CIT. Journal of Computing and Information Technology, Vol. 27, No. 4, December 2019, 1–16
doi: 10.20532/cit.2019.1004591

A Study of Feature Reduction Techniques and Classification for Network Anomaly Detection

Meenal Jain and Gagandeep Kaur

Department of Computer Science & Information Technology, Jaypee Institute of Information Technology, Noida, India

Due to the launch of new applications the behavior of internet traffic is changing. Hackers are always looking for sophisticated tools to launch attacks and damage the services. Researchers have been working on intrusion detection techniques involving machine learning algorithms for supervised and unsupervised detection of these attacks. However, with newly found attacks these techniques need to be refined. Handling data with large number of attributes adds to the problem. Therefore, dimensionality based feature reduction of the data is required. In this work three reduction techniques, namely, Principal Component Analysis (PCA), Artificial Neural Network (ANN), and Nonlinear Principal Component Analysis (NLPCA) have been studied and analyzed. Secondly, performance of four classifiers, namely, Decision Tree (DT), Support Vector Machine (SVM), K Nearest Neighbor (KNN) and Naïve Bayes (NB) has been studied for the actual and reduced datasets. In addition, novel performance measurement metrics, Classification Difference Measure (CDM), Specificity Difference Measure (S_PDM), Sensitivity Difference Measure (S_NDM), and F1 Difference Measure (F1DM) have been defined and used to compare the outcomes on actual and reduced datasets. Comparisons have been done using new Coburg Intrusion Detection Data Set (CIDDS-2017) dataset as well widely referred NSL-KDD dataset. Successful results were achieved for Decision Tree with 99.0 percent and 99.8 percent accuracy on CIDDS and NSL-KDD datasets respectively.

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

Information systems → Information systems applications → Data mining → Clustering

Networks → Network performance evaluation → Network performance analysis

Keywords: intrusion detection, dimensionality, reduction, principal component analysis, nonlinear principal component analysis, artificial neural network, CIDDS, NSL-KDD

1. Introduction

With the increase in everyday utilization of internet there has been a tremendous surge in network based attacks. According to M.V. Pawar and J. Anuradha [1] network attacks have been classified in two types, namely, active attacks and passive attacks. Distributed Denial of Service attacks (DDoS) are a type of active attacks and occur most frequently in the internet. The prime intent of a DDoS attack is to congest the network and affect the services of the victim server by sending large amounts of IP packets from multiple infected nodes, called bots. Different varieties of DDoS attacks inundate the network, consequently leading to unavailability of regular services to legitimate users, thus incurring financial losses and damaging goodwill of the service providers. Moreover, with advancement in technology and reduced data rates, these attacks have become more sophisticated and can be launched using lesser number of resources. Furthermore, these attacks are known to majorly exploit the vulnerabilities of the network protocols like TCP, UDP, IP, HTTP, DNS, *etc.* To dampen the services, hackers have been known to trace out newer and newer protocol weaknesses. Therefore, it is vital to look out for techniques and design systems that pro-

tect the network by identifying not only the existing ones but also to successfully identify new types of attacks. Intrusion Detection Techniques (IDTs) [2] are used to detect both known and unknown types of threats. IDTs have been divided into two types, namely, (1) Signature based IDTs (SbIDTs) [3] and Anomaly based IDTs (AbIDTs) [4-6]. Pre-identified signatures for normal and attack traffic in SbIDTs are used to detect attack patterns. In AbIDTs, intrusions are identified by making a profile of normal network activity while patterns deviating from normal behaviour are considered as anomalous and later studied for presence or absence of an attack. SbIDTs detect already known attack patterns only and fail to identify unknown or new attacks.

Various techniques like signal processing, statistical analysis, machine learning based approaches, *etc.* have been studied and used by researchers for tackling the menace of network based attacks. In recent times, Machine Learning Techniques (MLTs) have gained popularity [7]. MLTs find widespread use and are popular because of their capabilities to automatically detect attack patterns, identify hidden anomalies, maintain high detection accuracy with low false positive rate, and work on large data sets. Popular MLTs used for classifying network traffic are Support Vector Machine (SVM) [8], Decision Tree (DT), K Nearest Neighbours (KNN), Naïve Bayes (NB) [9] and so on. However, an adverse aspect of employing these classification algorithms for anomaly detection applications is their high complexity with respect to space and time, essentially due to the high dimension space in which these algorithms work. Besides, the number of input parameters required for training of these classifiers has also increased. Moreover, the rate of incoming and outgoing network traffic has also increased exponentially, thus leading to the need for studying large data sets. Therefore, study of feature reduction is required to reduce the size of the data sets in order to ensure fast and accurate application of machine learning algorithms [10]. Furthermore, traditional intrusion detection techniques have been confined to datasets having linear data only. It has been realised that present nature of network traffic data is non-linear and that appropriate techniques of machine learning should be explored.

Varied numbers of datasets related to computer networks traffic are available in the public domain. KDD Dataset which was later converted to NSL-KDD Dataset, after removing its inconsistencies for applying MLTs, has been a dataset widely studied by the researchers' community. However, it's quite old for studying new varieties of attacks that have cropped up on the internet. Henceforth, study of newer datasets is needed. Some of the new benchmark public datasets are CAIDA Dataset, LBNL Dataset, CIDDS Dataset, UNSW, CICIDS2017 Dataset, UNB-ISCX, *etc.* These are available in the network anomaly detection domain [11]. We have worked on CIDDS Dataset due to two main factors: firstly, it covers some of the new attack types and, secondly, due to its size.

In this paper, we have worked on three feature reduction methods, namely, Principal Component Analysis (PCA), Artificial Neural Network (ANN) and Nonlinear Principal Component Analysis (NLPCA). Four classifiers have been applied, namely, Support Vector Machine (SVM), Decision Tree (DT), Naïve Bayes (NB), and K Nearest Neighbours (KNN) to verify the effect of the new reduced sets of features on the detection accuracy and false positive rate. In doing so, the main contributions of our work are:

- reduction of the dimensionality of the network traffic of recent dataset so as to lessen computational time and space complexity;
- generation of new dataset from dimensionally reduced data while maintaining the relevant features required for successful identification of new anomalies;
- applying ML Classifiers to measure performance evaluation metrics;
- maintaining or increasing detection accuracy and reducing false positive rate;
- define novel performance measures, Classification Difference Measure (CDM), Specificity Difference Measure (SPDM), Sensitivity Difference Measure (SNDM), F1 Difference Measure (F1DM) and Combined Performance Measure (CPM) to analyse the outputs.

The rest of the paper covers: literature survey in Section 2, proposed methodology in Section 3, results & discussions in Section 4, conclusion in Section 5 and references at the end.

2. Literature Survey

J. P. Nziga in [12] has presented dimensionality reduction techniques and performed Naïve Bayes and J48 based classifications. The author has used PCA and Multidimensional Scaling for linear and nonlinear dimensionality reduction and reduced the data set to four and twelve dimensions respectively. The dataset used was KDD dataset [13]. Results showed that the Naïve Bayes with twelve dimensions reduced the original dataset to 95.11 percent and J48 with four dimensions reduced dataset to 99.87 percent. K. K. Vasan and B. Surendiran in [14] focused on the efficacy of PCA for anomaly detection and extracted ten Principal Components (PCs) for classification. Two real-time intrusion detection datasets, namely, UNB ISCX and KDD were used. Reduction Ratio (RR) was studied to analyse the importance of PCA in detecting anomalies. It showed that the RR of PCA for KDD and UNB ISCX dataset was 0.24 and 0.36, respectively. Results showed that the classification accuracies using Random Forest (RF) and C4.5 after applying reduced dimensions on both datasets were approximately the same as those obtained using original features, 98.8 percent and 99.7 percent respectively. I. S. Thaseen and C. A. Kumar in [8] have presented two-step PCA feature reduction algorithm. In the first step the variance of every attribute was calculated to find optimal principal components. Ten components with the highest variance were selected and were used in the second step as an input vector for classifier SVM to perform anomaly detection. KDD dataset was used for experiments. It was divided in two separate datasets, namely, D1 and D2. The test results showed that minimum False Positive Rates (FPR) of 0.15 percent and 0.30 percent, respectively, were achieved. F. Rahat and S. N. Ahsan in [9] have proposed a structure using two sampling methods: stratified remove folds and resample. In addition, the authors have proposed five different feature reduction techniques, namely, PCA, Info Gain, Gain Ratio, Chi Square and Filtered Attribute. Five different classifiers were used for classifying performance of the intrusion detection in data set, namely, J48, Naive Bayes, AdaBoost, Bagging, and Nearest Neighbour. It showed that Gain Ratio produced an optimal subset of features. Analysis was performed on KDD dataset. Results showed that KNN and J48 machine learning algorithms performed best, with regard to, processing time, 0.02 sec. and 0.39 sec. respectively. S. Mallissery, S. Kolekar and R. Ganiga in [15] have applied PCA technique for feature reduction. The classifiers used in this paper were Classification and Regression Tree, NB, SVM, ID3, and J48. The analysis was performed on NSL-KDD dataset, with and without dimension reduction technique. The results showed that after reduction the original dataset was reduced to approximately 56.09 percent. They also showed that SVM gave better accuracy of 99.8 percent after reduction. For anomaly detection, a hybrid machine learning algorithm was proposed by A.S.A. Aziz, A.E. Hassanien, S. Hanaf *et al*. in [16]. In the first step, 22 attributes were selected using PCA. For producing detectors, Genetic Algorithm was applied in the next step, which can differentiate between attack and normal behaviour. In the last step, various classifiers were used. The results showed that NB classifier achieved better detection accuracy for two types of attacks, namely, U2R and R2L. Decision Tree classifier achieved highest accuracy of 82 percent and 65 percent for DOS and Probe attacks respectively. PCA is an effective method to reduce dimensionality of data by providing a linear transformation of high dimension to low dimensional feature space as discussed by Cureton and D'Agostino in [17]. Because the time complexity of PCA was high and it also failed in nonlinear mapping, an Improved Principal Component Analysis (IPCA) method was proposed for feature reduction by B. Zhang, Liu, Jia *et al.* in [18]. They differentiated the proposed method with traditional PCA and showed that IPCA, along with Gaussian Naïve Bayes algorithm for classification, achieved better detection rate of 91.06 percent. Also, time was reduced by 60 percent in comparison to Naïve Bayes Classifier. A. Jahanbani and H. Karimi in [19] proposed a new classifying system Principal Component Analysis Neural Network (PCANN) for anomaly detection. KDD dataset was used for analysis and testing. The results showed that the proposed approach had either the same or higher detection and false positive rate of 99.59 percent and 0.40 percent respectively, in comparison with other approaches. Z. Elkhadir, K. Chougdali, and B. Mohammed in [20] applied two feature reduction techniques namely, PCA and Kernel PCA (KPCA) and compared their performances. After extracting the features, KNN

or Decision Tree (DT) algorithms were used for classification. Test result showed that KPCA with proposed kernel, that is power kernel, performed better in comparison with various varieties of kernels. In addition, the detection rate for two types of attacks that is probe attacks and DOS attacks was highest in comparison to PCA method. Y. Wang, H. Yao, and S. Zhao in [21] explained the concept of Auto Associative Neural Network (AANN) and focused on its ability in nonlinear feature reduction. M.A. Kramer in [22] presented a PCA technique for nonlinear feature reduction problem based on neural network model, which referred to the resulting technique as Non-Linear PCA (NLPCA), using Auto Associative Neural Network (AANN) in chemical engineering literature. Kramer's NLPCA has been applied to problems in data reduction and visualization, sensor validation, fault detection, quality control, principal component regression, *etc.* The results showed that NLPCA can be applied in a more general way than PCA. Also, NLPCA improves the performance of these tasks.

From these works we observed that the majority of these algorithms had been tested on old and obsolete datasets. Therefore, study of the new dataset was required. The CIDDS dataset is the most recent publicly available dataset [23] used by the researchers working in the area of anomaly detection. Performance of the said models has been evaluated on CIDDS dataset. Secondly, Artificial Neural Network (ANN) was previously used for classification purposes only. We have applied it for feature reduction.

In the next section, the proposed methodology is discussed.

3. Research Methodology

Main phases of our methodology are preprocessing, followed by feature reduction for identifying key features, and then machine learning based classification for attack detection. Novel performance measures, Classification Difference Measure (CDM), Specificity Difference Measure (S_PDM), Sensitivity Difference Measure (S_NDM), F1 Difference Measure (F1DM) and Combined Performance Measure (CPM) have been computed in this work. The details are discussed in this section.

Main phases of the proposed methodology have been shown in Figure 1 and their detailed description is explained next.

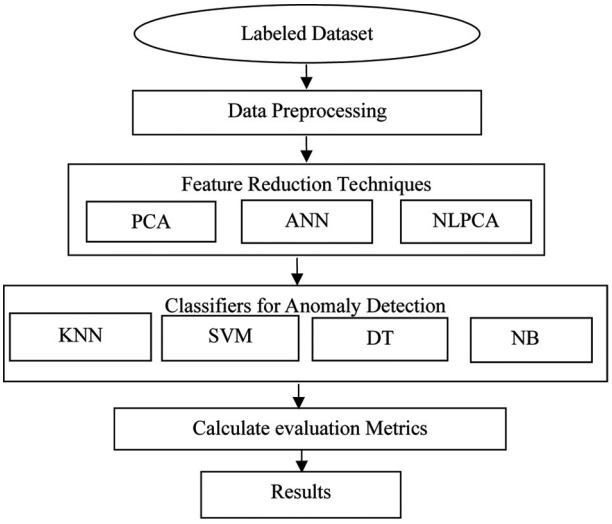

Figure 1. Different Phases of Proposed Methodology.

3.1. Data Preprocessing

3.1.1. Preprocessing of CIDDS Dataset

To transform captured data to desirable format for applying machine learning techniques, data processing was done. Most of the publicly available intrusion detection datasets have unwanted raw attributes which are not required for specific classification techniques. In this work, Coburg Intrusion Detection Data Sets (CIDDS) [24] has been used for anomaly detection. The CIDDS dataset consisted of both numerical and categorical attributes as shown in Table 1.

The following preprocessing steps were done on the dataset.

1. Out of these *AttackID*, *Date_first_seen*, *Duration*, *Attack_Type*, *Attack_Description* and *Flags* were dropped as these attributes did not contribute to classification. Binary data preprocessing was not required because there was no binary data.
2. Since the algorithms based on distance measure work on integer values only, we converted dotted decimal notation of *Src_IP*, *Dest_IP* to long integer (*e.g.* 192.16.68.44 to 192166844). There were addresses in string form to which numerical values were assigned (*e.g.* ext_server to 200000000).

Table 1. Description of Attributes of CIDDS Dataset.

S.no.	Attributes	Category	Description
1	Byt	Numerical	Total number of bytes
2	Pkts	Numerical	Total number of packets
3	Src_IP	Numerical	Internet protocol address of the source
4	Trans_Proto	Categorical	Transport_ Protocol (*e.g.* ICMP, TCP, or UDP)
5	Src_Port	Numerical	Source port number
6	Dest_IP	Numerical	Internet protocol address of the destination
7	Dest_Port	Numerical	Destination port number
8	Class	Categorical	Type of class (normal, attack)
9	Duration	Numerical	Duration of the flow.
10	AttackID	Numerical	Unique attack id.
11	TOS	Numerical	Type of services
12	Flows	Numerical	Sequence of packets from source to destination.
13	Date_first_seen	Numerical	Start time flow first seen.
14	Attack_Type	Categorical	Type of attacks
15	Attack_Description	Categorical	Additional information of attacks.
16	Flags	Categorical	OR concatenation of all TCP flags.

3. Values in the '*Byt*' field that were non-numeric like '2.1M' were converted to integer form like (2.1M to 2100000).
4. In order to convert the categorical field '*Trans_Proto*' to numeric field, following convention has been applied: TCP, ICMP, UDP, GRE were mapped to 1, 2, 3, 4 respectively.
5. '*Flows*' and '*TOS*' were also removed because both of them had a single constant value of 1.0.
6. The CIDDS dataset consisted of parameters like '*Byt*' where there were small numbers of instances with high byte count and large number of instances with very small byte count, like 21000000,76. Therefore, normalization was applied to scale down the value into the range of zero to one based on the equation given below:

$$X_n = \frac{x - mean(x)}{std(x)} \quad (1)$$

where x is the attribute value, *std* is the standard deviation and X_n is the calculated normalized value.

7. The statistical procedure called Pearson Correlation Coefficient has been used to analyze the linearity and nonlinearity of the dataset. It quantified *Byt* and *Pkts* attributes as linear in nature and the remaining attributes as nonlinear.

3.1.2. Preprocessing of NSL-KDD Dataset

The KDDCUP dataset is the most preferred publicly available dataset used by the researchers working in the field of network intrusion detection. However, Ghorbani and his team [25] did statistical analysis of this dataset and reported some inconsistencies. They found out that these irregularities could be affecting the performance of IDSs, especially the ones presumed on anomaly based network intrusion detection. Their team removed irrelevant records from the original files and proposed a new dataset, named, NSL-KDD. The dataset has 41 features with label class as 42nd feature and has been divided into nominal, binary and numeric values. The NSL-KDD dataset files have been divided into training dataset and testing dataset. Instances in these files are 'labeled' as 'normal' for regular traffic and 'attack' for attack traffic.

1. That dataset has six binary parameters, namely, *land*, *logged_in*, *root_shell*, *su_attempted*, *is_host_login*, and *is_guest_login* but *su_attempted* has 3 values (0, 1, 2). To convert *su_attempted* to binary values, the value 2.0 was replaced with 0.0 because there was no instance of value 2.0 in the training data and only 59 instances in the testing dataset. Therefore, it was appropriate to replace 2.0 with 0.0 for *su_attempted* parameter.
2. It was realized that the parameter '*num_outbound_cmds*' has only 0.0 values and therefore it was decided to remove the instances of this parameter.
3. Since most of machine learning algorithms use numerical data for their algorithms, label *encoding* was applied to convert categorical data to integer values. Three parameters, namely, *Protocol_type*, *Service*, and *Flag* were converted to numerical values using label *encoding*.
4. Training data was further divided into 80 : 20 ratios where 20 percent was used for *cross validation.*

3.2. Feature Reduction Techniques

In machine learning the complexity of the algorithms is dependent on two characteristics of the dataset: number of input variables (*i.e.* dimensions 'd') or size of the dataset (*i.e.* number of instances 'n'). Therefore, dimensionality reduction of any of the above two characteristics helps in reducing space complexity. This improves the performance of machine learning algorithms. Since CIDDS dataset has very large number of instances, dimensionality reduction is crucial before applying the algorithms. Two common approaches for handling large number of instances used in machine learning are: feature selection and feature reduction. Though feature selection leads to reduction in dimensionality by choosing a small set of attributes, this procedure is not effective in cases when all attributes are important for anomaly detection. Therefore, feature reduction was applied to CIDDS dataset to transform original attributes so as to generate other significant features.

Three feature reduction techniques, namely Principal Component Analysis (PCA), Artificial Neural Network (ANN), Nonlinear Principal Component Analysis (NLPCA), were applied on the dataset. After pre-processing the CIDDS dataset consisted of seven attributes, namely *Byt*, *Pkts*, *Trans_Proto* (*e.g. ICMP*, *TCP*, or *UDP*), *Src_IP*, *Src_Port*, *Dest_IP*, and, *Dest_Port*.

Details of the applied feature reduction technique are explained next.

3.2.1. Principal Component Analysis

PCA works on the basis of variances. Individual variances for various attributes in the dataset were computed and dimensionality reduction was done based on variance score. First five principal components, as given in Table 2, were selected with the highest variance of 91.56 percent and further used in the second step as an input vector for classifiers to perform anomaly detection. We divided the dataset into 80 : 20 ratios were 80 percent of the instances were used for training. The training data subset was used for finding out Principal Components (PCs).

Table 2. Principal Components (PCs) and Corresponding Eigen-Values Elected based on Outcomes of Scree Plot Test and Critical Eigenvalue Test.

Feature Names	Eigen-values
PC1	2.2398
PC2	1.7384
PC3	1.0066
PC4	0.7568
PC5	0.6680
PC6	0.3288
PC7	0.2616

The steps involved in calculating PCs are given below.

1. Covariance of seven attributes was calculated based on the equation given below:

$$\begin{aligned} &\operatorname{cov}(f_i f_j) = \\ &= E\left[E\left[f_i\right] - f_i\right] \cdot E\left[E\left[f_i\right] - f_i\right], \end{aligned} \tag{2}$$

where $E[f_i]$ and $E[f_j]$ denote the expected value of the attributes f_i, f_j respectively, and $1 \le i, j \le 7$.

2. Using the covariance matrix, the eigenvectors and eigenvalues were calculated.
3. The obtained eigenvalues were sorted in decreasing order as given in Table 2. These eigenvalues were used as PCs whereby the eigenvector with the highest eigenvalue ev_1 became first principal component PC_1, second highest eigenvector with the highest eigenvalue ev_2 became second principal component PC_2 and so on.
4. In order to decide the sufficient number (n) of features, we performed *Scree Plot Test* and *Critical Eigenvalue Test* (Cureton and D'Agostino, 1983). The remaining features were discarded as redundant data.
   a) In *Scree Plot Test* the differences $d_i f_i$ between respective PCs are computed using the sorted eigenvalues.

$$d_i f_j = ev_i - ev_{i+1} \tag{3}$$

   A graph of principal components vs eigenvalue differences was plotted as shown in Figure 2. From the plot peaks were observed at points $d_i f_2$ (0.7318) and $d_i\ f_5$ (0.3395). Therefore, the break in the trend happened between points $d_i f_2$ and $d_i f_3$ and between $d_i f_5$ and $d_i f_6$. Since two peak values were received, we performed another test *i.e.* critical eigenvalue test.

   b) The *Critical Eigenvalue Test* is used to compute the threshold of eigenvalues to detect the number of final principal components. Several experiments were conducted to determine the best threshold.

   For our tests we found $\tau_c = f^{\frac{0.9}{10}}$ was appropriate, where f is a feature of the dataset. For our test τ_c was 0.5762. Based on these two tests, the number of significant features was decided to be five. These five features were *Src_IP*, *Trans_Proto*, *Src_Port*, *Dest_IP*, and *Dest_Port*.
5. The obtained five feature vectors in step 4 were used to compute new features. The formula to calculate new features is given below:

$$nf_x = [f_1, f_2, f_3, f_4, f_5, f_6, f_7] \cdot [ev_{i,j}], \quad \text{for } x = 1 \text{ to } 5,\ i, j = 1 \text{ to } 7 \tag{4}$$

   PCA_DATA = $\{nf_1, nf_2, nf_3, nf_4, nf_5, nf_6, nf_7\}$, where PCA_DATA is new dataset and nf_x is new feature.

This new dataset PCA_DATA was further used as an input to the classifiers in the next phase.

PCA, being a linear dimension reduction technique, has its obvious limitation. Hence, we have used Non Linear Principal Component Analysis (NLPCA) using the auto-associative neural network model. The implementation of NLPCA is presented in the next section 3.2.3. Further, we have used the multi-layer perceptron neural network called Artificial Neural Network (ANN) model. Based on the accuracy, best attributes were selected. To get rid of correlations among these attributes, we again used the Auto Associative Neural Network model (AANN). The uncorrelated features, thus obtained, were used as input to classifiers. Implementation details are discussed in section 3.2.2.

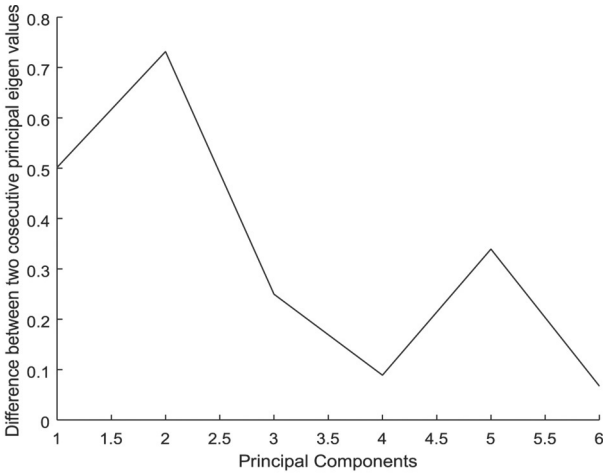

Figure 2. Scree Plot Test.

3.2.2. Artificial Neural Network

Artificial Neural Network (ANN) is used to solve classification problems, noise reduction, prediction, *etc.* ANN helps in prediction of future score based on past knowledge and specified learning on a training dataset. In this work ANN has been used for feature reduction.

In three-layer perceptron ANN with nonlinear transfer function, first layer consists of input attributes also called neuron, and the second layer is called hidden layer in which each neurons receive inputs from the first layer neuron output. Sigmoid nonlinear activation function has been used in the hidden layer for extracting significant features; the third layer is the output layer in which identity activation function has been used. Its basic function is defined as:

$$f(I) = \tau \sum_j W_j \cdot I_j, \tag{5}$$

where $f(I)$ is the predicted output of the class label, τ is the sigmoid activation function and W_j is a weight of each instance I_j.

The steps listed below were followed for feature reduction.

1. To determine the number of neurons required if hidden layer accuracy rate was computed. The values were computed by taking one to five neurons at a time. So the best accuracy of 97 percent was obtained for five neurons and therefore five neurons were fixed.
2. Since training of the network is largely dependent on the number of epochs required, Early Stopping Criteria (ESC) was used to determine the number of epochs. ESC is based on how accurately the training data is predicted and on the number of epochs used for achieving that accuracy. In this work different accuracy values were calculated by increasing the five epochs at every step. Best accuracy (approx. 97 percent) was obtained for 40 epochs and therefore the number of epochs was fixed at 40.
3. Iterative pruning of the input attributes was done by removing them one by one. Firstly, all seven attributes were taken, namely *Byt*, *Pkts*, *Src_IP*, *Trans_Proto*, *Src_Port*, *Dest_IP*, and *Dest_Port*, and the accuracy of the validation dataset was computed. Secondly, six attributes were taken by removing the first attribute which was *Byt*. Thirdly, five attributes were taken by removing the first two attributes which were *Byt* and *Pkts* and the process was continued till the best accuracy was achieved. And based on best results, the last four attributes were fixed, namely, *Trans_Proto*, *Src_Port*, *Dest_IP*, and *Dest_Port*, as shown in Figure 3.
4. In the output obtained from the model in step 3 based on accuracy memorization, four best attributes were selected. Further, to get rid of the correlations among these four attributes, we again used the ANN model on these attributes. The uncorrelated features, thus obtained in step 4, were used as a new reduced feature to form the new dataset (ANN_DATA). This new dataset became the input for the classification phase.

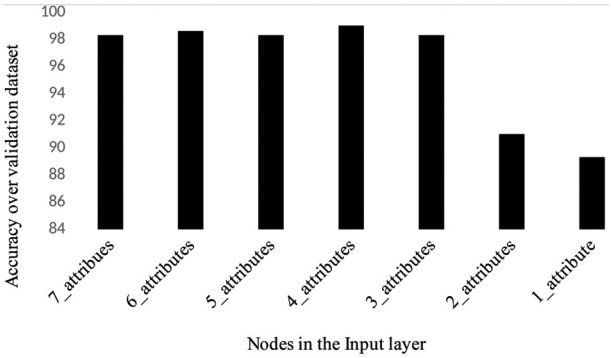

Figure 3. Variation in prediction accuracies of ANN works over validation data.

3.2.3. Nonlinear Principal Component Analysis

Nonlinear Principal Component Analysis (NLPCA) was introduced as a nonlinear feature reduction technique by (Kramer, 1991). Auto-Associative Neural Network (AANN) is used to generate NLPCA. It is a three-hidden-layer feed-forward neural network where the target data set is identical to input data set and the input and output layers are connected via weights. One of the hidden layers of the network works as a bottleneck layer of the network, which forces the reduction of data dimensionality for data interpretation and for anomaly detection.

Steps to perform nonlinear feature reduction using AANN are given below.

1. 80 percent of training dataset was used to perform the training of nonlinear components.

2. Custom auto-associative neural network is created to generate nonlinear features. The network consisted of seven neurons in the input layer, and three hidden layers, namely, hidden layer 1 (HL_1), bottleneck layer (BL), and hidden layer 3 (HL_3), respectively. In addition, there were seven neurons in the output layer.
3. Five neurons were considered in HL_1 and HL_3, as explained in ANN technique.
4. The number of neurons in bottleneck layer varied from the number of attributes from one to seven.
5. Activation functions *tan sigmoid* was used in hidden layers where as *pure linear* was used in output layer. *Trainlm* function was used for training the network.
6. The output of HL_1 was passed as input to *BL*. *BL* is the vital layer used for feature reduction by eliminating nodes. The output of the bottleneck was passed to HL_3.
7. Compute output based on the iterative pruning of the input attributes in the bottleneck layer, by removing attributes one by one and computing the accuracy of the training dataset. Based on the best results, last five attributes were fixed, namely, *Src_IP*, *Trans_Proto*, *Src_Port*, *Dest_IP*, and *Dest_Port*, as shown in Figure 4.
8. The output obtained from the model using five attributes selected in step 7 was fixed as a new reduced feature to form the new dataset (NLPCA_DATA). This new dataset became input for the classifiers.

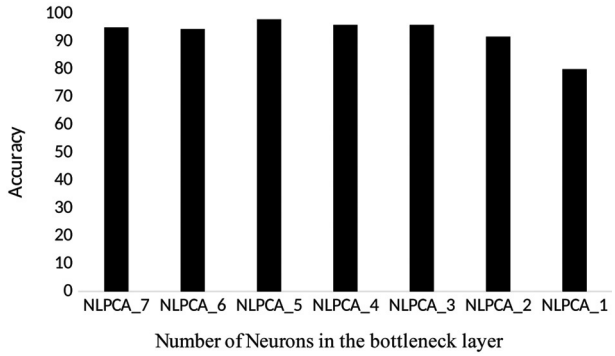

Figure 4. Predicted accuracies over test data by pruning the BL in NLPCA.

3.3. Classifiers for Anomaly Detection

Four supervised machine learning classifiers, namely Decision Tree (DT), K Nearest Neighbor (KNN), Support Vector Machine (SVM), and Naïve Bayes (NB) were applied on the datasets generated in phase 3.2. That is the data from three different independent dimension reducers, and an actual data is fed to the four classifiers simultaneously. And the information gain was evaluated to decide on the best technique. Respective classifiers are explained next.

3.3.1. Decision Tree Based Anomaly Detection

Decision Tree based classification is based on the construction of DT by deciding how the nodes are split. The vital part in DT construction is splitting node value. To decide on the splitting value, the steps followed in our algorithm are explained next.

1. Check the class of all instances in the dataset. If they belong to single class, then create a single node and stop.
2. For each feature (f) gain ratio was computed as ratio of feature information gain and feature split value using the formula given below:

$$Gain\ Ratio = \frac{information\ gain(f_i)}{s(f_i)} \quad (6)$$

where, $i \leq n$, n is the number of features in the dataset.

3. To compute feature information gain, individual entropy values were computed for attack and for normal classes. To compute entropy individual probabilities were calculated for all features for two classes namely normal and attack, the formula use in given equation is:

$$Entropy(f_i) = \\ = -\sum_{k=1}^{\alpha} \frac{frequency(c_k f_i)}{|f_i|} \cdot \frac{\log_2(c_k f_i)}{|f_i|}, \quad (7)$$

where $C = C_1$, C_2 is the set of classes and α = number of classes.

4. Similarly, feature information gain is calculated as shown below:

$$information\ gain(f_i) = \\ = Entropy(f_i) - \sum \frac{|f_i|}{|F|}, \tag{8}$$

where $F = f_1, f_2, \ldots, f_n$, where n is the number of features.

5. The node splitting of the tree was done based on the highest gain ratio for the particular feature.
6. Repeat steps 1 to 4 till no splitting is possible.

3.3.2. K-Nearest Neighbor Based Anomaly Detection

K-NN is one of the simplest supervised machine learning algorithm used for classification. It classifies a data point based on how its neighbours behave. K-NN stores all available cases and classifies new cases based on a similarity measure. The procedure of deriving best classification model involves the three following steps.

1. Pick the right value of K, where K is the number of nearest neighbors, in our experiment we choose $K = 1$.
2. Calculate the similarity measure (Euclidean distance) between all the input instances.
3. Sort the distances and determine the nearest neighbor based on the K^{th} minimum distance.

Euclidean distances were computed using the equation:

$$d(x, y) = \sqrt{\sum_{i=1}^{k} (x_i - y_i)^2}, \tag{9}$$

where x_i and y_i are the instances in a given set of attributes.

Similarly, ED was calculated for new data point x, y for all the instances in a given dataset. The new calculated value was compared with the ED of the old instances. The class of the instance for which the new calculated value was closest was considered as the resulting class of the new data point.

3.3.3. Support Vector Machine Based Anomaly Detection

Support Vector Machine is a supervised machine learning technique based on classification or regression in network anomaly detection. In network anomaly detection it is primarily used for classification. Using SVM, data instances are plotted as points in n-dimensional space, where n is the number of features. The coordinates represent the value of each feature individually. These coordinates are used for classification by finding hyperplane of attack and normal classes. In this work the two classes are normal class and attack class. The target was to use SVM so that separating margin of these classes could be maximized as well as training error could be minimized. SVM ability to generalize the result depended on the margins. These coordinate points in the hyperplane were used to find the support vectors which were further used to find the hyperplane. To do so, α coefficients for the kernel function were computed using the equation:

$$\sum_{i=1}^{n} \alpha_i \left(y^i\right) k\left(\left(x^i, x\right)\right) + b, \tag{10}$$

where, $1 \le i \le n$, x^i y^i is i^{th} coordinate, x^i is an input vector of any dimension, y^i is a class label (1 or 0), α_i is the associated coefficient, k is a kernel function that operates on two vectors and gives scalar output, b is a scalar value.

3.3.4. Naïve Bayes Based Anomaly Detection

For classification problems, Naïve Bayes (NB) is one of the most popular machine learning algorithms. It studies the interconnection between dependent and independent features to obtain a contingent probability for every connection. Therefore, a strong assumption has been established that the features are independent. Mathematical representation of NB is shown below:

$$P(c_i \mid F) = \\ = \frac{P(f_1 \mid c_i) P(f_2 \mid c_i) \ldots P(f_7 \mid c_i)}{P(F)} \tag{11}$$

where, c_i represents the type of classes ($c_1 = Normal$ and $c_2 = Attack$) and $F = f_1, f_2, \ldots, f_n$, where $n = 7$ is the number of features.

3.4. Performance Evaluation Metrics

Performance was measured in terms of performance metrics, namely Accuracy (ACC), and False Positive Rate (FPR). In addition to traditional performance metrics, novel performance measures such as Classification Difference Measure (CDM), Specificity Difference Measure (S_PDM), Sensitivity Difference Measure (S_NDM), and F1 Difference Measure (F1DM) have been defined and results were computed. Consider True Positive values as (TP), False Positive values as (FP), True Negative values as (TN), False Negative values as (FN), then TP, FP, TN and FN can be defined as:

- TP: the total count of "normal" instances in the dataset correctly classified as "normal" instances;
- FP: the total count of "normal" instances in the dataset wrongly classified as "attack" instances;
- TN: the total count of "attack" instances in the dataset correctly classified as "attack" instances;
- FN: the count of "attack" instances in the dataset wrongly classified as "normal" instances.

Accuracy (ACC) and False Positive Rate (FPR) scores were calculated from these metrics based on the following equations:

$$ACC = \frac{TP + TN}{TP + TN + FP + FN}$$
$$FPR = \frac{FP}{FP + TN} \quad (12)$$

To evaluate the performance of such reduced datasets with actual data, the main target was to reduce the dimensionality of the feature set (F) from 'd' to 'k' such that $F^k < F^d$. The difference in the detection accuracy (ACC) for the dataset with the dimension D and the dataset with the dimension K was computed as the CDM.

$$CDM = ACC_K - ACC_D \quad (13)$$

Similarly, S_PDM was computed as a difference in the false positive rate of D dimensional dataset and K dimensional dataset.

$$S_PDM = FPR_K - FPR_D \quad (14)$$

For CDM > 0, the information gain was achieved for the reduced dataset. For CDM < 0 loss of the information occurred in the reduced dataset. Also, S_PDM > 0 resulted in gain whereas S_PDM < 0 resulted in loss for the reduced dataset. If the values for CDM and S_PDM were zero, then the information retention was achieved.

Sensitivity has been defined as a measure of the ratio of negative cases that got predicted as true negative cases. To study the impact of sensitivity on actual and reduced datasets, a new metric *i.e.* difference of sensitivity was computed. Sensitivity Difference Measure (S_NDM) was computed as given in the equation below:

$$S_N\text{DM} = \left(\frac{TN}{FP + TN}\right)_K - \left(\frac{TN}{FP + TN}\right)_D \quad (15)$$

F1 measure is known to reflect the balance between precision and recall. For high detection performance low values of FP and FN are considered good, thus resulting in low F1. Therefore, F1 measure can be used to performance of detection methods and difference of F1 measures was used to study the impact on original and reduced datasets. A new metric, difference of F1 scores, was computed as F1 Difference Measure (F1DM) as given in the equation below:

$$F1DM = F1DM_K - F1DM_D, \quad (16)$$

where,

$$F1\,Score = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall},$$
$$Precision = \frac{TP}{TP + FP},$$
$$Recall = \frac{TP}{TP + FN}.$$

The results are discussed in the next section.

4. Results

Performance of an intrusion or anomaly detection technique is measured based on its ability to classify normal and attack instances correctly. We have applied four classifiers, namely

KNN, SVM, DT, and NB on four datasets (one actual, three derived) ACTUAL_DATA, PCA_DATA, ANN_DATA, and NLPCA_DATA. Performance of the aforesaid classifiers has been compared using popular metrics like detection accuracy (*ACC*) and false positive rate (*FPR*). Additionally, new metrics have been defined to study the impact of dimensionality based feature reduction in a dataset. Novel performance measures, namely Classification Difference Measure (CDM), Specificity Difference Measure (S_PDM), Sensitivity Difference Measure (S_NDM), and F1 Difference Measure (F1DM) have been computed and the results have been analyzed.

In this section the results are discussed in the following manner. Firstly, three feature reduction algorithms were applied to reduce the original dataset. Using these algorithms three new datasets were created for performance evaluation. In the second step, four classifiers, namely KNN, SVM, DT and NB were applied. These machine learning algorithms were tested on two datasets, namely, CIDDS and NSL-KDD.

4.1. Feature Reduction

Table 3. Data derived after feature reduction techniques.

Dataset	Features	Method
ACTUAL_DATA	7	None
PCA_DATA	5	PCA
ANN_DATA	4	ANN
NLPCA_DATA	5	AANN

Table 3 shows the number of features reduced using PCA, ANN and AANN algorithms. Our focus in this paper was to study the performance of machine learning algorithms on non-linear network data. Therefore, we focused on these three algorithms for feature reduction. As given in the table, the original CIDDS Dataset after pre-processing had seven features and with PCA it was reduced to five features and to four and five features with ANN and AANN respectively. Although the lowest number of four features was achieved with ANN, *i.e. Trans_Proto*, *Src_Port*, *Dest_IP*, and *Dest_Port*, one of the most important attributes, *i.e. Src_IP* got removed. Without knowing the source IP address, the source of the attack/attacks cannot be identified. Therefore, we measured the performance for five features with ANN as well.

Table 4. Detection accuracies for four classifiers on four datasets.

Classifier	ACTUAL_ DATA (7)	PCA_ DATA (5)	ANN_ DATA (5)	NLPCA_ DATA (5)
KNN	0.99	0.98	0.9631	0.99
SVM	0.88	0.88	0.9836	0.88
DT	0.80	0.79	0.9674	0.99
NB	0.80	0.93	0.9731	0.84

Table 4 shows the detection accuracies for KNN, SVM, DT, and NB. For ACTUAL-DATA achieved accuracies of KNN, SVM, DT, and NB were 0.99, 0.88, 0.80, and 0.80 respectively. The values obtained on PCA-DATA were 0.98, 0.88, 0.79, and 0.93. Similarly, accuracies achieved for ANN-DATA were 0.95, 0.97, 0.96, and 0.97 respectively. Lastly, for NLPCA-DATA classification, accuracies of KNN, SVM, DT, and NB were 0.99, 0.88, 0.99, and 0.84 respectively. In this work we have considered KNN as a reference classifier. It is known to have least or no training time with best results [26] and is therefore used as a reference classifier for comparing the performance of other algorithms. Therefore, from the values achieved, it was observed that SVM had best accuracy of 98 percent on ANN_DATA with five features; DT had best accuracy of 99 percent on NLPCA_DATA with five features whereas NB's best accuracy of 97 percent was on ANN_DATA. Therefore, out of the three algorithms, SVM and DT were shortlisted for further study.

Table 5. False positive rates for four classifiers on four datasets.

Classifier	ACTUAL_ DATA (7)	PCA_ DATA (5)	ANN_ DATA (5)	NLPCA_ DATA (5)
KNN	0.0080	0.0032	0.0032	0.0063
SVM	0.0833	0.1001	0.0021	0.1135
DT	0.1991	0.1995	0.0036	0.0020
NB	0.1991	0.0181	0.0024	0.0198

Table 5 shows FPR values for the classifiers. SVM had lowest FPR for ANN_DATA and DT had lowest FPR for NLPCA_DATA.

Upon comparing the results for all the classifiers for four datasets, based on SVM and DT values, ACTUAL_DATA and PCA_DATA datasets, further analysis was dropped. We now had SVM with best performance (98,0.0021) on ANN_DATA and DT with best performance (99,0.0020) on NLPCA_DATA. Further comparison was done for SVM and DT.

Table 6. Training times of classifiers.

Classifier	ANN_DATA (5 dim.)	NLPCA_DATA (5 dim.)
KNN	2.290 sec.	14.23 sec.
SVM	408.763 sec.	510.675 sec.
DT	9.266 sec.	4.074 sec.
NB	1.057 sec.	1.781 sec.

Table 6 shows the training times of SVM and DT. It was found that training time for SVM on ANN-DATA was 408.76 secs and on NLPCA_DATA it was 510.675 secs which was relatively high in comparison to training time of 9.266 secs, on ANN_DATA and 14.23 secs on NLPCA_DATA of DT. Therefore, DT was considered best under the given conditions. Figure 5 shows the ROC curve for the same.

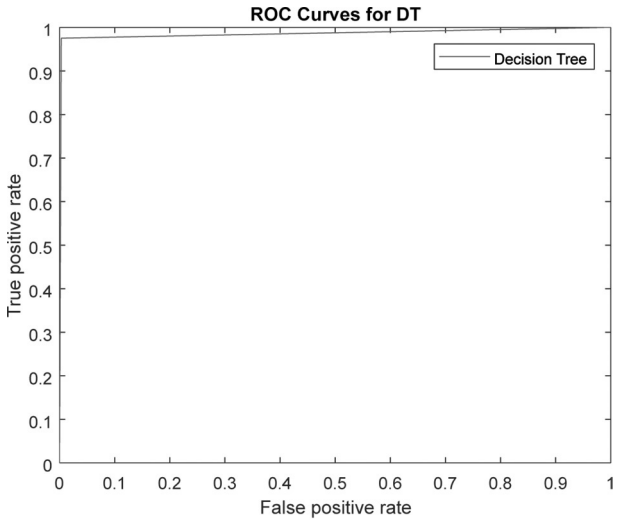

Figure 5. ROC curve for DT on NLPCA_DATA.

In addition to detection accuracy and FPR, novel performance metrics were also evaluated. Table 7 shows the values measured for CDM, S_PDM, S_NDM and F1DM. A CDM of 0.19 was achieved for DT. It meant that classification accuracy of DT on NLPCA_DATA reduced dataset with five features improved in comparison to classification accuracy on the original dataset with seven features. Similarly, specificity difference measure SPDM was measured and it was − 0.1971. This meant that false positive rate of DT on NLPCA_DATA reduced in comparison to ACTUAL_DATA. Furthermore, the scores for SNDM and F1DM were computed as 0.2781 and 0.2907, respectively.

Table 7. CDM, SPDM, SNDM and F1DM Scores for DT.

Proposed Metrics	Values
CDM	0.1900
SPDM	− 0.1971
SNDM	0.2781
F1DM	0.2907

Since SNDM was computed as number of attacks rightly predicted out of total number of attacks, the achieved value of 0.2781 meant that the results improved for NLPCA_DATA as compared to ACTUAL_DATA, *i.e.* more attacks were correctly predicted as attacks. F1DM is a measure used to analyze the combined impact of precision and recall measures. Positive value of 0.2907 meant that results received for DT classifier on NLPCA_DATA were more accurate as compared to ACTUAL_DATA. Figure 6 shows the graph for achieved values.

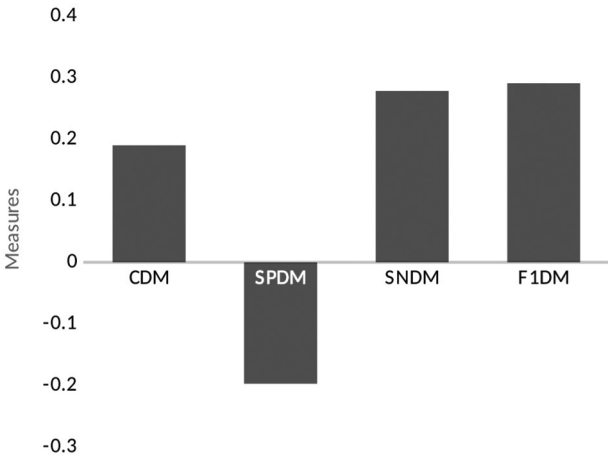

Figure 6. CDM, SPDM, SNDM and F1DM scores for DT

4.2. Performance Evaluation of DT on NSL-KDD Dataset

Performance of DT for NLPCA based feature reduction was also measured on NSL-KDD dataset. As mentioned in the earlier sections, KDD dataset has been extensively used for studying machine learning techniques applied for intrusion detection. However, NSL-KDD dataset was created from KDD after removing the said inconsistencies. There is not much literature available for comparison of non-linear feature reduction techniques and ML classifiers on NSL-KDD.

We therefore have tabulated our comparison results in Table 8 for KDD. As Table 8 shows, our proposed approach of feature reduction and DT based classification achieved high accuracy of 99.8 percent with reduced dataset with ten features, namely, *count*, *serror rate*, *srv error rate*, *dst host serror rate*, *dst host srv serror rate*, *service*, *rerror rate*, *srv rerror rate*, *diff srv rate*, and *dst host count*.

5. Conclusion

In this work three feature reduction techniques, namely PCA, ANN, and NLPCA were applied on the CIDDS dataset to reduce the attributes. Seven attributes in the actual dataset were respectively reduced to five features, with respect to PCA, ANN, and NLPCA. Machine learning classifiers, namely KNN, SVM, DT, and NB were applied on actual and reduced datasets for normal and attack classification. Based on detection accuracy and FPR, DT had best accuracy of 99 percent on NLPCA_DATA with 0.0020 FPR. In addition to detection accuracy and FPR, novel performance metrics namely, CDM, S_PDM, S_NDM and F1DM were also evaluated to study the impact of dimensionality based feature reduction in a dataset. A CDM of 0.19 was obtained for DT. Similarly, S_PDM was measured and it was -0.1971. Furthermore, the scores for SNDM and F1DM were computed as 0.2781 and 0.2907, respectively. Performance of DT for NLPCA based feature reduction was also compared on NSL-KDD dataset. The results showed that DT retained high accuracy of 99.8 percent and low FPR 0.0021 on NLPCA_DATA with a set of ten reduced features.

Our future work can further explore classification techniques for reducing dataset size and decrease training time without sacrificing accuracy.

References

[1] M. V. Pawar and J. Anuradha, "Network Security and Types of Attacks in Network", *Procedia Computer Science,* vol. 48, pp. 503–506, 2015. https://doi.org/10.1016/j.procs.2015.04.126

[2] V. V. Phoha, "The Springer Dictionary of Internet Security", New York: Springer Verlag, 2002.

[3] S. Kumar and E. H. Spaord, "A Pattern Matching Model for Misuse Intrusion Detection", in *Proc. of the 17th National Computer Security Conference*, 1994, pp.11–21.

[4] H. S. Javitz and A. Valdes, "The SRI IDES Statistical Anomaly Detector", in *Proc. of the IEEE Symposium on Security and Privacy*, 1991. https://doi.org/10.1109/RISP.1991.130799

[5] G. Kaur *et al.*, "Study of Self-Similarity for Detection of Rate-Based Network Anomalies", *International Journal of Security and Its Applications*, vol. 11, no. 8, pp. 27–44, 2017. http://dx.doi.org/10.14257/ijsia.2017.11.8.03

Table 8. Feature reduction technique on KDD dataset.

Paper Ref. No.	Dataset	Feature Reduction Algo.	Features Reduced	Classifier	Validation Acc.
[27]	Subset of KDD (31279 instances)	PCA	10	RF	0.997
[28]	NSL-KDD	PCA	31	SVM	0.997
[19]	10 percent KDD	PCA-NN	30	ANN	0.995
[29]	KDD	NLCA	12	DT	0.996
Proposed Approach	NSL-KDD	NLPCA	10	DT	0.998

[6] G. Kaur *et al.*, "Detection of TCP Targeted High Bandwidth Attacks Using Self-Similarity", *Journal of King Saud University – Computer and Information Sciences*, vol. 32, pp. 35–49, 2017. https://doi.org/10.1016/j.jksuci.2017.05.004

[7] T. Abraham, "IDDM: Intrusion Detection Using Data Mining Techniques", Tech. Rep. DSTO-GD-0286, Defense Science and Technology Organization, Australia, 2000.

[8] I. S. Thaseen and C.A. Kumar, "Intrusion Detection Model Using Fusion of PCA and Optimized SVM", in *Proc. of the IEEE International Conference on Contemporary Computing and Informatics (IC3I)*, 2014. https://doi.org/10.1109/IC3I.2014.7019692

[9] F. Rahat and S.N. Ahsan, "Comparative Study of Machine Learning Techniques for Preprocessing of Network Intrusion Data", in *Proc. of the IEEE International Conference on Open Source Systems & Technologies (ICOSST)*, 2015. https://doi.org/10.1109/ICOSST.2015.7396401

[10] S. Khalid *et al.*, "A Survey of Feature Selection and Feature Extraction Techniques in Machine Learning", in *Proc. of the Science and Information Conference (SAI)*, 2014, pp. 372–378. https://doi.org/10.1109/SAI.2014.6918213

[11] M. Ring *et al.,* "A Survey of Network-Based Intrusion Detection Data Sets", *Computers & Security*, 2019.

[12] J. P. Nziga, "Minimal Dataset for Network Intrusion Detection Systems via Dimensionality Reduction", in *Proc. of the IEEE 6th International Conference on Digital Information Management (ICDIM)*, 2011. https://doi.org/10.1109/ICDIM.2011.6093368

[13] KDD dataset. [Online]. Available: http://kdd.ics.uci.edu/databases/kddcup99 [Accessed: 2 August 2016]

[14] K. K. Vasan, and B. Surendiran, "Dimensionality Reduction Using Principal Component Analysis for Network Intrusion Detection", *Perspectives in Science*, vol. 8, 2016. https://doi.org/10.1016/j.pisc.2016.05.010

[15] S. Mallissery *et al.*, "Accuracy Analysis of Machine Learning Algorithms for Intrusion Detection System Using NSL-KDD Dataset", in *Proc. of the International Conference on Future Trends in Computing and Communication (FTCC)*, 2013. https://doi.org/10.13140/RG.2.1.5018.0247

[16] A. S. A. Aziz *et al.*, "Multi-Layer Hybrid Machine Learning Techniques for Anomalies Detection and Classification Approach", in *Proc. of the 13th International Conference on Hybrid Intelligent Systems (HIS)*, 2013. https://doi.org/10.1109/HIS.2013.6920485

[17] E. E. Cureton and R. B. D'Agostino, "Factor Analysis an Applied Approach", London: Lawrence Erlbaum Associates, 1983.

[18] B. Zhang *et al.*, "Network Intrusion Detection Method Based on PCA and Bayes Algorithm", *Security and Communication Networks*, 2018. https://doi.org/10.1155/2018/1914980

[19] A. Jahanbani and H. Karimi, "A New Approach for Detecting Intrusions Based on the PCA Neural Networks", *Journal of Basic and Applied Scientific Research*, pp. 672–679, 2012.

[20] Z. Elkhadir *et al.*, "Intrusion Detection System Using PCA and Kernel PCA Methods", in *Proc. of the Mediterranean Conference on Information & Communication Technologies*, 2015. https://doi.org/10.1007/978-3-319-30298-0_50

[21] Y. Wang *et al.*, "Auto-Encoder Based Dimensionality Reduction", *Neurocomputing*, vol. 184, pp. 232–242, 2016. https://doi.org/10.1016/j.neucom.2015.08.104

[22] M. A. Kramer, "Nonlinear Principal Component Analysis Using Autoassociative Neural Networks", *AIChE*, vol. 37, 1991. https://doi.org/10.1002/aic.690370209

[23] M. Ring *et al.*, "Creation of Flow-Based Data Sets for Intrusion Detection", *Journal of Information Warfare*, vol. 16, no. 4, pp. 41–54, 2017.

[24] M. Tavallaeeet *et al.*, "A Detailed Analysis of the KDD CUPT 99 Dataset", *IEEE Symp. Comp. Int. Secand Def. Appl.*, 2009. http://dx.doi.org/10.1109/CISDA.2009.5356528

[25] M. Mohanapriya, "Comparative Study Between Decision Tree and knn of Data Mining Classification Technique", *Journal of Physics: Conference Series*, vol. 1142, no. 1, 2018. http://dx.doi.org/10.1088/1742-6596/1142/1/012011

[26] A. R. Vasudevan *et al.*, "SSENet-2011: A Network Intrusion Detection System Dataset and Its Comparison with KDD CUP 99 Dataset", in *Proc. of the 2nd Asian Himalayas International Conference on Internet (AH-ICI)*, Kathmandu, 2011, pp. 1–5. http://dx.doi.org/10.1109/AHICI.2011.6113948

[27] S. T. Ikram and C. K. Aswani, "Improving Accuracy of Intrusion Detection Model Using PCA and Optimized SVM", *Journal of Computing and Information Technology*, vol. 24, no. 2, pp. 133–148, 2016. https://doi.org/10.20532/cit.2016.1002701

[28] G. K. Kuchimanchi *et al.*, "Dimension Reduction Using Feature Extraction Methods for Real-Time Misuse Detection Systems", in *Proc. of the 5th Annual IEEE SMC Information Assurance Workshop*, 2004, pp. 195–202. http://dx.doi.org/10.1109/IAW.2004.1437817

Received: January 2019
Revised: January 2020
Accepted: March 2020

Contact addresses:
Meenal Jain
Department of Computer Science & Information Technology
Jaypee Institute of Information Technology
Noida
India
e-mail: meenaljain.bpl1988@gmail.com

Gagandeep Kaur
Department of Computer Science & Information Technology
Jaypee Institute of Information Technology
Noida
India
e-mail: gagandeep.kaur@jiit.ac.in

Meenal Jain received the M. Tech degree in Computer Science & Engineering from the Inderprastha Engineering College, Ghaziabad. University of Dr. A. P. J. Abdul Kalam Technical University, Lucknow, India in 2015. She is currently a PhD student in Jaypee Institute of Information Technology, Noida, India, Computer Science branch. Her research interests include machine learning, Big Data, and information security.

Gagandeep Kaur received her B.E. degree in Computer Science & Engineering from Punjab Technical University Jalandhar, Punjab, India. She received the M. Tech degree in Information Technology from PEC Chandigarh under Panjab University Chandigarh and PhD degree in Computer Engineering from JIIT, Noida, India. She is currently an Assistant Professor (Senior Grade) in the Jaypee Institute of Information Technology, Noida, India, Computer Science branch. Her research interests include Big Data networks, computer networks & networks security, wireless networks, data visualization & integration in R, performance measurement & evaluation metrics in information security, application of wavelets, self-similarity and machine learning techniques in networks security.