Network-Based Intrusion Detection Using Unsupervised Adaptive Resonance Theory (ART)*

Morteza Amini
Computer Engineering Department,
Sharif University of Technology,
Azadi ave., Tehran, Iran
tel: +98-21-6164632
m_amini@ce.sharif.edu

Rasool Jalili
Computer Engineering Department,
Sharif University of Technology,
Azadi ave., Tehran, Iran
tel: +98-21-6164617
jalili@sharif.edu

Abstract

This paper introduces the Unsupervised Neural Net based Intrusion Detector (UNNID) system, which detects network-based intrusions and attacks using unsupervised neural networks. The system has facilities for training, testing, and tunning of unsupervised nets to be used in intrusion detection. Using the system, we tested two types of unsupervised Adaptive Resonance Theory (ART) nets (ART-1 and ART-2). Based on the results, such nets can efficiently classify network traffic into normal and intrusive. The system uses a hybrid of misuse and anomaly detection approaches, so is capable of detecting known attack types as well as new attack types as anomalies.

I. Introduction

Intrusion Detection Systems (IDSs) are amongst the main tools for providing security in computer systems and networks. They detect intrusions and attacks through analyzing audit and log file data. Based on the data source, IDSs are classified into host-based and network-based. Also based on the analyzing approach, IDSs are categorized into misuse detection and anomaly detection systems. Misuse detection systems detect known attacks using priori defined attack patterns and signatures. Anomaly detection systems detect attacks by observing deviations from normal behavior of the system, network or their users [3,13].

Some early research on IDS attempted to use neural nets for intrusion detection. Such systems were trained on normal or attack behavior information and then detect intrusions or attacks. Supervised and unsupervised nets have been used in IDSs. Most supervised neural net architectures requires retraining, in order to improve analysis capability due to changes in the input data. Unsupervised nets offer an increased level of adaptability to neural nets, and have been used in intrusion detection systems [4]. Adaptive Resonance Theory (ART) is a type of neural nets with the capability of unsupervised training as well as efficient classification of the input data [7]. Accordingly, we first designed an IDS named UNNID (Unsupervised Neural Net based Intrusion Detector), using unsupervised neural nets. Then we employed ART (Adaptive Resonance Theory) nets in the system for clustering and classifying of network traffic in order to detect intrusive or attack traffic along with its type. UNNID has flexibility to change structure and parameters of ART neural nets (including ART-1 and ART-2) for training and testing in different situations. We trained and tested our system using KDD Cup's 99 dataset which covers four categories of attacks: Denial of Service (DoS) attacks, User-to-Root (U2R) attacks, Remote-to-Local (R2L) attacks, and Probing [25].

The rest of the paper organized as follows: Section II discuses related works in the field of intrusion detection with neural nets. Section III discuses system architecture and main components of UNNID. Section IV describes characteristics of ART nets and how to use such nets for classifying intrusive network traffic from normal ones. Section V presents experimental results of using ART in intrusion detection. Section VI draws some conclusions and future works.

II. Related Works

Since 1990, many works and research have been

* This project is partially supported by Iran Telecommunication Research Center (ITRC).
This paper is partially supported by Bank Refah.

done in the field of intrusion detection using neural nets, for misuse detection as well as anomaly detection. According to the type of neural net being used, neural net-based IDSs can be classified into the following three categories.

The first category is systems built on Multi Layers Feed-Forward (MLFF) neural nets, such as MLP and BP. MLFF neural nets have been used in most of the prime research in neural net-based IDSs. Works including [20] and [22] used MLFF neural nets for anomaly detection based on user behaviors. MLFF nets trained on known attack patterns or signatures are used in misuse detection in works including [4] and [9]. Remainder of works including [1] and [18], focused on incorporating MLFF nets with other techniques (such as keyword selection and especially expert system) to achieve accurate detection of intrusions.

The second category is systems built on recurrent and adaptive neural nets such as ELMAN and CMAC. Giving a feed back from the output of the net or its protected system in such systems causes the neural net to preserve correlation of current system inputs with previous system inputs and states. Instances of systems in this category presented in [2, 5, 6]. Debar and his colleagues in [6] and [5] correspondingly used simplified ELMAN recurrent net (GENT) and multi layer recurrent net (with back propagation learning rule) to predict the next acceptable command. In [2], CMAC (Cerebellar Model Articulation Controller) net, which is a form of adaptive neural nets, has been applied to intrusion detection. The resulting system is capable to learn new attacks autonomously by modified reinforcement learning method that uses feed back from the performance of a protected system.

The third category uses unsupervised learning neural nets to classify and visualize system input data to separate normal behaviors from abnormal or intrusive ones. Most of the systems in this category used self-organizing maps (SOMs) neural nets. For the first time, Fox [8] used SOM to learn the characteristics of normal system activity and identify statistical variations from the normal ones that may be an indication of a virus. In [19], multiple SOMs are used for intrusion detection, where a collection of more specialized maps are used to process network traffic for each protocol separately. Each neural net was trained to recognize the normal activity of a single protocol. Gardian in [10] used SOM for visualizing the network activity that provides new ways for network administrators to explore, track and analyze intruders. This approach is different from anomaly and misuse detection and considers human factors to support the exploration of network traffic and judgment about anomaly packets. Hoglund et al in [11] trained SOM on a collection of normal data from UNIX audit data and used it for detecting abnormal or anomalous user activity. Jirapummin in [12] proposed an alternative methodology employing hybrid neural network for both visualizing intrusions using Kohenen's SOM and classifying intrusions using Resilient Propagation neural network (RPROP).

In [17] hierarchical SOMs are applied to examine session data by users on a UNIX system in order to find behavioral anomalies. In [24] a Hierarchical Intrusion Detection (HIDE) system is introduced which is able to detect network based attacks as anomalies using statistical preprocessing and neural net classification. Five different type of neural net classifiers were evaluated in this system and compared together. Evaluated types were Perceptron, Back Propagation (BP), Perceptron-Back propagation-Hybrid (PBH), Fuzzy ARTMAP and Radial-Based Function. In [16], a two-level hierarchical SOM is applied for intrusion detection. The system has emphasis on representation of time and incremental development of a hierarchy. The SOM in this system is able to detect attack patterns over a sequence of connections. The system developed in [14] (named NSOM), uses an structured SOM to classify real-time Ethernet network data. The system is able to classify DoS attacks graphically as opposed to normal traffic by demonstrating that the clustering of neurons is very different between the two. In [15] statistical-based methods are used for anomaly detection, where active users are compared to historical profiles. The user is identified as normal, if closely matched to his/her historical profiles. Using ART-2 net for clustering users by command profiles in this wok largely improved the prediction rate.

III. UNNID System Architecture

The architecture and main components of our UNNID system is shown in figure 1. The system is designed to 1) facilitate training, testing, tunning and evaluating different types of unsupervised neural nets for intrusion detection, 2) apply them for analyzing network traffic in on-line and off-line mode and classifying network traffic into normal and attack.

In UNNID, *Data Provider* collects data from network audited data file (off-line mode) or live network (on-line mode) and send text data to the *PreProcessor* component. PreProcessor converts text data into numeric and if needed convert numeric data into binary or normalized form, and send them to *Neural Net Based Analyzer*. The analyzer uses data either for training and testing its neural net or for analyzing and detecting intrusions/attacks. The analyzer output (normal or attack type) is given to *Responder* for recording in the system log file and generating alarm in case an attack is detected. The *IDS Evaluator*

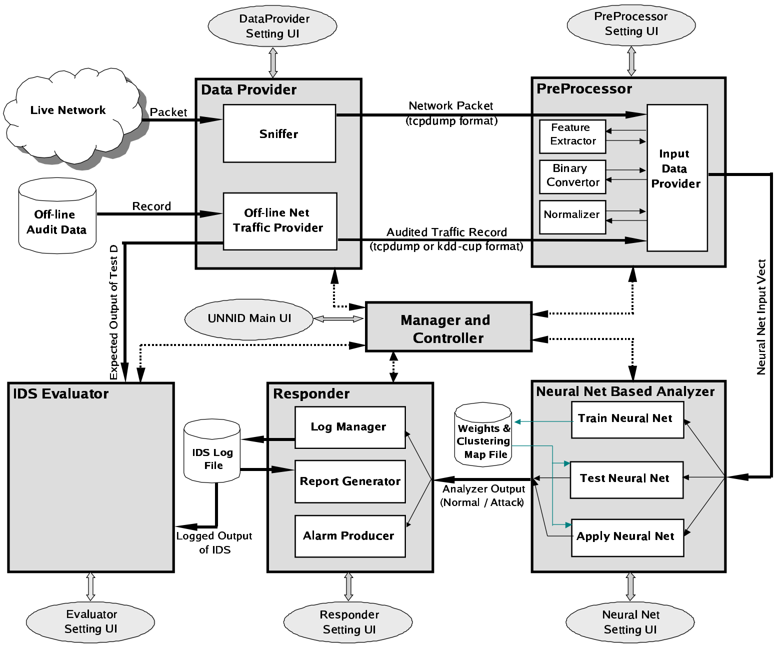

Figure 1. Unsupervised Neural Net-based Intrusion Detector (UNNID) System Architecture.

component provides a facility for reporting true detection rate, false positive detection rate, false negative detection rate, and other criteria to evaluate our unsupervised neural net-based intrusion detection system. In brief, UNNID works in four modes: 1) off-line training, 2) off-line testing, 3) as a real on-line IDS, and 4) as a real off-line IDS. The *Manager & Controller* component, manages and directs other component to work in one of the above modes based on the command and parameters delivered from the operator.

Our focus in this paper is on using UNNID for tuning and testing an ART net with KDD Cup's 99 dataset. Accordingly, the details of the corresponding component is described as follows.

A. Data Provider: This component has two sub-components: *Sniffer* and *Off-Line Net Traffic Provider*. Sniffer is used for on-line mode and can filter and capture live network packets in the *tcpdump* format by switching the network adapter to the promiscuous mode. This sub-component has not been used in this study. Off-Line Net Traffic Provider manages the network audit file and sends records (in the text format) to PreProcessor. The file can be either in the *tcpdump* or *kdd-cup* format. However the data with *kdd-cup* format, from KDD Cup's 99 dataset, has been used in this study.

The KDD Cup's 99 dataset is a feature extracted data source for experimental studies. The 1998 DARPA Intrusion Detection Evaluation Program provided a standard set of audited data, which includes a wide variety of intrusions simulated in a U.S. Air Force LAN environment. The 1999 KDD intrusion detection contest used a version of this dataset. The raw training data was about 4 GB of compressed binary TCP dump data from seven weeks of network traffic. This was processed into about five million connection records. Similarly, the two weeks of test data yielded around two million connection records [25]. For each connection, 41 features were defined, categorized as Basic TCP features, Content features, Time-based traffic features, and Host-based traffic features[16].

Each connection is labeled as either normal, or attack, with exactly one specific attack type. The dataset contains a total of 24 attack types with an additional 14 unseen attack types in the test data only. Attacks fall into the following main categories [25]:

- Denial of Service (DoS) attack: deny legitimate requests to a system, e.g. syn flood;
- Remote-to-Local (R2L) attack: unauthorized access from a remote machine, e.g. guessing password;

- User-to-Root (U2R) attack: unauthorized access to local super user (root) privileges, e.g. various buffer overflow attacks;
- Probing: surveillance and other probing, e.g. port scanning.

B. PreProcessor: The PreProcessor component gets traffic data from Data Provider, extracts appropriate features, converts features into numerical form and then convert into binary or normalized form in order to feed the neural net sensors in Neural Net based Analyzer component.

This study deals with *kdd-cup* format data. Each record in *kdd-cup* format has 41 features, each of which is in one of the continuous, discrete and symbolic form, with significantly varying ranges. Based on the type of neural nets, the input data may have different forms and so needs different preprocessing. Some neural nets only accept binary input and some can also accept continuous-valued data. In PreProcessor, after extracting *kdd-cup* features from each record, each feature is converted from text or symbolic form into numerical form. For converting symbols into numerical form, an integer code is assigned to each symbol. For instance, in the case of protocol_type feature, 0 is assigned to *tcp*, 1 to *udp*, and 2 to the *icmp* symbol. The next step in preprocessing is converting data into binary, or normalized and scaled form. For normalizing feature values, a statistical analysis is performed on the values of each feature based on the existing data from KDD Cup's 99 dataset and then acceptable maximum value for each feature is determined. According to the maximum values and the following simple formula, normalization of feature values in the range [0,1] is calculated.

```
If ( f > MaxF )   Nf=1;
Otherwise         Nf = ( f / MaxF)
```

F: Feature
f: Feature value
MaxF: Maximum acceptable value for *F*
Nf: Normalized or scaled value of *F*

For converting feature values into binary format, the nature of the values is considered. If the variation range of feature values is small and the values are integer, direct conversion from integer value into binary code will be done. Otherwise, based on the distribution of feature values, the [0 , MaxF] interval is divided into some subintervals and then a binary unique code is assigned to each subinterval. At this point, each feature value is converted into the binary code of its containing subinterval. As the smaller the subintervals are and more in number, the accuracy of this conversion is higher. However more subintervals leads to longer binary code, which in turn increases the training period as well as the response time of the neural net. To solve this problem, a higher resolution can be used for the rang of more integrated values and a lower resolution for the other range. Assuming that values of a feature have a normal distribution, the range of $[\overline{F}-\delta, \overline{F}+\delta]$ (i.e. $\overline{F}$ is average of feature F values and δ is standard deviation of them) is the place of more integration of feature values.

In addition to preprocessing the connection records in *kdd-cup* format, PreProcessor has facilities for preprocessing headers of packets in *tcpdump* format to feed the neural net and also processing the network packets (in *tcpdump* format) into connections (in *kdd-cup* format). This feature has not been used in this study.

C. Neural Net based Analyzer: The main component of UNNID is Neural Net based Analyzer, which analyzes the network traffic and detects intrusions and attacks (after getting the system into operation). Moreover, this component provides facilities for training and testing an unsupervised neural net for intrusion detection purpose (before bringing the system into application in the real environment). The analyzer receives appropriate preprocessed input data from PreProcessor and after analyzing the data, sends its results to the Responder component. As unsupervised neural nets can classify input data based on their similarity, ART nets are used in UNNID for clustering and classifying network traffic into normal and intrusive. Selecting the type of neural net and tuning its parameters can be done through the UNNID Neural Net Setting UI graphical user interface.

D. Responder: Response to the detected intrusions and attacks (by the analyzer) is achieved by this component. Responder has facilities for logging detected attacks, generating alarms (in different ways such as sending e-mail to system administrators and displaying the appropriate message on the screen) and generating detailed or statistical reports on the collected data in IDS log files.

E. IDS Evaluator: this component is included in the architecture for evaluating the IDS outputs in the test phase (after system training). IDS Evaluator calculates the following criteria by comparing the output of IDS and expected output of the system, which is determined by labels on records of test data:

- Exact True Type Detection Rate (detecting normal traffic from attack and recognizing the known attack type);

- True Detection Rate (only separating normal traffic from attack);
- False Positive Detection Rate (mis-detecting attack);
- False Negative Detection Rate (failing to detect attack when it is occurred).

IV. Adaptive Resonance Theory Classifier

Adaptive Resonance Theory (ART) was invented by Stephen Grossberg in 1976. Later on, ART came in several flavors, both supervised and unsupervised. There are various unsupervised ART algorithms such as ART-1, ART-2, ART-3 and Fuzzy ART; and various supervised ones named with the suffix "MAP" such as ARTMAP, Fuzzy ARTMAP and Gussian ARTMAP [23]. Our focus in this paper is on the unsupervised ART nets that developed before supervised ones.

In unsupervised ART nets, input patterns may be applied several times and in any order. Each time a pattern is applied, an appropriate cluster unit is chosen and related cluster weights are adjusted to let the cluster unit learn the pattern. In such nets, choosing a cluster is based on the relative similarity of an input pattern to the weight vector for a cluster unit, rather than the absolute difference between the vectors (that is used in SOM nets). As in most cases of clustering nets, the weights on a cluster unit may be considered to be an exemplar (or code vector) for the patterns placed on that cluster [7]. ART nets are designed to allow the user to control the degree of similarity of patterns placed on the same cluster. This can be done by tuning the *vigilance parameter* in such nets. In ART nets, the number of clusters is not required to be determined previously, so the vigilance parameter can be used to determine the proper number of clusters in order to decrease the probability of merging different types of clusters into the same cluster. Moreover, ART nets have two other main characteristics, stability and plasticity. *Stability* means a pattern not oscillating among different cluster units at different stages of training, and *plasticity* means the ability of net to learn a new pattern equally well at all stages of learning [7, 15].

Stability and plasticity of ART nets and the capability of clustering input patterns based on the user controlled similarity between them, made such nets more appropriate for using in IDSs, rather than the other types of unsupervised nets including SOM, for classifying network traffic into normal and intrusive/ attack. Accordingly, we used two types of unsupervised ART nets, ART-1 and ART-2 [7]. ART-1 is used for clustering binary inputs and ART-2 is used to accept continuous-valued vectors.

In UNNID, appropriate input vector (binary vector for ART-1 and normalized/scaled continuous-valued vector for ART-2) is fed into ART net by PreProcessor. In the training phase, input vectors are clustered through ART nets regardless of their nature (normal or intrusive). Following the training phase, system must determine the neurons of each type of cluster and assign name to each cluster using the label of connection records (in training data). Each cluster has the same name as its units. Each unit is named based on the type of the majority of input data that the unit represent the winning or best matching for. This reduces to constructing a *Clustering Map.* In the map, units are clustered together to indicate either the normal traffic, known trained attacks, or possibly a new attack. New attacks may appear in abnormal traffic, which is neither a normal traffic nor a known attack.

Considering the above architecture, we use both normal and known attacks network traffic for training the *Neural Net based Analyzer* and then detect known attacks and also abnormal traffic as new attacks. In the other words, we combined misuse detection and anomaly detection approaches together using an ART net. This characteristic of UNNID, offers the advantages and abilities of both approaches in detection and recognition of known attacks as well as novel or new ones.

V. Experimental Results

We implemented UNNID under Red Hat Linux 9.0 operating system using C++ as a programming language and Qt-Designer 3.1.1 software tools for designing GUI. For training UNNID, 10,000 connection records were selected randomly from the training dataset in the way that the result contains records of 22 attack types. Correspondingly, for testing the system, 5000 connection records were selected from the test data in the way that the result contains records of all the 22 known attack types plus 14 new unseen attack types.

We evaluated the performance of ART-1 and ART-2 based on the Exact True Type detection Rate (ETTR), True detection Rate (TR), False Positive detection Rate (FPR), and False Negative detection Rate (FNR). To evaluate these criteria, we first determined the best value of important parameters of each net (essentially number of epochs for training and vigilance parameter) and then evaluate capabilities of these nets from different aspects. The number of units in the output layer (F2 layer) in ART-1 set to 3000 and in ART-2 to 2000.

To determine the appropriate number of epochs in training, ART-1 with vigilance value of 0.9 was trained with different number of epochs. Having nets of no over-training and low-training problems, the result of experiments recommended the range 75 to 125 as the number of epochs. Accordingly we chose 100 epochs for performing the main evaluation. Repeating this experiment for ART-2 with vigilance value of 0.999 resulted the same. Again, we chose 100 number of epochs for training the net in the other experiments.

To specify the appropriate value of vigilance parameter and its effect on the system performance, the system was trained with different vigilance values and ETTR, TR, FPR and FNR criteria were evaluated. Results of the experiment is depicted in figure 3.

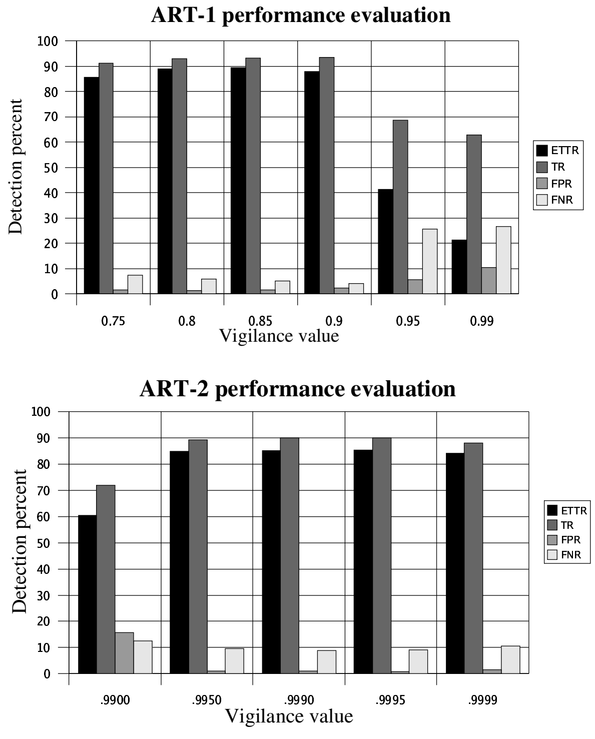

Figure 3. Performance evaluation of ART nets.

The results show that ART-1 with vigilance value of 0.9 and ART-2 net with vigilance value of 0.999 offers the best level of detection performance. Performance results of best performing instance of each ART nets are shown in table 1.

Table1. Detection performance of ART nets in best situation.

	ETTR	*TR*	*FPR*	*FNR*
UNNID ART-1	88.04	93.6	2.24	4.16
UNNID ART-2	85.01	90.04	1.06	8.9

For ART-1 and ART-2, true detection rate of the four attack categories are shown in table 2. Comparing the results with the best results which are presented in [21] is satisfactory. The results show that both ART nets have a good acceptable level of detection performance for DoS and Probe attack categories, ART-1 has had a better performance than ART-2. In the case of R2L, ART-1 offers an acceptable rate of detection while ART-2 dose not. In the case of U2R, both ART nets dose not offer an acceptable level of detection performance.

Table2. Detection rate of ART nets in best situation and best result of [21] for each attack category.

	DoS	*R2L*	*U2R*	*Probe*
UNNID ART-1	100	88.69	17.41	99.48
UNNID ART-2	96.17	36.31	10.71	96.88
Best result of [21]	97.3	9.6	29.28	88.7

Comparison between ART-1 and ART-2 shows that, ART-1 has a better performance than ART-2. However, considering the response time, ART-2 is 7 to 8 times faster than ART-1 and also ART-2 is more appropriate than ART-1 for using in real-time intrusion detection systems.

VI. Conclusion

In this paper, we introduced an Unsupervised Neural Net based Intrusion Detector (UNNID) system for classifying network traffic using different types of unsupervised neural nets. The system is used to tune, train and test two types of Adaptive Resonance Theory (ART) nets, (ART-1 and ART-2). We have experienced how to use ART nets for classifying network traffic into normal and attack and also recognizing the known trained attacks (along with their types) as well as new unseen attacks as anomalies. The results show that ART-1 in 93.5 percent of times and ART-2 in 90.7 percent were able to recognize attack traffic from normal one.

Using unsupervised neural nets in intrusion detection have many advantages rather than supervised nets. This includes the capability of unsupervised nets to improve their analysis of new data over time without the requirement of retraining over all the previous and new data. Accordingly, in our future investigations, we intend to apply and compare other types of unsupervised nets for classifying network traffic.

Acknowledgments

We're grateful to Dr. Saeed Bagheri of Sharif University of Technology for his useful comments and direction in performing this project.

References

[1] J.M. Bonifacio, "Neural Networks Applied in Intrusion Detection Systems", *Neural Networks Proceedings, IEEE World Congress on Computational Intelligence*, vol. 1, pages: 205-210, 1998.

[2] J. Cannady, "Applying CMAC-based Online Learning to Intrusion Detection", *Neural Networks, Proceeding of the IEEE-INNS-ENNS International Joint Conference*, vol 5, pages: 405-410, 2000.

[3] R. Coolen and H.A.M. Luiijf, "Intrusion Detection: Generics and State-of-the-Art", *Research and Technology Organization (RTO) Technical Report 49*, 2002.

[4] J. Cannady, "Artificial Neural Networks for Misuse Detection", *Proceedings of the National Information Systems Security Conference*, 1998.

[5] H. Debar and B. Dorizzi, "An Application of Recurrent Network to An Intrusion Detection System", *Proceeding of the International Joint Conference on Neural Networks,* pp. 478-483, 1992.

[6] H. Debar, M. Becker and D. Siboni, "A Neural Network Component for An Intrusion Detection System", *Proceedings of the IEEE Computer Society Symposium,* pp. 240-250, 1992.

[7] L. Fausett, "Fundamentals of Neural Networks", Prentice-Hall, 1994.

[8] L. Fox Kevin, R. Henning Rhonda and H. Reed Jonathan, "A Neural Network Approach Towards Intrusion Detection", *Proceeding of the 13th National Computer Security Conference*, 1990.

[9] A. K. Ghosh and A. Schwartzbard, "A Study in Using Neural Network For Anomaly and misuse Detection", *Proceedings of the 8th USENIX Security Symposium*, 1999.

[10] L. Girardin, "An Eye on Network Intruder-Administrator Shootouts", *Proceedings of the First USENIX Workshop on Intrusion Detection and Network Monitoring, Santa Clara*, USA, 1999.

[11] A.J. Hoglund *et al*, "A Computer Host-based User Anomaly Detection System using Self-Organizing Map", *Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks, Vol. 5, pp. 411-416*, 2000.

[12] C. Jirapummin, N. Wattanapongsakorn and P. Kanthamanon, "Hybrid Neural Networks for Intrusion Detection System", *Proceedings of the 2002 International Technical Conference on Circuits / Systems, Computers and Communications (ITC-CSCC 2002)*, pp. 928-931, Thailand, 2002.

[13] S. Kummar, "Classification and Detection of Computer Intrusions", *PhD Thesis, Purdue University*, 1995.

[14] K. Labib and R. Vemuri, "NSOM: A Real-Time Network-Based Intrusion Detection System Using Self-Organizing Maps", *Networks and Security,* 2002 .

[15] T. Li, "Behavioral Clustering and Statistical Intrusion Detection", *M.Sc. Thesis, Florida State University*, 1997.

[16] P. Lichodzijewski, A. N. Zincir-Heywood and M. I. Heywood, "Dynamic Intrusion Detection Using Self-Organizing Maps", *Proceedings of the 14th Annual Canadian Information Technology Security Symposium, CITSS*, 2002.

[17] P. Lichodzijewski, A. N. Zincir-Heywood and M. I. Heywood, "Host-Based Intrusion Detection Using Self-Organizing Maps", *Proceedings of the 2002 IEEE World Congress on computational Intelligence*, 2002.

[18] R. P. Lippmann and R. K. Cunningham, "Improving Intrusion Detection Performance Using Keyword Selection and Neural Networks", *RAID '99, Computer Networks*, Vol. 34, No. 4, 2000.

[19] B.C. Rhodes, J.A. Mahaffey, J. D. Cannady, "Multiple Self-Organizing Maps for Intrusion Detection", *Proceedings of the 23rd National Information Systems Security Conference*, 2000.

[20] J. Ryan, M. Lin, R. Miikkulainen, "Intrusion Detection with Neural Networks", *Advances in Neural Information Processing Systems*, Vol 10, The MIT Press, 1998.

[21] M. Sabhnani and G. Serpen, "Application of Machine Learning Algorithms to KDD Intrusion Detection Dataset within Misuse Detection Context", http://www.eecs.utoledo.edu/~serpen/professional/Research/Publication/MLMTA 2003 Manuscript Submission Version.pdf, July 2003.

[22] K. Tan, "The Application Of Neural Networks to UNIX Computer Security", *Proceedings of the IEEE International Conference on Neural Networks,* Vol 7, pp. 476-481, 1995.

[23] D. Tauritz, "ART: An overview of the field", http://web.umr.edu/~tauritzd/art/overview.html, April 2003.

[24] Z. Zhang, J. Li, C. N. Manikopoulos, J. Jorgenson and J. Ucles, "HIDE: A Hierarchical Network Intrusion Detection System Using Statistical Preprocessing and Neural Network Classification", *Proceedings of the 2nd Annual IEEE Systems, Mans, Cybernetics Information Assurance Workshop,* West Point, NY, 2001.

[25] The 3rd *International* Knowledge Discovery and Data Mining Tools Competition, http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html, 2003.