J. Basic. Appl. Sci. Res., 2(1)672-679, 2012
© 2012, TextRoad Publication
ISSN 2090-4304
Journal of Basic and Applied Scientific Research
www.textroad.com

A new Approach for Detecting Intrusions Based on the PCA Neural Networks

Adel Jahanbani, Hossein Karimi

Department of computer, Lamerd Branch, Islamic Azad University, Lamerd, Iran

ABSTRACT

Intrusion Detection System (IDS) is an effective tool that can help to prevent unauthorized access to network resources. A good intrusion detection system should have higher detection rate and lower false positive. A new classification system using Principal Component Analysis (PCA) neural networks for ID is proposed to detect intrusions from normal connections with satisfactory detection rate and false positive. Experiments and evaluations were performed with the KDD Cup 99 intrusion detection database. Comparison with other approach based on different evaluation parameters showed that proposed approach has noticeable performance with detection rate 99.596% and false positive 0.404% and can classify the network connections with satisfactory performance.

KEY WORDS: Intrusion detection; PCA neural network; Detection rate; Neural Network; KDD Cup 99.

INTRODUCTION

Heavy reliance networked computer resources and the increasing connectivity of these networks has greatly increased the potential damage that can be caused by attacks launched against computers from remote sources. These attacks are difficult to prevent with firewalls, security policies, or other mechanisms because system and application software is changing at a rapid pace, and this rapid pace often leads to software that contains unknown weaknesses or bugs. Intrusion detection systems are designed to detect those attacks that inevitably occur despite security precautions (Lippmann et al., 1999). Intrusion detection is the process of determining an intrusion into a system by the observation of the information available about the state of the system and monitoring the user activities. An intrusion detection system or IDS is any hardware, software or combination of both that monitors a system or network of systems for a security violation (Koziol, 2003). On the other hand, IDS is a system for raising attention towards potential misbehaviors of the system caused by external adversaries (Crescenzo, 2005). IDSs are classified into two types: misuse detection and anomaly detection. Misuse detection is a method in which intrusion pattern is hand-coded using expert knowledge for well-known attacks of the system, then through matching and identifying these known intrusions with patterns, intrusions of system are detected. Misuse detection has low false alarm because of its nature. But the main shortcomings of misuse detection are: known intrusion patterns have to be hand-coded; it is unable to detect any new or unknown attack that has no matched pattern stored in the system. Anomaly detection assumes that an attack will always reflect some deviation from normal patterns, which is designed to capture any deviations from the established profiles of the system's normal behavior. Anomaly detection can detect new and unknown intrusion, but it has the shortcoming of false alarm rate. A good detection system should have higher detection rate and lower false alarm. In order to detect intruders, many artificial intelligent (AI) techniques, such as rule-based expert system, immune principles, data mining technique and so on, have been applied to intrusion detection (Degang et al., 2007). Moreover, the use of the artificial neural networks in IDS has had a great successful, too. The main work of IDS is to classify normal and intrusive event from the original dataset. Neural networks have proven to be a promising modus operandi for ID. A wide variety of IDSs are using neural networks to address the ID problem. The main reason for using the neural networks in IDS is their generalization ability which makes it suitable to detect unknown attacks. In IDSs, we can use a simple neural network or combination of neural networks or combination of neural networks with other approaches.

The proposed system has different layers and training network is done in two stages. After preprocessing on the original data, these records classify in five different classes. In the first stage there are five PCA neural networks for five classes of records. Every PCA neural network contains only related

***Corresponding Author:** Adel Jahanbani, Department of computer, Lamerd Branch, Islamic Azad University, Lamerd, Iran
E-mail: Jahanbani_adel@yahoo.com

records. Then outputs of this stage is combined and transferred to a PCA as second stage. After these two stages training network finish. Applying two separated stages in training and combination of some neural networks improves the performance in comparison with existing systems.

The rest of the paper is organized as follows. Section 2 provides a background of related work. Section 3 describes KDD Cup 99 database that is used for doing experiments. In section 4 PCA neural network that is used in this work is described. The proposed system architecture is presented in Section 5. Section 6 explains performance evaluation for the comparison this work with other works. Finally, in section 7 and 8 we present experiments and conclusion.

Related works

Among the vast variety of techniques which have been researched for the IDS, the interest on AI techniques and data mining applications have received greater attention particularly the use of unsupervised leaning methods as they have the ability to address some of the shortcomings (Amini et al., 2006). This also helps to achieve the ultimate goal for the IDS i.e. the capability of novelty detection. Recently, the unsupervised learning method (SOM) has represented an excellent performance for sensors work on an unsupervised learning mode (Gunes-Kayacik et al., 2006), as well as it is efficient for real-time intrusion detection (Wang et al., 2006).

In (Jazzar & Jantan, 2008) a new approach that is anomaly based and utilizes causal knowledge inference based on fuzzy cognitive maps (FCM) and multiple self organizing maps (SOM) is represented. Detection rate and false alarm in this approach is 90% and 10.29% respectively.

A novel method using the Fisher's filter to select attributes and attack classification is suggested. This method is applied for four neural networks and results show that detection rate and classification rate are largely improved (Beghdad, 2005).

The techniques that are being investigated include fuzzy logic with network profiling, which uses simple data mining techniques to process the network data. The hybrid system that combines anomaly and misuse detection and using fuzzy rules is reported. The results showed that detection rate for all type of attacks is increased and Attack Accuracy with 99.90% is gained (Shanmugam & Idris, 2007).

A study that analyzes performance of some neural network when all of KDD dataset is used for training in order to classify and detect attacks is reported. The five types of neural networks that are studied: Multi Layer perception (MLP), Self-Organizing Feature Map (SOFM), Jordan / Elman neural network, recurrent neural network and RBF neural network. Their results showed that Percent of Correct Classification (PCC) is 99.16%, 98.28%, 98.36%, 98.44% and 79.23% respectively (Beghdad, 2007).

A serial multi-stage classification system for facing the problem of intrusion detection in computer networks is proposed. This approach offers the advantage that at each stage it is possible to use a set of features tailored for recognizing a specific attack category. Overall error and missed detection in this method is 0.68% and 0.67% respectively (Cordella & Sansone, 2007).

Some of the existing ID models use multi-level structure instead of single-level in order to detect all types of attacks. A hierarchical ID model using principal component analysis (PCA) neural networks is proposed. This method for anomaly detection and misuse detection has a good result in comparison with other methods (Liu et al., 2007). Two hierarchical IDS frameworks using Radial Basis Functions (RBF) are proposed. A serial hierarchical IDS (SHIDS) is proposed to identify misuse attack accurately and anomaly attacks adaptively. A parallel hierarchical IDS (PHIDS) is proposed to enhance the SHIDS's functionalities and performance. The experiments show that the two proposed IDSs can detect network intrusions in real-time, train new classifiers for novel intrusions automatically, and modify their structures adaptively after new classifiers are trained. The experimental results successfully showed that RBF network based IDS have a good performance in misuse detection with a 98% detection rate and a 1.6% false positive detection rate (Zhang et al., 2005). An approach to network intrusion detection is investigated, based purely on a hierarchy of Self-Organizing Feature Maps. This principal interest is to establish just how far such an approach can be taken in practice. In this paper a hierarchical SOM architecture is investigated under two basic feature sets, one is limited to 6 basic features whereas the other contains all 41-features (Kayacik et al., 2005).

In (Moradi & Zulkernine, 2004) a MLP is used for ID based on an off-line analysis approach. Different neural network structures are analyzed to find the optimal network with regards to the number of hidden layers. An early stopping validation method is also applied in the training phase to increase the generalization capability of the neural network. The result shows that the designed system is capable of

classifying records with about 91% accuracy with two hidden layers of neurons in the neural network and 87% accuracy with one hidden layer.

However, in order to refine the process and achieve better detection and performance, extra efforts are required.

The database

The 1998 DARPA Intrusion Detection Evaluation Program was prepared and managed by MIT Lincoln Labs. The objective was to survey and evaluate research in intrusion detection. A standard set of data to be audited, which includes a wide variety of intrusions simulated in a military network environment, was provided. The 1999 KDD intrusion detection contest uses a version of this dataset. A connection is a sequence of TCP packets starting and ending at some well-defined times, between which data flow to and from a source IP address to a target IP address under some well defined protocol. Each connection is labeled as either normal, or as an attack that includes 41 features. These features are in forms of continuous, discrete, and symbolic that fall into four categories (Http://kdd.ics.uci.edu/databases/kddcup99/kddcup):

- Basic Features: Basic features can be derived from packet headers without inspecting the payload.
- Content Features: Domain knowledge is used to assess the payload of the original TCP packets. This includes features such as the number of failed login attempts.
- Time-based Traffic Features: These features are designed to capture properties that mature over the 2 second temporal window. One example of such a feature would be the number of connections to the same host over the 2 second interval.
- Host-based Traffic Features: Utilize a historical window estimated over the number of connections-in this case 100 - instead of time. Host based features are therefore designed to assess attacks, which span intervals longer than 2 seconds.

But the attack types are classified into the following four categories:

- DoS (Denial of Service): making some computing or memory resources too busy to accept legitimate users access theses resources.
- R2L (Remote to Local): unauthorized access from a remote machine in order to exploit machine's vulnerabilities.
- U2R (User to Root): unauthorized access to super user or root functions.
- Probe: surveillance and other probing for vulnerabilities.

KDD database include two groups as training and testing record sets. Training database has about 5 million records. This is very large; for this reason in more investigation works, another training database is applied as 10% training database and we have used this database, too. Distribution of normal and attack types of connection records in 10% training and testing database is shown in TABLE 1.

Table 1. Distribution of normal and attack connection in the KDD database

Class	Train		Test	
	Number	*Percent*	*Number*	*Percent*
Normal	97278	19.69	60593	19.48
DOS	391458	79.24	229853	73.90
Probe	4107	0.83	4166	1.34
U2R	52	0.01	228	0.07
R2L	1126	0.22	16189	5.2

Principal Component Analysis (PCA) Neural Networks

As with the Radial Basis Function Networks, Principal Component Analysis (PCA) networks are a mixture of unsupervised and supervised networks. Principal component analysis is a linear procedure to find the direction in input space where most of the energy of the input lies. In other words, PCA performs feature extraction. The projections of these components correspond to the eigenvalues of the input covariance matrix. The unsupervised segment of the network performs the feature extraction and the supervised segment of the network performs the (linear or nonlinear) classification of these features using a MLP.

The principal component analysis is performed first, and then the MLP is trained. The reason for this is that the PCA network trains faster when it does not have to share computing resources with the MLP. There is no point in training the MLP until the eigenvalues are stable.

Advantages of PCA Neural Networks

Principal component analysis is a well known method of orthogonal zing data. It converges very fast and the theoretical method is well understood. Since the features are orthogonal, the MLP is able to train easily. There are usually fewer features extracted than there are inputs, so the unsupervised segment provides a means of data reduction.

Disadvantages of PCA Neural Networks

The most discriminate features are not always the features that have the largest eigenvalues. For these cases, the PCA is suboptimal and the choice of the number of features to extract is rather arbitrary.

PCA Theoretical

The fundamental problem in pattern recognition is to define data features that are important for the classification (feature extraction). One wishes to transform our input samples into a new space (the feature space) where the information about the samples is retained, but the dimensionality is reduced. This will make the classification job much easier.

Principal component analysis (PCA) also called Karhunen-Loeve transform of Singular Value Decomposition (SVD) is such a technique. PCA finds an orthogonal set of directions in the input space and provides a way of finding the projections into these directions in an ordered fashion. The first principal component is the one that has the largest projection (we can think that the projection is the shadow of our data cluster in each direction). The orthogonal directions are called the eigenvectors of the correlation matrix of the input vector, and the projections the corresponding eigenvalues.

Since PCA orders the projections, we can reduce the dimensionality by truncating the projections to a given order. The reconstruction error is equal to the sum of the projections (eigenvalues) left out. The features in the projection space become the eigenvalues. Note that this projection space is linear.

PCA is normally done by analytically solving an eigenvalue problem of the input correlation function. However, Sanger and Oja demonstrated that PCA can be accomplished by a single layer linear neural network trained with a modified Hebbian learning rule.

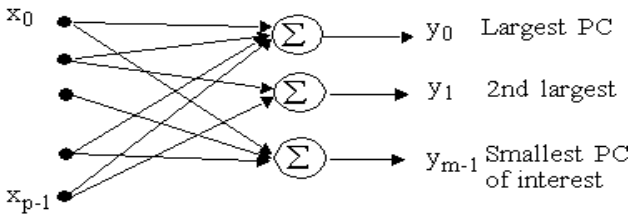
Let us consider the network shown in the figure below. Notice that the network has p inputs (we assume that our samples have p components) and m<p linear output PEs. The output is given by

$$y_i(n) = \sum_{i=0}^{p-1} w_{ij}(n) X_i(n) \qquad j = 0,1,\ldots, m-1$$

To train the weights, we will use the following modified Hebbian rule

$$\Delta w_{ji}(n) = \eta\left[y_j(n) X_i(n) - y_j(n) \sum_{k=0}^{j} w_{kj}(n) y_k(n) \right] \qquad \begin{matrix} i = 0,1,\ldots, p-1 \\ j = 0,1,\ldots, m-1 \end{matrix}$$

Where h is the step size.



PCA Network

Principal Component Analysis (PCA, also called Karhunen-Loeve transform) is one of the most widely used dimensionality reduction techniques for data analysis and compression. It is based on transforming a relatively large number of variables into a smaller number of uncorrelated variables by finding a few orthogonal linear combinations of the original variables with the largest variance. The first principal component of the transformation is the linear combination of the original variables with the largest variance; the second principal component is the linear combination of the original variables with the second largest variance and orthogonal to the first principal component and so on. In many data sets, the first several

principal components contribute most of the variance in the original data set, so that the rest can be disregarded with minimal loss of the variance for dimension reduction of the data. The transformation works as follows:

Given a set of observations $X_1, X_2, \ldots, X_n$ where each observation is represented by a vector of length m, the data set is thus represented by a matrix $X_{n \times m}$

$$X_{n \times m} = \begin{bmatrix} x_{11} & x_{12} & \ldots & x_{1m} \\ x_{21} & x_{22} & \ldots & x_{2m} \\ \ldots & \ldots & \ldots & \ldots \\ x_{n1} & x_{n2} & \ldots & x_{nm} \end{bmatrix} = \begin{bmatrix} X_1, & X_2, & \ldots, & X_n \end{bmatrix}$$

The average observation is defined as
The deviation from the average is defined as

$$\Phi_i = X_i - \mu$$

The sample covariance matrix of the data set is defined as

$$C = \frac{1}{N} \sum_{i=1}^{n} (X_i - \mu)(X_i - \mu)^T = \frac{1}{n} \sum_{i=1}^{n} \Phi_i \Phi_i^T = \frac{1}{n} AA^T$$

Where $A = \begin{bmatrix} \Phi_1, & \Phi_2, & \ldots & \Phi_n \end{bmatrix}$

To apply PCA to reduce high dimensional data, eigenvalues and corresponding eigenvectors of the sample covariance matrix C are usually computed by the Singular Value Decomposition (SVD). Suppose $(\lambda_1, u_1), (\lambda_2, u_2), \ldots, (\lambda_m, u_m)$ are m eigenvalue-eigenvector pairs of the sample covariance matrix C. We choose the k eigenvectors having the largest eigenvalues. Often there will be just a few large eigenvalues, and this implies that k is the inherent dimensionality of the subspace governing the "signal" while the remaining m-k dimensions generally contain noise. The dimensionality of the subspace k can be determined by

$$\frac{\sum_{i=1}^{k} \lambda_i}{\sum_{i=1}^{m} \lambda_i} \geq \alpha$$

Where α is the ratio of variation in the subspace to the total variation in the original space. We form a $m \times k$ matrix U whose columns consist of the k eigenvectors. The representation of the data by principal components consists of projecting the data onto the k-dimensional subspace according to the following rules (Http://kdd.ics.uci.edu/databases/kddcup99/kddcup).

$$y_i = U^T (X_i - \mu) = U^T \Phi_i$$

What is interesting in this network is that we are computing the eigenvectors of the correlation function of the input without ever computing the correlation function. Sanger showed that this learning procedure converges to the correct solution, i.e. the weights of the PCA network approach the first m principal components of the input data matrix. The outputs are therefore related to the eigenvalues and can be used as input to other neural networks for classification.

PCA networks can be used for data compression, providing the best m linear features. They can also be used for data reduction in conjunction with multilayer perceptron classifiers. In this case, however, the separability of the classes is not always guaranteed. If the data clusters are sufficiently separated, yes, but if the classes are on top of each other, the PCA will get the largest projections, but the separability can be in some of the other projections. Another problem with linear PCA networks is outlying data points. Outliers will distort the estimation of the eigenvectors and create skewed data projections. Nonlinear networks are better able to handle this case.

The importance of PCA analysis is that the number of inputs for the MLP classifier can be reduced a lot, which positively impacts the number of required training patterns, and the training times of the classifier.

The proposed system architecture

In this paper a new classification is introduced so that training network is done in two separated stages and serial form although the first stage for training can do parallel form, too. Our proposed architecture is illustrated in Figure 1.

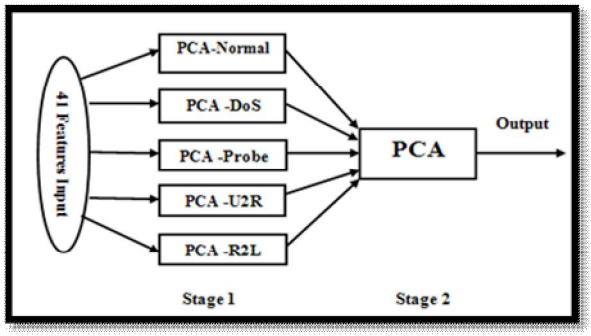


Figure 1. System architecture block diagram

Here, we use PCA neural network for study. After preprocessing on KDD database and using all 41 features of connections, these data are classified in five main groups. The proposed architecture includes two layers; it means that training network is done in two separated stages. In the first layer (stage 1), there are five PCA neural networks that are trained to cluster the input data for each connection type. Each PCA belongs to one of the groups in the database and each providing five outputs between 0 and 1. In this stage for training, the connections of every class are given to network of the same class. As connections of Normal are given to PCA-Normal, connections of DoS are given to PCA-DoS and so on. Every PCA has 41 processing elements (PE) in input layer, 5 PEs in hidden layer and 5 PEs in output layer- 4 attack categories and 1 normal connection- that is shown as [41; 5; 5]. At the end of this stage a little knowledge from database about different classes is provided. In the second layer (stage 2), there is a PCA neural network that uses output of stage 1 as input and has [5; 30; 30; 5]. Our system uses all these features and connections in database as training database that was introduced in section 2. After stage 2 training network is finished. Note that for testing network every connection is given to all networks and the main result is obtained at the end of stage 2 which shows the place of each input connection in these five classes.

Performance Evaluation

For evaluation and comparison works is applied different parameters but in this paper, we use all of evaluation parameters. In this section four parameters is introduced for the comparison this work with other works.

- Detection Rate: ratio between the number of correctly detected attacks and the total number of attacks.
- False Alarm (False Positive): ratio between the numbers of normal connections that is incorrectly misclassified as attacks and the total number of normal connections.
- Classification Rate: this parameter for each class of data is defined as the ratio between the number of test instances correctly classified and the total number of test instances of this class.
- Cost per Example (CPE): CPE is calculated using the following formula:

$$\text{CPE} = \frac{1}{N}\sum_{i=1}^{m}\sum_{j=1}^{m} CM(i,j) * C(i,j)$$

Where CM and C are Confusion Matrix and Cost matrix, respectively, and N represents the total number of test instances, m is the number of classes in classification. A Confusion Matrix is a square matrix in which each column corresponds to the output class, while rows correspond to the desired classes. An entry at row i and column j, CM (i, j) represent the number of misclassified instances that originally belong to class i, although incorrectly identified as a member of class j. The entries of the primary diagonal-CM (i, i) - stand for the number of properly detected instances. Cost matrix is similarly defined, as well as entry C (i, j) represents the cost penalty for misclassifying an instance belonging to class i into class j (Nadjaran-Toosi & Kahani, 2007). Cost matrix values employed for the KDD's classifier learning contest are shown in Table 2 (Http://kdd.ics.uci.edu/databases/kddcup99/kddcup). Lower values for CPE show better classification for IDS.

Table 2. Cost matrix values for the KDD's classifier learning contest.

Desired/Output	Normal	DoS	Probe	U2R	R2L
Normal	0	2	1	2	2
DoS	2	0	1	2	2
Probe	1	2	0	2	2
U2R	3	2	2	0	2
R2L	4	2	2	2	0

Experiments

NeuroSolution software (NeuroSolution, 1994-2006) was used for the implementation of the studied neural network, on a T5500 (1.66 GHz), with 0.99GB of memory. All connections of testing database (Table 1) are applied to evaluation. We discuss the results according to the performance evaluation. The system reaches its best performance for high value of detection rate and low value of false alarm rate (Beghdad, 2005). We did an experiment on the base of proposed system in this paper. In this experiment the transfer function was TanhAxon, and the learning rule was Momentum. The Mean Square Error (MSE) in the training step was 0.01. Results of experiments are shown in a CM in Table 3. Detection Rate (DR), False Positive (FP) and Cost per Example (CPE) are also added.

Table 3. Confusion Matrix for PCA with proposed system

Desired/Output	Normal	DoS	Probe	U2R	R2L
Normal	60593	0	0	0	0
DoS	1024	228829	0	0	0
Probe	0	0	4166	0	0
U2R	5	0	223	0	0
R2L	3	0	0	0	16186
FP (%)	1.67	0	5.008	0	0
DR (%)	99.596				
CPE	0.008105				

Table 4. Comparison between some IDS and our approach

Performance Evaluation (%) / Approach	Normal	DoS	Probe	U2R	R2L	DR	FP	CPE
[4]	-	85	69.3	64.9	70	76.3	0.87	-
[8]	99.51	98.37	88.40	84.33	71.13	90	10.29	-
[9]	94.9	99.46	95.17	0	90.76	98.49	-	-
[10]	99.70	99.95	99.45	79.96	100	99.90	-	-
[11]	99.85	99.36	92.45	0	0	99.16	-	-
[18]	98.2	99.5	84.1	14.1	31.5	95.3	1.9	0.1579
Our approach	100	99.554	100	0	99.981	99.596	0.404	0.008105

According to the above table, proposed system has good performance that is competitive with the other approaches from the point of all performance evaluation.

Conclusion

In this section we investigate the performance of our approach with other works in this field. This information is showed in Table 4.

Notice that we used all connections in database for training and results showed that our approach has better or the same level performance in comparison with other works. With regard to DR the best result is 99.90% in (Shanmugam & Idris, 2007) but DR in our approach is 99.594% that is competitive with the best approach. When we compare FP in different approaches, our approach with 0.404% has the least value. CPE parameter aren't calculated in all works but with regard to values of classification rate in Table 4, our approach has the lowest value. With regard to the fact that the numbers of R2L and Probe connections are used in network training are very limited, the value of classification rate for these classes is very low in most work. But in our method this amount has improved noticeably. Using and combining neural networks with other approaches and reducing number and feature connections for training neural network in order to learn only useful information will be our future work.

REFERENCES

Amini, M., Jalili, R. & Shahriari, H.R. (2006). RT-UNNID: A practical solution to real-time network-based intrusion detection using unsupervised neural networks. Comp. & Sec, *25*, 459-468.

Beghdad, R. (2005). Applying fisher's filter to select kdd connection's features and using neural network. Neu. Net. Wor. *15*, 35-52.

Beghdad, R. (2007). Training all the KDD data set to classify and detect attacks, Neu. Net. Wor, *17*, 81.

Cordella, L.P. & Sansone, C. (2007). A multi-stage classification system for detecting intrusions in computer networks, Patt. Anal. App, *10*, 83–100.

Crescenzo, G.D., Ghosh, A., & Talpade, R. (2005). Towards a theory of intrusion detection, Comp. Sec. – ESORICS, *3679*, 267-286.

Degang, Y., Guo, C., Hui, W. & Xiaofeng, L. (2007). Learning vector quantization neural network method for network intrusion detection, Wuhan university journal of natural sciences (WUJNS). Article ID: 1007-120201-0147-04, DOI 10.1007/s11859-006-0258-z.

Gunes-Kayacik, H., Zincir-Heywood, A.N. & Heywood, M.I. (2006). A hierarchal SOM-based intrusion detection system. Engg. App. Arti. Intelli., doi: 10.1016/j.engappai.2006.09.005.

Http://kdd.ics.uci.edu/databases/kddcup99/kddcup.

Jazzar, M. & Jantan, A. (2008). A novel soft computing inference engine model for intrusion detection, IJCSNS Int. J. Comp. Sci. Net. Sec. *8*.

Kayacik, H.G., Zincir-Heywood, A.N., Heywood, M.I. (2005). A hierarchical SOM based intrusion detection system.

Koziol, J. (2003). Intrusion detection with Snort, Sams Publishing.

Lippmann, R.P., Cunningham, R.K., Webster, S.E., Graf, I. & Fried, D. (1999). Using bottleneck verification to find novel new computer attacks with a low false alarm rate, Unpublished Technical Report.

Liu, G., Yi, Z. & Yang, S. (2007). A hierarchical intrusion detection model based on the PCA neural networks, Neurocomp, *70*, 1561–1568.

Moradi, M. & Zulkernine, M. (2004). A neural network based system for intusion detection and classification of attacks, In proceedings of 2004 IEEE International Conference on Advances in Intelligent System-Theory and Applications. Luxembourg-kirchberg, Luxembourg, November, 15-18.

Nadjaran-Toosi, A. & Kahani, M. (2007). A new approach to intrusion detection based on an evolutionary soft computing model using neuro-fuzzy classifiers. comp. connec, *30*, 2201-2212.

Shanmugam, B. & Idris, N.B. (2007). Improved hybrid intelligent intrusion detection system using AI technique. Neu. Net. Wor, *17*, 351.

Wang, W., Guan, X., Zhang, X. & Yang, L. (2006). Profiling program behavior for anomaly intrusion detection based on the transition and frequency property of computer audit data. Comp. & Sec, *25*, 539-550.

Zhang, C., Jiang, J. & Kamel, M. (2005). Intrusion detection using hierarchical neural networks, Patt. Recog. Lett, *26*, 779–791.