**NeID: Neural Network based Intrusion Detection System**

Rajkumar Pandi

Department of Computer Science, University of New Mexico

[rpandi@unm.edu](mailto:rpandi@unm.edu)

**Abstract**

This paper compares various methods that are used to learn the normal behavior of the program and finds the best fit to detect intrusions. Out of all methods so far, its been proven that neural networks show better performance but learning the sequences takes a longer amount of time. In order to overcome this, a legitimate intrusion detection system that uses evolutionary neural networks(ENN) is deployed. The ENN differs from conventional neural networks as they can discover the structure and weight of the neural network simultaneously and it takes less time for learning.

**1 Introduction**

With the rapid growth of the Internet and its potential, changes have been observed in the business model of many organizations . With the Internet, there are two types of users, the users who are harmless and the users who are harmful, eventually, when an organization makes its information available to everyone, it might be open to harmful users(attackers)as well. There are various ways an attacker can gain access to organization's internal system, some of the general ways are 1) exploiting software bugs (vulnerabilities) 2) setup or compliance problems 3) leaving default settings. This paper is only going to focus on host based anomaly detection.

As more complicated structures in an organization are being controlled by computer programs, there is a risk of new vulnerabilities. Many companies already started voting with their feet, as they are paying millions and still end up with insecurities. Current intrusion detection systems have laid an egg at newly emerging software bugs. Recently, Google's project zero team uncovered a zero day exploit in a Adobe kernel module and no anti viruses or any other intrusion detection softwares were able to detect it. Generally intrusion detection system are classified into two kinds 1)Anomaly detection and, 2)Misuse detection.

Anomaly detection first records the normal behavior of a program and measure deviation from the normal behavior. Basically, Anomaly detection presumes that intrusion are mutually related to the abnormal behavior displayed by either the user or the system.

Misuse detectionis also called as signature based detection. Only the intrusion whose signatures are available can be detected. The approaches include expert systems, model-based reasoning, state transition analysis and key stroke dynamics monitoring.

Misuse detection are static, it depends on the predefined rules that makes it hard to detect new attacks. On the other hand, Anomaly detection can react to new attacks but with more false positives, where the beauty of misuse detection comes into play which detects known attacks reliably and with less false positives.

The greatest challenge with current intrusion detection tools nowadays is how to extrapolate previously observed behaviors to find similar future behavior?. This paper is interested in discussing intrusion detection methods that are based on biological models(neurons and a bit about immune systems). The author is inspired by how brains process information and learn. Inside the human brain, a neuron collects signals from other neurons through dendrites, and sends out the spikes of electrical activity which breaks into thousands of branches through axon. There is a structure at the end of each branch called synapse which converts the activity from axon to electrical effects and passes it other neurons. By changing the effectiveness of synapses, learning occurs. Later on, models that simulate the real neurons have been developed. A artificial neural network consist of processing units called nodes, and the connection between them called edges. The weighted edges used to determine how one unit influence the other. A portion of the unit of the network act as input nodes and the other portion of the network act as output nodes. Activation is propagated through the network by assigning value to each input node. A functional mapping which itself gets stored in the weights of the network happens from one set of values that is assigned to the input nodes to the other set of values that is retrieved from the output node.

Previous work [4],[5] developed intrusion detection methods based on how human immune system distinguish self from others and how they respond. In human immune system, the foreign bodies are recognized by epitopes, an antigenic determinant and by this way, human immune system distinguish antigens and triggers immune response for the particular antigen and triggers automated response using system call delays. The same way with computers the self is defined by short range correlations in a process system call. But the problem with this approach is, some privileged programs after a day or two, started exhibiting few anomalies and converged to a stable normal.

The idea of learning the behavior of a program in host based anomaly detection is in demand. Machine learning methods like statistical technique, hidden markov model and rule learning techniques have been used to study the behavior of the program as it can be seen as a binary classification problem which is most common problem in pattern differentiation. This paper reports how neural networks based intrusion detection methods outperforms all other existing host based anomaly and misuse detection mechanism. Also, profiling normal behavior using neural network based intrusion detection method takes long time and this paper explains the method to overcome this drawback and gives some direction to successfully apply neural networks for both anomaly and misuse host based detection methods.

**2 Approaches**

The view of using biologically inspired methods in computer security is a boon to traditional cryptography and other deterministic approaches. All the papers I selected focus on biological based models for host based intrusion detection methods.

The papers reviewed are:

*A study in using neural networks for anomaly and misuse detection* authored by Anup K. Ghosh and Aaron Schwartzbard - This paper was selected on the 8th USENIX security symposiums, August 23-36, 1999, Washington, D.C. and has been funded by Defense Advanced Research projects.

*Evolutionary neural networks for anomaly detection based on the behavior of the program* authored by Sang-Jun Han and Sung-Bae Cho - This paper was published on IEEE Transactions on SYSTEMS, MAN AND CYBERNETICS, June 2006.

*Intrusion detection* authored by D. Endler - This paper was selected for ACSAC '98 Proceedings of the 14th Annual Computer Security conference and also on IEEE Computer Society, Washington DC, USA.

*A sense of self for UNIX processes* authored by Stephanie Forrest, Steven A. Hofmeyr, Anil Somayaji and Thomas A.Longstaff- This paper was published 1996 IEEE Symposium on security and privacy, IEEE Computer Society Press, Los Alamitos, CA.

*Automated Response using System-Call Delays* authored by Anil Somayaji and Stephanie Forrest – This paper was selected for 9th USENIX security symposium 2000.

The previous work on intrusion detection was mostly building user profiles on a per user basis. [3] In this work, they recorded the profiles of software behavior and classified them as normal and malicious software behavior. This really benefits a user who does not want to be under surveillance. They used artificial neural networks for addressing the challenge of extrapolating and classifying incomplete data that current intrusion detection faces. A artificial neural network consist of processing units called nodes, and the connection between them called edges. The weighted edges used to determine how one unit influence the other. A portion of the unit of the network act as input nodes and the other portion of the network act as output nodes. Activation is propagated through the network by assigning value to each input node. A functional mapping which itself get stored in the weights of the network happens from one set of values that is assigned to the input nodes to the other set of values that is retrieved from the output node. A classical feed-forward multilayer perceptron network has been implemented which leads to construct two different artificial neural network, one for anomaly detection and other for misuse detection. To use this network, first they need to find a way to encode the data that inputs to the network, they need to be aware of network topology, they need to know how to perform anomaly detection with supervised training algorithm and what they are planned to do with the data produced by neural network. In order to filter the necessary information while encoding, they used distance metrics for string of events. The events common to two strings as well as difference in positions are taken into account. Basically the encoding is a set of measured distances in which strings can be visualized as a point in a space where each point is mapped in space by plotting the distance from each dimensions. Once after finding a way to encode data, they started looking into network topology. A neural network can be constructed for each program. The more anomalous the input, the output of the network closer to 1.0 and its closer to 0.0 if its less anomalous. Different networks are constructed, tuned and trained for every program to be monitored. During training, networks with best performance was selected and the rest of the networks discarded. The hidden nodes for a program is unknown before training, so for each program networks are trained with range of 10-60 hidden nodes. Also the weights have been initialized randomly that leads to poor performance and in order to avoid it, for each number of hidden nodes, 10 networks are initialized differently and trained. So for program, 90 networks are trained. Only the network with accurate classification of data was saved.

*Neural Networks for Anomaly detection*

Once selection and training is done, neural networks is ready to use. But unfortunately neural network can classify only a single string but in order to classify multiple programs, they used leaky bucket algorithm to keep track of recent memory events. The output from a neural network which is the classification of the input string goes into the leaky bucket. For each time step, the height of the bucket decreased by a fixed amount. If the height increases above the threshold, the program is marked as anomalous.

*Neural Networks for Misuse detection*

Modifying leaky bucket trivially, the system is used for misuse detection. The lack of data and and marking intrusions make misuse detection difficult. The data we are using have less intrusion data which makes it difficult for training networks. Also the intrusion data are marked on a session-by-session basis. Generally all data marked are assumed either anomalous or normal. Its basically an assumption.

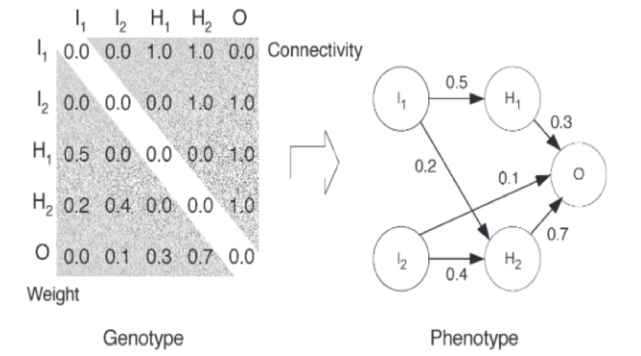
By following the simple neural network approach to learn previously observed behavior, they proved that this approach outperforms all other representative methods. Neural network hold all the aces for learning system call sequences. But it has its delimits as well. Profiling normal behavior due to large amount of audit data takes long time. As we already seen how important it is to determine the topology of the network and the hidden nodes inside it which directly affects the performance of the neural network. The reason for huge amount of delay to profile normal behaviors because the network structures are designed on trail and error cycles of previous problems. This is one major limitation in neural network based intrusion detection systems. In order to address this limitation, they developed a evolutionary neural network that does not require trial-and-error cycles for designing network structures.

|  |  |
| --- | --- |
| (a) | (b) |

**Figure 1:** In anevolutionary neural network (a)is the structure of ENN and (b)is the architecture of ENN. [2]

Evolutionary neural network based intrusion detection technique is deployed as its learning includes designing its own structure. Also it does not have any structural restrictions and with this, we can obtain better neural network in a short period of time. In this approach, a random population of individual neural network is generated with initial weights and full connection, each individual is assigned a fitness score and the next population is generated by genetic operations which means individual with best fitness goes into the next population. Before starting the genetic operation, each neural network information is learned beforehand using a back-propagation algorithm. The evolution is repeated for particular number of generations. To speed up evolutionary learning, a combination of Baldwinian and Lamarckian learning is used. In order to deploy neural network for evaluating abnormality of system-call sequences, they follow two approaches 1) classify the sequence into two classes: normal and attack behavior 2) predict next sequence at time n+1 with current input at time n which is kind of temporal modeling. But evolutionary neural network does not use temporal logics. By setting L input nodes with window length same as input nodes, the number of hidden nodes and their connections are determined dynamically by the evolutionary algorithm. In this approach, system call sequences are generated at random and training data by mixing normal and artificial sequences. There are three problem in using evolutionary neural network in practice they are 1)

representation of genotypes 2) genetic operations and 3) fitness evaluations.

**Figure 2: **Genotype and phenotype representations. [2]

*Representation of genotypes:*

Due to the use of matrix based genotype representations, it may include meaningless network structure and nodes. Also back propagation cannot be applied to network that contains forward links. But however these kinds of neurons are eradicated at the partial-learning stage.

*Genetic operations:*

Hidden node as pivot point is chosen randomly when doing crossover of two distinct neural networks. Also if there is no connection between two nodes, the mutation operator automatically connects the nodes and assigns random weights.

*Fitness Evaluation:*

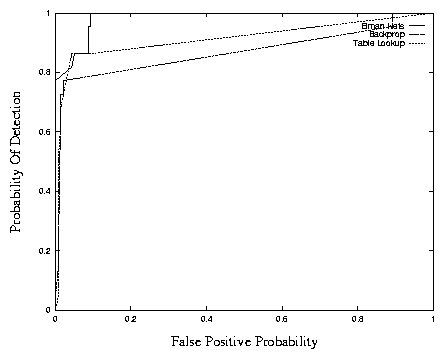
Rank based selection is used other than the traditional roulette and tournament selection. Rank of an individual are directly based on detection rate of the training data.

|  |  |
| --- | --- |
| (a) | (b) |

**Figure 3:** Genetic operation (a)crossover (b)mutation. [2]

**3 Results**

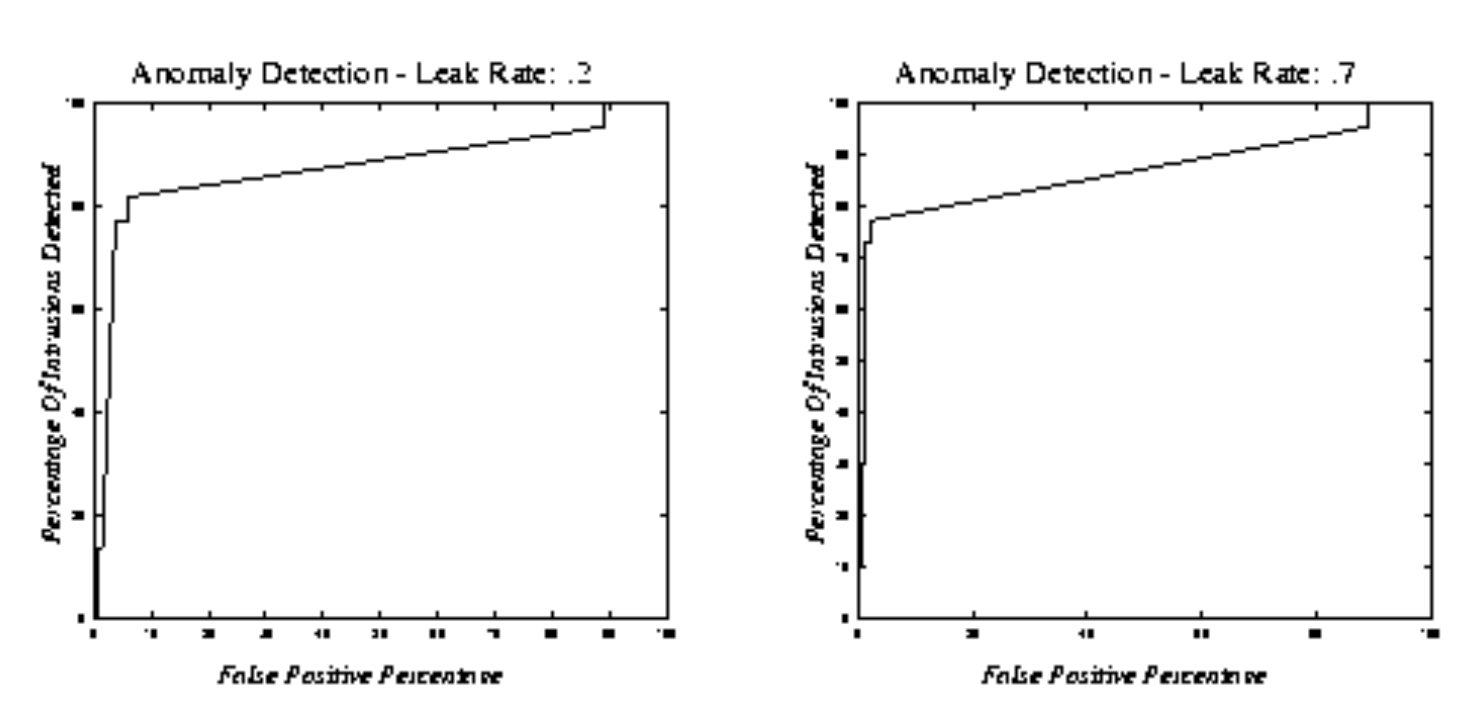
From [3], applying Elman recurrent network on the DARPA data along with equality matching shows better performance with reduced false rates. This becomes the first motivation for applying neural network for intrusion detection system.

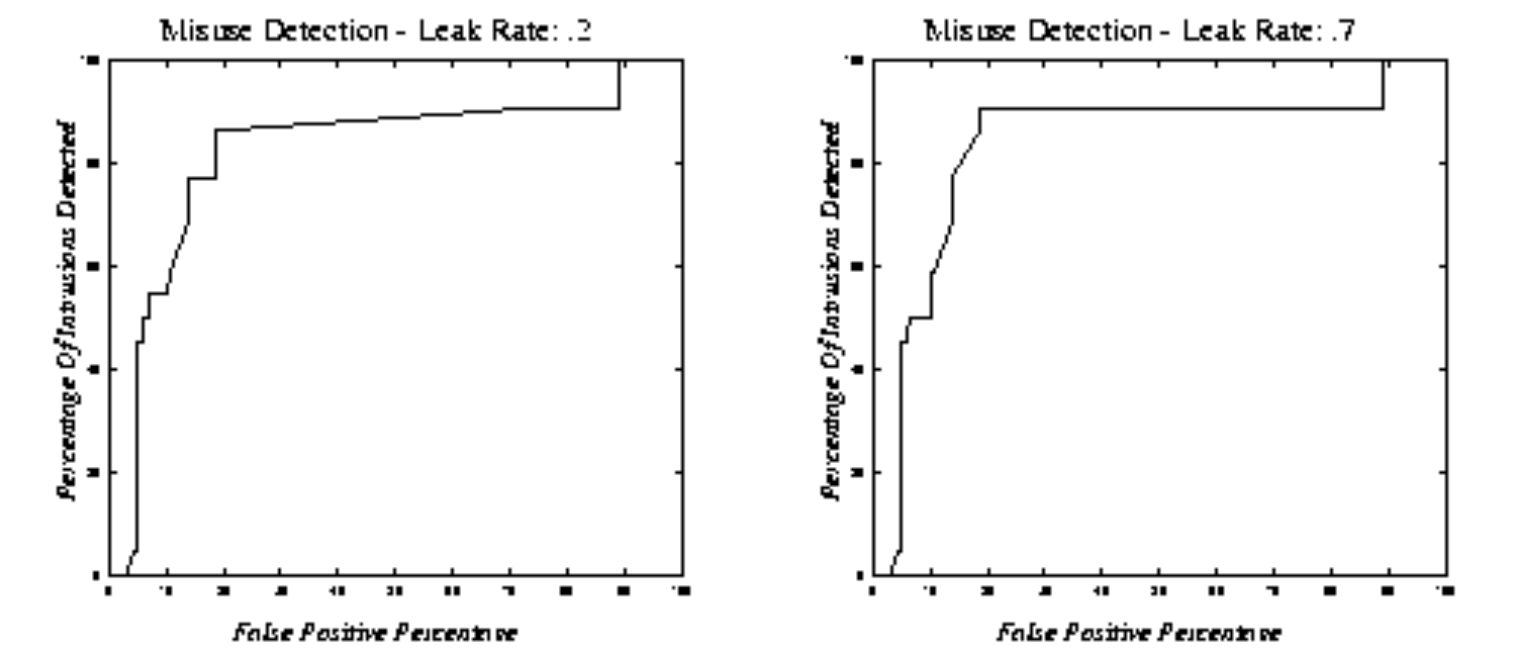
****

**Figure 4:** Performance measure of three intrusion detection algorithm, out of all three Elman network performs the best. [3]

MIT's Lincoln laboratory collects both network and audit data on host machines by setting up a private network. The laboratory already has a prior knowledge of the data and know which one is normal and which one is attack data. The network data is collected using a network sniffer and host machine audit data is collected using Sun Microsystem's Solaris Basic Security Module (BSM). They distributed the data to the project sites in two phases : training and testing data. The training data is data marked either as normal or attack and distributed to participating sites to train their corresponding intrusion detection system. After training, the test data is distributed in unlabeled form. The participating sites never knows which data in the test data is normal or attack. The data are separately analyzed by the participating sites again to find which sessions are normal and which has intrusions and the results are sent back to MIT's Lincoln Labs for evaluation.

The test data from these labs are tested for both anomaly and misuse detection. The test data contains 22 intrusive sessions and 139 non intrusive sessions. The IDS performance is judged by both of its false positives and its ability to detect intrusions. These factors can be observed by varying the leaky rate used by the leaky bucket algorithm. A receiver operating characteristic(ROC) curve is used to analyze the intrusion detection ability against false positives. This ROC curve basically used to evaluate the performance of a binary classifier system as its discrimination threshold is varied. Different leaky rates produce different ROC curves. The curve is a plot of the likelihood that an intrusion detected against the likelihood that a non intrusion is misclassified for a threshold. [1]For anomaly detection, with leak rate of 0.7, a detection rate of 77.3% can be achieved with false positive of 2.2%. For misuse detection, with leak of 0.7, a detection rate of 90.9% can be achieved with false positive of 18.7%. The false positives for the misuse detection. With these results, its proved that neural networks are best fit to perform intrusion detection.

**Figure 5:** Anomaly detection on two different leaky rates.[1]

**Figure 6:** Misuse detection on two different leaky rates.[1]

For evaluating the performance of evolutionary neural network, they used the BSM audit data which contains 280 system call events. Only critical system calls are used to model normal behavior of a program and rarely used system calls are ignored. There are totally 45 system calls that are numbered from 0 and 44. In order to verify the evolutionary neural network, they used the 1999 DARPA intrusion-evaluation dataset which contains four types of attacks: denial of service, probe,remove to local(R2L), and U2R. Specifically they monitor only the SETUID privileged programs that run on victim UNIX servers. The DARPA dataset consist of five weeks of audit data where 1-3 week dataset is used for training and 4-5 week dataset is used for testing. The weight to ans weigh to the output value of the attack node is set to 0.5. And also the weight to the output value to normal node is set to -0.5 to cut down of partially intrusive input. We characterize the results of evolutionary neural network in 5 different ways:1)change of fitness 2)change of network structure 3)comparison of training time 4)comparison of detection performance 5) comparison of network structures.

|  |  |  |
| --- | --- | --- |
| (a) | (b) | (c) |

**Figure 7:** Characterization of Results (a) fitness over evolution (b)network structure over evolution (c) ENN intrusion detection performance. [2]

*Analyses of fitness:*

In order to analyze whether neural network evolve or not, they observe fitness change. When evolution proceeds, fitness increases and the maximum converges to 0.9. From this, we can conclude, by evolution one can see better neural network that classified data with 90% accuracy.

*Analyses of Network Structure:*

In order to analyze whether neural network evolve with the change in hidden node and number of connections they observed evolutionary neural network with the set of full connections. Number of connections decrease while the evolution proceeds which shows that it optimizes the network structure by pruning unnecessary connections. There is no effect or change in the number of hidden nodes until the final generation. This shows that the evolutionary algorithm never eliminates hidden nodes.

*Analyses of training time:*

The training program run on a computer with duel Intel Pentium Zeon 2.4GHz processor, 1GB RAM and Sun Solaris 9 operating system. The basic neural network that repeats the trial and error cycle requires 17h and 50min, but evolutionary neural network only takes 1h and 14 min. Thus it proved evolutionary neural network reduce learning time.

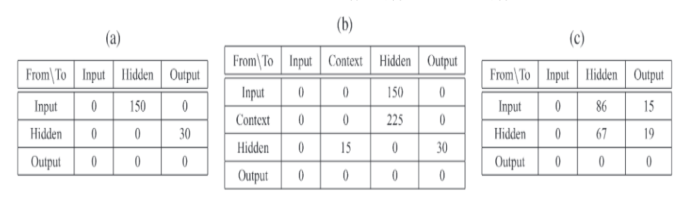
*Comparison of Detection Performance:*

A traditional way of representing the performance of classifier are using ROC curves. A neural network with highest fitness is selected and a curve is plotted with false alarm rate against detection rate. Evolutionary neural network produce 0.0011% of false alarms at 100% detection rate. The false alarms are raised by 'ffbconfig' of which training data is not sufficient. The previous Elman recurrent neural network showed 2.2% pf false alarms at 100% detection rate. This results proves that evolutionary neural network performs better than the conventional neural network of static and regular structures.

*Analyses of Network Structures:*

MLP, Elman and ENN (Ref. Table 1)runs on more complex network as ENN does not restrict the topology of the network. With Elman having large number of connections by making the context node receive input from single hidden node and output it to all other hidden nodes. By this way of adding context layer, Elman maintain its performance which is not revealed by [3].

**Table 1**

Comparing network structures (a)MLP (b)Elman network (c) ENN. [2]

**4 Discussion**

Since from 1990's, biological methods put its best foot forward by using finite state or graph models. However when you think of any biological related models, the main problem would be that developers do not even fully understand how they work. Then the results will be the logical implications of the assumptions that the model the developer interested in. Also biological models do have their own false positives, for example, immune system can wrongly differentiate self and non self in some cases which results in a autoimmune disease called rheumatoid arthritis. Validating a biological model is difficult and most researchers are caught between two stools of biological and formal models.

The papers [1],[2],[4],[5] contains audit data of sequences of system calls alone and all their implementation were based on collecting traces of system calls using strace ignoring their parameters and timing instructions. One such exploit is called race conditions. A TOCTOU exploit uses the gap between a test of a condition and use of a object which the authors completely ignored as their approach fails to detect this exploit or they might be ignorant towards it. TOCTOU is a series vulnerability and only recently have Microsoft and Apple found a mitigation for this attack.

In [1], the data from DARPA contains less intrusions, and they assumed that the data marked intrusive are normal and one marked non intrusive assumed to be abnormal. The data is not filtered and analyzed properly which ends up in poor results for signature based detection. Its not clearly mentioned in the paper, which version of leaky bucket algorithm is used, whether queue or meter based. But its understood they are using queue based leaky bucket algorithm but it is not discussed in the paper [1].

In [2], the output value that come from a attack node cycles periodically and exceeds the output value of a normal nodes for very short period of time which is not enough to classify the process as the attacked one. For this a threshold is used to check for abnormality. But applying a threshold to all neural network is not feasible. In order to enhance feasibility they used statistical for evaluation values which is given by,, if this evaluation value is greater than the threshold, then the process is said to be attacked, which makes ENN more efficient for intrusion detection. But still Elman network performed well for more complex structure than ENN which only tries to increase the modeling power for complex network structures and is discussed in section 3. However, ENN produce less false alarms with 100% detection rate than Elman network.

Moreover, speaking of genetic algorithm its been proved doing cross over at random points yield poor results than crossover at the defined fixed point. But in ENN's genetic operation cross over is done at random pivot points.

Apart from the suggestions given above, the authors were successful in finding a solution for the problem they want to solve. Now for Evolutionary neural network is good fit for profiling normal behavior at faster rate and anyone can use it detect intrusion efficiently than the conventional one.

**5 Summary and Conclusions**

This paper starts addressing the current challenge that current IDS facing is the capability to generalize the normal behavior which [3] proves that Elman network along with equality matching works fairly well. Motivated by the results of [3] , the authors of [1] started applying simple neural network to learn the previously observed behavior, and the results from this study paved the way to open the life of neural networks in detecting intrusions. However, it comes under the cost of long delay in profiling user behavior that has been addressed by[2] with its new approach of evolutionary neural networks. Evolutionary neural network not only improves the performance but also decreases the time for training. The experiment with same data used by [1] , ENN based detector showed better performance as it learns the structure and weights at the same time. In my opinion, regarding host based intrusion detection methods the best variants to look into, other than sequence of system calls might be the internal functioning of a program like considering the sequence of machine dependent register values and I/O's but the challenges to face with the assembly level instruction sequences will be that you might end up in sudden dead locks and context switching. In the future, I wish to extend the work with sequences of registers where the return of value of a system call ends up in storing, it might be EAX or EBX for 32 bit systems and RAX or RBX for 64 bit registers. Also in order to further improve the detection performance, one can do by joining many neural nets that evolves with speciation or developing a dynamic evolutionary algorithm that adapts to the change in network structure and window length. However results concludes that evolutionary neural network detectors are the best fit for intrusion detection and it outperforms all other conventional methods, which can now successfully apply to real world problems.

**6 Acknowledgements**

I would like to express my gratitude for Dr. Thomas Caudell, Professor of University of New Mexico who gave me an opportunity to research through the topic of biological based intrusion detection methods. I would also like to thank Dr. Stephanie Forrest, Professor at University of New Mexico who helped me in finding different models that already used for intrusion detection mechanism. And also I would like to thank Tony Amajest, Graduate Student at University of New Mexico for the ideas and suggestions to improve this literature relevant to the problem addressed.

**7 References**

[1] A.K.Ghosh and A. Schwartzbard, *“A study in using neural networks for anomaly and misuse detection,”* in *Proc.* *8th USENIX Secutity Symp*, Washington, D.C, August. 1999,

[2] Sang-Jun Han and Sung-Bae Cho, “*Evolutionary neural networks for anomaly detection based on the behavior of the program,”* in *Proc. IEEE Transactions on Systems, Man and Cybernetrics*, June. 2006.

[3] D.Endler, “I*ntrusion detection,”* in *Proc. 14th Annual Computer Security conference and in Proc. on IEEE Computer Society,* Washington DC, 1998.

[4] S. Forrest, S. A. Hofmeyr, A. Somayaji and T. A.Longstaff *, “A sense of self for UNIX processes,”* in *Proc.* *IEEE Symposium on Security and Privacy, IEEE Computer Society Press*, Los Alamitos, CA, 1996.

[5] A. Somayaji and S. Forrest ,”*Automated Response using System-Call Delays,”* in *Proc. 9th USENIX security symposium*, 2000.