrusion detection and inner illegal hosts detection and prevention. We strive for an overall solution for intrusion detection. Multiple neural networks, including a novel methods GNG, a clustering methods PCASOM and adaptive GHA based PCANN are hierarchically integrated to detect intrusions from the Internet; simulations show that the IIDS system can obtain obviously better performance than single method. The second part of IIDS has abilities to detect harmful insiders and prevent them from illegally accessing.

ACKNOWLEDGMENT

This work was supported by Chinese 863 High-Tech Program under Grant 2007AA01Z321.

REFERENCES

[1] Bace and G. Rebecca, Intrusion Detection, *Macmillan Technical Publishing*, 2000.
[2] A. Ghosh and A. Schwartzbard, A study in using neural networks for anomaly and misuse detection, In *Proc. 8th USENIX Security Symposium*, 1999, pp.141-151.
[3] J. Cannady, Artificial neural networks for misuse detection. *Proc. National Information Systems Security Conference(NISSC'98)*, Arlington VA, 1998, pp.443-456.
[4] J. Cannady, J. Mahaffy, The application of artificial intelligence to misuse detection. in *Proc. 1st Recent Advances in Intrusion Detection (RAID) Conference*, Louvain-la-Neuve, Belgium, 1998.
[5] T. Kohonen, Self-Organizing Maps, Berlin, Germany: Springer, 1995.
[6] T. Horeis, Combination of self-organizing maps and radial basis function networks for human expert integration. IEEE Computational Intelligence Society, 2003 Grant Recipients and Final Reports.
[7] S. T. Sarasamma, Q. A. Zhu and J. Huff, Hierachical kohonen net for anomaly detection in network security, *IEEE Trans. SMC - Part B*, vol. 35, no. 2, pp.302-312, Apr. 2005.
[8] A. Rauber, D. Merkl and M. Dittenbach, The growing hierarchical self-organizing map: exploratory analysis of high-dimensional data, *IEEE Trans. Neural Networks*, vol. 13, no. 6, Nov. 2002.
[9] M. Shyu, S. Chen, K. Sarinnapakorn, et al, A novel anomaly detection scheme based on principal component classifier. In *Proc. IEEE Foundations and New Directions of Data Mining Workshop, in Conjunction with the 3rd IEEE International Conference on Data Mining (ICDM'03)*, 2003, pp.172-179.
[10] G. Liu, Z. Yi, S. Yang, A hierarchical intrusion detection model based on the PCA neural networks. *Neurocomputing*, 2007, vol. 70, no. 7-9, pp.561-1568.
[11] G. Liu, Z. Yi, Intrusion detection using PCASOM neural Networks. *Lecture Notes in Computer Science*, 2006, vol. 3973, pp.240-245.
[12] M. Martinetz, S. Berkovich and K. Schulten, Neural-gas network for vector quantization and its application to time series prediction, *IEEE Trans. Neural Networks*, 1993, vol. 4, pp.558-569.
[13] L. Portnoy, E. Eskin, S. J. Stolfo, Intrusion detection with unlabeled data using clustering, In *Proc. ACM CSS Workshop on Data Mining Applied to Security (DMSA-2001)*, Philadelphia, PA, 2001.
[14] E. L. Rubio, J. M. Prez, J. Antonio, et al, A principal components analysis self-organizing map, *Neural Networks*, vol. 17, no. 2, 2004, pp.261-270.
[15] J. Lv, Z. Yi, K. K. Tan, Determining of the number of principal directions in a biologically plausible PCA model, *IEEE Trans. Neural Networks*, 2007, vol. 18, no. 2, pp.910-916.
[16] R. Lippmann, J. Haines, D. Fried, et al, The 1999 DARPA off-line intrusion detection evaluation, *Computer Networks*, vol. 34, 2000, pp.579-595.
[17] KDD Cup 1999 Data, http://kdd.ics.uci.edu/databases/kddcup99, Reference data: Dec. 2007.