Comparative Study of Principal Component Analysis Based Intrusion Detection Approach Using Machine Learning Algorithms

Krupa Joel Chabathula
Department of IT
NITK Surathkal
Mangalore, India
Email: krupajoel@gmail.com

Jaidhar C.D
Department of IT
NITK Surathkal
Mangalore, India
Email: jaidharcd@nitk.ac.in

Ajay Kumara M.A
Department of IT
NITK Surathkal
Mangalore, India
Email: ajayit13f01@nitk.edu.in

***Abstract*—This paper induces the prominence of variegated machine learning techniques adapted so far for the identifying different network attacks and suggests a preferable Intrusion Detection System (IDS) with the available system resources while optimizing the speed and accuracy. With booming number of intruders and hackers in todays vast and sophisticated computerized world, it is unceasingly challenging to identify unknown attacks in promising time with no false positive and no false negative. Principal Component Analysis (PCA) curtails the amount of data to be compared by reducing their dimensions prior to classification that results in reduction of detection time. In this paper, PCA is adopted to reduce higher dimension dataset to lower dimension dataset. It is accomplished by converting network packet header fields into a vector then PCA applied over high dimensional dataset to reduce the dimension. The reduced dimension dataset is tested with Support Vector Machines (SVM), K-Nearest Neighbors (KNN), J48 Tree algorithm, Random Forest Tree classification algorithm, Adaboost algorihm, Nearest Neighbors generalized Exemplars algorithm, Navebayes probabilistic classifier and Voting Features Interval classification algorithm. Obtained results demonstrates detection accuracy, computational efficiency with minimal false alarms, less system resources utilization. Experimental results are compared with respect to detection rate and detection time and found that TREE classification algorithms achieved superior results over other algorithms. The whole experiment is conducted by using KDD99 data set.**

Keywords : Intrusion Detection Systems, Machine Learning Algorithms, Principal Component Analysis.

I. INTRODUCTION

Todays computer system has become vulnerable to many unknown attacks due to system software vulnerabilities. The firewall is able to scrutinize incoming and outgoing traffic and block the traffic as per written policy. Even if it detects an attack, generation of an alert is not a property of the firewall. To detect any kind of attack accurately and defend against them, efficient intrusion detection and prevention technique is highly crucial for current Internet world. To protect the network all the time against known attacks as well as unknown attacks, the best solution is Intrusion Detection System (IDS). Its primary goal to is to detect an intrusion followed by triggering an alter pertaining to detected attack in timely manner. The IDS requirement is to detect attacks those compromise the goals of security such as confidentiality, integrity, availability. Two major types of intrusion detection approach are misuse based detection and anomaly based detection. The former deals with known attack patterns which are known to the IDS while the later deals with both known and unknown attacks. Examples of misuse detection techniques are IDIOT [11], STAT [12] and Snort [13]. Major concern with misuse detection is inefficient to detect new attacks without the corresponding signature in the database. So new attacks can successfully bypass the misuse detection based approach which results in system damage. Anomaly based detection deals both known and unknown attacks and its detection efficiency is more compared to misuse detection based approach due to new attack detection capability. Examples of anomaly detection techniques are IDES [14] and EMERALD [15]. Anomaly based intrusion detection approach detects network behavior either as normal or abnormal and it is inefficient to spot exact type of attack. Major concern in anomaly based intrusion detection is generation of false alarms due to improper design of detection approach.

IDS can be broadly classified into two types: Host based Intrusion Detection System (HIDS) and Network based Intrusion Detection Systems (NIDS). The HIDS identifies and detects the threats that are target towards particular machine rather than the entire network. Its major function is to analyze system activity like CPU utilization, file integrity, log content verification etc. NIDS on the other hand determines the attacks that are arriving through network traffic. Various methods have been proposed in the literature for network based anomaly detection using machine learning algorithms, data mining, neural networks and statistical method based approach.

Principal Component Analysis (PCA) for network based intrusion detection measures the Mahalanobis distance of each observation from the center of the data for anomaly detection. The Mahalanobis distance is calculated by the sum of squares of the standardized principal component scores. In this work, PCA [10] adopted to reduce the detection time of network based intrusions. KDD99 dataset [7] has been used in this work to measure the detection rate of the following machine learning algorithms Support Vector Machines (SVM), K-Nearest Neighbors (KNN), J48 Tree algorithm, Random Forest Tree classification algorithm, Adaboost algorihm, Nearest Neighbors generalized Exemplars algorithm, Navebayes probabilistic classifier and Voting Features Interval classification algorithms.

978-1-4673-6823-0/15/$31.00

The rest of the paper is organized as follows. Overview of PCA [6] technique and machine learning algorithms are explained in section II. Methodology adopted in this work is discussed in section III. Section IV depicts the experimental results and section V concludes the paper.

II. Background

A. Principal Component Analysis

PCA [16] is the most used and most popular dimension reduction technique so far available due to its calculation flexibility and reversible approach. PCA is achieved with elimination of less significant attributes in high dimensional space which constitutes the major computational cost and projecting the most useful relevant attributes into a low dimensional subspace making much simpler. In other words if the vector is of length 'm' and there are 'n' observations then the matrix (A) can be represented as

Every column is a vector, thus shown as in equation 1.

$$A_{m\times n} = [A_1, A_2, A_3, ..., A_n] \tag{1}$$

The average mean () of every vector is calculated using equation 2.

$$\mu = \frac{1}{n}\Sigma_{i=1}^{n} A_{ij} \tag{2}$$

Deviation is a metric used to calculate the variation of a vector from its mean. The deviation is calculated as in equation 3.

$$\Phi = A_i - \mu \tag{3}$$

The co variance is a metric used to measure the degree of relationship between the two variables. The positive value of the result indicates the two variables are positively correlated, negative value resembles the negatively correlated data and the zero value suggests the data is not correlated. We get to know the spread of the data through the co variance matrix. The co variance matrix is then constructed over the square matrix with number of classes as the dimension as in equation 4.

$$B_{m\times n} = \frac{1}{n-1}\Sigma_{i=1}^{n}\Phi_i\Phi_i^t\frac{1}{n-1}\Sigma_{i=1}^{n}(X_i-\mu)(X_i-\mu)^t \tag{4}$$

Where Φ^t is the transpose of the matrix Φ.

To perform PCA over the covariance matrix, calculation of Eigen-values and Eigen vectors is generally employed through Singular Value Decomposition (SVD). Say $(\lambda_1, u_1), (\lambda_2, u_2), ..., (\lambda_m, u_m)$ are the 'm' eigen-value eigen-vector pairs of the covariance matrix 'B'. Then we choose the most highest P eigen-values which contributes more to classification purpose while the remaining m-P values are of low significance which mostly contains noise data. The reduced subspace can be calculated using the equation 5

$$\frac{\Sigma_{i=1}^{p}\lambda_i}{\Sigma_{i=1}^{m}\lambda_i} \geq S \tag{5}$$

Where $'S'$ is the ratio of variation in the reduced subspace to the total variation in the high dimensional space. We thus obtain MXP matrix (Y_i) containing the P eigenvectors in the columns. The data represented by the principal components into the reduced P dimensional subspace is according to equation 6.

$$Y_i = U^t\Phi_i = U^t(X_i - \mu) \tag{6}$$

B. Machine Learning Algorithms

Intrusion detection is accomplished by classifying the packet into normal or one of the attacks. There are many prevailing classification algorithms but machine learning algorithms have added new era for unknown patterns classification to improve the detection rate [1]. A few machine learning algorithms are discussed in this section.

Naive-based Classification Algorithm : It is well-known probabilistic model for classification and uses a simple conditional probabilistic equation [9]. In general Nave-based algorithm uses the basic approach like Posterior = $\frac{PriorXLikelihood}{evidence}$. If there are 'n' features and dependent class variable is $'\alpha'$ then the Bayesian conditional distribution under independent assumptions is in equation 7.

$$p(\alpha_k|\beta_1, ..., \beta_n) = \frac{1}{z}p(\alpha_k)\Pi_{i=1}^{n}p(X_i|\alpha_k) \tag{7}$$

where $Z = p(\beta)$ is a scaling factor dependent on $\beta_1, ..., \beta_n$ The classifier model for some k can be formulated as in equation 8.

$$\Gamma = argmax_{(k\in 1,...,K)}p(\alpha_k)\Pi_{i=1}^{n}p(X_i|\alpha_k) \tag{8}$$

K-Nearest Neighbors (KNN) Algorithm: KNN is a lazy algorithm which calculates the K-nearest training samples in the feature space. The nearest neighbors are given the maximum weight where weight is assigned as 1/d, where $'d'$ is the distance to the neighbor. Euclidean distance is used as a distance metric for KNN and is calculated as $\sqrt{(x_2-x_1)^2+(y_2-y_1)^2}$ in a 2-dimensional space, where $(x_1, y_1) and (x_2, y_2)$are the coordinates in the 2-dimensional space. KNN primarily depends on the K-value chosen [8].

Support Vector Machines (SVM): It is a non probabilistic linear classifier which constructs the hyper planes in high dimensional space. The hyper plane be $f(x) = \beta_0 + \beta^t$x, where β is known as the weight vector and β_0 as the bias. By scaling β and β_0 an optimal plane can be constructed. Let $|\beta+\beta_0| = 1$ be a hyper plane and the training vectors closest to the training plane are called the support vectors. The distance between a point x and the hyper plane is denoted by $\frac{|\beta+\beta^t x|}{||\beta||}$. So the distance is $\frac{1}{||\beta||}$ has to be minimized to obtain optimal hyper-planes classification [4].

Adaboost Algorithm : This is the boosting algorithm which combines multiple weak classifiers and emerges as a strong classifier. It helps to determine the training dataset on its own based on the results of the previous classifier and also to determine the weight of each classifier when combining the results [5]. The final classifier is shown by the equation 9.

$$H(x) = sin\Sigma_{t=1}^{T}\alpha_t h_t(x) \tag{9}$$

Where T is the total weak classifiers and h(x) is the output of the weak classifier and $'\alpha'$ is weight associated to the classifier and $'\alpha'$ is computed using equation 10.

$$\alpha_t = \frac{1}{2}log(\frac{1-\epsilon_t}{\epsilon_t}) \tag{10}$$

Where ϵ_t is the number of misclassification of that weak classifier. The updation of the training weights is done using the equation 11.

$$D_{t+1}(i) = \frac{D_t(i)e^{-\alpha_t y_i h_t(x_i)}}{Z_t} \tag{11}$$

Where D_t is the vector of weights and 'i' is the training number.

J48 Tree Algorithm : In this method binary tree is constructed to model the classification process and compared to each tuple in the database which finally results in the classification. One primary disadvantage with this algorithm is that it ignores missing values.

J48 Tree Algorithm [17]

```
INPUT : D // Training Data
OUTPUT : T // Decision Tree
DT Build(*D)
{
   T = Φ
   T= Create root node and label with splitting attribute;
   T= Add arc to root node for each split predicate and label;
   For each arc do
    D= Database created by applying splitting predicate to D;
   If stopping point reached for this path, then
    T'= create leaf node and label with appropriate class;
   Else
    T'= DTBUILD(D);
   T= add T' to arc;
}
```

Nearest Neighbors Generalized Exemplars Algorithm (NNGE) : This is a probabilistic distribution classification model and is based on a point of reference for classifying. Let R^d be the Euclidean space and N be the point process [19]. To classify a point x , the nearest neighbor function is defined as in equation 12.

$$D_0(r) = 1 - P(N(b(o,r)) = 1|0) \tag{12}$$

Where $P(N(b(o,r)) = 1|0)$ denotes the conditional probability that there is one point of N located in b(o, r) given there is a point of N located at o. For the reference point not at origin but at point x is defined using equation 13.

$$D_x(r) = 1 - P(N(b(x,r)) = 1|0) \tag{13}$$

where $x \in R^d$.

Random Forest Tree Algorithm : It was the fastest algorithm since it builds trees of height log(k) where k is the number of attributes and provides better accuracy due to no pruning [3].

Voting Features Interval (VFI) : Each feature participates in the classification by distributing real-valued votes among classes and is considered to be a non-incremental classification algorithm. The class receiving the highest vote is declared to be the predicted class [2].

Voting Features Interval Algorithm [18]

```
classify (x): //x is the vector to be classified
begin:
      for each class c
         vote(c) = 0;
      for each feature f
         for each class c
            feature_vote[f,c] = 0; vote of feature f for class c
         if x_f value is known
            i = find_interval(f, e_f)
            feature_vote[f,c] = interval_class_count[f,i,c] / class_count[c]
            normalize_feature_vote[f];     //such that
                                  // Σ_c feature_vote[f, c] = 1
            for each class c
               vote[c] = vote [c] + feature_vote[f,c];
      return class c with highest vote[c];
end;
```

III. Methodology Adopted in this Work

Network attack analysis and identification is highly desirable for incoming traffic. Firstly header fields of every incoming packet is analyzed by aligning in the form of vector and every vector represents a data connection or a data packet. Network traffic is been monitored for some amount of time to prepare the dataset. The obtained data is transformed into vectors which serves as input for the PCA algorithm. In this work, the system is designed for two phases namely training phase and testing phase. In training phase, network data is collected and preprocessed such that the rotten attributes are eliminated. PCA is applied over preprocessed data to bring down high dimensional data to low dimension data. Dimension reduced data is used in training to build base profile of normal traffic as well as abnormal traffic. In testing phase also, the test records are preprocessed to reduce the dimension from high to low prior to classification using PCA. Applied test records are compared with base profile created during training phase and then record is classified accordingly. A confusion matrix is constructed over the obtained classification results. Accuracy and detection time of each algorithm is measured for each test record. Classification result is either normal pattern record or one among the trained attack patterns. Alarm is set to generate if the test record is classified into any of the attack patterns. The complete flow procedure followed in this experiment is as shown in Figure 1.

IV. Experimental results and Discussion

Machine learning algorithms such as Support Vector Machines (SVM), K-Nearest Neighbors (KNN), J48 Tree, Random Forest Tree classification, Adaboost, Nearest Neighbors generalized Exemplars, Navebayes probabilistic classifier and Voting Features Interval classification are used in this experimental work to test detection accuracy on network based attacks by utilizing KDD99 dataset as training data as well as testing data. Data set provides the data for the following attacks: A) DoS (Denial-of-Service, e.g. syn flood). B). R2L - Unauthorized access from a Remote Machine, e.g. guessing password. C) U2R-unauthorized access to local super user (root) privileges e.g., various "buffer overflow" attacks. D). Probing -surveillance and other probing, e.g., port scanning. The whole experiment is conducted on the machine with following specifications. Processor : Intel(r) Core(TM) i7 2600 CPU @ 3.40 GHz, Installed Memory 8 GB (RAM) and 64 bit System type. The classification is made using the prominent

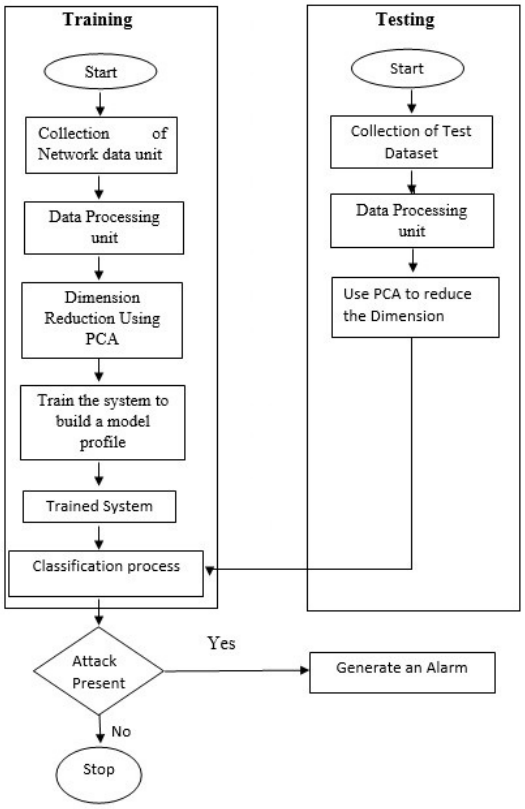


Fig. 1: Flow procedure followed in this experiment

tool such as Weka. In addition, Weka tool widely used for clustering as all machine learning algorithms are part of it.

Two different experiments namely without-PCA and with-PCA are conducted to demonstrate the comparison between them. The 10 percent of KDD99 dataset has been considered with 11 attack patterns and 150000 records into the file. The number of attack records taken for each attack pattern are shown in table 1. A testing file is used for testing accuracy and speed of with-PCA and without-PCA approach on 8 different machine learning algorithms. Figure 2 provides comparison results of 5 dimensioned with-PCA and 42 dimensioned without-PCA approach in terms of detection accuracy. Detection rate of Support Vector Machines (SVM), K-Nearest Neighbors (KNN), J48 Tree, Random Forest Tree classification, Adaboost, Nearest Neighbors generalized Exemplars is almost same in both with-PCA and without-PCA approach. However, identification rate of Navebayes probabilistic classifier is higher in with-PCA based approach compare to without-PCA approach. Detection rate of with-PCA in Voting Features Interval classification is lesser compared to without-PCA based approach. The results suggested that the tree based algorithms namely J48 and Random-forest outperformed the other algorithms both in terms of accuracy and speed. Figure 3 provides comparison results of 5-dimension with-PCA and 42- dimension without-PCA approach in terms of detection time. Detection time is much smaller for all the algorithms in with-PCA approach to that of without-PCA approach. This due to dimension reduction that eliminates overhead computation. Among all the algorithms, Voting features interval has incurred less detection time. Primarily two reasons for quick detection are: 1) normalization is not needed for the features because each feature is processed separately. 2) The features containing missing values are ignored because of their unknown characteristic.

TABLE I: Records Count For Attacks

Attack Types	Normal	Abnormal	Total
Recordsd used for training	29953	120047	150000
Records used for testing	2995	12005	15000

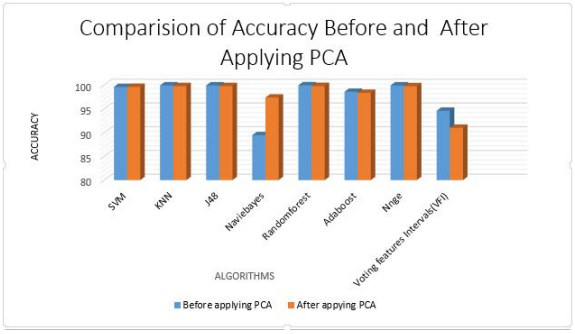


Fig. 2: Comparison of Detection Accuracy Before and After Applying PCA.

The output of PCA is reduced low dimensional dataset and the number of dimensions to be selected can be set manually by altering the standard deviation limit. The datasets with different dimensions namely 5, 6, 7, 8, 9, 10, 13, 14, 15, 18, 24, and 42 are taken to compare performance of eight machine learning algorithms to find the minimum dimensions which gives high accuracy rate and low detection time. Figure 4 depicts detection time of eight different machine learning algorithms for different dimensions. Figure 5 depicts detection accuracy of eight different machine learning algorithms for different dimensions. It has found that the detection time has

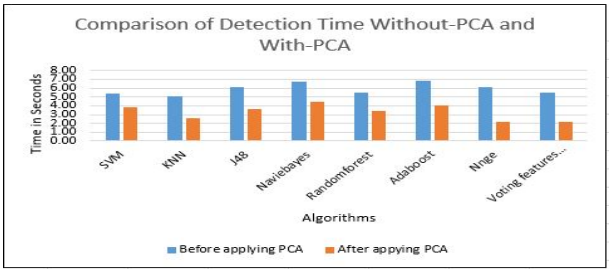


Fig. 3: Comparison of Detection Time before and After Applying PCA

linear growth till 18 dimensions and then showed a steady increase. Most algorithms showed rapid increase of accuracy till 7 dimensions and increased steadily thereafter. So this work suggests 7 dimensions are good enough for optimal accuracy as well as to reduce computational cost and system resources. The behavior of the 8 algorithms under PCA with different dimensions are shown separately in figures 6,7,8,9,10,11,12,13.

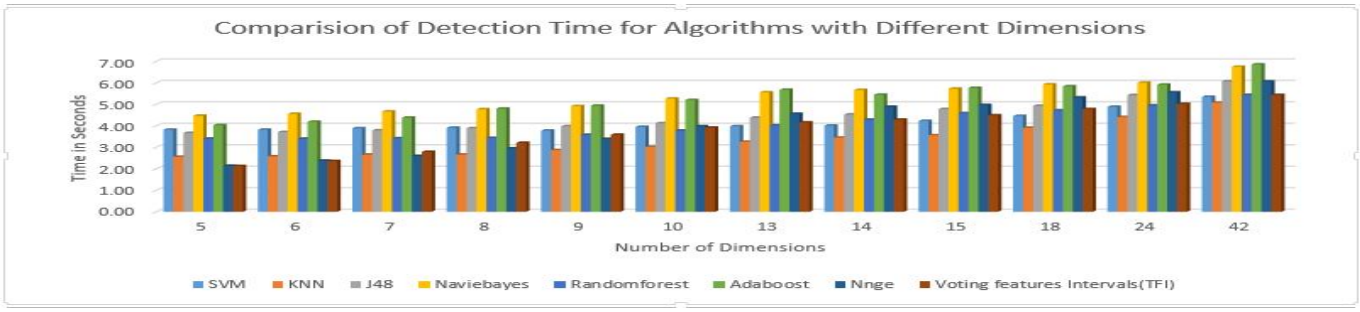
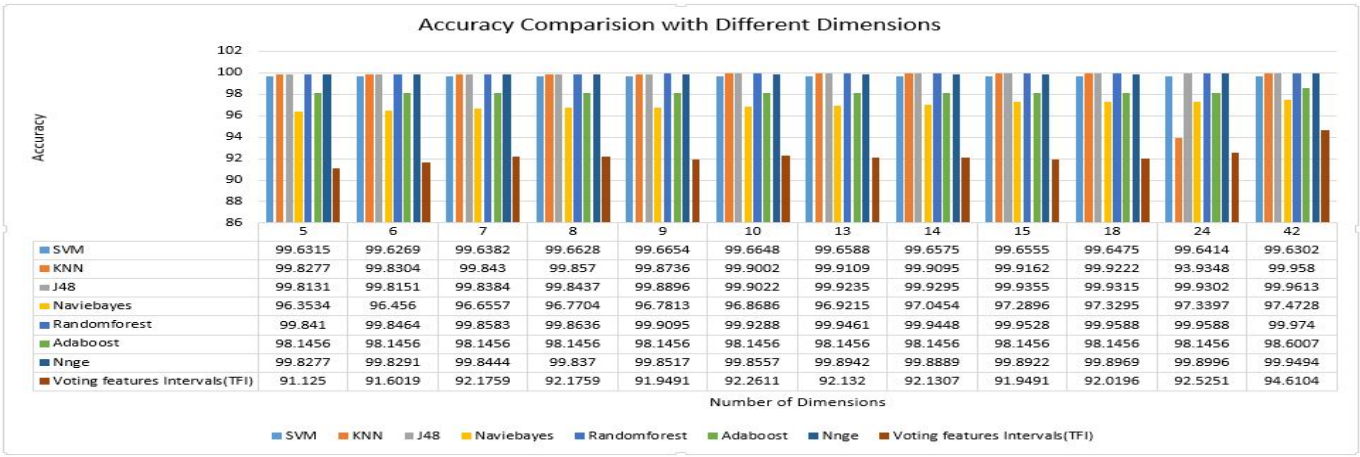


Fig. 4: Time Comparison with Different Dimensions



	5	6	7	8	9	10	13	14	15	18	24	42
SVM	99.6315	99.6269	99.6382	99.6628	99.6654	99.6648	99.6588	99.6575	99.6555	99.6475	99.6414	99.6302
KNN	99.8277	99.8304	99.843	99.857	99.8736	99.9002	99.9109	99.9095	99.9162	99.9222	93.9348	99.958
J48	99.8131	99.8151	99.8384	99.8437	99.8896	99.9022	99.9235	99.9295	99.9355	99.9315	99.9302	99.9613
Naviebayes	96.3534	96.456	96.6557	96.7704	96.7813	96.8686	96.9215	97.0454	97.2896	97.3295	97.3397	97.4728
Randomforest	99.841	99.8464	99.8583	99.8636	99.9095	99.9288	99.9461	99.9448	99.9528	99.9588	99.9588	99.974
Adaboost	98.1456	98.1456	98.1456	98.1456	98.1456	98.1456	98.1456	98.1456	98.1456	98.1456	98.1456	98.6007
Nnge	99.8277	99.8291	99.8444	99.837	99.8517	99.8557	99.8942	99.8889	99.8922	99.8969	99.8996	99.9494
Voting features Intervals(TFI)	91.125	91.6019	92.1759	92.1759	91.9491	92.2611	92.132	92.1307	91.9491	92.0196	92.5251	94.6104

Fig. 5: Detection Accuracy Comparison with Different Dimensions

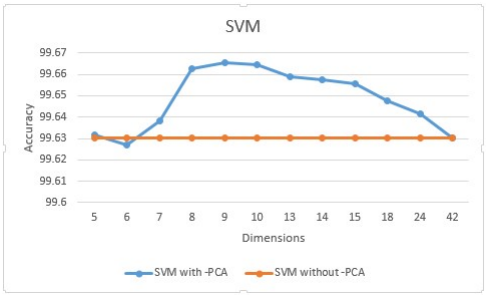


Fig. 6: Comparison of SVM under PCA

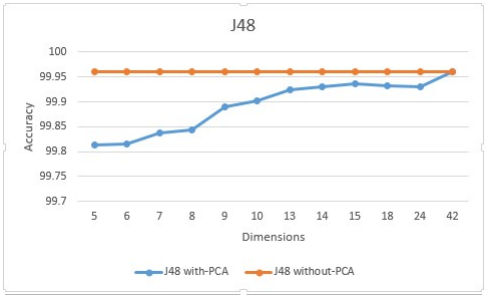


Fig. 8: Comparison of J48 under PCA

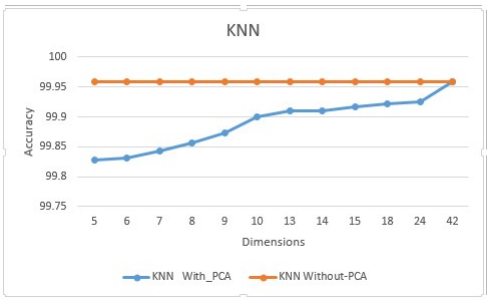


Fig. 7: Comparison of KNN under PCA

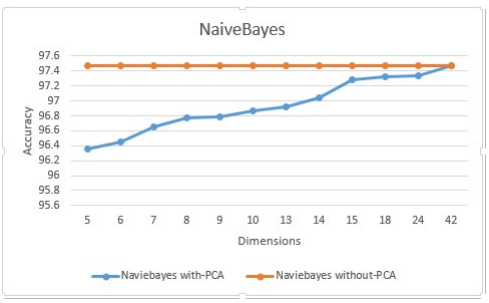


Fig. 9: Comparison of Naives-Bayes under PCA

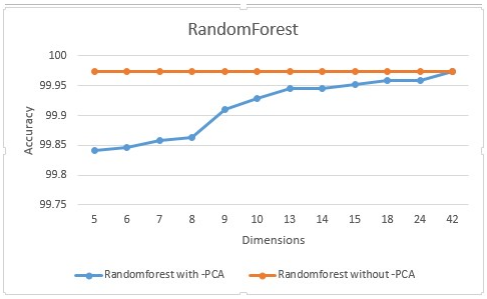


Fig. 10: Comparison of Random-Forest Under PCA

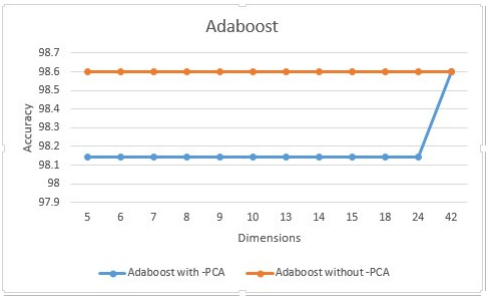


Fig. 11: Comparison of Adaboost under PCA

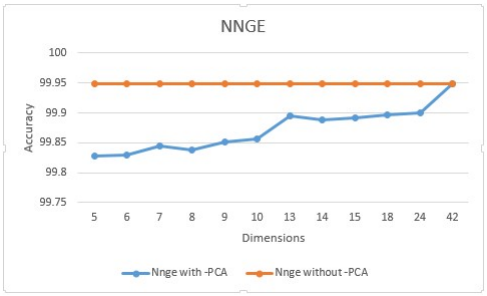


Fig. 12: Comparison of NNge under PCA

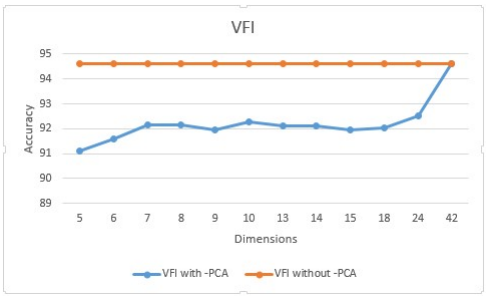


Fig. 13: Comparison of VFI under PCA

V. Conclusion

In this paper, with-PCA and without-PCA based network based intrusion detection approaches are compared using Support Vector Machines (SVM), K-Nearest Neighbors (KNN), J48 Tree, Random Forest Tree classification, Adaboost, Nearest Neighbors generalized Exemplars, Navebayes probabilistic classifier and Voting Features Interval classification machine learning algorithms. Both high dimensional dataset and low dimensional dataset have been used in classification. It is observed through experimental result that with-PCA based approach exhibits less detection time compared to without-PCA based approach. Detection rate of with-PCA based approach increases as dimension size increases. Tree algorithms outperformed all other algorithms used in this experiment in terms of accuracy and speed. Future work is plan to built a self-adaptive IDS onto which with-PCA based approach will be incorporated so that records updates dynamically thereby increasing the detection accuracy and reduces the training time.

References

[1] Witten IH, Eibe F, Hall MA. Data Mining: Practical Machine Learning Tools and Techniques. Thirdth. Morgan Kaufmann, San Francisco; 2011.

[2] Demirz, G., & Gvenir, H. A. (1997). Classification by voting feature intervals. In Machine Learning: ECML-97 (pp. 85-92). Springer Berlin Heidelberg.

[3] Biau, G. (2012). Analysis of a random forests model. The Journal of Machine Learning Research, 98888(1), 1063-1095.

[4] Shang-Fu, G., & Chun-lan, Z. (2012, July). Intrusion detection system based on classification. In Intelligent Control, Automatic Detection and High-End Equipment (ICADE), 2012 IEEE International Conference on (pp. 78-83). IEEE.

[5] Hu, Wei, and Weiming Hu. "Network-based intrusion detection using adaboost algorithm." Web Intelligence, 2005. Proceedings. The 2005 IEEE/WIC/ACM International Conference on. IEEE, 2005.

[6] Mechtri, Leila, F. Djemili Tolba, and Nacira Ghoualmi. "Intrusion detection using principal component analysis." Engineering Systems Management and Its Applications (ICESMA), 2010 Second International Conference on. IEEE, 2010.

[7] Araujo, Nelcileno, et al. "Identifying important characteristics in the KDD99 intrusion detection dataset by feature selection using a hybrid approach."Telecommunications (ICT), 2010 IEEE 17th International Conference on. IEEE, 2010.

[8] Kuang, Liwei, and Mohammad Zulkernine. "An Intrusion-Tolerant Mechanism for Intrusion Detection Systems." Availability, Reliability and Security, 2008. ARES 08. Third International Conference on. IEEE, 2008.

[9] Gumus, Fatma, et al. "Online Naive Bayes classification for network intrusion detection." Advances in Social Networks Analysis and Mining (ASONAM), 2014 IEEE/ACM International Conference on. IEEE, 2014.

[10] Jiang jian-chun,Ma heng-tai.Network Security Intrusion Detection: Review. Journal of software. 2000,11(11):460-1467.

[11] S.Kumar, E.H.Spafford, A Software architecture to support misuse intrusion detection, Proceedings of the 18th National Information Security Conference, pp.194-204, 1995.

[12] K.Ilgun, R.A.Kemmerer, P.A.Porras, State transition analysis: A rule-based intrusion detection approach, IEEE Transactions on Software Engineering, vol.21, no.3, pp.181- 199, 1995.

[13] J.Beale, Caswell (Editor), Snort 2.1 Intrusion Detection (Second Edition), Syngress, 2004.

[14] T.Lunt, A.Tamaru, F.Gilham, et al, A Real-time Intrusion Detection Expert System (IDES) - final technical report, Technical report, Computer Science Laboratory, SRI International, Menlo Park, California, Feb.1992.

[15] P.A.Porras, P.G.Neumann,EMERALD: Event monitoring enabling responses to anomalous live disturbances, Proceedings of National Information Systems Security Conference, Bal-timore MD, Oct. 1997.

[16] Jolliffe, Ian. Principal component analysis. John Wiley & Sons, Ltd, 2005.

[17] Dunham, Margaret H. Data mining: Introductory and advanced topics. Pearson Education India, 2006.

[18] Demirz, Glen, and H. Altay Gvenir. "Classification by voting feature intervals." Machine Learning: ECML-97. Springer Berlin Heidelberg, 1997. 85-92.

[19] D.Stoyan, W. S. Kendall, J. Mecke, and L. Ruschendorf. Stochastic geometry and its applications, volume 2. Wiley Chichester, 1995.