Intrusion Detection Model using PCA and Ensemble of Classifiers

Sumaiya Thaseen[1] and Ch.Aswani Kumar[2]

[1]*School of Computing Science and Engineering, VIT University, Chennai, Tamil Nadu.*
[2]*School of Information Technology and Engineering, VIT University, Vellore.*

Abstract
Most of the intrusion detection systems examine all network features to identify intrusions with different classification approaches. The major challenges for any intrusion detection model is to achieve maximum accuracy with minimal false alarms. While many ensemble techniques are present to improve the accuracy of intrusion detection models, building an ensemble that can be generically applied for any network traffic is still a difficult task. In this paper, we propose a hybrid model for intrusion detection integrating base classifiers such as SVM, Linear Discriminant and Quadratic discriminant analysis.The aim of this paper is to identify the class label by constructing an individual classifier for each of the attack type and merging the results of every classifier. The resultant decision of the class label is obtained using weighted majority voting approach. We analyzed the performance of the model on two different data sets such as NSL-KDD and UNSW-NB datasets. The experimental results indicate that the ensemble produces high accuracy in comparison to the base classifiers. As there is a huge class imbalance problem in network traffic, it is also observed that rather than relying on a single classifier, predicting the class label by weighted majority voting of SVM, Linear and Quadratic Discriminant classifier is an optimal solution which is proposed in this paper.
Keywords Accuracy; Intrusion; Linear discriminant; Quadratic discriminant; Support vector machine.

1 Introduction

Malicious intruders in the network are increasing day by day due to the rapid development of internet. The intruders can access, manipulate and disable the systems connected on the internet. Intrusion detection systems (IDS) are designed to discover the unauthorized access to computers in the network. Intrusion detection systems are classified in two categories Signature detection and anomaly detection. Signature detection is used to identify attacks based on the known pattern of attacks. Anomaly detection compares unknown profiles with known profiles and then identifies the unknown traffic profile as an attack.

Anomaly detection techniques have high false positive rates. Many machine learning techniques have been used by researchers to overcome the disadvantages of anomaly detection models. Several intelligent approaches such as SVMs [1],

ANNs [2], Petri nets and data mining approaches[3, 4] have been used to build an IDS. There are also many ensemble methods of machine learning which are more efficient than individual techniques that can reduce the false alarm rate and increase the classification accuracy. The different ensemble methods are bagging, boosting and stacking. Bagging and boosting are mostly used to implement intrusion detection models as the stacking technique requires more time.

There are two major limitations of existing approaches. The first limitation is even though there are many sophisticated detection techniques only few focus on feature representation for normal traffic and attack traffic which is a major issue to enhance performance of the classifier. The second issue is the computation time involved in integrating multiple techniques which may degrade the efficiency of on-line detection.

The contribution of this research is to to build a desirable IDS model with high accuracy using machine learning ensemble techniques with a feature reduction technique to identify the suitable features in IDS. Ensemble is preferred because the aggregation of multiple classifier predictions improves the accuracy of IDS.

In this paper we specify a manager as a combination of classifiers that will be generated depending on the class labels in each dataset. For instance, the manager will have 5 different classifiers in the SVM, MNB and LDC ensemble for NSL-KDD data set as there are five class labels and the manager will have 9 different classifiers in the SVM, LDC and QDC for UNSW- NB dataset. Thus the manager has a variable number of classifiers generated dynamically according to the number of class labels available in the dataset. Thus the ensemble model utilizes a individual classifier for every class type and is an integration of base classifiers SVM, Linear Discriminant and Quadratic Discriminant with the resultant class label predicted by Weighted Majority Voting ensemble which is deployed to classify the different kinds of network attacks.

A similar ensemble of classifiers was already developed [5] using four dif-ferent base classifiers namely Linear Discriminant Classifier (LDC), Quadratic Discriminant Classifier (QDC), k-nearest neighbor (KNN) and back propagation and tested on four datasets namely Hearth, Diabetes, Iris and Transfusion. The uniqueness of the proposed model over earlier developed ensemble techniques are 1) The model can be evaluated on any real time intrusion detection datasets and the managers can be extended based on the number of class labels in the samples. 2) Any combination of base classifiers can replace the existing techniques to improve the model.

The rest of this paper is summarized as follows. Section 2 provides a discussion on various developed intrusion detection models. Section 3 provides the background of various techniques utilized in the model such as SVM, Linear Dis-

criminant , Quadratic Discriminant Classifier and weighted majority approach. Section 4 gives the overview of proposed intrusion detection model. Experimental results and discussions are discussed in section 5. Section 6 finally concludes the paper.

2 Related Work

In this section we will discuss the various intrusion detection models developed using machine learning approaches, models developed by integrating classifiers and the ensemble techniques developed for intrusion detection.

Various artificial intelligence methods have been developed for intrusion detection models such as fuzzy logic [6], k-nearest neighbors [7], support vector machines [1], artificial neural networks [2], Naïve Bayes networks [8], decision trees [9], and genetic algorithms [10].

Sumaiya et al. developed an intrusion detection model by using PCA as the dimensionality reduction technique and SVM as the classifier [11, 12]. The kernel parameters of SVM are optimized by considering the variance of samples available in the same and different classes. This model provided a better classification accuracy. Ajith et al. built a light weight IDS using genetic programming approaches [13]. The experimental results proved that the accuracy was better in comparison to traditional intrusion detection models.

Kuanga et al. developed a hybrid KPCA SVM with GA model for intrusion detection [14]. The authors used KPCA to extract the primary features of intrusion detection dataset. SVM multilayer model is employed as the classifier to identify the attack. Chebrolu et al. evaluated the performance of two feature selection techniques such as Bayesian Networks (BN) and Classification and Regression Trees (CART) and also an integration of CART and BN [15]. Results illustrate that feature selection is very effective in the development of real world intrusion detection models. Sandhya et al. developed a hybrid intrusion detection model by combining decision trees and SVM and also an ensemble of other base classifiers [3]. This hybrid model maximized accuracy and minimized complexity and results illustrated that the proposed model provided an accurate IDS.

Perin et al. built a three layer multi classifier intrusion detection model to increase the overall accuracy [16]. The performances were analyzed from a vari-ety of combination techniques such as fuzzy k-NN classifier, na¨ıve bayes classifier and back propagation neural network classifier and the decision obtained from multiple classifiers are combined into a single result. The results proved that the detection performance is better than deploying a single classifier when using a full feature set or partial feature set. Chandra and Yao developed an ensemble based neural network wherein the outputs are combined in a form that resulted in a sig-nificant improvement in the generalization performance [17]. Srinivas et al. built

an ensemble model of SVM, Artificial Neural Network (ANN) and Multivariate Adaptive Regression Splines (MARS) and analyzed the performance which was superior to individual approaches with respect to classification accuracy [18].

Syarif et al. improved the accuracy and false positive rate of intrusion detec-tion by constructing an ensemble of bagging, boosting and stacking [19]. The base classifiers for these ensemble models were na¨ıve bayes, decision tree, rule induction and nearest neighbor. Their results indicated that the accuracy for known intrusions was more than 99% but novel intrusions were identified with accuracy levels of 60%. Thus bagging and boosting did not improve the accuracy significantly whereas stacking decreased the false positive rate by 46%. These ensembles increase the execution time and hence are not practical to be implemented in an IDS. Bahri et al built a ensemble method called Greedyboost and compared with Adaboost and C4.5 [20]. However the base classifier details are not specified in the paper but the results indicated that Greedyboost scores a higher precision and recall in comparison to probe, U2R and R2L attacks present in KDD'99 dataset.

Bukhtoyarov et al designed neural network classifiers by applying a probabilistic approach to the network intrusion detection. Genetic programming based ensembling (GPEN) was deployed to design neural network ensembles [21]. They also analyzed with the KDDcup 1999 dataset and classified the attacks. Cordeiro and Pappa utilized the Particle Swarm Optimization (PSO) by weighing the classifications obtained from different classifiers [22]. The four classification algorithms used were KNN, Naïve Bayes, Rocchio and SVM. They used datasets of users of video social network for classification. Their results outperformed the single classifiers.

The motivation for selecting algorithms in the ensemble is due to the fact that an ensemble based on the four expert algorithms: Linear Discriminant Classifier (LDC), Quadratic Discriminant Classifier (QDC), k-nearest neighbor (KNN) and back propagation. The ensemble is obtained by integrating the experts opinion with a weight coefficient assigned by weighted majority voting is already tested on four widely used datasets of Hearth, Diabetes, Iris and Transfusion. This ensemble resulted in better accuracy in comparison to simple majority voting approach, mean, maximum, minimum and median combiner [23].

Thus many hybrid models using ensemble of techniques for intrusion detection have been developed but the major issue in ensemble approaches is the models were built and tested only on KDD datasets and thus a generic model that can deploy and test for any real time datasets is the necessity for the current scenario.

The proposed model aims to overcome the issues in the existing ensemble approach and also with the advantage of developing modular structures that can have interchangeable positions. Another advantage of our proposed ensemble de-

signs is that the algorithms can be replaced anytime with a more precise one. Hence the approach aims to improve intrusion detection accuracy using simple techniques in ensemble learning integrated with PCA as a dimensionality reduction technique.

3 Background

3.1 Preprocessing

Data preprocessing is very essential for huge data such as network traffic. Reduction of redundant data and normalization are essential to be performed in preprocessing to build a balanced set of data. Normalization is the process of transforming the data within a small specified range. The different normalization techniques are min-max normalization, z-score normalization and normalization by decimal scaling. We select z-score technique because it considers the mean and standard deviation of the attribute.

$$d^1 = \frac{b - mean\,(f)}{std\,(f)} \tag{1}$$

Where ,
mean(f)= sum of all attribute values of f
std(f) = standard deviation of all values of f.

3.2 Dimensionality Reduction

Dimensionality reduction transforms the data in the high dimensional space to a lower dimension. PCA performs a linear mapping of the data to a lower dimension such that the maximum variance for the data is obtained. The procedure begins with the construction of correlation matrix and computation of eigen vectors. The eigen vectors that correspond to the highest eigen values will be deployed to reconstruct the variance of the original data. transformation Thus the original space is reduced to the space obtained by a few eigen vectors. The advantages of using dimensionality reduction are as follows:

- Time and storage space is reduced.
- Improves the performance of the machine learning model.
- Visualization of data is easier as the dimensions are reduced to 2D or 3D.

3.3 Datasets

We have analyzed the knowledge discovery and data mining 1999 standard dataset such as NSL-KDD data set which is widely used as intrusion detection benchmark datasets. The NSL-KDD data set contains roughly 33,300 samples. This dataset is chosen because of the following benefits [24]: 1) No redundant records in the training set. 2) Due to the reduction in the number of records, the complete data

can be used for both training and testing. Each packet in the dataset can be classified in any one of the classes namely normal, DoS, U2R, R2L and probe. All the class labels except normal indicate the different attacks in the dataset.

The other dataset analyzed in our model is UNSW-NB dataset obtained from the cyber range lab of the Australian Center for Cyber Security (ACCS). This dataset contains nearly 1,56,000 samples falling in one of the nine classifica-tion categories namely Normal, Analysis, Backdoor, Reconnaissance, Exploits, Fuzzers, Generic, DoS and Shellcode. This dataset has the advantages of con-taining current attacks in the network domain.

3.4 Support Vector Machine

Support Vector Machines are widely used for classification and regression problems. SVM is preferred over other techniques due to the low generalization error and less over fitting issues that arise from the training data set. There exist a possibility of high generalization error or overfitting if the model doesnt scale well on instances not available in the training set. SVM is very effective on data samples that are separable in a linear fashion. The objective is to identify the hyperplane H that can split the instances into two categories such that samples in one class fall entirely on one side of H. As we can determine unlimited number of candidate hyperplanes, SVM selects only the hyperplane that maximizes distance to the closest data samples in either class. This is known as margin maximization. The major features of SVM are:

- Deals with very large data sets efficiently.
- Multiclass classification can be done with any number of class labels.
- High dimensional data in both sparse and dense formats are supported.
- Expensive computing not required.
- Used in many applications like e-commerce, text classification, bioinformatics, banking and other areas.

There are many real time applications where such a hyperplane does not exist. In such cases, SVM utilizes a function to transform the data into a different feature space such that there is a possibility of separation. The function that performs such a transformation is called as kernel function. Kernels play a major role in SVM. The different kernel functions widely used along with SVM are [25] as given below:

i) Linear Kernel: $K\left(x_i, x_j\right) = x_i x_j$

j) Polynomial Kernel : $K\left(x, x^{'}\right) = \left(x x^{'} + 1\right)^{d}$

k) RBF Kernel: $K\left(x, x^{'}\right) = exp\left(-\gamma \| \ x - x^{'} \ \|^{2}\right)$

l) Sigmoid kernel: $k\left(x_i, x_j\right) = tanh\left(y {x_i}^{t} x_j + r\right) k\left(x_i, x_j\right) = tanh\left(y {x_i}^{t} x_j + r\right)$

SVM can be extended to multi-class classification, a set of binary classifiers are trained one for each class depending on the data set and its respective class labels.ie. If we train the NSL-KDD data set, then let i=15 be a index in the set S=(Normal, Probe, DoS, U2R and R2L) and let B_i denote the matching binary classifier for the target set S. Similarly if we train the UNSW-NB dataset, then let i= $1 \cdots 9$ be a index in the set S=(Normal, Analysis, Backdoor, Reconnaissance, Exploits, Fuzzers, Generic, DoS and Shellcode) and let B_i denote the matching binary classifier for the target set S. Thus the observations are classified using One-Versus-All approach in both the datasets. To distinguish among the binary classifiers, we deploy manager to denote one set of classification. SVM produces the best results when the RBF Kernel function is utilized. Experimental results show that the performance of SVM classifiers will differ with the selection of RBF function. Therefore in this paper we train the SVM manager with five different RBF values= [5, 2, 1, 0.5 ,0.1] for both datasets to ensure that the SVM algorithm is utilized maximally. This approach will ensure greater diversity of managers in ensemble classifier as the accuracy will vary for each binary classifier according to the selected RBF values in the vector.

Construct a set of binary classifiers f^{1} , $\mathrm{f}^{2} \cdots \mathrm{f}^{N}$ for $1 \cdots \mathrm{N}$ classes each trained to differentiate one class from the rest. A multi class categorization can be obtained by combining them according to the maximal output before applying the sgn function.

$$\begin{aligned} &argmax\ g^k(x) \\ &Where\ g^k(x) = \sum_{i=1}^{n} y_i {a_i}^k k(x, x_i) + b^k \\ &Where\ k = 1 \cdots\ N. \end{aligned} \tag{2}$$

wherein $g^k(x)$ returns a signed real value which is the distance from the hyper plane to the point x. This value is referred as the confidence value. The higher the value, the more is the confident that the point x belongs to positive class. Hence we need to assign x to the class having highest confidence value.

Given normal data $\chi = \{x_1, x_2, ...x_m\} \in R^d$ and let r be the radius of the hypersphere and $c \in R^d$ which is the center. The optimization problem can be solved by determining the minimum enclosing hypersphere.

Minimize r^2

Subject to

$$\| \phi(x_j) - c \|^2 \leq r^2, j = 1, ...m \tag{3}$$

$$L(c, r, \alpha) = r^2 + \sum_{j=1}^{m} \alpha_j \{ \| \phi(x_i - c) \|^2 - r^2 \} \tag{4}$$

Setting the derivatives

$$\frac{\delta L\left(c,r,\alpha\right)}{\delta c}=c\sum_{j=1}^{m}\alpha_j\left(\phi\left(x_j\right)-c\right)=0 \tag{5}$$

We can obtain the following equation,
$\sum_{j=1}^{m}\alpha_j=1$ and $c=\sum_{j=1}^{m}\alpha_j\phi\left(x_j\right)$
Hence the equation (4) becomes,

$$L\left(c,r,\alpha\right)=\sum_{j=1}^{m}\alpha_j k\left(x_j,x_j\right)-\sum_{i,j=1}^{m}\alpha_i\alpha_j k\left(x_i,x_j\right) \tag{6}$$

Which is the dual form of equation (4).
The dual form of α can be obtained by solving the optimization problem,
Maximizing,

$$W\left(\alpha\right)=\sum_{i=1}^{m}\alpha_i k\left(x_i,x_i\right)-\sum_{i,j=1}^{m}\alpha_i\alpha_j k\left(x_i,x_j\right) \tag{7}$$

Subject to
$\sum_{i=1}^{m}=\alpha_i=1$ and $\alpha_i\geq 0$, $i=1$ to m
It should be noted that lagrange multiplier can be non-zero only if the inequality constraint is an equality for the solution.
The complementarity conditions are satisfied by the optimal solutions α,(c,γ)

$$\alpha_i\{\parallel\phi\left(x_i\right)-c\parallel^2-r^2\},i=1...m \tag{8}$$

Hence it implies that the training samples x i lie on the surface of the optimal hypersphere corresponding to $\alpha_i>0$.
The decision function becomes,
$f\left(x\right)=sgn\left(r^2-\parallel\phi\left(x\right)-c\parallel^2\right)$
This implies,

$$\begin{aligned}&=sgn(r^2-\phi(x).\phi(x)-2\sum_{i=1}^{m}\alpha_i\phi(x).\phi(x_i)+\sum_{i,j=1}^{m}\alpha_i\alpha_j(\phi(x_i).\phi(x_j)))\\&=sgn(r^2-k(x,x)-2\sum_{i=1}^{m}\phi(x_i)k(x,x_i)+\sum_{i,j=1}^{m}\phi_i\phi_j k(x_i,x_j))))\end{aligned} \tag{9}$$

Thus the aim of obtaining minimum enclosing hypersphere containing all training samples is satisfied.

3.5 Linear Discriminant Classifier

Discriminant Analysis is a classification problem where more than two groups of populations are known a priori and one or more samples from the population are classified according to the characteristics measured.

The assumption is that the population π_i has a probability density function of x which has a mean vector u_i and variance-covariance matrix Σ (similar for all populations). It is specified as

$$f\left(x \mid \pi_i\right) = \frac{1}{\mid \Sigma \mid^{1/2}(2\pi)^{p/2}} \exp\left[-\frac{1}{2}(x-u_i)^{'}\Sigma^{-1}\left(x-u_i\right)\right] \tag{10}$$

We classify to the population for which p i f(x— i) is the highest.

LDA is used when the variance-covariance matrix is not dependent on the population from the available data. In such cases the decision rule is based on the linear score function which is a function of the means for each of our g population u_i and also the variance Ccovariance matrix.

The linear score function is as follows:

$$s_i{}^{L}\left(X\right) = -\frac{1}{2}u_i{}^{'}\Sigma^{-1}u_i{}^{'} + u_i{}^{'}\Sigma^{-1}x + \log P_i = \hat{d_{i0}} + \sum_{j=1}^{p}\hat{d_{ij}}x_j + \log P_i \tag{11}$$

Where

$d_{i0} = -\frac{1}{2}u_i{}^{'}\Sigma^{-1}u_i u_i{}^{'}\Sigma^{-1}$

$d_{ij} = jth$ element of $u_i{}^{'}\Sigma^{-1}$

The far left hand expression represents a linear regression with intercept d_{i0} and regression coefficients d_{ij}.

$d_i{}^{L}\left(X\right) = -\frac{1}{2}u_i{}^{'}\Sigma^{-1}u_i{}^{'} + u_i{}^{'}\Sigma^{-1}x = \hat{d_{i0}} + \sum_{j=1}^{p}\hat{d_{ij}}x_j$

Given a sample unit with measurements x_1,$x_2 \cdot x_p$, the sample unit is classified into the population that has the highest linear score. This is comparable to the population that has the highest membership of posterior probability. Linear score has to be calculated for each class of population and then the assignment of the sample to the population with highest score. But as this function utilizes unknown parameters u_i and Σ these parameters have to be determined from the data.

Hence discriminant analysis requires estimation of the following

Prior probabilities:

$p_i = Pr\left(\pi_i\right); i = 1, 2, ...g$

Population Means: These can be determined by sample vectors.

$u_i = E\left(X \mid \pi_i\right); i = 1, 2, ...g$

Variance-covariance matrix: This can be determined using the pooled variance-covariance matrix.

$$\Sigma = var\left(X \mid \pi_i\right); i = 1, 2, ...g \tag{12}$$

Generally these parameters are determined from training set for which the population membership is already calculated.

Conditional Density Function Parameters

Population Means: i can be determined by replacing in the sample means x_i

Variance-covariance matrix : Let S i represent the sample variance-covariance matrix for the population i. Thus the variance-covariance matrix Σ can be determined by substituting the pooled variance-covariance into the linear score as given below:

$$S_p = \frac{\sum_{i=1}^{g}\left(n_i - 1\right) S_i}{\sum_{i=1}^{g}\left(n_i - 1\right)} \tag{13}$$

To obtain the linear score,

$\hat{s_i^L}\left(X\right) = -\frac{1}{2} x_i sp^{-1} x_i + {u_i}' \Sigma^{-1} x + \log P_i = \hat{d_{i0}} + \sum_{j=1}^{p} \hat{d_{ij}} x_j + \log P_i$

Where,

$\hat{d_{i0}} = -\frac{1}{2} x_i sp^{-1} x_i$ and $d_{ij} = jth$ element of $x_i {s_p}^{-1}$

This is a function of the sample mean vectors, the variance-covariance matrix and prior probabilities for different populations. Thus the expression looks similar to linear regression formula with a term for intercept and a linear combination of response variables with the natural log of the probabilities. Thus the decision rule is to classify the sample item into the population that has the highest calculated linear score.

The steps to identify the class label is as follows

Step 1: Delete one observation from the sample.

Step 2: Compute the discriminant function using the remaining observations.

Step 3: Calculate the discriminant function from step 2 to identify the class label of the observation removed from sample in step 1. Steps 1-3 are repeated for all the samples. Calculate the misclassified observations.

3.6 Quadratic Discriminant Analysis

QDA is very similar to LDA where in the assumption is that the each class measurements are distributed normally. But in QDA there is no such assumption that the covariance of each of the class is identical. If the normality assumption holds true, then the best possible test for a hypothesis that the given measurement from a given class is named as the likelihood test. Assume that there are only 2 groups ($y \epsilon \{0, 1\}$) and the means of each class are specified as $u_y = 0$,$u_y = 1$ and the covariances are defined as $\Sigma_{y=0}$ and $\Sigma_{y=1}$. Then the likelihood ratio will be specified as,

$$Likelihoodratio = \frac{\exp\left(-\frac{1}{2}\left(x-u_{y=1}\right)^{T}\sum_{y=1}^{-1}\left(x-u_{y=1}\right)\right)\sqrt{2\pi \mid \Sigma_{y=1} \mid}^{-1}}{\exp\left(-\frac{1}{2}\left(x-u_{y=0}\right)\right)^{T}\sum_{y=0}^{-1}\sqrt{2\pi \mid \Sigma_{y=0} \parallel}^{-1}} < t \tag{14}$$

For a specific threshold 't'. The sample estimates of the mean vector and variance-covariance matrices will substitute the population quantities in the formula.

3.7 Ensemble approach using WMA

The basic idea of majority voting is that the votes are initialized to each managers opinion. The opinion with the highest votes is selected as the final result. Littlestone and Warmuth [37] have specified that the number of errors can be reduced in an ensemble model by introducing weights to the majority voting technique.

We utilize voting approach in this paper as given in [37]. Each manager is initialized with a weight obtained from managers accuracy in classifying the sample. As each manager based on the dataset contains different number of binary classifiers B_i, we need to consider each manager's opinion for every class i separately. Thus we can divide the manager's opinion into two categories:

- Managers which classify given sample as an object of class i (output value 1)
- Managers which assert that the given observation fits to some other class than i. (output value 0).

The voting approach is repeated for each sample x and for each binary classifier inside the manager. This results in an ensemble manager one for each class. We define a set of weight coefficients w as a 15 element vector for NSL-KDD dataset where each element j represents weight for jth manager in ensemble ie. $w = (w_1, w_2...w_{15})$ and similarly w as a 27 element vector for UNSW-NB dataset. ie. $w = (w_1, w_2...w_{27})$.

To obtain the final decision function, we consider the weights used in the voting approach. For every single observation x, we obtain fifteen output values $(y_1, y_2...y_{15})$ for NSL-KDD dataset and twenty seven output values $(y_1, y_2...y_{27})$ for UNSW-NB dataset, one output value per manager. Each value can be a positive or negative,i.e $y_j = \{-1, 1\}$ where value 1 correspond to managers output 1 and negative value -1 represents managers output 0. The final decision is evaluated by the equation given below

$$y = sgn\left(\sum_{i=1}^{n} w_i y_i\right) \tag{15}$$

Where n= $1 \cdots 15$ for NSL-KDD dataset and n=$1 \cdots 27$ for UNSW-NB dataset. Each coefficient w_i is multiplied with output from ith manager y_i and the final decision is determined by the sign of the sum of weight coefficients for all managers.

4 Proposed Model

The proposed model is an integrated intrusion detection system combining PCA as the dimensionality reduction technique and a hybrid model of base and ensemble classifiers. In stage 1, the raw data is sent to a preprocessing unit for performing normalization by z-score technique and the noisy attributes are removed using dimensionality reduction. The resultant subset is then fed to the stage 2 which is the ensemble layer for classification. Three classifiers are deployed in stage 2 namely SVM. Linear and Quadratic Discriminant Classifiers. Thus the ensembled approach is a merger of different supervised classifiers. A 5-fold cross validation is performed to split the data into training and testing sets. The class label is obtained by majority voting from the three classifier results. Fig 1 depicts the proposed intrusion detection model.

In the next subsection we discuss the approach for obtaining optimal subset using PCA and ensemble approach for classification of network traffic label.

Algorithm for obtaining the optimal feature subset using PCA:

Input(Training Set, Test Set)

Output(Optimal Training Set, Optimal Test Set)

Step 1: Determine the size of training and test data

Step 2: Scale the training and test data

Step 3: Subtract the mean for each row

$$m = \frac{\sum_{k=1}^{n} x_k}{n} \tag{16}$$

Wherein x specifies the individual elements and 'n' denotes the no. of samples.

Step 4: Determine the covariance matrix

$$C = \frac{X^I X^{I^T}}{n} \tag{17}$$

Where X represents the matrix after subtracting the mean and X^T is the transpose matrix and n is the total number of elements.

Step 5: Determine the eigenvectors and eigenvalues of the covariance matrix.

$$\Sigma v = \lambda v \tag{18}$$

Step 6: Obtain a feature $vector = \left(eig_1, eig_2 ... eig_p\right)$ where eig_1 is principal component and $p \leq n$. Select 'm' such eigen vectors that match to the largest

'm' eigenvalues in the set.

Algorithm for obtaining the class label using hybrid model

Given: Classifier M_1,(SVM)M_2,(LDC),M_3(QDC), Ensemble(WMA)

Input: Optimal attribute dataset D

Output: Class label

Step1: Initialize all the weights in D. $W_i = 1/n$, where n is the total number of elements.

Step2: For every sample data d_i

Fit the SVM classifier to (x_t, y_t) using weights w_i

For each class label $k = 1...k$ obtain the hypothesis

$x_s \leftarrow argmin\,(\lambda \mid f\,(x_j) \mid + (1 - \lambda))$

$max \frac{k(x_i,x_j)}{\sqrt{k(x_i,x_j)k(x_i,x_j)}}$

$D \leftarrow D \bigcup \{x_s\}$

Label(D)

$L \leftarrow L \bigcup \{S\}$

Step 3: For every sample data d_i

i) Compute the sample estimates $\hat{\pi_m}$,$\hat{u_m}$,$\hat{\Sigma}$

ii) Make two transformations: Sphere the data points based on factoring $\hat{\Sigma}$ and project to the subspace by the centroids.

Thus a transformation of $A \epsilon R^{p(K-1)}$

iii) Given any data, $x \epsilon R^p$ transform to $\bar{x} = Ax \epsilon R^{K-1}$

$\bar{x} = Ax \epsilon R^{K-1}$ and classify according to class $m = 1...K$ for which

$\frac{1}{2} \| \tilde{x} - \tilde{u_m} \|^2 - \log \hat{\pi^m}$ is lowest

Where $\tilde{u_j} = A\hat{u_j}$

Step 4: For every sample data di, Perform QDA by repeating the process in step 3 but with different and for each class and compute the class label.

Step 5: The class label is predicted by weighted majority voting from results of steps (2), (3)and (4) given in equation (15)

Step 6: Determine the performance of the model and analyze the accuracy and time complexity of the different algorithms.

4.1 Class label prediction using Weighted Majority Approach (WMA)

In this paper, we first deploy SVM, LDC and QDC classifiers individually to result in a good generalization performance. After passing through each of the individual classifiers, a majority voting technique is adopted to predict the resultant class label from the different classifier labels.

The major advantage of using ensemble approach is that the performance is improved because the approach selects only the class label which are correctly identified by all the classification techniques. The efficiency of existing approaches is compared by deploying a validation set to obtain weights for each classifier. All managers assign initial weight with value 1 and all the managers weights

are combined using WMV. The predicted class label is then compared with the target label present in the validation set. If the manager has made a mistake in identifying the class label then the weights will be subtracted by a learning factor β. Learning factor is user defined and the values range between 0 and 1. This process is repeated for every sample in the dataset. After each test in which a mistake arises the sum of the weights is at most 'u' times the sum of the weights before the test W_{start} for specific $u < 1$. If the initial eight is W_{begin} and the final weight is W_{fin} , then $W_{begin}u^f \geq W_{fin}$ must be true where 'f' is the number of faults.

$$f \leq \frac{\log\left(W_{begin} - W_{fin}\right)}{\log\left(1/u\right)} \tag{19}$$

Then semble approach overcomes the difference in misclassification. The overall complexity of the model is o(n^4 *m) where n is the number of attributes and m is the number of training instances. LDC performs computation in O(n), QDC performs computation in O(n^2) and SVM performs computation in O(nm) time.

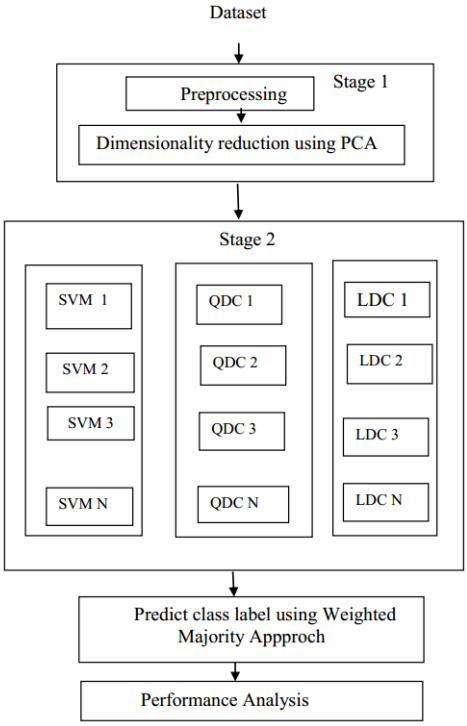

Fig. 1 Proposed intrusion detection model

5 Experiment Setup and Results

The experimental study was conducted with NSL-KDD and UNSW-NB datasets as discussed in the section 3.2. The experiments utilized Matlab-2013b installed on windows 7 ultimate 64-bit machine.

A 5-fold cross validation is performed on all the classifiers for splitting the dataset into 5 non repeated subsets for training, validation and testing. In this paper we compare the efficiency of the proposed algorithms with the metrics namely accuracy rate and elapsed time.

- Accuracy: Number of test instances correctly classified by the model.
- Elapsed time : Time to complete the detection by a classification approach.

An important advantage of intergrating complementary classifiers is to improve accuracy and generalization performance. We assume the accuracy of each classifier separately. The ensemble approach is compared by considering the average score from each classifier. We analyzed the efficiency of each classifier in the base and ensemble. Thus the managers accuracy is defined by

$$E_i = \frac{\sum_{i=1}^{n} A_i}{n} \tag{20}$$

Where 'n' is the number of classes.

5.1 Study 1: NSL-KDD dataset

The initial step is to perform preprocessing on the data and dimensionality reduction using PCA to remove the noisy attributes in the traffic that do not contribute for classification. Table 1 shows the features retrieved after dimensionality reduction on the NSL-KDD dataset. The symbolic features and the features with less variance are removed from the dataset. Experimental results for each manager separately for SVM, LDC, QDC and ensemble managers on the NSL- KDD datasets are specified in tables 2-5 respectively. The results show that by deploying different classifiers for each class, higher accuracy is achieved for all classes namely normal, DoS, probe, U2R and R2L which is highlighted in each table. It is also shown that in comparison to base classifiers and ensemble managers, the latter results in high accuracy above 99% for all the class labels. Time required for classification for each of the managers is presented in table 6. The time consumption for the ensemble WMA is relatively higher in comparison to base classifiers but it can be neglected as the accuracy is high. Figs 2-5 represent the accuracies of SVM, LDC, QDC and ensemble WMA managers respectively. It is evident that WMA produces highest accuracy in comparison to all base classifiers though individually SVM, LDC and QDC produces optimal results for a single or two class label only.

Table 1 Features selected by PCA in NSL-KDD dataset

22 features selected after dimensionality reduction in NSL-KDD data set.	Service, Dst_bytes, dst_host_diff_srv_rate, flag, dst_host_serror_rate, dst_host_srv_count, same_srv_rate , dst_host_same_srv_rate, serror_rate, src_bytes, dst_host_srv_diff_host_rate, host, dst_host_rerror_rate duration, srv_diff_host_rate, dst_host_srv_rerror_rate er ror_rateprotocol_type, srv_rerror_rate, is_guest_login, srv_count, num_compromised.

Table 2 Accuracy obtained using SVM manager with different RBF Kernel in NSL-KDD dataset

Manager	Normal	Probe	DoS	U2R	R2L
SVM 1 (RBF:5)	92.9	98.08	74.41	90.08	87.5
SVM 2 (RBF:2)	91.93	97.79	51.42	84.1	100
SVM 3 (RBF: 1)	90.39	99.8	85.71	85.2	88.88
SVM 4 (RBF:0.5)	89.73	97.78	71.82	100	83.3
SVM5 (RBF:0.2)	99.54	97.94	56.25	99.2	89.47

Table 3 Accuracy obtained by linear discriminant analysis in NSL-KDD dataset

Manager	Normal	DoS	Probe	U2R	R2L
LDC 1	98.87	99.83	82.05	96.99	99.12
LDC 2	93.22	97.79	94.12	86.77	99.65
LDC 3	96.58	98.56	100	99.45	98.56
LDC 4	99.58	99.6	86.48	100	81.81
LDC 5	70.4	93.92	71.84	55.86	97.42

Table 4 Accuracy obtained by quadratic discriminant analysis in NSL-KDD dataset

Manager	Normal	Probe	DoS	U2R	R2L
QDC 1	92.898	97.94	82.75	83.81	89.83
QDC 2	91.73	98.28	86.29	86.81	93.32
QDC 3	87.17	97.55	67.92	91.71	92.21
QDC 4	90.59	97.92	94.05	87.44	92.45
QDC 5	89.83	97.8	78.98	86.02	95.31

Table 5 Accuracy obtained by WMA managers in NSL-KDD dataset

Manager	Normal	Probe	DoS	U2R	R2L
WMA 1	98.93	99.47	94.61	99.45	99.11
WMA 2	78.77	94.47	94.59	82.78	99.6
WMA 3	78.89	94.31	88.1	82.36	84.85

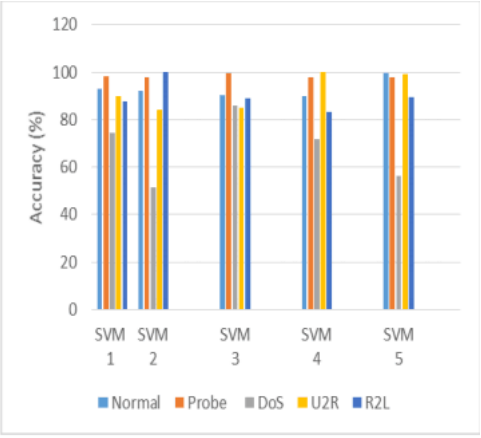

Fig. 2 Accuracy of SVM managers in KDD dataset

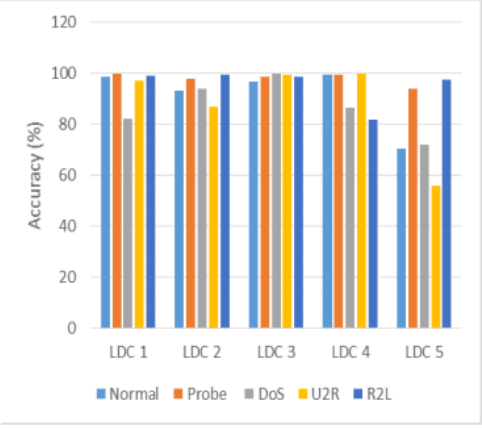

Fig. 3 Accuracy of LDC managers in NSL-KDD dataset

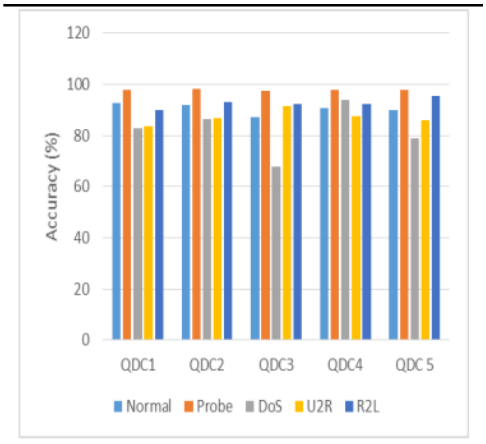

Fig. 4 Accuracy of QDC managers in NSL-KDD dataset

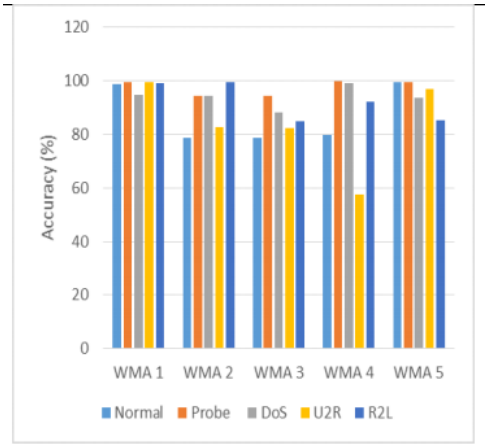

Fig. 5 Accuracy of WMA managers in NSL-KDD dataset

Table 6 Elapsed time for managers in NSL-KDD

Managers	Elapsed Time (Secs)
SVM	31.96 s
LDC	63.95 s
QDC	66.09 s
WMA	53.36 s

5.2 Study II: UNSW-NB Dataset

After preprocessing on the dataset, dimensionality reduction using PCA is performed and the results are shown in table 7. Experimental results for each manager separately for SVM, LDC, QDC and ensemble manager on the UNSW-NB datasets are specified in tables 8- 11 respectively. The results show that by deploying unique classifiers for each class higher accuracies are obtained for all the class labels namely normal, analysis, backdoor, exploits, reconnaissance, fuzzers, exploits, DoS and shellcode respectively. The highest accuracy for all the class labels are obtained using ensemble manager higher than the base classifiers as weighted majority approach reduces the misclassification rate. Time required for classification for each of the managers is presented in table 12. The time consumption for the ensemble WMA is relatively higher in comparison to base classifiers but as the dataset is huge the time span is relatively huge in comparison to the NSL-KDD dataset.

Figs 6-9 represent the graphical accuracies using SVM, LDC, QDC and WMA managers for the UNSW-NB dataset respectively. It is inferred that SVM performs poorly for attacks namely reconnaissance, DoS and Analysis with accuracy of 35%, 29% and 40% respectively. LDC performs poorly for attacks such as shellcode, reconnaissance, DoS and Fuzzers with accuracy of 40%,36%,38% and 28% respectively. QDC performs poorly for backdoor, analysis, exploits, shellcode and reconnaissance with accuracy of 34%,33%,30% and 33% respectively whereas WMA results in an average accuracy for reconnaissance,exploits, shellcode and fuzzers more than 50% which is a considerable increase in comparison to base classifiers and other class labels also produce higher accuracy.

Table 7 Attributes retrieved after dimensionality reduction using PCA in UNSW-NB dataset.

21 attributes selected after PCA	Id, dur, service, state, spkts, dpkts,sbytes,dbytes, rate, sttl, dttl, sload, dload, sloss, dloss, sinpkt, dinpkt, sjit, djit, swin, stcpb, dtcpb, dwin, tcprtt, synack, ackdat, smean, dmean, trans_depth, response_body_len, ct_srv_src, ct_state_ttl, ct_dst_ltm, ct_src_dport_ltm, ct_dst_sport_ltm, is-2_login, ct_2_cmd, ctflw_4_mthd, ct_src_ltm, ct_srv_dst, is_sm_ips_ports.

Table 8 Experimental results using SVM manager in UNSW-NB dataset

Manager	Normal	Analysis	Backdoor	Reconna issance	Exploits	Fuzzers	Generic	DoS	Shellcode
SVM 1	99.94	66.21	59.14	90.28	86.76	91.35	99.61	91.35	91.26
SVM 2	51.66	60.94	50.81	71.89	64.86	98.33	84.06	76.66	86.66
SVM 3	69.14	93.79	45.81	48.21	56.17	99.72	86.89	56	78.87
SVM 4	99.03	39.28	45.95	60.79	87.76	53.05	98.4	70.2	27.83
SVM 5	93.76	43.33	65.51	25.97	47.81	57.88	72.64	58.62	43.13
SVM 6	82.34	34.76	62	30.39	52.22	63.57	87.37	80.76	73.22
SVM 7	99.55	31.33	47.36	33.55	75.49	42.67	99.26	47.42	42
SVM 8	96.19	32.15	47.07	33.4	52.54	33.84	98.83	28.95	34.18
SVM 9	99.65	79.63	63.73	70.78	87.38	96.28	99.65	93.97	87.51

Table 9 Experimental results using LDC manager in UNSW-NB dataset

Manager	Normal	Analysis	Backdoor	Reconnaissance	Exploits	Fuzzers	Generic	DoS	Shellcode
LDC 1	87.91	53.49	54.15	75.07	67.88	63.91	99.09	71.95	39.61
LDC 2	86.41	50.22	52.95	36.07	84.34	55.66	94.71	54.16	84.94
LDC 3	98.67	75.14	62.59	36.84	65.45	56.08	92.36	38.67	50
LDC 4	97.59	80.11	92.3	32.98	65.79	42.87	99.26	46.25	41.37
LDC 5	96.59	89.2	90.44	50.89	84.26	47.55	63.43	35.26	36.74
LDC 6	94.22	76.56	85.66	36	75.44	94.59	80.82	48.21	66.99
LDC 7	87.56	44.7	59.53	37.39	55.4	28.41	99.57	38.39	43.27

Table 10 Experimental results using QDC manager in UNSW-NB dataset

Manager	Normal	Analysis	Backdoor	Reconnaissance	Exploits	Fuzzers	Generic	DoS	Shellcode
QDC 1	82.11	46.41	77.41	26.38	48.33	50	65.21	67.1	67.16
QDC 2	97.86	28.67	25.03	35.8	60.34	57.06	80.68	75.49	58.76
QDC 3	86.85	32.98	34.03	33.42	83.7	60.34	59.06	85.68	50
QDC 4	91.73	43.43	52.76	36.34	46.27	51.37	94.03	77.36	30.17
QDC 5	84.21	83.33	63.37	42.19	51.76	45.83	80.94	74.15	42.23
QDC 6	81.24	35.82	81.51	24.4	52.8	53.95	87.59	81.92	41.95
QDC 7	88.39	40.52	78.94	49.49	38.58	54.98	87.58	68.85	64.19
QDC 8	87.48	44.45	58.95	35.43	54.54	66.31	81.54	91.66	50.63
QDC 9	88.25	46.7	63.85	36.72	62.33	62.83	96.43	82.48	53.56

Table 11 Experimental results using WMA manager in UNSW-NB dataset

Manager	Normal	Analysis	Backdoor	Reconnaissance	Exploits	Fuzzers	Generic	DoS	Shellcode
WMA1	99.99	78.41	86.71	65.97	76.66	91.35	91.85	58.94	86.66
WMA1	98.51	51.95	64.71	71.89	68.84	98.33	90.88	50.46	91.26
WMA 3	88.93	52	62.09	48.21	71.14	99.72	81.45	90.75	78.87
WMA 4	99.41	50.8	100	60.79	73.97	53.05	97.65	‘100	57.83
WMA 5	93.2	67.32	84.61	90.28	59.9	57.88	90.19	67.67	53.13
WMA 6	54	100	50	60.39	46.84	63.57	84.32	68.8	73.22
WMA 7	95.25	55.89	57.09	63.55	62.67	42.67	99.15	58.8	62
WMA 8	81.71	74.72	50.22	63.4	88.21	63.84	91.4	69.34	64.18
WMA 9	99.21	58.67	65.62	70.78	87.92	96.28	84.13	64.96	87.51

Table 12 Elapsed time for managers in UNSW-NB data set

Managers	Elapsed Time (Secs)
SVM	464.82 s
LDC	485.56 s
QDC	622.50 s
WMA	751.61 s

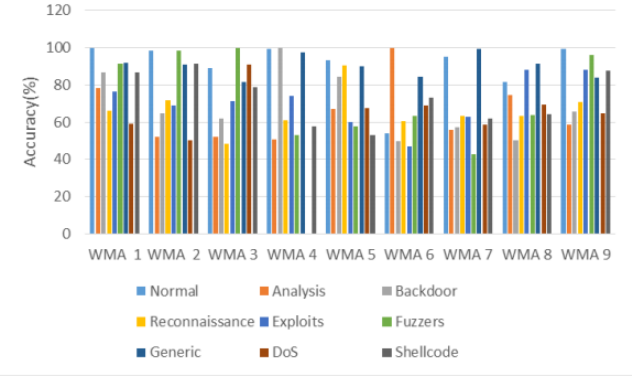

Fig. 6 Proposed intrusion detection model

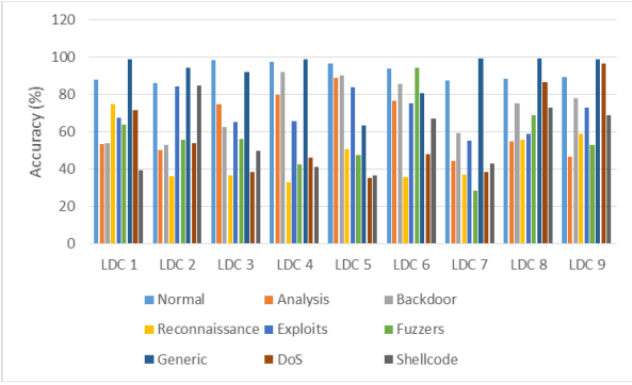

Fig. 7 Proposed intrusion detection model

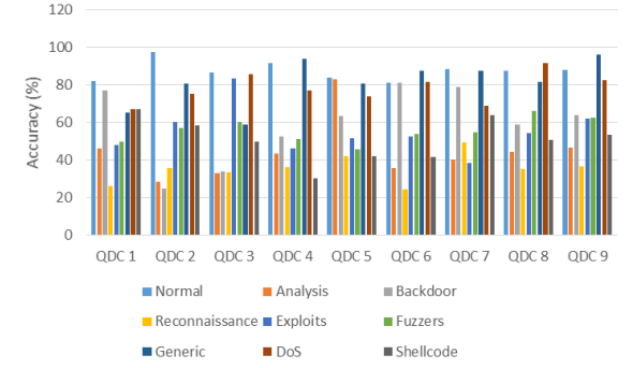

Fig. 8 Proposed intrusion detection model

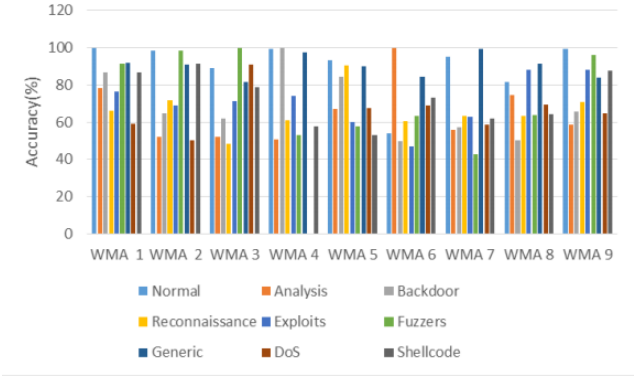

Fig. 9 Proposed intrusion detection model

5.3 Discussion

In this paper, we have analyzed the accuracy of the base and ensemble classifier managers. It is to be noted that the ensemble managers accuracy is relatively high in comparison to the base classifiers and managers accuracies are above 99% for NSL-KDD dataset and the range is from 88-100% for UNSW-NB dataset. The variation is due to the imbalance of samples in the dataset for the classes exploits, shellcode and reconnaissance. The reason for the high accuracy in both the datasets are 1) proper selection of training data 2) well-selected parameters in WMA technique namely the learning factor and weight assignment w_i for each sample. 3) Misclassification error is reduced due to weight assignment to each sample.

There were poor results obtained for LDC and QDC manager only in the UNSW-NB dataset. But SVM performs well for NSL-KDD dataset and in specific for class labels normal, fuzzers, generic and DoS in the UNSW-NB dataset. SVM produced poor accuracy of 85% for DoS and QDC produced poor accuracy of 91% and 95% for U2R and R2L attacks in the NSL-KDD dataset. The accuracy was as low as 49.49% for reconnaissance class label because the test data contains samples that were not utilized while training the classifiers. Thus in comparison to WMA, the base classifiers SVM, LDC and QDC perform poorly and also with increase in elapsed time. The constraint on the weights generated by WMA also is a major factor in improving the accuracy of the intrusion detection model. The weights can lie between 0 and 1, the higher the value the more is the possibility of correctly predicted.

Thus this model can be extended to include other ensemble techniques but it should consider the time complexity as many ensemble approaches complete the task in extremely long time.

6 Conclusions

The classification accuracy can be improved with minimal elapsed time by integrating opinions from multiple managers into single using an ensemble approach. We have deployed weighted majority voting to integrate results from different managers. The three base classifiers namely SVM, LDC and QDC were experimentally compared using two different datasets namely NSL-KDD and UNSW-NB. Thus WMA results in good accuracy for both the datasets.

The accuracy improvement is of 1% in comparison to base classifiers. Thus the success of the model is due to the generated weights in WMA which were tuned by the learning factor. Thus the integration of base classifiers in to an ensemble manager with WMA will be very well utilized in intrusion detection as it has been tested on recent datasets with modern day attacks. The other improvement in our approach is instead of relying on binary classification techniques, we have utilized multi class classification for the base managers.

In future we can integrate optimization techniques with WMA to tune the parameters for generating weights.

References

[1] Khan L., Awad M. and Thruaisingham B. (2007), "A new intrusion detection system using support vector machines and hierarchical clustering", *The VLDB journal*, Vol. 16, pp. 507-521.

[2] Wang G., Hao J., Ma J. and Huang L. (2010), "A new approach to intrusion detection using artificial neural networks and fuzzy clustering, Expert Systems with Applications", Vol. 37, pp. 6225-6232.

[3] Sandhya Peddabachigiri, Ajith Abraham, Crina Grosan and Johnson Thomas. (2007), "Modeling intrusion detection system using hybrid intelligent systems", *Journal of Network and Computer Applications*, Vol. 30, pp. 114-132.

[4] Lee W, Stolfo S and Mok K. (1999), "A data mining framework for building intrusion detection model", *In: Proc. of IEEE symposium on security and privacy*, pp. 120-32.

[5] A. Kausar, M. Ishtiaq, M.A.Jaffar and A.M.Mirza. (2010), "Optimization of ensemble based decision using PSO", *in: Proceedings of the World Congress on Engineering*, Vol. 10.

[6] Tsang C.H., Kwong S. and Wang H. (2007), "Genetic fuzzy rule mining approach and evaluation of feature selection techniques for anomaly intrusion detection", *Pattern Recognition*, Vol. 40, pp. 2373-2391.

[7] Li Y. and Guo L. (2007), "An active learning based TCM-KNN algorithm for supervised network intrusion detection", *Computer and Security*, Vol. 26, pp. 459-467.

[8] Amor N.B., Benferhat S., Elouedi Z.and Zhang X.T. (2004). *Naïve Bayes vs decision trees in intrusion detection systems*, SAC04:Proceedings of the 2004 ACM Symposium on Applied Computing, New York,NY,USA: ACM Press.

[9] Xiang C., Yong P.C. and Meng L.S. (2008),"Design of multiple-level hybrid classifier for intrusion detection system using Bayesian Clustering and decision trees" , *Pattern Recognition Letters*, Vol. 29, No. 7, pp. 918-924.

[10] Shafi K. and Abbass H.A. (2009), "An adaptive genetic based signature learning system for intrusion detection", *Expert Systems with Applications*, Vol. 36, No. 10, pp. 12036-12043.

[11] Sumaiya Thaseen and Ch.Aswani Kumar. (2014), " Intrusion detection model using fusion of PCA and optimized SVM", *2014 International Conference on Computing and Informatics (IC3I) 27-29 Nov 2014*, pp. 879-884.

[12] I.Sumaiya Thaseen and Ch.Aswani Kumar. (2016), "Improving accuracy of intrusion detection model using PCA and optimized SVM", *CIT Journal of Computing and Information Technology*, Vol.24, No.2 ,pp. 133-148

[13] Ajith Abraham, Crina Grosan and Carlos Martin-Vide. (2007), "Evolutionary design of intrusion detection programs", *International Journal of Network Security*, Vol. 4, No. 3, pp. 328-339.

[14] Fangjun Kuanga, Weihong Xua and SIyang Zhang.(2014). "A novel hybrid KPCA and SVM with GA model for intrusion detection", *In Applied Soft Computing* , Vol. 18, pp. 178-184.

[15] Srilatha Chebrolu, Ajith Abraham and Johnson P.Thomas. (2005), " Feature deduction and ensemble design of intrusion detection systems", *Computers and Security*, Vol. 24, pp. 295-307.

[16] Alexandre Balon-Perin. (2012), "Ensemble-based methods for intrusion detection", *Norwegian University of Science and Technology.*

[17] Chandra A. and Yao X.(2006), "Evolving hybrid ensembles of learning machines for better generalization", *Neurocomputing*, Vol. 69, No. 7, pp. 686-700.

[18] Srinivas Mukkamala, Andrew H. Sung, Ajith Abraham. (2007). "Intrusion detection using and ensemble of intelligent paradigms", *Journal of Network and Computer Applications*, Vol. 28, No. 2, pp. 167-182.

[19] I.Syarif, E.Zaluska, A.Prugel-Bennett and G.Wills. (2012), "Application of bagging, boosting and stacking to intrusion detection", *in: Machine Learning and Data Mining in pattern recognition*, pp. 593-602.

[20] E. Bahri, N.Harbi and H.N.Huu. (2011), "Approach based ensemble methods for better and faster intrusion detection", *in: Computational Intelligence in Security for Information Systems*, pp. 17-24.

[21] V.Bukhtoyarov and V.Zhukov. (2014), "Ensemble-distributed approach in classification problem solution for intrusion detection systems", *in: Intelligent Data Engineering and Automated Learning-IDEAL 2014*, pp. 662-668.

[22] Cordeiro Junior, Z. and G.L.Pappa. (2011), "A PSO algorithm for improving multi-view classification", *in: 2011 IEEE Congress on Evolutionary Computation (CEC)*, pp. 925-932.

[23] A.Kausar, M.Ishtiaq, M.A.Jaffar and A.M.Mirza. (2010), "Optimization of ensemble based decision using PCO", *in Proceedings of the World Congress on Engineering, WCE*, Vol. 10.

[24] http://nsl.cs.unb.ca/NSL-KDD/

[25] S.J.Horng, M.Y.Su, Y.H.Chen, T.W.Kao, R.J.Chen, J.L.Lai and C.D.Perkasa. (2011), "A novel intrusion detection system based on hierarchical clustering and support vector machines", *Expert Systems and Applications*, Vol. 38, No. 1, pp. 306-313.

Corresponding author
Sumaiya Thaseen can be contacted at: sumaiyathaseen@gmail.com