Improved Detection System of Denial of Service Attack

Omar I. Alsaleh and Abdulkader A. Alfantookh

Computer Science Department, College of Computer and Information Sciences, King Saud University, Riyadh, {o.alsaleh, fantookh}@ccis.ksu.edu.sa, Saudi Arabia

***Abstract* — A problem with current intrusion detection systems is that they have many false positive and false negative events. Most of the existing Intrusion detection systems implemented nowadays depend on rule-based expert systems where new attacks are not detectable. In this paper, a possible application of Neural Networks is presented as a component of an intrusion detection system.**

***Index Terms* — Computer security, Artificial intelligence, Intrusion detection, Neural networks, Feedforward neural networks.**

I. Introduction

The potential damage that can be inflicted by attacks launched over the Internet keeps increasing due to growing reliance on the Internet and more widespread connectivity. Intrusion detection systems (IDSs) have now become an essential component of computer security: to detect attacks that occur despite the best preventive measures. Some approaches detect attacks in real-time and can be used to monitor and, possibly, stop an attack in progress. Others provide after-the-fact forensic information about attacks and can help repair damage, understand the attack mechanism, and reduce the possibility of future attacks of the same type. More advanced IDSs detect never-before-seen (unknown) attacks, while the more typical systems detect previously seen (known) attacks.

There are two general approaches to ID namely: misuse detection and anomaly detection. Methods of the first approach are dealing with prior prepared patterns, also called signatures, of known attacks that are used to detect intrusions by pattern matching on audit information. And, methods of the second approach are dealing with profiling user behavior. In other words, they define a certain model of a normal user activity. Any deviation from this model is regarded as anomalous. DoSID uses the anomaly detection approach.

Also, IDSs are classified according to the kind of input information they analyze. These classes are: application-based IDS, host-based IDS and network-based IDS. DoSID is network-based IDS.

In this paper, an intrusion detection system called Denial of Service Intelligent Detection (DoSID) is developed. The type of Neural Network used to implement DoSID is feed forward which uses the backpropagation learning algorithm. The data used in training and testing is the data collected by Lincoln Labs at MIT for an intrusion detection system evaluation sponsored by the U.S. Defense Advanced Research Projects Agency (DARPA). Results show that the best detection rate for new attacks is 68%. Also it has been shown in the final experiment that the false positive of the system has been reduced considerably.

In the following section, a brief introduction to Neural Network concepts is given. In section 3, DoSID framework and related improvements are explained. Then, the experimental results and analysis will be presented in section 4. Finally, the conclusion is in section 5.

II. *Neural Network Concepts*

A Neural Network is a structure which is composed of a number of simple elements or nodes called *neurons* as shown in Fig.1.

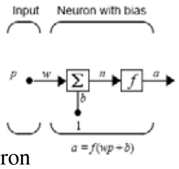

Fig. 1. Simple neuron

These elements are always operating in parallel. The function of the Neural Network is determined largely by the connection between the neurons. These neurons are connected by links and each link is adjusted by values called *weights*. The process of updating the weights is called *learning*.

Neuron showed in Fig1 is composed of: input *p* associated with weight *w* and there is a scalar bias *b*. The equation

$$n=wp+b \quad (1)$$

forms an input to the second main component which is the *transfer function*. The output of the neuron is the output of the transfer function. The general equation is

$$a = f\,(wp+b). \quad (2)$$

Here *f* is a transfer function which takes the argument *n* and produces the output *a*. The Neural Network will exhibit the desired or interested behavior by adjusting its parameters. That means, the Neural Network can be trained to a particular job by adjusting the weight or bias parameters or perhaps the network itself will adjust these parameter to achieve some desired results.

One of the most commonly used Neural Networks is the multilayer feed-forward network. It falls under the category called “Networks for Classification and Prediction”. The DoSID is built using this specific type of Neural Network.

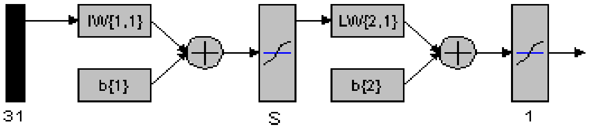

Fig. 2. Neural Network Architecture for DoSID

Feed-forward networks usually consist of two to three layers in which the neurons are logically arranged. The last layer is the output layer and there are usually one or more hidden layers before the output layer. The DoSID Neural Network as shown in Fig 2 is composed of two layers (the hidden and the output layer), a variable number of neurons in the hidden layer and there is one neuron in the output layer. Each output vector element value is in the range [-1,1]. The transfer functions of neurons on both layers are "tan-sigmoid" function. This function takes the input, which may have any value between plus and minus infinity, and squashes the output into the range [-1,1]. The input vector contains 31 elements. These elements are the result of converting the 18 features in the DARPA dataset to Neural Network format.

The most common and widely used learning algorithm for multilayer feed forward Neural Networks is the backpropagation algorithm. It is based on the *Delta Rule* that basically states that if the difference (delta error) between the user's desired output (target) and the network's actual output is to be minimized, the weights must be continually modified. The result of the transfer function changes the delta error in the output layer. The error in the output layer has been adjusted, and therefore it can be used to change the input connection weights so that the desired output may be achieved. This is why feed-forward networks are also often called "backpropagation feed-forward networks". The learning mechanism is illustrated in Fig 3.

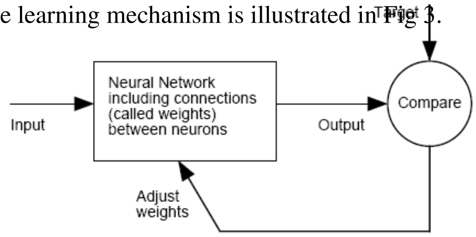
Target
Neural Network including connections (called weights) between neurons
Compare
Input
Output
Adjust weights

Fig. 3. Neural Network learning mechanism

The input connection weights are adjusted in such a way that the delta error will be minimized. This process is repeated several times (*Iterations*). The training stops if: the number of iterations exceeds a certain number of iterations, the training performance function drops below certain threshold of MSE, or the training time is longer than certain threshold of seconds. The mean squared error (*MSE*) is computed by summing the squared differences between the target and the network's actual output, and then dividing the sum by the number of components (input vector elements) that went into the sum.

III. Denial of Service Intelligent Detection System Description

A. DoSID Framework

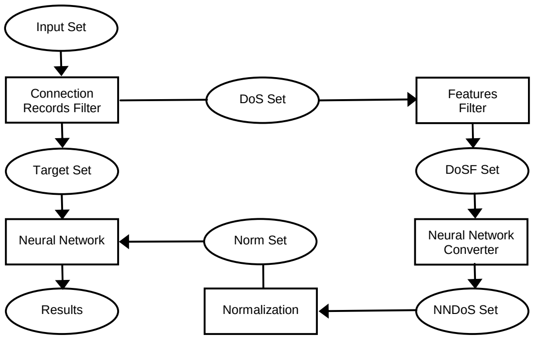

Fig. 4. DoSID Architecture

DoSID is dedicated to DoS attacks. Therefore, the types of connection records needed in our experiments are only normal traffic and any DoS attacks. The role of *Connection Records Filter* module is to filter the *Input set* to contain only normal and DoS attacks. The filtered set is called *DoS Set*. For each connection record in *DoS Set*, the *Connection Record Filter* module prints in separate set the class of that record. It prints either 1 for normal or -1 for DoS attacks in separate line. This set is called *Target set*.

Each connection record contains 41 features. Only 18 features that are useful for detecting DoS attacks are used. *Features Filter* module extracts the needed features (18 features) from each connection record in *DoS Set* and stores this record in *DoSF Set*. The *DoSF Set* is converted to Neural Network format to be readable by the Neural Network and this is the role of *Neural Network Converter* module. The result of this module will be in *NNDoS Set*. To improve the decision making of Neural Network, a module called *Normalization* has been implemented to make all values in each connection record homogenous by normalizing each value in *NNDoS Set*. The normalized connection records in are *Norm Set*. The last set is used as input to *Neural Network* module. As described in the previous section, this module is a Neural Network composed of two layers (the hidden and the output layer), with a variable number of neurons in the hidden layer and one neuron in the output layer as depicted in Fig 2.

Three improvements to DoSID are added by developing and implementing different techniques in order to enhance detection accuracy, decision making, and Neural Network performance (MSE) for testing phase.

B. Improvement 1: the Gray Area

Neural Network predicts the type of each connection record. Its output is a vector that consists of one element which falls in the range [-1, 1]. The connection record is classified as normal when the vector value is around 1, where values around -1 indicate a DoS attack connection record.

Wherever the output value is closer to 1 or -1, the Neural Network decision becomes more accurate. The further the value from 1 and -1 toward 0 indicates non-accurate decision. Therefore the gray area concept is proposed to improve the accuracy of Neural Network as depicted in Fig 5.

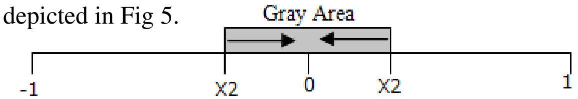

Fig. 5. the Gray Area Concept

The gray area is an area inside the range of the output value. The value that gets in this area [x1, x2] is not accurate because it is far from 1 or -1. So, the value is changed to zero which means that connection record is unrecognized.

The most critical issue in the gray area concept is its boundaries x1 and x2. The values of boundaries are selected based on the desired objective of the gray area. For example, in a strict environment where any possible intrusion is to be reported regardless of the high false positive warnings, the value of x1 is increased.

The size of the gray area depends on overall Neural Network results or decisions for each connection record in the training set. In order to specify the boundaries of the gray area, the Distribution concept is introduced.

C. Improvement 2: Distribution

The Neural Network gives highly accurate decisions for connection records that were used in the training phase. The output value of each connection record during the training phase is distributed over the range of output value.

To do this distribution, first, the range is divided into small intervals. The length of each interval is 0.1. A counter is assigned for each interval to count the number of connection records that the corresponding Neural Network vector output value lies in.

The gray area will include each interval that has small counter value. For example, in one of the experiments, the distribution of 9979 connection records is as in Table 1. So, the range of the gray area will be [-0.8, 0.9)

Table 1: Distribution training data with 24 neurons and 1000 iterations

Intervals	***[-1,-0.9)***	***[-0.9,-0.8)***	***[-0.8,-0.7)***	***[-0.7,-0.6)***	***[-0.6,-0.5)***
Records No.	5974	6	2	0	0
Intervals	***[-0.5,-0.4)***	***[-0.4,-0.3)***	***[-0.3,-0.2)***	***[-0.2,-0.1)***	***[-0.1,0)***
Records No.	0	0	0	2	0
Intervals	***[0,0.1)***	***[0.1,0.2)***	***[0.2,0.3)***	***[0.3,0.4)***	***[0.4,0.5)***
Records No.	0	0	0	0	0
Intervals	***[0.5,0.6)***	***[0.6,0.7)***	***[0.7,0.8)***	***[0.8,0.9)***	***[0.9,1]***
Records No.	0	2	0	1	3999

D. Improvement 3: Normalization

Usually the input vector should be normalized before train or testing sessions. The normalization method which called *Normalization to Zero Mean and Unit Standard Deviation* method has been used for DoSID. This normalization algorithm ensures that all elements of the input vector are transformed into the output vector in such a way that the mean of the output vector is approximately Zero, while the standard deviation (as well as the variance) is in a range close to unity.

The mean(μ)of all elements(x_i)of a vector are calculated according to following formula:

$$\mu = \frac{1}{N}\sum_{i=1}^{N} x_i \qquad (3)$$

The standard deviation can be expressed as:

$$\sigma = \sqrt{\frac{\sum_{i=1}^{N}(x_i-\mu)^2}{N-1}} \qquad (4)$$

The use of this formula depends on a pre-calculated (μ). From a programmer's point of view, the following equivalent formula would be computationally less expensive, since it requires only one iteration over the input vector:

$$\sigma = \sqrt{\frac{N\left(\sum_{i=1}^{N} x_i^2\right) - \left(\sum_{i=1}^{N} x_i\right)^2}{N(N-1)}} \qquad (5)$$

The input vector x has to transform into the (normalized) output vector x' by applying

$$x_i' = \frac{x_i - \mu}{\sigma} \quad i \in N \qquad (6)$$

yielding for the output vector the conditions:

$$\sigma_{x'} \approx 1 \qquad (7)$$
$$\mu_{x'} \approx 0$$

IV. Experimental Results And Analysis

In this section, experiments are conducted and their results are presented along with the different improvements proposed in the previous section.

For training and testing the Neural Network, connection records are collected randomly from the DARPA training dataset. Therefore, all connection records are labeled. They are used in the training phase to inform the Neural Network about the type of the connection (attack or normal), so the Neural Network will learn from each connection record (i.e. by adjusting its weights).

On the other hand, the labeled connection records are useful in the testing phase for measuring the accuracy. This is done by feeding each connection record to the Neural Network. The normal and attack labeled records are fed separately. By computing the average output, the accuracy of the Neural Network decision is obtained.

In order to test the Neural Network against known and unknown attacks, it is trained with specific attacks. Some other attacks are left for testing unknown attacks. The six DoS attacks are categorized as follows:

1) **Training Attacks:** Four attacks are used for training the Neural Network. These attacks are Back, Neptun, Smurf and Teardrop. The labeled connection records using these attacks are used to train the Neural Network.
2) **Testing Attacks:** Two training attacks are used for testing the Neural Network. These attacks are Back and Smurf, which have been recognized by the Neural Network in the training phase. These are called

Known Attacks. Also, there are two attacks which are not seen by the Neural Network in the training phase. These attacks are Land and Pod which are called *unknown* attacks.

A. Training Experiments

In the experiments conducted, various Neural Networks are trained using the training set. This set contains about 10000 connection records. 4000 connection records are labeled with Normal and 6000 connection records are labeled with one of training attacks namely Back, Neptun, Smurf or Teardrop.

In the training phase, diverse methods are used to train the Neural Networks in order to achieve good performance. Actually, there are many factors affecting the Neural Network. Some of the factors that are considered as the most significant factors affecting the neural network decision making are

1) **Number of neurons per layer:** The Neural Networks is built using one hidden layer that contains 24 or 64 neurons.
2) **Number of iterations (epochs):** Each Neural Network is trained twice, using 1000 iterations and 5000 iterations.
3) **The initial weights and bias:** The initial weights of each Neural Network training session are selected based on the following methods:
    - *Zeros initial*: The initial weights are zeros.
    - *Training initial*: The initial weights are the resultant weights from a Neural Network that has been trained using 200,000 iterations.
    - *Random initial*: The initial weights are generated randomly.

To show the effects of these factors, the experiments are conducted using all combinations. The training performance is measured using the mean square error (MSE). As mentioned before, the MSE is the difference between the target and the Neural Network's actual output. So, the best MSE is the closest to 0. If MSE is 0, this indicates Neural Network's output is equal to the target which is the best situation.

B. Training Experiments

To show the accuracy of the Neural Network decisions with each type of connection records, the connection records have been randomly collected from DARPA training dataset where these records did not exist in the training set. The connection records are separated into three sets. Table 2 lists these sets.

Table 2: Distribution training data with 24 neurons and 1000 iterations

Set Name	*Connection records*	*Possible label(s)*
Normal Set	70 records	Normal
Known Set	60 records	back, smurf
Unknown Set	50 records	land, pod

The average of the Neural Network output for each connection record is used as a measure of the Neural Network accuracy. In case using the Normal set, the best average will be 1. If Known set or Unknown set are using, the best average will be -1.

C. Experiments with Un-Normalized Data Set

C.1 Experimental Results for Un-Normalized Training Data Set

Table 3 shows the training performance (MSE) that results from training some Neural Networks by using regular training set.

Table 3: The MSE for training sessions by using non-normalized training data set

Non-normalized Training Set	**24 neuron**		**64 neuron**	
	1000 iterations	**5000 iterations**	**1000 iterations**	**5000 iterations**
Zeros initial	0.0021588	0.0016041	0.0019362	0.0015045
Training initial	0.0001617	0.00015721	0.00101543	0.00101404
Random initial	1.60335	0.00227416	0.001334	1.60337

So, the MSE for training by using training initial weights is better than zeros or random initial weights. This is because the training initial weights are very close to the ideal weights because the training initial weights are generated by using 200000 iterations, regular training set and random initial weights.

The shaded MSE in table 3 is for a Neural Network with 24 neurons and has been trained using 5000 iteration. This Neural Network has the best performance. This particular Neural Network is used in the testing phase. The next section shows the results of the testing phase.

C.2 Experimental Results for Un-Normalized Testing Data Set and Without the Gray Area

The Neural Network is tested without using the gray area. The Neural Network decision for each connection record must be normal or attack. There are no undefined connection records because there is no the gray area to indicate uncertainty. Any connection record is considered normal if its output lies in the positive side. If the connection record output lies in the negative side, it will be considered an attack as depicted in Fig 6.

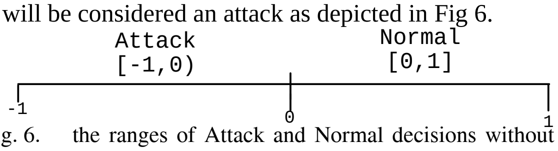

Fig. 6. the ranges of Attack and Normal decisions without the gray area

The results of testing the Neural Network using the testing sets are shown in Table 4.

Table 4: Results for testing a Neural Network with 24 neurons and 5000 iterations without the gray area

Without the Gray Area	Detection Rate	False Alarm		Connection records		Average	MSE
		False Positive	False Negative	DoS	Normal		
Normal (70)	91.42%	8.57%	▨	6	64	0.835604	0.3103
Known (60)	100%	▨	0%	60	0	-0.9936	0.0000747
Unknown (50)	60%	▨	40%	30	20	-0.38181	0.91189

As shown in Table 4, the neural network correctly detected 100% of the known attacks and, for the normal traffic; the false positive indicator is low (less than 9%). It has been noticed that the Neural Network can detect 60% of the unknown attacks, which proves that the Neural Network can detect new attacks, but the false negative indicator is still high.

C.3 Experimental Results for Un-Normalized Testing Data Set and With the Gray Area

To see the effect of the gray area, the Neural Network is tested again using the same testing sets that are used in the previous tests but using the gray area. The Distribution is used to determine the boundaries of the gray area. Table 5 shows the distribution of simulation of the training set using the Neural Network under testing.

Table 5: The distribution of simulation of the training set

Intervals	***[-1,-0.9)***	***[-0.9,-0.8)***	***[-0.8,-0.7)***	***[-0.7,-0.6)***	***[-0.6,-0.5)***
Records No.	***5975***	***4***	***0***	***0***	***0***
Intervals	***[-0.5,-0.4)***	***[-0.4,-0.3)***	***[-0.3,-0.2)***	***[-0.2,-0.1)***	***[-0.1,0)***
Records No.	***0***	***0***	***0***	***0***	***0***
Intervals	***[0,0.1)***	***[0.1,0.2)***	***[0.2,0.3)***	***[0.3,0.4)***	***[0.4,0.5)***
Records No.	***1***	***0***	***0***	***0***	***0***
Intervals	***[0.5,0.6)***	***[0.6,0.7)***	***[0.7,0.8)***	***[0.8,0.9)***	***[0.9,1]***
Records No.	***0***	***0***	***0***	***2***	***3997***

From table 5, the density of records exists in intervals: [-1,0.8) which means that 5979 records are attacks, and [0.8,1] which means that 3999 records are normal. From this distribution, the range of the gray area is [-0.8,0.8). The connection records lying in this range as "Unrecognized" records are considered. The connection records lying in the positive side are "Normal" and in the negative side are "Attack". (See Fig 7)

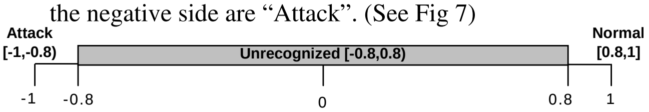

Fig. 7. the gray area range for experiment

Table 6 shows the result of testing the Neural Network using the gray area and testing sets.

Table 6: Results for testing a Neural Network with 24 neurons and 5000 iterations without the gray area

With Gray Area	Detection Rate	False Alarm		Connection records			Average	MSE
		False Positive	False Negative	DoS	Normal	Unrecognized		
Normal (70)	91.42%	5.71%		4	64	2	0.856167	0.285189
Known (60)	100%		0%	60	0	0	-0.9936	0.0000747
Unknown (50)	58%		8%	29	4	17	-0.5074	0.608

By applying the gray area concept, there is considerable improvement in the results in two aspects. First, it minimizes the false negative indicator from 40% to 8%. Second, it shows the high level of accuracy of the Neural Network's decision where most of the output fell very close to 1 in case of Normal connection records and very close to -1 in case of Attack connection records.

D. Experiments with Normalized Data Set

D.1 Experimental Results for Normalized Training Data Set

As the previous section, same combinations of training sessions is used but with normalized training set. Table 7 shows the results.

Table 7: The MSE for training sessions by using the normalized training data set

Normalized Training Set	**24 neuron**		**64 neuron**	
	1000 iterations	**5000 iterations**	**1000 iterations**	**5000 iterations**
Zeros initial	0.056107	0.0016532	0.049531	0.001558
Training initial	0.000033697	0.0000330	0.0000210	0.0000205
Random initial	0.0096744	0.0024496	0.0047671	0.0020184

By comparing the best MSE for training sessions in table 3, which based on un-normalized training set, and table 7, it has been found significant improvement in the MSE values. This approves that the normalization concept improves the Neural Network decision making.

In table 7, the shaded MSE is for a Neural Network with 64 neurons and has been trained using 5000 iteration. This Neural Network has the best performance. This particular Neural Network has been used in the testing phase. The next section shows the results of the testing phase.

D.2 Experimental Results for Normalized Testing Data Set and Without the Gray Area

The neural network has been tested without the gray area. The results of testing the Neural Network using the normalized testing sets are shown in Table 8.

Table 8: Results for testing a Neural Network with 24 neurons and 5000 iterations without gray area

Without the Gray Area	Detection Rate	False Alarm		Connection records		Average	MSE
		False Positive	False Negative	DoS	Normal		
Normal (70)	90%	10%		7	63	0.654565	2.88339139
Known (60)	100%		0%	60	0	-1.00277	0.00044842
Unknown (50)	32%		68%	16	34	-0.58117	0.395

D.3 Experimental Results for Normalized Testing Data Set and With the Gray Area

The same neural network has been tested again but with the gray area. Table 9 shows the distribution of simulation the training set by using the Neural Network under testing.

Table 9: Results for testing a Neural Network with 24 neurons and 5000 iterations without the gray area

Intervals	***[-1,-0.9)***	***[-0.9,-0.8)***	***[-0.8,-0.7)***	***[-0.7,-0.6)***	***[-0.6,-0.5)***
Records No.	***3872***	***5***	***2***	***1***	***0***
Intervals	***[-0.5,-0.4)***	***[-0.4,-0.3)***	***[-0.3,-0.2)***	***[-0.2,-0.1)***	***[-0.1,0)***
Records No.	***0***	***0***	***0***	***0***	***0***
Intervals	***[0,0.1)***	***[0.1,0.2)***	***[0.2,0.3)***	***[0.3,0.4)***	***[0.4,0.5)***
Records No.	***0***	***0***	***0***	***0***	***0***
Intervals	***[0.5,0.6)***	***[0.6,0.7)***	***[0.7,0.8)***	***[0.8,0.9)***	***[0.9,1]***
Records No.	***0***	***0***	***4000***	***0***	***2099***

Table 9 shows the density of records exists in two intervals: [-1,-0.6) which means that 3880 records are attacks, and [0.7,0.8) which means that 6099 records are normal. From this distribution the range of the gray area is [-0.6,0.7). The connection records that lie in this range will be considered as "Unrecognized" records. The connection records that lie in the positive portion are “Normal” and that in the negative portion are “Attack”. (See Fig.8)

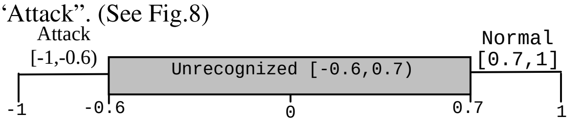

Fig. 8. The ranges of Attack, Normal and Unrecognized decisions with the gray area

Table 10 shows the result of testing the Neural Network by using the gray area and testing sets.

Table 10: Results for testing a Neural Network with 64 neurons and 5000 iterations with the gray area

With Gray Area	Detection Rate	False Alarm		Connection records			Average	MSE
		False Positive	False Negative	DoS	Normal	Unrecognized		
Normal (70)	90%	1.42%		1	63	6	0.690109	0.04119131
Known (60)	100%		0%	60	0	0	-1.00277	0.00044842
Unknown (50)	68%		0%	34	0	16	-0.60436	0.344

With comparing table 9 and table 10, the effect of the gray area is obvious. The gray area improves the detection rate of un-known attacks from 32% to 68%. Also, it improves the false positive rate by decreasing it from 10% to 1.42% for normal connections.

V. Conclusion

The Neural Networks provide a number of advantages in the detection of new attacks. In this paper, the DoSID system as a network-based IDS is introduced using Neural Network to detect Denial of Service attacks. The training dataset from DARPA is used to train and test our Neural Network.

The ability of a feed-forward Neural Network is tested to classify normal traffic correctly and to detect attacks. It has been found that the Neural Network detects the known attacks which have been used in the training of the Neural Network. Also, it has been found that the Neural Network can detect unknown attacks which have never been used in the training phase. These results mean that the Neural Networks are a significant technique to detect new attacks.

The gray area improvement is proposed which uses the distribution concept to determine the boundaries of the gray area. The two gray area experiments have improved the false negative indicator from 40% to only 8% for un-known attacks, and increased the accuracy of Neural Network decisions.

Also, the Normalization of the training/testing set is introduced to improve the Neural Network decision making. The experiments show how the normalization increases the detection rate and decreases the false positive rate with/without the gray area.

In the experiments, various Neural Networks have been trained using different combinations of factors. The experiments are separated into groups. First group is without normalization and the second group is with normalization. Both groups of experiments have been conducted with/without the gray area. As a result, the effect of each improvement of the detection rate or on the false positive rate can be analyzed separately.

While this research is about improving anomaly IDS, more emphasis is done on the improvement of detection rate for un-known attacks. The experiment resulted in the highest detection rate for un-known attacks is the one that used the gray area with normalized data set. In addition, this experiment considerably reduces the false positive rate to 1.42%. Also, for this particular neural network, we had the best detection rate of 84% on 110 attacks (both known and unknown attacks). Comparing this with the work in [10] where the detection rate is 77.3% or with the work in [11] where the detection rate is 80%.

References

[1] **Verwoerd, T. and Hunt, R. “Intrusion Detection Techniques and Approaches”,** *Computer Communications*, **vol. 25, no. 15, pp. 1356-1365, September 2002.**

[2] **Julia, A., Christie, A., Fithen, W., McHugh, J., Pickel, J. and Stoner, E.** “State of the Practice of Intrusion Detection Technologies (CMU/SEI-99/TR-028)”. *Pittsburgh,PA: Software Engineering Institute,CarnegieMellonUniversity*, **2000.**

[3] **Debar, H., Dacier, M. and Wespi. A.** “Towards a Taxonomy of Intrusion-Detection Systems”. *Computer Networks*, **vol. 31, no. 8, pp. 805-822, April 1999.**

[4] **Zhang, Z., Li, J., Manikopoulos, C., Jorgenson, J. and Ucles, J.** “A Hierarchical Anomaly Network Intrusion Detection System Using Neural Network Classification”, *WSES International Conference on: Neural Networks and Applications* **(NNA’01), pp. 85-90, Feb. 2001.**

[5] **Cannady, J.** “Artificial Neural Networks for Misuse Detection.” *National Information Systems Security conference (NISSC'98),* **October 5-8 1998. Arlington, VA.**

[6] **Ghosh, A.K., Wanken, J. and Charron, F.** “Detecting Anomalous and Unknown Intrusions Against Programs” . *Annual Computer Security Applications Conference (ACSAC'98)*, **December 1998.**

[7] **Hettich, S. and Bay, S. D.** “The UCI KDD Archive” [http://kdd.ics.uci.edu]. *Irvine, CA: University of California, Department of Information and Computer Science.***1999**

[8] **Richard P. Lippmann, David J. Fried, Isaac Graf, Joshua W. Haines, Kristopher R. Kendall, David McClung, Dan Weber, Seth E. Webster, Dan Wyschogrod, Robert K. Cunningham, and Marc A. Zissman** “Evaluating Intrusion Detection Systems: The 1998 DARPA Off-line Intrusion Detection Evaluation” *IEEE*, **1999.**

[9] **R K Cunningham, R P Lippmann, D. J. Fried, S. L. Garfinkel, I Graf, K R Kendall, S. E. Webster, D. Wyschogrod and M. A. Zissman** “Evaluating Intrusion Detection Systems without Attacking your Friends: The 1998 DARPA Intrusion Detection Evaluation”, *3rd Third Conference and Workshop on Intrusion Detection and Response*, **San Diego, CA, 1999**

[10] **A. K. Ghosh and A. Schwartzbard,** “A Study in using Neural Networks for Anomaly and Misuse Detection”, *Proc. 8th USENIX Security Symposium, Washington*, **USA, 23-26 August 1999.**

[11] **R. F. Erbacher, K. L. Walker and D. A. Frincke**, “Intrusion and Misuse Detection in Large-Scale Systems”, *IEEE Computer Graphics and Applications,* **Vol. 22, No. 1, pp. 38-48, January/February 2002.**