Ensemble based Classification Model for DDoS Detection using Machine Learning Classifiers

[1]Vinay Kumar, [2]Abhinav Bhandari
[1]M.Tech Student, Dept. of Computer Engineering, Punjabi University, Patiala
[2]Assistant Professor, Dept. of Computer Engineering, Punjabi University, Patiala

ABSTRACT—The Internet has grown over the past two decades, and billions of active users and devices has been added to the Internet to make the use of various applications and available data. The rise of Internet has also exaggerated the interest of hackers to disrupt the online services and data. The online services such as financial, defense, etc are considered the highly sensitive datasets, which are always under threat from the online attacks. Also, a number of online services such as social media, email servers, etc, are prone to several forms of attacks due to different reasons. The service disruption attacks are generally launched from a single attacker node or group of nodes, which is known as denial of service (DoS) and distributed denial of service (DDoS) respectively. The proposed model primarily focuses upon the DDoS attack detection, which is considered the most dangerous and effective attacks on resource availability. The performance of ensemble classification based proposed model is evaluated over the CAIDA dataset, which is collection of real-time DDoS attack data. This data is coupled with normal network traffic occurred during telnet and TCP connection. The ensemble classification, neural network and other machine learning models are deployed on the extracted features to detect DDoS attacks. The Random Forest is found most efficient classification model, whereas KNN & Naïve Bayes classifiers are the second best classifiers in different situations. However, the Multilayer Perceptron (MLP) classifier underperformed on the network data for DDoS detection.

KEYWORDS—DDoS Detection, Intrusion Detection, Supervised Classification, Ensemble Classification.

1. INTRODUCTION

There are several types of networks across the globe, which are used to connect the different types of nodes and offers variety of services. A big share of the networks across the globe is connected to the internet now-a-days, which is marking them prone and vulnerable to various types of attacks. The network attacks are launched for various purposes, which may include the denial of service, data or session hijacking, sniffing, etc. The intension behind the network attack may vary from causing the economic losses to the service provider, to taking over the target's computer systems to take control of it. A conclusive result of such intensions is always seen as a major hazard, which increases the important of security. The security measures primarily include the intrusion detection methods, which are either designed to detect the malicious codes or data flooding. The following section explains the important elements of the intrusion detection models:

Elements of Intrusion Detection: There are several intrusion detection models deployed in variety of networks, which utilizes the different mechanisms to detect the anomaly data. However, all of the intrusion detection techniques inherit some basic elements, which are described as following:

a) Data supply

The first part of associate degree intrusion detection system may be a knowledge supply. Knowledge supply is that from that network the traffic is coming back. It may be classified into four classes specifically Host-based monitors, Network-based monitors, Application-based monitors and Target-based monitors.

b) Analysis engine

This part takes info from the info supply and examines the info for symptoms or options of attacks or different abnormalities, violations. The analysis engine will analysis the misuse or signature based mostly detection that detects the legendary patterns. Different is anomaly based mostly detection that detects the unknown pattern.

c) Response manager

The response manager is activated once intrusion detection method is completed. This sends some signals within the type of alarm to the administrator that some intrusions area unit detected [4].

The ensemble classification model is deployed in the proposed model, which is the combination of multiple classification instances before taking the final classification decision. There are several ensemble classification algorithms such as voting, boosting and bootstrap aggregation (bagging) classification models. In this paper, the random forest classification algorithm is deployed for the purpose of classification, which is a bagging based ensemble classification algorithm. The bagging methods rely upon the decision based upon cloud of individual decisions taken by analyzing the small datasets against the testing signature. The testing signature is the testing data row, which is compared to the different training sets obtained randomly under each individual classification instance. The micro-instances of decision tree are created under the random forest classification, and the final decision is made after analyzing the result of all decision tree instances.

2. LITERATURE REVIEW

E.Kesavulu et al. defines an intrusion detection system is the process of monitoring and analysing the events occurring in a computer system in order to detect signs of security problems. The intrusion detection and other security technologies such as cryptography, authentication and firewalls have gained in importance in last ten years. Intrusion detection is an area growing in relevance as more and more sensitive data are stored and processed in networked systems. An intrusion detection system monitors devices and looks for anomalous or malicious behaviour in the pattern of activity in the audit stream [1].

Yao et al. describes the functional component and techniques of intrusion detection system. He said that typical intrusion detection system consists of three functional components: an information source, an analysis engine and a decision maker. The information source provides a stream of event records. This component can also be considered as an event generator. It monitors different data sources and generates data that are well formatted and suitable for analysis. The analysis engine finds signs of intrusions. In this paper the fuzzy intrusion detection system is explained and its techniques are examined. They said that fuzzy logic and support vector machine develops a dynamic decision boundary for intrusion detection [2].

Witcha et al. finds the abnormal behaviour using the fuzzy c mean and rough set. In fuzzy c mean the different degree of membership is defined. This approach not only detects the known intrusion as well as unknown intrusions. He performs the intrusion detection on KDD dataset. The experimental result shows in this are that rough set method is suitable and good for network security. By using fuzzy c mean we can differentiates the normal or abnormal data. This paper defines the anomaly detection and unsupervised clustering [26].

B.Benet al. differentiates the two basic approaches used for intrusions detection. The first approach is called misuse detection. A system using this approach detects intrusion events which follow well-known patterns. The patterns may describe a suspect set of sequences of actions or takes other forms. The primary limitation of this approach is that it cannot detect novel intrusions, i.e., events that are never happened in the system. The second approach is called anomaly detection. An anomaly detection based system analyses the event data of a training system, recognizes patterns of activities that appear to be normal [14].

Wei Li et. al. states that intrusion detection system may be divided into 2 teams betting on wherever they give the impression of being for intrusive behaviour :Network-based intrusion detection system and Host-based intrusion detection system. The network primarily based} intrusion detection detects the intrusions from the network and host based detects the intrusions of 1 host. This paper describes the network primarily based intrusion detection exploitation the genetic algorithmic rule. Genetic algorithmic rule generates a rule exploitation the choice, crossover and mutation operators. He takes four parameters for intrusion detection like mutation rate, crossover rate, population size and range of generations [3].

Liao et al. describes the soft computing technique i.e fuzzy logic which was introduced by Zadeh. Fuzzy logic is a powerful technique for dealing with human reasoning and decision making process. A fuzzy expert system uses fuzzy logic instead of Boolean logic. A fuzzy expert system is a collection of membership functions and rules that are used to reason about data. They define the steps of fuzzy logic which are used for intrusion detection: the traffic capture, the feature extractor, the fuzzification, the fuzzy inference engine, the knowledge base, the defuzzification, and the forensic analyzer. The traffic capture component is responsible for network traffic capture and preparation for traffic analysis. Then feature extractor performs extracting features on the ‘‘network traffic” captured by the traffic capture component. Feature extraction and selection from the available data are important to the effectiveness of the methods employed. Under the network and system environment, there are many traffic features that can be used for attack detection and analysis. Then in fuzzification each input variable’s sharp (crisp) value needs to be first fuzzified into linguistic values before the fuzzy decision processes with the rule base [4].

3. EXPERIEMTNAL DESIGN

The proposed DDoS Attack Detection (DDoS-AD) model has been designed for the detection of the network anomalies and assigns the access on the basis of the security measure of the target network nodes to the other network resources. The simulation of the DDoS attack affected network will produce a specific dataset containing several parameters, which includes current throughput, overall throughput, total anomaly count, detection time, etc. In this paper, the SVM, KNN, Random Forest and other classifiers will be applied to the dataset in order to obtain the results.

1. A document or an individual entity is broken on the basis of quantitative and qualitative variables.
2. The quantitative and qualitative variables are handled differently in order to create the balanced version of these variables.
3. Finally, both of the feature matrixes, both obtained from quantitative and qualitative, are combined together to create the feature matrix.

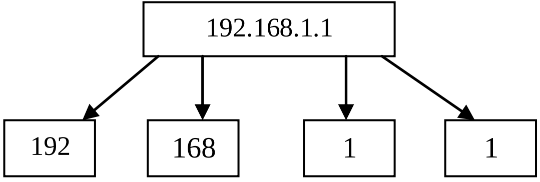
Afterwards, the data is divided into training and testing dataset, which is done using the random selection by creating the random number series. The cross validation split works on the 10-Fold, 5-Fold and 4-Fold split ratio configurations, which divided all the samples into training and testing subsets. Anomaly classification for network security system involves various steps like feature extraction, feature selection, feature vector generation, training and classification. The flow chart in figure 1 demonstrates all the phases system goes through.

Feature Engineering: The Internet protocol (IP) version 4 is being used for the purpose of tracking the attackers and targets, which are defined using the standard IPv4 format, which consists of four octets divided by a dot. The standard IPv4 is defined as the following:

IPv4 example	192.168.1.1

In the above example, the four octet based IPv4, which consists of total 32-bit (pr 4 bytes) address on total, which is divided in four octets 1 byte each. Each one byte of the octet carries are values ranging between 0 and 255, which can also defined between the range of 2^0 and 2^7.

The IP address is divided into the four numeric values, which are defined by each of the octet individually. This IP division policy in the proposed feature engineering model can be defined with the following example:



The above four separated numerical values describe the different octet values described in the form of independent numerical values, which can undergo any numerical methodology. The primary number handling method is known as the scaling, which generally shifts the values between 0 and 1, or -1 and 1, depending upon the application using the data.

In this data, the MaxMin scaling is being used to transform the value sequence produced from the IPv4 values, which is converted in the value between 0 and 1. The proposed model uses the similar technique for the source and destination node addresses.

Categorical (Qualitative) Variable Handling (CVH): The categorical or qualitative variables contain the categories or classes, which can't be defined numerically. Hence the categorical variables are handled by using the dummy variable mechanism, which is considered as the binary representation of the different categories after converting the string formatted categories to numerical labels. The following sequence is performed to process the categorical variables:

1. At first step, the example values of the categorical values are shown in the following table:

Protocol	**Class**
ICMP	0
TCP	1
-NA-	0
ICMP	1

2. In the above example, the two column table is taken into account with protocol and class features, out of which the protocol is considered as the categorical variables. There are two primary categories in the protocol column, which lists ICMP and TCP. Also, one missing values is depicted in the table, which must be filled before beginning with classification.

Protocol	**Missing Value Makrer**
ICMP	0
TCP	0
-NA-	**1**
ICMP	0

3. On the second step, the missing values are marked in the variable (or column), and the missing value filling is applied in the next step. In the above example, the missing value is marked with 1, which is detected in an iterative method through all of the variable values.

Protocol
ICMP
TCP
OTHER
ICMP

4. In the next step, the marked missing values are filled with provided option, which might be relevance based or static. In the proposed model the missing categorical values is replaced with another category titled "OTHER", because relevance is not determinant in the case of network packets. The above example shows the result after replacing the missing values.

Protocol	**Unique values (Numerical Labels)**	**Numeric Matrix**
ICMP	ICMP (1)	1
TCP	TCP (2)	2
OTHER	**OTHER (3)**	**3**
ICMP		1

5. After that, the unique values are extracted from the categorical variable, and a numerical label is assigned to each unique option. Finally, all of the values in the variable are replaced with respective numerical labels, which can be clearly seen in the third column of above example. The numerical labeling is necessary because almost the classification algorithms consider the numerical data for classification decisions.

Numerical (Quantitative) Variable Handling (NVH):A typical numerical variable contains a list of numerical values, which can possibly contain any value. The missing values can be present in any column of the classification data, which must be filled in order to execute the classification algorithm. The missing values generally cause the execution errors, which must be avoided for the implementation of a successful data model. The following example is taken from the

Length	Class
60	0
68	1
93	0
-NA-	1

1. In the above example, the packet length is taken as the numerical variables, which may contain roughly any value. In this exemplar, the length of 60, 68 and 93 are described, and one missing values on the end of variable.

Length	Missing Value Marker
60	0
68	0
93	0
-NA-	1

2. The values of the variable are marked with parallel binary vector, which describes presence of missing values in the target variable. The parallel vector in the above example marks the last value as the missing values.

Length	Mask
60	0
68	0
93	0
0	1

3. Then, an average must be computed over the variable. The missing value must be converted to zero initially, because a missing value can't take part in the calculation of average. Hence, the value in above exemplar is temporarily replaced by zero.

Length	Mask
60	0
68	0
93	0
55.25	1

4. Then, the average value is computed, which results a value of 55.25, which must be assigned to the actual missing value holder replacing zero, as its shown in the above example.

Length (A)	Min. value (B)	Length (A-B)
60	55.25	4.75
68	55.25	12.75
93	55.25	37.75
55.25	55.25	0

5. Afterwards, the find the lowest value in the matrix and subtract the minimum value from all of the values in the given quantitative variable. The above example shows the calculation of minimum value, and subtraction of the missing value from all data points.

Length (A)	Max. value (B)	Length (A-B)
4.75	37.75	0.1258
12.75	37.75	0.3377
37.75	37.75	1.0000
0	37.75	0

6. In the above example, the maximum value is used to divide the numerical values in the next step. The maximum value of 37.75 is marked from the variable (Length), which is further used to divide the data points by maximum values, and the final values are calculated in third column, which is known as scaled variable.

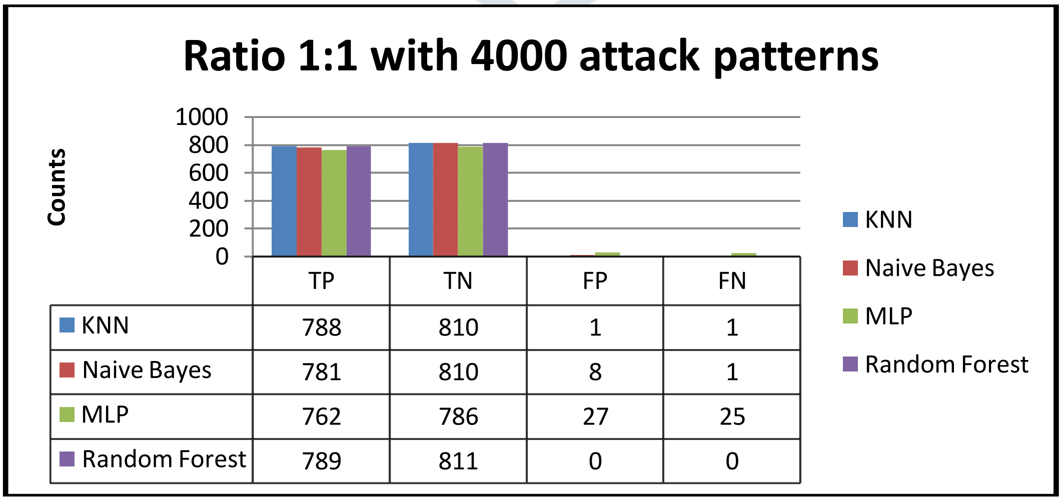
Classification Modeling: The classification modeling is defined in the following algorithm, which includes several steps. The complete classification procedure describes the training and testing procedures in detail on the given data to detect the DDoS attacks. At first, the network data is acquired from the data source, and perform the feature extraction by using the quantitative and qualitative processing methods as described in this section earlier. The feature matrix is prepared and divided in two parts, training & testing. The training data is used to train the algorithm and test data is used to determine the performance of the classification data model for detection DDoS attack.

Algorithm 1: DDoS Attack Detection Algorithm (DDoS-ADA)

```
1.  Perform the data acquisition from the local data source on fixed intervals
2.  Mark the categorical variables in the acquired dataset, and perform the following step on each variable individually
    1. Extract the categorical variable from the dataset one by one
    2. Search the variable for missing values, and mark the decision with missing values marker (0 for non-existent, 1 for existent)
    3. Fill the missing values with the target value
    4. Return the categorical variable
3.  Mark the numerical variables in the acquired dataset, and perform the following step on each variable individually:
    1. Extract the numerical variable from the dataset one by one
    2. Search the variable for missing values, and mark the decision with missing values marker (0 for non-existent, 1 for existent)
    3. Calculate the average of all of the variables individually
    4. Fill the missing values with respective average value of each of the column
    5. Return the numerical variable
4.  Combine all numerical and categorical variables to create the classification dataset
5.  Split the data into two subsets on specific ratio (Eg 10%) to create the training and testing subsets, and respective label vectors (i.e. training & testing label vectors)
6.  Initialize the classification algorithm with environmental parameters
7.  Train the classification algorithm with training data subset and training labels
8.  Test the classification algorithm with testing data subset
9.  Return the decision vector (predictions) by the classification model
10. Compare the classification predictions with the original testset labels
11. Compute the performance of the classifier with statistical parameters
12. Return the performance evaluation report
```

RESULT ANALYSIS

In this section the scenarios are planned with 4000 attack patterns per class with different split ratio in order to test the classification performance using the cross validation method. The different data split ratios are used to create the subset from the target dataset, which involves the 1:1, 1:2 and 1:4 ratio based subsets. The following tables shows the results obtained from the target datasets for different cross validation split ratios, which involves 10-fold, 5-fold and 4-fold testing subsets.



	TP	TN	FP	FN
🟦 KNN	788	810	1	1
🟥 Naive Bayes	781	810	8	1
🟩 MLP	762	786	27	25
🟪 Random Forest	789	811	0	0

Figure 4.1 (a): Result analysis of ratio 1:1 with 10-Fold Cross validation with 4000 attack patterns per class with statistical type 1 and 2 errors

Out of total 16000 rows of the input data, the 1600 rows are extracted for the testing set, and rest remain in the training set to test the classifier performance. The Random Forest is recorded with the highest accuracy of 100%, whereas the rise in the number of attack samples has devastated the performance, as it has been recorded with 27 false positive and 25 false negative cases. KNN and Naive Bayes has been also performed lower comparatively to the similar case of 2000 attack patterns. KNN and Naive are recorded 1 false negative each, and 1 & 8 false positive cases.

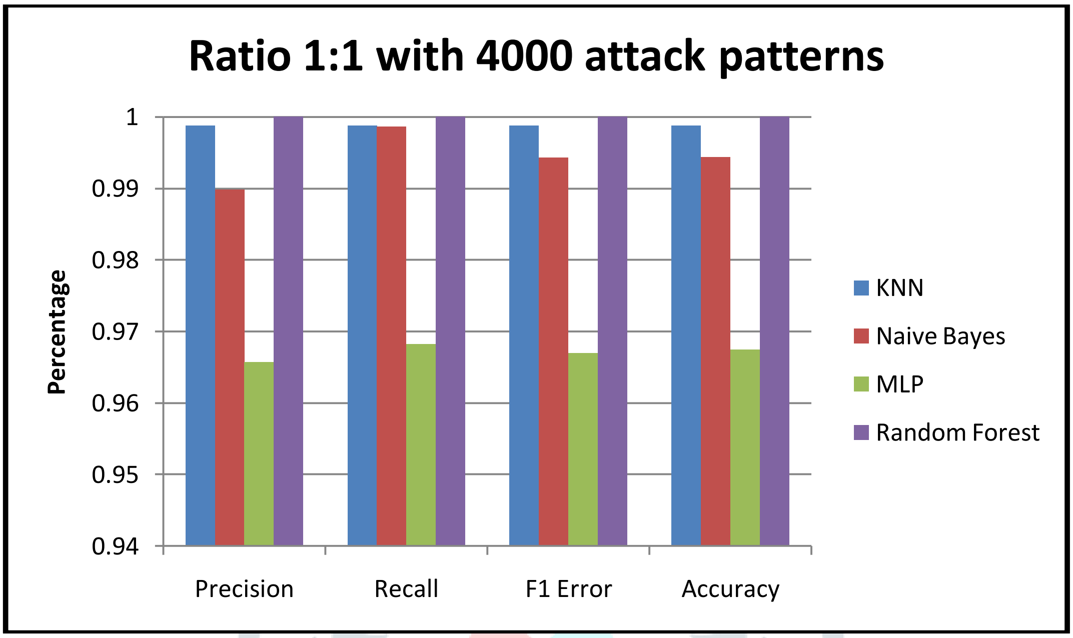
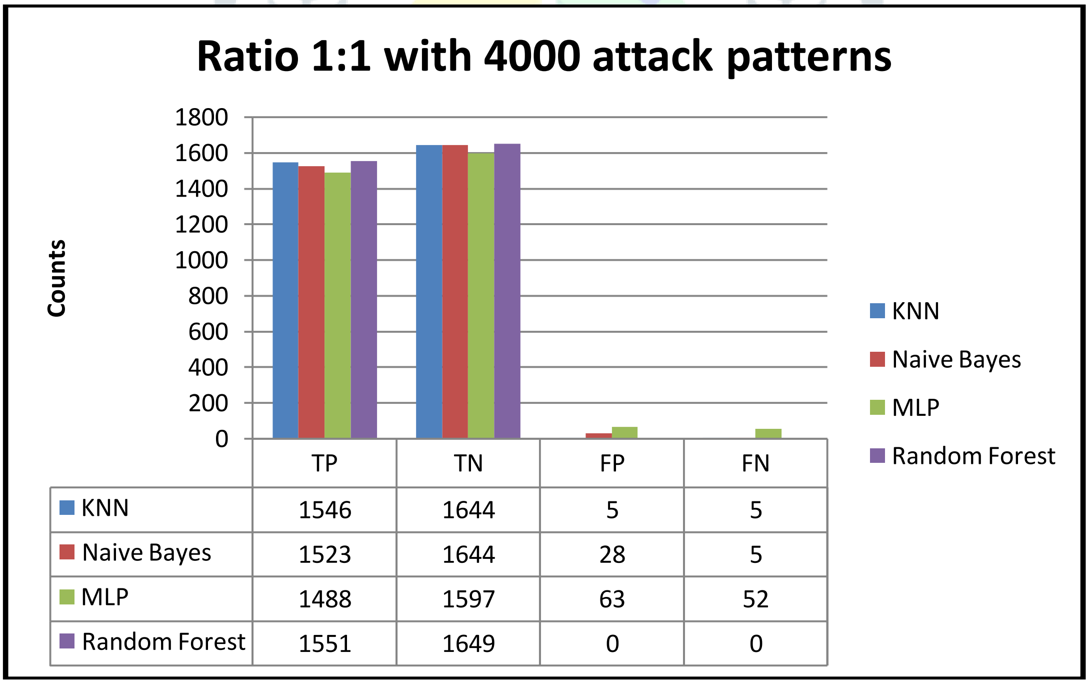


Figure 4.1 (b): Result analysis of ratio 1:1 with 10-Fold Cross validation with 4000 attack patterns per class with statistical accuracy based parameters



	TP	TN	FP	FN
KNN	1546	1644	5	5
Naive Bayes	1523	1644	28	5
MLP	1488	1597	63	52
Random Forest	1551	1649	0	0

Figure 4.2 (a): Result analysis of ratio 1:1 with 5-Fold Cross validation with 4000 attack patterns per class with statistical type 1 and 2 errors

Out of total 16000 rows of the input data, the 3200 rows are extracted for the testing set, and rest remain in the training set to test the classifier performance. Again, the Random Forest is recorded with the highest performance of 100%, whereas performance of MLP is decreased further (63 FP and 52 FN). KNN and Naive Bayes are recorded with 5 false negatives each, and 5 & 28 false positive cases respectively.

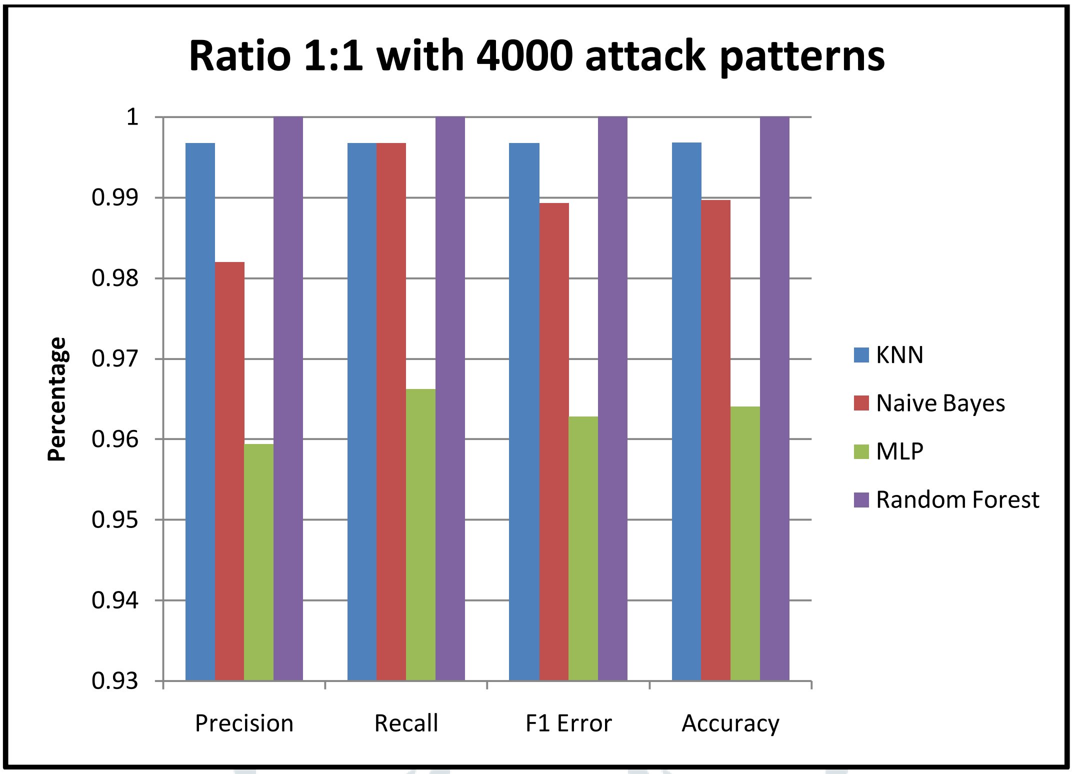
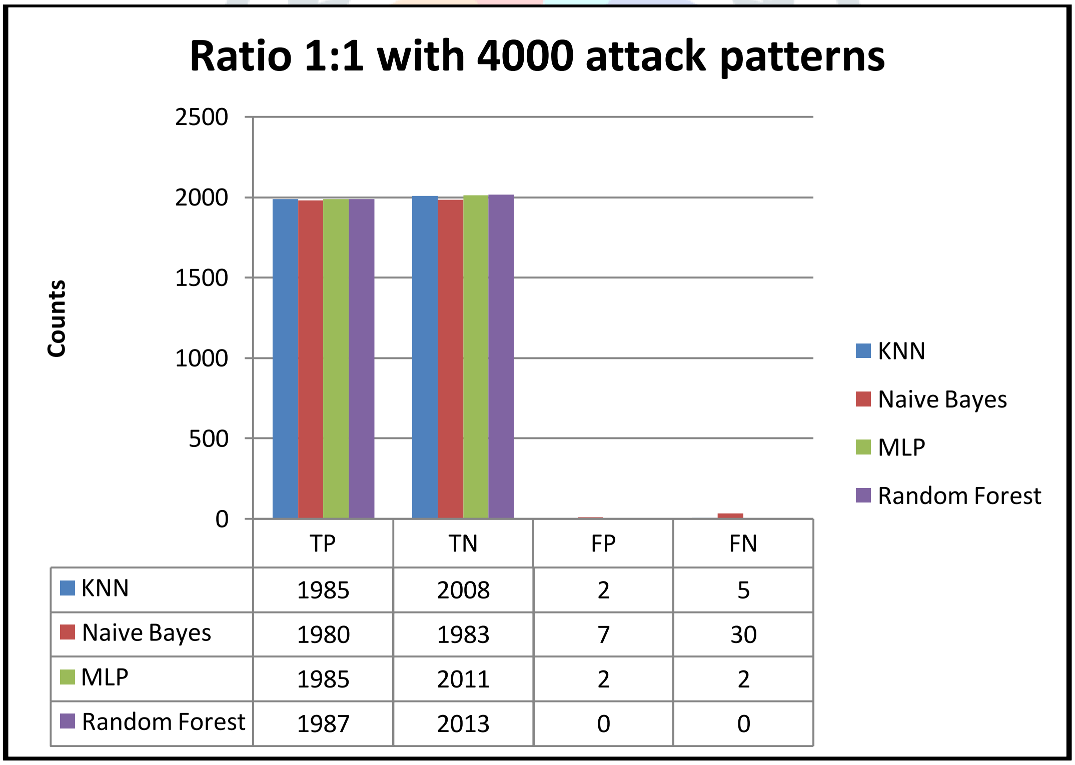


Figure 4.2 (b): Result analysis of ratio 1:1 with 5-Fold Cross validation with 4000 attack patterns per class with statistical accuracy based parameters



	TP	TN	FP	FN
🟦 KNN	1985	2008	2	5
🟥 Naive Bayes	1980	1983	7	30
🟩 MLP	1985	2011	2	2
🟪 Random Forest	1987	2013	0	0

Figure 4.3 (a): Result analysis of ratio 1:1 with 4-Fold Cross validation with 4000 attack patterns per class with statistical type 1 and 2 errors

Out of total 16000 rows of the input data, the 4000 rows are extracted for the testing set, and rest remain in the training set to test the classifier performance. Again, the Random Forest is recorded with the highest performance of 100%, whereas performance of MLP is improved further (2 FP and 2 FN). KNN and Naive Bayes are recorded with 5 and 30 false negatives respectively, and 2 & 7 false positive cases respectively. In this case, all of the performance indicators for MLP have boosted.

In contrasting factor, the Naive Bayes has shown the alternative degradation of the performance in comparison with previous case (5-fold), where the false positive cases are reduced from 28 to 7, and false negatives increased from 5 to 30. This shows an interesting transformation, which occurred due to the problems related to selection of error prone patterns for different classes. MLP has also shown an interesting transformation, where the results are strongly improved. MLP results are changed from false positives from 63 to 2 cases, and false negatives from 52 to 2 cases. This interesting change is possible due to the sample optimizations only. The training and testing samples are when extracted in highly uneven patterns, MLP's performance varies, because the difference of entropy rises due to small variations in the data. The MLP was constantly better for the 2000 attack pattern based dataset in previous sections, because the dataset was compact in size, and MLP (cross entropy) performs better with the compact samples, which is again the reason in this case in the performance boosting of the classifier.

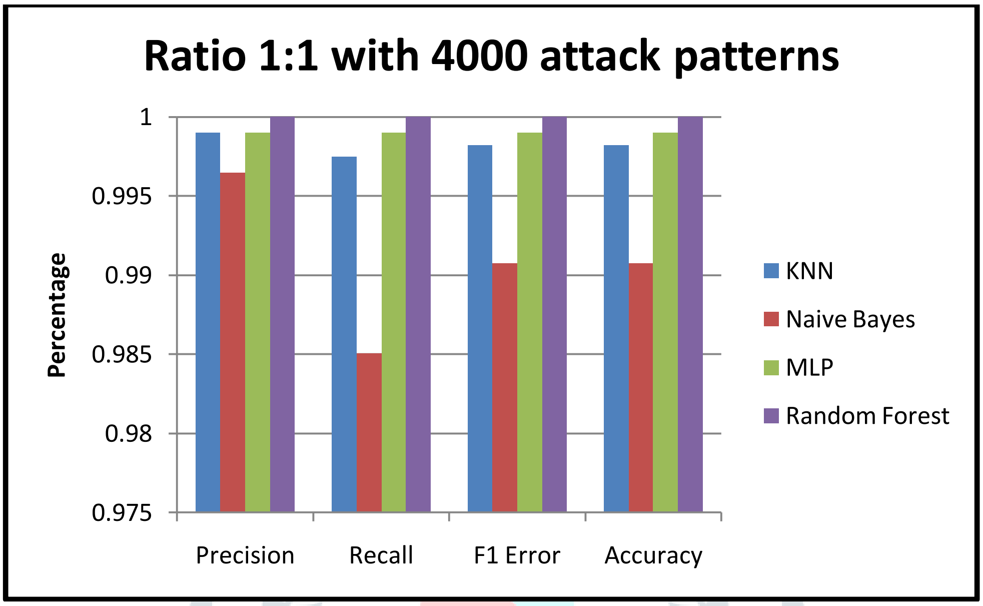
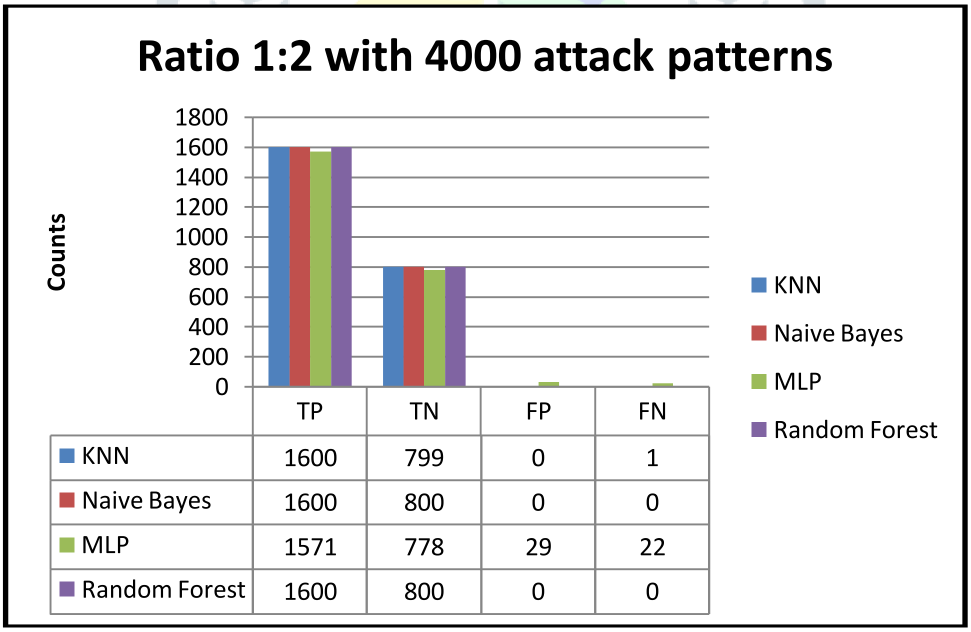


Figure 4.3 (b): Result analysis of ratio 1:1 with 4-Fold Cross validation with 4000 attack patterns per class with statistical accuracy based parameters



	TP	TN	FP	FN
■ KNN	1600	799	0	1
■ Naive Bayes	1600	800	0	0
■ MLP	1571	778	29	22
■ Random Forest	1600	800	0	0

Figure 4.4 (a): Result analysis of ratio 1:2 with 10-Fold Cross validation with 4000 attack patterns per class with statistical type 1 and 2 errors

Out of total 24000 rows of the input data, the 2400 rows are extracted for the testing set, and rest remain in the training set to test the classifier performance. The Naive Bayes and Random Forest are recorded with the highest performance of 100%. The MLP performance is again decreased from the previous case, as the deviation of testing samples decreased due to the reduction in sample count. KNN's performance in marginally decreased and it has been recorded with only 1 false negatives. MLP's performance shows only 29 false positive and 22 false negative cases.

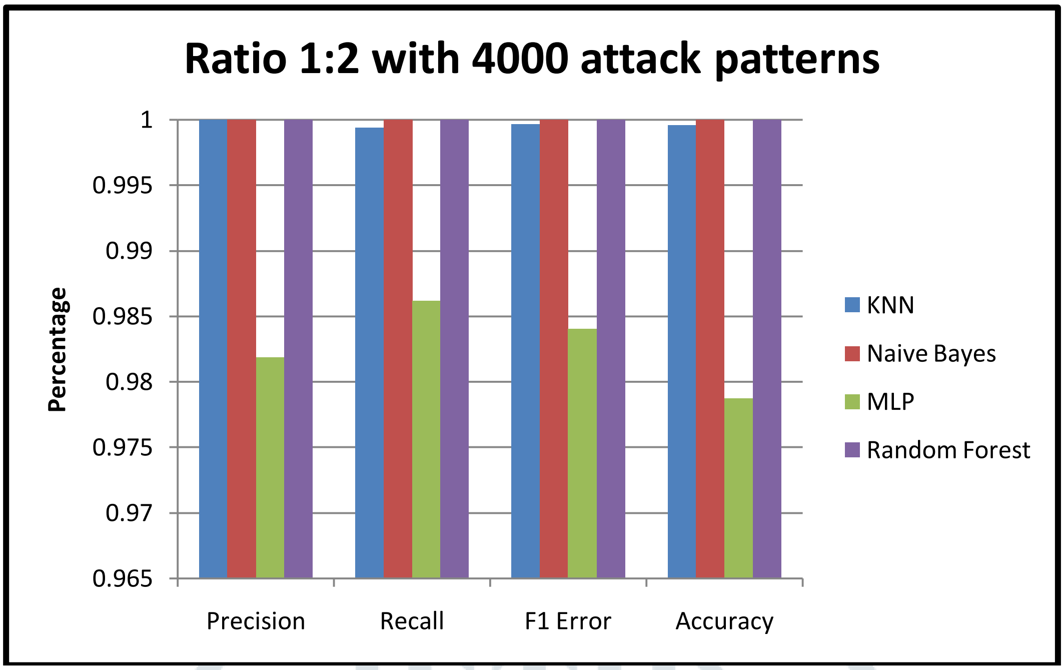
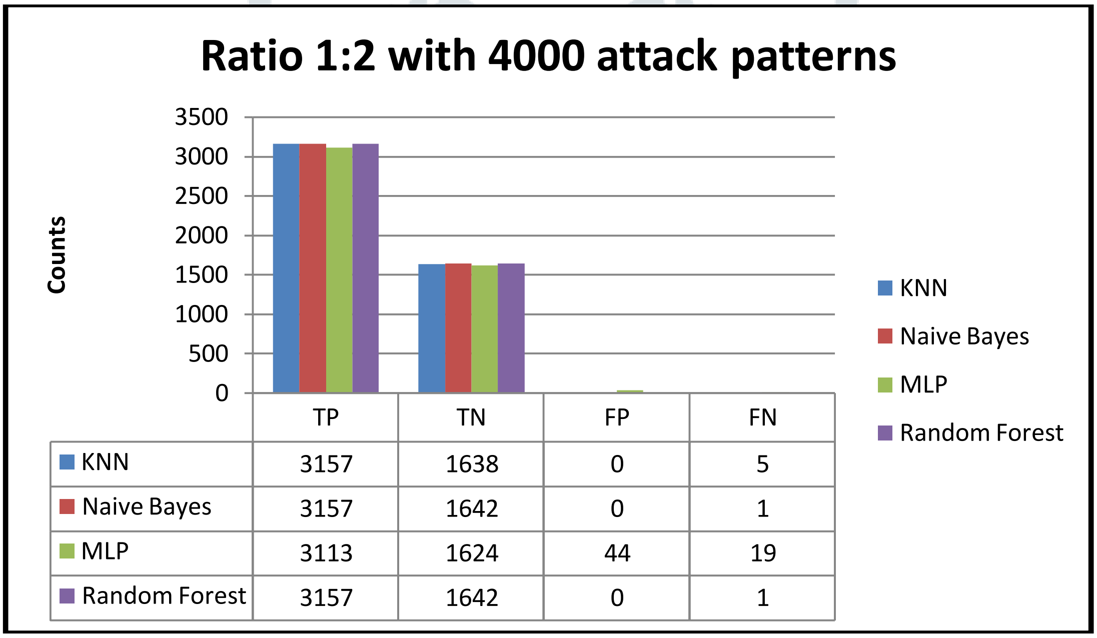


Figure 4.4 (b): Result analysis of ratio 1:2 with 10-Fold Cross validation with 4000 attack patterns per class with statistical accuracy based parameters



	TP	TN	FP	FN
KNN	3157	1638	0	5
Naive Bayes	3157	1642	0	1
MLP	3113	1624	44	19
Random Forest	3157	1642	0	1

Figure 4.5 (a): Result analysis of ratio 1:2 with 5-Fold Cross validation with 4000 attack patterns per class with statistical type 1 and 2 errors

Out of total 24000 rows of the input data, the 4800 rows are extracted for the testing set, and rest remain in the training set to test the classifier performance. For the first time in all of the performance evaluation testing, the performance of Random Forest has been marginally decreased from its over-fitting behaviour with only 1 false negative cases. The decrease in the performance of Random Forest is caused due to the randomization of the sample selection. MLP is performing the similar to the previous case, and recorded with higher false negative and false positive cases. Only false negative cases are detected for KNN, Naive Bayes and Random Forest, and false negative cases are recorded at 5, 1 and 1 respectively.

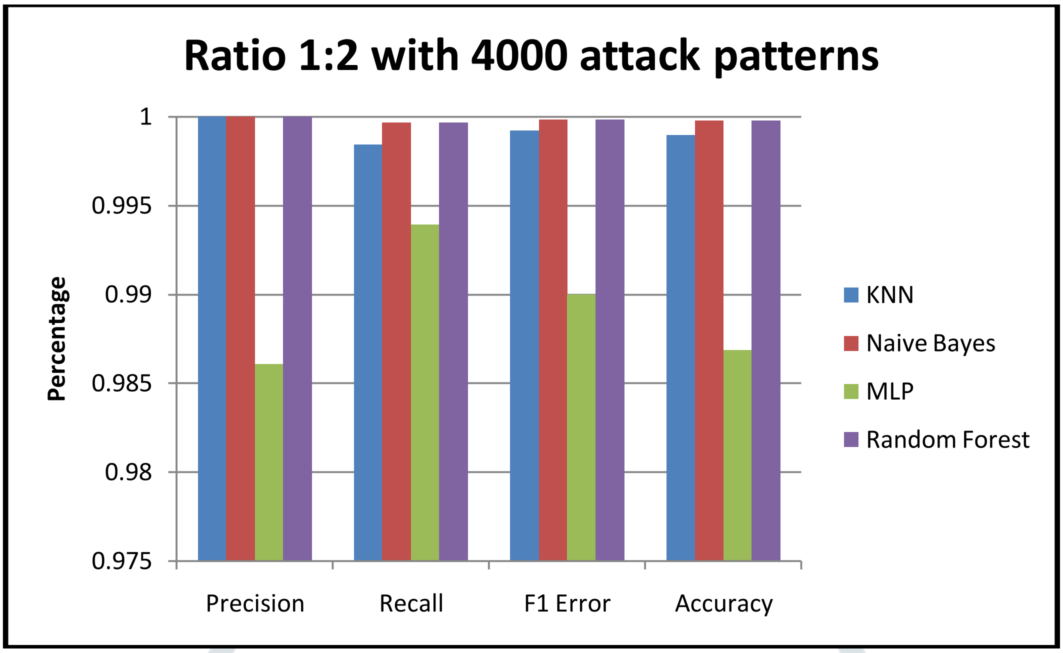
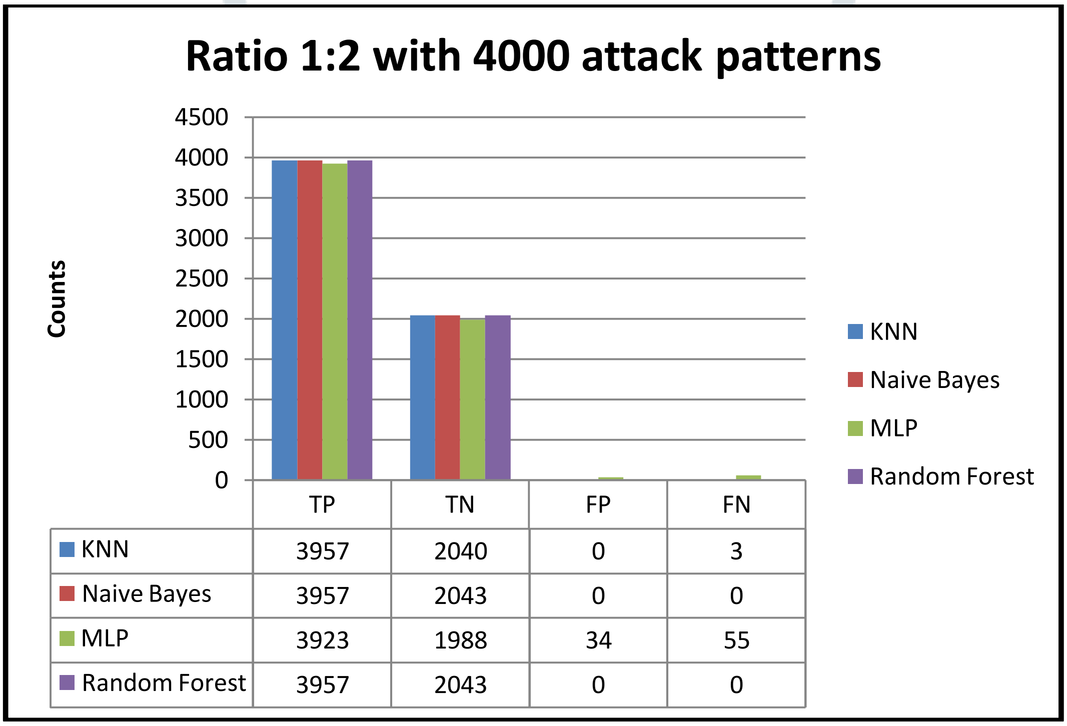


Figure 4.5 (b): Result analysis of ratio 1:2 with 5-Fold Cross validation with 4000 attack patterns per class with statistical accuracy based parameters



	TP	TN	FP	FN
KNN	3957	2040	0	3
Naive Bayes	3957	2043	0	0
MLP	3923	1988	34	55
Random Forest	3957	2043	0	0

Figure 4.6 (a): Result analysis of ratio 1:2 with 4-Fold Cross validation with 4000 attack patterns per class with statistical type 1 and 2 errors

Out of total 24000 rows of the input data, the 6000 rows are extracted for the testing set, and rest remain in the training set to test the classifier performance. The Random Forest has gained the performance again in comparison with the previous case, which is again caused by randomized sample selection. Both Naive Bayes and Random Forest are detected with 100% accuracy, whereas the performance of KNN is decreased marginally (3 FN). MLP's performance is being tumbled down with 34 FP, and 35 FP.

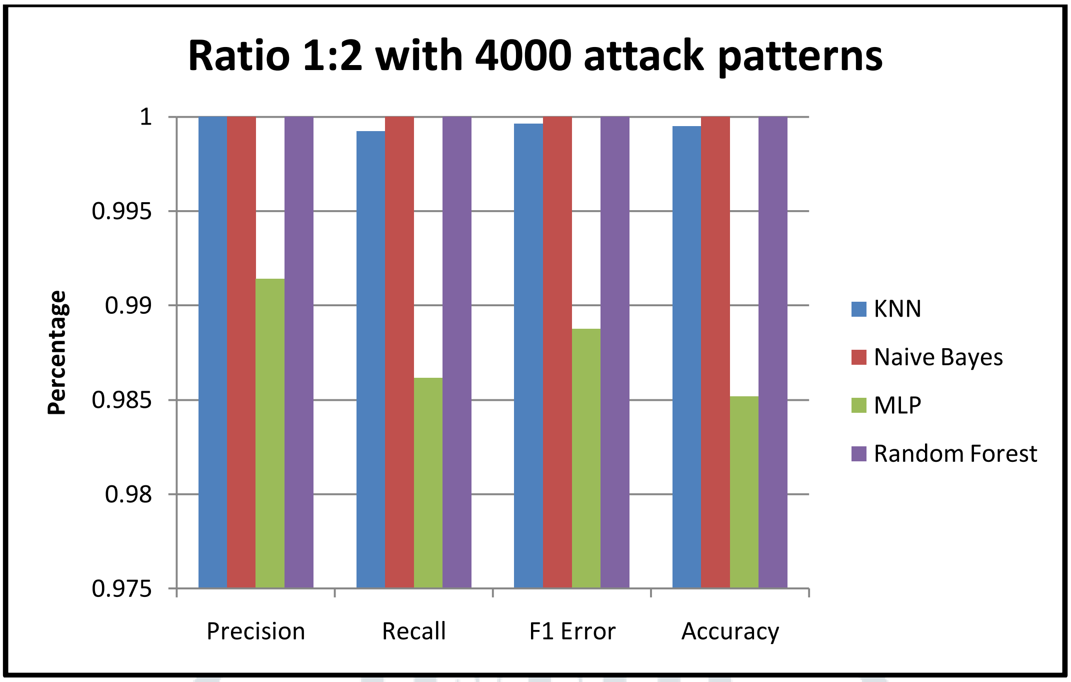
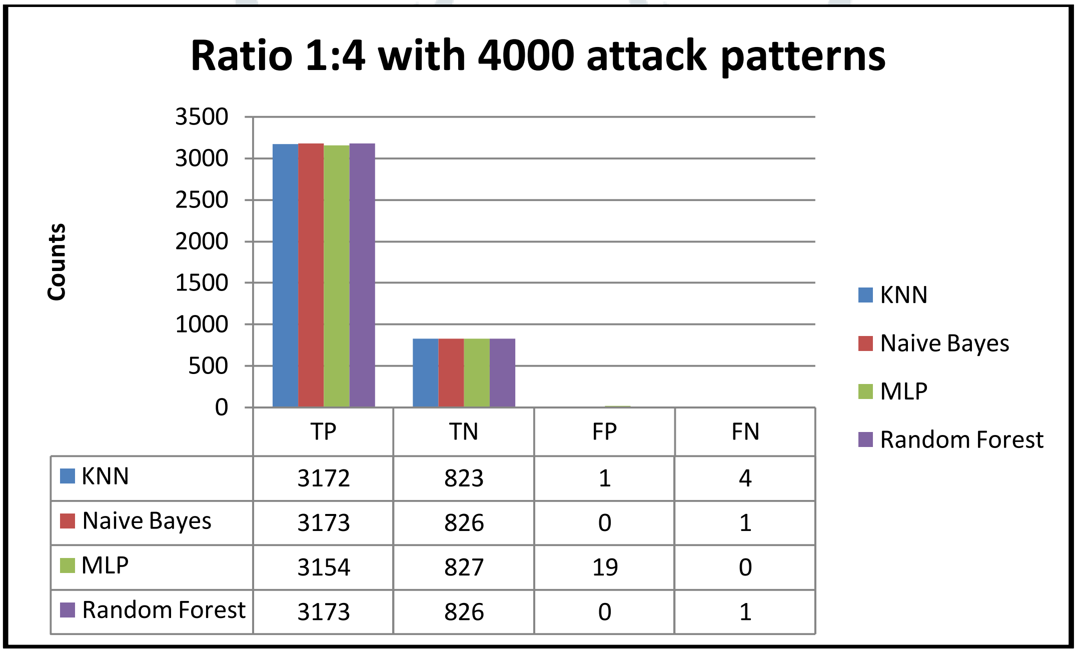


Figure 4.6 (b): Result analysis of ratio 1:2 with 4-Fold Cross validation with 4000 attack patterns per class with statistical accuracy based parameters



	TP	TN	FP	FN
🟦 KNN	3172	823	1	4
🟥 Naive Bayes	3173	826	0	1
🟩 MLP	3154	827	19	0
🟪 Random Forest	3173	826	0	1

Figure 4.7 (a): Result analysis of ratio 1:4 with 10-Fold Cross validation with 4000 attack patterns per class with statistical type 1 and 2 errors

Out of total 40000 rows of the input data, the 4000 rows are extracted for the testing set, and rest remain in the training set to test the classifier performance. The best performers are Random Forest and Naive Bayes, as both are recorded with only 1 false negative cases. The performance of KNN is underperforming with 1 false positive and 4 false negative cases. MLP is recorded with 30 false positive cases and no false negative case.

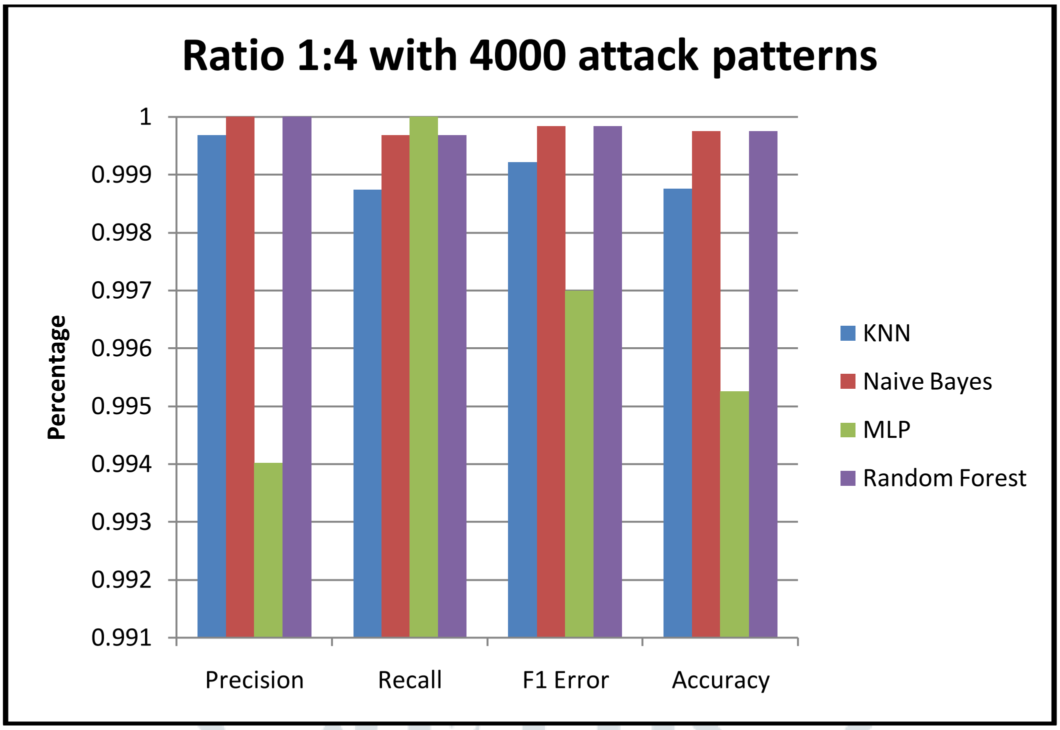
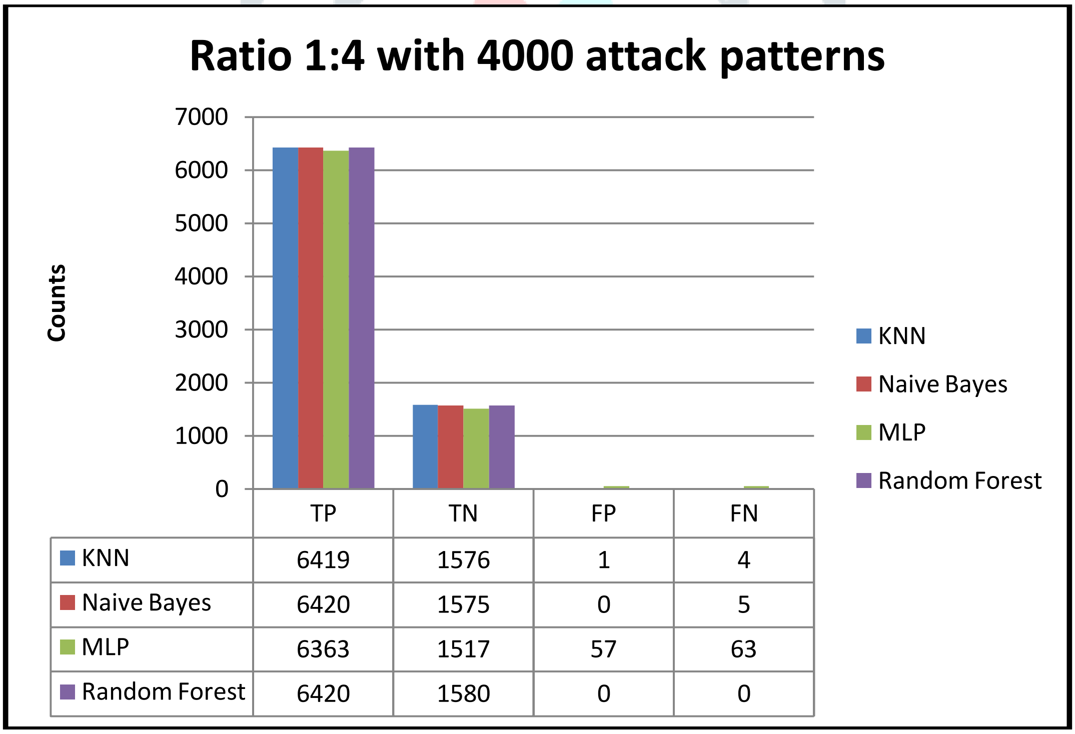


Figure 4.7 (b): Result analysis of ratio 1:4 with 10-Fold Cross validation with 4000 attack patterns per class with statistical accuracy based parameters



	TP	TN	FP	FN
🟦 KNN	6419	1576	1	4
🟥 Naive Bayes	6420	1575	0	5
🟩 MLP	6363	1517	57	63
🟪 Random Forest	6420	1580	0	0

Figure 4.8 (a): Result analysis of ratio 1:4 with 5-Fold Cross validation with 4000 attack patterns per class with statistical type 1 and 2 errors

Out of total 40000 rows of the input data, the 8000 rows are extracted for the testing set, and rest remain in the training set to test the classifier performance. Random Forest has been recorded with highest performance at 100% accuracy. KNN and Naive Bayes are recorded above 99% accuracy (all accuracy parameters), which are recorded with 4 and 5 false negative cases respectively. Additionally, 1 false positive case is recorded for the KNN, which caused a marginal decrease in the precision.

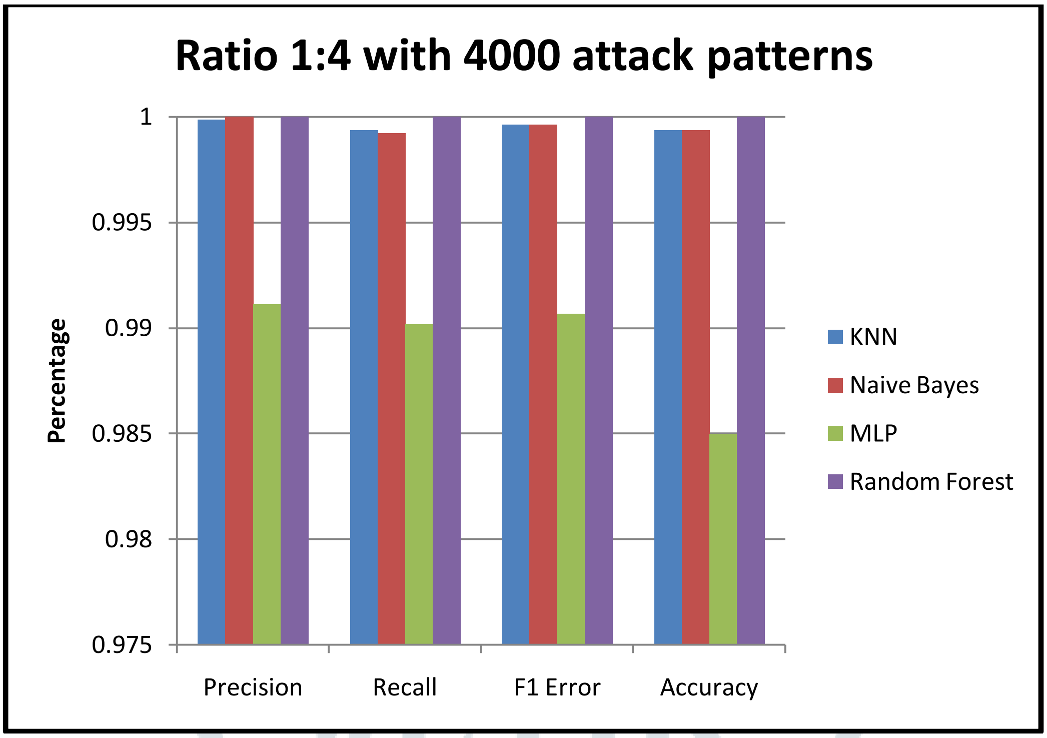
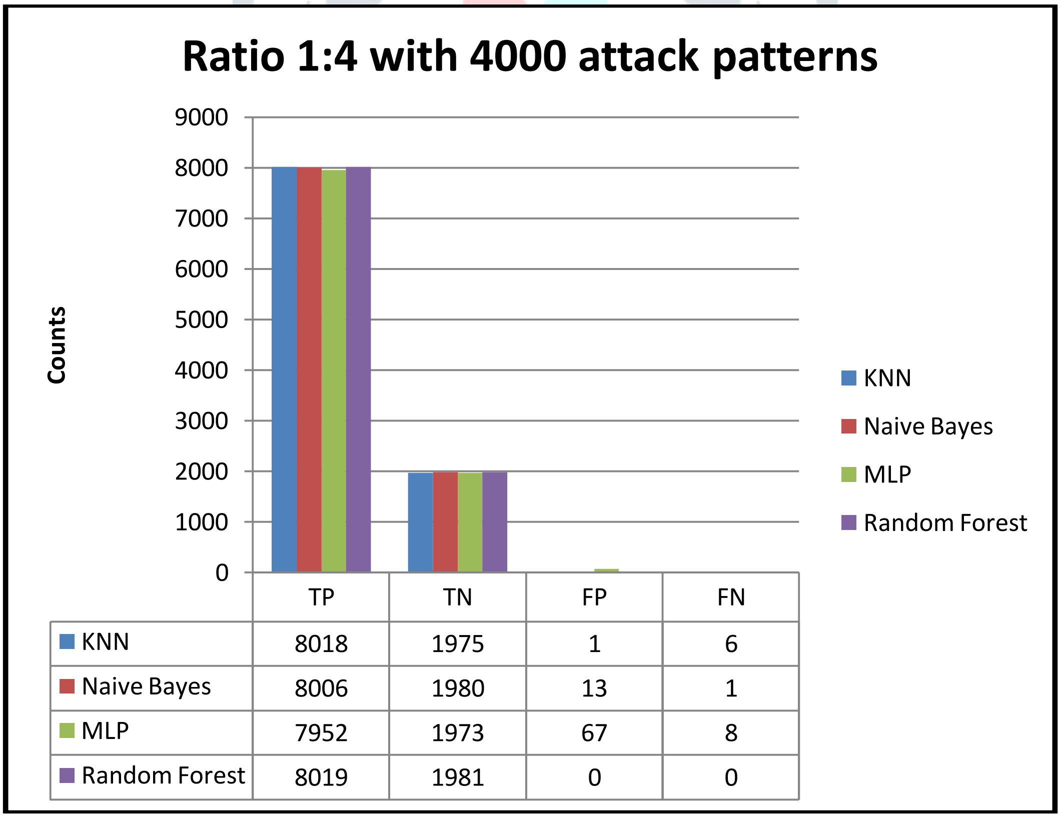


Figure 4.8 (b): Result analysis of ratio 1:4 with 5-Fold Cross validation with 4000 attack patterns per class with statistical accuracy based parameters



	TP	TN	FP	FN
KNN	8018	1975	1	6
Naive Bayes	8006	1980	13	1
MLP	7952	1973	67	8
Random Forest	8019	1981	0	0

Figure 4.9 (a): Result analysis of ratio 1:4 with 4-Fold Cross validation with 4000 attack patterns per class with statistical type 1 and 2 errors

Out of total 40000 rows of the input data, the 10000 rows are extracted for the testing set, and rest remain in the training set to test the classifier performance. Random Forest has been recorded with highest performance at 100% accuracy. KNN and Naive Bayes are recorded above 99% accuracy (all accuracy parameters), which are recorded with 6 and 1 false negative cases respectively. Also for the KNN and Naive Bayes are recorded with 1 and 13 false positive cases respectively.

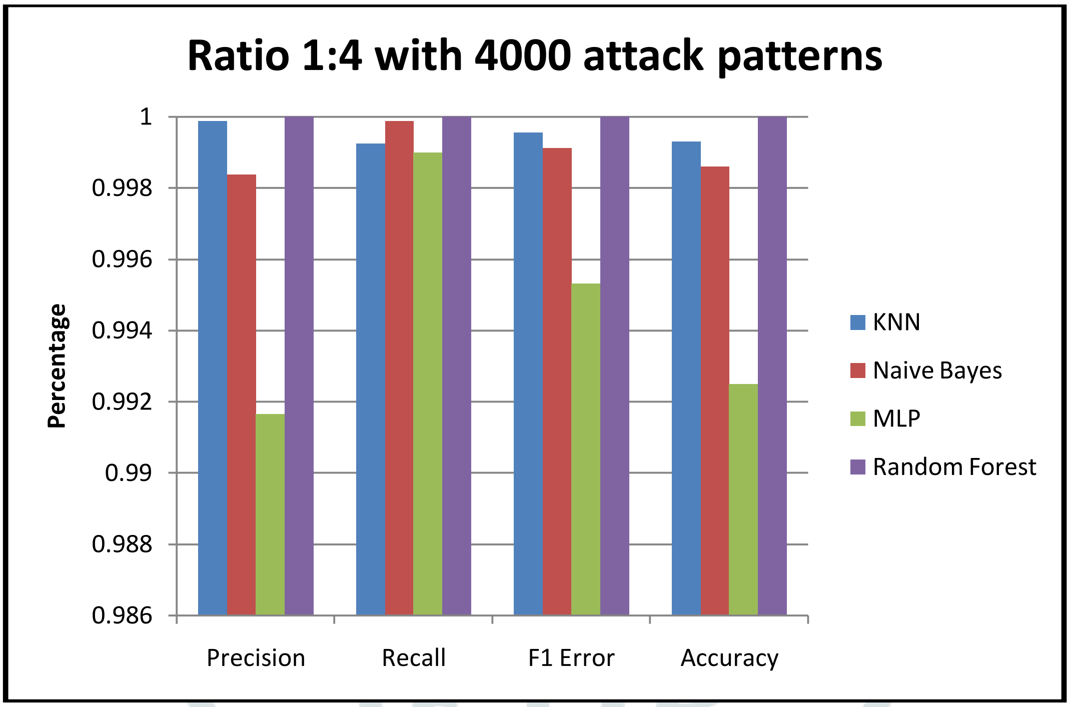


Figure 4.9 (b): Result analysis of ratio 1:4 with 4-Fold Cross validation with 4000 attack patterns per class with statistical accuracy based parameters

4. CONCLUSION

In this paper, different classification models are planned with KNN, Naive Bayes, Random Forest and Multilayer Perceptron (MLP). The evaluation of all of the classification models is performed on the prepared data using proposed dataset preparation methodology, which is used to handle the layer-3 information. The IP addresses of source and destination nodes are tracked from the dataset, and processed under the IP address decoding method, which extract and convert the IP address to numerical vector. The numerical vector is further processed using the numerical value handling procedure along with other features such as length, protocol and timestamp. The results are obtained with 4000 attack patterns with normal traffic patterns under different ratio based subsets, which include 1:1, 1:2 and 1:4. The Random Forest is found the best among all the classification models, which is followed mostly by KNN or Naive Bayes varying from case to case. MLP is analyzed an underachiever in almost all of the cases, except rare cases, which clearly discards the preference MLP classification on DDoS dataset. The performance of Random Forest is evaluated between 99.9% and 100% for all subsets of the dataset, where precisely the smaller data samples are included to testify its efficiency for DDoS attack detection. KNN classification model is also found at par with Random Forest, as it has been achieved above 99% accuracy for all of the evaluated parameters, whereas Naive Bayes has been observed with accuracy below 99% in some of the cases. Hence, the Random Forest is observed as the best classification algorithm for the detection of DDoS attacks over CAIDA dataset in this paper.

REFERENCES

[1] Reddy Kesavulu, “A study of intrusion detection in data mining”, 2011.
[2] Yao and Zhao, “A study on fuzzy intrusion detection”, 2007.
[3] Wei li, “Using genetic algorithm for network intrusion detection” 2003.
[4] Liao, Tianshengfeng, “Network forensics based on fuzzy logic and expert system”, science direct:1881-1892, 2009.
[5] Simranjeet and Neeta, “Soft computing in intrusion detection”, 2010.
[6] John and ali, “Network intrusion detection using an improved competitive learning neural network”, 2008.
[7] K. Ilgun and A. Kemmerer, “State transition analysis: A rule-based intrusiondetection approach”, IEEE Transaction on Software Engineering 21(3): 181-99 (1995).
[8] Kandeeban and S rajesh, “Integrated intrusion detection system using soft computing”, IJNS: 87-92, 2010.
[9] ZoranaBankovic, stepnvic, “Improved network security using genetic algorithm approach” , science direct: 438-451, 2007.
[10] Abadehsaniee, licas.c “Intrusion detection using a fuzzy genetic based learning algorithm”, science direct: 414-428, 2007.
[11] Toosiadel and Kahani, " A new approach on intrusion detection using genetic algorithm and neural network” science direct : 2007.
[12] Anupgoyal and chetan, “A genetic algorithm network intrusion detection system”, 2008.
[13] Islam and ahmed ,“ Fuzzy grid based intrusion detection in neural network”. 2012.
[14] Ben, Roja and Paramesvri, “Intrusion detection using fuzzy genetic algorithm”:IJARCSSE, 2012.
[15] Luo, susan M bridge, “Mining fuzzy association rules and fuzzy frequency episodes for intrusion detection”, vol-15. 2000.
[16] Kshirsagar and M. Tidke, “Intrusion detection system using genetic algorithm and data mining”: 2231-2232, 2012.

[17] Ramesh babu, “Intrusion detection using data mining along fuzzy logic and genetic algorithms”, IJCSNS:vol-8, 2008.
[18] Disha Sharma, “An intrusion detection using clustering techniques” IEEE: 2011.
[19] VarunChandola Anomaly Detection for Symbolic Sequences and Time Series Data,PhD. Dissertation. Computer Science Department, University of Minnesota,2009.
[20] Ojugo, A.O Eboka, “ Genetic algorithm rule based intrusion detection system” , CIS journal:1182-1190, 2012.
[21] Morgan Kaufman, “An Immunity-based Technique to Characterize Intrusions in Computer Network”, IEEE Transactions on Evolutionary Computation, Vol. 6(3), pp. 281-291.GECCO, pp.1081-1088.
[22] R.A. Kemmerer and G. Vigna “Intrusion Detection: A Brief History and Overview”, IEEE Computer, Vol. 1(1), pp. 27 – 30,2002.
[23] PrzemyslawKazienko and PiotrDorosz “Intrusion Detection Systems (IDS) Part 2 – Classification, methods; techniques", web white paper, 2004.
[24] Tarek S. Sobh and Wael M. Mostafa, “A cooperative immunological approach for detecting network anomaly", Applied Soft Computing, Elsevier, Vol. 11(1), pp. 1275-1283, 2011.
[25] P. Garcia Teodorro, J. Diaz-Verdejo, G. Marcia- Fernandez, E. Vazquez, “Anomaly-based network intrusion detection: Techniques, systems and challenges, Computers and Security” Vol.28(1-2), pp.18-28, 2009.
[26] Chimphlee, Abdullah hanan, “Unsupervised clustering methods for identifying rare events in anomaly detection” :740-747, 2007.
[27] Wei Li, “Using Genetic Algorithm for Network Intrusion Detection", Proceedings of the United States Department of Energy Cyber Security Grou, Training Conference, Vol. 8, pp. 24-27, 2004.
[28] K.S. Tang, K.F. Man, Z.F. Liu, S. Kwong, “Minimal fuzzy memberships and rules usinghierarchical genetic algorithms”, IEEE Trans. Ind. Electron, 162–169,1998.
[29] K.F. Man, K.S. Tang, S. Kwong, “Genetic algorithms: concepts and applications”, IEEE Trans. Ind. Electron, 519–534, 1996.
[30] W. Lee, S.J. Stolfo, K.W. Mok, “Adaptive intrusion detection: a data mining approach”, Artif. Intell. 533–567,2000.