available at www.sciencedirect.com

ScienceDirect

journal homepage: www.elsevier.com/locate/cose

Computers & Security

RT-UNNID: A practical solution to real-time network-based intrusion detection using unsupervised neural networks

Morteza Amini[*], *Rasool Jalili*, *Hamid Reza Shahriari*

Department of Computer Engineering, Sharif University of Technology, Azadi Avenue, Tehran, Iran

ARTICLE INFO

Article history:
Received 25 April 2005
Revised 4 April 2006
Accepted 18 May 2006

Keywords:
Network security
Intrusion detection systems
Misuse detection
Anomaly detection
Unsupervised neural network
Self-organizing map
Adaptive resonance theory

ABSTRACT

With the growing rate of network attacks, intelligent methods for detecting new attacks have attracted increasing interest. The RT-UNNID system, introduced in this paper, is one such system, capable of intelligent real-time intrusion detection using unsupervised neural networks. Unsupervised neural nets can improve their analysis of new data over time without retraining. In previous work, we evaluated Adaptive Resonance Theory (ART) and Self-Organizing Map (SOM) neural networks using offline data. In this paper, we present a real-time solution using unsupervised neural nets to detect known and new attacks in network traffic. We evaluated our approach using 27 types of attack, and observed 97% precision using ART nets, and 95% precision using SOM nets.

1. Introduction

The increasing reliance on networked computers, and the growing expertise in subverting such systems, makes intelligent and adaptive threat detection vital.

Computer security revolves around confidentiality, integrity, and availability. Integrity refers to the trustworthiness of data or resources, and is usually phrased in terms of preventing improper or unauthorized change. Integrity mechanisms fall into two classes: prevention or detection (Bishop, 2003).

Prevention mechanisms try to maintain the integrity of data by blocking unauthorized attempts to change data. On the other hand, detection mechanisms do not try to prevent violations of integrity, but simply report that data integrity can no longer be assumed (Bishop, 2003). Intrusion Detection Systems (IDSs) attempt to detect intrusion and attacks through analyzing events in computer systems or networks. IDSs can be classified as being based on anomaly detection or misuse detection depending on how they analyse data (Cannady, 1998; Coolen and Luiijf, 2002).

Misuse detection systems detect known attacks using attack patterns and signatures known a priori, while anomaly detection systems detect attacks by observing deviations from normal behaviour of the system, network, or users (Amini and Jalili, 2004).

Some early research on IDSs explored neural nets for intrusion detection. These can be used only after training on normal or attack behaviours, or combination of the two. Both supervised and unsupervised neural nets have been

[*] *Corresponding author*. Tel.: +98 21 66164019; fax: +98 21 66164020.
E-mail addresses: m_amini@ce.sharif.edu (M. Amini), jalili@sharif.edu (R. Jalili), shahriari@ce.sharif.edu (H.R. Shahriari).

0167-4048/$ – see front matter © 2006 Elsevier Ltd. All rights reserved.
doi:10.1016/j.cose.2006.05.003

used. Most supervised neural net architectures require retraining to improve analysis on varying input data, but unsupervised nets, which offer greater adaptability, can improve their analysis capability dynamically (Cannady, 1998).

In this paper, we introduce RT-UNNID (Real-Time Unsupervised Neural-Net-based Intrusion Detector). This can detect network-based attacks using unsupervised neural nets in real-time, and has facilities for training, testing, and tuning of unsupervised nets for intrusion detection purpose. Using the system, we evaluated two types of unsupervised Adaptive Resonance Theory nets (ART-1 and ART-2) and a traditional unsupervised Self-Organizing Map (SOM) net. We present a practical solution for using unsupervised neural nets for real-time intrusion detection, compare the performance of such neural nets in real-time intrusion detection, and introduce ART nets as a better solution for dynamic IDSs.

The remainder of the paper is organised as follows: Section 2 discusses related work on intrusion detection using neural networks. Section 3 describes a practical way toward using unsupervised neural networks in intrusion detection. Section 4 introduces the RT-UNNID system and describes its main components. Section 5 focuses on data feature selection and preprocessing in this system. Section 6 discusses the unsupervised neural-net-based engine and how ART and SOM neural nets may be used. Section 7 presents experimental results, and Section 8 draws conclusions and describes future work.

2. Related work

Neural-net-based IDSs can be classified into the following four categories.

2.1. MLFF neural-net-based IDSs

The first category includes the systems built on Multi-Layer Feed-Forward (MLFF) neural nets, such as the Multi-Layer Perceptron (MLP) and Back Propagation (BP). MLFF neural nets have been used in most early research in neural-net-based IDSs. Works including Ryan et al. (1998) and Tan (1995) used MLFF neural nets for anomaly detection based on user behaviours. MLFF nets that trained on known attack patterns or signatures were used for misuse detection in Cannady (1998) and Ghosh and Schwartzbard (1999), while Bonifacio (1998) and Lippmann and Cunningham (2000) focused on incorporating MLFF nets with techniques such as keyword selection and expert systems. Other researchers have compared the effectiveness of MLFF neural nets to other methods such as Support Vector Machine (SVM) and Multivariate Adaptive Regression Splines (MARS) (Mukkamala et al., 2002, 2004); MLFF neural nets have been shown to have lower detection performance than SVM and MARS.

2.2. Recurrent and adaptive neural-net-based IDSs

This category includes systems built on recurrent and adaptive neural nets such as ELMAN and CMAC. By getting feedback from its output or its protected system, the neural net preserves the correlation of current system inputs with previous system inputs and states (Cannady, 2000; Debar et al., 1992; Debar and Dorizzi, 1992). Debar et al. used a simplified ELMAN recurrent net (GENT) and multi-layer recurrent net with back-propagation to predict the next acceptable command (Debar et al., 1992; Debar and Dorizzi, 1992). Cannady has applied the CMAC (Cerebellar Model Articulation Controller) net – a form of adaptive neural nets – to learn new attacks autonomously by modified reinforcement learning (Cannady, 2000).

2.3. Unsupervised neural-net-based IDSs

The third category uses unsupervised learning neural nets to classify and visualize system input data to separate normal behaviours from abnormal or intrusive ones. Most of the systems in this category use Self-Organizing Maps (SOMs), while a few use other types of unsupervised neural nets. Fox (Kevin et al., 1990) was the first to apply an SOM to learn the characteristics of normal system activity and identify statistical variations from the normal trends. In Rhodes et al. (2000), multiple SOMs are used for intrusion detection, where a collection of specialized maps are used to process network traffic for each protocol such as TCP, UDP, and ICMP. Each neural net is trained to recognise the normal activity of a single protocol. Girardin in Girardin (1999) used SOM for visualizing the network activity that provides new ways for network administrators to explore, track, and analyse intruders. This approach is different from both anomaly and misuse detection and considers human factors to support the exploration of network traffic and judgment about anomaly packets. Höglund et al. (2000) trained SOM on a collection of normal data from UNIX audit data and used it for detecting anomalous user activity. Li used statistical methods for anomaly detection; active users are compared to historical profiles, and are classified as normal if their behaviour closely matches their historical profiles (Li, 1997). Using ART-2 net for clustering users by command profiles in this system greatly improved the prediction rate.

Some recent research has explored using multiple neural nets in a hierarchical structure to improve classification accuracy. In Lichodzijewski et al. (2002b), hierarchical SOMs are applied to examine session data by users on a UNIX system in order to find behavioural anomalies. In Zhang et al. (2001), a Hierarchical Intrusion Detection (HIDE) system is introduced which can detect network-based attacks as anomalies using statistical preprocessing and neural net classification. Five different types of neural net classifiers – Perceptron, Back Propagation (BP), Perceptron–Back propagation-Hybrid (PBH), Fuzzy ARTMAP, and Radial-based Function – were evaluated. In Lichodzijewski et al. (2002a), a two-level hierarchical SOM was applied to detect intrusions. The system has emphasis on the 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 NSOM system described in Labib and Vemuri (2002) uses a structured SOM to classify real-time Ethernet network data, and can classify DoS attacks graphically as opposed to normal traffic by demonstrating that the clustering of neurons is very different between them.

2.4. Hybrid neural-net-based IDSs

The last category of neural-net-based IDSs encompasses systems that combine supervised and unsupervised neural nets. Jirapummin in Jirapummin et al. (2002) proposed employing hybrid neural network for both visualizing intrusions using Kohenen's SOM and classifying intrusions using a Resilient Propagation neural network (RPROP). Horeis (2003) used a combination of SOM and Radial Basis Function (RBF) nets. The system offers generally better results than IDSs based on RBF nets alone.

Integration and combination of neural-net-based IDSs (as an intelligent component in detecting variations of known and especially unknown attacks), with other preventive techniques such as firewalls and access control is a new research area. A sample of this research has been introduced by Yoo and Ultes-Nitsche (2002). The main purpose of their research was integrating a smart detection engine (based on neural nets) into a firewall. The presented system not only detects anomalous network traffic as in classical IDSs, but also detects unusual structures in data packets that suggest the presence of virus data.

3. Toward a practical neural-net-based IDS

Surveying the research performed on intrusion detection based on neural nets, raises the following two questions:

1. Why did the research often use Self-Organizing Map (SOM) nets, and not use other more complicated types of unsupervised neural nets, which probably have higher capability?
2. Why did not the research, result in a practical system for intrusion detection? There exist many IDSs which are merely put forward in theoretical and laboratorial system forms.

To answer the above questions, the idea of designing a flexible system named UNNID was conceived for applying various types of unsupervised neural nets in intrusion detection. This system was constructed to provide the facilities for tuning, testing, and applying Adaptive Resonance Theory (ART) and Self-Organizing Map (SOM) neural nets in intrusion detection. The system was used to detect the malicious attacks in the real network which are possible to take place beyond the laboratorial environment.

In our previous work (Amini and Jalili, 2004), we compared detection performance of SOM and two types of ART nets (ART-1 and ART-2) using the KDD-Cup's 99 data set. The KDD-Cup's 99 data set is an offline and connection based data which has been extracted from DARPA 98 standard data set and covers four categories of attacks: Denial of Service (DoS), User-to-Root (U2R), Remote-to-Local (R2L), and Probing attacks (KDD, 2003).

To detect intrusions in real-time, we extended the UNNID system to the RT-UNNID system. This system receives a network traffic through sniffing the network and detects predefined as well as new attacks in real-time.

Using the implicit representation of time in the system substantially increased the performance of detecting certain types of attacks (especially denial of service attacks). In the following sections, we introduce the RT-UNNID system and describe its design aspects, and present the results of using the system in comparing the efficiency of SOM, ART-1 and ART-2 in detecting network-based attacks and intrusions.

4. The RT-UNNID system

Main components and data-flow diagram of the RT-UNNID system are shown in Fig. 1. The Sniffer component collects all network traffic by setting network adapter in promiscuous mode. Preprocessor extracts numerical features from delivered packets and sends them to Unsupervised Neural-Net-based Engine (UNN-Engine) after converting these features into the binary or normalized form. In the training phase, UNN-Engine uses these data to train its neural network, but in the operation phase, UNN-Engine uses these data for real-time attack detection. In the operation phase, the output of UNN-Engine (which can be normal or be the type of an attack) is given to Responder for recording in the system log file and generating appropriate alarm in the case of detecting an attack.

5. Data features and preprocessing

For real-time detection of attacks in RT-UNNID, content of packet headers have been used. Since the main goal in designing RT-UNNID is to detect attacks and intrusions against TCP/IP protocol, it is necessary to convert the header data of different types of packets (UDP, TCP, and ICMP) into a canonical form with fixed length. Each delivered packet from Sniffer

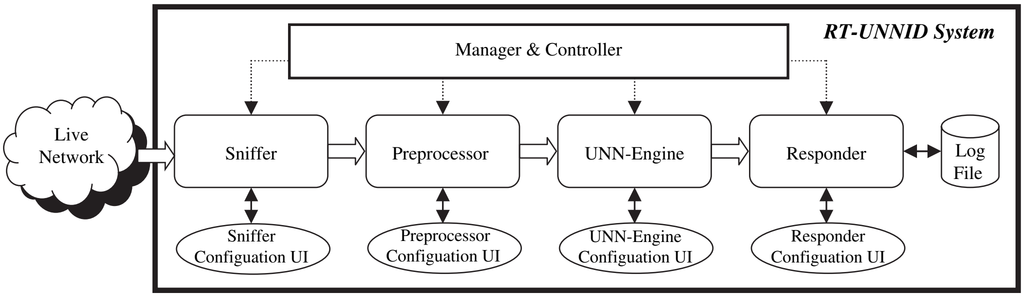

Fig. 1 – RT-UNNID main components and data-flow diagram.

(in the tcpdump format (TcpDump, 2004)) is converted into the canonical form consisting of 27 features. The features are partitioned into the following four categories:

- IP header features
- TCP header features
- UDP header features
- ICMP header features

Details of each category are shown in Table 1.

For example, to convert a TCP packet into the canonical form, it is required to extract the IP header and TCP header features from the packet and use zero for the UDP and ICMP header features. Fig. 2 shows a sample record in the canonical form resulted from a TCP packet.

Preprocessing in RT-UNNID comprises of extracting features from each packet, converting them into the canonical form, and then transforming the resulted record into a numerical vector, and finally converting the numerical vector into the binary or normalized (in interval [0,1]) form depending on the type of the unsupervised neural net which is used in UNN-Engine. The complete process of preprocessing in RT-UNNID is shown in Fig. 3.

5.1. *Feature selection*

Two significant practical issues have been considered for feature selection in RT-UNNID. The first issue is substitution of *source-ip-address* and *destination-ip-address* with *is-home-source-ip* and *is-home-destination-ip* consequently in the selected features. In some previous related work (Bonifacio, 1998; Cannady, 1998; Girardin, 1999; Labib and Vemuri, 2002), source and destination IP addresses were used directly in intrusion detection. Direct use of source and destination IP addresses extremely reduces the flexibility of neural-net-based IDSs in detecting new or modified known attacks. Another reason for considering such a gist is due to the potential of IP spoofing in many new attacks specially Denial of Service (DoS) attacks. Direct use of source and destination IP addresses in misuse detection systems, produces more false negative alarms because of training the neural net based on special IP addresses as the source of the attacks. Accordingly, similar to many other practical IDSs (e.g. Snort, 2004); we determine whether the source/destination IP address belongs to the local network or external network in RT-UNNID. The network mask and subnet address of the local IP address are used for this purpose.

Table 1 – Features of the canonical format classified in four categories

Category	Features
IP header fields	diff-time-stamp, ip-id, ip-tos, ip-ttl, ip-headerlen, ip-len, is-home-src-ip, is-home-dest-ip, is-land, ip-frag-flag
TCP header fields	tcp-src-port, tcp-dest-port, tcp-fin, tcp-syn, tcp-rst, tcp-push, tcp-ack, tcp-urg, tcp-offset, tcp-win-size
UDP header fields	udp-src-port, udp-dest-port
ICMP header fields	icmp-type, icmp-code, icmp-id, icmp-sequence

```
0,3,10438,0,64,20,52,1,1,0,2,32803,139
,0,0,0,0,1,0,32,5840,0,0,0,0,0,0
```

Fig. 2 – The canonical format of a sample TCP packet.

The second significant practical issue we have considered in feature selection is the effect of time and time correlation of packets stream in intrusion detection. This feature was considered in many neural-net-based IDSs (Bivens et al., 2002; Labib and Vemuri, 2002; Lichodzijewski et al., 2002b; Zhang et al., 2001). There are two approaches to represent time in intrusion detection systems (Lichodzijewski et al., 2002b):

- *Explicit representation of time*; through assigning a time stamp to each packet to indicate its receiving time.
- *Implicit representation of time*; by the way of putting the received packets in an FIFO queue and feeding all the queued packets information simultaneously to a neural-net-based detection engine.

In RT-UNNID, we initially considered time explicitly as one of the packet IP header features, but faced up to no improvement on detection performance of the system. Consequently, we examined implicit representation of time using the conventional method of representing implicit time (i.e. using FIFO queue of packets), which imposed a great overhead on our system. So, we implemented the representation of time in a more efficient approach using the time difference between each packet and its previous packet in the incoming packet stream. Implicit representation of time in this manner had a significant effect on detection of many attacks especially denial of service attacks. In denial of service attacks, an attacker sends many packets in a short time to the victim system. Using this feature, the detection performance of our system was increased by more than 2%, which is very significant in marginal rates of detection.

6. Unsupervised neural-net-based engine (UNN-Engine)

The most important component of RT-UNNID is UNN-Engine whose function is to analyse and detect intrusions using unsupervised neural nets. To utilize such capability, some features and procedures which are necessary for training and testing the neural net, have been installed in the system. The most important factors which influence the efficiency of a neural net include three items: the features which are selected as the network input, network structure, and the values of the network parameters.

In Section 5, we explained the selected features of the network input, the characteristics of some of the features, and also the method of preprocessing the input data. Regarding the proper structure and plausible parameters value, we

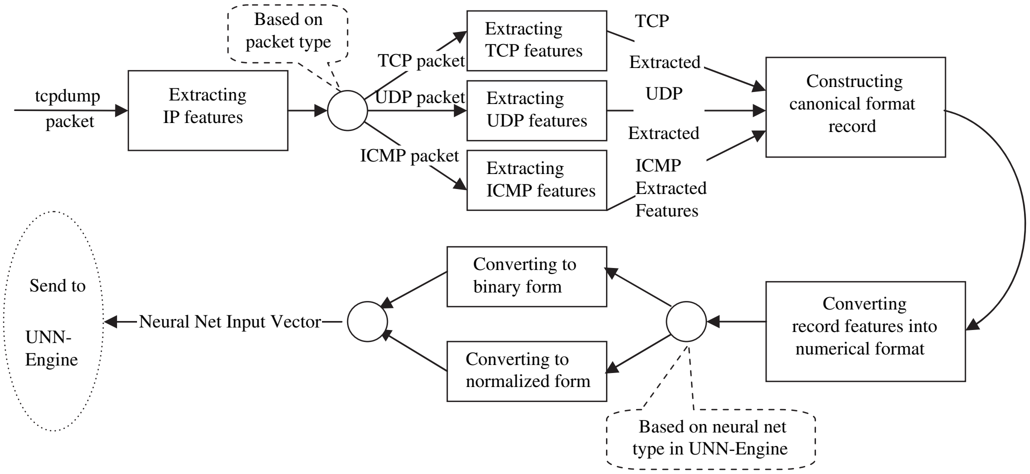

Fig. 3 – Feature extraction and data preprocessing in RT-UNNID.

used the common approach of trial and error by performing various experiments. To facilitate such task, the user interface "UNN-Engine UI" has been developed, which provides the chance of any change in structure and parameters of the unsupervised neural nets implemented in RT-UNNID.

The first step in applying RT-UNNID for intrusion detection is training the neural-net-based analyzer. Unsupervised neural nets are able to classify the input data according to their similarities. We used such a capability for classifying network traffic into normal traffic and abnormal or intrusive one.

RT-UNNID facilitates assessment of unsupervised neural nets, especially ART nets, in intrusion detection. Our focus in this paper is presenting a practical solution in application of unsupervised ART nets in intrusion detection and comparing them with SOM under the identical and real conditions. On this basis, three types of unsupervised neural nets namely SOM, ART-1, and ART-2 were implemented in the RT-UNNID detection engine.

As shown in Fig. 4, the common procedure used in training these three sorts of neural nets is as follows. Initially, the neural net automatically learns and classifies the input data based on their similarities. After finishing the clustering phase, the system determines the neurons of each cluster and assigns a name to each cluster using the label of packets (which exist in the 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 represents the winning or best matching for. Applying of the above procedure constructs a *Clustering Map*. In this map, units are clustered together to indicate either the normal traffic, known trained attacks, or possibly a new attack. New attacks may appear in the abnormal traffic, which is neither a normal traffic nor a known attack.

Considering the above training process, we can use both normal and known attacks traffic for training UNN-Engine and consequently for detecting both known attacks and the abnormal traffic as new attacks. In other words, we combined misuse detection and anomaly detection approaches together using unsupervised neural nets. This characteristic of RT-UNNID offers the advantages and abilities of both approaches in detection and recognition of known attacks as well as new ones.

6.1. *Self-Organizing Map (SOM) classifier in UNN-Engine*

According to experimental results, information in human brain is stored in a two-dimensional surface. Similar and related data reside in spaces close to each other on this surface. Based on this feature of human brain, self-organizing neural nets have been developed. Mathematical models of self-organizing neural networks were introduced by Malzberg and then they were developed by Kohenen in 1989 (Fausett, 1994; Sadati, 2002).

The self-organizing neural nets, also called topology-preserving maps, assume a topological structure among the cluster units. In this structure, there are m cluster units, arranged in a one- or two-dimensional array. Each input signal is an n-tuple vector. The weight vector for a cluster unit serves as an exemplar of the input patterns associated with that cluster. During the self-organization process, the cluster unit whose weight vector is more similar than others (its Euclidean distance is smaller) is chosen as the winner of this competition. The winning unit and its neighbouring units update their weights toward input signal patterns (Fausett, 1994).

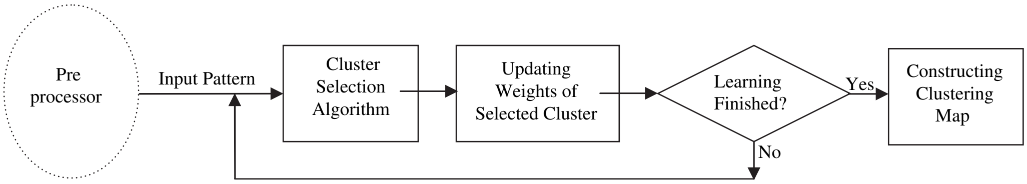
Pre processor
Input Pattern
Cluster Selection Algorithm
Updating Weights of Selected Cluster
Learning Finished?
Yes
No
Constructing Clustering Map

Fig. 4 – UNN-Engine training process in RT-UNNID.

In many unsupervised neural-net-based IDSs, SOM has been employed to classify network traffic into attack and normal. In the RT-UNNID system, this type of neural net has been used for investigating the capabilities of SOM networks in separating the attack traffic from normal one and also for comparing other types of unsupervised neural networks with it.

6.2. *Adaptive Resonance Theory (ART) classifier in UNN-Engine*

Adaptive Resonance Theory (ART) was invented by Stephen Grossberg in 1976. Later on, ART came in several flavours 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 Gaussian ARTMAP (Tauritz, 2003). Our focus in this paper is on using unsupervised ART nets in real-time intrusion detection systems.

In unsupervised ART nets, the input patterns may be presented several times and in any order. Each time a pattern is presented, an appropriate cluster unit is chosen, and related cluster weights are adjusted to let the cluster unit learn the pattern. 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 the most cases of clustering nets, the weights on a cluster unit may be considered as an exemplar (or code vector) for the patterns placed on that cluster (Fausett, 1994). ART nets are designed to allow the user to control the degree of similarity of patterns placed on the same cluster through tuning the *vigilance parameter*. The vigilance parameter can be used to determine the proper number of clusters in ART nets, in order to reduce the probability of merging different types of clusters to the same cluster (Fig. 5). Moreover, ART nets have two other main characteristics: *stability* that means a pattern does not oscillate among different cluster units at different stages of training, and *plasticity* that means ART nets are able to learn a new pattern equally well at any stage of learning (Fausett, 1994; Li, 1997).

Stability and plasticity of ART nets and the capability of clustering input patterns based on the user controlled similarity between them, made these nets more appropriate for IDSs, rather than most of the other types of unsupervised neural nets (such as SOM). Accordingly, we used two types of unsupervised ART nets, ART-1 and ART-2. ART-1 is aimed to cluster binary inputs and ART-2 is aimed to accept continuous-valued vectors (Fausett, 1994).

In the following sections, we discuss the practical comparison of the classifiers based on ART-1, and ART-2 with the SOM classifier as well as their capability in this application from various aspects.

7. Experimental results

We implemented RT-UNNID in a Red Hat Linux 9.0 operating system environment. Using the system, we evaluated the detection performance of different types of unsupervised neural nets in intrusion detection systems. We will describe the process of evaluating our system followed by the results.

7.1. *Train and test data sets*

Firstly, we tried to use the DARPA data sets for training and testing our system. However, due to the enormity of these data and their inaccessibility, we had to generate the necessary data within a local network. Using some of the existing attack tools, we generated a group of attacks against a local network server and collected the produced traffic as known attacks traffic. To gather the normal traffic, we recorded samples of the usual traffic of the network within a 4 days period. Thus, we had a training data collection of over 5000 packets including 20 known attack types and a test data collection, more than 3000 packets including the aforementioned 20 attack types plus 7 other attack types. The existing attacks in the test data set along with the corresponding tools for their generation are presented in Table 2. More information about the attacks and the tools is available in Attacks Tools and

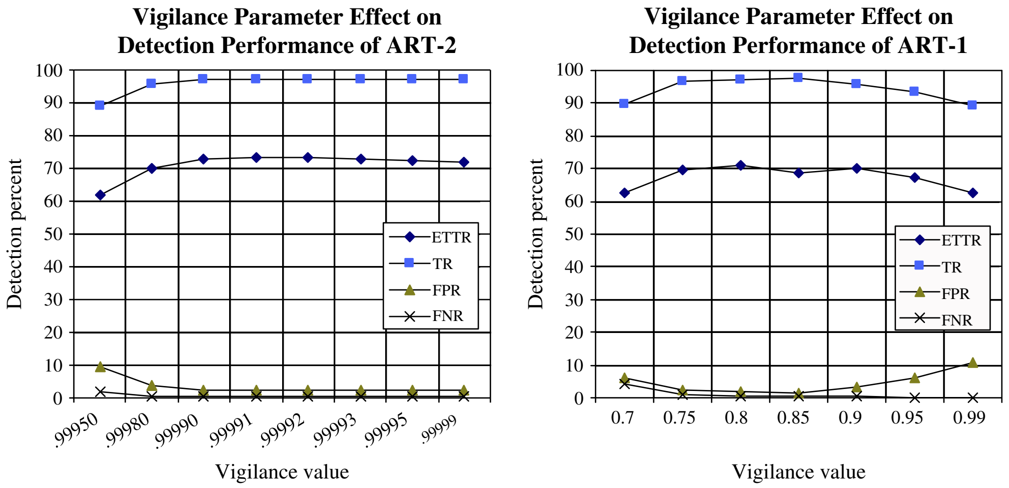

Fig. 5 – Effect of vigilance parameter value on detection performance of ART-1 and ART-2 nets.

Table 2 – Train and test data attack types with their tools

Attack name	Attack generation tools	Train data set	Test data set
Bonk	targa2.c	✓	✓
Jolt	targa2.c	✓	✓
Land	targa2.c	✓	✓
Saihyousen	targa2.c	✓	✓
TearDrop	targa2.c	✓	✓
Newtear	targa2.c	✓	✓
1234	targa2.c	✓	✓
Winnuke	targa2.c	✓	✓
Oshare	targa2.c	✓	✓
Nestea	targa2.c	✓	✓
SynDrop	targa2.c	✓	✓
Octopus	Octopus.c	✓	✓
KillWin	Killwin.c	✓	✓
Twinge	Twinge.c	✓	✓
TcpWindowScan	Nmap	✓	✓
SynScan	Nmap	✓	✓
Neptune	FireHack	✓	✓
Dosnuke(NetBios)	FireHack	✓	✓
Smbdie	Smbdie.exe	✓	✓
XmassTree-Scan	Nmap	✓	✓
LinuxICMP	linux-icmp.c	–	✓
Moyari13	moyari13.c	–	✓
Sesquipedalian	sesquipedalian.c	–	✓
Smurf	smurf4.c	–	✓
OverDrop	overdrop.c	–	✓
OpenTear	opentear.c	–	✓
EchoChargen	FireHack	–	✓

Information (2003), MIT Lincoln Laboratory – DARPA Intrusion Detection Evaluation Data Sets (2003).

7.2. *Evaluation criteria*

For evaluating the IDS outputs (in the test phase), an IDS Evaluator component was added to RT-UNNID. This component, by comparing the output of IDS and expected output of the system (which is determined for a test data set in a separate file), calculates the following evaluation metrics:

1. ETTR: exact true type detection rate (detecting normal traffic from attacks and recognising the known attack type);
2. TR: true detection rate (only separating normal traffic from attacks);
3. FPR: false positive detection rate (mis-detecting attacks);
4. FNR: false negative detection rate (failing to detect attacks when they are occurred).

7.3. *Neural net structure and parameters value*

Based on the above criteria, before evaluating the system, we determined the best values of important parameters for each neural net model. These include the number of cluster units in the output layer, the number of epochs for training, and the vigilance parameter in ART nets. For this purpose, some primary experiments were carried out and the values of Table 3 were achieved.

A noticeable point realized from these primary experiments was the considering influence of vigilance parameter on detection performance of the system. As depicted in Fig. 5, neither small nor big amount of this parameter is suitable for this purpose. This is due to the fact that the vigilance parameter determines the similarity degree of patterns that are placed on the same cluster. Hence, the low value of this parameter causes dissimilar patterns to be placed in the same cluster and so the neural net is unable to precisely distinguish some of the patterns from each other. The high value for the parameter also causes increasing the sensitivity of the network and reduction of its flexibility in placing a new pattern in the previously formed clusters (of normal and known attacks). As a result, by finding out a proper value of the vigilance parameter, it is possible to determine the optimum sensitivity level of the system and the appropriate number of produced clusters, during the training phase.

7.4. *Detection performance evaluation*

After determining appropriate structure and parameter values for SOM, ART-1, and ART-2, we evaluated their performance in real-time detection of network-based attacks using the RT-UNNID system. The evaluation results are shown in Table 4.

According to the results, ART-1 has the highest performance and SOM has the lowest performance. In our previous paper (Jalili and Amini, 2003), we investigated the capability of SOM and ART nets in intrusion detection based on the KDD-Cup's 99 standard collection of network connection records. Both research results show that ART nets have higher intrusion detection performance than SOM nets, using either offline connection based data or on-line packet based data.

Table 3 – Best achieved values for neural net parameters

SOM		ART-1		ART-2	
Cluster units number	2500	Cluster units number	2500	Cluster units number	400
Epochs	100	Epochs	100	Epochs	100
Learning rate	0.5	Vigilance	0.8	Vigilance	0.99991
Neighbourhood type	Rectangular	L	2.0	Learning Rate	0.5
Neighbourhood number	7			a, b	10
Distance	Euclidian			c	0.2
				d	0.8

Table 4 – Detection performance of SOM, ART-1, and ART-2 in RT-UNNID

	ETTR	TR	FPR	FNR
RT-UNNID ART-1	71.17	97.42	1.99	0.59
RT-UNNID ART-2	73.18	97.19	2.30	0.51
RT-UNNID SOM	83.44	95.74	3.50	0.77

7.5. *Training time and detection time evaluation*

One of the main challenges in using neural nets in intrusion detection is their training time problem. Previous research demonstrates that neural nets in IDSs need a long time for their training (Cannady, 1998; Coolen and Luiijf, 2002). On the other side, the response time of neural nets is low and they are very fast in the deployment phase. This is due to the low usage of system resources, in comparison to the other intelligent techniques such as expert systems (Cannady, 1998; Coolen and Luiijf, 2002; Tan, 1995). This makes neural nets, specially the unsupervised ones, as one of the most appropriate candidates to be used in real-time intrusion detection systems.

We measured the training time of 5000 packets in different iterations (in 100 iterations) and computed the average training time of each packet per iteration. The results are presented in Table 5 for the three different neural nets.

To measure the detection time of unsupervised neural nets, we ran RT-UNNID on a Pentium IV 1.8 GHz machine with 256 MB RAM connected to a local network. The network topology for this measurement is shown in Fig. 6.

Average detection time of categorising each packet to normal or attack was measured in these experiments. The results are shown in Table 6. This time includes processing and recognising of normality or abnormality of each packet, and logging intrusive ones.

The results show that ART-2 offers the highest speed in training and detection and so it is the best choice for real-time network-based IDSs engine, especially for high traffic networks. To verify this result, we saturated our local network using *netperf* benchmarking software (NetPerf, 2004) and generated attack traffic using one of the hosts in the LAN. Employing RT-UNNID in this situation showed that only ART-2 could process almost all the attack packets and recognise them. However, ART-1 and SOM can detect just a little and no considerable number of intrusive packets, because they are unable to process all the network packets.

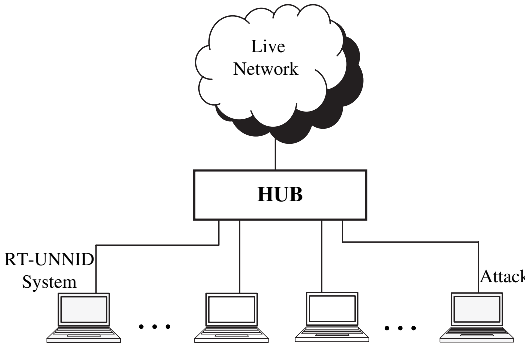

Fig. 6 – Network architecture for measuring detection time of RT-UNNID.

8. Conclusions

Using unsupervised neural nets in intrusion detection have many advantages rather than supervised ones. The main advantage is the capability of unsupervised nets to improve their analysis of new data without retraining. Moreover, unsupervised neural nets have higher speed and lower response time in the deployment phase.

In this paper, we presented a practical solution to using unsupervised neural nets in real-time intrusion detection systems, which are fed with live network packets. We designed and implemented a Real-Time Unsupervised Neural-Net-based Intrusion Detector system named RT-UNNID. The system is able to employ unsupervised neural nets for classifying and separating normal traffic from the intrusive and attack traffic. In its detection process, RT-UNNID uses combination of *misuse detection* and *anomaly detection* approaches. Therefore, this system is able to detect known attacks with their type as well as new unknown attacks.

RT-UNNID was used for tuning, training, and testing three types of unsupervised neural nets including SOM, ART-1, and ART-2 in intrusion detection. Evaluation of the above neural nets efficiency in intrusion detection was performed from three different aspects: detection performance, training time, and detection time. The results showed that SOM in 95.74%, ART-1 in 97.421%, and ART-2 in 97.19% of times were able to recognise attack traffic from the normal one. Although in our experiments, ART-2 offered a little lower detection performance than ART-1, its higher speed (both in training phase and application phase) makes it the most appropriate unsupervised neural net for applying in real-time intrusion detection systems.

Table 5 – Training time of RT-UNNID using SOM, ART-1, and ART-2

	Training time	
	(μs/Packet)	Relative time
RT-UNNID ART-1	8730	19.84
RT-UNNID ART-2	440	1.00
RT-UNNID SOM	17,580	39.95

Table 6 – Real-time detection time of RT-UNNID using SOM, ART-1, and ART-2

	Real-time detection time and performance		
	(μs/Packet)	Relative time	Performance (kB/s)
RT-UNNID ART-1	15,000	46.80	6.5
RT-UNNID ART-2	320	1.00	305.2
RT-UNNID SOM	18,500	57.81	5.3

REFERENCES

Amini M, Jalili R. Network-based intrusion detection using unsupervised adaptive resonance theory (ART). In: Proceedings of the fourth conference on engineering of intelligent systems (EIS 2004), Madeira, Portugal; 2004.

Attacks tools and information, <http://packetstormsecurity.nl/index.html>; 2003 [accessed November 2003].

Bishop M. Computer security, art and science. Addison-Wesley; 2003.

Bivens A, Embrechts M, Palagiri C, Smith R, Szymanski BK. Network based intrusion detection using neural networks. In: Intelligent engineering systems through artificial neural networks. Proceedings of ANNIE, vol. 12; 2002.

Bonifacio JM. Neural networks applied in intrusion detection systems. In: Proceedings of the IEEE world congress on computational intelligence and neural networks; 1998. p. 205–10.

Cannady J. Artificial neural networks for misuse detection. In: Proceedings of the national information systems security conference; 1998. p. 443–56.

Cannady J. Applying CMAC-based online learning to intrusion detection. In: Proceedings of the IEEE–INNS–ENNS international joint conference, vol. 5; 2000. p. 405–10.

Coolen R, Luiijf HAM. Intrusion detection: generics and state-of-the-art. Research and Technology Organization (RTO) Technical report 49; 2002.

Debar H, Dorizzi B. An application of recurrent network to an intrusion detection system. In: Proceedings of the international joint conference on neural networks; 1992. p. 478–83.

Debar H, Becker M, Siboni D. A neural network component for an intrusion detection system. In: Proceedings of the IEEE computer society symposium; 1992. p. 240–50.

Fausett L. Fundamentals of neural networks. Prentice-Hall; 1994.

Ghosh AK, Schwartzbard A. A study in using neural network for anomaly and misuse detection. In: Proceedings of the eighth USENIX security symposium; 1999.

Girardin L. An eye on network intruder–administrator shootouts. In: Proceedings of the first USENIX workshop on intrusion detection and network monitoring, Santa Clara, USA; 1999. p. 19–28.

Höglund AJ, Hätönen K, Sorvari AS. A computer host-based user anomaly detection system using self-organizing map. In: Proceedings of the IEEE–INNS–ENNS international joint conference on neural networks, vol. 5; 2000. p. 411–16.

Horeis T. Intrusion detection with neural network – combination of self-organizing maps and redial basis function networks for human expert integration, <http://ieee-cis.org/_files/EAC_Research_2003_Report_Horeis.pdf>; 2003.

Jalili R, Amini M. ART neural net based intrusion detection system (in comparison with self-organizing maps). In: Proceedings of the fifth conference on intelligent systems (CIS 2003), Mashhad, Iran; 2003.

Jirapummin C, Wattanapongsakorn N, Kanthamanon P. Hybrid neural networks for intrusion detection system. In: Proceedings of the international technical conference on circuits/systems, computers and communications (ITC–CSCC 2002), Thailand; July 2002. p. 928–31.

The third international knowledge discovery and data mining tools competition, <http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html>; 2003 [accessed April 2003].

Kevin LF, Rhonda RH, Jonathan HR. A neural network approach towards intrusion detection. In: Proceedings of the 13th national computer security conference; 1990. p. 125–34.

Labib K, Vemuri R. NSOM: a real-time network-based intrusion detection system using self-organizing maps. Technical report. Davis: Dept. of Applied Science, University of California; 2002.

Li T. Behavioral clustering and statistical intrusion detection. MS dissertation, Florida State University, Spring; 1997.

Lichodzijewski P, Zincir-Heywood AN, Heywood MI. Dynamic intrusion detection using self-organizing maps. In: Proceedings of the 14th annual Canadian information technology security symposium, CITSS, Ottawa, Canada; 2002a.

Lichodzijewski P, Zincir-Heywood AN, Heywood MI. Host-based intrusion detection using self-organizing maps. In: Proceedings of the IEEE world congress on computational intelligence; 2002b. p. 1714–9.

Lippmann RP, Cunningham RK. Improving intrusion detection performance using keyword selection and neural networks, RAID 99. Comput Netw 2000;34(4):597–603.

MIT Lincoln Laboratory – DARPA Intrusion Detection Evaluation Data Sets, <http://www.ll.mit.edu/IST/ideval/data/data_index.html>; 2003 [accessed June 2003].

Mukkamala S, Janoski G, Sung A. Intrusion detection using neural networks and support vector machines. In: Proceedings of the IEEE international joint conference on neural networks; 2002. p. 1702–7.

Mukkamala S, Sung AH, Abraham A, Ramos V. Intrusion detection systems using adaptive regression splines. In: Proceedings of the sixth international conference on enterprise information system (ICEIS-04). Kluwer Academic Publishers; 2004. p. 26–33.

NetPerf, <http://www.netperf.org>; 2004 [accessed November 2004].

Rhodes BC, Mahaffey JA, Cannady JD. Multiple self-organizing maps for intrusion detection. In: Proceedings of the 23rd national information systems security conference, Baltimore, MD; 2000.

Ryan J, Lin M, Miikkulainen R. Intrusion detection with neural networks. In: Advances in neural information processing systems, vol. 10. The MIT Press; 1998.

Sadati N. Artificial neural networks. Sharif University Press in Electrical Department; 2002.

Snort: the open source network intrusion detection system, <http://www.snort.org>; 2004 [accessed October 2004].

Tan K. The application of neural networks to UNIX computer security. In: Proceedings of the IEEE international conference on neural networks, vol. 7; 1995. p. 476–81.

Tauritz D. ART: an overview of the field, <http://web.umr.edu/~tauritzd/art/overview.html>; 2003 [accessed April 2003].

TcpDump, <http://www.tcpdump.org>; 2004 [accessed November 2004].

Yoo I, Ultes-Nitsche U. An intelligent firewall to detect novel attacks – an integrated approach based on anomaly detection against virus attacks. In: Proceedings of the SOFSEM conference, SOFSEM 2002 student research forum, Milovy, Czech Republic; 2002. p. 59–64.

Zhang Z, Li J, Manikopoulos CN, Jorgenson J, Ucles J. HIDE: a hierarchical network intrusion detection system using statistical preprocessing and neural network classification. In: Proceedings of the second annual IEEE systems, mans, cybernetics information assurance workshop, West Point, NY; June 2001.

Morteza Amini received his MSc in Computer Software Engineering from Sharif University of Technology, Tehran, Iran, in 2004. He is currently a PhD student in Software Engineering in Sharif University of Technology. His research interests are Information Security, Semantic Web Security, and Neural Networks.

Rasool Jalili received his PhD in Computer Science from The University of Sydney, Australia in 1995. He joined the Department of Computer Engineering, Sharif University of Technology, Tehran, Iran, as an assistant professor. His research interests are Distributed Systems and Information Security.

Hamid Reza Shahriari received his MSc in Computer Science from Amir-Kabir University of Technology, Tehran, Iran, in 2000. He is currently a PhD student in Computer Science in Sharif University of Technology, working on his thesis about vulnerability analysis of computer networks. His research interests are Information Security, and Formal Methods in Security.