Applied Mathematics & Information Sciences
An International Journal

http://dx.doi.org/10.12785/amis/080222

Enhancing MLP Performance in Intrusion Detection using Optimal Feature Subset Selection based on Genetic Principal Components

*Iftikhar Ahmad**

Department of Software Engineering, College of Computer and Information Sciences, King Saud University, P.O. Box 51178, Riyadh 11543, Saudi Arabia

Received: 26 Mar. 2013, Revised: 27 Jul. 2013, Accepted: 29 Jul. 2013
Published online: 1 Mar. 2014

Abstract: Computer and network systems nowadays are facing many security issues, one of which considered important is intrusion. To prevent such intrusion, a mechanism for optimal intrusion detection is deemed necessary. A number of tools and techniques are available, yet most of them still face a main problem that is on performance. The performance, in essence, can be increased by reducing false positives and increasing accurate detection rate. What has made the performance terrible in the existing intrusion detection approaches is due to the use of a raw dataset that includes redundancy and leads the classifier to be confused. To overcome this issue, Principal Component Analysis (PCA) has been used to project a number of raw features on principal feature space and to select the features based on their sensitivity determined by the magnitude of eigenvalues. Here, only the features corresponding to the highest eigenvalues are selected; the remaining features, by contrast, are ignored. Due to the ignorance of many important and sensitive features for the classifier for their lowest eigenvalues, this method comes to be not optimal. Therefore, a suitable method is necessary to select a subset of features, which, in turn, can enhance the classifier performance. The focus of this research is to observe a space of principal features to find a subset of sensitive features to the classifier, which can optimize the detection accuracy. Genetic Algorithm (GA) has been applied to solve an optimization problem. The raw features have, afterwards, been transformed through PCA into principal features space. GA, in this case, was used to search this features space to obtain principal components called genetic principal components (GPC). The feature set obtained through this process was, in turn, presented to the classifier. The Multilayer Perceptron (MLP), meanwhile, was used for classification considering its proven ability. Additionally, Knowledge Discovery and Data mining (KDD) cup dataset was used for the validation of the proposed approach, which is considered as a benchmark to evaluate the intrusion detection approaches. The performance of this approach has been analyzed and compared with a number of existing approaches. The results then show that proposed method outperforms the existing approaches. Not only does it significantly reduce the dimension of the feature space but also improves the detection accuracy.

Keywords: Intrusion Detection System (IDS), Multilayer Perceptron (MLP), Principal Component Analysis (PCA), Genetic Algorithm (GA), Genetic Principal Component (GPC), Detection Rate (DR), False Positive (FP), False Negative (FN), True Negative (TN), True Positive (TP) and Dataset.

1 Introduction

Security breach in network today has become one of the major problems since a single of it may cause a significant loss or damage to the information systems. Hence, an effective intrusion detection system is deemed essential in use to address such incidents or intrusions. In response, a wide variety of intrusion detection systems are now available. However, the main issue on their poor performance still remains due to false positives.

One of the reasons for this is related to the use of raw dataset. The performance of intrusion detection principally can be increased by using a proper feature selection method. A number of previous intrusion detection techniques, in response, have attempted to focus on the issues of feature extraction and classification, yet less importance unfortunately has been given to the serious issue of feature selection. In past, a subset of features was selected using PCA based on some percentage of the top principal components. Further, the

* Corresponding author e-mail: wattoohu@gmail.com

features corresponding to the highest eigenvalues were selected and those corresponding to the lowest ones, by contrast, were ignored. This method in selecting an appropriate set of features might be not effective for being potential to omit certain features that, for their sensitivity, might be very important to the classifier. To cope with this problem, GA is proposed for selecting good subsets of features from the PCA space.

This study clarifies that feature selection is a significant issue in intrusion detection. Further, the combination of PCA and GA provide a simple, general, and powerful framework in selecting a number of important and sensitive features, leading to improved performance of the classifier. In this work, raw features have been presented to PCA for transformation to principal feature space. This transformation makes such features more visible and organized in PCA feature space. At this point, the feature space was searched through GA to find some optimal features based on genetic eigenvectors. The resultant feature set was then presented to the classifier. On the other hand, MLP (Multilayer Perceptron) was used as a classifier and tested on the selected feature set. The standard dataset, KDD cup dataset, also was used to validate the proposed model. The experimental results then illustrate significant performance improvements.

The rest of the paper is organized as follows. Section 2 is designed to present some related work. It is then followed by Section 3 dealing with the proposed model including the explanation of dataset, feature selection process using PCA and GA, classification and training and testing. Section 4 and Section 5 are to discuss the experiments and results and to draw a conclusion respectively.

2 Related Work

Feature selection is a serious issue in intrusion detection. In some previous approaches, it was considered insignificant and depended on powerful classification algorithms to deal with redundant and irrelevant features. Moreover, feature extraction and classification have been more focused in intrusion detection in which the features have been extracted from the raw features using PCA. The raw features were then projected to principal space to select a subset of features. Features corresponding to highest eigenvalues were included in the subset, and those corresponding to the lowest eigenvalues, oppositely, were ignored. In this method, some percentage of the total features was included in the feature subset [11, 12, 13].

In fact, this is not an effective scheme to select an appropriate set of features in this space in consideration to the potential of missing important and sensitive features to the classifier. Related work in intrusion detection is presented in which the prime concentration is on classification.

In [2], PCA and neural networks have been used to detect intrusion. The PCA here was applied for classification and neural networks for online computing. The features were selected based on 22 principal components while others were ignored for being less important - the importance of features was determined based on the highest eigenvalues. Such feature selection, for some reasons, is not effective for a possibility to miss many important features having sensitive information for classifier or intrusive analysis engine [11, 12].

The importance of a feature is defined differently in the existing research work of intrusion detection. It can be determined based on eigenvalues, accuracy, detection rate, and false alarms. In [3], the importance of a feature is determined based on the accuracy and the number of false positives of the system with and without the feature. The feature selection was based on method: leaving-one-out; removing one feature from the feature set, performing experiment, and comparing the new results with the original result. If any case of these described cases occurs, the feature is regarded as important; otherwise, unimportant. To illustrate, with 200 features in the original set, an experiment might be repeated 200 times to ensure whether each feature is important or not. As a consequence, not only does this method involve complexity but also overheads on huge dataset.

In [4], real-time pattern classification has been performed using a radial basis function (RBF) network. The Elman network here was used to restore the memory of past events and full featured DARPA dataset was used in experiment of this work. Such method, consequently, might increase training and testing overheads on the system as well as make the classifier confused to produce false alarms.

In [5], PCA was used to determine a subset of features based on a feature reduction concept. The feature reduction accelerated training and testing process for the classifier. However, this can affect on the efficiency and accuracy of the system. For example, few numbers of principal components speed up the training efficiency while a large number of principal components make the classifier confused to produce false alarms. Such compromise in fact is not suitable in intrusion detection mechanism. Hence, other method suitable in feature section to avoid such compromise is deemed essential.

In [6], GA was in use for features and parameters selection for intrusion detection model. The Support Vector Machines (SVM) has been applied as an intrusive analysis engine. Even though this method was capable of minimizing a number of features and maximizing the detection rates,, features uniformity still become the problem. Since the features in original forms are not consistent, they must be transformed into a feature space for a well organized form.

In [11], a mechanism of intrusion detection was proposed using a number of soft computing techniques: SVM, GA and PCA. The proposed model was tested on

two subsets of features, one of which was obtained using PCA and GA while another one was selected on the basis of top percentage of principal components using PCA only. The first subset consisted of 12 features while the second one did consist of 22 features all directly taken from the principal space. Both sets were then tested on SVM, and compared for their performance. The focus was performance comparison on both feature sets. In fact, this approach was still needed further experimentation to validate the proposed model.

The above mentioned approach was explored further in [15]. The principal space was concentrated on selecting features for a suitable subset. The GA was used to search the PCA space for feature selection. Several experiments were conducted for optimal feature subset selection, and their results were analyzed. The results showed that the proposed approach outperformed the existing approaches. This work used SVM which is not suitable for large dataset and it also increases training time of the classifier.

In [13], an attempt of a feature subset was initiated using the MLP as classifier. However, the presented approach was not sufficient to verify the concept. Three different subsets were selected with 12, 20 and 27 features. These subsets were tested on MLP, and their results were analyzed. This approach had a number of issues, which were not considered and tested to validate the concept. For example, the classifier may perform better on the feature sets consisting of raw features, transformed features and conventionally PCA selected features. This work then will be further explored, extended and verified in the model proposed in this research.

3 Proposed Model

The proposed model consists of four sections; Dataset, Feature selection, Classification and Training & Testing. The selection of dataset is considered as an important issue in the intrusion detection considering that an accurate dataset can result in accurate results. Dataset collection could be conducted in several ways; those are real time, simulated and test-bed, each of which has various issues. To avoid such issues, the standard dataset; KDD cup dataset has been used to validate the model of this work. Feature selection, meanwhile, was accomplished through GA and PCA due to their proven ability in feature selection. The selected features sets were presented to the classifier to determine their sensitivity and importance. Further, the classification was performed through a well-known classifier such as MLP. The classifier, subsequently, was trained and tested to analyze the feature subsets. Figure 1 illustrates the proposed model. The detail of the proposed model is described as follows.

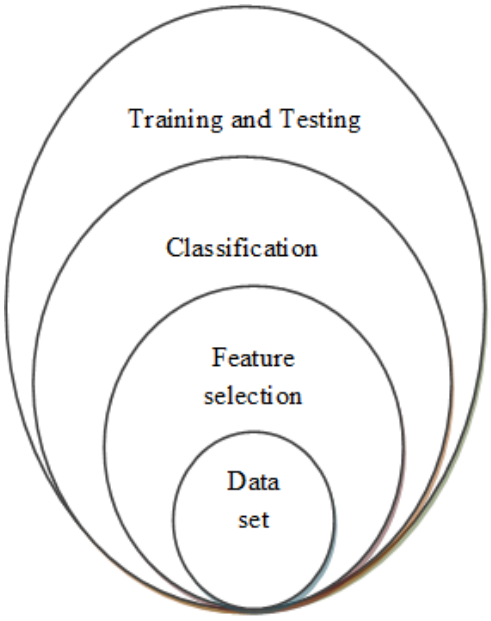

Fig. 1: Proposed model for intrusion detection

3.1 Dataset

This work used KDD cup dataset, considered as a standard in the evaluation of intrusion detection techniques. From the dataset, 20.000 connections were randomly selected in which each connection of raw dataset consisted of 41 features.

$$f_1, f_2, f_3, f_4, .. f_n \quad where \quad n = 41 \tag{1}$$

Three symbolic features; protocol-type, service and flag were discarded for having no any impact on the classifier. Thus, the features remained 38 in each record.

$$f_1, f_2, f_3, f_4, .. f_m \quad where \quad m = 38 \tag{2}$$

3.2 Feature selection

The feature selection is an important task in this work. The suitable feature set simplified the classifier architecture as well as improved its overall performance. Figure 2 presents the flow of feature selection algorithm. The feature selection algorithm used two types of techniques: PCA and GA, which has been being widely used in the process of feature selection in many various fields such as image processing, data mining or medical. Below is the algorithm proposed for feature selection.

Algorithm
Step 1: Let Fs is a feature set, which consists of 38 features.
$Fs = f_1, f_2, f_3, f_4, f_m$ where $m = 38$ representing a total

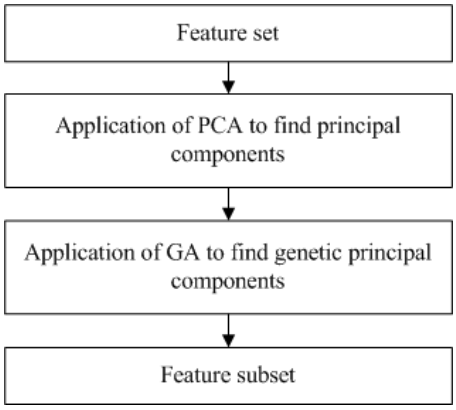

Fig. 2: Feature subset setection algorithm flow

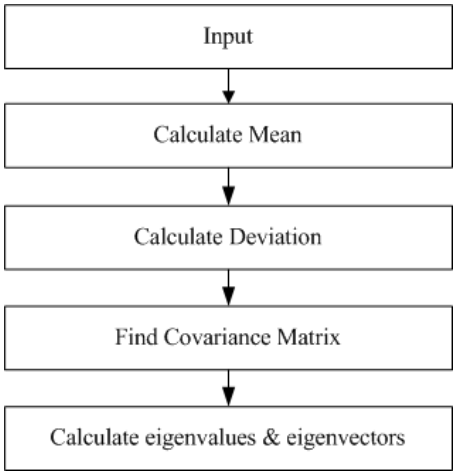

Fig. 3: PCA algorithm flow

number of features.
Step 2: The feature set (Fs) is projected into another feature space called the principal feature space. This can be expressed as
$Fs \rightarrow PCA \rightarrow \sum_{i=1}^{N=38} PCs_i$
Step 3: The principal feature space is searched through GA for the selection of genetic eigenvectors. This is represented in the following expression.
$PCs \rightarrow GA \rightarrow \sum_{k=1}^{r} GPCs_k$
Step 4: A set of features $(F's)$ is obtained based on genetic principal components, which are the subset of original feature set (Fs). This can be expressed as.
$GPCs_k \rightarrow F's_l$
$F's_l \subset Fs$
The term r represents random number and l indicates the number of features less than 38. This step is repeated several times until obtaining a set with a maximum accuracy and minimum number of features.

The feature selection process and its applied techniques; PCA and GA are explained respectively as follows. PCA is a valuable statistical method that has an application in many fields such as face recognition and image compression. It is a common technique in finding patterns in high dimension data. It has also been used to analyze a large data set and the relationship between the individual points in that set [4,11,22]. PCA purposely is to reduce the dimension of the data while retaining as much as possible of the variation present in the original dataset. It provides a way of identifying patterns in data, and expressing the data in such a way as to highlight their similarities and differences [6,11]. However, PCA here was used to transform the input vectors to the new search space. On the other hand, the choosing of number of principal components is done by GA. The flow of applied PCA is sown in Figure 3 . Below is the PCA algorithm applied in this work.

Algorithm
Let $x_1, x_1, x_1, x_1, ..x_M$ are $N \times 1$ vectors
Step 1: $\bar{x} = \frac{1}{M} \sum_{i=1}^{M} x_i$
Step 2: Subtract the mean: $\phi(x1 - \bar{x})$
Step 3: From the matrix $A = [\phi 1, \phi 2, \phi 3.....\phi M]$ $(N * M)$
Matrix then compute $C = \frac{1}{M} \sum_{N=1}^{M} \phi n \phi n = AA^T$
Step 4: Compute the eigenvalues of $C : \lambda 1 > \lambda 2 > .....\lambda N$
Step 5: Compute the eigenvectors of $C = \mu_1, \mu_2.....\mu_N$
Since C is symmetric, $C = \mu_1, \mu_2.....\mu_N$ form a basis i.e. any vector x or actually $(x1 - \bar{x})$ can be written as a linear combination of the eigenvectors.
$(x_1 - \bar{x}) = b_1\mu_1 + b_2\mu_2 + ... + b_N\mu_N = \sum_{i=1}^{N} b_i\mu_i$

In order to overcome the issue in optimal feature selection, GA was applied to search the principal components space in order to select an optimal subset of features. This is a main contribution that positively impact on the performance of intrusion detection analysis engine. GA is inspired by the biological mechanisms of reproduction [5,11,12,13,24]. GAs operate iteratively on a population of structures, each of which represents a candidate solution to the problem.

In this work, the initial population was generated randomly in which each individual approximately contained the same number of 1s and 0s. All experiments used a population size of five thousands and hundred generations. In most cases, the GA converged in less than hundred generations [11,13]. The GA started its search on the initial population. Three basic genetic operators, namely selection, crossover, and mutation have guided this search. The genetic search process is iterative: evaluating, selecting, and recombining strings in the population during each iteration or generation until reaching some termination conditions. The applied GA algorithm is given.

Algorithm
Step 1: Create initial population. The chromosomes are

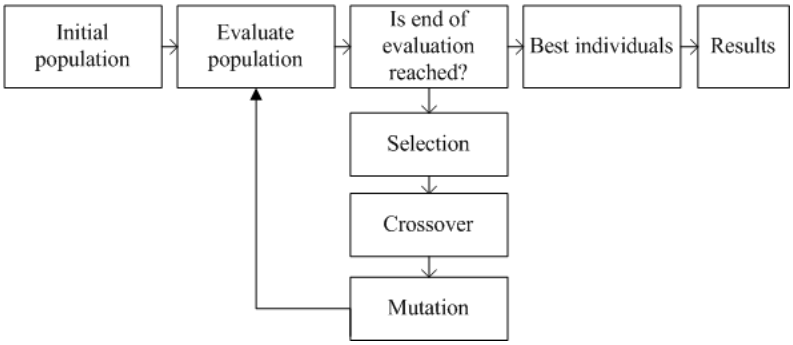

Fig. 4: GA algorithm flow

selected randomly from principal components.
Step 2: Evaluate the population
Step 3: Check the termination condition?
i. If condition is satisfied than select best individuals
ii. else go to step 2 (selection crossover mutation)
Step 4: Find subset of genetic principal components
Step 5: Select feature subset

Each string was evaluated based on a fitness function determining the suitability of the candidate solutions. This was inspired by the idea of survival of the fittest in natural selection. Selecting a string depends on its fitness relative to other strings in the population. So, strings with high fitness were selected and the remaining ones with low fitness were removed from the population. Selection acted as a filter, removing poor performance solutions and selected high performance one. Further, selection as a genetic operator chose chromosomes from the current generations population for inclusion in the next generations population. Five selection operators; namely roulette, tournament, top percent, best and random were used. Tournament used roulette selection N times (the Tournament Size) to produce a tournament subset of chromosomes. The best chromosome in this subset was then chosen as the selected chromosome. This selection method applied an additional selective pressure over plain roulette selection. There was also an option to specify whether the chance of being selected was based on fitness or on rank. In Best, the best chromosome was selected (as determined by the lowest cost of the training run). If there are two or more chromosomes with the same best cost, one of them is chosen randomly. In random, a chromosome from the population was randomly selected. Similarly, in top percent, a chromosome from the top N percent (the Percentage) of the population was randomly selected. Top percent selection method was used in the experiments for giving a better performance compared to other selection operators. So, the selection strategy was GA generational. Assuming a population of size N, the offspring then doubled the size of the population and selected the best top 10 percent individuals from the combined parent-offspring population.

Before making into the next generations population, selected chromosomes may undergo crossover and mutation. Fundamentally, crossover is categorized into three: one-point crossover, two-point crossover, and uniform crossover [12]. For one-point crossover, the parent chromosomes were divided at a common point chosen randomly and the resulting sub-chromosomes are swapped. For two-point crossover, the chromosomes were thought of as rings with the first and last gene connected. In this case, the rings were divided into two common points chosen randomly and the resulting sub-rings were swapped. Uniform crossover, meanwhile, was different from the above two schemes. In this case, each gene of the offspring was selected randomly from the corresponding genes of the parents. For simplicity, we used one-point crossover here. The crossover probability used in all experiments was 0.9.

Crossover is applied with high probability and allows information exchange between points. Its goal is to preserve the fittest individuals without introducing any new value. Mutation, in contrast, is a low probability operator, which flips a specific bit to restore the lost genetic material [11, 12].

Mutation is a genetic operator that alters one or more gene values in a chromosome from its initial state [13]. This can result in entirely new gene values being added to the gene pool. With these new gene values, the genetic algorithm may be able to arrive at a better solution than the previous one. Mutation is an important part of the genetic search as it helps to prevent the population from being stagnant at any local optima. It occurs during evolution according to the defined probability. This probability should usually be set fairly low. If it is set too high, the search will turn into a primitive random search [3]. The traditional mutation operator is used which just flips a specific bit with a very low probability. The mutation probability used in all experiments was 0.01.

Crossover and mutation generate new solutions for exploration through string operations. Genetic algorithms do not guarantee a global optimum solution. However, they have an ability to search through very large search spaces and come to nearly optimal solutions fast [1].

In this work, a simple encoding scheme was used where the chromosome was a bit string whose length was determined by the number of principal components. Each principal component, computed using PCA, was associated with one bit in the string. If the ith bit is 1, the ith principal component is selected; otherwise, that component is ignored. Each chromosome thus represents a different subset of principal components [11,12].

The main goal of feature subset selection is to use less features to achieve the same or better performance. Therefore, the fitness evaluation contains two terms: (i) accuracy and (ii) the number of selected features. The performance of MLP is estimated using a validation dataset which guides the GA search. Each feature subset contains a certain number of principal components. If two subsets achieve the same performance with different number of principal components, the subset with fewer principal components is preferred [1,12]. Between accuracy and feature subset size, accuracy is a major concern. The fitness function combine the two terms is shown:

$$f(t) = 10^4 CA + 0.5CNS \tag{3}$$

Where CA corresponds to the classification accuracy on a validation set for a particular subset of principal components and CNS corresponds to the number principal components not selected. The CA term ranges roughly from 0.50 to 0.99, thus, the first term assumes values from 5000 to 9900. The CNS term ranges from 0 to $l-1$ where l is the length of the chromosome, thus, the second term assumes values from 0 to $37(l=38)$. The CA term dominates the fitness value based on the weights as assigned to each term. This implies that individuals with higher classification accuracy will outweigh those with lower one - no matter how many features they contain. On the whole, the higher the accuracy is, the higher the fitness is [1]. Also, the fewer the number of features is, the higher the fitness is. Selecting the weights for the two terms of the fitness function is more objective dependent than application dependent.Many factors are considered in the design of an intrusion detection system. However, there must be the best balance between model compactness and performance accuracy. Under some scenarios, performance is preferred, no matter what the cost might be. If this is the case, the weight associated with the CA term should be very high. Under different situations, compact models are favored over accuracy as long as the accuracy is within a satisfactory range. In this case, choose a higher weight for the CNS term. Thus, these experiments are the best balance between model compactness and performance accuracy using GA and the classifier. The fitness is calculated based on above mentioned formula. Three different feature sets are obtained such as;

$$F's_l \subset Fs \;\; where\; l = 1,2,3 \tag{4}$$

Further, two more sets were included to validate the technique on PCA transformed set and raw dataset.

$$F's_{pca} = f_1, f_2, f_3, f_4, ..., f_{38} \tag{5}$$

$$F's_{raw} = f_1, f_2, f_3, f_4, ..., f_{38} \tag{6}$$

3.3 Classification

The selected features were presented to MLP for classification. Here, the short detail of MLP was given. There are two important characteristics of the multilayer perceptron (MLP). First, its processing elements (PEs) or neurons are nonlinear. The nonlinearity functionality is provided by the functions; logistic and hyperbolic tangent. Second, they are massively interconnected such that any element of a given layer feeds all the elements of the next layer. Using MLP in this problem bring some following advantages [12,13]. First, MLPs are very powerful pattern classifiers. Second, with one or two hidden layers they can approximate virtually any input-output map. Third, they show a better performance to other classifier in difficult problems. Fourth, they efficiently use the information contained in the input data.

A MLP is a feed forward neural network that maps the sets of input data onto a set of appropriate output. MLP architecture used consists of three layers; namely input, hidden and output. In this architecture, hidden layer and output layer consist of neurons (processing elements), each of which has a nonlinear activation function. The layers are fully connected from one layer to the next. MLP is an amendment of the standard linear perceptron, which can discriminate data that is not linearly separable [12,13,21]. Figure 5 shows the MLP architecture used in this work and Figure 6 illustrates its implemented architecture. The pseudo code of the back propagation algorithm is given below.

Algorithm
Input: $training-examples, \eta, \phi, net$
Output: $trainednetwork$
$Initialize\ all\ weights\ of\ net$;
for each pair $< \overrightarrow{x}, \overrightarrow{t} >\in$ training-examples do
Step 1: *Forward phase* :
Present the input $< \overrightarrow{x}$ to the input layer of the *net*;
for each unit $u \in net$ do
calculate the output o_u of unit u;
Step 2: *Backward phase* :
Calculate errors:
for each unit $k \in$ output layer, calculate its error δ_k do
$\delta_(k) \leftarrow o_(k)(1-o_(k)(t-o_(k)$;
for each hidden unit h, calculate its errors δ_h do
$\delta_h \leftarrow o_h(1 \leftarrow o_h)\sum_k \in outputs\omega khk\delta$
Step 3: $Update\ weights$:
for each weight $\omega_i j \in net$ do
$\Delta\omega_i j(t+1) = \alpha\delta x_j i + \eta\Delta\omega_i j(t)$;

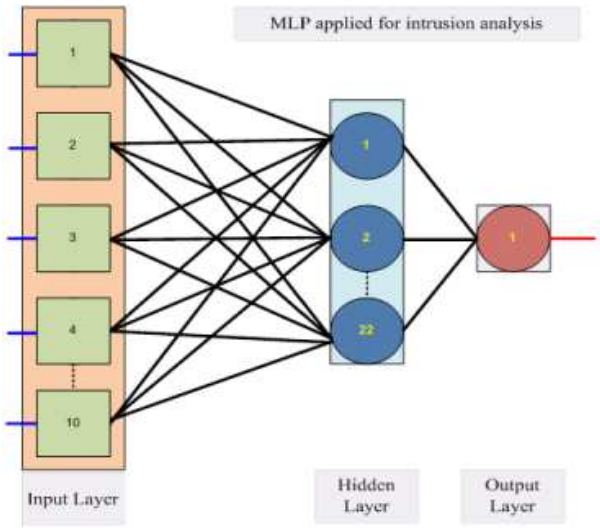

Fig. 5: Classifier architecture for intrusion analysis

$$\omega_i j(t+1) = \omega_i j(t) + \Delta\omega_i j(t+1)\ ;$$

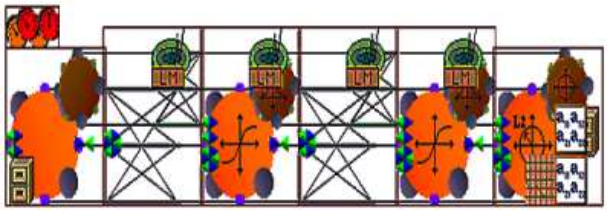

Fig. 6: Implemented Classifier architecture

3.4 Training and Testing

In the training phase, input patterns and desired outputs are given related to each input vector. It is aimed to minimize the error output produced by the classifier and the desired output [12]. To achieve this goal, weights are updated by carrying out certain steps known as training. Table 1 presents the parametric specification used for MLP architecture during training phase. When the system is trained well, the weights of the system are frozen and performance of the system is evaluated. Testing of trained system involves two steps; (i) verification step, and (ii) generalization step.

In verification step, a trained system is tested against the data, which are used in training. The purpose of this step is to investigate how well trained system has learned the training patterns in the training dataset. If a system was trained successfully, the outputs produced by the system would be similar to the real outputs. In this research work, 30% of the training dataset (5000) was used as verification (i.e. 1500).

In generalization step, testing is conducted with data not used in training. It is purposely to measure the generalization ability of the trained network. After training, the system only involves the computation of the feed forward phase. For this purpose, a production dataset that has input data but no desired data is used. This work used a dataset of fifteen thousand $(15,000)$ as a production dataset. Further, the system performance was also tested on total dataset $(20,000)$ consisting of both training dataset and production dataset. The parametric specification used for MLP architecture during testing phase is given in Table 2.

The purpose of testing phase is to observe the system how well the system learned the training dataset after the training process. The sensitivity analysis of confusion matrix of training, cross validation and testing dataset is shown in Table 3. The training time, training epochs and performance of testing phase is presented in Table 4.

In verification phase, in order to observe generalization performance of the trained system the trained MLP with different feature set is tested on production dataset, which is not a part of the training set. The overall performance of MLP during verification phase is shown in Table 5. Table 6 shows a comparative analysis among various feature sets with some results indicating that MLP with ten features (subset) has increased its performance based on genetic principal components as compared to other feature sets.

The comparison of different cases in Table 6proved that the mechanism using GA to search the PCA features space for genetic principal components provides an optimal performance as compared to the traditional

Table 1: Classifier parameters during training

S.No	Parameter Name	Value
1	Architecture	MLP Feedforward
2	Layers	03 (input, hidden and output)
3	Input samples features	38 (original), 22 (PCA), and 10 (GA)
4	PEs in Input layer	It depends on features subset selections. For examples; 38, 22,12, & 10.
5	PEs in Hidden Layer	If number of features are 10 than PEs are 22 in hidden layer.
6	Epochs	1000
7	PE in output layer	One that has value 0 or 1
8	Activation function	Tanh
9	Training algorithm	Backpropagation (Forward & Backward)
10	Training dataset	5000 connections in which 20% for cross-validation and 30% for testing
11	Production dataset	20,000 connections

Table 2: Classifier parameters during Testing

S.No	Parameter Name	Value
1	Architecture	MLP Feedforward
2	Layers	03 (input, hidden and output)
3	Input samples features	38 (original), 22 (PCA),GPC-12 and GPC-10 (GA)
4	PEs in Input layer	It depends on features subset selections
5	PEs in Hidden Layer	If number of features are 10 than PEs are 22 in hidden layer
6	Epochs	1
7	PE in output layer	1
8	Activation function	Tanh
9	Training algorithm	NO. But it involves feedforward phase only
10	Testing dataset	3000 connections for testing and 2000 for cross-validation
11	Production dataset	20,000 connections

Table 3: Sensitivity analysis of training, cross-validation and testing dataset

Feature set	Dataset (s)	True Positive (%)	False Positive (%)	False Negative(%)	True Negative(%)
Raw-38	Training	100	0.0	2.59	97
	Cross Validation	98.71	1.28	2.25	97.74
	Testing	97.07	2.92	2.54	97.45
PC-38	Training	100	0.0	0.0	100
	Cross Validation	99.84	0.153	0.0	100
	Testing	100	0.0	0.0	100
PC-22	Training	100	0.0	0.0	100
	Cross Validation	100	0.0	0.0	100
	Testing	100	0.0	0.0	100
GPC-12	Training	100	0.0	0.0	100
	Cross Validation	100	0.0	0.0	100
	Testing	100	0.0	0.0	100
GPC-10	Training	100	0.0	0.0	100
	Cross Validation	100	0.0	0.0	100
	Testing	100	0.0	0.0	100

feature selection from PCA search space. The key focus of the research was to select sensitive features and minimum features as well as to increase accuracy of the system.

Thus, research work achieved this objective by using GA and PCA that made the classifier simpler as well as more efficient in performance. Hence, this method shows that proposed method provides MLP based intrusion detection mechanism that outperforms the existing approaches and has the capability to minimize the number of features and maximize the detection rates.

4 Experimental Results

The MLP based intrusion analysis engine was evaluated on different feature subsets. This section presents MLP results and their sensitivity analysis in different scenarios. First of all, MLP was tested on original dataset without

Table 4: Training time, training epochs and performance of testing phase

Feature set	Training Time(H:M:S)	Training Epochs(Number)	Detection rate(%)	False Alarm(%)
Raw-38	01:29:36	1000	43.28	56.72
PC-38	01:28:07	1000	99.96	0.035
PC-22	01:08:07	1000	99.95	0.055
GPC-12	00:23:00	217	99.985	0.015
GPC-10	00:20:00	173	99.99	0.01

Table 5: Performance of verification phase

Feature set	Features(num.)	True Positive(num.)	True Negative(num.)	False Alarm(num.)
Raw-38	760000	1456	18544	11344
PC-38	760000	12793	7207	07
PC-22	440000	12789	7211	11
GPC-12	440000	12797	7203	03
GPC-10	200000	12798	7202	02

Table 6: Classifier performance on different feature sets

Feature set	GPC-10	GPC-12	PC-22	PC-38	Raw-38
False alarm	02	03	11	07	11344
Epochs	173	217	1000	1000	1000
Time	00:20:00	00:23:00	01:08:07	01:28:07	01:29:36
Features size	564 KB	2.17 MB	5.15MB	8.37MB	8.37MB
Features in numbers	200,000	240,000	440,000	760,000	760,000
False positive	02	03	11	07	11344
False negative	0	0	0	0	0
True positive	12798	12797	12789	12793	1456
True negative	7202	7203	7211	7207	18544

using PCA and GA, which consisted of 38 features. Five thousand exemplars or input samples were randomly selected from twenty thousand dataset. Five thousand exemplars contained two types of connections; normal and intrusive, in which $3,223$ were normal and $1,777$ were intrusive. The five thousand dataset was further divided into three subsets; training dataset (2500), cross-validation dataset (1000) and testing dataset (1500). The sensitivity analysis of MLP in terms of true positive, false positive, false negative and true negative are discussed.

4.1 Comparison with existing Approaches

Comparison of the performance of the developed system is done with some other intrusion detection approaches introduced in related work section. SVMs use a Gaussian function for each input sample in the training set. This can slow the training process. The training time of SVM is one hour and sixteen minutes on the dataset of size 564KB whereas MLP takes twenty minutes to train itself on the same dataset. The SVM converges to the optimal solution in 1000 epochs while MLP converge in 173 epochs. The intrusive classification accuracy of MLP is 99.99% that is higher than SVM which was 99.96%[15],99.94%[15],99.60%[11] on the same dataset. Further, SVMs are not suitable for huge dataset. Therefore, MLP is considered a good classifier for intrusion analysis due to its proven ability to handle large data such as traffic data on networks, less number of epochs and time in training process. The selection of principal components is different from previous work [11, 13]. In this work, eigenvalues and eigenvectors are not arranged in descending order because this is necessary step in traditional way of selecting principal components. Therefore, this step is eliminated in this work and the selection is done using genetic algorithm. This process reduced the number of features to ten features as compared to previous approach [13] which has twelve features. Thus, the size of dataset with ten features of $20,000$ samples is reduced to 564KB from 2.17 MB [11, 13]. Table 7 shows the comparative analysis of applied approach with other approaches. The results indicate that adopting MLP based on genetic principal components is a feasible solution that satisfies optimal performance in intrusion detection.

5 Conclusion

In this research, a performance enhancement model is proposed for intrusion detection system based on an optimal feature subset selection using several genetic

Table 7: Performance with existing approaches

Approach(s)	Detection rate (%)
(MLP+GPC-10)[applied approach]	**99.99**
(MLP+FS-12)[13]	99.98
(SVM+GPC-12)[15]	99.96
(SVM+GPC-10)[15]	99.94
(SVM-FS-12)[11]	99.60
(MLP+PCA)[16]	98.57
(GA+SVM) [8]	98
(ART1)[20]	97.42
(ART2)[20]	97.19
(SOM)[20]	95.74
(RBF/Elman) [19]	93
(PCA+NN)[18]	92.2
(SVM) [17]	83.2
(MLP) [17]	82.5

principal components. The feature selection has been accomplished using the techniques of PCA and GA. PCA, in this case, was applied to transform the input samples into a new feature space. Since the selection of an appropriate number of principal components is a critical problem, GA in this research was also used in the optimum selection of principal components instead of using the traditional method. The selected principal components called genetic principal components are the basis of feature subsets. The selected feature subsets are presented to MLP for classification purpose. The KDD-cup dataset used is a benchmark to evaluate the security detection mechanisms. The performance of applied approach was then addressed. Further, a comparative analysis is made with some existing approaches. As a result, this method provides a performance enhancement in intrusion detection, reduces feature subset size and maximizes the detection rates.

Acknowledgement

This work is supported by College of Computer & Information Sciences Research Center, Deanship of Scientific Research, King Saud University, Saudi Arabia. The author is grateful for this support.

References

[1] Z. Sun, G. Bebis, R. Miller, Object detection using feature subset selection, Pattern Recognition, **37**, 2165-2176 (2004).
[2] G. Liu, Z. Yi, S.Yang, A hierarchical intrusion detection model based on the PCA neural networks, Neurocomputing, Advances in Computational Intelligence and Learning,14th European Symposium on Artificial Neural Networks, **70**, 1561-1568 (2006).
[3] S. Horng, S. Ming-Yang, C. Yuan-Hsin, K. Tzong-Wann, C. Rong-Jian, L. Jui-Lin, P. Citra Dwi, A novel intrusion detection system based on hierarchical clustering and support vector machines, Expert Systems with Applications, **38**, 306-313 (2011).
[4] X. Tong, Z. Wang, Y. Haining, A research using hybrid RBF/Elman neural networks for intrusion detection system secure model, Computer Physics Communications, **180**, 1795-1801 (2009).
[5] H. F. Eid, A. Darwish, A. E. Hassanien, and A. Abraham, Principle components analysis and Support Vector Machine based Intrusion Detection System,10th International Conference on Intelligent Systems Design and Applications (ISDA), 363-367 (2010).
[6] L. J. Cao, K. S. Chua, W. K. Chong, H. P. Lee, and Q. M. Gu, A comparison of PCA, KPCA and ICA for dimensionality reduction in support vector machine,Neurocomputing, **55**, 321-336 (2003).
[7] G. Zargar and P. Kabiri, Selection of Effective Network Parameters in Attacks for Intrusion Detection, Advances in Data Mining. Applications and Theoretical Aspects, Lecture Notes in Computer Science, **6171**, 643-652 (2010).
[8] D. Kim, H. Nguyen, O. Syng-Yup and P. Jong Sou, Fusions of GA and SVM for Anomaly Detection in Intrusion Detection System, Advances in Neural Networks, Lecture Notes in Computer Science, **3498**, 415-420 (2005).
[9] I.Ahmad, A. Abdullah, and A. Alghamdi. Application of artificial neural network in detection of DOS attacks. In Proceedings of the 2nd international Conference on Security of information and Networks, SIN .ACM, New York, NY, 229-234 (2009).
[10] Li-Xin Wang, Adaptive fuzzy systems and control: design and stability analysis, Prentice-Hall, Inc. Upper Saddle River, NJ, USA, (1994)
[11] I. Ahmad, A. Abdullah, and A. Alghamdi, M. Hussain, Optimized Intrusion Detection Mechanism Using Soft Computing Techniques, Telecommunication Syst., **52**, 2187-2195 (2013).
[12] I. Ahmad, Feature Subset Selection in Intrusion Detection, LAMBERT Academic Publishing,Germany, (2012).
[13] I. Ahmad, A. Abdullah, and A. Alghamdi, M. Hussain, K. Nafjan, Intrusion Detection Using Feature Subset Selection based on MLP, Scientific Research and Essays, **6**, 6804-6810 (2011).
[14] M. Hussain, SK. Wajid, A. Elzaart, and M. Berbar, A Comparison of SVM Kernel Functions for Breast Cancer Detection,in Computer Graphics, Imaging and Visualization (CGIV), 2011 Eighth International Conference on, 145-150 (2011).
[15] I Ahmad, M Hussain, A Abdullah, A Alelaiwi,Enhancing SVM performance in intrusion detection using optimal feature subset selection based on genetic principal components,Neural Computing and Applications,in press.DOI 10.1007/s00521-013-1370-6.
[16] G. Liu, Y. Zhang, S. Yang, A hierarchical intrusion detection model based on the PCA neural networks. Neurocomputing, **70**, 1561-1568 (2007).
[17] A. Osareh, B.Shadgar,Intrusion Detection in Computer Networks based on Machine Learning Algorithms, International Journal of Computer Science and Network Security(IJCSNS), **8**, 15-23 (2008).
[18] S.lakhina, S.Joseph, B. Verma, Feature Reduction using Principal Component Analysis for Effective Anomaly-Based Intrusion Detection on NSL-KDD, International

Journal of Engineering Science and Technology, **2**, 1790-1799 (2010).
[19] X. Tong, Z. Wang, Y. Haining , A research using hybrid RBF/Elman neural networks for intrusion detection system secure model, Computer Physics Communications, **180**, 1795-1801 (2009).
[20] M. Amini, R. Jalili, H. Shahriari, RT-UNNID: A practical solution to real-time network-based intrusion detection using unsupervised neural networks, Computers Applicaton and Security, **25**, 459-468 (2006).
[21] Tamer Shahwan and Raed Said, A Comparison of Bayesian Methods and Artificial Neural Networks for Forecasting Chaotic Financial Time Series, Journal of Statistics Applications & Probability, **1**, 89-100 (2012)
[22] T. D. X. Duong and V. N. Duong, Principal Component Analysis with Weighted Sparsity Constraint, Appl. Math. Inf. Sci., **4**, 79-91 (2010).
[23] Mary Iwundu and Polycarp Chigbu, A Hill-Climbing Combinatorial Algorithm for Constructing N-Ppint D-OptimaL Exact Designs, Journal of Statistics Applications & Probability, **1**, 133-146 (2012)
[24] M. R. Girgis, T. M. Mahmoud, H. F. Abd El- Hameed and Z. M. El-Saghier, Routing and Capacity Assignment Problem in Computer Networks Using Genetic Algorithm, Information Science Letters, **1**, 13-25 (2013).
[25] Bo Zhang and Zhicai Juan, Modeling User Equilibrium and the Day-to-day Traffic Evolution based on Cumulative Prospect Theory, Information Science Letters, **1**, 9-12 (2013)

Iftikhar Ahmad received the B.Sc. degree in Mathematics and Physics from Islamia University, Bahawalpur, Pakistan, in 1999 and the M.Sc. Computer Science from University of Agriculture, Faisalabad, Pakistan in 2001. He obtained his MS/M.Phil degree in Computer Science from COMSATS Institute of Information Technology, Abbottabad, Pakistan in 2007. He received the Ph.D. degree in IT from Universiti Teknologi PETRONAS, Malaysia. He has published several papers in highly reputed international conferences and journals. His research interests include Soft Computing, Network Security, Intrusion Detection, Analytic Hierarchy Process, and C4I systems.