HYBRID SYSTEM OF LEARNING VECTOR QUANTIZATION AND ENHANCED RESILIENT BACKPROPAGATION ARTIFICIAL NEURAL NETWORK FOR INTRUSION CLASSIFICATION

Reyadh Shaker Naoum[1] & Zainab Namh Al-Sultani[2]
[1]Middle East University, Jordan
[2]Baghdad, Iraq

ABSTRACT

Network-based computer systems play increasingly vital roles in modern society; they have become the target of intrusions by our enemies and criminals. Intrusion detection system attempts to detect computer attacks by examining various data records observed in processes on the network. This paper presents a hybrid intrusion detection system models, using Learning Vector Quantization and an enhanced resilient backpropagation artificial neural network. The proposed system is divided into five phases: environment phase, dataset features and pre-processing phase, Learning Vector Quantization phase, enhanced resilient backpropagation neural network phase and testing the hybrid system phase. A Supervised Learning Vector Quantization (LVQ) as the first stage of classification was trained to detect intrusions; it consists of two layers with two different transfer functions, competitive and linear. A multilayer perceptron as the second stage of classification was trained using an enhanced resilient backpropagation training algorithm. Best number of hidden layers and hidden neurons were calculated to train the enhanced resilient backpropagation neural network. One hidden layer with 32 hidden neurons was used in resilient backpropagation artificial neural network training process. An optimal learning factor was derived to speed up the convergence of the resilient backpropagation neural network performance. The evaluations were performed using the NSL-KDD99 network anomaly intrusion detection dataset. The experiments results demonstrate that the proposed system (LVQ_ERBP) has a detection rate about 97.06% with a false negative rate of 2%.

Keywords: *Intrusion Detection System, Learning Vector Quantization, Resilient Backpropagation, Artificial Neural Network*

1. INTRODUCTION

The importance of protecting systems from attacks and intrusions is critical, especially with the coming of Internet age, and because of the increasing dependence which companies and government agencies have on their computer networks [1]. A single intrusion of a computer network can result in the loss or unauthorized utilization or modification of large amounts of data and cause users to question the confidentiality, reliability and the availability of all of the information and the resources on the network. Intrusion Detection Systems have become the key foundation of network security.

The two main intrusion detection techniques are misuse detection and anomaly detection. Misuse detection systems, use patterns of well known attacks or weak spots of the system to match and identify known intrusions. Anomaly detection systems, flag observed activities that deviate significantly from the established normal usage profiles as anomalies, that is, possible intrusions. Anomaly detection techniques can be effective against unknown or novel attacks since no a priori knowledge about specific intrusions is required. However, anomaly detection systems tend to generate more false alarms than misuse detection systems because an anomaly can just be a new normal behavior [2].

Neural networks are a uniquely powerful tool in multiple class classification, especially when used in applications where formal analysis would be very difficult or even impossible, such as pattern recognition, nonlinear system identification, and control [3]. 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 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 [4].

2. RELATED WORKS

Depren et al. (2005) [5] proposed an intelligent intrusion detection system for anomaly detection system using Self Organizing Map (SOM) to model the normal behavior. They used KDD Cup99 data set for implementation. Their results showed that their module achieved an accuracy rate of 98.96% (2 Classes) a false positive rate of 1.01%. The main advantage of this method is using the powerful unsupervised SOM which results a low false positive rate, but on the other hand the system didn't classify the records into 5 classes. Ahmad, Swati & Mohsin (2007) [6] designed an intrusion detection mechanism using resilient backpropagation. The ANN architecture was input and output layers and two hidden layer, with 41, 14, 9, and 2 neurons respectively. KDD Cup 99 was used as the dataset that contains both training and testing sets. They achieved an accuracy rate of 95.93% (2 Classes). The proposed system had a very good accuracy rate but on the other hand they have used 2 hidden layers which are not necessary especially if the neural network parameters were selected optimally. Naoum, Abid and Al-Sultani (2012) [7] proposed a hybrid intrusion detection system based on k-Nearest Neighbor and an enhanced resilient backpropagation artificial neural network. An optimal learning factor was derived to enhance the performance (speed up the convergence) of the enhanced resilient backpropagation. First Norm was used in the k-Nearest Neighbor implementation instead of Euclidean distance and they have used the first nearest neighbor where k equals 1. The enhanced resilient backpropagation neural network trained using an optimal number of hidden layers and neurons; therefore it was trained with only one hidden layer and 34 hidden neurons. The evaluation was performed on the NSL-KDD99 anomaly intrusion detection dataset. The proposed system has a classification rate (5 classes) of 97.2% with false negative rate of about 1%.

3. PROPOSED SYSTEM

This paper presents two classifiers to classify intrusions, using Learning Vector Quantization and a multi layer perceptron trained using an enhanced resilient backpropagation as the first and second classifier respectively. The system will be tested and evaluated using the NSL-KDD dataset. The proposed system is divided into five phases: environment phase, dataset features and pre-processing phase, Learning Vector Quantization phase, enhanced resilient backpropagation neural network phase and testing the hybrid system phase.

3.1 The Environment Phase

This unit presents records from NSL KDD99 dataset [8]. It's divided into two subsets, training subset and testing subset. The NSL KDD dataset includes a wide variety of intrusions together with normal activities simulated in a military network environment. NSL KDD records belong to one of the following five categories: Normal, DoS (denial of service), R2L (root to local), U2R (user to root) and Probing (surveillance). There are 41 features columns and they are either symbolic or continuous.

3.2 Data Pre-processing Phase

The data from the environment unit will be processed before entering the classification unit. Feature columns are processed at 2 steps as follows:

1. Transformation: Symbolic columns are transformed to numeric values using transformation table for each column. Table 1 demonstrate the customize transformation table for flag feature column.

Table 1 Flag Column Feature Transformation Table

Flag-4	No
OTH	1
REJ	2
RSTO	3
RSTO 0	4
RSTR	5
S0	6
S1	7
S2	8
S3	9
SF	10
SH	11

Label column (column 42) contains either normal or the sub-type attack label. Transforming this column was applied at two steps. First the sub-attack type was represented with the main attack type, and then the main attack type was transformed to numeric using 5 columns, each class is represented with value one using one column. Table 2 represents the customization transformation for the main classes.

Table 2 Label Transformation Table

Label -42	Column1	Column2	Coulmn3	Column4	Column5
Normal	**1**	**0**	**0**	**0**	**0**
DoS	**0**	**1**	**0**	**0**	**0**
U2R	**0**	**0**	**1**	**0**	**0**
R2L	**0**	**0**	**0**	**1**	**0**
Prob.	**0**	**0**	**0**	**0**	**1**

2. Standardization: Training subset matrix is processed by mapping each row's means to 0 and standard deviations to 1. It’s important to mention that the main testing dataset also should be standardized using the mean and the variance of the training dataset before performing the simulation.

3.3 Learning Vector Quantization Phase

In this paper a Learning Vector Quantization was trained to detect intrusions as the first step. Learning vector quantization (LVQ) is a method for training competitive layers in a supervised manner. LVQ network has a first competitive layer and a second linear layer. The competitive layer learns to classify input vectors in much the same way as the competitive layers of Self-Organizing Feature Maps. The linear layer transforms the competitive layer's classes into target classifications defined by the user. The classes learned by the competitive layer are referred to as subclasses and the classes of the linear layer as target classes [9].

In the training process of LVQ different computational paradigms were used. First we have mentioned before that LVQ only consists of one competitive (hidden) layer and one output layer containing sub-class and main-class neurons respectively. Therefore in the competitive and output layers 23 (normal and 22 sub attack types) and 5 neurons were used respectively.

During the training of LVQ it was clearly that LVQ highly affected about how many patterns corresponding for each main class, therefore in designing the training dataset it was very critical to select approximately equal numbers of patterns to represent each class otherwise the LVQ will classify one main class and neglects the others.

Learning rate and number of epochs was selected iteratively where the best performance is the main criteria.

After the training process, the LVQ is ready to classify the testing dataset. LVQ will classify the dataset into five classes (Normal, DoS, U2R, R2L and Prob). Then the results of LVQ will be combined later with the results of the neural network trained using the enhanced resilient backpropagation classifier to provide maximum classification rates.

3.4 Enhanced Resilient Backpropagation Neural Network Phase

The enhanced resilient backpropagation neural network will be trained by adjusting the weights until the error between the desired output and the neural output is below some predefined value (e.g. e^{-10}). Mean Square Error (MSE) will be used to find the norm between the desired output and the neural output. The learning process is essentially an optimization process in which the parameters of the best set of connection coefficients (weights) for solving a problem are found [10]. It is very difficult to know which training algorithm will be the fastest for a given problem. Our experiments have shown that the resilient backpropagation (trainrp) may be the fastest on detecting intrusions, and the memory requirements for this algorithm are relatively small in comparison to the other algorithms considered. In order to improve the convergence speed of the resilient backpropagation an optimal learning factor parameter was derived, and the algorithm was enhanced and it is called the enhanced resilient backpropagation.

The general learning rule formula is identified as:

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

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*

The optimal weight $\mathbf{w}^*$ which is the correct weight solution will be used to improve the convergence speed where the neural network will settle in the global minima instead the local. Using the above equation $\mathbf{w}^*$ is subtracted at both sides of the equation. The learning rule, where assuming $\mathbf{w}^*$ is the correct weight solution becomes:

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

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 $\| \mathbf{.} \|$ can be any norm, however in this research 1- norm was used. The term $(\mathrm{t}^{\mathrm{m}} - \mathrm{d}^{\mathrm{m}})^2$ equals 1, because when the 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:

$$(t^m - d^m)(w^*)^T z^m = |(w^*)^T z^m| \geq 0 \qquad \boldsymbol{Reinforcement\,Learning}$$
$$and$$
$$(t^m - d^m)(w^m)^T z^m = -|(w^m)^T z^m| \leq 0 \qquad \boldsymbol{Anti-Reinforced\,Learning}$$

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}:

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

Substitute ξ_{opt} in the learning rule equation:

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

We can't use ξ_{opt} since the optimal weight value can't be determined in advance. Therefore a relaxation method will be used by replacing the unknown term $(\mathrm{w}^*)^{\mathrm{T}}\mathrm{z}^{\mathrm{m}}$ by δ, where $0 \leq \delta \leq \delta^*$: $\delta^* = min_m|w^{*T}z^m|$
Thus the learning rule becomes:

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

To provide maximum generalization, we started with only one hidden layer using different number of hidden neurons iteratively. We used the iterative process because using high number of hidden neurons will lead to over-fitting problem, where the neural network will not be able to classify new records. Generally if there are no good results then a second layer can be added to improve the neural performance.
Experiments have shown that when using only one hidden layer with 32 hidden neurons, the enhanced resilient backpropagation neural performance gave the best classification rate.

3.5 Testing the hybrid system (LVQ_ERBP) Phase

In this phase, testing dataset will be classified by both Learning Vector Quantization and the enhanced resilient backpropagation neural network which was trained during the training phase using the best number of hidden neurons and layers. The designed system will be evaluated by calculating the Detection Rate (DR), False Positive Rate (FPR) etc.

4. EXPERIMENTS RESULTS

In this paper a hybrid system of learning vector quantization and an enhanced resilient backpropagation neural network was trained to detect intrusions using NSL-KDD99 dataset. Testing set contains some attacks that it is not represented in the training set. The testing dataset details are shown in the table 3:

Table 3 Testing Datasets (Labeled) Analysis Details

Testing Dataset (Labeled)	***Class Size***
Normal	1000
Denial of Service (DoS)	2200
User to Root (U2R)	37
Root to Local (R2L)	2200
Prob	2200
Total	7637

LVQ network as the first classifier was trained using the parameters shown in the table 4:

Table 4 Learning Vector Quantization Learning Parameter

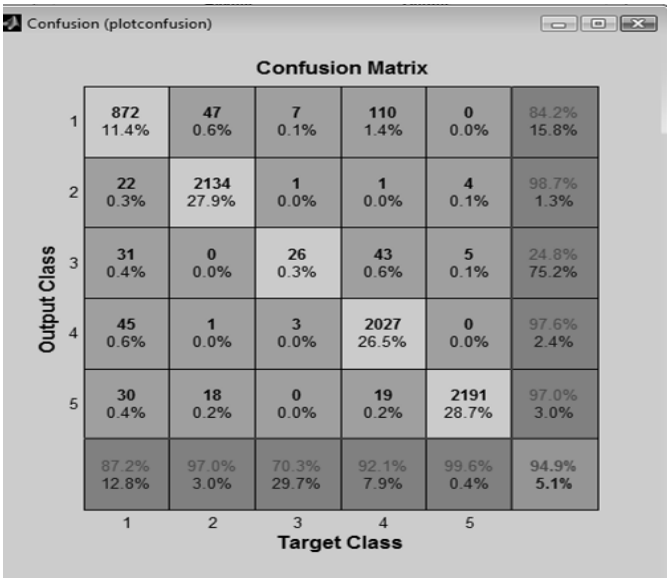
Parameters	**Details**
Learning	Supervised
Input Nodes	Input Dimensionality: 41
Hidden Nodes	Number of sub-classes: 23
Output Nodes	Number of Main classes: 5
Distance Function	Negative Euclidean Distance (**negdist**)
Transfer Function in the Hidden Layer	Competitive Transfer Function (**compet**)
Transfer Function in the output layer	Linear Transfer Function (**purelin**)
Learning Function	Learning Vector Quantization 1
Training Function	Random Weight/Bias Rule
Learning Rate	0.008
Number of epochs	6
Network Performance	Mean Square Error (**MSE**)

An enhanced resilient backpropagation neural network (ERBP) as the second classifier will be used also to classify the testing dataset into 5 classes. The neural network was trained using the parameters shown in table 5:

Table 5 Enhanced Resilient Artificial Neural Network Learning Parameters

Parameters	***Details***
Learning	Supervised
Input Layer	One input layer with 41 neurons (input dimensionality)
Hidden Layer	One hidden layer with 32 neurons
Output Layer	One output layer with 5 neurons (Classes)
Number of epochs	364
Transfer Function	Hyperbolic tangent sigmoid (tansig)
Network Performance	Mean Square Error (MSE)

Figure below demonstrates the confusion matrix of the enhanced resilient backpropagation neural network as a separated phase:



Output Class \ Target Class	1	2	3	4	5	
1	872 11.4%	47 0.6%	7 0.1%	110 1.4%	0 0.0%	84.2% 15.8%
2	22 0.3%	2134 27.9%	1 0.0%	1 0.0%	4 0.1%	98.7% 1.3%
3	31 0.4%	0 0.0%	26 0.3%	43 0.6%	5 0.1%	24.8% 75.2%
4	45 0.6%	1 0.0%	3 0.0%	2027 26.5%	0 0.0%	97.6% 2.4%
5	30 0.4%	18 0.2%	0 0.0%	19 0.2%	2191 28.7%	97.0% 3.0%
	87.2% 12.8%	97.0% 3.0%	70.3% 29.7%	92.1% 7.9%	99.6% 0.4%	94.9% 5.1%

Figure 1Enhanced Resilient Backpropagation Neural Network Confusion Matrix

After combining the results of both classifiers the class detection rate of the hybrid (LVQ_ERBP) is shown in table 6:

Table 6 Hybrid (LVQ_ERBP) Detection Rate

Testing(Labeled) Datasets	***Class Size***	***Detected Size***	***Detection Rate***
Normal	1000	910	91%
DoS	2200	2165	98.4%
U2R	37	26	70.27%
R2L	2200	2121	96.4%
Prob.	2200	2191	99.59%
Total	7637	7413	97.06%

False Positive Rate, False Negative Rate, Recall, and Precision metrics are used to evaluate the performance of learning algorithms. These metrics have been widely used for comparisons. Table 7 shows the values of these metrics for the hybrid system (LVQ_ERBP):

Table 7 Hybrid System (LVQ_ERBP) Evaluation Metrics

Testing(Labeled) Datasets	***Percentage***
Recall	98%
Precision	99%
False Negative Rate	2%
False Positive Rate	9%

Table 8 shows the comparison between the proposed hybrid system (LVQ_ERBP) and the hybrid system (LVQ_kNN) proposed by [7].

Table 8 Hybrid (LVQ_ERBP) vs. Hybrid (LVQ_kNN)

Dataset	**Hybrid (LVQ_ERBP)**	**%**	**Hybrid (LVQ_kNN)**	**%**
Normal	910 - 1000	91%	938 – 1000	94%
DoS	2165 – 2200	98.4%	1187 -1200	99%
U2R	26 – 37	70.3%	30 – 37	81%
R2L	2121 – 2200	96.4%	194 – 500	39%
Prob.	2191 – 2200	99.59%	1157 – 1200	96%
All	7425 - 7637	97.06%	3506 - 3937	89%

5. CONCLUSION

In this paper a Learning Vector Quantization and an enhanced resilient backpropagation artificial neural network were trained to detect intrusion. The main issue in LVQ training, it requires a long time to be trained especially when comparing to other networks such as Multilayer Perceptron or Self Organizing Maps. Table 8 shows that the hybrid system of (LVQ_ERBP) had better results than (LVQ_kNN). Artificial neural network as a machine learning algorithm has a better performance in classification problems than the k-Nearest Neighbor which uses norm distance to classify the records. One of the main issues in detecting intrusions is the low-frequent attacks. User to Root as a low-frequent has the lowest detection rate among other classes. For low-frequent attacks, the leaning sample size is too small compared to high-frequent attacks, it makes ANN not easy to learn the characters of these attacks and therefore detection precision is much lower. In practice, low-frequent attacks do not mean they are unimportant. Instead, serious consequence will be caused if these attacks succeeded. For example, if the U2R attacks succeeded, the attacker can get the authority of root user and do everything he likes to the targeted computer systems or network device [11].

6. REFERENCES

[1] 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

[2] LEE, W. & STOLFO, S (2000). A Framework for Constructing Features and Models for Intrusion Detection Systems. ACM Transactions on Information and System Security, 3 (4).

[3] 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

[4] Kukiełka, P. & Kotulski, Z. (2010). Adaptation of the neural network-based IDS to new attacks detection. arXiv - Cornell University. Retrieved October 26, 2011, from http://arxiv.org/ftp/arxiv/papers/1009/1009.2406.pdf

[5] Depren, O., Murat, T., Anarim, E. & Ciliz, M. (2005). An intelligent intrusion detection system (IDS) for anomaly and misuse detection in computer networks. *Expert Systems with Applications, Elsevier*,713-722. Retrieved October 20, 2011, from http://www.ft.unicamp.br/RedesComplexas/downloads/An_intelligent_intrusion_detection_system_for_anomaly_and_misuse_detection_in_computer_networks.pdf

[6] Ahmad, I., Swati, S. & Mohsin, S. (2007). Intrusions Detection Mechanism by Resilient Back Propagation (RPROP). *European Journal of Scientific Research, EuroJournals Publishing, Inc*, *17*, 523-531. Retrieved November 2, 2011, from http://www.eurojournals.com/ejsr%2017%204.pdf

[7] Naoum,R. Abid,N. & Al-Sultani,Z. (2012). A Hybrid Intrusion Detection System Based on Enhanced Resilient Backpropagation Artificial Neural Network and K-Nearest Neighbor Classifier. International Journal of Academic Research IJAR, 4 (2). Retrieved from http://www.ijar.lit.az/en.php?go=march2012

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

[9] MathWorks Matlab Help, "Learning Vector Quantization Networks", 2011.

[10] 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

[11] Wang. G., Hao, J., Ma,J. and Huang, L. (2010). A new approach to intrusion detection using Artificial Neural Networks and fuzzy clustering Gang. Expert Systems with Applications Journal, Elsevier