Data Mining Approaches for Network Intrusion Detection: from Dimensionality Reduction to Misuse and Anomaly Detection

Iwan Syarif [1,2], Adam Prugel-Bennett[1], Gary Wills[1]
[1]School of Electronics and Computer Science
University of Southampton, UK
[2]Eletronics Engineering Polytechnics Institute of Surabaya
Indonesia
{is1e08,apb,gbw}@ecs.soton.ac.uk, iwanarif@eepis-its.edu

ABSTRACT: *This paper describes the use of data mining techniques to solve three important issues in network intrusion detection problems. The first goal is finding the best dimensionality reduction algorithm which reduces the computational cost while still maintains the accuracy. We implement both feature extraction (Principal Component Analysis and Independent Component Analysis) and feature selection (Genetic Algorithm and Particle Swarm Optimization) techniques for dimensionality reduction. The second goal is finding the best algorithm for misuse detection system to detect known intrusion. We implement four basic machine learning algorithms (Naïve Bayes, Decision Tree, Nearest Neighbour and Rule Induction) and then apply ensemble algorithms such as bagging, boosting and stacking to improve the performance of these four basic algorithms. The third goal is finding the best clustering algorithms to detect network anomalies which contains unknown intrusion. We analyze and compare the performance of four unsupervised clustering algorithms (k-Means, k-Medoids, EM clustering and distance-based outlier detection) in terms of accuracy and false positives.*

Our experiment shows that the Nearest Neighbour (NN) classifier when implemented with Particle Swarm Optimization (PSO) as an attribute selection algorithm achieved the best performance, which is 99.71% accuracy and 0.27% false positive. The misuse detection technique achieves a very good performance with more than 99% accuracy when detecting known intrusion but it fails to accurately detect data set with a large number of unknown intrusions where the highest accuracy is only 63.97%. In contrast, the anomaly detection approach shows promising results where the distance-based outlier detection method outperforms the other three clustering algorithms with the accuracy of 80.15%, followed by EM clustering (78.06%), k-Medoids (76.71%), improved k-Means (65.40%) and k-Means (57.81%).

Keywords: Intrusion detection system, Anomaly detection, Misuse detection, Feature selection, Clustering, Ensemble classifiers

Received: 17 January 2012, Revised 21 March 2012, Accepted 28 March 2012

1. Introduction

Intrusion detection is a process of gathering intrusion-related knowledge occurring in the process of monitoring events and analyzing them for signs of intrusion [8][17]. There are two basic IDS approaches: misuse detection (signature-based) and anomaly detection. The misuse detection system uses patterns of well-known attacks to match and identify known intrusions. It performs pattern matching between the captured network traffic and attack signatures. If a match is detected, the system generates an alarm. The main advantage of the signature detection paradigm is that it can accurately detect instances of known attacks. The main disadvantage is that it lacks the ability to detect new intrusions or zero-day attacks [17][6].

The anomaly detection model works by identifying an attack by looking for behaviour that is out of the normal. It establishes a baseline model of behaviour for users and components in a computer or network. Deviations from the baseline cause alerts that direct the attention of human operators to the anomalies [6][4][18]. This system searches for anomalies either in stored data or in the system activity. The main advantage of anomaly detection is that it does not require prior knowledge of an intrusion and thus can detect new intrusions. The main disadvantage is that it may not be able to describe what constitutes an attack and may have a high false positive rate [17][6][4].

2. Dimensionality Reduction

Most of the current IDS handle huge amount of data with many features which are derived from network traffics. Some of the features may be redundant or make less of a contribution to the detection process [3]. Selecting the best dimensionality reduction algorithm is one of the most important factors that affect the IDS performance [14].

Dimensionality reduction is the process of reducing the number of random variables under consideration. There are two techniques of dimensionality reduction. The first technique is feature extraction which refers to the mapping of the original high-dimensional data onto a lower-dimensional space. In this technique, all the original features are combined into a new reduced set of features. The second technique is feature selection which is a process that chooses an optimal subset of features according to an objective function. In other words, this technique selects only the most relevant features/attributes.

2.1 Feature Extraction

There are many feature extraction reduction algorithms, but in this paper we only select two examples of them which are Principal Component Analysis (PCA) and Independent Component Analysis (ICA). PCA is one of the most widely used dimensionality reduction techniques for data analysis and compression. It is based on transforming a large number of variables into a smaller number of uncorrelated variables by finding a few orthogonal linear combinations of the original variables with the largest variance.

Independent Component Analysis (ICA) is a statistical and computational technique for revealing hidden factors that underlie sets of random variables, measurements, or signals. Draper et. al. [5] have compared the performance of PCA and ICA in the face/ image recognition problems. They reported that ICA outperforms PCA on visible light image, but on the other hand PCA outperforms ICA in another different type of images. In this report, we would like to compare both algorithms in the field of IDS. The details of PCA and ICA algorithms are explained in [5][24].

2.2 Feature Selection

Feature selection is a popular technique used to find the most important and optimal subset of features for building powerful learning models. An efficient feature selection method can eliminate irrelevant and redundant data; hence it can improve the classification rate. There are a lot of feature selection techniques, but in this paper we only select two algorithms: Genetic Algorithm (GA) and Particle Swarm Optimization (PSO). GA is a heuristic search or optimization technique for obtaining the best possible solution in a vast solution space. GA can be applied in feature selection because this problem has an exponential search space. The detail information of feature selection using GA is explained in [15].

Particle Swarm Optimization (PSO) was first introduced by Kennedy and Eberhart [9] and was inspired by the social behaviour of bird flocking or fish schooling. Compare to the GA, PSO is simpler and easier to implement with few parameters. This algorithm is a very powerful and widely used to solve optimization problems as well as feature selection problems [11]. The algorithm is explained in more detail in [9] [11].

3. Misuse Detection System

In the first experiment, we apply four basic machine learning algorithms to the misuse detection module then we apply some ensemble algorithms to improve the performance.

3.1 Misuse Detection System Design

Our misuse detection module consists of four phases: dimensionality reduction, classification algorithms, performance measurement and performance analysis as shown in Figure 1 below.

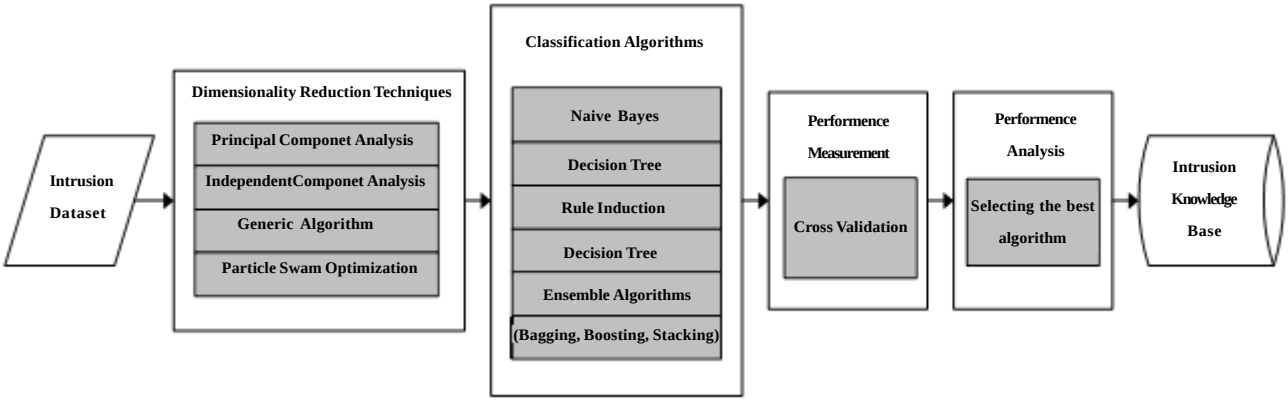


Figure 1. Misuse Detection System Design

We applied four different classifiers: Naïve Bayes (NB), Decision Tree (DT), Rule Induction (RI) and Nearest Neighbour (NN) into the misuse detection module in order to find the best method of detecting intrusion based on accuracy, false positive and speed (computation time).

3.2 Ensemble Approaches for Misuse Detection System

Ensemble approaches [13][19][7] have the advantage that they can be made to adopt the changes in the stream more accurately than single model techniques. The ensemble classification technique is more advantageous and has better accuracy than the single classification method. It is a combination of several base models and is used for continuous learning. In this paper we evaluated and analyzed three different ensemble classifier techniques, called bagging, boosting and stacking, using various weak classifiers, such as NN, DT, RI and NB.

Bagging was first introduced by Leo Breiman [1] to reduce the variance of a predictor. It uses multiple versions of a training set which is generated by a random draw with the replacement of N examples where N is the size of original training set. Each of these data sets is used to train a different model. The outputs of the models are combined by voting to create a single output. Details of the bagging algorithm and its pseudo-code were given in [26].

Boosting, which was introduced by Schapire et al. [20], is an ensemble method for boosting the performance of a set of weak classifiers into a strong classifier. This technique can be viewed as a model averaging method and it was originally designed for classification, but it can also be applied to regression. Boosting provides sequential learning of the predictors. The first one learns from the whole data set, while the following learns from training sets based on the performance of the previous one. The misclassified examples are marked and their weights increased so they will have a higher probability of appearing in the training set of the next predictor. This results in different machines being specialized in predicting different areas of the dataset[7].

Stacking or *stacked generalization* is a different technique of combining multiple classifiers. Unlike bagging and boosting, stacking is usually used to combine various different classifiers, e.g. DT, NN, RI, NB, logistic regression, etc. Stacking consists of two levels which are base learner as level-0 and stacking model learner as level-1. Base learner (level-0) uses many different models to learn from a dataset. The outputs of each of the models are collected to create a new dataset. In the new dataset, each instance is related to the real value that it is supposed to predict. Then that dataset is used by stacking model learner (level-1) to provide the final output [7].

4. Anomaly Detection System

Unlike misuse detection which requires labeled data set and supervised algorithms, our proposed anomaly detection works on unlabeled dataset and uses unsupervised clustering algorithms. We apply four different clustering algorithms as explained in the following section.

4.1 Clustering Algorithms

Clustering is a technique for finding patterns in unlabelled data with many dimensions. Clustering has attracted interest from

researchers in the field of intrusion detection [18][2]. The main advantage of clustering algorithm is the ability to learn from and detect intrusions in the audit data without explicit descriptions (intrusion signatures) which usually provided by security experts. In this paper, we implement and compare the performance of four different clustering algorithms in our anomaly detection module which are k-Mean, k-Medoids, EM clustering and distance-based outlier detection algorithm.

4.1.1 k-Means

k-Means which is firstly proposed by James MacQueen, is a well-known and widely used clustering algorithm. k-Means is one of the simplest clustering algorithms in machine learning which can be used to automatically recognize groups of similar instances/items/objects/points in data training. The algorithm classifies instances to a pre-defined number of clusters specified by the user (e.g. assume *k* clusters). The first important step is to choose a set of k instances as centroids (centres of the clusters) randomly, usually choose one for each cluster as far as possible from each other. Next, the algorithm continues to read each instance from the data set and assigns it to the nearest cluster. There are some methods to measure the distance between instance and the centroid but the most popular one is Euclidian distance. The cluster centroids are always recalculated after every instance insertion. This process is iterated until no more changes are made.

4.1.2 k-Medoids

k-Medoids is a clustering algorithm similar to k-Means, which attempts to minimize the distance between points and its centre (centroid). A medoid is a data point which acts as an exemplar for all other data points in the cluster. The k-Means algorithm is very sensitive to outliers because if there is an object with a very large value, the data distribution may be biased or distorted [23]. In this case, k-Medoids is more robust to noise and outliers because in this algorithm the partitioning method is performed based on the principle of minimizing the sum of dissimilarities between each object in a cluster. The detail of k-Medoids algorithm is explained in [23].

4.1.3 EM Clustering

Expectation Maximization (EM) clustering is a variant of k-Means clustering and is widely used for density estimation of data points in an unsupervised clustering [21]. In the EM clustering, we use an EM algorithm to find the parameters which maximize the likelihood of the data, assuming that the data is generated from k normal distributions. The algorithm learns both the means and the covariance of the normal distributions. This method requires several inputs which are the data set, the total number of clusters, the maximum error tolerance and the maximum number of iteration.

The EM can be divided into two important steps which are Expectation (E-step) and Maximization (M-step). The goal of E-step is to calculate the expectation of the likelihood (the cluster probabilities) for each instance in the dataset and then re-label the instances based on their probability estimations. The M-step is used to re-estimate the parameters values from the E-step results. The outputs of M-step (the parameters values) are then used as inputs for the following E-step. These two processes are performed iteratively until the results convergence. The mathematical formulas of EM clustering are described in [21][12] and the pseudo codes can be found in [12] .

4.1.4 Outlier Detection Algorithms

Outlier detection is a technique to find patterns in data that do not conform to expected behaviour [2]. Most of the clustering algorithms do not assign all points to clusters but account for noise objects, in other words clustering algorithms are optimized to find clusters rather than outliers. Outlier detection algorithms look for outliers by applying one of the clustering algorithms and retrieve the noise set, therefore the performance of outlier detection algorithms depends on how good the clustering algorithm captures the structure of clusters.

The distance-based outlier detection approach, which is based on the Nearest Neighbour algorithm was first introduced by Ng et al [10] and implements a well-defined distance metric to detect outliers, the greater the distance of the object to its neighbour, the more likely it is to be an outlier. This method calculates the distance between each pair of objects using a nested loop (NL) algorithm and then the objects which are far away from the majority are signed as outliers. The mathematical formulas of distance-based outlier detection methods and their pseudo codes are described in more details [10][16].

4.2 Anomaly Detection Module

We designed the anomaly detection module as shown in Figure 2 below. This module implements several unsupervised clustering algorithms which do not required labeled dataset. In the feature extraction module we select only numerical data and handle missing value, then we transform the data into normal form. Normalization is a popular method used to convert all attributes/

variables to a common scale with an average of zero and standard deviation of one.

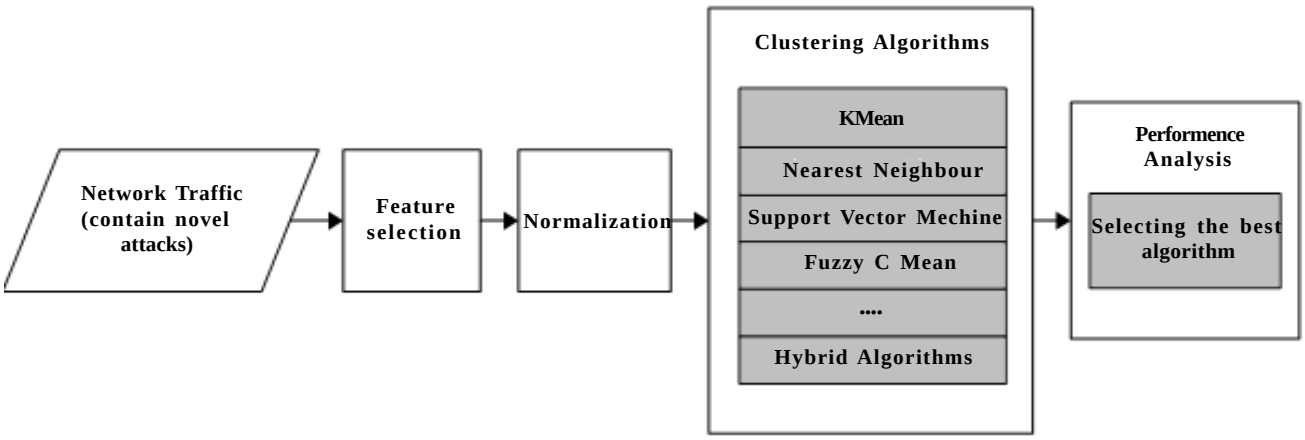


Figure 2. Anomaly Detection System Design

5. Experimental Setup

The following section describes the intrusion data sets used in the experiment, the performance metric used to evaluate the proposed system and the experimental settings and its results.

5.1 Intrusion Dataset

One of the most widely used data sets for evaluating IDS is the DARPA/Lincoln Laboratory off-line evaluation dataset or IDEVAL[1]. IDEVAL is the most comprehensive test set available today and it was used to develop the 1999 KDD Cup data mining competition[2]. In this experiment, we use the NSL-KDD intrusion data, which was provided to solve some problems in KDD'99, particularly that its training and test sets contained a huge number of redundant records with about 78% and 75% of the records being duplicated in the training and test sets, respectively. This may cause the classification algorithms to be biased towards these redundant records and thus prevent it from classifying other records [22].

Intrusions which exist in both training and testing data	Intrusions which only exist in testing data
back, buffer_overflow, ftp_write, guess_passwd, imap, ipsweep, land, loadmodule, multihop, neptune, nmap, phf, pod, portsweep, rootkit, satan, smurf, spy, teardrop, warezclient, warezmaster	apache2, httptunnel, mailbomb, mscan, named, perl, processtable, ps, saint, sendmail, snmpgetattack, snmpguess, sqlattack, udpstorm, worm, xlock, xsnoop, xterm

Table 1. List of intrusions in training and testing data

The intrusion data set consists of forty different intrusions classified into four main categories: DoS (Denial of Service), R2L (Remote to Local Attack), U2R (User to Root Attack) and Probing Attack. The training dataset consists of 25,191 instances and the testing dataset consists of 11,950 instances. The testing data set has many intrusions, which do not exist in the training data, as shown in Table 1.

5.2 Performance Metric

We use accuracy rate and false positive rate as the performance criteria based on the following metric shown in Table 2 below.

[1]http://www.ll.mit.edu/mission/communications/ist/corpora/ideval/data/index.html

[2]http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html

		Actual Result	
		Intrusion	Normal
	Intrusion	True Positive (TP)	False Positive (FP)
	Normal	False Negative (FN)	True Negative (TN)

Table 2. Performance metric

True Positive (TP) is a condition when an actual attack is successfully detected by the IDS. True Negative (TN) is a condition where normal traffic is detected as a normal, in other words there is no attack nor IDS alert is raised. False Positive (FP) is an alert that indicates that an attack is in progress when in fact there was no such attack. False Negative (FN) is a failure of IDS to detect an actual attack [25]. The accuracy rate and false positive rate are measured using the following formulae:

$$\text{AccuracyRate} = \frac{FP + FN}{TP + TN + FP + FN} \quad (1)$$

$$\text{False Positive} = \frac{FP}{TP + FP} \quad (2)$$

6. Experimental Results and Discussion

The following section discusses and analyses the results of the dimensionality reduction module, the misuse detection module and the anomaly detection module.

6.1 Dimensionality Reduction Module

The following sections describe the experimental results of various dimensionality reductions including PCA, ICA, GA and PSO.

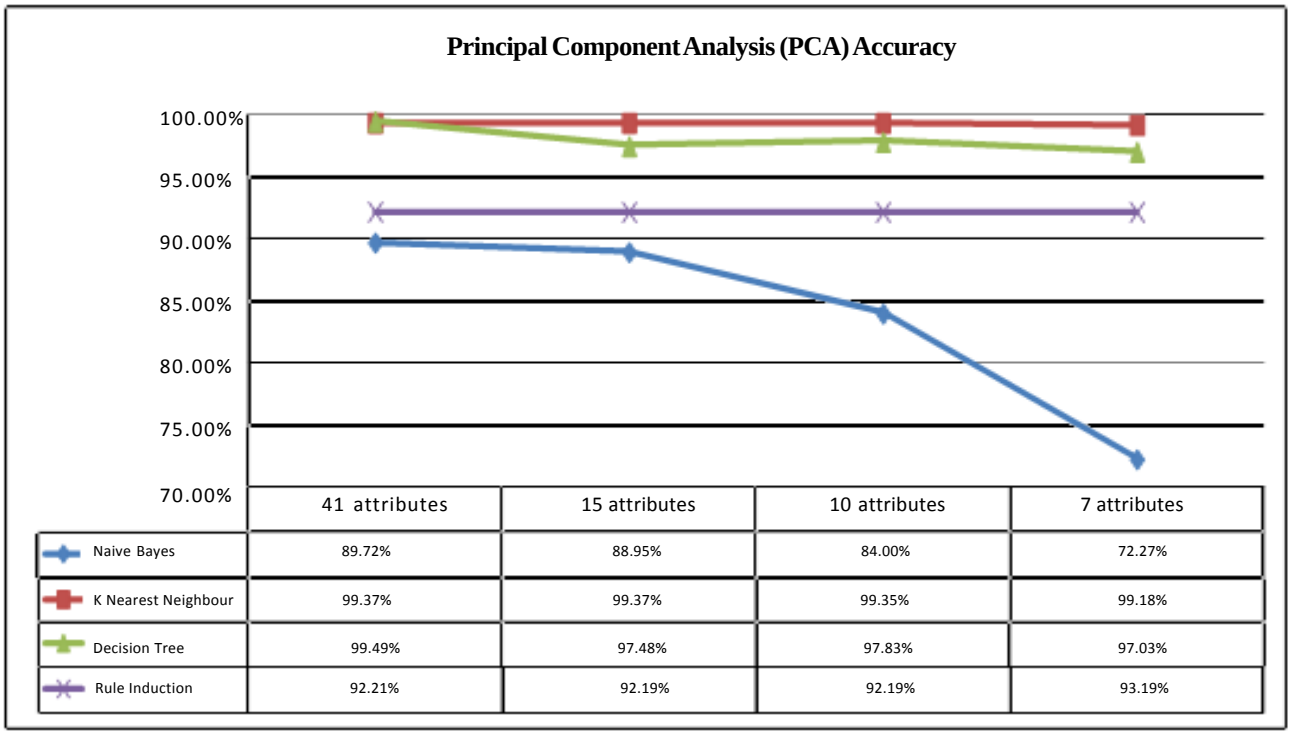
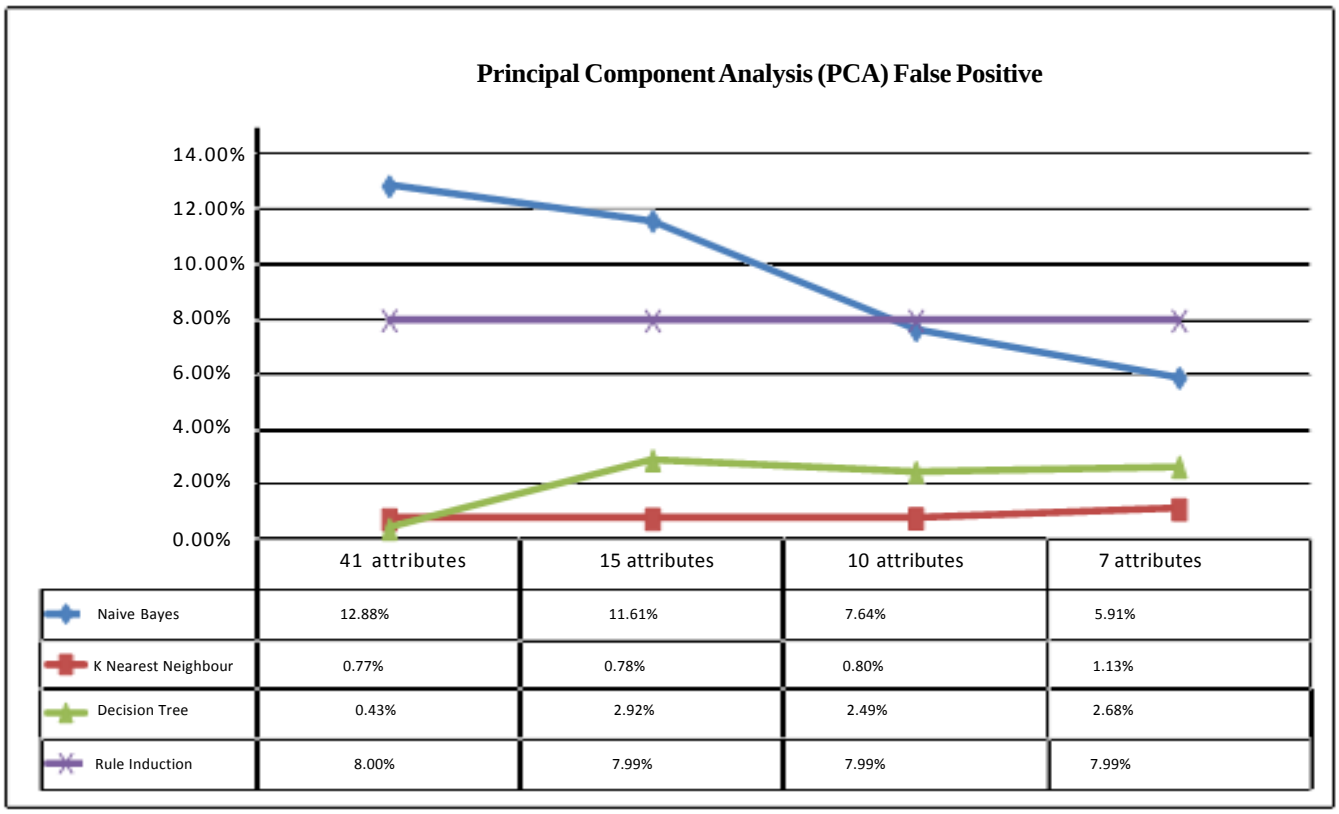


Figure 3. Principal Component Analysis (PCA) Accuracy

6.1.1 Feature Extraction Results

The intrusion dataset which has 41 attributes was reduced by PCA and ICA into a smaller dimension. In order to find the ideal number of attributes, we run several experiments with a different number of attributes which are 15, 10 and 7 attributes. We use RapidMiner Data Mining Tools[3] both for dimensionality reduction (PCA and ICA) and four classification algorithms and the results are shown in Figure 3 below.

Figure 4 shows that reducing the number of attributes from 41 attributes to 15 new attributes or 10 new attributes can still maintain the accuracy. Among 4 classification algorithms, only NB did not perform well while NN achieved the best accuracy and outperformed other algorithms.



	41 attributes	15 attributes	10 attributes	7 attributes
Naive Bayes	12.88%	11.61%	7.64%	5.91%
K Nearest Neighbour	0.77%	0.78%	0.80%	1.13%
Decision Tree	0.43%	2.92%	2.49%	2.68%
Rule Induction	8.00%	7.99%	7.99%	7.99%

Figure 4. Principal Component Analysis (PCA) False Positive

In Figure 4, we can see the use of PCA was able to reduce the false positive significantly on NB and maintains the same value of false positive on RI and NN even though the dimensionality was reduced from 41 into 15 and 10. Overall, the NN algorithms achieved the best performance both in accuracy (99.37%) and false positive (0.78%) compare to the three other algorithms.

In Figure 5, we can see that the accuracy of 4 algorithms was decreased significantly if the number attributes were 10 and 7. Once again, NN achieved the best accuracy compared to the others.

Figure 6 show that ICA outperformed PCA when implemented with NN as a classifier and the number of attributes was set to 15, it achieved better accuracy (99.57%) and produced a lower false positive (0.38%). The ideal number of attributes is 15 or 10 in PCA and 15 in ICA.

6.1.2 Feature Selection using GA and PSO

We applied GA and PSO feature selection algorithms provided by WEKA Data Mining Tools[4] to find the most important attributes in intrusion data set, then we applied these selected attributes into two different classifiers (NN and DT) provided by

[3]Rapid Miner Data Mining Tools http://rapid-i.com

[4]Weka Data Mining Tools http://www.cs.waikato.ac.nz/ml/weka/

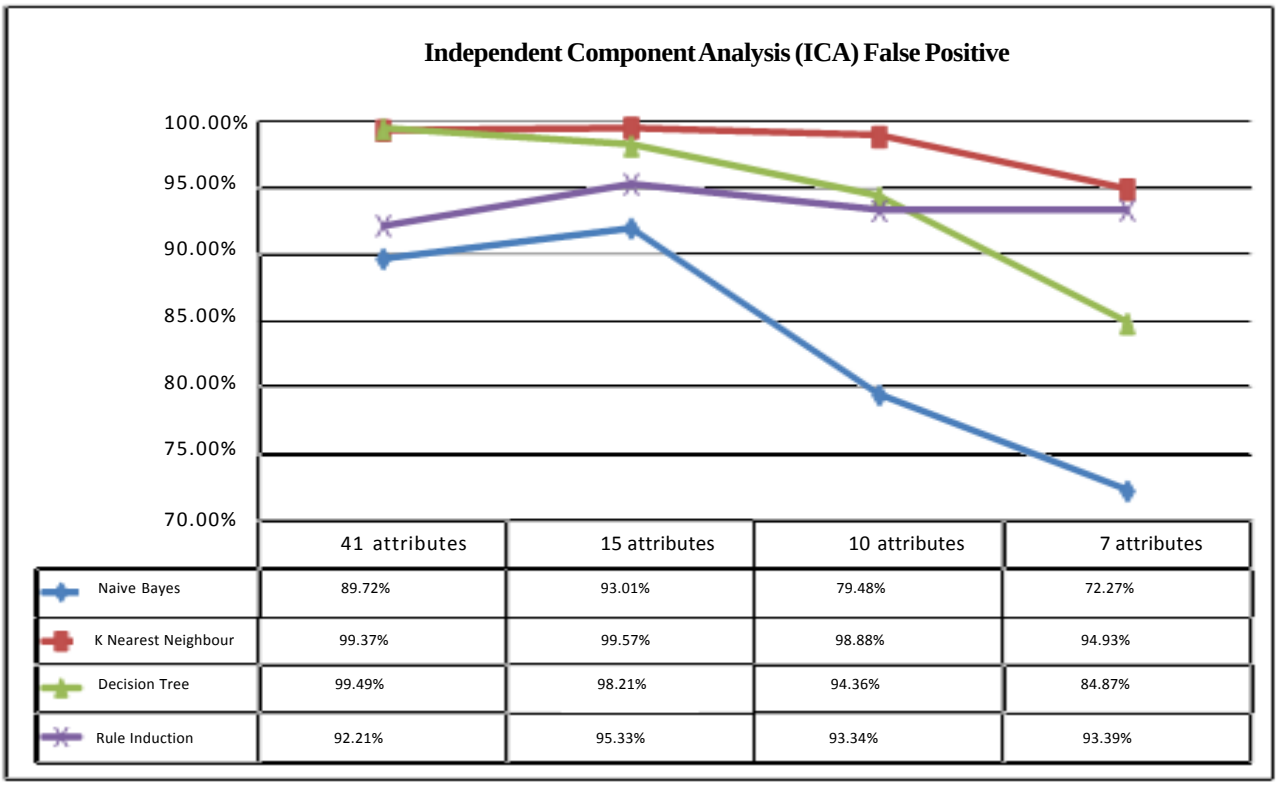


Figure 5. Independent Component Analysis (ICA) Accuracy

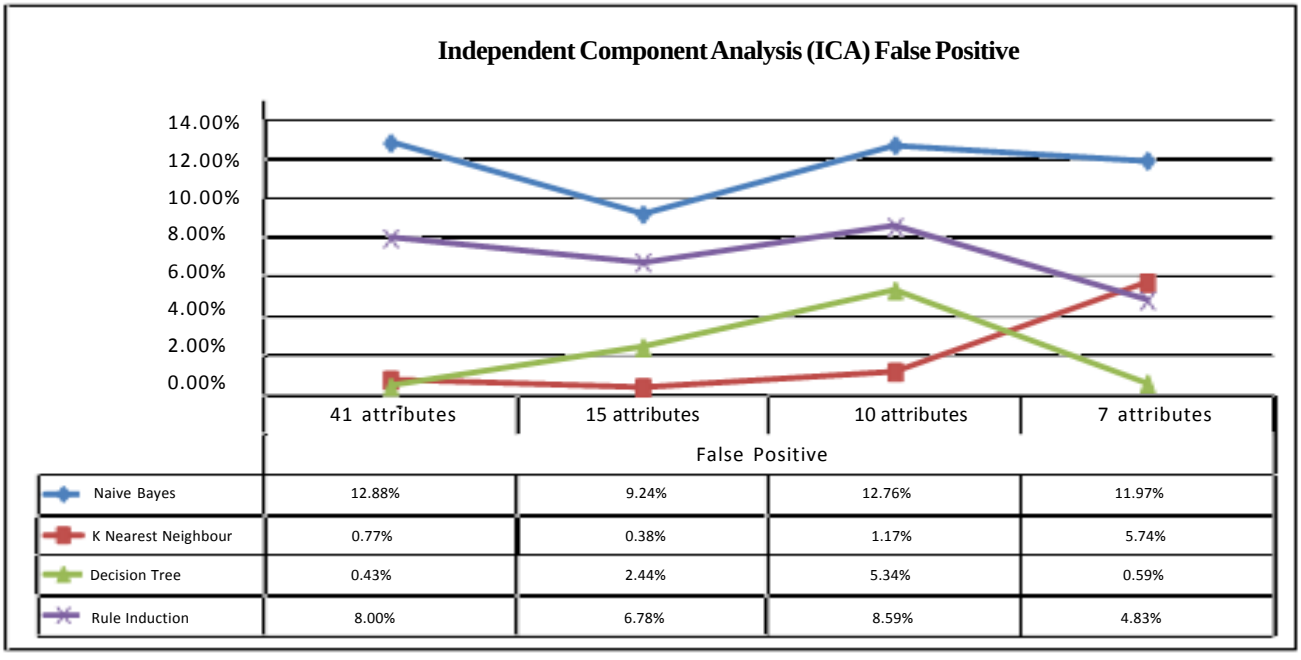


Figure 6. Independent Component Analysis (ICA) Accuracy

classifiers (NN and DT) provided by RapidMiner. From the 41 attributes of KDD Intrusion Dataset[1], GA has selected the 15 most important attributes and PSO selected the best 9 of them which are explained in the Table 3 below.

[5]Full Attributes of Intrusion Data http://kdd.ics.uci.edu/databases/kddcup99/task.html

Algorithms	The most important attributes	Nearest Neighbour		Decision Tree	
		Accuracy	FP	Accuracy	FP
Original Data Set	41 attributes	99.37%	0.77%	99.49%	0.43%
Genetic Algorithm (GA)	14 attributes service, flag, src_bytes, dst_bytes, logged_in, num_root, num_shel ls, serror_rate, srv_serror_rate, same_srv_rate, diff_srv_rate, srv_diff_host_rate, dst_host_same_src_port_rate, dst_host_srv_di ff_host_rate, dst_host_serror_rate	99.71%	0.27%	99.27%	0.64%
Particle Swarm Optimization (PSA)	9 attributes src_bytes, dst_bytes, serror_rate, srv_serror_rate, same_srv_rate, di ff_srv_rate, dst_host_srv_di ff_host_rate, dst_host_serror_rate, dst_host_srv_serror_rate	99.72%	0.25%	99.03%	0.88%

Table 3. Feature Selection using GA and PSO

Table 3 shows that the NN classifier when implemented with PSO as an attribute selection algorithm, achieved the best performance and outperformed other algorithms. Even with only 9 attributes, NN-PSO achieved 99.72% accuracy and 0.25% false positive. These results are much better than NN-GA, DT-GA or any classifiers which used PCA or ICA as a dimensionality reduction algorithm. This experiment proved that the use of dimensionality reduction algorithm is able to improve the accuracy, reduce the false positive and reduce the computation time.

6.2 Misuse Detection Module

In the previous section, we have applied four basic machine learning algorithms into misuse detection module using 10-fold cross validation. In this following section, we will test it using testing data which contains new intrusions. After that, we tried to improve the classification performance using ensemble approach.

Algorithm	Accuracy	False Positive
Naïve Bayes	55.77%	34.80%
Nearest Neighbour	62.84%	20.90%
Rule Induction	63.69%	18.00%
Decision Tree	63.97%	17.90%

Table 4. Misuse Detection performance using testing data

6.2.1 Basic Algorithm

In the experiments described in Section 6.1, misuse detection has achieved very good results when detecting known intrusion. Three of the four algorithms (NN, RI and DT) achieve more than 99% accuracy and the false positive rates are less than 1%. In

Algorithm	Accuracy			False Positive		
	Single	Bagging	Boosting	Single	Bagging	Boosting
Naïve Bayes	89.59%	89.57%	94.56%	10.60%	10.70%	5.30%
Nearest Neighbour (iBK)	99.44%	99.44%	99.44%	0.60%	0.60%	0.60%
Rule Induction (JRip)	**99.58%**	**99.71%**	99.73%	**0.40%**	**0%**	0.30%
Decision Tree (J48)	99.56%	99.67%	**99.80%**	**0.40%**	0.30%	**0.20%**

Table 5. The performance of ensemble classifiers using 10-fold cross validation

the second experiment, we now use a testing data to evaluate the performance of the intrusion model in the misuse detection module. The testing data contains 22 types of known intrusions and 18 types of unknown intrusions. The results of the second experiment are shown in Table 4 below. Table 4 shows that the misuse detection module does not perform well in detecting data which contains a large number of unknown intrusions where the highest accuracy is only 63.97% and the lowest false positive is 17.90%.

6.2.2 Ensemble Approach

In order to improve the performance of basic classifiers (NB, NN, RI and DT), we applied tree different ensemble classifiers:

Base Learner	Stacking Model Learner	Accuracy	False Positive
NB, NN, DT	Rule Induction (RI)	99.64%	0.40%
RI, NN, DT	Naive Bayes (NB)	**99.75%**	**0.30%**
RI, NB, DT	Nearest Neighbour (NN)	99.51%	0.50%
RI, NB, NN	Decision Tree (DT)	99.63%	0.40%

Table 6. The performance of stacking algorithm using 10-fold cross validation

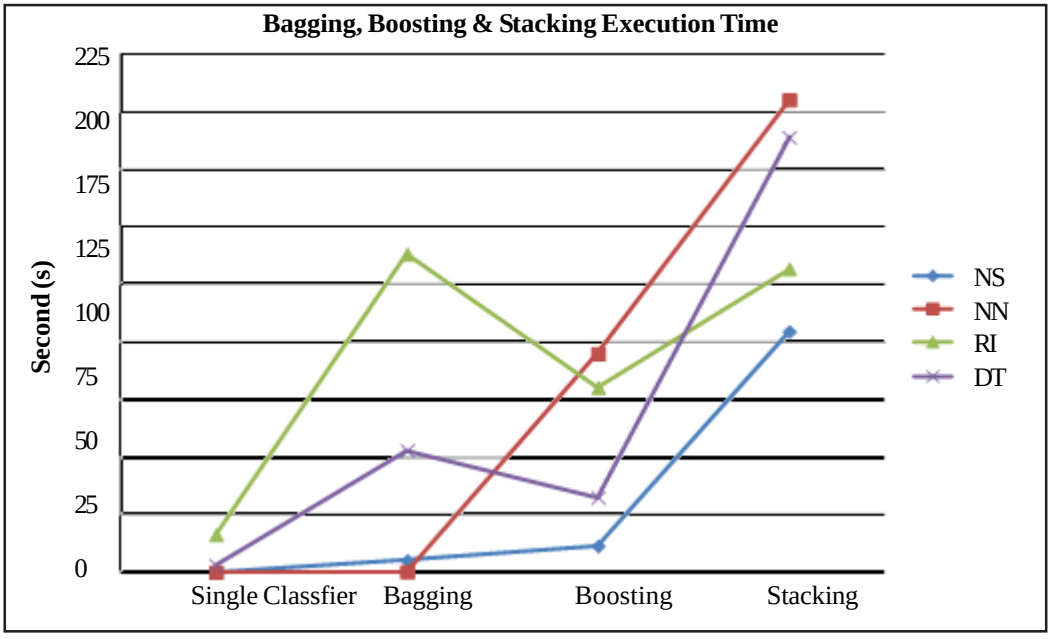


Figure 7. Execution time comparison for single classifier bagging, boosting and stacking

bagging, boosting and stacking. We applied Bagging and AdaBoost (Boosting) algorithms from WEKA and Stacking algorithm from RapidMiner. The bagging and boosting results are explained in Table 5 below.

In the stacking method, we use three different algorithms as base learners and an algorithm as a stacking model learner. We use various combinations of NB, NN, DT and RI. The classifications predicted by the base learners will be used as input variables into a stacking model learner. Each input classifier computes predicted classifications using cross validation from which an overall performance characteristic can be computed. Then the stacking model learner will attempt to learn from the data how to combine the predictions from the different models to achieve maximum classification accuracy. The stacking algorithm experiment results are given in Table 6.

Overall, all of the classification algorithms achieved good results, with the highest accuracy being 99.80% and the lowest being 89.59%. Tables 5 and 6 above show that Adaboost when implemented with DT as a weak classifier, achieves the highest accuracy, which is 99.80%, with a false positive (FP) rate of 0.30%. On the other hand, the RI Bagging algorithm achieves the lowest FP rate of 0%. Bagging was able to reduce the false positive rate by up to 25% when implemented with RI and DT, boosting by up to 50% for NB and DT, and stacking by up to 96.23% for NB.

Algorithm	Accuracy	False Positive
k-Means	57.81%	22.95%
improved k-Means	65.40%	21.52%
k-Medoids	76.71%	21.83%
EM clustering	78.06%	20.74%
Distance-based outlier detection	80.15%	21.14%

Table 7. Anomaly Detection accuracy using clustering algorithms

Figure 7 shows that the use of bagging, boosting and stacking significantly increases the execution time. The slowest is stacking followed in turn by bagging and boosting. The stacking method was able to reduce the false positive rate, but it would be too slow to implement in a misuse detection module. The bagging method, especially when applied to the NN and NB algorithms, did not increase the execution time significantly and only improves the accuracy by 0.18% (NN) and 0.59% (NB).

6.3 Anomaly Detection Module

We applied five unsupervised clustering algorithms which are k-Means, improved k-Means, k-Medoids, Expectation-Maximization (EM) clustering and distance-based outlier detection algorithm into the anomaly detection module and used an unlabelled dataset as an input and the results are shown in Table 7 below.

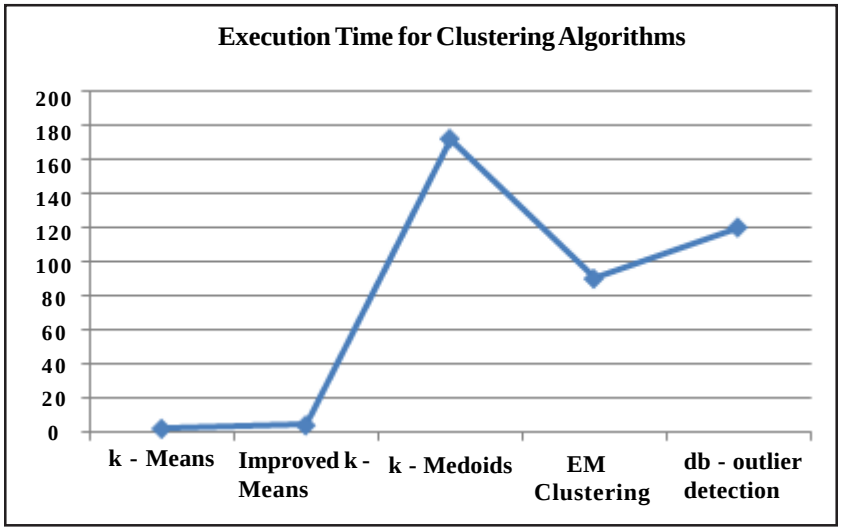


Figure 8. Clustering algorithms execution time

Compared to the misuse detection module (Table 4) which has an accuracy of only 63.97% (evaluated using testing data), the anomaly detection module has a better performance in detecting novel intrusion. These clustering algorithms are able to detect intrusions without prior knowledge. In this experiment, the distance-based outlier detection algorithm achieves the best accuracy with 80.15%, followed by EM clustering 78.06%, k-Medoids with 76.71%, improved k-Means 65.40% and k-Means 57.81%. Unfortunately, all of these algorithms have quite high positive rates with more than 20%. This means that there are around 20% of normal traffics predicted as intrusions.

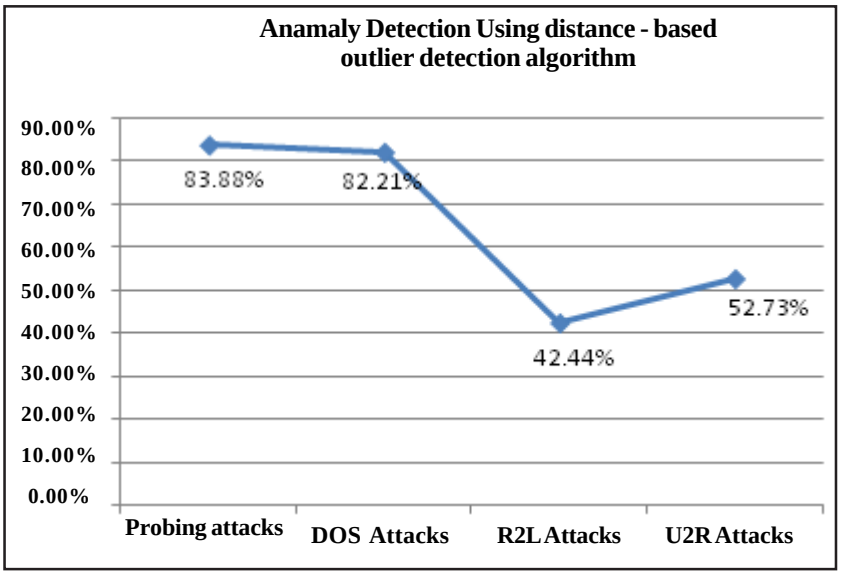


Figure 9. Anomaly detection using distance-based outlier detection algorithm

Even though the distance-based outlier detection algorithm outperforms the other four algorithms in accuracy, unfortunately its computation time is relatively high. The k-Means algorithm is the fastest but its accuracy is the worst (57.81%), in contrast the k-Medoids algorithm is the slowest but its accuracy is relatively high (76.71%).

Since the distance-based outlier detection algorithm has achieved the highest accuracy, we continue our experiment by applying this algorithm in the anomaly detection module. Now we classify the intrusion dataset into four types of intrusion which are probing attacks, DoS attacks, R2L attacks and U2R attacks. The results are shown in Figure 9 below.

This experiment shows that the distance-based outlier detection algorithm is able to detect probing attacks with 83.8% accuracy and DoS attacks with 82.21% accuracy. Unfortunately, this algorithm failed to accurately detect R2L attack (42.44%) and U2R attacks (52.73%). One reason is that the R2L attacks and U2R attacks have very similar behaviour with normal traffics which makes them very difficult to distinguish. Furthermore, the number of R2L and U2R attacks in intrusion dataset is very small compare to the whole data set. The number of R2L attacks is only 0.83% and U2R is only 0.04%.

7. Conclusions

Our experiment shows that the Nearest Neighbour (NN) classifier when implemented with Particle Swarm Optimization (PSO) as an attribute selection algorithm, achieved the best performance and outperformed other algorithms. Even with only 9 attributes, NN-PSO achieved 99.72% accuracy and 0.25% false positive. These results are better than NN-GA (99.71% accuracy and 0.27% false positive), Decision Tree - GA (99.27% accuracy and 0.64% false positive), NN-ICA (99.57% accuracy and 0.38 false positive) and NN-PCA (99.37% accuracy and 0.78% false positive). The NN classifier when applied into a full intrusion dataset with 41 attributes, achieved 99.37 % accuracy and 0.77% false positive. These results proved that the use of proper dimensionality reduction algorithm is able to improve the accuracy, reduce the false positive as well as reduce the computation time.

We investigated the possibility of using ensemble algorithms (bagging, boosting and stacking) to improve the performance on

misuse detection systems. Our experiment shows that Boosting when implement with Decision Tree as a weak classifier achieves the highest accuracy, which is 99.80% , with a false positive (FP) rate of 0.30%. On the other hand, the Rule Induction Bagging algorithm achieves the lowest FP rate of 0%. Stacking was able to reduce the false positive rate by a relatively high amount; unfortunately, this method has the longest execution time which is a serious disadvantage in the intrusion detection field.

The misuse detection technique achieves a very good performance with more than 99% accuracy when detecting known intrusion but it fails to accurately detect data set with a large number of unknown intrusions where the highest accuracy is only 63.97%. In contrast, the anomaly detection approach shows promising results where the distance-based outlier detection method outperforms the other three clustering algorithms with the accuracy of 80.15%, followed by EM clustering (78.06%), k-Medoids (76.71%), improved k-Means (65.40%) and k-Means (57.81%). Further experiment shows that the distance-based outlier detection performs very well in detecting probing attacks (83.88%) and DoS attacks (82.21%) but it fails to detect R2L attacks (42.44%) and U2R attacks (52.73%).

References

[1] Breiman, L., Breiman, L. (1996). Bagging Predictors. *Machine Learning*, 123–140.

[2] Chandola, V. et al. (2009). Anomaly detection: A survey. *ACM Comput. Surv.* 41 (3) 1–58.

[3] Chebrolu, S. et al. (2004). Hybrid Feature Selection for Modeling Intrusion Detection Systems. *LNCS 3316*, p. 1020–1025.

[4] Davis, J. J., Clark, A. J. (2011). Data preprocessing for anomaly based network intrusion detection: A review. *Computers & Security*. 30, 6–7, 353–375.

[5] Draper, B. A. et al. (2003). Recognizing faces with PCA and ICA. *Computer Vision and Image Understanding*. 91, 1–2 (Jul. 2003) 115–137.

[6] García-Teodoro, P. et al. (2009). Anomaly-based network intrusion detection: Techniques, systems and challenges. *Computers & Security*. 28, 1–2 , 18–28.

[7] Graczyk, M. et al. (2010). Comparison of Bagging, Boosting and Stacking Ensembles Applied to Real Estate Appraisal. *Intelligent Information and Database Systems*. N. Nguyen et al., eds. Springer Berlin / Heidelberg. p. 340–350.

[8] Gudadhe, M. et al. (2010). A new data mining based network Intrusion Detection model. *Computer and Communication Technology (ICCCT),International Conference on* 2010, p. 731–735.

[9] Kennedy, J., Eberhart, R. 1995. Particle swarm optimization. *Neural Networks, 1995. In*: *Proceedings., IEEE International Conference on* (Dec. 1995), 4, 1942 –1948.

[10] Knorr, E. M., Ng, R.T. (1999). Finding Intensional Knowledge of Distance-Based Outliers. *In*: *Proceedings of the 25th International Conference on Very Large Data Bases*, 211–222.

[11] Liu, Y. et al. (2006). An Improved Particle Swarm Optimization for Feature Selection. *Engineering*. 8 (2) 924–928.

[12] Lu, W., Tong, H. (2009). Detecting Network Anomalies Using CUSUM and EM Clustering. *In*: *Proceedings of the 4th International Symposium on Advances in Computation and Intelligence*, p. 297–308.

[13] Mukkamala, S. et al. (2005). Intrusion detection using an ensemble of intelligent paradigms. *J. Network and Computer Applications*. 28 (2) 167–182.

[14] Nguyen, H. et al. (2010). Improving Effectiveness of Intrusion Detection by Correlation Feature Selection. *Availability, Reliability, and Security, 2010. ARES '10 International Conference on* (Feb. 2010), p. 17 –24.

[15] Oh, I.-S. et al. 2004. Hybrid genetic algorithms for feature selection, *In*: *Transactions on Pattern Analysis and Machine Intelligence, IEEE* 26 (11) 1424-1437.

[16] Orair, G. H. et al. (2010). Distance-based outlier detection: consolidation and renewed bearing. *In*: *Proceedinds VLDB Endow.* 3, 1-2, 1469–1480.

[17] Panda, M. and Patra, M.R. 2009. Ensemble of classifiers for detecting network intrusion. *In*: Proceedings of the International Conference on Advances in Computing, Communication and Control (Mumbai, India, 2009) 510–515.

[18] Patcha, A. and Park, J.-M. (2007). An overview of anomaly detection techniques: Existing solutions and latest technological trends. *Computer Networks*. 51 (12) 3448–3470.

[19] Rasoulifard, A. et al. (2009). Incremental Hybrid Intrusion Detection Using Ensemble of Weak Classifiers. *Advances in Computer Science and Engineering*. H. Sarbazi-Azad et al., eds. Springer Berlin Heidelberg. p. 577–584.

[20] Schapire, R.E. et al. (1997). Boosting the Margin: A New Explanation for the Effectiveness of Voting Methods.

[21] Seetha, J. et al. 2012. Unsupervised Learning Algorithm for Color Texture Segmentation Based Multiscale Image Fusion. European Journal of Scientific Research. 67 (4) 506–511.

[22] Tavallaee, M. et al. (2009). A detailed analysis of the KDD CUP 99 data set. In: Proceedings of the Second IEEE international conference on Computational intelligence for security and defense applications, p. 53–58.

[23] Velmurugan, T., Santhanam, T. Computational Complexity between K-Means and K-Medoids Clustering Algorithms for Normal and Uniform Distributions of Data Points. *Journal of Computer Science*. 6 (3) 363–368.

[24] Wang and Battiti, R. (2006). Identifying intrusions in computer networks with principal component analysis. 3 (8).

[25] Whitman, M. E., Mattord, H. J. (2011). *Principles of Information Security*. Cengage Learning.

[26] Zhou, Z.-H. (2009). Ensemble Learning. *Encyclopedia of Biometrics*. Li, S. Z., Jain, A, eds. Springer US. 270–273