Volume: 07 Issue: 04 Pages: 2828-2834 (2016) ISSN: 0975-0290

Decision Tree Based Algorithm for Intrusion Detection

Kajal Rai
Research Scholar, Department of Computer Science and Applications, Panjab University, Chandigarh, India
Email: kajalrai@pu.ac.in
M. Syamala Devi
Professor, Department of Computer Science and Applications, Panjab University, Chandigarh, India
Email: syamala@pu.ac.in
Ajay Guleria
System Manager, Computer Center, Panjab University, Chandigarh, India
Email: ag@pu.ac.in

----------**ABSTRACT**----------

An Intrusion Detection System (IDS) is a defense measure that supervises activities of the computer network and reports the malicious activities to the network administrator. Intruders do many attempts to gain access to the network and try to harm the organization's data. Thus the security is the most important aspect for any type of organization. Due to these reasons, intrusion detection has been an important research issue. An IDS can be broadly classified as Signature based IDS and Anomaly based IDS. In our proposed work, the decision tree algorithm is developed based on C4.5 decision tree approach. Feature selection and split value are important issues for constructing a decision tree. In this paper, the algorithm is designed to address these two issues. The most relevant features are selected using information gain and the split value is selected in such a way that makes the classifier unbiased towards most frequent values. Experimentation is performed on NSL-KDD (Network Security Laboratory Knowledge Discovery and Data Mining) dataset based on number of features. The time taken by the classifier to construct the model and the accuracy achieved is analyzed. It is concluded that the proposed Decision Tree Split (DTS) algorithm can be used for signature based intrusion detection.

Keywords **- Decision Tree, Information Gain, Gain Ratio, NSL-KDD, Signature-based IDS**

Date of Submission: Dec 22, 2015 Date of Acceptance: Dec 28, 2015

1. INTRODUCTION

An Intrusion Detection System (IDS) is a system that monitors network to check harmful activities in the network and reports events that does not meet the security criteria to the network administrator. IDSs are categorized as Signature based and Anomaly based. Signature or Misuse based IDS uses various techniques to locate the similarity among system behavior and previously known attacks stored in the signature database. Anomaly based IDS detects activities in a network which deviates from normal behaviors stored in system profiles database. There are various classifiers that are applicable to misuse based detection. Some are tree based such as decision tree [1], and random forest [2], whereas some are rule based such as oneR [3], while some are function based such as SVM (Support Vector Machine) [4]. In this paper, the decision tree classifier is used to classify input data as normal or anomalous.

A Decision Tree is a tree-like graph consisting of internal nodes which represent a test on an attribute and branches which denote the outcome of the test and leaf nodes which signify a class label. The classification rules are formed by the path selected from the root node to the leaf. To divide each input data, first the root node is chosen as it is the most prominent attribute to separate the data. The tree is constructed by identifying attributes and their associated values which will be used to analyze the input data at each intermediate node of the tree. After the tree is formed, it can prefigure newly coming data by traversing, starting from a root node to the leaf node visiting all the internal nodes in the path depending upon the test conditions of the attributes at each node [5]. The main issue in constructing decision tree is, which value is chosen for splitting the node of the tree. This issue is taken care in section 3.

Decision trees can analyze data and identify significant characteristics in the network that indicate malicious activities. It can add value to many real-time security systems by analyzing large set of intrusion detection data. It can recognize trends and patterns that support further investigation, the development of attack signatures, and other activities of monitoring. The main advantage of using decision trees instead of other classification techniques is that they provide a rich set of rules that are easy to understand, and can be effortlessly integrated with real-time technologies [6].

NSL-KDD is the latest dataset for intrusion detection. This dataset consists of 41 features, however not all the

Volume: 07 Issue: 04 Pages: 2828-2834 (2016) ISSN: 0975-0290

features are of equal importance. If complete feature set is used for classification input data, then the classifier will take more time to detect intrusion and they can also affect the accuracy of the classifier. That's why before performing any classification, we need to reduce this set by applying some feature selection method. Feature selection is done to remove irrelevant and redundant features. In the literature, there are various feature selection methods such as information gain [7], PCA (Principle Component Analysis), and GA (Genetic Algorithm). For classification of network data several classifiers are available such as KNN (k-nearest neighbor), SVM, ANN (Artificial Neural Network), and decision tree. C4.5 builds decision tree by using the notion of information entropy from a set of training data. At each node of the tree, the algorithm picks out an attribute which most efficiently divides the set of given data into smaller subsets associated with any class in the given training set. The dividing factor here is the gain ratio. The attribute with the highest gain ratio is selected to do the judgment [8].

The rest of the paper is structured as follows: Section 2 gives a brief related work of intrusion detection based on feature selection and classifiers. In section 3 provides the proposed algorithm DTS for developing decision tree. In section 4, the experimental results using NSL-KDD dataset is shown. Section 5, includes conclusion and scope for future work.

2. RELATED WORK

Literature review has been done including latest papers that perform training and testing of the system on NSL-KDD dataset. The review is performed based on feature selection and classification.

2.1 REVIEW BASED ON FEATURE SELECTION

Gaikwad and Thool [9] have applied Genetic Algorithm (GA) on NSL-KDD data set to select relevant features. The GA selects 15 features out of 41 from the available data set. These 15 features gives 79% accuracy on test data with decision tree as a classifier and it takes 176 seconds to build the model.
Bajaj and Arora [10] discuss various feature selection methods such as information gain, gain ratio and correlation based feature selection. In their paper, they select 33 features out of 41 for classification and the results for various classifiers are compared. The Simple Cart algorithm gives the highest accuracy 66.77% whereas the classification result of C4.5 decision tree is 65.65% only. Alazab et al. [11] also select features using information gain and decision tree to detect both the old and the new attacks.

Thaseen and Kumar [12] used two useful methods for feature selection, namely, Correlation based Feature Selection (CFS) and Consistency-based Feature Selection (CONS). In this paper, 8 features are selected using CFS and the classification is done using Naïve Bayes, C4.5 decision tree, and AD (Alternating Decision) tree and their results have been compared. CONS selects 10 useful features out of 41 and then the classification using various techniques such as Random Forest and Random Tree has been analyzed.

Revathi, and Malathi[13] select 15 features using CFS and test using various classifiers such as Random Forest, C4.5 decision tree, SVM, CART and Naïve Bayes. The results of above mentioned classifiers have been compared and the outcome shows that Random Forest gives highest accuracy in detecting attacks.
Jyothsna and Prasad [14] evaluates and compares the performance of IDSs for different feature selection techniques such as information gain, gain ratio, Optimized Least Significant Particle based Quantitative Particle Swarm Optimization (OLSP-QPSO), and Principle Component Analysis (PCA). Results show that the OLSP-QPSO technique has more number of attribute reduction and low false alarm rate with high detection rate when compared with the remaining feature selection techniques.

In [15], the authors proposed hybrid KNN and Neural Network based multilevel classification model. In this model, KNN was used as a classifier for anomaly detection with two classes, namely, 'normal' and 'abnormal'. After that a neural network was used to detect a specific type of attack in 'abnormal' class. For experiments, the NSL-KDD dataset was used. First, all the features of the dataset were used for classification. Then classification is performed on 25 selected features. Selection has been done by Rough Set Theory and Information Gain separately. In this classification model, Information Gain with 25 features of the NSL-KDD dataset produced better results as compared to 25 features with Rough Set Theory as well as 41 features of NSL-KDD dataset.

2.2 REVIEW BASED ON CLASSIFIERS

Elekar, and Waghmare [16] implement different classifiers such as C4.5 decision tree, Random Forest, Hoeffding Tree and Random Tree for intrusion detection and compare the result using WEKA. The results show that the Hoeffding Tree gives the best result among the various classifiers for detecting attacks on the test data.
Aggarwal and Sharma [1] evaluate ten classification algorithms such as Random Forest, C4.5, Naïve Bayes, and Decision Table. Then they simulate these classification algorithms in WEKA with KDD'99 dataset. These ten classifiers are analyzed according to metrics such as accuracy, precision, and F-score. Random Tree shows the best results overall while the algorithms that have high detection rate and low false alarm rate were C4.5 and Random Forest.
In [17] the authors show how useful is the NSL-KDD for various intrusion detection models. For

Volume: 07 Issue: 04 Pages: 2828-2834 (2016) ISSN: 0975-0290

dimensionality reduction, PCA technique was used in this paper. Six different algorithms, namely, ID3, Bayes Net, J48, CART, SVM, and Naïve Bayes were used for the experimentation with and without feature reduction, and from the results it was clear that SVM gives the highest accuracy for the above two cases.

In [18], the authors designed a multi-layer hybrid machine learning IDS. PCA was used for attribute selection and only 22 features were selected in the first layer of the IDS. GA was used in the next layer for generating detectors, which can distinguish between normal and abnormal behavior. In the third layer, classification was done using several classifiers. Results demonstrate that the Naive Bayes has good accuracy for two types of attacks, namely, User-to-Root (U2R) and Remote-to-Local (R2L) attacks however the decision tree gives higher accuracy up to 82% for Denial-of-Service attacks and 65% of probe attacks.

The system proposed by Raeeyat et al. in [19] consists of 4 modules namely, Data pre-processing module, Misuse detection module, anomaly detection module and Evaluation and comparison module. Data were pre-processed before passing to the other modules by data pre-processing module. In the misuse detection module, pre-processed data is given to PCA to take out important features. After that the data were examined using Adaboost algorithm based on C4.5 decision tree to know whether it is a normal packet or an intrusion. Then the outcome of decision tree is passed on to the next module for evaluation and comparison. When the data is sent to misuse detection module it is simultaneously sent to anomaly detection module also. The correlation among features was also found out by the correlation unit by using Pearson Correlation. Data correlation graph is used to show deviation of behavior from the normal behavior. Then, the evaluation and comparison module determine whether the instance is an intrusion or not by taking the output from misuse and anomaly detection module and if both the module shows that it is an intrusion then only that instance is considered as an intrusion.

In [20], a hybrid IDS is proposed. Random Forest is used for classification in misuse detection to build patterns of intrusion from a training dataset. Weighted k-means clustering is used in anomaly detection. Due to less correlation between clusters, there is high false alarm rate as the number of clusters increases for detecting larger number of attacks.

3. DECISION TREE SPLIT (DTS) ALGORITHM

Decision Tree Split (DTS) algorithm is based on C4.5 decision tree algorithm [11] [21]. The main issue in constructing decision tree is the split value of a node. The proposed algorithm gives a novel approach in selecting the split value. The steps of the algorithm are as follows:

1. If all the given training examples belong to the same class, then a leaf node is created for the decision tree by choosing that class.

2. For every feature 'a', calculate the gain ratio by dividing the information gain of an attribute with splitting value of that attribute. The formula for gain ratio is $\text{GainRatio(a)} = \frac{\text{IG(a)}}{\text{Split(a)}}$
   where, S is the set of all the examples in the given training set.

3. Information gain of an attribute is computed as $\text{IG(a)} = \text{Ent(S)} - \sum_{\text{a\_val} \in \text{values(a)}} \frac{|\text{S\_a}|}{|\text{a}|} * \text{Ent(S\_a)}$
   where, S_a is the subset of S, values (a) is the set of all possible values of attribute ‘a’ and |a| is the total number of values in attribute ‘a’

4. Entropy can be calculated as $\text{Ent(S)} = -\sum_{j=1}^{\text{num\_class}} \frac{\text{freq(Lj,S)}}{|\text{S}|} * \log 2 (\frac{\text{freq(Lj,S)}}{|\text{S}|})$
   where, $L = L_1, L_2, \ldots, L_n$ is the set of classes, andnum_class is the number of distinct classes. For our consideration num_class has only two values, namely, ‘normal’, and ‘anomaly’.

5. Split value of an attribute is chosen by taking the average of all the values in the domain at that particular attribute. It can be formulated as $\text{Split(a)} = \frac{\left(\sum_{i=1}^{m} (a\_val)i \right)}{m}$
   where m is the number of values of an attribute ‘a’.

6. Find the attribute with the highest gain ratio. Suppose, the highest gain ratio is for the attribute 'a_best'.

7. Construct a decision node that divides the dataset on the attribute 'a_best'.

8. Repeat steps from 1 to 4 on each subsets produced by dividing the set on attribute 'a_best' and insert those nodes as descendant of parent node.

C4.5 algorithm uses the following function for calculating the split value of an attribute

$$\text{Split(a)} = -\sum_{a\_val \in values(a)} \frac{|S\_a|}{|a|} * \log 2 \left(\frac{|S\_a|}{|a|} \right)$$

Volume: 07 Issue: 04 Pages: 2828-2834 (2016) ISSN: 0975-0290

3.1 SIGNIFICANCE OF THE PROPOSED ALGORITHM

To select the split value, C4.5 algorithm first sorts all the values of an attribute. Then from these sorted values, say, P_i, P_{i+1}, … P_n , the gain ratio of all the values is calculated by choosing the lower value of P_i and P_{i+1} as threshold value and then calculate split value by using above mentioned formula. The value which gives the highest gain ratio is chosen as the split value for that particular node. Instead of using all these calculations which makes technique more complex and difficult to understand, we use a simple and effective approach. In our approach, there is no need to sort the attribute values to calculate the split value. We calculate the split value by taking the average of the values in the domain of a particular attribute at each node. It gives uniform weightage to all the values in the domain, making the classifier totally unbiased towards the most frequent values in the domain of an attribute. Sometimes, gain ratio may choose an attribute as a split attribute just because its intrinsic information is very low. This limitation can be overcome by considering only those attributes that have greater value of information gain than average information gain.

4. IMPLEMENTATION AND TESTING

The DTS algorithm is implemented on a 64-bit Windows 8.1 operating system, with 8 GB of RAM and a Pentium(R) processor with CPU speed of 2.20GHz using tools WEKA and MATLAB. The proposed algorithm is compared with the existing ones such as Classification and Regression Tree (CART), C4.5, and AD Tree.

The experiments were done for performance comparison of different tree based classifiers and the DTS algorithm. The analysis is done based on different parameters such as how many seconds the classifier takes to construct the model, false positive rate, true positive rate, and accuracy. True Positive (TP) represents the examples that are correctly predicted as normal. True Negative (TN) shows the instances which are correctly predicted as an attack. False Positive (FP) identifies the instances which are predicted as attack while they are not. False Negative (FN) represents the cases which are prefigured as normal while they are attack in reality. Accuracy can be defined as the number of correct predictions. It can be computed as

$$\text{Accuracy} = \frac{(TP + TN)}{(TP + TN + FP + FN)}$$

The Receiver Operating Characteristic (ROC) curve is also plotted for various techniques. ROC plots the curve between true positive rate (TPR) and false positive rate (FPR) of an algorithm. TPR and FPR are computed as

$$TPR = \frac{TP}{(TP + FN)} \text{ and } FPR = \frac{FP}{(FP + TN)}$$

4.1 DATASET

In this section, the efficiency of the proposed technique is evaluated cautiously by experimentations with the NSL-KDD data set, which is a revised version of KDD'99 data set. The reason for using NSL-KDD dataset for our experiments is that the KDD'99 data set has a large number of redundant records in the training and testing data set. For binary classification, the NSL-KDD classifies the network traffic into two classes, namely, normal and anomaly. The experiments were performed on full training data set having 125973 records and test data set having 22544 records.

First, we compute information gain of all the attributes of the data set. We found that there are 16 attributes whose information gain is greater than the average information gain. That's why in the preprocess step, we can choose 16 or less than 16 attributes for further processing based on information gain because the remaining features will not have much effect on classification of the dataset. Then, the data set with these selected attributes is passed to the algorithm for constructing, training and testing the decision tree.

4.2 RESULTS AND ANALYSIS

The performance of the proposed algorithm is compared with the performance of various techniques. The comparison of results is done based on the accuracy in detecting attacks on the test dataset of NSL-KDD. The results are taken from the literature which uses various techniques such as Self Organizing Maps (SOM), hoeffding tree, and Ripple Down Rule learner Intrusion Detection (RDRID) for training their detection model and testing. It is observed that our proposed algorithm for constructing decision tree is efficient in attack detection as shown in Fig. 1.

Various other classifiers such as CART, Naïve Bayes (NB) Tree, and AD Tree along with the proposed algorithm are tested using NSL-KDD test dataset. ROC curves of AD Tree, C4.5, CART and DTS algorithm without feature selection on test data of NSL-KDD are plotted as shown in Fig. 2. The time taken by several classifiers is also measured and bar graph is plotted in Fig. 3. It is observed from Fig. 2 and Fig. 3, that the true positive rate of DTS is better than C4.5 technique, however CART shows the best performance in terms of true positive rate. But if we compare the results in terms of delay to build the model, we can see that CART takes very high time as compared to other techniques. The results of comparison of various classifiers with different number of features are presented in Fig. 4. It can be seen from the results that with the proposed technique, instead of training with all the features we get good accuracy with even less number of features selected using information gain.

Volume: 07 Issue: 04 Pages: 2828-2834 (2016) ISSN: 0975-0290

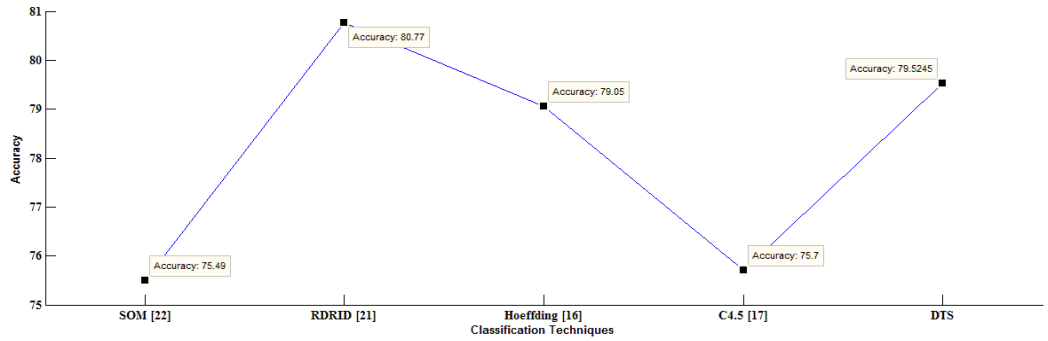

Figure1. Results of comparison of proposed algorithm with various other techniques

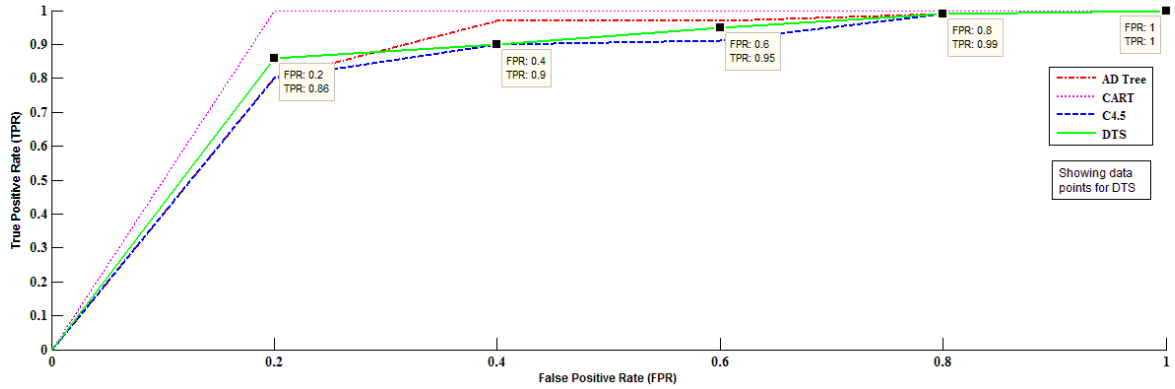

Figure 2. Comparison of ROC Curve of proposed algorithm with various other classifiers

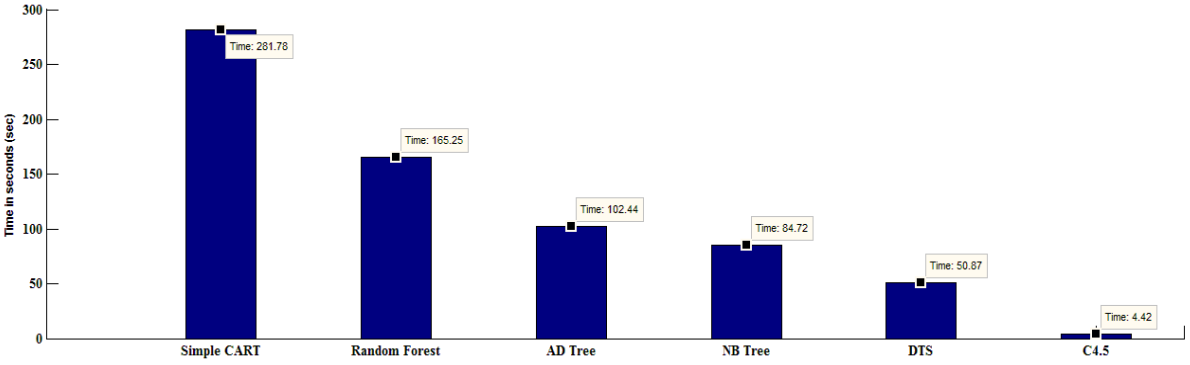

Figure 3. Time of construction of model of the proposed algorithm with various tree based classifiers

Volume: 07 Issue: 04 Pages: 2828-2834 (2016) ISSN: 0975-0290

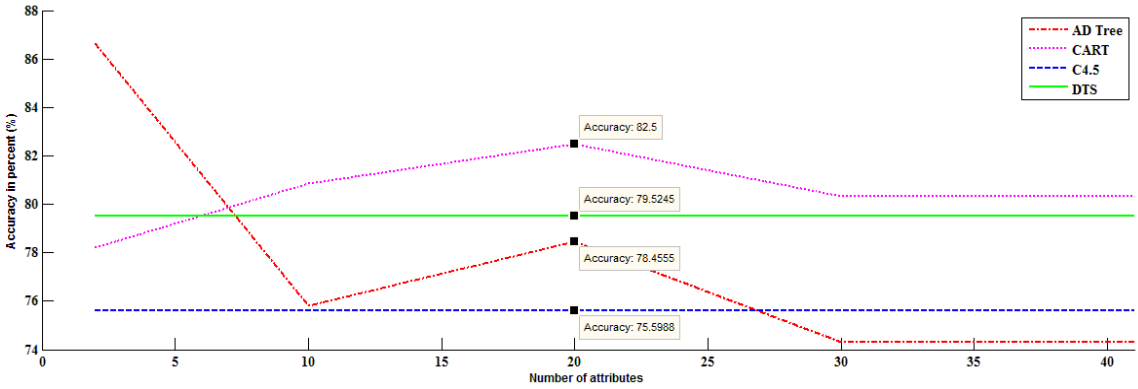

Figure 4. Comparison of accuracy of several classifiers with proposed algorithm

5. CONCLUSION AND FUTURE SCOPE OF WORK

Decision tree assists the network administrator to decide about the incoming traffic, i.e., whether the coming data is malicious or not by providing a model that separates malicious and non-malicious traffic. By modified the split value calculation by taking the average of all the values in the domain of an attribute. The algorithm provides uniform weightage to all the values in the domain. It allows taking less number of attributes and provides acceptable accuracy in reasonable account of time. From the results of the experiments, it is concluded that the proposed algorithm for signature based intrusion detection is more efficient with respect to finding attacks in the network with less number of features and it takes less time to construct the model. It is also concluded that the efficiency depends on the size of the data set and the number of features used to construct the decision tree. The formula used in DTS to calculate gain ratio can also be used in attribute selection for feature reduction. Our future scope of work is to improve the split value by using concepts such as geometric mean which also gives uniform weightage to the domain values.

REFERENCES

[1] P. Aggarwal, and S.K. Sharma, An Empirical Comparison of Classifiers to Analyze Intrusion Detection, *Proc. of Fifth International Conference an Advanced Computing and Communication Technologies*, 2015.

[2] Ho, and Tin Kam, Random Decision Forests,*Proc. of the 3rd International Conference on Document Analysis and Recognition*, Montreal, QC, 14–16 August 1995. pp. 278–282.

[3]http://www.saedsayad.com/oner.htm

[4]C. Cortes, and V. Vapnik, (1995). Support-vector networks, *Machine Learning* **20** (3): 273.

[5] P.N. Tan, M. Steinbach, V. Kumar, Introduction to Data Mining, *Pearson Addison Wesley*, 2005.

[6] J. Markey, Using Decision Tree Analysis for Intrusion Detection: A How-To Guide, *SANS Institute InfoSec Reading Room*, June, 2011.

[7] T. M. Mitchell, (1997). Machine Learning. *The Mc-Graw-Hill Companies*, Inc. ISBN 0070428077.

[8] J.R. Quinlan, C4.5: Programs for Machine Learning, *Morgan Kaufmann Publishers*, 1993.

[9] D.P. Gaikwad, and R.C. Thool, Intrusion Detection System Using Bagging with Partial Decision Tree Base Classifier,*Proc. of the 4th International Conference on Advances in Computing, Communication and Control,* 2015.

[10] K. Bajaj, and A. Arora, Improving the Intrusion Detection using Discriminative Machine Learning Approach and Improve the Time Complexity by Data Mining Feature Selection Methods, *International Journal of Computer Science, vol. 76*, Aug, 2013.

[11] A. Alazab, M. Hobbs, J.Abawajy, and M. Alazab, Using Feature Selection for Intrusion Detection System, *International Symposium on Communications and Information Technologies*, 2012.

[12] S. Thaseen, and Ch. A. Kumar, An Analysis of Supervised Tree Based Classifiers for Intrusion Detection System, *In Proc. of the 2013 International Conference on Pattern Recognition, Informatics and Mobile Engineering*, Feb, 2013.

[13] S. Revathi, and A. Malathi, A detailed Analysis on NSL-KDD Dataset Using Various Machine Learning Techniques for Intrusion Detection, *International*

Journal of Engineering research and Technology, vol 2, Issue 12, Dec, 2013.

[14] V. Jyothsna, and V.V. Prasad, A Comparative Study on Performance Evaluation of Intrusion Detection System through Feature Reduction for High Speed Networks, *Global Journal of Computer Science and Technology: E Network, Web and Security, vol. 14*, Issue 7, Version 1.0, 2014.

[15] P. Ghosh, C. Debnath, D. Metia, and Dr. R. Dutta, An Efficient Hybrid Multilevel Intrusion Detection System in Cloud Environment, *IOSR Journal of Computer Engineering (IOSR-JCE) e-ISSN: 2278-0661, p-ISSN: 2278-8727, Volume 16*, Issue 4, Ver. VII (Jul – Aug. 2014), PP 16-26.

[16] K.S. Elekar, and M.M. Waghmare, Comparison of Tree base Data Mining Algorithms for Network Intrusion Detection, *International Journal of Engineering, Education and Technology, vol 3* Issue 2, 2015.

[17] S. Mallissery, S. Kolekar, and R. Ganiga, Accuracy Analysis of Machine Learning Algorithms for intrusion Detection System using NSL-KDD Dataset, *Proc. International Conference on Future Trends in Computing and Communication -- FTCC 2013*, July 2013, Bangkok.

[18] A.S.A. Aziz, A.E. Hassanien, S. El-Ola Hanafy, M.F. Tolba, Multi-layer hybrid machine learning techniques for anomalies detection and classification approach, *13th International Conference on Hybrid Intelligent Systems (HIS)*, 2013, IEEE.

[19] A. Raeeyat, and H. Sajedi, HIDS: DC-ADT: An Effective Hybrid Intrusion Detection System based on Data Correlation and Adaboost based Decision Tree classifier, *Journal of Academic and Applied Studies, vol. 2*(12), Dec. 2012, pp.19-33.

[20] R.M. Elbasiony, E.A. Sallan, T.E. Eltobely, and M.M. Fahmy, A hybrid network intrusion detection framework based on random forests and weighted k-means, *Ain Shams Engineering Journal, vol.4,* Issue 4, Dec, 2013, pp. 753-762.

[21] D.P. Gaikwad, and R.C. Thool, Intrusion Detection System using Ripple Down Rule learner and Genetic Algorithm, *International Journal of Computer Science and Information Technologies, vol. 5,* 2014, pp. 6976-6980.

[22] L.M. Ibrahim, D.T. Basheer, and M.S. Mahmod, A comparison study for intrusion database (KDD99, NSL-KDD) based on Self Organization Map (SOM) Artificial Neural Network, *Journal of Engineering Science and Technology, vol. 8*, No. 1, 2013, pp. 107-119.