2015 International Conference on Smart Technologies and Management for Computing, Communication, Controls, Energy and Materials (ICSTM), Vel Tech Rangarajan Dr. Sagunthala R&D Institute of Science and Technology, Chennai, T.N., India. 6 - 8 May 2015. pp.89-95.

Comparative Analysis of Machine Learning Algorithms along with Classifiers for Network Intrusion Detection

Sumouli Choudhury[1] and **Anirban Bhowal**[2]

Department of Information Technology, Indian Institute of Information Technology, Allahabad, India
Email: [1]sumouli143@gmail.com, [2]anirbanb45@gmail.com

***Abstract*— Intrusion detection is one of the challenging problems encountered by the modern network security industry. A network has to be continuously monitored for detecting policy violation or suspicious traffic. So an intrusion detection system needs to be developed which can monitor network for any harmful activities and generate results to the management authority. Data mining can play a massive role in the development of a system which can detect network intrusion. Data mining is a technique through which important information can be extracted from huge data repositories. In order to spot intrusion, the traffic created in the network can be broadly categorized into following two categories- normal and anomalous. In our proposed paper, several classification techniques and machine learning algorithms have been considered to categorize the network traffic. Out of the classification techniques, we have found nine suitable classifiers like BayesNet, Logistic, IBK, J48, PART, JRip, Random Tree, Random Forest and REPTree. Out of the several machine learning algorithms, we have worked on Boosting, Bagging and Blending (Stacking) and compared their accuracies as well. The comparison of these algorithms has been performed using WEKA tool and listed below according to certain performance metrics. Simulation of these classification models has been performed using 10-fold cross validation. NSL-KDD based data set has been used for this simulation in WEKA.**

***Keywords*— intrusion detection, network, classification, data mining, machine learning.**

I. INTRODUCTION

In modern world, network performance has become an intriguing issue due to the ever-growing burden of network traffic. A network has to handle enormous amount of data traffic which includes malicious data as well. It is important for an organization to monitor the network flow and detect any network intrusion which is violating the policies of the organization. So an intrusion detection system needs to be developed which would be efficient enough to monitor network traffic. It is also essential for a network to be protected against possible attacks.

The systems which are used for detection of intrusion can be broadly categorized into two different types- NIDS and HIDS which stands for Network based and host based intrusion detection system respectively. NIDS are strategically placed at nodes within the network such that they can perform an analysis of the passing traffic on an entire subnet and match it with its own library of predefined attacks. On sensing of an abnormal behavior of the network or on revelation of an attack, an alert is sent to the administrator. On the contrary, HIDS runs only on individual hosts or network devices. It performs the monitoring of the inbound and outbound packets from the device and sends a warning to the administrator on spotting any suspicious packets. NIDS are of broadly two types:- anomaly based and signature based [1]. A signature based system is predefined for a particular vulnerability, so it has a reduced number of false positives, thereby offering less flexibility. Whereas, an anomaly based system is more dynamic in nature and will search for possible attacks that are out of the predefined ones, hence resulting in a greater number of false positives. It can detect attacks only without recognizing the precise type of attack.

An intrusion detection system categorizes the network traffic, for example normal and anomalous. Network traffic is considered to be anomalous when the behavior of network traffic deviates from the standard network traffic pattern. The effectiveness of the NIDS depends on the classification algorithm being used. The accuracy and time consumption of the algorithm are important parameters in the selection process of an algorithm.

Data mining approach is used for classifying the network traffic into the two above mentioned categories [2]. It involves extraction of huge volumes of data and exploring them for further analysis. Machine learning techniques are also used to build a model from sample inputs and use the model for decision making and predictions. The accuracy and speed of such techniques need to be analyzed before

978-1-4799-9855-5/15/$31.00 ©2015 IEEE

they can be deployed for highly sensitive applications like network intrusion detection.

In this paper, NSL-KDD dataset [3] has been considered for analysis. NSL-KDD compatible classification algorithms have been analyzed using WEKA tool and their performance has been evaluated in this paper. The performance metrics which have been used are ROC area, sensitivity, specificity, precision, accuracy, Kappa, mean absolute error, F1 score, FPR, NPV, FDR and training time.

In section II, WEKA tools and its classification algorithms have been discussed in detail. The performance metrics have been analyzed in section III. Dataset description has been explained in section IV while results along with further analysis and conclusion have been portrayed in section V and section VI respectively.

II. WEKA TOOLS ALONG WITH DIFFERENT ALGORITHMS

A tool which is used for both Data mining and Machine Learning is WEKA. It was first implemented by The University of Waikato, New Zealand, in 1997 [4]. It is a collection of an enormous number of Machine Learning and Data Mining algorithms. One drawback of this software is that it supports data files only written in ARFF (attribute relation file format) and CSV (comma separated values) format. Initially, it was written in C but later on it was rewritten in JAVA language. It comprises of a GUI interface for interaction with the data files. It possesses 49 data pre-processing tools, 15 attribute evaluators, 76 classification algorithms and 10 search algorithms for the purpose of feature selection. It comprises of three different types of graphical user interfaces (GUI's):- "The Explorer", "The Experimenter", and "The Knowledge Flow". WEKA provides the opportunity for the development of any new Machine Learning algorithm. It contains visualization tools and a set of panels to execute the desired tasks.

Classification algorithms or classifiers are used to basically sort out the network traffic into normal and anomaly categories. The objective behind classification techniques is to achieve high accuracy and precision and to classify the objects. Classifiers can be broadly classified into eight types in WEKA, where various machine learning algorithms reside in each category. The classifiers have been described in brief below.

Bayes Classifier— It originates from previous works in pattern recognition and is linked to the family of probabilistic Graphical Models. For each class, a probabilistic summary is stored. The conditional probability of each attribute and the probability of the class are stored in this summary. The graphical models are used to display knowledge about domains which are uncertain in nature. In the graphs [5], nodes depict random variables and the edges which connect corresponding random variable nodes are assigned weights which represent probabilistic dependencies. On encountering a new instance, the algorithm just creates an update of the probabilities stored along with the specific class [6]. The sequence of training instances and the existence of classification errors do not have any role in this process. Thus basically it has to predict the class depending on the value of the members of the class. This category consists of 13 classifiers, but only 3 of those are compatible with our chosen dataset.

Function classifier— It deploys the concept of regression and neural network. Input data is mapped to the output. It employs the iterative parameter estimation scheme. Overall there are 18 classifiers under this category, out of which only 2 are compatible with our dataset.

Lazy classifier— This requires the storage of the entire training instances and supports inclusion of new data only after classification time. The prime advantage of this classification scheme is the local approximation of the target function [5]. For each query to the system, the objective function is approximated locally thereby enabling lazy learning systems to solve multiple problems concurrently. But, the disadvantage is that it consumes a huge amount of storage space to store the entire training instance at once. It is time consuming also. Five classifiers are available under this category, but only two of those are compatible with our data set.

Meta Classifier— These sets of classifiers are essential to find the optimal set of attributes which can be used for training the base classifier [7]. New adaptive machine learning algorithms can be constructed using these classifiers and those new models can be further used for making predictions. 26 classifiers reside in this category, out of which 21 of them are compatible with our dataset.

Mi Classifier— Mi represents Multi-Instance Classifiers [8]. It consists of multiple instances in an example, but observation of one class is possible only for all the instances. Thus, it is an improvised learning technique. There are 12 classifiers under this category but all of them are incompatible with our dataset.

Misc Classifier— This category consists of different types of classifiers. Only two of them, out of three are compatible with our dataset.

Rules Classifier— Some kind of association rule is used for correct prediction of class among all the attributes. The amount of correct prediction is defined by the term coverage and is expressed in percentage or accuracy form. The association rules are mutually exclusive. More than one conclusion can be predicted. A total of 11 classifiers are available under this category, but 8 are compatible with our dataset.

Trees— It is a technique in which a flow-like tree diagram is generated where each node depicts a test on the attribute value and the outcome of each test is represented by each branch. The model generated is both predictive and descriptive in nature. The predicted classes are being signified by the tree leaves. There are 16 classifiers available out of which 10 are deemed to be compatible with this dataset.

The classification algorithms which we have executed in this paper are discussed in detail below.

BayesNet— It is a widely used technique which works on the basic Bayes theorem and forms a Bayesian network [9] after calculating conditional probability on each node. It is a graphical model which is probabilistic in nature and portrays a group of arbitrary variables along with their conditional dependencies through a directed acyclic graph.

Logistic— This technique employs regression to predict the probability of an outcome which can have only two values. One or several predictors are used to make the prediction. Logistic regression produces a logistic curve that is confined to values between 0 and 1. The curve is constructed using the natural logarithm of the odds of the target variable and not the probability.

IBK— It stands for instance based knowledge representation of the training instances [10] and does not conclude or predict a rule set or a decision tree. After a set of training instances has been stored, the memory is searched for the new training instance. So it is time consuming and requires space also.

JRip— This technique executes a proposed rule learner and cumulative error pruning method to reduce error. It is based on association rules with reduced error pruning techniques, thus making it an effective technique.

PART— It uses a divide and conquer approach to construct a C4.5 decision tree partially for each iteration specifying the optimal rule association. Using an entropic distance measure technique, it performs instance based learning.

J48— It is an enhanced version of C 4.5 which revolves on the ID3 algorithm with some extra functionalities to resolve issues that ID3 was incompetent in [11]. However, this technique is time and space consuming. Initially, it builds a tree using the divide and conquer algorithm and then applies heuristic criteria. The rules according to which the tree is generated are precise and intuitive.

Random Forest— This classification algorithm uses ensemble methods to obtain better predictive performance. It produces output in the form of individual trees and is based upon the decision tree algorithm. It is considered to be a highly accurate classifier and can handle multiple variables.

Random Tree— It generates a tree by randomly selecting branches from a possible set of trees. The trees are distributed in a uniform way so chances of getting sampled are equiprobable.

REPTree— It is a rapid decision tree learner. A decision tree is constructed with the help of information obtained on gain/ variance [12] and uses reduced error pruning techniques to reduce the error. The sorting of the values for numeric attributes is done exactly once by the algorithm and then it deals with the missing ones by splitting the subsequent instances into pieces.

In this paper, we have used certain machine learning algorithms. They construct a model from example inputs and use it for decision and prediction making. The algorithms which we have used are AdaBoost, Stacking and Bagging. They have been discussed in details below.

AdaBoost— It actually stands for adaptive boosting algorithm [13]. It is an ensemble based method initiating with a base classifier which is built on the training data. Then a second classifier is established behind it to concentrate on the instances in the training data which were obtained wrongly from the base classifier. Addition of further classifiers continues till a specified limit is reached in number of models or accuracy. Boosting uses the J48 algorithm for the base classifier. Boosting helps in enhancement of accuracy of any machine learning algorithm.

Bagging— Bagging or bootstrapping aggregating [14] is an ensemble method which creates different samples of the training dataset and for each sample a classifier is created. Finally, the results of these various classifiers are combined using average or majority voting. Since each sample of the training set is different from the other, so each trained classifier is given a different focus and outlook to the problem. It also uses the J48 as the base classifier. Bagging reduces variance and helps in avoidance of over fitting. It improves the accuracy and stability of machine learning algorithms.

Stacking— Stacking or blending is another ensemble method where preparation of different multiple algorithms takes place on the training data. A Meta classifier is prepared which learns to take predictions of each classifier and make precise predictions on data which cannot be seen. J48 and IBk are the two classifiers which are used and the Meta classifier used is Logistic Regression. Blending is basically the combination of different types of algorithms. So we are using J48, under tree section, and IBk (k-nearest neighbor), under lazy section, which are entirely different sets of algorithms. They can have a different perspective on the problem and can make varying useful predictions. Logistic regression is a simple and reliable method to learn how the predictions from the above two methods can be combined. It produces binary outputs, so it is suitable for binary classification problems.

III. Performance Measures

The performance of the classifiers can be compared according to certain metrics like accuracy, specificity, sensitivity, training time etc. A confusion matrix forms the basis from which different parameters can be calculated. The number of instances accurately or inaccurately predicted by a classification model can be tabulated in the form of a confusion matrix. The confusion matrix is generally represented by 4 values which are TP, FN, FP and TN [7] as shown in Table I. The parameters are discussed in brief below.

True positive (TP)— It indicates the instances which are predicted as normal correctly.

False negative (FN)— It indicates wrong prediction i.e. it detects instances which are attacks in reality, as normal.

False positive (FP)— It gives a hint of the number of detected attacks which are normal in reality.

True negative (TN)— It indicates instances which are correctly detected as an attack.

Table I

	Predicted		
Actual		*Normal*	*Anomaly*
	Normal	TP	FN
	Anomaly	FP	TN

ROC (Receiver operating characteristics)— In order to design the curve between true positive rate (TPR) and false positive rate (FPR), this term is required. The area under the curve is termed as AUC, which gives the value of ROC. The greater the area the curve occupies, greater will be the value of ROC.

Sensitivity— It is also known as true positive rate and gives an indication of the actual positives which are correctly identified. Thus, it gives the likelihood that the algorithm can foretell positive instances correctly [1].

Sensitivity=TP / (TP+FN)

Specificity (SPC)— It is also known as true negative rate and gives a measure of the actual negatives which are identified correctly. Thus it gives the likelihood that the algorithm can foretell negative instances correctly.

Specificity=TN / (FP+TN)

Precision— It estimates the probability of a positive prediction being correct.

Precision=TP / (TP+FP)

Accuracy— The number of correct predictions when expressed in percentage terms indicates the accuracy. It can be calculated from the confusion matrix by the formula:

Accuracy= (TP+TN) / (TP+TN+FP+FN)

Kappa— It is used to check the amount of reliability of the classification algorithm on the dataset. It is represented by 2 values-0 and 1. 0 indicates total disagreement and 1 indicates full agreement.

Mean absolute error (MAE)— This error should be minimum for an algorithm to be the best in performance. It is the mean of the overall error which a classification algorithm makes.

F1 score— It is defined as the harmonic mean of sensitivity and precision. The test performance can be evaluated by means of this performance metric.

F=2*TP / (2TP+FP+FN)

False positive rate (FPR)— It indicates the possibility of an algorithm to predict instances as attacks which are actually normal.

FPR=FP / (TN+FP) =1-SPC

False discovery rate (FDR)— It specifies the possibility of a positive prediction made being incorrect.

FDR=FP/ (FP+TP) =1-PPV

Negative predictive rate (NPV)— It indicates the possibility of an algorithm to correctly detect instances as attack.

NPV=TN / (TN+FN)

Training time— It is the time the classifier consumes to build the model on the dataset. It is usually measured in seconds. The lesser the value of this parameter, the better will be the classifier.

IV. Description of the Data Set

The dataset used for performance evaluation in our paper has been incorporated from NSL-KDD dataset. It classifies the network traffic into two categories-normal and anomaly. Both the training and testing dataset are in ARFF file format. The training set comprises of 42 attributes and 1166 instances while the testing set comprises of 42 attributes and 7456 instances. This dataset has certain advantages over the original KDD data set, so we have chosen the above mentioned dataset.

V. Result Analysis

A 10 fold cross validation assumption has been made for testing and evaluation of the results. In 10 fold cross validation process, the full data set is split into 10 subsets. The NSL-KDD data set is taken as testing set with 10 fold cross validation. Table II summarizes the performance metrics of all the classifiers i.e. BayesNet, Logistic, IBk, JRip, PART, J48, RandomForest, RandomTree and REPTree. Fig. 1 shows the comparative analysis of various classifiers in terms of sensitivity, specificity and accuracy. From the graph, we can observe that the sensitivity is highest and same for IBk and RandomForest ie 88.68% and lowest for REPTree i.e. 79.78%. The specificity is maximum for BayesNet i.e. 96.18% and minimum for Logistic i.e. 82.2%. The accuracy is maximum for RandomForest i.e. 91.523% and minimum for Logistics i.e. 84.96%. Fig. 2 shows the comparative analysis of various classifiers in terms of Kappa, F1 score and Negative Predictive Rate (NPV). The Kappa is highest for RandomForest i.e. 0.8306 and lowest for Logistic i.e. 0.6975. The F1 score is highest for RandomForest i.e. 0.9175 and lowest for Logistic and REPTree i.e. 0.86. The NPV is maximum for RandomForest i.e. 0.8804 and minimum for REPTree i.e. 0.8027. Fig. 3 analyses the study between precision and area under ROC. The precision is high for BayesNet having a value of 0.9623 and low for Logistic having value of 0.848. The area under ROC is highest for BayesNet with a value of 0.976 and low for Logistic with a value of 0.806. Fig. 4 shows the comparative study of various classifiers in terms of MAE, FPR and FDR. The MAE is highest for REPTree with a value of 0.1627 and lowest for RandomTree and IBk having similar value of 0.0932. The FPR is highest for Logistic and lowest for BayesNet with a value of 0.178 and 0.038 respectively. The FDR is maximum and minimum in case of Logistic and BayesNet having a value of 0.152 and 0.0377 respectively. Fig. 5 shows the time graph of various classification algorithms. The highest time is taken by Logistic consuming a time of 0.71 seconds and lowest time is taken by IBk and RandomTree consuming 0.01 seconds only.

According to the above analysis, Logistic is the worst classifier as its value is maximum for FPR, FDR, and time and minimum for specificity, accuracy, kappa, F1 score, and precision. The good classifiers noted from the above results are BayesNet and RandomForests.

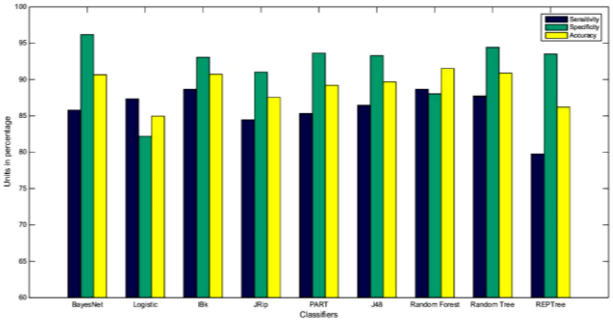

Fig. 1. Comparison of Classifiers Based on Sensitivity, Specificity and Accuracy.

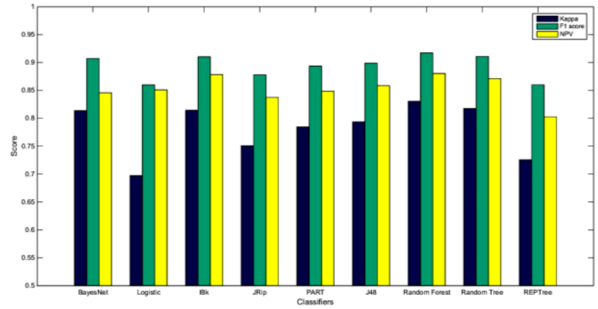

Fig. 2. Comparison of Classifiers Based on Kappa,F1 Score and NPV

Table II

Classifiers	*ROC curve*	*Sensitivity (%)*	*Specificity (%)*	*Precision*	*Accuracy (%)*	*Kappa*	*Minimum Absolute Error*	*F1 score*	*FPR*	*NPV*	*FDR*	*Training Time (seconds)*
BayesNet	0.976	85.8	96.18	0.962	90.66	0.8139	0.1002	0.907	0.038	0.846	0.037	0.02
Logistic	0.806	87.34	82.2	0.848	84.9651	0.6975	0.1542	0.860	0.178	0.851	0.152	0.71
IBk	0.909	88.68	93.06	0.936	90.7323	0.8145	0.0934	0.910	0.069	0.878	0.064	0.01
JRip	0.896	84.47	91.02	0.915	87.54	0.7511	0.1294	0.878	0.089	0.837	0.085	0.38
PART	0.933	85.32	93.63	0.938	89.2167	0.7848	0.1095	0.893	0.064	0.849	0.062	0.23
J48	0.91	86.48	93.29	0.936	89.6727	0.7937	0.1096	0.899	0.067	0.858	0.064	0.18
Random Forest	0.984	88.68	88.04	0.951	91.5236	0.8306	0.1135	0.917	0.120	0.880	0.049	0.49
RandomTree	0.912	87.74	94.43	0.947	90.8798	0.8178	0.0932	0.911	0.056	0.871	0.053	0.01
REPTree	0.903	79.78	93.52	0.933	86.2124	0.7258	0.1627	0.860	0.065	0.803	0.067	0.56

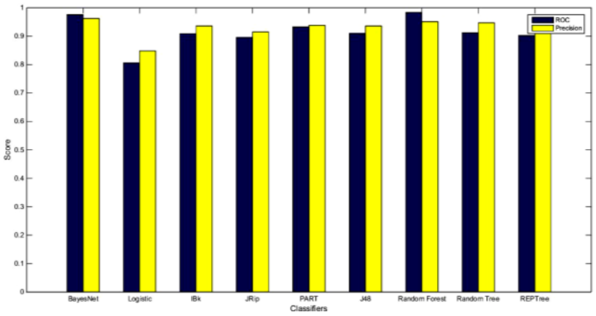

Fig. 3. Comparison of Classifiers Based on Area under ROC and Precision

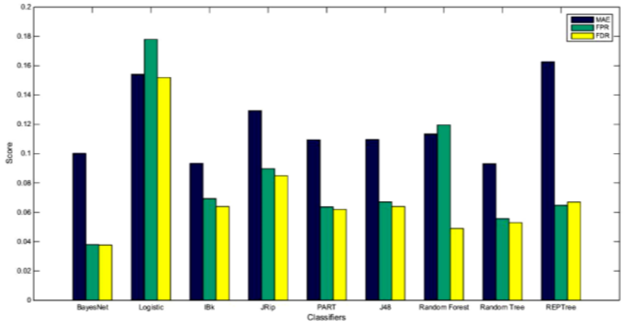

Fig. 4. Comparison of Classifiers Based on MAE, FPR and FDR.

The performance of Boosting (AdaBoostM1), Bagging and Blending (Stacking) are compared in terms of accuracy, true positive value (TP), true negative value (TN), false positive value (FP) and false negative value (FN). This comparison is portrayed in Fig. 6. '>' symbolizes the win of one algorithm over the other,' <' represents the loss and' >-< 'signifies the difference between win and loss. AdaBoostM1 wins by 3 over the other machine learning algorithms. Stacking and bagging both lost by a margin of 1, hence we obtained a difference of -1. TPR is highest for AdaBoostM1 having a win factor of 2, marginal for Bagging and lowest for Stacking. The TNR result is exactly same as that of TPR. The FPR is lowest for AdaBoostM1 with a loss of 2 and Bagging has win of 1 over Stacking. The FNR result is matching with the result of FPR. It signifies that AdaBoostM1 is the best machine learning algorithm over our data set.

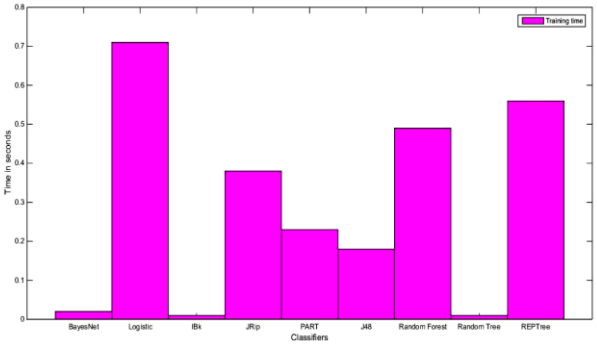

Fig. 5. Comparison of Classifiers Based on Training Time.

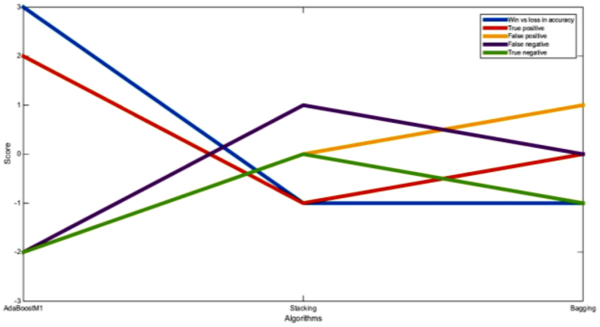

Fig. 6. Comparison of Machine Learning Algorithms Based on Accuracy, TPR, FPR, FNR and TNR.

VI. CONCLUSION

In our paper, we have put forth improved machine learning algorithms necessary for proper detection of network intrusion. We have also compared the performance of various classifiers in WEKA and concluded that RandomForest and BayesNet are suitable for this purpose. The machine learning algorithms have also been compared and it can be deduced that Boosting is the best algorithm. These improvised algorithms can be used to devise efficient network intrusion detection devices which can be used for security purposes in an organization.

REFERENCES

[1] Himadri Chauhan, Vipin Kumar, Sumit Pundir and Emmanuel S.Pilli,"A Comparative Study of Classification Techniques for Intrusion Detection", *IEEE International Symposium on Computational and Business Intelligence,2013*.

[2] Tanya Garg, "Analysis of Various Features Selection Techniques for Network Intrusion Detection Dataset in WEKA",*CT International Journal of Information & Communication Technology,2014,*Vol 2,Issue 1.

[3] NSL-KDD dataset: http://nsl.cs.unb.ca/NSL-KDD/

[4] G.Jenitha and V.Vennila, "Comparing the Partitional and Density Based Clustering Algorithms by Using Weka Tool", *2nd International Conference on Current Trends in Engineering and Technology,*2014,pp: 328-331.

[5] Ms S.Vijayarani, Ms M.Muthulakshmi,"Comparative Analysis of Bayes and Lazy Classification Algorithms", *International Journal of Advanced Research in Computer and Communication Engineering*, 2013, Vol.2, Issue 8, August 2013.

[6] Pat Langley, Wayne Iba and Kevin Thompson, "An Analysis of Bayesian Classifiers*", Tenth National Conference on Artificial Intelligence*, 1992.

[7] Tanya Garg and Surinder Singh Khurana, "Comparison of Classification Techniques for Intrusion Detection Dataset using WEKA", *IEEE International Conference on Recent Advances and Innovations in Engineering (ICRAIE-2014)*, May 09-11, 2014, Jaipur, India.

[8] Multi-Instance Classifiers at http://weka.wikispaces.comlMultiinstance+classification

[9] Harsimran Kaur, "Algorithm used in Intrusion Detection System: a Review", *International Journal of Innovative Research in Computer and Communication Engineering*, 2014, Vol.2, Issue 5, May 2014.

[10] G. MeeraGandhi, "Machine Learning Approach for Attack Prediction and Classification using Supervised Learning Algorithms", *International Journal of Computer Science & Communication,* vol. 1, no. 2, 2010, pp. 247-250.

[11] Mrutyunjaya Panda and Manas Ranjan Patra,"A Comparative Study of Data Mining Algorithms for Network Intrusion Detection", *First IEEE International Conference on Emerging Trends in Engineering and Technology*, 2008.

[12] Kailas Shivshankar Elekar and Prof. M.M. Waghmare, "Study of Tree Base Data Mining Algorithms for Network Intrusion Detection*", International Journal on Recent and Innovation Trends in Computing and Communication*, Volume: 2 Issue: 10, October 2014.

[13] Mrutyunjaya Panda and Manas Ranjan Patra," Ensembling Rule Based Classifiers For Detecting Network Intrusions", *IEEE International Conference on Advances in Recent Technologies in Communication and Computing*, 2009.

[14] Youqin Pan and Zaiyong Tang," Ensemble methods in bank direct marketing", *IEEE International Conference,* 2014.