CIT. Journal of Computing and Information Technology, Vol. 24, No. 2, June 2016, 133–148
doi: 10.20532/cit.2016.1002701

Improving Accuracy of Intrusion Detection Model Using PCA and Optimized SVM

Sumaiya Thaseen Ikram[1] and Aswani Kumar Cherukuri[2]

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

Intrusion detection is very essential for providing security to different network domains and is mostly used for locating and tracing the intruders. There are many problems with traditional intrusion detection models (IDS) such as low detection capability against unknown network attack, high false alarm rate and insufficient analysis capability. Hence the major scope of the research in this domain is to develop an intrusion detection model with improved accuracy and reduced training time. This paper proposes a hybrid intrusion detection model by integrating the principal component analysis (PCA) and support vector machine (SVM). The novelty of the paper is the optimization of kernel parameters of the SVM classifier using automatic parameter selection technique. This technique optimizes the punishment factor (C) and kernel parameter gamma (γ), thereby improving the accuracy of the classifier and reducing the training and testing time.

The experimental results obtained on the NSL-KDD and gurekddcup dataset show that the proposed technique performs better with higher accuracy, faster convergence speed and better generalization. Minimum resources are consumed as the classifier input requires reduced feature set for optimum classification. A comparative analysis of hybrid models with the proposed model is also performed.

ACM CCS (2012) Classification: Security and privacy → Intrusion/anomaly detection and malware mitigation → Intrusion detection systems

Keywords: cross validation, dimensionality reduction, intrusion detection system, principal component analysis, radial basis function kernel, support vector machine

1. Introduction

Intrusion Detection Systems (IDS) are developed to identify unauthorized attempts to access or manipulate the computer systems. IDS collects network data to identify different kinds of malware and attacks against services and applications. IDS has been classified into two major categories, namely signature based detection and anomaly based detection. In signature based IDS, attack pattern of intruders are modeled and the system will notify once the match is identified. All known attacks are identified with reduced false positive rate. Signature databases have to be updated frequently so as to identify the new attack pattern. However, anomaly detection systems create a profile of normal activity. Any pattern that deviates from the normal profile is treated as an anomaly. Hence even unknown attack patterns are identified without any manual intervention.

Data mining techniques have been used recently in the development of intrusion detection models to minimize information overloading. These models extract the useful knowledge by searching for patterns and relationships from the data collected, thereby improving decision making. Data mining technologies such as neural networks [1], naïve bayes networks [2], genetic algorithms [3], fuzzy logic [4] and support vector machine [5] are used for classification and pattern recognition in many industries as they

Extended version of the paper "Intrusion Detection Model Using Fusion of PCA and Optimized SVM" previously presented at International Conference on Computing and Informatics (IC3I), held on November 27–29, 2014, in Mysore, India.

have improved the performance of the models that deploy such algorithms. In classification, the features of newly present objects are examined and are assigned to one of the existing set of classes. Classifier models gain knowledge from the training data and identify the class label for the new instances. Many supervised learning models are used to solve classification problems.

Support Vector Machine (SVM) is one of the efficient techniques used as the generalization capability is higher even when the sample training data is small. In the recent years, many hybrid intelligent systems have been proposed to improve the accuracy in comparison to individual techniques.

Anomaly detection models have the difficulty of "curse of dimensionality" which is a very important issue. To overcome this issue, an optimal feature subset has to be obtained to improve accuracy and remove noise. In the proposed model, a SVM classifier is combined with PCA for identifying the anomalies. PCA is one of the extensively used statistical techniques to reduce the dimensionality and SVM has the advantage of achieving fine performance for the classification of abnormal patterns.

The remainder of the paper is organized as follows: Section 2 discusses various machine learning techniques, SVM techniques used in various intrusion detection models and recent hybrid techniques developed integrating SVM and dimensionality reduction. The background of various techniques used in the model is discussed in Section 3.The proposed methodology is discussed in Section 4. The experiments and results of the model are reported in Section 5. Section 6 contains the conclusion.

2. Related Work

In this subdivision, we analyze the literature about traditional intrusion detection models, intrusion models using machine learning techniques, intrusion models using SVM classifiers and integrated intrusion models using SVM and dimensionality reduction techniques.

Mahoney and Chan [6] developed an Application Layer Anomaly Detector that considers low level traffic features and payload for identifying the anomalies. Mahoney and Chan [7] also proposed learning rules for anomaly detection which learns the rules by considering network traffic rather than employing a predetermined set of rules. The system is unique because of the following aspects: the model uses a large number of attributes in addition to user behavior. The system is non stationary in nature, which specifies that the time event is significant but frequency is not. The model efficiently determines less number of rules with huge possibilities. The limitation of the model is that it has not been tested on a live environment and it is understood that no attack traffic is present in the training set. This would not be applicable in a real time environment.

Weijun [8] employed SVM with normalization for intrusion detection. Min-Max normalization method was chosen as it produces better performance, cross validation accuracy and increased number of support vectors. The authors indicate that normalization is a crucial stage in preprocessing to reduce calculation time and improve the performance of a classifier. SVM without normalization will increase calculation time and results in many support vectors. Carlos et al. [9] used autonomous labeling approach for normal traffic to deal with imbalance of class distribution which is not appropriate for SVM. The major advantage of this technique is that, under some attack distributions, it has superior results over SNORT [10].

Aswani [11] analyzed different dimensionality reduction techniques in the preprocessing stage. Some of the supervised techniques analyzed are Linear Discriminant Analysis (LDA), Maximum Margin Criterion (MMC) and Orthogonal Centroid (OC). Unsupervised approaches such as singular value decomposition (SVD) assign original data to a new dimension without considering label information. Sumaiya and Aswani Kumar [12] analyzed various unsupervised tree based classifiers for intrusion detection system wherein different classifier models along with feature selection resulted in an optimized record set that determined normal or anomaly type of the packet.

Kuang et al. [13] integrated Kernel Principal Component Analysis (KPCA) and Support Vector Machine (SVM) for intrusion detection. Multi-layer classifier model is employed

to determine if any action results in attack. The classification accuracy of the proposed model is superior to the models employing SVM classifiers where parameters are randomly selected and hence better generalization performance. KPCA can investigate higher order original input information and obtains large quantity of principal components. Sumaiya and Aswani Kumar [14] [15] [16] integrated various feature selection techniques along with SVM to build a hybrid intrusion detection model which had a better accuracy and reduced false alarm rate. Srinoy [4] employed particle swarm optimization (PSO) to extract intrusion features and classify using SVM. Horng et al. [17] integrated hierarchical clustering along with SVM to provide few abstracted and high quality training instances. This model reduces training time and improves the performance.

Sandhya et al. [18] developed a hierarchical hybrid intelligent system model integrating decision tree, support vector machines and an ensemble of various base classifiers. The hybrid model maximized the detection accuracy and reduced the complexity of computations. The major advantage of the ensemble approach is the fact that information from different classifiers is merged to decide whether an intrusion has occurred or not. The effectiveness of the model is based on the diverse nature of the base classifiers. Eskin [19] designed an unsupervised anomaly detection model and implemented it using three unsupervised techniques such as clustering, k-nearest neighbor and Support Vector Machine. The advantage of these algorithms is that they can be applied to any feature space to model diverse kinds of data. The model can be extended to more feature maps with various kinds of data and perform widespread experiments on the data. Li et al. [20] proposed an efficient intrusion detection method combining clustering, ant colony algorithm and SVM that resulted in an efficient model that determined whether a network data packet is classified as normal or abnormal. However, the model had a strategy of selecting a small training data set which may not be suitable to multiple classification problems. Bhavesh et al. [21] developed a hybrid technique to identify anomalies by integrating Latent Dirichlet Allocation (LDA) and genetic algorithm (GA). LDA determines the subset of attributes for anomaly identification. GA module computes the initial score of samples and breeding. It then evaluates the fitness criteria to construct a new generation. The hybrid technique was applied on a generic data set which resulted in reduction in accuracy, but if the model had been trained with specific anomaly dataset, then the accuracy could have been higher. Shih et al. [22] constructed an intelligent intrusion detection model wherein the goal was to combine SVM, Decision Tree (DT) and Simulated Annealing (SA). SVM and SA techniques can identify the optimal selected features to improve the accuracy level of intrusion detection. The best parameters for both DT and SVM are tuned without human intervention by SA. The intelligent technique can efficiently discover attacks with their appropriate types.

Liu et al. [23] built an anomaly model which performed feature selection by PCA and classification by neural networks. Only 22 features were extracted from the 38 feature set. Principal components selected were based on the highest eigenvalues. This technique minimized the total number of features and increased the detection rate. However, the approach of selecting the principal components is not globally optimal as a certain subset of principal components are only investigated. The increased generalization performance of PCA is obtained with the trade--off of large amount of computation time. Han et al. [24] investigated the effective nature of SVM by identifying masquerade behavior using various UNIX commands. Experiments revealed that SVM is a successful technique for masquerade identification. However, there is a trade-off between variety and efficiency of features analyzed, including session and time information.

The recent intrusion detection approaches discussed in the literature are as follows: Mamalakis et al. [25] deployed two different machine learning techniques, namely support vector machines and guassian mixture models to build a host based anomaly detection system and the experiments were conducted on real world data sets collected from three different web servers and a honeynet. The authors evaluated the model and proved that the results were very effective to other state-of-the-art file system based IDS. Ravale et al. [26] combined data mining approaches such as K-Means clustering and Radial Basis Function (RBF) kernel of sup-

port vector machine as a classifier technique for intrusion detection. The proposed model achieves better results in terms of detection rate and accuracy. John et al. [27] constructed a network intrusion detection model by choosing the Extreme Learning Machine (ELM) as the major learning algorithm and integrated it with multiple kernel boosting. The model was named as MARK-ELM. The IDS was tested on several machine learning datasets including KDD-Cup-99 datasets and results indicated that the approach works well for majority of the University of California, Irvine (UCI) repository datasets. The model performs better with lower detection rates and lower false alarm rates in comparison with other traditional approaches on intrusion detection data. Eduardo et al. [28] developed an intrusion detection model using Principal Component Analysis (PCA) for feature selection and removal of noise and integrated it with Self Organizing Map (SOM) to distinguish between normal and anomalous connections. This resulted in faster implementation of intrusion detection models.

Hence, from the literature, it is very clear that most of the intrusion detection systems deployed SVM with feature reduction techniques. Therefore, in this paper, we integrate SVM with dimensionality reduction techniques such as PCA for building an intrusion detection model. The novelty of this approach is to optimize the SVM parameters such as kernel parameter (γ) and punishment factor (C) using an automatic parameter selection technique which can improve the classification rate and detect the intrusions in a faster manner.

Any real time application that requires intrusion detection monitoring can deploy this approach as these systems are able to identify the intruders on the internet whose purpose is to breach the network and make it vulnerable. The proposed model can be deployed as a network based intrusion detection model to secure industrial networks and filter out all the intruder traffic trying to enter the network.

3. Background

In this section we briefly review the data mining techniques that are employed in our proposed model.

3.1 Scaling

Huge volumes of network traffic have to be processed for identifying the anomalies and hence classification may not be accurate. Therefore, data packets undergo normalization technique wherein data is sanitized. The purpose of normalization is to record the data to a diverse scale. Different techniques used for normalization are Z-score, Decimal and Min-Max scaling. The Min-Max normalization technique is chosen for the proposed model as it has less number of misclassification errors [29] compared to other techniques. Min-Max normalization achieves a linear modification on the original data. Normalization is carried out for a given range. To perform mapping for a value v of a feature f within range [$\min_f$, $\max_f$] to a new range [new_$\min_f$, new_$\max_f$], the calculation is given by

$$v' = \frac{(v - \min_f)(new\_\max_f - new\_\min_f) + new\_\min_f}{\max_f - \min_f} \quad (1)$$

where v' is the new value in the specified range. The benefit of this technique is that all values are concealed within certain ranges.

3.2 Principal Component Analysis (PCA)

PCA is one of the broadly used statistical techniques in the field of data mining to reduce dimensionality and to identify data points with the highest possible variance [30] [31]. Lakhina et al. [32] employed PCA to differentiate network traffic data into normal and anomalous sub regions. In this method, the focus is on detection of volume based anomalies in origin-destination flow aggregated in backbone networks and it is a vital component within several IDS systems today. The PCA approach identifies anomalous traffic volume on a particular link by comparing it with past values. Thus, PCA separates link traffic measurements into sub regions representing normal and abnormal traffic. The outcome of the PCA is to project a feature space onto a smaller subspace that represents data by reducing the dimensions of feature space. This reduces computational costs and the error of parameter estimation.

The standard PCA approach can be summarized in six simple steps:

(i) Determine the covariance matrix of the normalized *d*-dimensional dataset.

(ii) Determine the eigenvectors and eigenvalues of the covariance matrix.

(iii) Sort the eigenvalues in descending order.

(iv) Select the *k* eigenvectors that correspond to the *k* largest eigenvalues where *k* is the number of dimensions of the new feature subspace.

(v) Construct the projection matrix from the *k* selected eigenvectors.

(vi) Transform the original dataset to build a new *k*-dimensional feature space.

3.3 Support Vector Machine Classification Model

Support Vector Machines (SVMs) are a set of supervised learning techniques mainly used for classification, outlier detection and regression. The major advantages of SVM are:

- Efficient results in high dimensional spaces
- Helpful wherein the quantity of dimensions is higher than the quantity of data samples
- Efficient usage of memory as SVM uses only a subset of training points in the decision making function
- Different kernel functions can be used for the decision function. Common kernels are available, but we can also develop custom kernels [33]

3.3.1 Linear Support Vector Machine

Consider the categorization of two classes that can be separated in a linear fashion as shown in Figure 1 [34]. Figure 1 shows that the hyperplane for the linear classifier is of the form $(\mathbf{w}.\mathbf{x} + b) = 0$ having the maximum margin (both expressions $\langle \mathbf{w},\mathbf{x} \rangle$ and $\mathbf{w}.\mathbf{x}$ denote the scalar product of two vectors, i.e. represent the same operation). The classifier is described by the set of pairs $(\mathbf{w}, b)$, where $\mathbf{w}$ is a weight vector and b is the bias, that can specify the inequality for any sample $\mathbf{x_i}$ in the training set, while y_i represents the class label:

$$\begin{aligned} \mathbf{w}.\mathbf{x_i} + b \geq +1 \text{ if } y_i = +1 \\ \mathbf{w}.\mathbf{x_i} + b \leq -1 \text{ if } y_i = -1 \end{aligned} \tag{2}$$

Here, where $\mathbf{w}$ is normalized with respect to a set of points $\mathbf{x}$ such that: $\min_i | \mathbf{w}.\mathbf{x_i} | = 1$. Minimizing $\| \mathbf{w} \|^2$ subject to equation (2) and representation of constraints in a compact form are as follows:

$$\begin{aligned} y_i(\mathbf{w}.\mathbf{x_i} + b) \geq +1 \\ y_i(\mathbf{w}.\mathbf{x_i} + b) - 1 \geq 0 \end{aligned} \tag{3}$$

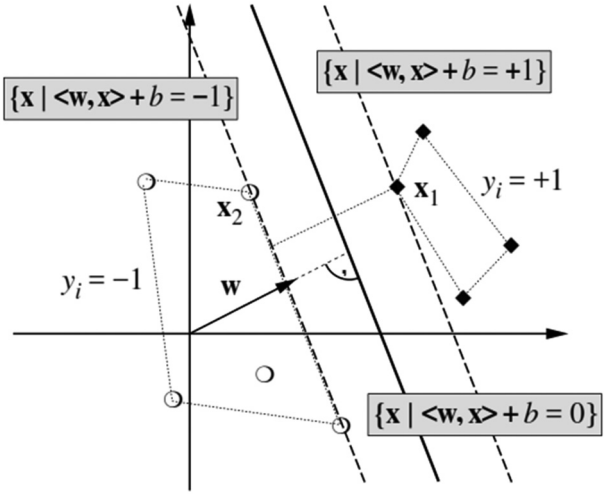

Figure 1. Linear support vector machine [34].

Every hyperplane $(\mathbf{w}, b)$ is a classifier that separates all patterns from the training set.

To deal with non separable case, the problem is rewritten as:

Minimize

$$\| \mathbf{w} \|^2 + C \sum_{i=1}^{m} \xi_i \tag{4}$$

(where C is a regularization parameter: small C allows constraints to be easily ignored, large C makes constraints hard to ignore) with respect to

$$y_i(\mathbf{w}.\mathbf{x_i} + b) \geq 1 - \xi_i, \text{where } \xi_i \geq 0 \tag{5}$$

Hence the decision function is of the form

$$f(\mathbf{x}) = \mathbf{w}.\mathbf{x} + b \tag{6}$$

The fundamental equation is obtained by

$$\min P(\mathbf{w}, b) = \frac{1}{2} \| \mathbf{w} \|^2 + C \sum_i H_1(y_i f(\mathbf{x_i})) \tag{7}$$

where $P(\mathbf{w},b)$ represents the primal formulation to minimize the training error, and H_1 would identify the number of errors.

3.3.2 Non-linear Support Vector Machine

Linear classifers are not complex. The preprocessing of data is done with:

$$\mathbf{x} \rightarrow \Phi(\mathbf{x})$$

Then it changes to the form $\Phi(\mathbf{x})$ to y:

$$f(\mathbf{x}) = \mathbf{w}.\Phi(\mathbf{x}) + b \tag{8}$$

The dimension of $\Phi(\mathbf{x})$ can be very large. Thus $\mathbf{w}$ will be hard to represent precisely in memory space and also difficult to solve the quadratic problem.

The theorem shows that

$$\mathbf{w} = \sum_{i=1}^{m} \alpha_i \Phi(\mathbf{x_i}) \tag{9}$$

We can optimize α (which represents the lagrange multiplier) directly instead of optimizing $\mathbf{w}$.

The decision rule is modified as:

$$f(\mathbf{x}) = \sum_{i=1}^{m} \alpha_i \Phi(\mathbf{x_i}).\Phi(\mathbf{x}) + b \tag{10}$$

We specify the term $K(\mathbf{x_i}, \mathbf{x}) = \Phi(\mathbf{x_i}).\Phi(\mathbf{x})$ as the kernel function.

We can rewrite all SVM equations with equation (9).

Hence the decision function becomes

$$f(\mathbf{x}) = \sum_{i} \alpha_i \Phi(\mathbf{x_i}).\Phi(\mathbf{x}) + b \tag{11}$$

$$= \sum_{i} \alpha_i K(\mathbf{x_i}, \mathbf{x}) + b \tag{12}$$

The dual formulation is obtained by:

$$\min P(\mathbf{w},b) = \frac{1}{2} \| \sum_{i=1}^{m} \alpha_i \Phi(\mathbf{x_i})) \|^2 + C \sum_{i} H_1[y_i f(\mathbf{x_i})] \tag{13}$$

where $P(\mathbf{w},b)$ represents the primal formulation to minimize the training error of the SVM.

Kernel function K(. , .) is used to build implicit non linear feature map,

(i) Polynomial Kernel: $K(\mathbf{x}, \mathbf{x_i}) = (\mathbf{x}.\mathbf{x_i} + 1)^d$

(ii) RBF Kernel: $K(\mathbf{x}, \mathbf{x_i}) = \exp(-\gamma \| \mathbf{x} - \mathbf{x_i} \|^2)$

3.3.3 RBF – SVMs

The RBF Kernel $K(\mathbf{x}, \mathbf{x_i}) = \exp(-\gamma \| \mathbf{x} - \mathbf{x_i} \|^2)$ is one of the most widely used kernel functions.

$$f(\mathbf{x}) = \sum_{i=1}^{m} \alpha_i \exp(-\gamma \| \mathbf{x_i} - \mathbf{x} \|^2) + b \tag{14}$$

where $\| \mathbf{x_i} - \mathbf{x} \|^2$ is the squared Euclidean distance between two vectors and σ is an unrestricted parameter. It is defined as

$$\gamma = \frac{1}{2\sigma^2}$$

3.4 Cross Validation

Cross validation is a model evaluation technique that uses a partial data set for training which will be used by the learner. Certain records are removed before the training begins. The data already removed is used as a test set to measure performance of the model on the new data.

In K-fold cross validation, the data set is divided into k subsets. In every iteration, one of the k subsets is considered as the test set and the other $k - 1$ subsets are considered as the training set. The advantage of this method is that the data point will be in the test set at least once and in the training set $k - 1$ times.

4. Proposed Work

In this section we propose a novel hybrid model for intrusion identification.

4.1 Proposed Methodology

The proposed model integrates PCA with SVM for classification of anomalies in the network traffic. The SVM parameters such as the punishment factor (C) and kernel parameter gamma (γ) are optimized to obtain higher levels of accuracy. Figure 2 illustrates the block diagram of the proposed system. The proposed approach employs two stages: In the first stage, PCA finds an optimal subset of all attributes by removing noisy information from attributes

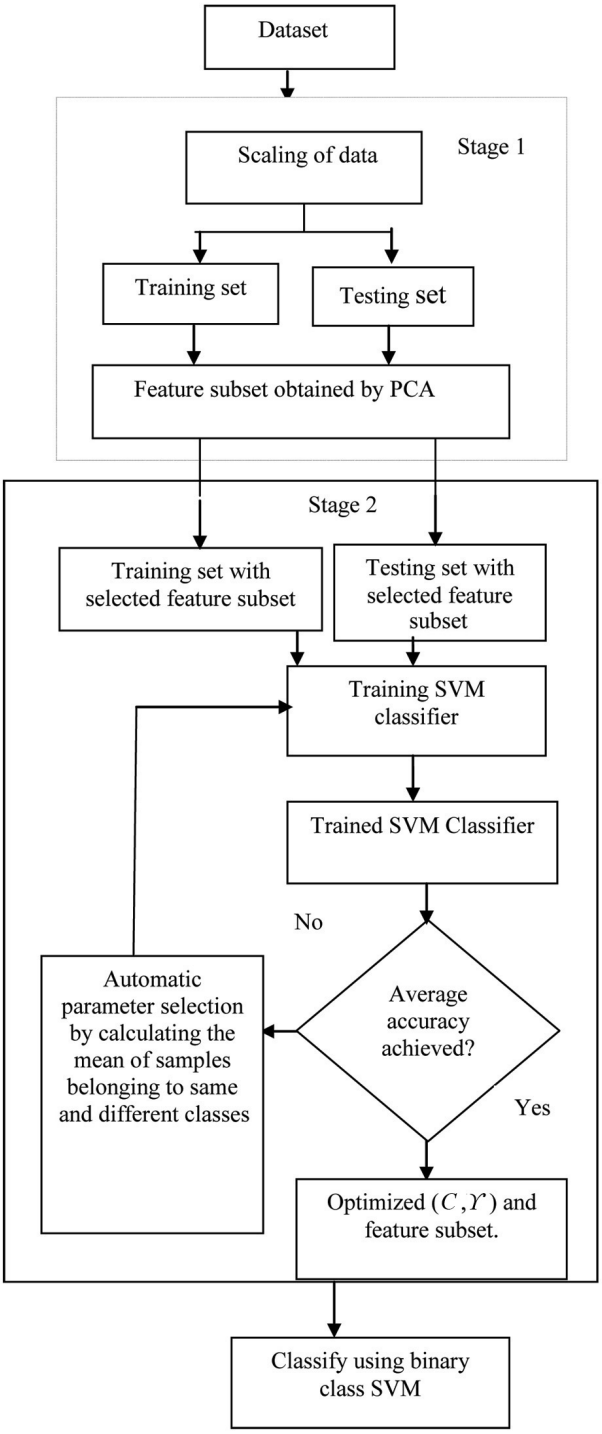

Figure 2. Proposed intrusion detection model.

that contain it. Variance threshold is usually set to a higher value so that the cumulative variance of the various principal components falls above the threshold and can be selected to form the feature vectors. The second stage uses the optimal subset obtained from PCA as training data set and test data set for SVM to perform classification. RBF kernel is adopted in this model and the optimal parameters of SVM are obtained using grid search with automatic parameter selection as depicted in Figure 3.

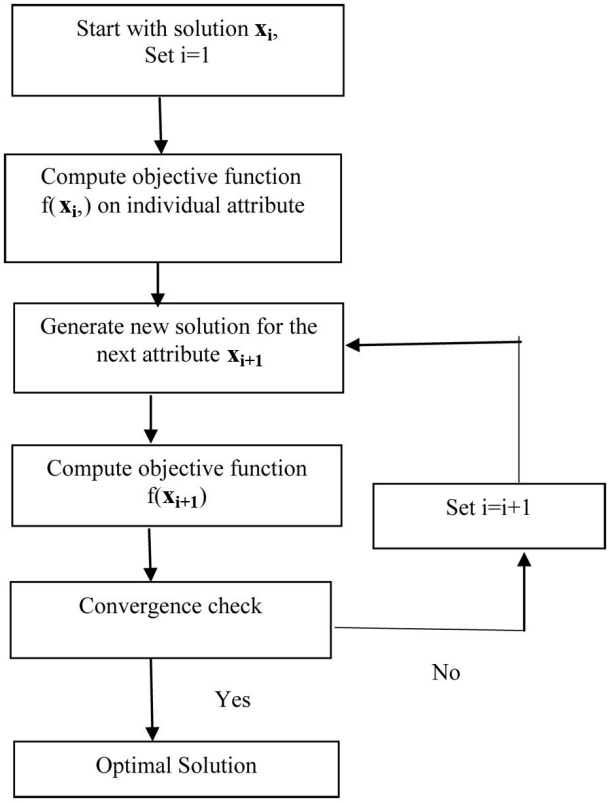

Figure 3. Optimization of SVM parameters using automatic parameter selection technique.

An initial set of C and Υ values are specified in the objective function to check whether the parameter combination converges to a best optimal subset. If the subset converges, then the parameter combination is retained; otherwise, the process is repeated to obtain a best optimal subset value.

In the next section, we discuss the methodology for obtaining the optimal subset from PCA and optimization performed to obtain the optimal SVM kernel parameters for classification.

4.2 Preprocessing of Dataset

This paper analyzes the National Scientific Laboratory–Knowledge Discovery and Data Mining (NSL-KDD) dataset [35] and gurekddcup [36] for the experiments. Preprocessing is performed by normalization of the discrete attributes into continuous ones by Min-Max technique on both datasets. Every network data has

41 attributes wherein 34 attributes are continuous and 7 attributes are discrete in nature. Preprocessing is performed on the dataset and then the data is divided into training and test sets.

4.3 Principal Component Analysis

Figure 4 shows the step by step analysis of feature vector generation using PCA. A normalized feature matrix is obtained after preprocessing and is fed as input to obtain the mean and covariance of the individual features. Eigenvectors are generated for every feature and the highest eigenvalues and respective eigenvectors are retained (in the vector set) to obtain the optimal feature subset.

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 from each feature **x** its respective mean *m*

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

wherein x_k specifies the individual element of **x** and *n* denotes the number of elements.

Step 4: Determine the covariance matrix **C**

$$\mathbf{C} = \frac{\mathbf{X}_{(m)}\mathbf{X}_{(m)}^{T}}{n}$$

where $\mathbf{X}_{(m)}$ represents the feature matrix after subtracting the respective means, $\mathbf{X}_{(m)}^{T}$ is the transpose matrix, and *n* is the total number of elements.

Step 5: Determine the eigenvectors $\mathbf{v}_j$ and eigenvalues λ_j of the covariance matrix **C**

$$\mathbf{C}\mathbf{v}_j = \lambda_j \mathbf{v}_j \qquad j = 1,..., p, \quad p \le n$$

Step 6: Obtain a feature vector using a set of eigenvalues (λ_1, λ_2 … λ_p) and respective eigenvectors, where λ_1 is the highest eigenvalue. Select *k* such eigenvectors that match the largest *k* eigenvalues in the set.

The computational complexity of the PCA is $O(p^2n + p^3)$ *p* is the number of features and *n* is the number of data points. Covariance matrix computation is $O(p^2n)$ and the corresponding eigenvalue decomposition is $O(p^3)$.

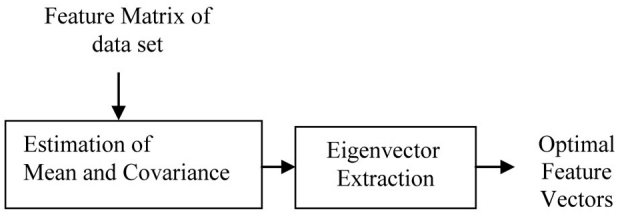

Figure 4. Optimal feature subset generation using PCA.

4.4. Support Vector Machines

Figure 5 shows the stages of SVM classifier for predicting the class label of the network traffic. The stages are divided into two phases: Training and prediction. In training phase, the feature matrix is fed in to the classifier model to identify the class label. The testing phase obtains the learning rules from training phase to identify the pattern of the unknown traffic.

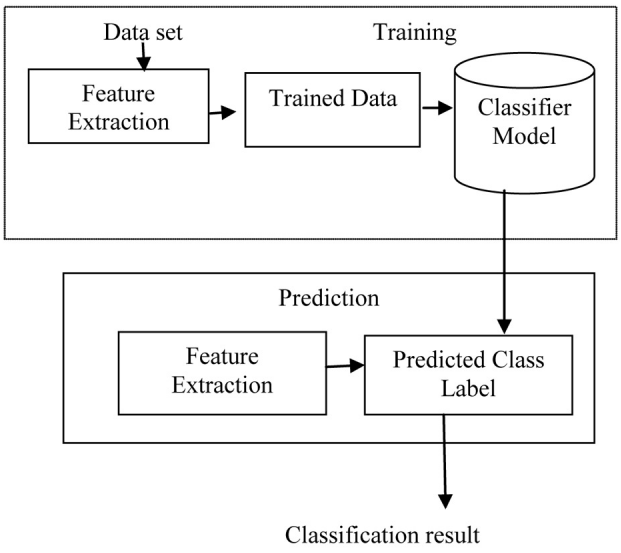

Figure 5. Predicting the class label using support vector machine.

Algorithm for obtaining the optimal kernel parameters of SVM and classification of network data

---

Input (C, Υ, Training set, Test set)

Output (Optimized C, Υ, predicted test label, accuracy)

Step 1: Specify an objective metric for calculating score using mean of the attribute values.

Step 2: Build composite estimators whose parameter space can be searched.

Step 3: Evaluate the parameter settings after computing an objective function on the parameter C and Υ. If the objective function converges, then that parameter combination is selected.

Step 4: Train the SVM classifier with optimized C and Υ.

Step 5: Predict the test data label.

Step 6: Determine the performance metrics using the equations (15), (16) and (17).

The computational complexity of the support vector machine is $O(n_{\text{features}} \cdot n^2_{\text{samples}})$ where n is the number of data elements.

4.5. Optimization of SVM Parameters

The SVM parameters possibly in the kernel can have a major influence on the outcome of the training and misclassification and hence it is very natural that these have to be fine-tuned to improve performance. The basic approach is to control the punishment factor penalty weight (C) and also to identify the best trade-off between misclassification errors and generalization. A higher value of C leads to hard margin SVM behavior whereas lower value of C increases misclassifications. When using kernel other than linear kernel, there might be some tunable parameters. The most common strategy is to use a Radial Basis Function (RBF) kernel and perform optimization on gamma parameter (Υ) along with C.

As cross validation is very time consuming, the major parameters C and Υ of RBF kernel of SVM are optimized by a method that includes searching or sampling candidate parameters and calculating a score function. If the score function is higher than the threshold limit, then the parameter combination is selected as the best combination for usage in the SVM classifier. This method is similar to grid search wherein we choose a starting point x and a step size s.

Then the optimal parameters are deployed in training the model. This parameter optimization avoids misclassification of training sample.

5. Experiments

5.1. Experimental Setup

5.1.1. Datasets

The standard datasets used for intrusion detection are deployed in the proposed model which are the variants of the KDD-Cup-99 data set namely NSL-KDD data set containing 33,300 samples and 10% of gurekddcup data set containing 1,78,810 records. A description of NSL-KDD cup data set and gurekddcup datasets can be obtained from [17] and [32]. NSL-KDD dataset is chosen because of the following advantages: i) It does not include redundant records in the training set, hence the classifiers will not be biased towards frequent records. This results in a better detection rate. ii) The number of selected records from each group is inverse to the number of records in the dataset. iii) Significant reduction in the redundancy of data records allows the experiments to run on the complete set of training and testing data without selecting a random sample data set.

A portion of gurekddcup database is chosen for analysis because the entire database is too huge. Both datasets contain a total of 41 attributes in which the features are divided into intrinsic type, content type and traffic type. Each pattern of the NSL-KDD data set falls into any one of the following classes, namely, Normal and four different kinds of attacks such as Probe, Denial of Service (DoS), User to Root (U2R) and Remote to Local (R2L), but in gurekddcup the class attribute can take any one of the following values such as *normal*, *format_clear*, *ffb_clear*, *load_clear*, *perl_clear*, *dict_simple*, *teardrop*, *guest*, *land*, *ftp-write*, *imap*, *syslog*, *phf*, *ffb*, *multihop*, *warez*, *warezmaster*, *warezclient*, *rootkit*, *spy*, *format*, *loadmodule*, *eject*, *perlmagic*, *format-fail*, *anomaly*, *dict* and *eject-fail*. Apart from the normal class value, all the other class labels indicate the different kinds of attacks in the dataset.

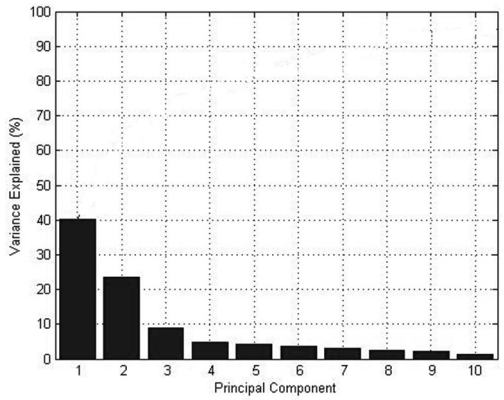

Figure 6. Variance estimation of the principal components.

The dataset is very large and high-dimensional in nature. Hence both datasets undergo dimensionality reduction using PCA. Figure 6 shows the variance estimation of the ten principal components obtained for both datasets as they contain the same 41 features. The principal components having a cumulative variation of more than 90% are retained and the remaining ones are discarded on the basis of cumulative variation which is more than 90% whereas the remaining components are discarded because of the remaining 8.91% variation which is negligible. The components with higher variance will be retained in the parameter selection results after dimensionality reduction using PCA.

Table 2 shows the different attributes selected by the dimensionality reduction technique, namely PCA. The basic ten features removed by PCA contribute to significantly less variance and hence the remaining 31 attributes selected are shown in Table 2.

5.1.2. Classifier Design

The final step of the proposed model is based on the optimization of SVM kernel parameter and punishment factor using the validation set and then training and testing the model to obtain a better accuracy and reduced false positive rate.

The datasets are split into 10 non duplicated subsets and any nine of the subsets will be used for training the model and the remaining one for testing. This is termed as 10-fold cross validation. Hence the SVM classifier will be trained and then tested for 10 intervals.

The cross validation accuracy before optimization is 85%, with an initial value of C and Υ being 1. After many iterations of varying C and keeping Υ constant and vice versa, the best accuracy rate of 98% is achieved when C and Υ is 4.This result is achieved nearly after 600 iterations of modifying the kernel parameter values. Table 3 shows certain iterations of the automatic parameter selection technique.

5.1.3 Evaluation Methods

In this paper, we consider the detection rate (*DR*), false alarm rate (*FAR*) and correlation coefficient (*CC*) which are mostly used in literature to estimate the performance of intrusion detection. They can be determined from the confusion matrix, as given in Table 1. The values obtained in Table 1 are as follows:

True Positives (*TP*): Total instances of anomalies correctly classified as anomalies

True Negatives (*TN*): Total number of normal instances correctly classified as normal

False Positives (*FP*): Total number of normal instances falsely classified as anomalies

False Negative (*FN*): Total number of anomalies wrongly classified as normal instance

Table 1. Confusion Matrix.

	Predicted (normal)	**Predicted (attack)**
Actual(normal)	*TP*	*FN*
Actual(attack)	*FP*	*TN*

The derived metrics obtained from the confusion matrix are as follows:

(i) Detection rate $\left(DR\right) = \dfrac{TN}{TN + FP}$ (15)

(ii) False Alarm rate $\left(FAR\right) = \dfrac{FN}{TN + FP}$ (16)

(iii) Correlation Coefficient: $(CC) =$

$$\frac{TP \cdot TN - FP \cdot FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \quad (17)$$

These parameters are essential in evaluating the performance of the intrusion detection model. The other parameter *CC* indicates the correlation among forecast result and the actual result

ranging from –1 to 1 where 1 specifies the estimated result and is stable with the real calculation and –1 is based on a random calculation.

5.2 Study I: The 41 and 31 Dimensional NSL-KDD Cup Dataset

Study I is based on the 41 and 31 attributes selected from the NSL-KDD cup dataset. Table 6 shows the confusion matrix of the NSL-KDD cup dataset using the entire dataset without dimensionality reduction wherein the accuracy levels of the Probe, DoS, R2L, U2R and Normal are 85.65, 46.41, 33.33, 87.19 and 97.38 respectively in terms of percentage. It is evident that the accuracy levels of the various attack categories are very small. Hence dimensionality reduction is performed using PCA and the results are summarized in Table 7. Table 7 shows the confusion matrix of NSL-KDD dataset after optimization wherein the accuracy levels of Probe, DoS, U2R , R2L and Normal are 90.84, 91.22, 80.43, 78.35 and 63.62 respectively in terms of percentage. Hence it is very evident that the accuracy level of majority attacks such as DoS and Probe are high in comparison to the minority attacks such as U2R and R2L. This is due to the fact that the number of training samples of U2R and R2L are very small in comparison to Normal, Probe and DoS samples.

Table 4 shows the performance metrics of the proposed model on NSL-KDD dataset after dimensionality reduction. The performance metrics of two different experiments on the same data set before and after optimization are obtained. It is evident that the accuracy of the classifier improves when the kernel parameter and punishment factor are tuned using automatic parameter selection technique. The other metrics evaluated are Precision, Recall and F-Score. Precision is the ratio of the number of relevant records classified to the total number of irrelevant and relevant records. Recall is the ratio of the number of relevant records classified to the total number of records in the dataset. F-Score is a statistical technique for determining accuracy based on both precision and recall. Table 5 shows the extended performance metrics of the proposed model. Figure 7 shows the performance metrics of the proposed model before and after optimization. The metrics obtained before optimization have a decreased level of accuracy, precision and recall whereas the metrics after optimization have a constant increase in accuracy, precision and recall. The x-axis represents the different metrics and y-axis represents the accuracy levels. Hence the values fall in a straight line.

Table 2. Features selected from PCA.

Content Features	Hot, num_failed_logins, logged_in, Num_compromised, root_shell, Su_attempted, num_root, num_file_creation, Num_shells, num_access_files, num_outbound_cmds, is_host_login, is_guest_login
Traffic Features	Count, srv_count, serror_rate, srv_serror_rate, rerror_rate, same_srv_ rate, diff_srv_rate, srv_diff_host_rate, dst_host_count, dst_host_srv_count, dst_host_same_srv_ rate, dst_host_diff_srv_rate, dst_host_same_src_port_rate dst_host_srv_diff_host_rate, dst_host_serror_rate, dst_host_srv_serror_rate, dst_host_reeror_rate, dst_host_srv_rerror_rate, dst_host_rerror_rate, dst_host_srv_rerror_rate

Table 3. Accuracy obtained by automatic parameter selection technique.

C	**Gamma** Υ	**Mean Accuracy**	**False Alarm Rate**
1	0.001	88%	12%
1	0.01	94%	6%
1	0.03	96%	4%
1	0.05	96.3%	3.7%
1	0.07	98.1%	1.9%
10	0.001	97.7%	2.3%
10	0.01	96.3%	3.7%

Table 4. Performance metrics of the model on NSL-KDD dataset.

Datasets	***CC***	**Accuracy**	***FAR***
D1	0.9314	0.9655	0.0012
D2	0.9940	0.9970	0.0030

Table 5. Performance metrics of the proposed model on NSL-KDD dataset.

Datasets	Precision	Recall	F-Score
D1(Before Optimization)	0.9471	0.9809	0.9637
D2 (After optimization)	0.9970	0.9970	0.9970

Table 6. Confusion matrix of NSL-KDD data set with 41 dimensions.

Probe	DoS	U2R	R2L	Normal
1212	164	0	0	0
13	4758	0	1	3
2	1	3	4	6
0	51	0	385	7
188	5277	6	125	596

Table 7. Confusion matrix of NSL-KDD dataset with 31 dimensions.

Probe	DoS	U2R	R2L	Normal
3006	65	0	0	238
94	10611	0	0	927
0	0	37	9	0
1	18	0	1835	488
311	157	0	358	14445

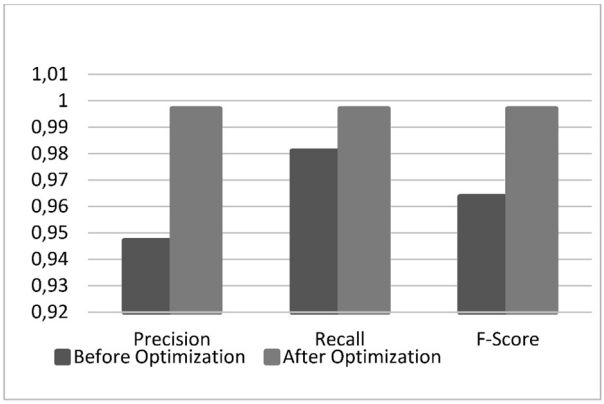

Figure 7. Performance metrics of the proposed model on NSL-KDD dataset.

5.3 Study II: The 31-dimensional Gurekddcup Dataset

The second study is based on the 31 attributes of gurekddcup dataset. The entire dataset is too large to be used for experimentation. Hence most of the experiments are performed on the 10% of the database. Our proposed model also analyzes the 10% of the dataset. The 41 attributes of gurekddcup dataset also had decreased levels of accuracy for individual attacks and hence we deployed the 31 attribute set of gurekddcup for the SVM classifier. Table 8 shows the different metrics obtained for the gurekddcup dataset D1 before optimization and dataset D2 after optimization. Table 11 shows the confusion matrix of gurekddcup after optimization wherein the accuracy of normal instance and most of the attack instances are 99 and above 80 percent respectively. The 27 different attack categories are analyzed in the matrix and it is observed that the minority attacks such as ftp-write, imap, syslog, phf, multihop, loadmodule, eject and perlmagic are 50, 33.33, 50, 40, 66.66, 37.5, 63.63 and 25 respectively in terms of percentage. The results show that the minority attacks are hard to detect in any model.

Table 8. Metrics obtained for the gurekddcup dataset before and after optimization.

Datasets	Precision	Recall	F-Score
D1 (Before optimization)	0.833	0.862	0.847
D2 (After optimization)	0.997	0.999	0.998

5.4 Discussion

Considering the previous results, it is evident that the accuracy, detection rate and false alarm rate for the 31 dimensional dataset preform better whereas the entire dataset results in reduced accuracy. We further examine both the datasets over the reduced dataset with optimization of SVM parameters. Both datasets perform better in terms of classification of majority attacks, however the detection rate of minority attacks is comparatively less, which can be improved by oversampling of the minority samples in the dataset. The critical attacks are the majority at-

tacks in any network environment. Hence this paper aims to identify the majority attacks with high detection rate and low false positive rate.

Table 9 shows the runtime of both the datasets before and after dimensionality reduction. There has been a steady decrease in the runtime after dimensionality reduction and hence minimal resources are required resulting in higher levels of accuracy.

Table 9. Run time of different datasets.

	NSL-KDD dataset (Training and Testing)	**gurekddcup dataset (Training and Testing)**
41 Dimensional dataset	2100 secs	14 729 secs
31 Dimensional dataset	1171 secs	10 235 secs

Figure 8 shows the performance metrics of the proposed model on gurekddcup before and after optimization on the SVM classifier. The x-axis represents the various metrics and the y-axis represents the unit of metrics from 0 to 1. The higher the metric, the better is the performance of the model. Table 10 shows the accuracy and correlation coefficient of the different hybrid intrusion detection models. Higher levels of accuracy and correlation coefficient are achieved in the proposed model which has been specified in Table 10.

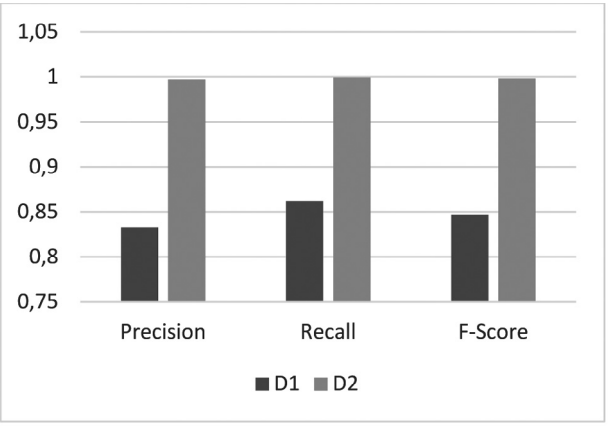

Figure 8. Performance metrics of the proposed model on gurekddcup.

Table 10. Comparison of accuracy and correlation coefficient metric on various hybrid models of intrusion detection.

Models	**Accuracy**	**Correlation Coefficient**
Single-SVM	0.78	0.74
KPCA-GA-SVM	0.93	0.82
PCA-GA-SVM	0.86	0.85
N-KP-CA-GA-SVM	0.92	0.95
Proposed Model	0.99	0.99

6. Conclusion

This paper proposes an intrusion detection model integrating Principal Component Analysis (PCA) and Support Vector Machines (SVM) using RBF kernel. Dimensionality reduction using PCA removes noisy attributes and retains the optimal attribute subset. SVMs construct classification models based on training data obtained from PCA. Optimization of SVM parameters C and Υ for RBF kernel by proposed automatic parameter selection technique reduces the training and testing time and produces better accuracy. Two different datasets NSL-KDD and gurekddcup were applied to the model to analyze the performance. The experimental results indicate that the classification accuracy of the proposed model outperforms other classification techniques using SVM as the classifier with PCA as the dimensionality reduction technique. Minimum resources are consumed as the classifier input requires reduced feature set and thereby minimizes training and testing overhead time.

Table 11. Confusion matrix of gurekddcup dataset.

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	aa
173988	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	876	1	0	0	0	0	0	0	7
0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	1084	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	50	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	30	0	0	0	0	0	0	0	0	0	5	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	4	0	0	0	0	0	0	0	0	4	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	4	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0	8	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0	0	0	6	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0
25	0	0	0	0	0	0	0	6	0	0	0	0	0	0	0	0	1703	4	0	0	0	0	0	0	0	11
4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	25	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	1
4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	7	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	9	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	879	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

References

[1] G. Wang *et al.*, "A new approach to intrusion detection using artificial neural networks and fuzzy clustering", *Expert Syst. Appl.*, vol. 37, pp. 6225–6232, 2010. http://dx.doi.org/10.1016/j.eswa.2010.02.102

[2] W. Wang and R. Battiti, "Identifying intrusions in computer networks with principal Component analysis", in *Proceedings of the First International Conference on Availability Reliability and Security (ARES'06)*, 2006, pp. 270–279. http://dx.doi.org/10.1109/ARES.2006.73

[3] K. Shafi and H. A. Abbass, "An adaptive genetic based signature learning system intrusion detection", *Expert Syst. Appl.*, vol. 36, no. 10, pp. 12036–12043, 2009. http://dx.doi.org/10.1016/j.eswa.2009.03.036

[4] S. Srinoy *et al.*, "Anomaly based intrusion detection using fuzzy rough clustering", in *Paper Presented at the International Conference on Hybrid Information Technology (ICHIT'06)*, 2006, pp. 329-334.

[5] L. Khan *et al.*, "A new intrusion detection system using support vector machines and hierarchial clustering", *Journal on very large databases*, vol. 16, no. 4, pp. 507–521, 2007.

[6] M. V. Mahoney and P. K. Chan, "Learning non stationary models of normal network traffic for detecting network attacks", in *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining KDD'02*, New York, NY, USA, pp. 61–72.

[7] M. Mahoney and P. Chan, "Learning models of network traffic for detecting novel attacks", Florida Institute of Technology, Technical report CS-2001–2, 2002.

[8] L. Weijun and L. Zhenyu, "A method of SVM with normalization in intrusion detection", *Procedia Environmental Sciences*, pp. 256–262, 2011. http://dx.doi.org/10.1016/j.proenv.2011.12.040

[9] A. C. Catania *et al.*, "An Autonomous labeling approach to support vector machines algorithms for network traffic anomaly detection", *Expert Systems with Applications*, pp. 1822–1829, 2012.

[10] Snort [Online]. Available: https://www.snort.org

[11] C. A. Kumar, "Analysis of Unsupervised Dimensionality Reduction Techniques", *Computer Science and Information Systems*, vol. 6, no. 2, pp. 217–227, 2009. http://dx.doi.org/10.2298/CSIS0902217K

[12] S. Thaseen and C. A. Kumar, "An Analysis of supervised tree based classifiers for intrusion detection systems", in 2013 International Conference on Pattern Recognition, Informatics and Mobile Engineering (PRIME), Salem, 2013, pp. 294–299.

[13] F. Kuang *et al.*, "A novel hybrid KPCA and SVM with GA model for intrusion detection", *Applied Soft Computing*, pp. 178–184, 2014. http://dx.doi.org/10.1016/j.asoc.2014.01.028

[14] L. S. Thaseen and C. A. Kumar, "Intrusion Detection Model using fusion of PCA and optimized SVM", in *Proceedings of 2014 International Conference on Computing and Informatics (IC3I)*, Mysore, India, 2014, pp. 879–884.

[15] I. S. Thaseen and C. A. Kumar, "Intrusion Detection Model using fusion of chi-square feature selection and multi class SVM", *Journal of King Saud University – Computer and Information Sciences*, in press, 2016.

[16] I. S. Thaseen and C. A. Kumar, "Intrusion Detection model using chi-square feature selection and modified naïve bayesian classifier", in *Third International Symposium on Big Data and Cloud Computing*, March 10–11 2016.

[17] S. Horng *et al.*, "A novel intrusion detection system based on hierarchical clustering and support vector machines", *Expert Systems with Applications*, vol.38, no.1, pp. 306–313, 2011. http://dx.doi.org/10.1016/j.eswa.2010.06.066

[18] S. Peddabachigiri *et al.*, "Modeling Intrusion detection system using hybrid intelligent systems", *Journal of Network and Computer Applications*, vol. 30, pp. 114–132, 2007. http://dx.doi.org/10.1016/j.jnca.2005.06.003

[19] A. M. Eskin and L. Stolfo, "A Geometric Framework for Unsupervised Anomaly Detection : Detecting Intrusions in Unlabeled Data", in *Proceedings of Applications of Data Mining in Computer Security*, 2002, pp. 174–194.

[20] Y. Li *et al.*, "An efficient intrusion detection system based on support vector machines and gradually feature removal method", *Expert Systems with Applications*, vol. 39, 424–430, 2012. http://dx.doi.org/10.1016/j.eswa.2011.07.032

[21] B. Kasliwal *et al.*, "A hybrid anomaly detection model using G-LDA", in *2014 IEEE International Advance Computing Conference*, 2014, pp. 288–293.

[22] S.-W. Lin *et al.*, "An intelligent algorithm with feature selection and decision rules applied to anomaly intrusion detection", *Applied Soft Computing*, pp. 3285–3290, 2012.

[23] G. Liu *et al.*, "An intrusion detection model based on the PCA and neural networks", *Neurocomputing*, vol. 70, pp. 1561–1568, 2007. http://dx.doi.org/10.1016/j.neucom.2006.10.146

[24] H. S. Kim and S.-D. Cha, "Empirical evaluation of SVM based masquerade detection using UNIX commands", *Computers and Security*, vol. 24, no. 2, pp. 160–168, 2005. http://dx.doi.org/10.1016/j.cose.2004.08.007

[25] G. Mamalakis *et al.*, "Of daemons and men: A file system approach towards intrusion detection", *Applied Soft Computing*, vol. 25, pp. 1–14, 2014. http://dx.doi.org/10.1016/j.asoc.2014.07.026

[26] U. Ravale *et al.*, "Feature Section Based Hybrid Anomaly Intrusion Detection System Using K Means and RBF Kernel Function", *Procedia Computer Science*, vol. 45, pp. 428–435, 2015. http://dx.doi.org/10.1016/j.procs.2015.03.174

[27] J. M. Fossaceca *et al.*, "MARK-ELM: Application of a novel Multiple Kernel Learning framework for improving the robustness of Network Intrusion Detection", *Expert Systems with Applications*, vol. 42, pp. 4062–4080, 2015. http://dx.doi.org/10.1016/j.eswa.2014.12.040

[28] E. De La Hoz *et. al.*, "PCA filtering and probabilistic SOM for network intrusion detection", *NeuroComputing*, vol. 164, pp. 71–81, 2015. http://dx.doi.org/10.1016/j.neucom.2014.09.083

[29] C. Saranya and G. Manikandan, "A study on normalization techniques for Privacy Preserving Data Mining", *International Journal of Engineering and Technology (IJET)*, vol. 5, no. 3, pp. 2701–2704, 2013.

[30] J. E. Jackson, A User's Guide to Principal Components. New York: John Wiley and Sons, 1991. http://dx.doi.org/10.1002/0471725331

[31] L. T. Jolliffe. Principal Component Analysis. Springer-Verlag, 1986. http://dx.doi.org/10.1007/978-1-4757-1904-8

[32] A. Lakhina *et al.*, "Diagnosing Network-Wide Traffic Anomalies", *ACM SIGCOMM Compurer Communication Review*, vol. 34, no. 4, pp. 219-230, October 2004.

[33] T. Ambwani, "Multi Class support vector machine implementation to intrusion detection", in *The international joint conference on neural networks*, vol. 3, July 2003, pp. 2300–2305. http://dx.doi.org/10.1109/ijcnn.2003.1223770

[34] J. Wetson. Support Vector Machine Tutorial [Online]. Available: http://www.cs.columbia.edu/~kathy/cs4701/documents/jason_svm_tutorial.pdf

[35] Information Security Centre of Excellence (ISCX). UNB ISCX NSL-KDD DataSet [Online]. Available: http://www.unb.ca/research/iscx/dataset/iscx-NSL-KDD-dataset.html

[36] Gure KDD cup database [Online]. Available: http://www.sc.ehu.es/acwaldap/

Received: May, 2015
Revised: August, 2015
Accepted: August, 2015

Contact addresses:
Sumaiya Thaseen Ikram
School of Computing Science and Engineering
VIT University
Vandalur
Kelambakkam Road
Chennai – 600 127
India
e-mail: isumaiyathaseen@vit.ac.in

Aswani Kumar Cherukuri
School of Information Technology and Engineering
VIT University
Vellore
India
Near Katpadi Rd Vellore
Tamil Nadu – 632 014
India
e-mail: aswani@vit.ac.in

Sumaiya Thaseen Ikram is an Assistant Professor at School of Computing Science and Engineering, VIT University, Chennai, India. She is currently pursuing PhD in the area of intrusion detection at VIT University. She also holds Bachelor's and Master's degrees in Computer Science from Madras University and VIT University respectively. Sumaiya Thaseen has published 10 research papers so far in various national, international journals and conferences.

Aswani Kumar Cherukuri is Professor of Network and Information Security Division, School of Information Technology and Engineering, VIT University, Vellore, India. Aswani Kumar holds a PhD degree in Computer Science from VIT University, India. He also possesses Bachelor's and Master's degrees in Computer Science from Nagarjuna University, India.
His current research interests are data mining, formal concept analysis, information security, and machine intelligence. Aswani Kumar has published 75 refereed research papers so far in various national, international journals and conferences. He was principal investigator in major research projects sponsored by the Department of Science and Technology, Govt. of India, during 2006–2008 and National Board of Higher Mathematics, Dept of Atomic Energy, Govt. of India during 2011–2013. Presently he is the principal investigator in a major research project funded by Dept. of Science and Technology, Govt. of India under Cognitive Science Research Initiative Program.
Aswani Kumar is a senior member of ACM and is associated with other professional bodies including ISC, CSI, ISTE, IEEE. He is a reviewer in many reputed international journals and conferences. He is an editorial board member of several international journals.