An Enhanced Resilient Backpropagation Artificial Neural Network for Intrusion Detection System

Reyadh Shaker Naoum[1], Namh Abdula Abid[2] and Zainab Namh Al-Sultani[3]

[1, 3]College of Information Technology, Middle East University
[2]College of Science for Women, Baghdad University

Summary

The potential threats and attacks that can be caused by intrusions have been increased rapidly due to the dependence on network and internet connectivity. In order to prevent such attacks, Intrusion Detection Systems were designed. Different soft computing based methods have been proposed for the development of Intrusion Detection Systems. In this paper a multilayer perceptron is trained using an enhanced resilient backpropagation training algorithm for intrusion detection. In order to increase the convergence speed an optimal or ideal learning factor was added to the weight update equation. The performance and evaluations were performed using the NSL-KDD anomaly intrusion detection dataset. The experiments results demonstrate that the system has promising results in terms of accuracy, storage and time; the designed system was capable to classify records with a detection rate about 94.7%.

Key words:

Input here the part of 4-5 keywords.

1. Introduction

Attacks on computer infrastructures are becoming an increasingly serious problem nowadays, therefore several information security techniques are available today to protect information systems against unauthorized use, duplication, alteration, destruction and viruses attacks [1].

Researcher in [2] agrees that detection of computer and network system intrusions has always been an elusive goal for system administrators and information security researchers. The individual creativity of attackers, the wide range of computer hardware and operating systems, and the ever-changing nature of the overall threat to target systems have contributed to the difficulty in effectively identifying intrusions.

Given the level and nature of modern network security threats, the question for security professionals should not be whether to use intrusion detection, but which intrusion detection features and capabilities to use. Intrusion Detection Systems have gained acceptance as a necessary addition to every organization's security infrastructure [3].

2. Related Work

The ability of soft computing techniques for dealing with uncertain and partially true data makes them attractive to be applied in intrusion detection. Some studies have used soft computing techniques other than ANNs in intrusion detection. Li, Zhang & Gu [7] proposed an anomaly based network intrusion detection system based on Multilayer perceptron with single hidden layer trained by Backpropagation learning algorithm. The system operation was divided into three stages: Input Data Collection and Preprocessing, Training, and Detection stage. The result for the proposed module was 95% detection rate. Sammany et al. [10] developed an intrusion detection system and classification attacks using artificial neural networks. They were able to design a multilayer perceptron capable to distinguish only 2 types of attacks (Neptune, Satan) from normal. The proposed MLP architecture was trained using Backpropagation algorithm with two hidden layers and three class output neurons. The results showed that the system was able to classify records with 93.43% detection rate. Al-Rashdan [4] has proposed an intelligent model using Hybrid Artificial Neural Networks, supervised and unsupervised learning capabilities to classify and / or detect network intrusions from the KDDCup'99 dataset. She designed three cooperative phases by using an enhanced k-means clustering algorithm in Phase-1 "clustering phase", a Hybrid Artificial Neural Network (Hopfield and Kohonen-SOM with Conscience Function) in Phase-2 "training phase" and a Multi-Class Support Vector Machines in Phase-3 "testing phase". The Hybrid Neural Network Machine Learning Model achieved a detection rate of 92.5%.

3. Proposed System

Expert systems and Artificial Neural Networks (ANN) are the most commonly used approaches in Intrusion Detection Systems [8]. Neural networks are a uniquely powerful tool in multiple class classification, especially

Manuscript received March 5, 2013
Manuscript revised March 20, 2013

when used in applications where formal analysis would be very difficult or even impossible, such as pattern recognition and nonlinear system identification [10]. Because of their generalization feature, neural networks are able to work with imprecise and incomplete data. It means that they can recognize also patterns not presented during a learning phase. That is why the neural networks could be a good solution for detection of a well- known attack, which has been modified by an aggressor in order to pass through the firewall system. In that case, traditional Intrusion Detection Systems, based on the signatures of attacks or expert rules, may not be able to detect the new version of this attack [6].

The proposed system is divided into 3 stages: Dataset features and pre-processing, training enhanced resilient backpropagation neural network and testing the system.

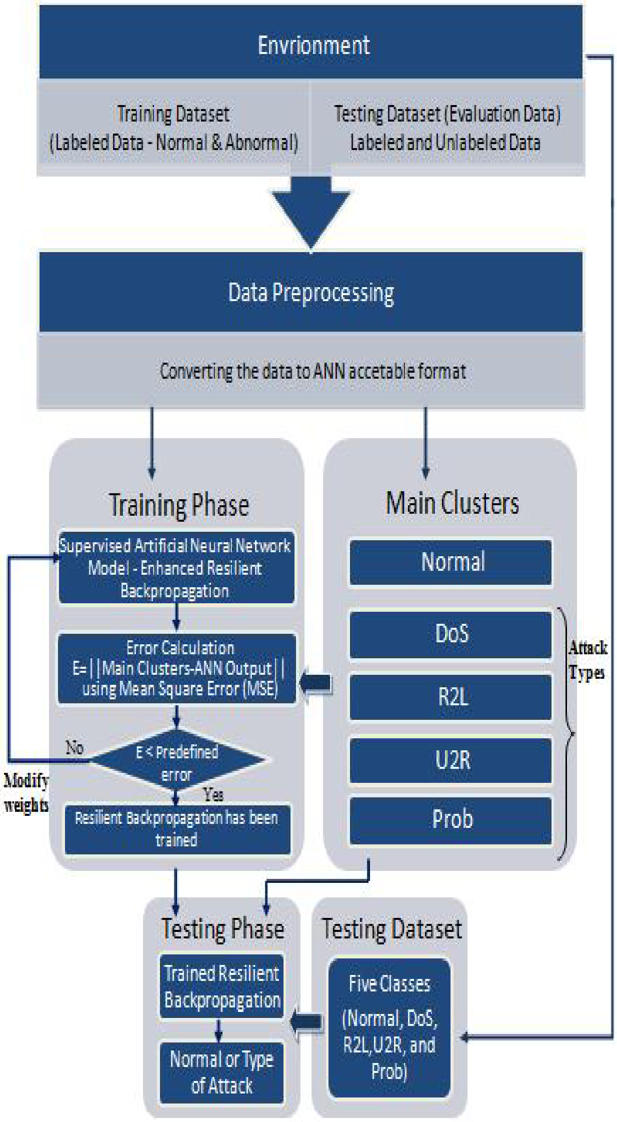

Envrionment
Training Dataset (Labeled Data - Normal & Abnormal)
Testing Dataset (Evaluation Data) Labeled and Unlabeled Data
Data Preprocessing
Converting the data to ANN accetable format
Training Phase
Supervised Artificial Neural Network Model - Enhanced Resilient Backpropagation
Error Calculation E=||Main Clusters-ANN Output|| using Mean Square Error (MSE)
No
E < Predefined error
Yes
Modify weights
Resilient Backpropagation has been trained
Main Clusters
Normal
DoS
R2L
U2R
Prob
Attack Types
Testing Phase
Trained Resilient Backpropagation
Normal or Type of Attack
Testing Dataset
Five Classes (Normal, DoS, R2L,U2R, and Prob)

Fig. 1 Proposed System ERBP

3.1 Dataset Features & Pre-Processing Stage

The dataset that will be used for training and testing is "NSL KDD-99" [5, 11]. The NSL KDD dataset includes a wide variety of intrusions together with normal activities simulated in a military network environment. The simulated attacks fall in one of four major categories: DoS (denial of service), R2L (unauthorized access from remote machine), U2R (unauthorized access to local superset privilege) and Probing (surveillance and other probing). Each instance in the dataset consists of the extracted features of a connection record. There are 41 features and they are either symbolic or continuous. The following operations are applied to the training and testing datasets:

Transformation: Symbolic columns which are protocol, service, flag and label must be converted to a numeric values using a customization transformation table, in order to be in an appropriate format before entering the classification phase.

Dividing: after transformation, the dataset which it is used to train the neural network is divided into 3 subsets: training, validation and testing. Training subset is used to tune the weights of the connections, validation subset is used to find out how the net would perform on data it has never been seen, while testing subsets is used to stop the training process. Using random division the main subset is divided where: 70% is the training subset, validation subset is 15% and testing subset is 15%.

Standardization: means subtracting a measure of location and dividing by a measure of scale. Means that subtract the mean and divide by the standard deviation, so the training subset matrix is processed by mapping each row's means to 0 and standard deviations to 1. The mean and variance of the training subset are applied to the validation and testing subsets. It’s important to mention that the main testing dataset also should be standardized before performing the simulation. The standardization can be done using the Matlab function mapstd.

3.2 Training Enhanced Resilient Backpropagation Artificial Neural Network Stage

Riedmiller and Braun [9] defined RPROP which stands for 'resilient propagation' as an efficient new learning scheme, which performs a direct adaptation of the weight step based on local gradient information. The main difference to the ordinary backpropagation is that the effort of adaptation is not blurred by gradient behavior whatsoever, it only depends on the sign of the derivative not its value and therefore it will converge from ten to one hundred times faster than the simple backpropagation algorithms.

Many algorithms have been proposed to deal with the problem of the ideal weight-update by performing some parameter adaption during the learning process. The RPROP deals with the local adaption instead of the global adaption, where it uses only the partial derivative of the error.

3.2.1 Enhanced Resilient Propagation (ERBP)

The general learning rule formula is identified as [13]:

$$w^{(m+1)} = w^m + \xi(t^m - d^m)z^m$$

Where

$w^{(m+1)}$ *is the new weight,*
w^m *is the previous weight,*
ξ *is a positive learning factor,*
t^m *is the target(desired)output*
d^m*is the neural output*
and finally z^m *is a training pattern*

In order to improve the convergence speed where the neural network will settle in the global minima instead the local, the above equation is improved by defining the optimal value for the learning factor. The optimal factor is derived from the learning rule, where assuming w^* is the correct weight solution [13]:

$$w^{(m+1)} - w^* = w^m - w^* + \xi\,(t^m - d^m)z^m$$

Now if z^m is correctly classifed there is no need to update the weights, but if z^m is misclassified, then:

$$\left\|w^{(m+1)} - w^*\right\|^2 = \|w^m - w^*\|^2 + \xi^2\|z^m\|^2 + 2\xi(t^m - d^m)(w^m - w^*)z^m$$

Where $\|.\|$ is any norm and $(t^m - d^m)^2$=1, because when target is one the neural output will be zero and vice versa. The target and the neural output will never be equal because we assumed from the beginning that z^m is missclassified.

Now it can be shown that:

$$(t^m - d^m)(w^*)^T z^m = |(w^*)^T z^m| \geq 0$$

$$\boldsymbol{Reinforcement\ Learning}$$

and

$$(t^m - d^m)(w^m)^T z^m = -|(w^m)^T z^m| \leq 0$$

$$\boldsymbol{Anti - Reinforced\ Learning}$$

Then substitute the above two formulas in the main equation, we have:

$$\left\|w^{(m+1)} - w^*\right\|^2 = \|w^m - w^*\|^2 + \xi^2\|z^m\|^2 - 2\xi(|(w^*)^T z^m| + |(w^m)^T z^m|)$$

Then the optimal step size can be derived by minimizing the mean square error (MSE), where $\left\|w^{(m+1)} - w^m\right\| \to 0$ over ξ_{opt}:

$$\xi_{opt} = \frac{|(w^*)^T z^m| + |(w^m)^T z^m|}{\|z^m\|^2}$$

The finite convergence is guaranteed because the total squared error always decreases with each update. But in case of misclassification the learning rule becomes:

$$w^{(m+1)} = w^m + \frac{(w^* - w^m)^T z^m}{\|z^m\|^2} z^m$$

But since w^* is not known then we can't use ξ_{opt} directly we need to use a relaxation method where the unknown term $(w^*)^T z^m$ is substituted by a lower bound lets say $\delta, 0 \leq \delta \leq \delta^*$ where:

$$\delta^* = min_m |w^{*T} z^m|$$

Thus leading to the well known relaxation method:

$$w^{(m+1)} = w^m + \frac{(t^m - d^m)(\delta + |w^{mT} z^m|)}{\|z^m\|^2} z^m$$

Using ξ_{opt}, the pseudo code for the enhanced resilient propagation becomes [9, 13]:

```
$For\ all\ weights\ and\ biases\{$
$if\left(\frac{\partial E}{\partial w}(m-1) * \frac{\partial E}{\partial w}(m) > 0\right) then\ \{$
$\Delta(m) = minimum(\Delta(m-1) * \eta^+, \Delta_{\max})$
$\Delta w(m) = -sign\left(\frac{\partial E}{\partial w}(m)\right) * \Delta(m)$
$w(m+1) = w(m) + \xi_{opt}\Delta w(m)$
}
$else\ if\ \left(\frac{\partial E}{\partial w}(m-1) * \frac{\partial E}{\partial w}(m) < 0\right) then\ \{$
$\Delta(m) = maximum(\Delta(m-1) * \eta^-, \Delta_{\min})$
$w(m+1) = w(m) - \xi_{opt}\Delta w(m-1)$
$\frac{\partial E}{\partial w}(m) = 0$
}
$else\ if\ \left(\frac{\partial E}{\partial w}(m-1) * \frac{\partial E}{\partial w}(m) = 0\right)\ then\ \{$
 $\Delta w(m) = -sign\left(\frac{\partial E}{\partial w}(m)\right) * \Delta(m)$
$w(m+1) = w(m) + \xi_{opt}\Delta w(m)$
}
}
```

In the training phase of the enhanced resilient backpropagation neural network, different computational intelligence paradigms were constructed using the training dataset to give maximum generalization accuracy on the unseen data. First of all we started with only one hidden layer using ten neurons, then we incremented the units by 2 neurons and repeated the training process. We have used the iterative process because using high number of hidden neurons will lead to over-fitting problem, where the neural will not be able to classify new records. Generally if the there are no good results then a second layer can be added to improve the neural performance. In our experiments we only needed one hidden layer with 26 hidden neurons. The optimum number of hidden neurons was selected iteratively where [16]:

I (input dimensionality) =41, O (classes) =5 and Ntrn (training vectors) = 2471

Neq (Number of output equations) = Ntrn*O=2471*5= 12355

Hmax1=floor ((Neq-O)/(I+O+1))=floor((12355-5)/(41+5+1))=262

To select the optimum number of hidden neurons Neq>>Nw, where:

Nw (Number of unknown weights) = (I+1)*H+ (1+H)*O; where H is the number of hidden neurons.

Suppose that Neq>r*Nw, where r=10, then:
Hmax10=round ((Neq/r-O)/(O+I+1)) =26

Therefore using H=26, Neq is really large than Nw:
Nw = (41+1)*26+(1+26)*5=1227
12355>>1227

According to the above equations the maximum number of hidden neurons is 26, therefore we started the training process with 10 incremented by 2 neurons ending with the maximum which is 26. The results have shown that the optimum number of hidden neurons was 26.
The neural network architecture consists of 41 neurons in the input layer, 26 hidden neurons in the hidden layer and 5 neurons in the output layer. The neural network was trained by adjusting the weights until the error between the desired output and the neural output is below some predefined value (e^{-6}). Mean Square Error (MSE) will be used to find the norm between the desired output and the neural output.

3.3 Testing Stage

In this stage, testing dataset will be classified by the enhanced resilient backpropagation neural network. The designed system will be evaluated by calculating the Detection Rate (DR), Accuracy Rate, False Positive Rate (FPR), False Negative Rate (FNR), Recall Rate (NPV) and Precision Rate (PPV).
A false-positive occurs when the system classifies an action as anomalous (a possible intrusion) when it is a legitimate action. Although this type of error may not be completely eliminated, a good system should minimize its occurrence to provide useful information to the users. A false-negative occurs when an actual intrusive action has occurred but the system allows it to pass as non-intrusive behavior. While the true-positives (TP) and true-negatives (TN) are correct classifications. Recall Rate measures the proportion of actual positives which are correctly identified. Precision Rate is the ratio of true positives to combined true and false positives [12].

Intrusion Detection Evaluation Formulas [15]:

$$Detection\ Rate = number\ of\ corrected\ classified\ /total\ number\ of\ \ records$$

$$Accuracy = \ (TP + TN)/(\mathrm{TP} + \mathrm{TN} + \mathrm{FN} + \mathrm{FP})$$

$$Recall(Sensitivity\ NPV) = TP/(\mathrm{TP} + \mathrm{FN})$$

$$Precision(PPV) = TP/(\mathrm{TP} + \mathrm{FP})$$

$$False\ Positive\ Rate\ (FPR) = FP/(\mathrm{TN} + \mathrm{FP})$$

$$False\ Negative\ Rate\ (FNR) = FN/(\mathrm{TP} + \mathrm{FN})$$

4. Experiments and Results

In this paper an enhanced resilient backpropagation neural network was trained to detect intrusions using NSL-KDD99 dataset. Training dataset was used to tune the weights and testing dataset was used for the network evaluation. Testing set contains some attacks that it is not represented in the training set. The testing dataset (Labeled and Unlabeled) details are shown in the tables below:

Table: 1 Testing Datasets (Labeled) Analysis Details

Testing Dataset (Labeled)	***Class Size***
Normal	1000
Denial of Service (DoS)	1200
User to Root (U2R)	37
Root to Local (R2L)	1200
Prob	1200
Total	4637

Table: 2 Testing Datasets (Unlabeled) Analysis Details

Testing Dataset (Unlabeled)	***Class Size***
Unknown	3750

An enhanced resilient backpropagation neural network (ERBP) will be used also to classify the testing set into 5 classes. The neural network was trained using the following parameters:

Table: 3 Enhanced Resilient Artificial Neural Network Parameters

Parameters	***Details***
Learning	Supervised
Input Layer	One input layer with 41 neurons (input dimensionality)
Hidden Layer	One hidden layer with 26 neurons
Output Layer	One output layer with 5 neurons (Classes)
Number of epochs	203
Transfer Function	Hyperbolic tangent sigmoid (tansig) and Log-sigmoid (logsig)
Network Performance	Mean Square Error (MSE)

As mentioned in the training stage section, the number of hidden neurons was selected carefully and precisely according to the confusion matrix results, therefore 26 hidden neurons were selected. The classification rate for different number of hidden neurons is described in the following table:

Table: 4 Number of Hidden Neurons vs. Detection Rate

Hidden Neurons	***Detection Rate***
10	90.3%
12	90.1%
14	92%
16	92.8%
18	91.7%
20	93.3%
22	93.5%
24	91.4%
26	94.7%
28	92.7%
30	93.9%
32	91.7%

Using the enhanced resilient backpropagation has improved the performance of the system in terms of classification rate and number of epochs in comparison with the ordinary resilient backpropagation. The table below demonstrates the difference between them:

Table: 5 Enhanced resilient backpropagation vs. Ordinary resilient backpropagation

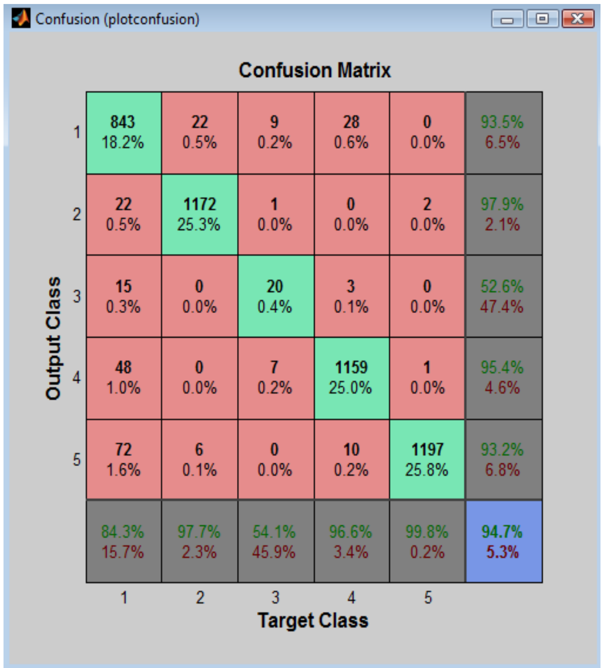
Algorithm	***Detection Rate***	***Epochs***
Ordinary resilient backpropagation	94.5%	244
Enhanced resilient backpropagation	94.7%	203

Enhanced resilient backpropagation neural network (ERBP) was able to classify the testing dataset labeled and unlabeled as follows:

Table: 6 Labeled Testing dataset Results

Testing(Labeled) Datasets	***Class Size***	***Detected Size***	***Detection Rate***
Normal	1000	843	84.3%
DoS	1200	1172	97.7%
U2R	37	20	54.1%
R2L	1200	1159	96.6%
Prob.	1200	1197	99.8%
Total	4637	4391	94.7%

Classifiers are best judged by the distribution of classification error rates in the confusion matrix. The following figure represents the confusion matrix of the enhanced resilient backpropagation.



Confusion Matrix

	1	2	3	4	5	
1	843 18.2%	22 0.5%	9 0.2%	28 0.6%	0 0.0%	93.5% 6.5%
2	22 0.5%	1172 25.3%	1 0.0%	0 0.0%	2 0.0%	97.9% 2.1%
3	15 0.3%	0 0.0%	20 0.4%	3 0.1%	0 0.0%	52.6% 47.4%
4	48 1.0%	0 0.0%	7 0.2%	1159 25.0%	1 0.0%	95.4% 4.6%
5	72 1.6%	6 0.1%	0 0.0%	10 0.2%	1197 25.8%	93.2% 6.8%
	84.3% 15.7%	97.7% 2.3%	54.1% 45.9%	96.6% 3.4%	99.8% 0.2%	94.7% 5.3%

Fig. 2 Enhanced Resilient Backpropagation Neural Network Confusion Matrix

The system was able to classify the unlabeled test dataset with a detection rate about 89%:

Table: 7 Unlabeled Testing dataset Results

Testing (Unlabeled) Datasets	***Class Size***	***Detected Size***	***Detection Rate***
Unknown attacks	3750	3340	89%

Finally the system performance is compared to other intrusion detection systems that use either neural network (supervised, unsupervised) or Iterative Dichotomiser3 (ID3) which it's a decision tree method.

Table: 8 Intrusion Detection System Evaluation Rates vs. Other systems

Method	***DR***	***AR***	***NPV***	***PPV***	***FPR***	***FNR***
ERBP	94.7 %	95.3 %	98.4 %	95.8 %	15.7 %	1.6%
ART1 [14]	71.1 %	97.4 %	-	-	1.9%	0.5%
SOM [14]	83.44 %	95.7 %	-	-	3.5%	0.7%
ID3 [15]	-	99%	98%	-	-	-

5. Conclusion

In this paper an intrusion detection system was designed. The proposed system classifies intrusions using an enhanced resilient backpropagation neural network. The enhanced resilient backpropagation was able to classify the records into 5 classes with a reasonable good detection rate about 94.7% and with a false positive rate of 15.7%. As noticed from table 8 supervised learning (ERBP & ID3) and unsupervised learning (ART1 & SOM) have close results in terms of Accuracy Rate and Recall Rate. False Positive Rate was less using unsupervised SOM & ART1 than supervised ERBP. Detection rate for the ERBP was greater than unsupervised ART1 & SOM. The conclusion of this comparison is that the intrusion detection system can be designed with high detection, accuracy, recall and precision rates while maintaining low false negative and false positive rates, if hybrid system of supervised models or supervised and unsupervised models is used. The power of integrating models is that the system will combine the power of the combined models, therefore designing a very powerful and reliable intrusion detection system.

During experiments, User to Root attack always suffers from low detection rate due to, there are a few records in the training dataset and this usually will lead the classifiers, especially neural networks to converge to the other attacks.

References

[1] Abraham, A, Grosan, C, & Chen, Y. (2006). Evolution of Intrusion Detection Systems. Retrieved October 18, 2011,from http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.161.5620

[2] Cannady, J. (1998). Artificial Neural Networks for Misuse Detection. National Information Systems Security Conference. Retrieved October 18, 2011, from http://citeseer.ist.psu.edu/viewdoc/summary?doi=10.1.1.39.5179

[3] Ali, A, Saleh, A & Badawy, T. (2010). Intelligent Adaptive Intrusion Detection Systems Using Neural Networks (Comparative study). International Journal of Video & Image Processing and Network Security IJVIPNS-IJENS, 10 (01). Retrieved October 25, 2011, from http://www.ijens.org/101701-6363%20IJVIPNS-IJENS.pdf

[4] Al-Rashdan, W. (2011). A Hybrid Artificial Neural Network Model (Hopfield-SOM with Conscience) for Effective Network Intrusion Detection System. (Doctoral dissertation, The Arab Academy for Banking and Financial Sciences, 2011).

[5] Information Security Center of Excellence (ISCX). (2009). The NSL-KDD Data Set. Retrieved October 26, 2011, from http://www.iscx.ca/NSL-KDD/

[6] Kotulski, Z & Kukiełka, P. (2008). Analysis of Different Architectures of Neural Networks for Application in Intrusion Detection Systems. Proceedings of the International Multi conference on Computer Science and Information Technology, 807 – 811. Retrieved October 25, 2011, from http://www.proceedings2008.imcsit.org/pliks/197.pdf

[7] Li, J, Zhang, G & Gu, U. (2004). The Research and Implementation of Intelligent Intrusion Detection System Based on Artificial Neural Network. Proceedings of the Third International Conference on Machine Laming and Cybernetics, Shanghai. Retrieved October 18, 2011, from http://svn.assembla.com/svn/odinIDS/.../01378582.pdf

[8] Moradi, M & Zulkernine, M. (n.d.). A Neural Network Based System for Intrusion Detection and Classification of Attacks. University of British Columbia. Retrieved October 18, 2011, from http://research.cs.queensu.ca/~moradi/148-04-MM-MZ.pdf

[9] Riedmiller, M & Braun, H. (1993). A Direct Adaptive Method for Faster Backpropagation Learning: The RPROP Algorithm. IEEE INTERNATIONAL CONFERENCE ON NEURAL NETWORKS. Retrieved October 31, 2011, from http://paginas.fe.up.pt/~ee02162/dissertacao/RPROP%20paper.pdf.

[10] Sammany, M, Sharawi, M, El-Beltagy, M & Saroit, I. (2007). Artificial Neural Networks Architecture For Intrusion Detection Systems and Classification of Attacks. Faculty of Computers and Information Cairo University. Retrieved October 18, 2011, from http://infos2007.fci.cu.edu.eg/Computational%20Intelligence/07177.pdf

[11] Tavallaee, M, Bagheri, E, Lu, W & Ghorbani, A. (2009). A Detailed Analysis of the KDD CUP 99 Data Set. Proceedings of the 2009 IEEE Symposium on Computational Intelligence in Security and Defense Applications (CISDA 2009). Retrieved October 25,2011, from http://www.tavallaee.com/publications/CISDA.pdf

[12] AL-Rashdan, W, Naoum, R, Al_Sharafat, W & Al-Khazaaleh, M. (2010). Novel Network Intrusion Detection System using Hybrid Neural Network (Hopfield and Kohonen SOM with Conscience Function). IJCSNS International Journal of Computer Science and Network Security, 10(11). Retrieved January 26, 2012, from http://paper.ijcsns.org/07_book/201011/20101103.pdf

[13] Naoum, R. (2011). Artificial Neural Network [Acrobat Reader], Middle East University (MEU), Jordan

[14] Amini, M, Jalili, R & Shahriari, H. (2006). RT-UNID: A practical solution to real-time network-based intrusion detection using unsupervised neural networks. Computers & Security Journal, ELSEVIER, 459-468. Retrieved October 20, 2011, from

http://techlab.bu.edu/files/resources/articles_tt/RT-UNNID,%20A%20practical%20solution%20to%20real-time.pdf

[15] Revathi, M & Ramesh, T. (2011). NETWORK INTRUSION DETECTION SYSTEM USING REDUCED DIMENSIONALITY. Indian Journal of Computer Science and Engineering (IJCSE). 2(1), 61-67. Retrieved January 29, 2012, from http://www.ijcse.com/docs/IJCSE11-02-01-056.pdf

[16] Heath, G. (2011, December 28). Validation set and Parameters in Backpropagation ANN, [Newsreader]. December 28, 2011, Answer posted to http://www.mathworks.com/matlabcentral/newsreader/view_thread/315487#865508