HYBRID ENSEMBLE TECHNIQUE FOR SCREENING OF NETWORK ATTACKS

Muskan Mittal [1], Yogesh Kumar [2]

BGIET Sangrur

***Abstract*: Due to excessive use of internet the problem of intrusion is also increased. So, to detect the intrusion in the network traffic, various AI based intrusion detection techniques are used but there is no such technique is available which is used for detecting the network attacks or monitors system activities for malicious activities and produces reports to a management station that can detect various types of network attacks with high accuracy. So the idea of this research paper is to find promising AI based method which classify each type of network traffic class and combine them by proposing an effective combination technique i.e. ensemble technique which can detect all network attacks, so as to increase the overall accuracy and performance of the IDS.**

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

1 INTRODUCTION

An intrusion detection system (IDS) defined as "an effective security technology, which can detect, prevent and possibly react to the various computer attacks [8]" is one of the standard components in security infrastructures. It monitors target sources of activities, such as audit and network traffic data in computer or network systems and then provides various techniques in order to provide security services. The main objective of IDS is to classify intrusive and non-intrusive network activities in an efficient manner. The process of intrusion detection involves various tasks that are as follows: (1) data acquisition/ collection; (2) data Preprocessing & feature selection; (3) model selection for data analysis; (4) classification and result analysis. Figure 1 show the organization of IDS where dotted arrows indicate a response to intrusive activities while solid arrow indicates data flow.

The rest of the paper is organized as follows: Section 2 summaries related works on various data mining, data classification techniques and ensemble techniques. Section 3 summaries the various types of data classification techniques used. Section 4 provides a general description of the tools and software under test and dataset used. Section 5 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.

2 LITERATURE SURVEY

Krishan Kumar et al. [12] in 2013 proposed that at present, network security needs to be concerned to provide secure information channels due to increase in potential network attacks. Intrusion Detection System (IDS) is a valuable tool for computer networks. However, building an efficient ID faces a number of problems. Current IDS finds all data features to detect intrusion or misuse patterns. Some of the features may be duplicate or contribute little to the detection process; their usage can decrease the intrusion detection efficiency as well as taking more computational time for the effective response in real time environment. The purpose of this paper is to identify important input features in building IDS that is relatively efficient and effective. In this work, the feature selection methods are used by ranking them providing the various feature selection algorithms like OneR, RELIEF etc. are proposed. Combining the features of the best algorithms whose performance is better by comparing the result with each other using J48 classifier. The empirical results indicate that input features are too important to detect the intrusions and reduces the dimensionality of the training time, features and increases overall accuracy.

Laheeb M. Ibrahim et al. [13] in 2013 suggested that Detecting anomalous traffic on the internet has remained an issue of security researchers over the years. The advances in the area of computing performance, in terms of storage and processing power, have fostered their ability to host resource-intensive intelligent algorithms, to detect the intrusive activity, in a proper time. The performance of Self Organization Map (SOM), Artificial Neural Network are studied and analyzed, when implemented as part of an Intrusion Detection System in Databases KDD 99 and NSL-KDD datasets of internet traffic activity simulation. Results obtained are compared and analyzed based on several performance metrics.

Gulshan Kumar et al. [8] in 2012 proposed that in supervised learning-based classification, ensembles have been successfully used in different application domains. In the literature, many researchers have proposed different ensembles by considering different combination techniques, training datasets, base classifiers, and many other factors. Artificial-intelligence-(AI-) based techniques play important role in development of ensemble methods for intrusion detection (ID) and have many benefits over other techniques. However, there is no such review of ensembles in general and AI-based ensembles for ID to examine and understand their current research to solve the ID problem. Here, an updated review of ensembles and their respective taxonomies has been presented in general. This paper also presents the updated review of various AI-based ensembles for ID. The related studies of AI-based ensembles are compared by set of evaluation metrics driven from (1) different methods utilized in different phases of ensemble learning; (2) architecture & approach followed; (3) other measures used to evaluate classification performance of the ensembles.

Shelly Xiaonan Wu et al. [29] in 2010 proposed that Intrusion detection based upon computational intelligence is currently attracting interest from the research community. Characteristics of computational intelligence (CI) systems, such as fault tolerance, adaptation, high computational speed and error resilience in the face of noisy information fit the requirements of building a required intrusion detection model. An overview of the research progress in applying CI methods to the problem of intrusion detection is

provided. The scope of this review will be on core methods of CI including fuzzy systems, evolutionary computation, artificial neural networks, artificial immune systems, swarm intelligence, and soft computing. The research contributions in each field are systematically summarized and compared, providing us to clearly define existing research challenges, and to highlight promising new research directions.

3 TECHNIQUES USED

3.1 Naive Bayes: The naive Bayes classifier [3] finds 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.

3.2. IBK Algorithm: Instance-based knowledge representation [4] uses the instances themselves to recognize what is learned, rather than inferring a rule set or decision tree. 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.

3.3. J48: Perhaps C4.5 algorithm 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.

3.4. RandomForest: The random forest [6] is an ensemble of various classification or regression trees. Random forest generates many classification trees. By using a tree classification algorithm, each tree is constructed by a different bootstrap sample from the original data. An on-line alert tells that 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 to indicate the trees decision about the class of the object. The forest chooses the class with the most votes for the object.

3.5. AttributeSelectedClassifier(ASC) : One of Weka meta learners, 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.

3.6. 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 discretized 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.

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

3.8. REPTree: REPTree builds a decision or regression tree using information gain or variance reduction and by using reduced-error pruning method prunes it. It only sorts values for numeric attributes once. It deals with missing values by splitting the instances into pieces, as C4.5 does.

3.9. 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, Rapidly exploring random tree, Brownian tree, Random forest and branching process.

3.10. 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.

3.11. 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.

3.12. RandomizableFilteredClassifier (RFC): It is a Class for running an arbitrary classifier on data that has been passed through an ordinary filter. Like the classifier, the structure of the filter is totally based on the training data and test instances will be processed by the filter without changing their structure.

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

3.14. RandomCommittee: Class for building an ensemble of randomizable base classifiers. Each base classifier is constructed using a different random number seed. The final prediction is a straight average of the predictions that is generated by the individual base classifiers.

3.15. KNN (k nearest neighbor): KNN is part of supervised learning that has been used in many applications in the field of data mining, image processing, statistical pattern recognition etc. It works based on finding of the minimum distance from the query instance to the training samples to get the K-nearest neighbors. After gathering K nearest neighbors, a simple majority of these K-nearest neighbors to be the prediction of the query instance is taken. The KNN prediction of the query instance is based on simple majority of the category of nearest neighbors.

4 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.

4.1 Tools Description

Weka 3.6 is a collection of machine learning algorithms for data mining tasks. Weka stands for Waikato Environment for Knowledge Analysis [13]. 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 Weka's 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 experiments 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.

4.2 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. 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:

1. Basic Features: 9 basic features were used to describe each individual TCP connection.
2. Content Features: 13 domain knowledge related features were used to indicate suspicious behavior having no sequential patterns in the network traffic.
3. 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.
4. 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 30000 connection records as a training data set and 20000 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 1: details of connection records in used dataset

Label	Training set	Testing set
Normal	20103	6839
Probe	1117	3033
DOS	8679	9750
U2R	29	36
R2L	72	342
Total Records	30000	20000

4.3 Ensemble technique used

The ensembles involve the uses of multiple base classifiers and combine their outputs to obtain reliable and more accurate predictions. By keeping the benefits of AI techniques and performance enhancement by using ensemble approach in mind, I develop a new ensemble technique, in which a training set and testing set are used to train the pool of base classifiers. After that, performance measures of different base classifiers for various attacks are being postulated. Then promising classifiers according to TP rate, roc area and other performance measures are find out. According to which values of different attacks obtained from promising classifiers. Then by using Union ensemble technique , new training set is get obtained which is used to send the secure data over the network. Algorithm of Union ensemble technique is given below:

Algorithm to find the ensemble of the base classifiers:

1. Take 5 .arff files to merge.
2. Extract data of files from where "@data" starts.
3. Find distinct elements of each file by using function

file= file. Distinct().ToList();

4. Then merge all files using union function

file1= file1.Union (file2).ToList();

5. End

Therefore, by using ensemble technique it can detect all network attacks, so as to increase the overall accuracy and performance of the IDS.

5 EXPERIMENTS AND EVALUATIONS

5.1 Result evaluation parameters

1) The correctly and incorrectly classified instances show the percentage of test instances that were correctly and incorrectly classified. The percentage of correctly classified instances is often called accuracy or sample accuracy.

2) Root Mean Squared Error (RMSE): The RMSE is a quadratic scoring rule which measures the average magnitude of the error.

RMSE= sqrt $((p_1.a_1)^2+\ldots..+ (p_n-a_n)^2/n)$

3. Relative Absolute Error (RAE): It is just the total, absolute error, with the same kind of normalization.

RAE= $(|p_1-a_1|+\ldots+|p_n-a_n|)/(|\bar{a}-a_1|+\ldots+|\bar{a}-a_n|)$

(4) Root Relative squared error (RRSE): The root relative squared error takes the total root of squared error and normalizes it by dividing the total squared error of the default predictor. Root relative squared error Ei of an individual program i is evaluated by the equation:

$E_i=\sqrt{\sum_{j=1}^{n}(P_{ij}-T_j)2/\sum_{j=1}^{n}(T_j-\bar{T})\,2}$

Where P(ij) is the value predicted by the individual program i for sample case j (out of n sample cases); Tj is the target value for sample case j; and is given by the formula:

$$\bar{T}=1/n\sum_{j=1}^{n}T_j$$

(5) Mean absolute error (MAE): The mean absolute erroris less sensitive to outliers than the mean squared error. The error rates are used for numeric prediction rather than classification.

MAE=$|p_1-a_1|+\ldots..+|p_n-a_n|/n$

(6) 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)

(7) False positive rate(FPR) is the fraction of negative examples predicted as a positive class, ie.,

FPR = FP / (TN + FP)

(8)Recall and Precision: are two widely used metrics employed 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)

(9)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)

(10)Receiver Operating Characteristic (ROC): In signal detection theory, ROC curve is a graphical plot which illustrates the performance of a binary classifier system as its discriminated 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.

5.2 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, and R2L attacks from the confusion matrix of each algorithm that is shown in below tables 2, 3, 4, 5, 6. These algorithms are classified according to the various performance measures TP rate, FP rate, Precision, F-Measure, ROC area.

Table 2: Performance measures of different algorithms for normal attack

S.NO.	Classifier name	TPR	FPR	Precision	F-Measure	ROC Area
1	Naïve bayes	0.876	0.009	0.98	0.925	0.991
2	IBK	0.997	0.073	0.876	0.933	0.963
3	J48	0.996	0.043	0.923	0.96	0.968
4	RandomForest	0.942	0.021	0.959	0.998	0.994
5	ASC	0.994	0.021	0.96	0.977	0.99
6	CVR	0.94	0.024	0.953	0.946	0.984
7	Decision stump	0.968	0.074	0.871	0.917	0.947
8	REPTree	0.941	0.025	0.952	0.946	0.953
9	RandomTree	0.94	0.018	0.964	0.952	0.962
10	Filteredclassifier	0.994	0.055	0.904	0.947	0.99
11	HoeffdingTree	0.206	0.026	0.802	0.328	0.942
12	RFC	0.988	0.043	0.923	0.955	0.976
13	Jrip	0.997	0.054	0.905	0.95	0.973
14	RandomCommitee	0.998	0.019	0.965	0.95	0.997

Table 3: Performance measures of different algorithms for probe attack

S.NO.	Classifier name	TP Rate	FP Rate	Precision	F-Measure	ROC Area
1	Naïve bayes	0.968	0.112	0.606	0.746	0.94
2	IBK	0.859	0.088	0.635	0.73	0.929
3	J48	0.859	0.102	0.6	0.706	0.852
4	RandomForest	0.976	0.133	0.568	0.718	0.995
5	ASC	0.92	0.112	0.614	0.759	0.943
6	CVR	0.882	0.114	0.58	0.699	0.874
7	Decision stump	0	0	0	0	0.368
8	REPTree	0.882	0.005	0.971	0.925	0.986
9	RandomTree	0.976	0.13	0.568	0.718	0.922
10	Filteredclassifier	0.861	0.109	0.585	0.697	0.921
11	HoeffdingTree	0.994	0.004	0.978	0.978	0.959
12	RFC	0.888	0.005	0.967	0.926	0.955
13	Jrip	0.485	0.006	0.937	0.639	0.701
14	RandomCommitee	0.97	0.109	0.614	0.752	0.996

Table 4: Performance measures of different algorithms for dos attack

S.NO.	Classifier name	TP Rate	FP Rate	Precision	F-Measure	ROC Area
1	Naïve bayes	0.81	0.037	0.954	0.876	0.821
2	IBK	0.81	0.002	0.9	0.894	0.869
3	J48	0.81	0.001	0.98	0.895	0.905
4	RandomForest	0.81	0.001	0.997	0.895	0.996
5	ASC	0.81	0.002	0.97	0.894	0.904
6	CVR	0.805	0.063	0.924	0.861	0.983
7	Decision stump	0.996	0.271	0.776	0.869	0.997
8	REPTree	0.98	0.07	0.93	0.954	0.987
9	RandomTree	0.805	0.001	0.978	0.892	0.902
10	Filteredclassifier	0.81	0.001	0.996	0.894	0.905
11	HoeffdingTree	0.98	0.541	0.633	0.769	0.955
12	RFC	0.987	0	0.996	0.94	0.858
13	Jrip	0.978	0.111	0.893	0.934	0.934
14	RandomCommitee	0.81	0.001	0.996	0.894	0.996

Table 5: Performance measures of different algorithms for u2r attack

S.NO.	Classifier name	TP Rate	FP Rate	Precision	F-Measure	ROC Area
1	Naïve bayes	0.944	0.028	0.057	0.107	0.985
2	IBK	0.944	0.004	0.301	0.456	0.979
3	J48	0.778	0.013	0.097	0.172	0.949
4	RandomForest	0.861	0.001	0.544	0.667	0.996
5	ASC	0.556	0	0.8	0.656	0.808
6	CVR	0.667	0.001	0.522	0.585	0.996
7	Decision stump	0	0	0	0	0.797
8	REPTree	0.944	0.001	0.576	0.716	0.996
9	RandomTree	0.833	0.003	0.361	0.504	0.915
10	Filteredclassifier	0.361	0	0.489	0.531	0.877
11	HoeffdingTree	0.5	0.009	0.087	0.149	0.989
12	RFC	0.778	0	0.63	0.824	0.924
13	Jrip	0.944	0.001	0.654	0.773	0.972
14	RandomCommitee	0.972	0.002	0.507	0.667	0.997

Table 6

Performance measures of different algorithms for r2l attack

S.NO.	Classifier name	TP Rate	FP Rate	Precision	F-Measure	ROC Area
1	Naïve bayes	0.275	0.003	0.553	0.367	0.946
2	IBK	0.237	0	0.988	0.302	0.689
3	J48	0.178	0	0.612	0.303	0.86
4	RandomForest	0.316	0	0.964	0.278	0.751
5	ASC	0.196	0	0.78	0.328	0.816
6	CVR	0.254	0	0.935	0.4	0.694
7	Decision stump	0	0	0	0	0.816
8	REPTree	0.412	0	0.874	0.414	0.639
9	RandomTree	0.345	0.003	0.678	0.314	0.678
10	Filteredclassifier	0.167	0.002	0.606	0.261	0.809
11	HoeffdingTree	0.161	0.001	0.714	0.447	0.902
12	RFC	0.275	0.001	0.87	0.418	0.754
13	Jrip	0.447	0.004	0.614	0.263	0.97
14	RandomCommitee	0.412	0	0.589	0.42	0.761

5.3 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. Decision stump 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, and 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 no. 9 shows the values of different attacks obtained from the promising classifiers.

Table 7: Result analysis of different algorithms

S.no	Performance measures	NORMAL	PROBE	DOS	U2R	R2L
1	TP Rate	Random Commitee	Hoeffding Tree	Decision stump	Random Commitee	JRip
2	FP Rate	Decision stump	Random Forest	HoeffdingTree	Naïve bayes	JRip
3	Precision	Naïve bayes	Hoeffding Tree	RandomForest	AttributeSelected Classifier	IBK
4	F-Measure	Random Forest	Hoeffding Tree	REPTree	Randomizable filteredclassifier	HoeffdingTree
5	ROC Area	Random Commitee	Random Commitee	Decision stump	Random Commitee	JRip

Table 8: Promising classifiers according to tp rate, roc area and Other performance measures

Identified Attacks	Promising Classifiers
Probe	HoeffdingTree
Normal/U2R	Random committee
DOS	Decision stump
R2L	JRip

Table 9: Values of different attacks obtained from promising classifiers

Normal	6837
Probe	2790
DOS	9747
U2R	35
R2L	153
total	19562

5.4 After applying ensemble technique on attacks obtained from promising classifiers

The below table no. 10 and 11 enable us to analyze the training set that is obtained by applying ensemble technique on predicted testing set and then we got the instances of new training set and testing set obtained from ensemble training set.

Table 10: Training set obtained by applying ensemble technique on predicted testing set

Normal	6628
probe	403
DOS	4313
U2R	35
R2L	145
total	11524

Table 11: Instances of new training set and testing set obtained from above training set

Attacks	New Training set	New Testing set
Normal	4610	2018
probe	134	269
DOS	1766	2547
U2R	20	15
R2L	37	108
Sub total	6567	4957
Total	11524	

5.5 Result for KNN algorithm

In this I have taken instances of ensemble dataset as a training set and a testing set shown in table no. 11. By implementing KNN algorithm on this training set and testing set, I have find the % of correctly classified instances , incorrectly classified instances, Mean absolute error, Root mean squared error, Root Relative squared error, Relative absolute error by using majority vote classification among the classification of the K objects. Finally I got the result as shown in table no. 12 and 13.

Table 12: Output given by knn algorithm

Label	Testing set	Output set
Normal	2018	2018
Probe	269	125
DOS	2547	2814
U2R	15	0
R2L	108	0

Table 13: Correctly classified instances given by knn algorithm

Attacks	Frequency
Normal	2018
Probe	125
DOS	2547
U2R	0
R2L	0

Table 14: Correctly classified instances given by knn algorithm

Parameters	Result
% of correctly classified instances	94.6
% of incorrectly classified instances	5.4
Mean absolute error	0.0107
Root mean squared error	0.0649
Root Relative squared error	0.1346
Relative absolute error	0.1034

5.6 Result of different classification algorithms on Weka

In this I have taken upper defined KDD dataset as a training set and a testing set in weka that is shown in table no. 1. By implementing different algorithm on this training set and testing set, I have find the % of correctly classified instances , incorrectly classified instances, Mean absolute error, Root mean squared error, Root Relative squared error, Relative absolute error that are shown in .below table no. 15.

Table 15: Performance of different algorithms on weka

S.NO.	Classifier name	% of correctly classified instances	% of incorrectly classified instances	Mean absolute error	Root mean squared error	Relative absolute error	Root Relative squared error
1	Naïve bayes	84.75	15.25	0.0608	0.2451	24.3511	61.972
2	IBK	87.19	12.8	0.051	0.2263	20.52	57.21
3	J48	87.115	12.885	0.0516	0.226	20.66	57.136
4	RandomForest	87.175	12.825	0.0439	0.1651	17.576	41.7504
5	ASC	88.98	11.02	0.0444	0.2095	17.8	52.97
6	CVR	85.33	14.67	0.0572	0.2078	22.89	52.53
7	Decision stump	81	19	0.09	0.26	36.87	67.18
8	REPTree	84.2	15.8	0.02	0.14	10.53	36.03
9	RandomTree	86.95	13.04	0.05	0.22	20.9	57.72
10	Filteredclassifier	86.88	13.12	0.052	0.223	20.97	56.53
11	HoeffdingTree	69	30.8	0.08	0.22	33.31	57.01
12	RFC	83	17	0.013	0.11	5.45	29.45
13	JRip	87	13	0.03	0.19	15.81	50.16
14	RandomCommitee	89.2	10.7	0.04	0.14	16.01	37.21

4.2: Comparison of Results obtained by KNN and other 14 Algorithms

The below table no. 16 shows the comparison of KNN algorithm with other algorithms. The below fig. enable us to analyze the different algorithm results with better perception.

Table 16: Result analysis of knn & other 14 algorithms

Parameter	KNN	NB	IBK	J48	RF	ASC	CVR	DS	REPT	RT	FC	HT	RFC	JRip	RC
% of correctly classified instances	94.6	84.75	87.19	87.115	87.175	88.98	85.33	81	84.2	86.95	86.88	69	83	87	89.2
% of incorrectly classified instances	5.4	15.25	12.8	12.885	12.825	11.02	14.67	19	15.8	13.04	13.12	30.8	17	13	10.7
Mean absolute error	0.0107	0.0608	0.051	0.05116	0.0439	0.0444	0.0572	0.09	0.02	0.05	0.052	0.08	0.013	0.03	0.04
Root mean squared error	0.0649	0.2451	0.2263	0.226	0.1651	0.2095	0.2078	0.26	0.14	0.22	0.223	0.22	0.11	0.19	0.14
Root Relative squared error	13.46	61.97	57.21	57.136	41.7504	52.97	52.53	67.18	36.03	57.72	56.53	57.01	29.45	50.16	37.21
Relative absolute error	10.34	24.351	20.52	20.66	17.576	17.8	22.89	36.87	10.53	20.9	20.97	33.31	5.45	15.81	16.01

From the results of these experiments, K-Nearest Neighbour algorithm proved to have better results of finding the 94.6 % of correctly classified instances by using ensemble technique on the KDD dataset as compared to the other 14 algorithms.

6 CONCLUSIONS

In this work, I compare the basic classification algorithms. The goal of this study is to provide a comprehensive review of different classification techniques in data mining. In order to compare these algorithms based on the correctly classified instances, Relative absolute error, Relative squared error, Mean absolute error, Mean squared error, Root mean squared error and other parameters, we came to the conclusion which algorithm is more efficient to use for classifying each type of network traffic class by using effective combination technique i.e. ensemble technique which can detect all network attacks, so as to increase the overall accuracy and performance of the IDS. The performance of the each algorithm is tested on a KDD data set. After the execution of each classification algorithm, I got the numbers of correctly classified instances and the incorrectly classified instances. This gave the accuracy of the algorithm. The overall evaluation shows that K-nearest neighbor algorithm is far better than other Algorithms. In future studies, we can enhance the accuracy of the KNN algorithm to achieve better results than the previous methodology that I have discussed.

REFERENCES

[1] A. A. Olusola, A. S. Oladele, and D. O. Abosede, "*Analysis of kdd99 intrusion detection dataset for selection of relevance features*," in Proceedings of the World Congress on Engineering and Computer Science, **vol. 1**, 2010, pp. 20–22.

[2] 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.

[3] A. H. Wahbeh, Q. A. Al-Radaideh, M. N. Al-Kabi, and E. M. Al-Shawakfa, "*A Comparison Study between Data Mining Tools over some Classification Methods*", (IJACSA) International Journal of Advanced Computer Science and Applications,2011.

[4] A. Lazarevic, L. Ert¨oz, V. Kumar, A. Ozgur, and J. Srivastava, "*A comparative study of anomaly detection schemes in network intrusion detection.*" in SDM. SIAM, 2003, pp. 25–36.

[5] A. Satheesh, R. Patel, "*Dynamic Nearest Neighbours Classifier For Integrated Data Using Object Oriented Concept Generalization*", IJSSST, **Vol.11**, No. 1,2010.

[6] D. L. AL-Nabi, S. S. Ahmed,"*Survey on Classification Algorithms for DataMining:(Comparison and Evaluation)*", Computer Engineering and Intelligent Systems, ISSN 2222-1719 (Paper) ISSN 2222-2863 (Online), **Vol.4**, No.8, 2013.

[7] D. TIGABU, "*Constructing predictive model for network intrusion detection*."

[8] 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.

[9] 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.

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

[11] K. H. Raviya, BirenGajjar, "*Performance Evaluation of Different Data Mining Classification Algorithm Using WEKA*", ISSN - 2250-1991, **Volume 2**, Issue 1 , January 2013.

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

[13] 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, pp. 107–119, 2013.

[14] 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.

[15] 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.

[16] M. Sharma, S. K. Sharma, "*Generalized K-Nearest Neighbour Algorithm- A Predicting Tool*", International Journal of Advanced Research in Computer Science and Software Engineering, **Volume 3**, Issue 11, November 2013.

[17] O. Depren, M. Topallar, E. Anarim, and M. K. Ciliz, "*An intelligent intrusion detection system (ids) for anomaly and misuse detection in computer networks*," Expert systems with Applications, **vol. 29**, no. 4, pp. 713–722, 2005.

[18] Oliver Sutton, "*Introduction to k nearest Neighbour Classification*", Februar 2012.

[19] O. Veksler, "*Machine Learning in computer vision*",2008.

[20] P. De Boer and M. Pels, "*Host-based intrusion detection systems*," Amsterdam University, 2005.

[21] 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.

[22] P. T. B. Fomby, "*K-Nearest Neighbors Algorithm: Prediction and Classification*", Department of Economics, Southern Methodist University, Dallas, TX 75275, February 2008.

[23] S. Bishnoi, "*Comparison of classification techniques*", ISSN 2231-4334, IJRIM, **Volume 1**, Issue 2 (June, 2011).

[24] 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.

[25] 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.

[26] S. Neelamegam, Dr.E.Ramaraj, "*Classification algorithm in Data mining*", International Journal of P2P Network Trends and Technology (IJPTT) – **Volume 4** , Issue 8,Sep 2013

[27] S. Pandya, Dr. P. V. Virparia, "*Comparing the Applications of Various Algorithms of Classification Technique of Data Mining in an Indian University to Uncover Hidden Patterns*", International Journal of Advanced Research in Computer Science and Software Engineering, ISSN: 2277 128X, Volume 3, Issue 5, May 2013.

[28] S. Thaseen and C. A. Kumar, "*An analysis of supervised tree based classifiers for intrusion detection system*," in Pattern Recognition, Informatics and Mobile Engineering (PRIME), 2013 International Conference on. IEEE, 2013, pp. 294–299.

[29] S. X. Wu and W. Banzhaf, "*The use of computational intelligence in intrusion detection systems: A review*," Applied Soft Computing, **vol. 10**, no. 1, pp. 1–35, 2010.

[30] T. N. Phyu,"*Survey of Classification Techniques in Data Mining*", Proceedings of the International Multi Conference of Engineers and Computer Scientists 2009, **Vol 1**, IMECS 2009, March 18 - 20, 2009, Hong Kong.

[31] V. Kumar, H. Chauhan, and D. Panwar, "*K-means clustering approach to analyze nsl-kdd intrusion detection dataset*," International Journal of Soft Computing and Engineering (IJSCE) ISSN, pp. 2231–2307, 2013.

[32] 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.

[33] Y. B. Bhavsar and K. C. Waghmare, "*Intrusion detection system using data mining technique: Support vector machine*," International Journal of Emerging Technology and Advanced Engineering, **vol. 3**, no. 3, pp. 581–586, 2013.

[34] Y. Kumar and I. Bala, "*Identify Promising Classifiers for Each Type of Attack Class*" Fourth International Conference on Advances in Computer Science and Application, Grenze Scientific society, CSA-2015.

[35] Y. Kumar and I. Bala, "*Comparative analysis of various data mining classification algorithms*", 3rd International Conference on Advancements in Engineering & Technology (ICAET-2015), March 2015.