A Novel Soft Computing Inference Engine Model for Intrusion Detection

Mahmoud Jazzar† and Aman Jantan††,

School of Computer Sciences, Universiti Sains Malaysia, Penang, 11800 Penang, Malaysia

Summary
The main purpose of this paper is to propose a novel soft computing inference engine model for intrusion detection. Our approach is anomaly based and utilizes causal knowledge inference based fuzzy cognitive maps (FCM) and multiple self organizing maps (SOM). A set of parallel neural network classifiers (SOM) are used to do an initial recognition of the network traffic flow to detect abnormal behavior. The FCM incorporate to eliminate ambiguities of odd neurons and making final decisions. Initially, each neuron is mapped to its best matching unit in the self organizing map and then updated by the fuzzy cognitive map framework. This updating is achieved through the weights of the neighboring neurons. Based on the domain knowledge of network data (network packets) the SOM/FCM combination presents quantitative and qualitative matching correspondences which in turn reduce the number of suspicious neurons i.e. reduce the number of false alerts. This method work as a unique fuzzy clustering approach and we demonstrate its performance using DARPA 1999 network traffic data set.

Key words:
Intrusion detection, False alerts, Self organizing maps, Fuzzy cognitive maps, Security

1. Introduction

Intrusion detection systems (IDS) technologies are considered as the last line of defense in computer network security infrastructure. However, the problem of detecting novel attacks and the increasing number of false alarms is one of the most dramatic drawbacks in most of network-based IDS sensors. Moreover, the lack of novelty detection and the false alert generation problem result in detection deficiencies and deterioration of the IDS basic function [1].

Usually, false alerts are generated from harmless events. This condition is motivated when the attacker has some prior knowledge about IDS sensor function thus deliberately craft a network data to trigger fake alerts. Ultimately, this will help attackers to over control and even overwhelm the function of the security sensor due to the large number of traffic that matches its rules or other triggering alert mechanisms [2]. Moreover, attackers may paralyze sensors from detecting real attack when they lunch a high volume of fake attacks such that massive DoS attack to defer sensors function [3].

According to [4], the high volume of false positive logs and alerts is a very time consuming task for network security analysts to extract true alerts and determine whether it is an attack when it is a benign. Thus, result in low data quality alerts during security alert production and analysis level which remarkably affect the analysis at higher stages. On the other hand, as network attack may not happen at single action such that one massive attack may be start by seemingly innocuous or by small probe actions to take palace [5]. Thus, tracing ongoing or existing attacks is a very important issue to be considered.

We believe that by strengthening the IDS main engine we can improve the detection accuracy and performance of the IDS system. The critical gap is obvious within the main processing unit of the IDS system rather than increasing the number of sensors in network state. Thus, we propose to enrich the IDS main engine with the supplement of an inference engine. This study is conducted to prove the hypothesis that the detection deficiency of IDS sensors can be improved by the supplement of a defense-in-depth strategy at sensor level to elevate higher level analysis operations.

In this paper, we focus on anomaly detection aspect. Therefore, we are modeling this part of detection as a supplementary adaptive inference engine. The core components of the inference engine combine the emergence of several computing techniques particularly unsupervised learning methods based on neural networks and fuzzy logic.

In this study, we propose the use of FCM and SOM to address the problem of detection deficiency in the IDS sensors discussed above. Our approach focuses on the adoption, integrity and information sharing among the IDS components. Figure 2 shows the inference engine general system overview. The figure also illustrates the existing IDS process elements flow and the gap where the proposed inference engine supplement is located.

Manuscript received April 5, 2008
Manuscript revised April 20, 2008

Initially, each neuron is mapped to its best matching unit in the self organizing map and then updated through the weights of the neighboring neurons. Later, the weights of odd neurons are considered based on its relevance to the clusters and/or to the relevant error caused by odd neurons. Based on the effect value of odd neurons, benign concepts which are not relevant to attacks or certain error caused are dropped. The approach highlights fuzzy clustered neurons (in SOM) in order to build a network of concepts where matching constraints are mapped.

The rest of the paper is organized as follows. Section 2 provides a background of related work. In Section 3 we present the general inference engine model. The inference engine architecture and design are available in section 4 and 5. In section 6 and 7 we present experimental results and discussion. Finally, in section 8 we provide conclusion and future work.

2. Related Works

Among the vast variety of techniques which have been researched for the IDS sensors, the interest on AI techniques and data mining applications have received greater attention particularly the use of unsupervised leaning methods as they have the ability to address some of the short comings [6]. This is also helps to achieve the ultimate goal for the IDS i.e. the capability of novelty detection. Recently, the unsupervised learning method (SOM) has represented an excellent performance for sensors work on an unsupervised learning mode [7], as well as it is efficient for real-time intrusion detection [8]. However, in order to refine the process and achieve better detection and performance, extra efforts are required.

Our work was motivated by the work done recently on SOM ensembles [9] and hierarchal SOM [5,7] for intrusion detection. Ensemble SOM are able to identify computer attacks and characterize them appropriately with levels of confidence where as Hieratical SOM provide an amazing detection rate and false positive rate under test conditions. The SOM method is attractive because of considering the properties of events and its capability of processing large amount of data with low computational overhead i.e. suitable for real-time intrusion detection [8]. However, our work is different from these approaches in tackling the internal properties of SOM i.e. retesting the properties that are out of norm internally using the FCM framework.

False positive alerts have been addressed by various studies at sensor level [10,11,12] by improving the sensor outputs. These studies whether are too general or concentrate on certain product improvement. On the other hand, false alerts have been tackled at higher levels of the IDS operations. One such prototype is the Toolkit for Intrusion Alert Analysis [13], and the Intrusion Alert quality Framework [4] that uses certain quality parameters to improve the false positives by 35.04% using DARPA 2000 data set. The various techniques used include data mining [14], AI techniques [15], fuzzy logic [16], neural networks [17] and neuro-fuzzy approach [1]. These techniques and approaches work on logs/alerts directly and indirectly by building new strategies to tackle intrusions of various types to improve the detection process.

The potentials of unsupervised learning techniques in anomaly detection can be demonstrated through the use of the SOM and FCM as the basis for anomaly detection. Thus, it is important to exhibit how these methods can support the current IDS specifically by building a purely data driven inference engine able to provide timely and accurately details and notifications of activities going on the system network.

The biggest challenge here is to develop an intelligent inference engine model to defense-in depth i.e. able to deal with uncertainty and detect novel attacks with low rate of false alerts. Moreover, any optimal solution of an adaptive IDS system should provide the means of real-time detection and response as well as high level trust among the IDS components.

3. The Inference Engine Model

Basically, the inference engine is a computer program that attempt to infer or derive a deep insights or answers from the knowledge base. Moreover, the inference engine is considered as the brain of the expert systems which has the ability of reasoning depending on the methods applied or used [18]. Typically, expert systems analyze the knowledge base in the (brain) the inference engine which is designated to simulate human like expertise and reason for problem solving in a certain domain. As such, expert system could explain the reasoning process and handle levels of confidence and uncertainty, which straight algorithms do not do [29]. Figure 1 illustrates the basic components of a conventional expert system.

Expert and knowledge based systems are widely used in different computer and network security applications such as intrusion detection. However, in this study we are modeling our proposed soft computing components as an adaptive inference engine model for the support of intrusion detection process.

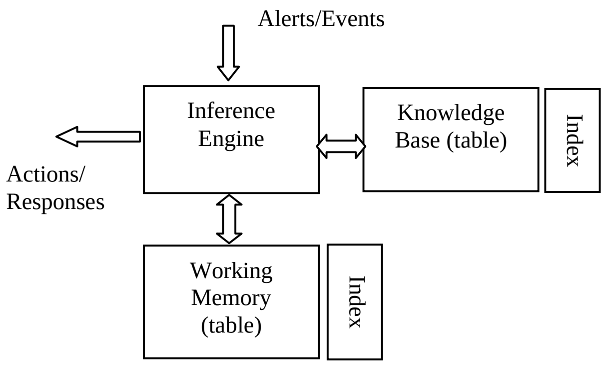

Fig. 1: Inference Engine Block Diagram [28].

The combined SOM-FCM model encompasses the inference engine components including the technical soft computing environments which provide the features and support required. It provides the core mathematical and graphical tools for classification, clustering, analysis, visualization and application development. However, the details about our inference engine internal design and architecture are illustrated in the following sections.

Soft computing paradigms such as neural networks, fuzzy inference systems and neuro-fuzzy methods are used for intrusion detection in many ways. However, in this study we focus on applying methods that combine the functionality and unsupervised learning ability from different soft computing paradigms. Therefore, we are modeling this part as a novel inference engine model of intrusion detection. There are different methods for modeling neuro-fuzzy systems based on structure, functionality and the degree of connectivity. One of the most popular neuro-fuzzy modeling methods is ANFIS and CANFIS described in details in Jang and Jang et al. [19,30].

The general design and architecture of our proposed inference engine combines advantages of the general design and architecture of both of modular hybrid system (MHS) and coactive neuro-fuzzy inference system (CANFIS). MHS provide a parallel neural network and fuzzy rule design. CANFIS provide the advantage of being linguistically interpretable fuzzy inference system that allows prior knowledge to be embedded in its construction and allows the possibility of understanding the results of learning. The Core components of our inference engine consists of a set of parallel soft computing classifiers as set of SOM and FCM are chosen to represent the parallel architecture of the inference engine design as shown in figure 3.

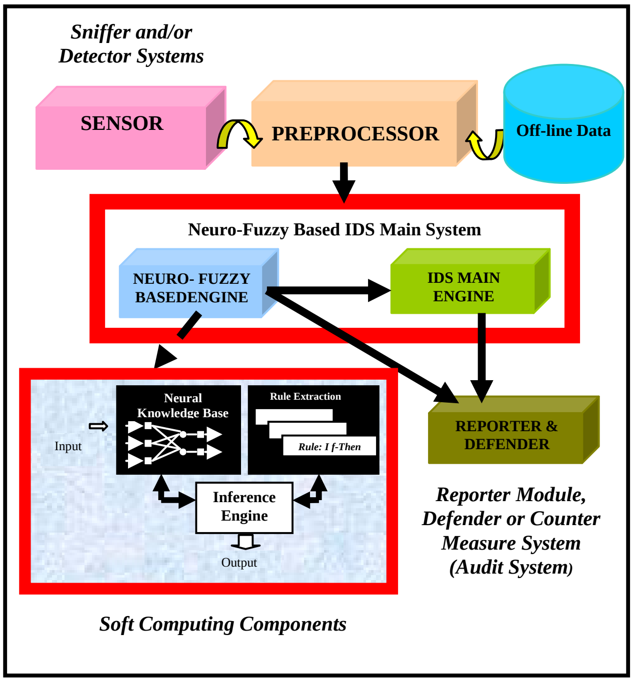

Fig. 2: General System Overview

4. The Inference Engine Architecture

The ability of detecting/preventing new attacks without prior knowledge of the attack behavior is a tough task, especially the way of determining the input features to monitor for normal versus intrusive behavior. To do so, we choose unsupervised learning techniques as they are the best suited for such situation [20].

To build the inference engine we use unsupervised learning method so called kohonen's maps (SOM) [31] for clustering and recognition of input data. Multi layer soft computing classifiers (SOM) work together to recognize abnormal behavior as shown in figure 3. FCM use causal reason to assess the SOM output and model the final decision.

Our proposed inference engine architecture is illustrated in figure 3 below. There are five parallel SOM layers trained to cluster the input data for each connection type. Each layer SOM belongs to one of five classes of the dataset and each provide an output relevant to specific class. The second layer is the FCM framework. This layer uses a causal reason to measure the severity of odd neurons (SOM alerts). The main advantage here is to call attention to how domain knowledge of neurons (network packets) can contribute on tracing new attacks or find path of on-going or existing attack.

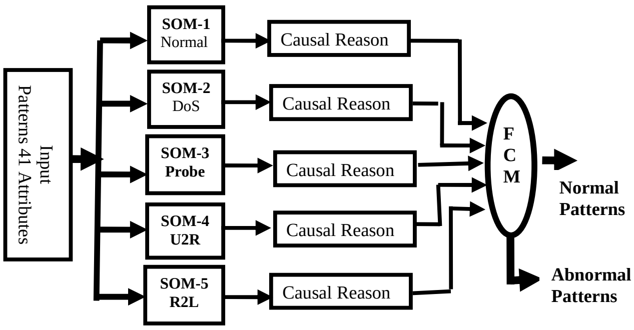

Fig. 3: The Inference Engine Architecture

5. The Inference Engine Design

Neurons can be organized in any topological manner. In the case of SOM, neurons usually are located on regular one or two dimensional topology. The Kohonen's Maps so called self organizing maps (SOM) [31] is a competitive and cooperative learning neural network. Thus, SOM retains the quality of a competitive and cooperative learning network to learn from a data set without supervision.

SOM can reduce dimensions by producing a map of usually one or two dimensions which plot the similarity of the data by grouping similar data items together. A wining neuron is one of neurons such that very similar or close to the neighborhood data in which later on can be classified as or belong to clusters. In this way the SOM can provide specialized platform for data representation from the input space. Figure 4 show some examples.

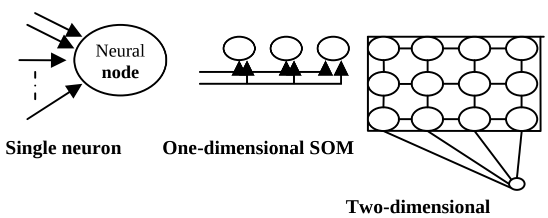

Fig. 4: SOM Structure Example

A typical SOM structure is usually an array of neurons arranged in rectangular or hexagonal method. Each of the neurons on the map is represented by a set of weights $w = [w_1..w_n]$ where n is equal to the dimension of the input vector. Two stages are required in order to create the SOM are initialization and training of the SOM. The initialization process sets up the map with the desired dimensions and initial weights for each unit of the map. The training process allows the map to adapt to the features of the data set during a number of epochs. At each epoch one input vector x is compared to all neurons weights w with a distance function (Euclidean or Manhattan) to identify the most similar nodes so called the best matching unit (BMU). Once the BMU has been found, the neighboring neurons and the BMU itself are updated according to the following rule:

$$w_i(t+1) = w_i(t) + h_{ci}(t)[x(t) - w_i(t)] \quad \ldots\ldots\ldots\ldots (1)$$

Where t is an integer that denotes time, $h_{ci}(t)$ is the neighborhood function around the winner unit c and $x(t)$ is the input vector drawn at time t. By updating the BMU and other units in the neighborhood, the distance between the BMU and the neighbors are brought closer together. The neighborhood function consists of two parts, one that define the form of the neighborhood and the other is the learning rate.

To increase the correlation among the neurons in the produced map grid, we minimize the neighborhood function and the learning rate by considering the minimum time interval according to the following rule:

$$h_{ci}(t) = h(\|r_c - r_i\|, t) \propto (t) \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (2)$$

Where r_c the location of winner unit, r_i is the location of the unit i on the grid map and $\propto (t)$ is the learning rate factor over minimum time t interval. Later, at this stage the map converge to an inactive stage which approximates the probability density function of the high dimensional input data. The learning rate and the neighborhood proceed by time until convergence.

The problem arose with neighboring neurons which are out of clusters and didn't reflect exactly the severity of attack-ness in network connections. That is because a network attack may not happen at a single action such that one massive attack may be start by seemingly innocuous or by a small probe actions to take place [5]. In SOM classification process per example in [9] a genetic or clustering algorithm used at the certain attack zone to classify each attack by class were as suspicious neurons which near attack zone or out of clusters are not analyzed and remain suspicious were they might be benign. As one potential solution to this problem in the hierarchal SOM [7], they consider the potential of studying the domain knowledge of features to be applied to the whole SOM concepts.

Here, we suggest an improvement to this process by considering the domain knowledge of particular neurons (odd neurons). Thus, we use the FCM to calculate the severity/relevance of odd concepts (neurons) to attacks. Therefore, benign concepts can be dropped or/and others can be addressed as a potential risk of error caused out of the cluster.

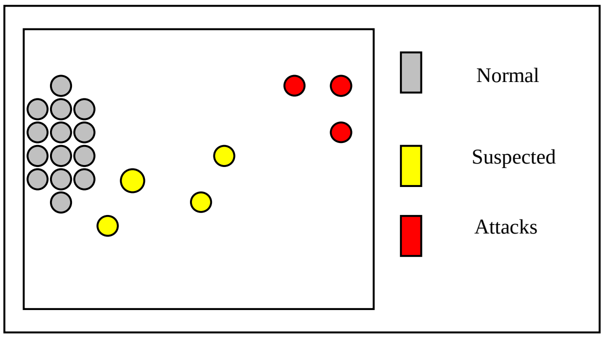

Fig. 5: SOM Connection Features

Each neuron is a network packet or connection which has certain identities either qualitative or quantitative such that (IP, Port, Flag, time, data, etc.). These identities could be related or not related to certain attack connection. We consider these identities as concepts at the FCM. The FCM module calculates the errors caused out of these concepts and the degree of relevance to certain or measure error (attack). Thus, later we can estimate how much these concepts related to attacks. The main advantage here is to call attention to how domain knowledge of neurons (network packets) can contribute on tracing new attacks or find path of on-going or existing attack.

We pick neurons with less relation to clusters and test its correlation to clusters. In other words, each feature parameter of odd neurons is measured based on a comparison criteria to detect interrelation between neurons i.e. determine the attack detection. To calculate the abnormality factor per packet we need to estimate the effect value of each feature parameter. The total degree of abnormality of odd neurons is calculated according to the following factors.

Table 1: Factors, rules and effect value

Factor	Factor Rule	Effect (E)
Availability	$\begin{cases} 1 & X \in S \\ 0 & X \notin S \end{cases}$ Where X : Comparison S: set of features	0.1
Similarity	$\begin{cases} 1 & X = S_i \\ 0 & X \neq S_i \end{cases}$ Where s_i is a feature of set S and X is comparison	0.1
Occurrences	$\log_2 \frac{1}{p(x)}$ Where p(x) is x's probability	0.2
Relevancy	$\frac{MaxF_i(x)}{\sum_{x \in s} F_i(x)}$ Where: x : Comparison, $MaxF_i(x)$ is the maximum frequency of occurrences and $\sum_{x \in s} F_i(x)$ is the total sample size (number of trials)	0.2
Independency	$P(x)P(y)$ Where p(x) is the x's probability and p(y) is the y's probability	0.2
Correlation	$Cov(X_t Y_t)/S_{X_t} S_{y_t}$ Is the covariance of X and Y comparisons at time t and the standard deviation	0.2

Now we can estimate the total degree of abnormality per packet according the following rule:

$$Un(x) = \sum_{i=1}^{n} E_i \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad (3)$$

Where:

$Un(x)$: Abnormality per packet

E_i : Effect value of packet

n : Total feature number of abnormality

Once the abnormalities per unclustered packets are calculated, the low malicious packets are dropped or ignored and the rest are considered as concepts in the FCM. It now is important to measure the effect/ influence value among the suspicious concepts to determine the path of the existing or ongoing attack. If the effect value is zero then there is no relationship among these concepts. Table2 show the total degree of effect value and relations between neurons.

Table 2: Effect and relation value trace

Normal	0
Slight	0.2
Low	0.4
Somehow	0.6
Much	0.8
High	1

5.1 FCM Procedure

FCM are a soft computing modeling techniques generated from the combination of fuzzy logic and neural networks [21,22,32,33]. FCM consist of nodes (concepts) and causal relations between the nodes formed in a structured collection (graph). The structure can be presented as an associative single layer neural networks which work on unsupervised mode whose neurons are assigned to concepts meanings and the interconnection weights represent the relationship among these concepts.

According to [23] in the FCM model, the directional influences are presented as all-or-none relationships i.e. FCM provide qualitative as oppose to quantitative information about relationships. In this work, the task of FCM is to determine the casual relation between the suspicious or odd neurons noted by the SOM to quantify the causal inference process. By quantifying the causal inference process we can determine the attack detection and the severity of odd neurons as such neurons with low causal relations can be dropped i.e. reduce the false alerts. The following steps are the general FCM procedure:

Inputs: SOM alerts (anomalies, false positives, false negatives, un-clustered packets, odd neurons …etc.)
Outputs: Reduced Alerts

Method:
1. Define the number of odd neurons or concepts which are produced after the SOM clustering (SOM Alerts)
2. Calculate the abnormality per neuron [Un(x)]
   2.1 Assign effect and relational values
   2.2 Drop neuron if the abnormality is low
3. Call FCM Initialization
5. Call FCM Simulation

Fig. 6: FCM Procedure

The number of neurons includes all those unidentified and attack neurons (SOM Alerts). At every epoch we process 100 neurons from the data set, later we pick the alert neurons and calculates the abnormality per each and every neuron to drop the low attack related and consider the rest as concepts for the FCM framework.

FCM Initialization

Initializing the FCM includes the definition of the FCM concepts and building the relations among these concepts by building a global matrix which can be calculated according to [21,32]. However, in order to build that matrix we define the weight of odd neurons according to the total effect factor $Un(x)$ and the grade of causality W_{ij} between the nodes C_i and C_j according to the following assumptions:

1. If $C_i \neq C_j$ and $E_{ij} \rangle 0$ then $W_{ij}^{+} = \max\{E_{ij}^{t}\}$
2. If $C_i \neq C_j$ and $E_{ij} \langle 0$ then $W_{ij}^{-} = \max\{E_{ij}^{t}\}$
3. If $C_i \neq C_j$ then $E_{ij} = 0$ and W_{ij} is zero

FCM Simulation

Once the FCM constructed its important now to measure the overall simulation of the system which consists of s input states such that M= {s1, s2 …s} where $s_i \in [0,1]$. After n number of iterations the output is $\overline{M}$ i.e. the predictions of the FCM model. The simulation of FCM follows the following steps:

1. Read from input state
2. Calculate the Effect factors
   2.1 Drop low effect factors
3. Until the system convergence
   3.1 Show the link of related factors

Fig. 7: FCM Simulation Steps

6. Experiment and Evaluation Description

SOM-FCM model is a defense-in-depth network based intrusion detection scheme. The model utilizes the domain knowledge of network data to analyze the packet information. Based on the analysis given, benign packets are dropped and high risk packets can be highlighted or blocked using a causal knowledge reason in FCM. The flowchart of the detection module is illustrated in figure 8.

6.1 Data Collection and Preprocessing

In this experiment, we use the most popular IDS evaluation data in which most of researchers aware of and use for evaluating their research, the KDD Cup 1999 intrusion detection contest data [24] followed by the success of the 1998 DARPA Intrusion Detection Evaluation program by MIT Lincoln Labs (MIT Lincoln Laboratory) [25].

The aim of DARPA evaluation was to assess the current state of Intrusion Detection Systems at the Department of Defense at the U.S. by simulating a typical U.S. Force LAN. However, Lincoln Labs acquired 9 weeks of raw data collection for the evaluation. The collected raw data processed into connection records, about 5 million of record connection. The data set contain 41 attributes for each connection record plus one class label and 24 attack types which fall into four main attack categories [26] as follows:

1. Probing: surveillance attack categories
2. DoS: denial of service
3. R2L: unauthorized access from a remote machine
4. U2R: unauthorized access to local super user (root) privileges

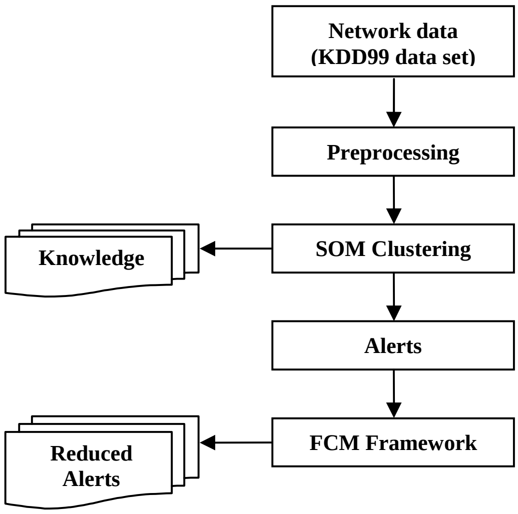

Fig. 8: Flowchart of the detection module

The data set was established to evaluate the false alarm rate and the detection rate using the available set of known and unknown attacks embedded in the data set [27]. We select a subset for testing our experiment. The selected subset contain 2020 records with non zero values as shown in table 3 below because some attacks are represented with few examples and the attack distribution in the large data set is unbalanced. However, collection, preprocessing and calculation of false and true alert of test data are followed as in [5]. We implement and run our experiment on a system with 2.667GHz Pentium4 processor 506 and 256MB PC3200 DDR400 RAM running widows XP.

Table 3: Selected data set records

Attack Category	Attack Name	# Records	Total
Normal		1020	1020
DoS	Neptune	105	367
DoS	Smurf	124	
DoS	Back	42	
DoS	Land	40	
DoS	Pod	33	
DoS	Teardrop	23	
Probe	Ipsweep	79	319
Probe	Nmap	59	
Probe	Portsweep	77	
Probe	Satan	44	
Probe	Mscan	36	
Probe	Saint	24	
U2R	buffer_overflow	82	217
U2R	sqlattack	79	
U2R	Perl	8	
U2R	Xterm	22	
U2R	Rootkit	26	
R2L	guess_passwd	41	97
R2L	Imap	2	
R2L	ftp_write	22	
R2L	Phf	20	
R2L	Sendmail	12	

6.2 SOM Clustering

In SOM clustering module the selected data set clustered based on the BMU and neighborhood accommodation as explained in section 5. SOM have presented a homogeneous clusters which represent normal data and odd neurons which represent alerts of certain attack type or suspicious nodes.

6.3 Alert Collection

Alert collection part is the SOM output part of odd neurons. Each alert contain detail information about the alert type, data and time happened. The collected alerts attributes are then transformed as an input for the FCM framework. Alerts attributes such as alert ID, date, time, srcIP, srcPort, dstIP, dstPort…

6.4 FCM Framework

In this module the received alerts attributes will be carried for fine-tuning in the FCM framework as discussed in section 5 and 5.1. In this module, neurons which represent low effect or less correlated to other attack like neurons are dropped or ignored and the high suspicious nodes are highlighted.

7. Discussion

In our experiment, the performance measure of both SOM and combined SOM-FCM are carried out solely on the selected data subset from the corrected.gz file of the KDD'99 data set [24] which contains test data with corrected labels. For instance, we calculate the detection rate and the false alarm rate according to [5] the following assumptions:

FP: the total number of normal records that are classified as anomalous
FN: the total number of anomalous records that are classified as normal
TN: the total number of normal records
TA: the total number of attack records
Detection Rate = [(TA-FN) / TA]*100
False Alarm Rate = [FP/TN]*100

The primarily results show that it's possible to reduce false alerts in SOM-based intrusion detection sensors using FCM causal reason. We believe that, further improvement on the SOM structure with FCM will improve the detection accuracy and expose more information about the attack details. Table 4 and table 5 shows the experimental results obtained.

Table 4: Experimental results

Attack Type	# Records	# Detection Records	
		SOM	SOM-FCM
Normal	1020	1018	1015
Probe	319	276	282
DoS	367	352	361
U2R	217	183	183
R2L	97	72	69
Overall	2020	1901	1910

Table 5: False alarm comparison

Method	Detection Rate	False Alarm Rate
SOM	88.30%	11.66%
SOM-FCM	90%	10.29%

8. Conclusion

In this paper, we describe the general design and architecture of our proposed inference engine. Our approach uses a set of parallel soft computing based classifiers (SOM and FCM) for detecting abnormal behaviors of network data. We also describe the possibility of establishing link between SOM and FCM and using the combination for building better IDS sensor. The immediate result of this research is to improve the detection deficiency issue in the SOM-based IDS sensor by reducing the false alerts and increasing the detection accuracy at the sensor level. For future work, experiment should be done on real time traffic data and investigating methods for proper feature selection and presentation.

References

[1] R. Alshammari, S. Sonamthiang, M. Teimour, D. Riordan, "Using neuro-fuzzy approach to reduce false positive alerts", Fifth Annual Conference on Communication Networks and Services Reseach (CNSR'07), IEEE Computer Society Press, 2007. pp. 345-349.
[2] W. Yurcik, "Controlling intrusion detection systems by generating false positives: Squealing proof-of-concept", 27th Annual IEEE Conference on Local Computer Networks, 2002.
[3] T.S. Sobh, "Wired and wireless intrusion detection system: Classification, good characteristics and state-of-the-art", Computer Standards & Interfaces 28(2006) 670-694.
[4] N.A. Bakar, B. Belaton and A. Samsudin, "False Positive Reduction via Intrusion Alert Quality Framework", 13th IEEE international Conference on Networks, Kuala Lampur, Malaysia, Vol. 1, 2005, pp. 547-552.
[5] S.T. Sarasamma, Q.A. Zhu, and J. Huff, "Hierarchal Kohonenen Net for Anomaly Detection in Network Security", IEEE Transactions on Systems, Man, and Cybernetics-Part B: Cybernetics, 35(2), 2005, pp. 302-312.
[6] M. Amini, R. Jalili, and H.R. Shahriari, "RT-UNNID: A practical solution to real-time network-based intrusion detection using unsupervised neural networks", Computers & Security, 25(2006) 459-468.
[7] H. Gunes Kayacik, A.N. Zincir-Heywood, and M.I. Heywood, "A hierarchal SOM-based intrusion detection system", Engineering Applications of Artificial Intelligence, 2006, doi: 10.1016/j.engappai.2006.09.005
[8] W. Wang, X. Guan, X. Zhang, and L. Yang, "Profiling program behavior for anomaly intrusion detection based on the transition and frequency property of computer audit data", Computers & Security, 25(2006) 539-550.
[9] L. DeLooze, "Attack Characterization and Intrusion Detection using an Ensemble of Self-Organizing Maps", Proceeding of the 2006 IEEE workshop on Information Assurance, United States Military Academy, West Point, NY, 2006.
[10] K. Timm, "Strategies to Reduce False Positives and False Negatives in NIDS", SecurityFocus Article, 2001 www.securityfocus.com/infocus/1463 (10 Dec. 2007).
[11] M.J. Ranum, "False Positives: A User's Guide to Making Sense of IDS Alarms", ICSA Labs IDSC, 2003.
[12] M. Norton and D. Roelker, "Snort 2.0 Rule Optimizer", Sourcefire Network Security White Paper, April 2004.
[13] "TIAA: A Toolkit for Intrusion Alert Analysis (Version 0.4)", http:/discovery.csc.ncsu.edu/software/correlator/ver0.4
[14] W. Lee, S.J. Stolfo, and K.W. Mok, "Adaptive intrusion detection: A data mining approach", Artificial Intelligence Review 14(6), 2000, pp. 533- 567.

[15] A. Siraj, R.B. Vaughn, and S.M. Bridges, "Intrusion Sensor Data Fusion in an intelligent Intrusion Detection System Architecture", Proceeding of the 37th Hawaii International Conference on System Sciences, 2004.
[16] J.E. Dickerson, J. Juslin, O. Koukousoula, and J.A. Dickerson, "Fuzzy intrusion detection", IFSA World Congress and 20th North American Fuzzy Information Processing Society (NAFIPS) International Conference, Vancouver, British Columbia, 2001.
[17] Y. Liu, D. Tian, and A. Wang, "ANNIDS: Intrusion Detection System Based on Artificial Neural Network", Proceedings of the second international conference on machine learning and cybernetics, Xi'an, 2003.
[18] N.L. Griffin and F.D. Lewis, "A rule-based inference engine which is optimal and VLSI implementable", IEEE Int'l Workshop on Tools for AI Architectures, Languages and Algorithms, Oct. 23-25, 1998, pp. 246-251.
[19] J.-S.R. Jang, "ANFIS: Adaptive network-based fuzzy inference systems", IEEE Transactions on Systems, Man, and Cybernetics 23(3) 1993 665-685, doi:10.1109/21.256541.
[20] O. Depren, M. Topallar, E. Anarim, and M.K. Ciliz, "An intelligent intrusion detection system (IDS) for anomaly and misuse detection in computer networks", Expert Systems with Applications, 29(2005), pp. 713-722.
[21] B. Kosko, "Fuzzy Cognitive Maps", International Journal of Man-Machine Studies, Vol. 24, 1986, pp. 65-75.
[22] C.D. Stylios and P.P. Groumpos, "Mathematical Formulation of Fuzzy Cognitive Maps", Proceedings of the 7th Mediterranean Conference on Control and Automation (MED99), Haifa, Israel, 1999.
[23] J. Aguilar, "A Dynamic Fuzzy-Cognitive-Map Approach Based on Random Neural Networks", International Journal of Computational Cognition, Vol. 1, number 4, 2003, pp. 91-107.
[24] KDD Cup 1999 Data. Knowledge Discovery in Databases DARPA Archive. http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html. Accessed December 2007.
[25] MIT Lincoln Lab, DARPA Intrusion Detection Evaluation Plan. http://www.ll.mit.edu/IST/ideval/data/2000/2000_data_index.html. Accessed December 2007.
[26] S. Peddabachigari, A. Abraham, C. Grosan, and J. Thomas, "Modeling Intrusion Detection System Using Hybrid Intelligent Systems", Journal of Network and Computer Applications, 2005.
[27] K. Kendall, "A Database of Computer Attacks for the Evaluation of Intrusion Detection Systems". Master's thesis, Massachusetts Institute of Technology, Department of Electrical Engineering and Computer Science, Cambridge, MA, 1999.
[28] P. Kabiri and A. Ghorbani, "A rule-based temporal alert correlation system", International Journal of Network Security, Vol.5, No.1, pp. 66-72, July 2007.
[29] J. Giarratano and G. Riley, Expert systems: principles and programming, PWS-Kent Publishing Co, Boston, MA, 1989.
[30] J.-S.R. Jang, C.-T Sun, and E. Mizutani, Neuro-fuzzy and soft computing: A computational approach to learning and machine intelligence. MATLAB Curriculum Series. Prentice-Hull, Upper Saddle River, NJ, 1997.
[31] T. Kohonen, Self-Organizing Maps, Third ed. Springer, Berlin, 2000.
[32] B. Kosko, Fuzzy Engineering, Prentice-Hall, New Jersey, 1997.
[33] R. Axelrod, Structure of Decision: The Cognitive Maps of the Political Elites, Princeton University Press, New Jersey, 1976.

Mahmoud Jazzar received the B.Sc. (Hons.) degree in computer science and application from Aligarh Muslim University, India, in 1998. He obtained his MSc. degree in computer science from the University of Mysore, India, in 2002. During 2002-2004, he stayed at Al-Quds Open University-West Bank as an instructor of computer science and ICT as well as part time lecturer at the Arab American University-Jenin. Later, he worked as a lecturer of computer science and network at Curtin University of Technology, Sarawak Campus until Dec 2005. Currently, he is a PhD student with Universiti Sains Malaysia. His research interests include computer and network security, intelligent systems, knowledge discovery and data mining.

Aman Jantan is a senior lecturer at the school of computer sciences, Universiti Sains Malaysia. He received the BComp.Sc (Hons.) and M.Sc. degree in computer science from the Universiti Sains Malaysia in 1993 and 1996 respectively. He obtained his PhD in computer science in 2002 from the same university. His research interests are in the fields of AI, computer and network security, E-commerce/web intelligence, compilers design and development techniques.