I. J. Computer Network and Information Security, **2016, 9, 34-40**
Published Online September 2016 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijcnis.2016.09.05

Comparative Analysis of Classification Algorithms on KDD'99 Data Set

Iknoor Singh Arora and Gurpriya Kaur Bhatia
Systems Engineer, Infosys Technologies Ltd, India and USICT, GGSIPU, New-Delhi, India
E-mail: Iknoor99@hotmail.com, gurpriya06@gmail.com

Amrit Pal Singh
Assistant Professor, GTBIT, GGSIPU, New-Delhi, India
E-mail: amritpal1986@gmail.com

Abstract—Due to the enormous growth of network based services and the need for secure communications over the network there is an increasing emphasis on improving intrusion detection systems so as to detect the growing network attacks. A lot of data mining techniques have been proposed to detect intrusions in the network. In this paper study of two different classification algorithms has been carried out: Naïve Bayes and J48. Results obtained after applying these algorithms on 10% of the KDD'99 dataset and on 10% of the filtered KDD'99 dataset are compared and analyzed based on several performance metrics. Comparison between these two algorithms is also done on the basis of the percentage of correctly classified instances of different attack categories present in both the datasets as well as the time they take to build their classification models.Overall J48 is a better classifier compared to Naïve Bayes on both the datasets but it is slow in building the classification model.

Index Terms—Intrusion detection system, Naïve Bayes, J48, DD`99(Knowledge Discovery and Data Mining).

I. Introduction

With the enormous growth of technology, the number of applications running on top of the computer network has increased drastically. Due to this network security is becoming increasingly more important as well as complex. Hence, intrusion detection systems (IDS) are used to detect anomalies and attacks in the network. These systems are dynamic in nature that is they gather and analyzes information from various areas within a computer or a network to identify possible security breaches. There are three elements that are central to intrusion detection systems, the first being resources that are to be protected in the target system, second is the model that characterises the behaviour of the system to be normal or illicit and the third are the techniques that compares the actual system activities with the established model[1]. The goal of such a system is to have a high detection rate while keeping the false alarm rate as minimum as possible. There are two types of intrusion detection systems which are Host-Based (HIDS) and Network-Based (NIDS) [2]. Host-Based intrusion detection system resides on the host (computer). It gains knowledge of user activity and generates an alarm when it encounters any deviation from the learned profile and reports it to system administrator. Network-Based intrusion detection system is used to analyze and monitor network traffic in order to protect a system from network-based threats. A NIDS reads all incoming packets and searches for any suspicious patterns.

Further there are two intrusion detection techniques which are Misuse-Based and Anomaly-Based intrusion detection technique. Misuse-Based intrusion detection scheme maintains patterns or signatures that represent known attacks. It examines the network traffic for such patterns in order to detect attacks. It fails to detect attacks whose patterns are not known. In anomaly-Based detection scheme any action that is different from the normal behaviour is termed as anomaly. It checks for the normal and abnormal behaviour of the system [3].It classifies using rules or heuristics.

Classification is considered an instance of supervised learning in which a training set of correctly labeled instances are used to build a model. This model is then used to solve the problem of identifying to which set of categories a new instance belongs, on the basis of training set of data containing instances whose category membership is known.

This paper attempts to analyse two classification algorithms that are Naive Bayes and J48. 10% of the original KDD`99 data set has been used for training as well as for testing purpose. Use of pre processing filters to remove the duplicate instances has been carried out. The resulting data set without duplicate instances is also used for training and testing purpose. The test results of both the algorithms applied on both the data set are then compared.

The structure of the paper is as follows: section 2 discusses related work, section 3 specifies our approach in detail section 4 provides our result and section 5 draws the conclusion and states the future work.

II. Related Work

This section covers the work that has been carried out to evaluate and analyse various classifiers in order to

detect intrusions in the datasets.

Panda and Patra[4] evaluated the performance of three well known data mining classification algorithms namely, ID3, J48 and Naïve Bayes based on the 10-fold cross validation test, using the KDDCup'99 IDS data set.

Gharibian and Ghorbani[5] employed two probabilistic techniques Naive Bayes and Gaussian and two predictive techniques Decision Tree and Random Forests. They constricted Different training datasets constructed from the KDD99 dataset that were used for training. They compared the ability of each technique for detecting the four attack categories. A. Adebowale, S.A Idowu , A. Amarachi [6] evaluated the performance of well known classification algorithms for attack classification. Their focus was on five of the most popular data mining algorithms that are: Decision trees, Naïve bayes, artificial neural network, K-nearest neighbour algorithm and Support vector machines. They have also discussed the advantages and disadvantages of these algorithms. Sinha, Kumar and Kumar in their work [7], have implemented various Artificial intelligence based techniques in IDS. They used normal and anomalous classes to classify the network traffic in order to detect intrusion, they tried to identify the best techniques for the different attack categories.

Amor, Benferhat, Elouedi [8] have done their experimental study on KDD'99 intrusion data set. They closely analyzed the importance of Naive Bayes in intrusion detection. Three levels of attack granularities were considered depending on whether dealing with whole attacks, or grouping them in four categories or just considering normal and abnormal behaviors. Comparison between Naive Bayes networks and decision tree was carried out.Lee, Stolfo and Mok [9] have proposed a data mining framework to build intrusion detection models. According to them, learning rules that precisely capture the normal and intrusive behavior of activities can be used for detecting intrusions.

Chandolikar and Nandavadekar [10] have evaluated the performance of J48 classification algorithm based on the correctly classified instances, Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), Root relative squared error and kappa statistics measures. They have applied feature selection on KDD cup data set before evaluating the performance of the algorithm. Patil and Sherekar [11] have evaluated Naïve Bayes and J48 classification algorithm in the context of bank data set. Their focus was on measuring the performance of classification algorithm based on True Positive rate and False Positive rate. Jalil and Masrek [12] in their paper evaluated the performance of J48 classification algorithm and compared its result to two other machine learning algorithm that are Neural Network and support Vector Machine based on the detection rate, false alarm rate and accuracy of classification based on attack type.

This paper compares the performance of Naïve Bayes and J48 algorithms on KDD'99 data set and studies the effects of removing redundancy from the dataset by applying preprocessing filter (i.e. RemoveDuplicate) present in Weka version 3.7.

III. Experimental Approach

A. Algorithm

Naive Bayes: It is a probabilistic classifier and is based on applying Bayesian theorem. It works on a strong assumption that the features in the dataset all are independent of each other. This assumption makes the algorithm quick and easy as well as it proves to be a limitation since features are not actually independent of each other.

According to Bayes rule, the expression [13] for probability that class Y will have value Y_i given the value of feature vector (X_1 ...X_n), is calculated using (1).

$$P(Y = y_k|X_1 \dots X_n)=\frac{P(Y=Y_i)*P(X_1 \dots X_n|Y=y_i)}{\sum_j P(Y=y_j)*P(X_1 \dots X_n|Y=y_j)} \quad (1)$$

Since Naive Bayes assumes that all feature values are independent of each other equation (1) can be written as equation (2)

$$P(Y = y_k|X_{1\dots}X_n)= \frac{P(Y=y_i)\prod_k(X_k|Y=y_i)}{\sum_j P(Y= y_j)* \prod_k P(X_k|Y=y_j)} \quad (2)$$

We need to find the most probable value of class attribute that is the value of variable Y, which can be found using equation (3)

$$Y \leftarrow \arg\max P(Y = y_k)\prod_i P(X_i |Y = y_k) \quad (3)$$

The denominator in (2) is not considered in (3) for simplicity since the denominator is independent of y_k.

J48: It builds a decision tree from feature value of training set using info gain in order to classify new instances. It uses a recursive divide and conquers strategy to build the decision tree, starting with attribute having the highest information gain. The internal nodes of the tree represent different attributes of the data set, the branches denote the possible values that these attribute can take as observed from the samples and the leaf or the terminal nodes tells us the predicted value of the class for that instance.

In order to classify a new instance, a decision tree is created if it already does not exist, based on the attribute values of the training set. We traverse the tree depending on the attribute value of the instance until we reach the leaf node that tells us the class label for that instance. If some ambiguity exists the branch is assigned the target value that majority of the items under this branch possess [14].

B. Pre-Processing filter

Real-world data is often incomplete and lacks in certain behaviours or trends, may also contain many errors. Data pre-processing transforms the raw data into a format that is useful for analysis and is easily understandable [15]. The weka.filters package includes classes that transform datasets by removing instances, resampling the dataset, removing or adding attributes, and so on. This package is organized into supervised and unsupervised filtering, which are further subdivided into

instance and attribute filter [16]. In this paper we have used unsupervised instance filter called RemoveDuplicates. It removes all the duplicate instances from the first batch of data it receives.

C. Data set

KDD'99 Dataset was prepared by Stolfo et al, based on the data captured in DARPA'98 IDS evaluation program. KDD'99 Dataset is about 4 gigabytes of compressed raw(binary) TCP dump data prepared by monitoring 7 weeks of network traffic, which contains about 5 million connection records with each record taking about 100 bytes of memory[17].

It contains 4,940,200 instances each of which contains 41 features. The 42nd feature is the class label which shows the attack category the instance belongs to and is determined by these 41 features. This data set contains large number of intrusions which were mapped in a military network environment [18].

KDD'99 features are divided into three groups

1. Basic Features: It includes all the features that can be linked to a TCP/IP Connection. Considering these features for attack detection takes a lot of time.
2. Traffic Features: These types of features are related to the time interval a connection is examined. They include 'same host' features and 'same service' features.
3. Context Features: This type of features searches for the dubious behaviour in the data set, so as to classify certain uncommon attack categories in the dataset.

This data set contains 4 types of intrusions

1. Denial of Service (DoS): These attacks directly target the server infrastructure. They make the online resources unavailable to the legitimate users.
2. Probe: The probe attacks are aimed at monitoring or collecting information about the vulnerability of a network or host. This information can later be used to exploit the privacy and security of the system.
3. User to Root: The attacker starts out as a user to the system. It then exploits various vulnerabilities of the system to gain the root access of the system.
4. Remote To Local (R2L):In this type of attack the attacker gains unauthorized access to a local account on a remote machine on which it can send packets through a network.

The class label determines whether the instance is a normal connection or an intrusion. The above type of intrusions can be subcategorised into 22 types of attacks.

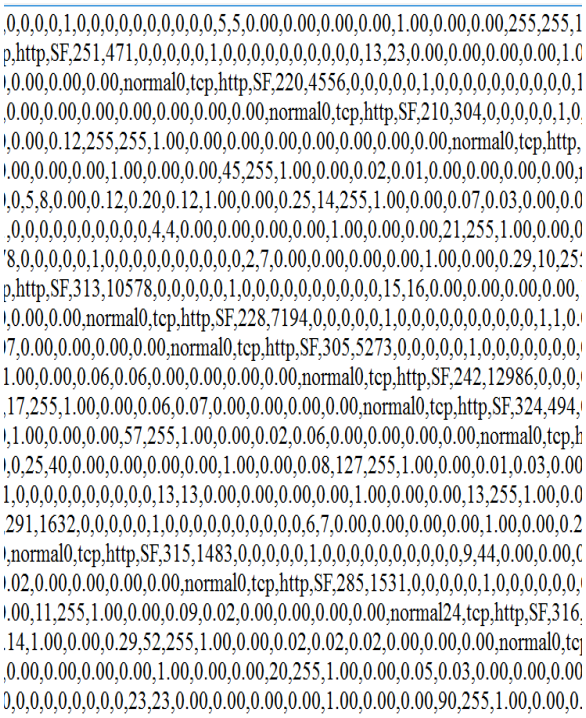
,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,5,5,0.00,0.00,0.00,0.00,1.00,0.00,0.00,255,255,1
p,http,SF,251,471,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,13,23,0.00,0.00,0.00,0.00,1.0
,0.00,0.00,0.00,normal0,tcp,http,SF,220,4556,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,1
,0.00,0.00,0.00,0.00,0.00,0.00,0.00,normal0,tcp,http,SF,210,304,0,0,0,0,0,1,0,
,0.00,0.12,255,255,1.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,normal0,tcp,http,
.00,0.00,0.00,1.00,0.00,0.00,45,255,1.00,0.00,0.02,0.01,0.00,0.00,0.00,0.00,
,0,5,8,0.00,0.12,0.20,0.12,1.00,0.00,0.25,14,255,1.00,0.00,0.07,0.03,0.00,0.0
,0,0,0,0,0,0,0,0,0,4,4,0.00,0.00,0.00,0.00,1.00,0.00,0.00,21,255,1.00,0.00,0
'8,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,2,7,0.00,0.00,0.00,0.00,1.00,0.00,0.29,10,25
p,http,SF,313,10578,0,0,0,0,1,0,0,0,0,0,0,0,0,0,15,16,0.00,0.00,0.00,0.00,
,0.00,0.00,normal0,tcp,http,SF,228,7194,0,0,0,0,1,0,0,0,0,0,0,0,0,0,1,1,0.
7,0.00,0.00,0.00,0.00,normal0,tcp,http,SF,305,5273,0,0,0,0,1,0,0,0,0,0,0,
1.00,0.00,0.06,0.06,0.00,0.00,0.00,0.00,normal0,tcp,http,SF,242,12986,0,0,0,
,17,255,1.00,0.00,0.06,0.07,0.00,0.00,0.00,0.00,normal0,tcp,http,SF,324,494,
,1.00,0.00,0.00,57,255,1.00,0.00,0.02,0.06,0.00,0.00,0.00,0.00,normal0,tcp,h
,0,25,40,0.00,0.00,0.00,0.00,1.00,0.00,0.08,127,255,1.00,0.00,0.01,0.03,0.00
1,0,0,0,0,0,0,0,0,0,13,13,0.00,0.00,0.00,0.00,1.00,0.00,0.00,13,255,1.00,0.0
,291,1632,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,6,7,0.00,0.00,0.00,0.00,1.00,0.00,0.2
,normal0,tcp,http,SF,315,1483,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,9,44,0.00,0.00,0
.02,0.00,0.00,0.00,0.00,normal0,tcp,http,SF,285,1531,0,0,0,0,0,1,0,0,0,0,0,0,
.00,11,255,1.00,0.00,0.09,0.02,0.00,0.00,0.00,0.00,normal24,tcp,http,SF,316,
.14,1.00,0.00,0.29,52,255,1.00,0.00,0.02,0.02,0.02,0.00,0.00,0.00,normal0,tcp
,0.00,0.00,0.00,0.00,1.00,0.00,0.00,20,255,1.00,0.00,0.05,0.03,0.00,0.00,0.00
0,0,0,0,0,0,0,0,0,23,23,0.00,0.00,0.00,0.00,1.00,0.00,0.00,90,255,1.00,0.00,0.

Fig.1. Snapshot of Portion of 10% KDD '99 Data set

Table 1 compares the number of intrusion instances present for specific attack category in the original 10% KDD`99 dataset and the filtered 10% KDD`99 dataset. The Filtered KDD`99 dataset was obtained by applying the pre processing unsupervised instance filter on 10% KDD`99 dataset. The number of instances present in the 10% KDD`99 dataset and filtered 10% KDD`99 dataset are 4, 94,020 and 1, 45,585 respectively.

Table 1. Number of Intrusion Instances of a Particular Attack Type

ATTACKS	ORIGINAL DATA SET	FILTERED DATA SET	INTRUSIONS
BACK	2203	968	DOS
LAND	21	19	DOS
NEPTUNE	107201	51820	DOS
POD	264	206	DOS
SMURF	280790	641	DOS
TEARDROP	979	918	DOS
SATAN	1589	906	PROBE
IPSWEEP	1247	651	PROBE
NMAP	231	158	PROBE
PORTSWEEP	1040	416	PROBE
GUESS_PASSWD	53	53	R2L
FTP_WRITE	8	8	R2L
IMAP	12	12	R2L
PHF	4	4	R2L
MULTIHOP	7	7	R2L
WAREZMASTER	20	20	R2L
WAREZCLIENT	1020	893	R2L
SPY	2	2	R2L
BUFFER_OVERFLOW	30	30	U2R
LOADMODULE	9	9	U2R
PERL	3	3	U2R
ROOTKIT	10	10	U2R

From Fig. 2 and 3 it can be seen that there were large number of duplicate Dos and Probe attacks in the original 10% KDD`99 dataset where as there were no duplicate entries present for U2R attack category.

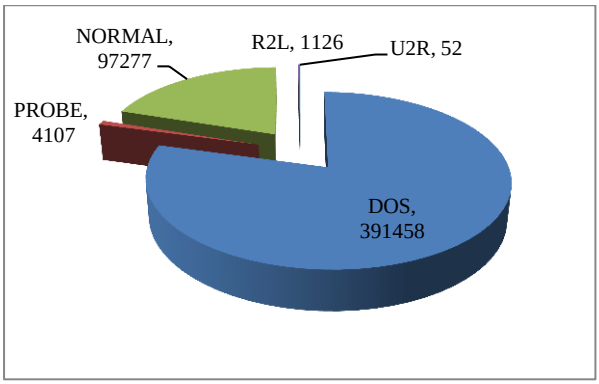

Fig.2. Number of Normal As Well As Intrusion Instances Present In Original 10% KDD'99 Dataset.

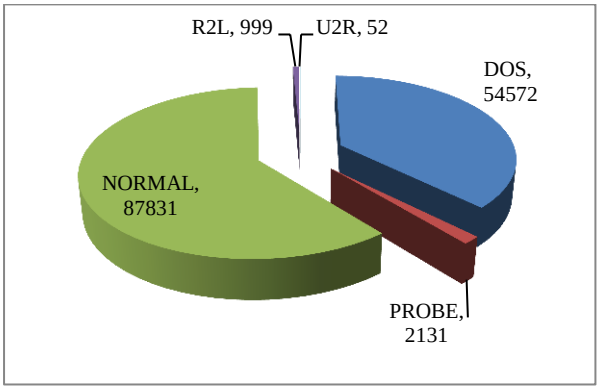

Fig.3. Number of Normal As Well As Intrusion Instances Present In Filtered 10% KDD'99 Dataset.

Table 2 shows the percentage of instances of each attack type in 10% KDD'99 data set and filtered 10% KDD'99 data set.

Table 2. Percentage of Intrusion Instances of a Particular Attack Type

% INSTANCES IN ORIGINAL KDD	% INSTANCES IN FILTERED KDD	ATTACK TYPE
79.2	37.4	DOS
0.83	1.4	PROBE
19.6	60.3	NORMAL
0.22	0.68	R2L
0.010	0.035	U2R

D. Parameter Used

To evaluate the performance of the classifiers we have used k-fold cross validation [19]. In this method the data set is divided randomly into k disjoint set of instances resulting into k trials.

In each trial k-1 sets are used for training purpose and the remaining one is used for testing.

The metrics computed using k=2:

True Positive Rate

It is defined as the ratio of instances that are correctly classified for a class to the total no. of instances belonging to that class.

$$\text{True positive rate} = \frac{\text{True positive}}{\text{True positive+False negative}} \quad (4)$$

Precision

It is the proportion of instances that actually belong to a class to the total number of instances that are classified as that class.

$$\text{Precision} = \frac{\text{True positive}}{\text{True positive+False positive}} \quad (5)$$

Recall

It is the ratio of instances that are classified as a given class to the actual number of instances that belong to that class.

$$\text{Recall} = \frac{\text{True positive}}{\text{True positive+False negative}} \quad (6)$$

F-Measure

This depends on two measures precision and recall.

F-Measure = (2* Precision * Recall) / (Precision +Recall). (7)

IV. Result

We have used Waikato Environment for Knowledge Analysis (WEKA) as a tool for comparative study. WEKA is a popular machine learning software coded in java, developed at University of Waikato, New Zealand. WEKA version 3.6[20] has many new features as compared to version 3.4. Furthermore the results compiled in this paper are analyzed and collected using version 3.7.This version of weka provides us with an additional functionality in the form of a pre processing Unsupervised filter used to remove duplicate instances from the data set. The results present in the following section were acquired on Intel core i5 CPU, 1.7GHz, 4GB RAM.

Our goal was to evaluate the performance of two classification algorithms that are Naive Bayes and J48 on the original 10% KDD`99 dataset as well as on filtered 10% KDD`99 dataset. Filtered 10% KDD`99 dataset was obtained by applying the unsupervised filter RemoveDuplicates on original 10% KDD`99 dataset.

Table 3 gives us the time to build the model in seconds for J48 as well as Naive Bayes algorithm.We can see a notable increase in speed for building up the model while using filtered KDD`99 data set for both the algorithms. The larger the number of instances to train the model the greater is the time to build the model.

Table 3. Time to Build the Model

ALGORITHM	ORIGINAL 10% KDD(sec)	FILTERED 10% KDD(sec)
Naive Bayes	5.12	1.8
J48	86.21	47.00

Figure 4 and 5 shows the performance of Naive Bayes and J48 on 10% original KDD'99 data set and on 10% filtered KDD'99 data set respectively. It appears that removing the duplicate instances has significantly reduced the true positive rate for Naïve Bayes which comes down to 0.77 from 0.927. It is because from equation 1 for Naive Bayes we can see that the probability that an instance belongs to a specific class is directly proportional to the probability of that class. So a class having higher probability would be favourable when the product terms are almost similar. Hence, all unique instances in the data set lead to lesser accuracy in terms of the correctly classified instances for Naïve Bayes. On the contrary the percentage of correctly classified instances for J48 remains almost same after the filter is applied on the dataset, which shows that J48 is better at training the records with Duplicate data and is not biased towards more frequently occurring records.

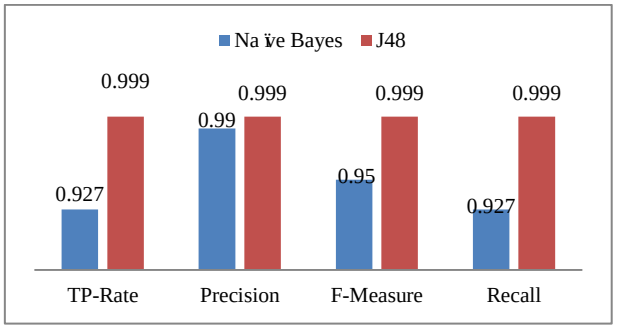

Fig.4. Comparison between Naive Bayes and J48 on Original 10% KDD'99 Dataset.

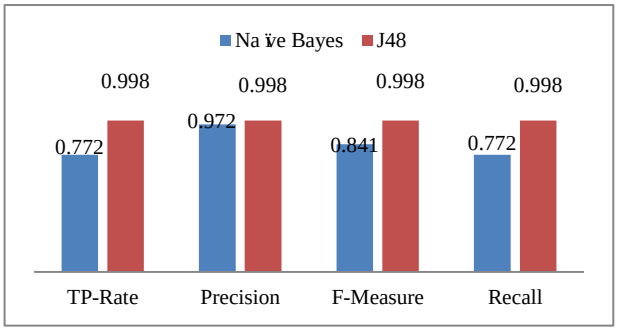

Fig.5. Comparison between Naïve Bayes and J48 on Filtered 10% KDD'99 Dataset

Table 4. Number of Intrusion Instances Correctly Classified By Algorithms

ATTACK TYPE	Number of instances in original 10%KDD'99 dataset	Number of Correctly classified instances in original 10% KDD'99 dataset		Number of instances in filtered 10% KDD'99 dataset	Number of Correctly classified instances in filtered 10% KDD'99 dataset		INTRUSION TYPE
		Naïve Bayes	J48		Naïve Bayes	J48	
Back	2203	2143	2190	968	933	962	DOS
Teardrop	979	974	978	918	914	917	
Neptune	107201	106805	107200	51820	51576	51801	
Land	21	19	17	19	18	16	
Smurf	280790	280383	280784	641	639	634	
Pod	264	259	260	206	203	204	
Satan	1589	1514	1569	906	852	882	PROBE
Ipsweep	1247	1190	1239	651	562	639	
nMap	231	111	223	158	29	133	
Portsweep	1040	966	1025	416	324	402	
Loadmodule	9	6	1	9	0	0	U2R
Rootkit	10	5	0	10	3	0	
Buffer overflow	30	15	22	30	12	19	
Perl	3	0	0	3	1	1	
Phf	4	3	4	4	3	4	R2L
ftp_write	8	5	0	8	4	0	
Spy	2	2	0	2	2	0	
Multihop	7	2	2	7	2	0	
Wareclient	1020	459	987	893	437	830	
Imap	12	11	2	12	11	3	
Warezmaster	20	16	16	20	17	16	
Guess password	53	50	50	53	50	50	
Normal (not an attack)	97277	63176	97203	87831	55796	87773	NORMAL

Table 5. Percentage of Intrusion Instances Correctly Classified by the Algorithms

INTRUSION	Correctly classified intrusion % in Original 10% KDD'99 dataset		Correctly classified intrusion % in filtered 10% KDD'99 dataset	
	NAÏVE BAYES	J48	NAÏVE BAYES	J48
DOS	99.77	99.99	99.47	99.93
PROBE	92.06	98.75	82.91	96.48
U2R	50.00	44.23	30.76	38.46
R2L	48.66	93.87	52.65	90.39
NORMAL	64.94	99.92	63.52	90.22

Table 4 and 5 shows that the efficiency of both the algorithms to detect and classify the Dos attacks has been excellent on both the datasets, though there is minute difference in the percentage of correctly classified instances by both algorithms i.e a decrease from 99.7% in original KDD`99 dataset to 99.4% in filtered KDD`99 dataset when classification is done using Naïve Bayes and decrease from 99.99% in original kdd99 dataset to 99.93% in filtered KDD`99 dataset when classification is done using J48 algorithm. More prominent difference in the percentage is seen while classifying the Probe attacks, a reduction in the correctly classified instances by 9.15% is seen when classification is done by Naïve Bayes and 2.27% when done by J48 algorithm. This difference in the percentage of correctly classified instances is due to the large number of duplicate instances of Dos and Probe present in the original kdd99 dataset. Further observation of results from both the tables tells us that both the algorithms are weak in correctly detecting U2R and R2L attacks. Efficiency of Naïve Bayes and J48 is reduced on U2R and R2L since the numbers of instances of these intrusions are least in the dataset and hence the model cannot be properly trained to detect them.

V. Conclusion and Future Work

We statistically analyzed the entire 10% KDD'99 data set, filtered 10% KDD'99 dataset in Weka. There is large number of redundant Dos and Probe attacks in original KDD'99 dataset whereas NORMAL instances and R2L attacks are less redundant in the original set. There are no duplicate entries present for U2R attack category in the original KDD99 dataset. Our results show that Naïve Bayes classifier is biased towards duplicate records and is weak at training the model with less frequent records, J48 tree classifier is also biased towards duplicate records but not to the extent as compared to Naïve Bayes classifier. In all we can see that J48 is better classifier than Naïve Bayes for KDD`99 data set. Due to the reduction in the number of instances from 4,94,020 in 10% KDD`99 dataset to 1,45,585 instances after the removal of redundant records the time to build the model has been reduced for both the algorithms, which indicates quick analysis by the classifiers. Naive Bayes algorithm is fast and easy to implement whereas algorithms with least percentage of errors are complex and slow such as J48 which considers the features with the highest information gain while constructing the decision tree. Dos and probe attacks have high accuracy rate because of the large number of instances present in the training set for these type of intrusions which makes the model learn easily about them and hence precisely detect such attacks. Accuracy of J48 was much higher than Naïve Bayes to detect different types of attacks in both the datasets whereas none of them could properly classify the U2R attacks in the KDD'99 dataset. Our Future work would be based on the concept of Feature Selection so as to remove irrelevant and redundant features from the data set as well as to select the best features in order to increase the accuracy rate of U2R and R2L.

Reference

[1] K.Lahre, T. Diwan, P. agrawal, S. K. Kashyap, "Analyze different approaches for IDS using KDD'99 data set", *International Journal on Recent and Innovation Trends in Computing and Communication*, August 2013,pp. 645-651.

[2] Bilal Maqbool Beigh, "A New Classification Scheme for Intrusion Detection Systems", *I.J. Computer Network and Information Security*, 2014, 8, 56-70

[3] Ashutosh Gupta, Bhoopesh Singh Bhati, Vishal Jain," Artificial Intrusion Detection Techniques: A Survey", *I.J. Computer Network and Information Security*, 2014, 9, 51-57.

[4] M. Panda,M. R. Patra, "A comparative study of data mining algorithms for network intrusion detection", *First International Conference on Emerging Trends in Engineering and Technology*, IEEE, 2008, pp.504-507

[5] F. Gharibian and A. Ghorbani, "Comparative Study of Supervised Machine Learning Techniques for Intrusion Detection", *Fifth Annual Conference on Communication Networks and Services Research (CNSR'07)*, IEEE, 2007.

[6] A. Adebowale, S.A Idowu, A. Amarachi, "Comparative Study of Selected Data Mining Algorithms Used For Intrusion Detection", *International Journal of Soft Computing and Engineering (IJSCE)*,Volume-3, Issue-3, July 2013, pp.237-241.

[7] N. K. Sinha, G. Kumar,K. Kumar, "A Review on Performance Comparison of Artificial Intelligence Techniques Used for Intrusion Detection", *International Conference on Communication, Computing & Systems (ICCCS)*, 2014, pp.209-214.

[8] N. B. Amor,S. Benferhat,Z. Elouedi, "Naive Bayes vs Decision Trees in Intrusion Detection Systems", *Symposium on Applied Computing*, ACM, 2004, pp 420-424.

[9] W. Lee, S. Stolfo, and K. Mok, "A Data Mining Framework for Building Intrusion Detection Models", *Proc, Of the 1999 IEEE Symposium on Security and Privacy*, IEEE, May 1999.

[10] N. S. Chandolikar, V. D. Nandavadekar," Efficient Algorithm for Intrusion Attack Classification by

Analyzing KDD Cup 99", *Wireless and Optical Communications Networks (WOCN)*, 2012 Ninth International Conference, IEEE, Sept. 2012, pp 1-5.
[11] T. R. Patil, S. S. Sherekar,"Performance Analysis of Naive Bayes and J48 Classification Algorithm for Data Classification", *International Journal Of Computer Science And Applications* Vol. 6, No.2, pp-256-261, Apr 2013
[12] Kamarularifin Abd Jalil, Mohamad Noorman Masrek, "Comparison of Machine Learning Algorithms Performance in Detecting Network Intrusion", *International Conference on Networking and Information Technology*, IEEE, 2010, pp 221-226.
[13] Tom M. Mitchell, *Machine Learning*, McGrawHill, 2015
[14] Tan et al, *Classification, 3rded.*, vol 1, Gerstein Lab, 2005.
[15] Data Preprocessing. Available on: https://www.techopedia.com/definition/14650/data-preprocessing.
[16] Weka. filters package. Available on: http://weka.sourceforge.net/doc.dev/weka/filters/Filter.html
[17] M.Tavallaee, E. Bagheri, W. Lu and A. A. Ghorbani, "A Detailed Analysis of the KDD CUP 99 Data Set", *Proceedings of the 2009 IEEE Symposium on Computational Intelligence in Security and Defence Applications*, IEEE, 2009.
[18] KDD Cup 1999. Available on: http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html
[19] N.williams, S.Zander, G. Armitage, "A preliminary performance comparison of five machine learning algorithms for practical IP traffic flow classification", *SIGCOMM computer communication review*, ACM, October 2006, pp. 7-15.
[20] M. Hall, E. Frank, "The WEKA data mining software: An update", *SIGKDDExplorations*, Volume II, pp.10-18.

Authors' Profiles

Iknoor Singh Arora, Born in 1993, currently working as a Systems Engineer at Infosys Technologies Ltd. He has done his B.Tech in Computer Science from GTBIT, GGSIPU, New Delhi.He has been doing research on KDD'99 Dataset since 6 months. email:iknoor99@hotmail.com

Gurpriya Kaur Bhatia, Born in 1993, currently pursuing M.Tech in Information Technology from USICT, GGSIPU, New Delhi. She has done her B.Tech in Computer Science from GTBIT, GGSIPU, New Delhi. She has been doing research on KDD'99 Dataset since 6 months. email:gurpriya.kaur@gmail.com

Amrit Pal Singh is Assistant Professor, GTBIT, GGSIPU, New Delhi, India and Pursuing his Ph.D from GGSIPU. He obtained his M.Tech degree in Information Technology from USICT, GGSIPU, New Delhi and B.Tech in Information Technology from GTBIT, GGSIPU, New Delhi, e-mail: amritpal.ipu@gmail.com**.**

How to cite this paper: Iknoor Singh Arora, Gurpriya Kaur Bhatia, Amrit Pal Singh,"Comparative Analysis of Classification Algorithms on KDD'99 Data Set", International Journal of Computer Network and Information Security(IJCNIS), Vol.8, No.9, pp.34-40, 2016.DOI: 10.5815/ijcnis.2016.09.05

