Proc. of Int. Conf. on Advances in Computer Science and Application, CSA

Identify Promising Classifiers for Each Type of Attack Class

[1]Yogesh Kumar and [2]Indu Bala
[1]Department of Computer Science & Engineering
Bhai Gurdas Institute of Engineering & Technology, Sangrur, Punjab 148001 Email: yksingla37@gmail.com
[2]Department of Computer Science & Engineering
Bhai Gurdas Institute of Engineering & Technology, Sangrur, Punjab 148001 Email: indu.bala30@gmail.com

***Abstract*—Most current offline intrusion detection systems are focused on unsupervised and supervised machine learning approaches. Due to the variety of network behaviors and the rapid development of attack fashions, it is necessary to find the best machine-learning-based intrusion detection algorithms with high attack rates. There are many Classification mechanisms used in data mining that would help with intrusion detection system such as Naive Bayes Algorithm, IBK, J48, Random forest, AttributeSelectedClassifier(ASC), ClassificationviaRegres-sion(CVR), Decision stump, REPTree, Random Tree, Filtered classifier, RandomCommitee, JRip, HoeffdingTree. This paper presents a comparison of 14 classification techniques based on the performance measures TP rate,FP rate,Precision, F-measure, ROC area. The goal of this research is to enumerate the high attack rates techniques from above fourteen analyzed algorithms under a given data set and provide a fruitful comparison result.**

***Index Terms*— TP rate, FP rate,Precision, F-measure, ROC area.**

I. INTRODUCTION

An intrusion detection system (IDS) [1] defined as an effective security technology, which can prevent, detect and possibly react to the various types of computer attacks is one of the standard components in security infrastructures. It monitors and detect target sources of activities, such as network traffic data in computer or network systems and then deploys various techniques in order to provide security services. The main purpose of IDS is to classify intrusive and nonintrusive network activities in an efficient manner. The process of intrusion detection(ID) involves the tasks like data acquisition/ collection; data Preprocessing and feature selection; model selection for data analysis; classification and result analysis. Two main Intrusion detection methods are defined as follows:

1) Anomaly detection: Anomaly detection [2]assumes that an intrusion will always reflect some deviations from normal patterns. Anomaly detection may be divided into static and dynamic anomaly detection. A static anomaly detection is based on the assumption that there is a portion of the system being monitored that does not change with time. Mostly, static detectors only audit the software portion of a system and are based on the assumption that the hardware need not be checked. The static portion of a system is that code for the system and the constant portion of data upon which the correct functioning of the system depends.

2) Misuse detection: Misuse detection [2]is based on the knowledge of system vulnerabilities and known

Grenze ID: 02.CSA.2015.4.8

attack patterns. Misuse detection is concerned with finding intruders who are attempting to break into a system by exploiting some known vulnerability. Ideally, a system security administrator should be aware of all the known vulnerabilities and eliminate them. The term intrusion
scenario is used as a description of a known kind of intrusion; it is a sequence of events that would result in an intrusion without some outside preventive intervention. An intrusion detection system continually compares recent activity to known intrusion scenarios to ensure that one or more attackers are not attempting to exploit known vulnerabilities.
This research has conducted a comparison study between a number of available data mining algorithms and tools depend-ing on their ability for classifying network attacks correctly and accurately. The rest of the paper is organized as follows: Section 2 summaries the various types of data classification techniques used. Section 3 provides a general description of the tools and software under test and dataset used. Section 4 reports experimental results and compares the results of the different algorithms. Finally, I close this paper with a summary and an outlook for some future work.

II. TECHNIQUES USED

A. Naive Bayes

The naive Bayes classifier [3] computes the likelihood that a program is malicious given the features that are contained in the program. This method used strings and byte sequence data to compute a probability of a binary Maliciousness given its features.

B. IBK Algorithm

Instance-based knowledge representation [4]uses the in-stances themselves to represent what is learned, rather than inferring a rule set or decision tree and storing it instead. Once a set of training instances has been memorized, on encountering a new instance the memory is searched for the training instance. This is known as instance-based learning.

C. J48

Perhaps C4.5 algorithm which was developed by Quinlan is the most popular tree classifier. Weka classifier package has its own version of C4.5 known as J48. J48 is an optimized implementation of C4.5 rev. 8. J48 [5]is experimented is this study with the parameters: confidenceFactor = 0.25; numFolds = 3; seed = 1; unpruned = False.

D. RandomForest

The random forest [6]is an ensemble of unpruned classi-fication or regression trees. Random forest generates many classification trees. Each tree is constructed by a different bootstrap sample from the original data using a tree classifica-tion algorithm. After Patterns Network Traffic Preprocessors Detector Training Dataset Pattern Builder On-line Off-line Alerts the forest is formed, a new object that needs to be classified is put down each of the tree in the forest for classification. Each tree gives a vote that indicates the trees decision about the class of the object. The forest chooses the class with the most votes for the object.

E. AttributeSelectedClassifier(ASC)

One of Wekas metalearners, which allows an attribute selection method and a learning algorithm to be specified as part of a classification scheme. ASC ensures that the chosen set of attributes is selected based on the training data only.

F. ClassificationviaRegression(CVR)

It performs classification using a regression method by binarizing the class and building a regression model for each value. RegressionByDiscretization is a regression scheme that discretizes the class attribute into a specified number of bins using equal-width discretization and then employs a classifier. The predictions are the weighted average of the mean class value for each discretized interval, with weights based on the predicted probabilities for the intervals.

G. Decision stump

A decision stump [7]is a decision tree with a root node and two leaf nodes. For each feature in the input data, a decision stump is constructed. The following points support our selection of decision stumps as the weak classifiers: 1) the model that decision stumps use is very simple; 2) there is only one comparison operation in

each decision stump for testing a sample; thus, the test time for each decision stump is very low.

H. REPTree

REPTree builds a decision or regression tree using infor-mation gain/variance reduction and prunes it using reduced-error pruning. Optimized for speed, it only sorts values for numeric attributes once. It deals with missing values by splitting instances into pieces, as C4.5 does. You can set the minimum number of instances per leaf, maximum tree depth (useful when boosting trees), minimum proportion of training set variance for a split (numeric classes only), and number of folds for pruning.

I. RandomTree

Trees built by RandomTree [8]test a given number of random features at each node, performing no pruning. Types of random trees include Uniform spanning tree, Random minimal spanning tree, Random binary tree, Random recursive tree, Treap, Rapidly exploring random tree, Brownian tree, Random forest and branching process.

J. Filteredclassifier

FilteredClassifier applies the filter to the data before running the learning algorithm. This builds the filter using the training data only, and then evaluates it on the test data using the discretization intervals computed for the training data.

K. HoeffdingTree

Hoeffding trees are based on a simple idea known as the Hoeffding bound. It makes intuitive sense that, given enough independent observations, the true mean of a random variable will not differ from the estimated mean by more than a certain amount. In fact, the Hoeffding bound states that with probability 1 -, a random variable of range R will not differ from the estimated mean after n observations.

L. RandomizableFilteredClassifier(RFC)

Class for running an arbitrary classifier on data that has been passed through an arbitrary filter. Like the classifier, the structure of the filter is based exclusively on the training data and test instances will be processed by the filter without changing their structure.

M. JRip

JRip implements RIPPER, including heuristic global opti-mization of the rule set. RIPPER, an acronym for repeated incremental pruning to produce error reduction. Classes are examined in increasing size and an initial set of rules for a class is generated using incremental reduced-error pruning. An extra stopping condition is introduced that depends on the description length of the examples and rule set. The description-length DL is a complex formula that takes into account the number of bits needed to send a set of examples with respect to a set of rules, the number of bits required to send a rule with k conditions, and the number of bits needed to send the integer ktimes an arbitrary factor of 50 percent to compensate for possible redundancy in the attributes.

N. RandomCommitee

Class for building an ensemble of randomizable base clas-sifiers. Each base classifiers is built using a different random number seed (but based one the same data). The final predic-tion is a straight average of the predictions generated by the individual base classifiers.

III. The Comparative Study

The methodology of the study consists of collecting a set of data mining and knowledge discovery tools to be tested, specifying the data set to be used, and selecting a various set of the classification algorithm to test the tools performance.

A. Tools Description

Weka 3.7 is a collection of machine learning algorithms for data mining tasks. Weka stands for Waikato Environment for Knowledge Analysis [9]. The algorithms can either be applied directly to a dataset or called from the Java code. Weka contains various tools for data pre-processing, classification, regression, association rules, clustering, and visualization. The Weka GUI Chooser (class weka.gui.GUIChooser) provides a starting point for launching Wekas main GUI applications and supporting tools. The GUI Chooser

consists of four buttons: one for each of the four major Weka applications and four menus. The buttons can be used to start the applications that are explained as follows:
Explorer: It is an environment used for exploring data with WEKA (the rest of this documentation deals with this application in more detail).
Experimenter: It is an environment for performing exper-iments and conducting statistical tests between learning schemes.
KnowledgeFlow: This environment supports essentially the same functions as the Explorer, but with a drag-and-drop interface. It supports incremental learning.
SimpleCLI: It provides a simple command-line interface that allows direct execution of WEKA commands for operating systems that do not provide their own command line interface.

B. Data Set Description

To verify the efficiency of 15 classification algorithms, I have used NSL-KDD dataset. NSL-KDD dataset is a reduced version of the original KDD 99 dataset. NSL-KDD consists of the same features as KDD 99.The KDD CUP 1999 benchmark datasets are used in order to evaluate different feature selection method for Intrusion detection system [10] . In the KDDCup99 dataset, any network connection (or instance) is comprised of 41 attributes and each instance is labeled either as normal or as an attack-specified type [11]. In KDD99 database, there are 494,021 instances in which 97,278 are considered normal and 396,744 are labeled as attacked by 22 different types that can be classified into 4 main categories as follows:

- Probing is a class of attacks where an attacker scans a net-work to gather information or find known vulnerabilities. An attacker with a map of machines and services that are available on a network can use the information to look for exploits. There are different types of probes: some of them abuse the computers legitimate features; some of them use social engineering techniques. This class of attacks is the most commonly heard and requires very little technical expertise.
- DOS(Denial of service) is a class of attacks where an attacker makes some computing or memory resource too busy or too full to handle legitimate requests, thus denying legitimate users access to a machine There are different ways to launch DOS attacks: by abusing the computers legitimate features; by targeting the implementations bugs; or by exploiting the systems misconfigura-tions.
- U2R(User to Root) exploits are a class of attacks where an attacker starts out with access to a normal user account on the system and is able to exploit vulnerability to gain root access to the system. Most common exploits in this class of attacks are regular buffer overflows, which are caused by regular programming mistakes and environment assumptions.
- R2L(Remote to User) attack is a class of attacks where an attacker sends packets to a machine over a network, then exploits machines vulnerability to illegally gain local access as a user.

Each TCP connection has 41 features [6]with a label which specifies the status of a connection as either being normal or a specific attack type. There are 38 numeric features and 3 symbolic features, falling into the following four categories:
Basic Features: 9 basic features were used to describe each individual TCP connection.
Content Features: 13 domain knowledge related features were used to indicate suspicious behaviour having no sequential patterns in the network traffic. Time-based Traffic Features: 9 features were used to summarize the connections in the past two seconds that had the same destination host or the same service as the current connection. Host-based Traffic Features: 10 features were constructed using a window of 100 connections to the same host instead of a time window, because slow scan attacks may occupy a much larger time interval than two seconds.
In order to test the classifiers, I randomly selected 28660 connection records as a training data set and 19763 connection records as a testing data set. Below Table 1 shows the detail of connection records in these both datasets. NSL-KDD dataset contains symbolic as well as continuous features.

TABLE I. DETAILS OF CONNECTION RECORDS IN USED DATASET

Label	Training set	Testing set
Normal	18763	6602
Probe	1117	3033
DOS	8679	9750
U2R	29	36
R2L	72	342
Total Records	28660	19763

IV. EXPERIMENTS AND EVALUATIONS

A. Result evaluation parameters

The classification models can be evaluated using misclassification error rate and the area under ROC curve. A Confusion Matrix One of the methods to evaluate the performance of a classifier is using confusion matrix. A Confusion matrix [12] that summarizes the number of instances predicted correctly or incorrectly by a classification model. For example we have two classes + and -, and therefore a 2x2 confusion matrix, the matrix could be arbitrarily large. The number of correctly classified instances is the sum of diagonals in the matrix; all others are incorrectly classified (class a gets misclassified as b exactly twice, and class b gets misclassified as a three times). The following terminology is often used when referring to the counts tabulated in a confusion matrix.

a) The True Positive (TP) [13]: corresponds to the number of positive examples correctly predicted by the classifi-cation model.
b) The False Negative (FN) [13]: corresponds to the num-ber of positive examples wrongly predicted as negative by the classification model.
c) The False Positive (FP) [13]: corresponds to the number of negative examples wrongly predicted as positive by the classification model.
d) The True Negative (TN) [13]: corresponds to the number of negative examples correctly predicted by the classifi-cation model.The counts in a confusion matrix can also be expressed in terms of percentages.
   The true positive rate (TPR) [13]or sensitivity is defined as the fraction of positive examples predicted correctly by the model, i.e.,
   TPR = TP / (TP + FN)
   Similarly, the true negative rate (TNR)[13] is defined as the fraction of negative examples predicted correctly by the model, i.e.,
   TNR = TN / (TN + FP)
   False positive rate(FPR) is the fraction of negative examples predicted as a positive class, ie.,
   FPR = FP / (TN + FP)
   Finally the false negative rate (FNR)[13] is the fraction of positive examples predicted as a negative class. ie., FNR = FN / (TP + FN)
e) Recall and Precision: are two widely used metrics em-ployed in applications where successful detection of one of the classes is considered more significant than detection of the other classes.
   Precision, p =TP/ (TP+FP) Recall, r = TP/ (TP+FN)
f) F-measure: A measure [14]that combines precision and recall is the harmonic mean of precision and recall, the traditional F-measure or balanced F-score is :
   F = 2 x (Precision x Recall) / (Precision + Recall)
g) Receiver Operating Characteristic (ROC): In signal de-tection theory, ROC curve is a graphical plot which illustrates the performance of a binary classifier system as its discrimina tion threshold is varied. It is created by plotting the fraction of true positives out of the total actual positives (TPR = true positive rate) vs. the fraction of false positives out of the total actual negatives (FPR = false positive rate), at various threshold settings. The ROC is also known as a relative operating characteristic curve, because it is a comparison of two operating characteristics (TPR and FPR) as the criterion changes.

B. Result of different classification algorithms on Weka

In this I have taken upper defined NSL- KDD dataset as a training set and a testing set in the weka. By implementing different algorithms on this training set and testing set, I have found the performance measures of Normal, Probe, DOS, U2R, R2L attacks from the confusion matrix of each algorithm that is shown in below tables 2 ,3, 4, 5, 6 and their respective figures are 1 ,2 ,3 ,4 ,5. These algorithms are classified according to the various performance measures TP rate,FP rate, Precision, F-Measure, ROC area.

C. Results Analysis of different Algorithms

The below table no. 7 and 8 enable us to analyze the different algorithm results with better perception based on TP rate, ROC area and other performance measure. From the results of these experiments, It is found that Random committee is best for normal according to TP RATE , ROC area. HoeffdingTree is best for detecting Probe attack according to TP rate, Precision , F-MEASURE. RFC is best for detecting DOS attack according

to TP RATE , ROC area.
Randomcommitte is best for detecting U2R attacks according to TP RATE, ROC area. JRip is best for detecting R2L attacks according to TP RATE , FP RATE, ROC area as shown in table no. 7. According to the TP rate, ROC area and other performance measures , various promising classifiers are shown in table no. 8.

TABLE II. PERFORMANCE MEASURES OF DIFFERENT ALGORITHMS FOR NORMAL ATTACK

Classifier name	TP Rate	FP Rate	Precision	F-Measure	ROC Area
Nave bayes	0.876	0.009	0.98	0.925	0.991
IBK	0.997	0.073	0.876	0.933	0.963
J48	0.996	0.043	0.923	0.96	0.968
RandomForest	0.942	0.021	0.959	0.998	0.994
ASC	0.994	0.021	0.96	0.977	0.99
CVR	0.94	0.024	0.953	0.946	0.984
Decision stump	0.968	0.074	0.871	0.917	0.947
REPTree	0.941	0.025	0.952	0.946	0.953
RandomTree	0.94	0.018	0.964	0.952	0.962
Filteredclassifier	0.994	0.055	0.904	0.947	0.99
HoeffdingTree	0.206	0.026	0.802	0.328	0.942
RFC	0.988	0.043	0.923	0.955	0.976
JRip	0.997	0.054	0.905	0.95	0.973
RandomCommitee	0.998	0.019	0.965	0.95	0.997

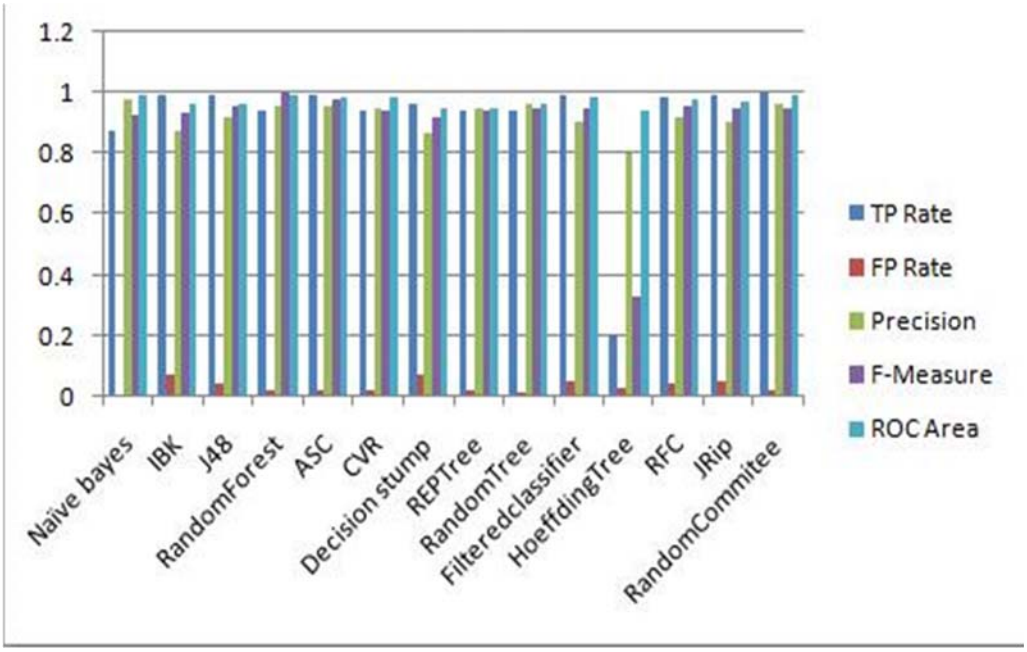

Fig. 1. Chart for Comparative analysis for normal attack

TABLE III. PERFORMANCE MEASURES OF DIFFERENT ALGORITHMS FOR PROBE ATTACK

Classifier name	TP Rate	FP Rate	Precision	F-Measure	ROC Area
Nave bayes	0.968	0.112	0.606	0.746	0.94
IBK	0.859	0.088	0.635	0.73	0.929
J48	0.859	0.102	0.6	0.706	0.852
RandomForest	0.976	0.133	0.568	0.718	0.995
ASC	0.92	0.112	0.614	0.759	0.943
CVR	0.882	0.114	0.58	0.699	0.874
Decision stump	0	0	0	0	0.368
REPTree	0.882	0.005	0.971	0.925	0.986
RandomTree	0.976	0.13	0.568	0.718	0.922
Filteredclassifier	0.861	0.109	0.585	0.697	0.921
HoeffdingTree	0.994	0.004	0.978	0.978	0.959
RFC	0.888	0.005	0.967	0.926	0.955
Jrip	0.485	0.006	0.937	0.639	0.701
RandomCommitee	0.97	0.109	0.614	0.752	0.996

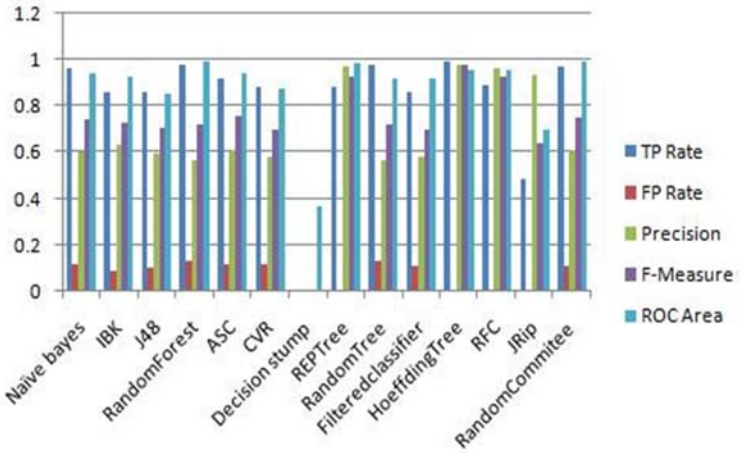

Fig. 2. Chart for Comparative analysis for Probe attack

TABLE IV. PERFORMANCE MEASURES OF DIFFERENT ALGORITHMS FOR DOS ATTACK

Classifier name	TP Rate	FP Rate	Precision	F-Measure	ROC Area
Nave bayes	0.81	0.037	0.954	0.876	0.821
IBK	0.81	0.002	0.9	0.894	0.869
J48	0.81	0.001	0.98	0.895	0.905
RandomForest	0.81	0.001	0.997	0.895	0.996
ASC	0.81	0.002	0.97	0.894	0.904
CVR	0.805	0.063	0.924	0.861	0.983
Decision stump	0.987	0.271	0.776	0.869	0.858
REPTree	0.98	0.07	0.93	0.954	0.987
RandomTree	0.805	0.001	0.978	0.892	0.902
Filteredclassifier	0.81	0.001	0.996	0.894	0.905
HoeffdingTree	0.98	0.541	0.633	0.769	0.955
RFC	0.996	0	0.996	0.94	0.997
Jrip	0.978	0.111	0.893	0.934	0.934
RandomCommitee	0.81	0.001	0.996	0.894	0.996

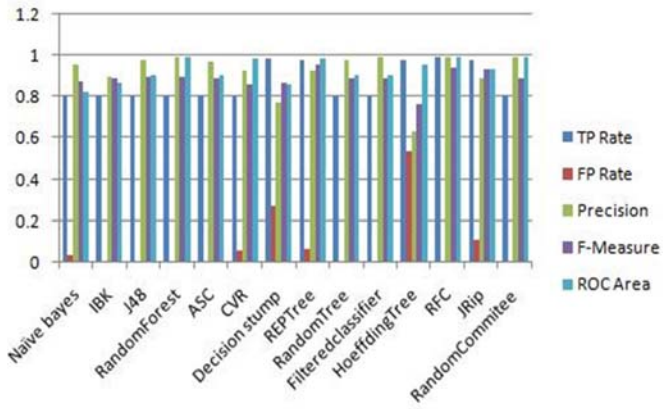

Fig. 3. Chart for Comparative analysis for DOS attack

TABLE V. PERFORMANCE MEASURES OF DIFFERENT ALGORITHMS FOR U2R ATTACK

Classifier name	TP Rate	FP Rate	Precision	F-Measure	ROC Area
Nave bayes	0.944	0.028	0.057	0.107	0.985
IBK	0.944	0.004	0.301	0.456	0.979
J48	0.778	0.013	0.097	0.172	0.949
RandomForest	0.861	0.001	0.544	0.667	0.996
ASC	0.556	0	0.8	0.656	0.808
CVR	0.667	0.001	0.522	0.585	0.996
Decision stump	0	0	0	0	0.797
REPTree	0.944	0.001	0.576	0.716	0.996
RandomTree	0.833	0.003	0.361	0.504	0.915
Filteredclassifier	0.361	0	0.489	0.531	0.877
HoeffdingTree	0.5	0.009	0.087	0.149	0.989
RFC	0.778	0	0.63	0.824	0.924
Jrip	0.944	0.001	0.654	0.773	0.972
RandomCommitee	0.972	0.002	0.507	0.667	0.997

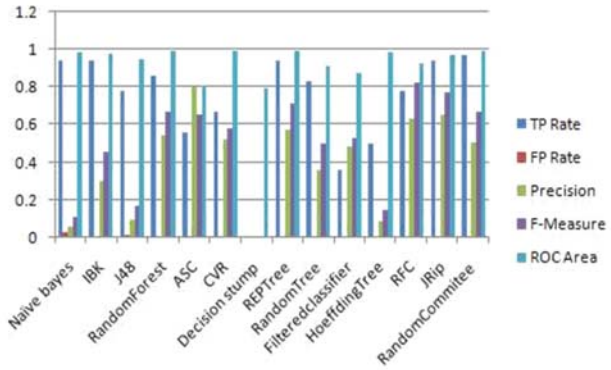

Fig. 4. Chart for Comparative analysis for U2R attack

TABLE VI. PERFORMANCE MEASURES OF DIFFERENT ALGORITHMS FOR R2L ATTACK

Classifier name	TP Rate	FP Rate	Precision	F-Measure	ROC Area
Nave bayes	0.275	0.003	0.553	0.367	0.946
IBK	0.237	0	0.988	0.302	0.689
J48	0.178	0	0.612	0.303	0.86
RandomForest	0.316	0	0.964	0.278	0.751
ASC	0.196	0	0.78	0.328	0.816
CVR	0.254	0	0.935	0.4	0.694
Decision stump	0	0	0	0	0.816
REPTree	0.412	0	0.874	0.414	0.639
RandomTree	0.345	0.003	0.678	0.314	0.678
Filteredclassifier	0.167	0.002	0.606	0.261	0.809
HoeffdingTree	0.161	0.001	0.714	0.447	0.902
RFC	0.275	0.001	0.87	0.418	0.754
Jrip	0.447	0.004	0.614	0.263	0.97
RandomCommitee	0.412	0	0.589	0.42	0.761

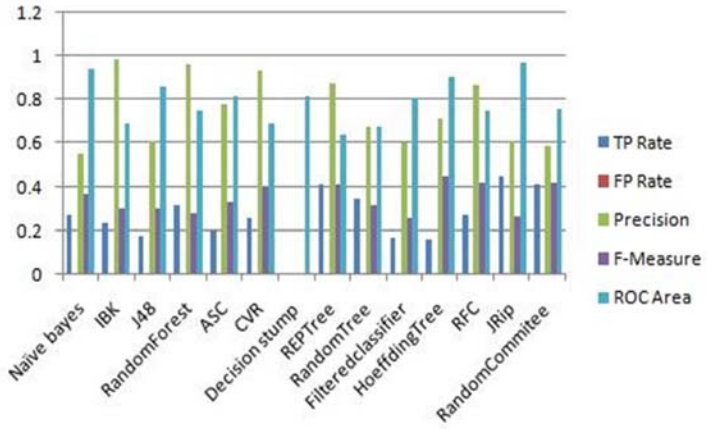

Fig. 5. Chart for Comparative analysis for R2L attack

TABLE VII. RESULT ANALYSIS OF DIFFERENT ALGORITHMS

Performance measures	Normal	Probe	DOS	U2R	R2L
TP Rate	Random Commi-Tee	Hoeffding Tree	RFC	Random Commi-tee	JRip
FP Rate	Decision stump	Random Forest	Hoeffding Tree	Nave bayes	JRip
Precision	Nave bayes	Hoeffding Tree	Random Forest	ASC	IBK
F-Measure	Random Forest	Hoeffding Tree	REPTree	RFC	Hoeffding Tree
ROC Area	Random Commi-Tee	Random Commi-Tee	RFC	Random Commi-tee	JRip

V. CONCLUSION

In this work, I compare the basic classification algorithms. The goal of this study is to provide a comprehensive review of different 14 techniques that are described above in data mining. In order to compare these 14 algorithms based on the TP rate,FP rate,Precision,F-measure,ROC area, we came to the conclusion which algorithm is more efficient to use for detecting the various types of attacks. The performance of the each algorithm is tested on a NSL-KDD data set. After the execution of each classification algorithm, It is found that Random committee is best for normal according to TP RATE , ROC area. HoeffdingTree is best for detecting Probe attack according to TP rate, Precision , F-MEASURE. RFC is best for detecting DOS attack according to TP RATE , ROC area. Random committee is best for detecting U2R attacks according to TP RATE, ROC area. JRip is best for detecting R2L attacks according to TP RATE , FP RATE, ROC area. This gave the accuracy of the detecting the attacks. In future studies, we can ensemble the accuracy of the these resultant algorithm to achieve better results.

TABLE VIII. PROMISING CLASSIFIERS ACCORDING TO TP RATE, ROC AREA AND OTHER PERFORMANCE MEASURES

Attack	Promising classifiers
Normal	Random Committee
Probe	Hoeffding Tree
DOS	RFC
U2R	Random Committee
R2L	JRip

REFERENCES

[1] G. Kumar and K. Kumar, "The use of artificial-intelligence-based ensembles for intrusion detection: a review," Applied Computational Intelligence and Soft Computing, vol. 2012, p. 21, 2012.

[2] S. Chebrolu, A. Abraham, and J. P. Thomas, "Feature deduction and ensemble design of intrusion detection systems," Computers & Security, vol. 24, no. 4, pp. 295–307, 2005.

[3] M. G. Schultz, E. Eskin, E. Zadok, and S. J. Stolfo, "Data mining methods for detection of new malicious executables," in Security and Privacy, 2001. S&P 2001. Proceedings. 2001 IEEE Symposium on. IEEE, 2001, pp. 38–49.

[4] G. MeeraGandhi, "Machine learning approach for attack prediction and classification using supervised learning algorithms," Int. J. Comput. Sci. Commun, vol. 1, no. 2, 2010.

[5] H. A. Nguyen and D. Choi, "Application of data mining to network intrusion detection: classifier selection model," in Challenges for Next Generation Network Operations and Service Management. Springer, 2008, pp. 399–408.

[6] J. Zhang and M. Zulkernine, "Network intrusion detection using random forests." in PST. Citeseer, 2005.

[7] W. Hu, W. Hu, and S. Maybank, "Adaboost-based algorithm for network intrusion detection," Systems, Man, and Cybernetics, Part B: Cybernet-ics, IEEE Transactions on, vol. 38, no. 2, pp. 577–583, 2008.

[8] S. Thaseen and C. A. Kumar, "An analysis of supervised tree based clas-sifiers for intrusion detection system," in Pattern Recognition, Informat-ics and Mobile Engineering (PRIME), 2013 International Conference on. IEEE, 2013, pp. 294–299.

[9] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The weka data mining software: an update," ACM SIGKDD explorations newsletter, vol. 11, no. 1, pp. 10–18, 2009.

[10] K. Kumar, G. Kumar, and Y. Kumar, "Feature selection approach for intrusion detection system."

[11] S. Mukkamala, A. H. Sung, and A. Abraham, "Intrusion detection using an ensemble of intelligent paradigms," Journal of network and computer applications, vol. 28, no. 2, pp. 167–182, 2005.

[12] P. Srinivasulu, D. Nagaraju, P. R. Kumar, and K. N. Rao, "Classifying the network intrusion attacks using data mining classification methods and their performance comparison," International Journal of Computer Science and Network Security, vol. 9, no. 6, pp. 11–18, 2009.

[13] M. Revathi and T. Ramesh, "Network intrusion detection system us-ing reduced dimensionality," Indian Journal of Computer Science and Engineering (IJCSE), vol. 2, no. 1, pp. 61–67, 2011.

[14] A. H. M. Ragab, A. Y. Noaman, A. S. Al-Ghamdi, and A. I. Madbouly, "A comparative analysis of classification algorithms for students college enrollment approval using data mining," in Proceedings of the 2014 Workshop on Interaction Design in Educational Environments. ACM, 2014, p. 106.