Indian Journal of Science and Technology, *Vol 8(17), DOI: 10.17485/ijst/2015/v8i17/70324, August 2015*
ISSN (Print) : 0974-6846
ISSN (Online) : 0974-5645

Performance Analysis of Multiple Classifiers on KDD Cup Dataset using WEKA Tool

S. Venkata Lakshmi[1*] and T. Edwin Prabakaran[2]

[1]Department of Computer Science, Loyola College, Chennai - 600034, Tamil Nadu, India; jaibv2012@gmail.com
[2]Department of Statistics, Loyola College, Chennai - 600034, Tamil Nadu, India; teprabakaran@yahoo.com

Abstract

Background/Objectives: The objective of this work is to find the best among the ten classification algorithms considered to classify the connection records into normal or abnormal in the KDDCup20% training data set using WEKA tool. **Methods/ Statistical Analysis**: In this work, the experiment is carried out by the application of 10 classification algorithms on the KDDCup 20% training dataset comprising of 25192 instances through an experiment type of 10-fold cross validation. The tests were configured with Paired T Tester (corrected) and the level in the test of significance was taken as 0.05. The comparison fields Percent_correct, fmeasure, irrecall, irprecision and auc (area under roc) were taken for evaluation. Tests were also performed for ranking and summary. **Findings**: As per the results obtained by the Weka Experimenter with the 10 classifiers on the KDD 20% training dataset, it has been analysed that Random forest classifier works best with the comparison fields percent_correct, fmeasure and AUC (Area under ROC). Simplecart classifier ranks next to Randomforest classifier with the comparison fields percent_correct and measure. Simplecart classifier outperforms all other classifiers with respect to the comparison field irprecision. ZeroR is found to be the worst classifier in terms of all the comparison fields other than irrecall. Thus it has been found that with the dataset that is taken for experiment, further detailed study could be restricted only with the five classifiers namely Random Forest, Simple cart, J48, Bagging and IBk. This will definitely reduce computational time and increase the efficiency of classification of the KDDCup20% data set.

Keywords: Connection Records, Intrusion Detection, Multiple Classifiers, Normal, Testing, Testing Dataset, Training Dataset, Weka Experimenter

1. Introduction

Intrusion is defined as any set of action that can compromise the integrity, confidentiality and availability of system resources[1]. There are two types of intrusion detection namely, misuse detection and anomaly detection. Misuse detection refers to the identification of the already known intrusion patterns in the dataset. Known attack patterns are easily identified using their signatures in misuse detection models. They are also called as Signature based Intrusion Detection Systems (IDS). Signature based IDS are unable to detect unknown and emerging attacks since signature database has to be manually revised for any new attack. The other type named Anomaly detection refers to the detection of novel intrusion patterns in data[2,3]. This could be used to identify known and unknown attacks[4].

There are several papers which deal with Intrusion detection in various angles. Intrusions are normally difficult to identify since there are various threats which are not real intrusions. Sometimes, user may not be able to identify the real intrusion[4-6].

*Author for correspondence

2. Intrusion Dataset

The KDD Cup dataset[7] is considered to be the benchmark data in Intrusion detection. The dataset was a collection of simulated raw TCP dump data over a period of nine weeks on a local area network. The known attack types are those present in the training dataset while the novel attacks are the additional attacks which are not present in the training dataset. There are various attacks like Buffer overflow, Perl, Port sweep, Neptune, Smurf, Teardrop, Guess password, IP Sweep etc. The training dataset consists of 4,94,021 records[7]. The testing dataset consists of 3,11,029 records. In each connection record there are 41 attributes describing different features of the connection. In the training dataset, along with the 41 attributes a class attribute is also given. In our study, we have taken KDD Cup 20% training dataset for experimenting with multiple classifiers.

3. Data Mining in Classification

Data mining is a finding process of significant non-intuitive correlations and patterns, making possible to get high level knowledge information from low level data. Data mining is also Knowledge Discovery in data. It is the non-trivial process of identifying valid and novel useful and understandable patterns of data. Data mining is more than collection of data. It involves analysis and predictions.

Classification is a data mining task that maps the data into predefined groups and classes. It is also called as supervised learning. It consists of two steps. First step is the model construction which consists of set of predetermined classes. Each tuple is assumed to belong to a predefined class. The set of tuple used for model construction is training set. The model is represented as classification rules, decision trees, or mathematical formulae. Second step is model usage which is used for classifying future or unknown objects. The known label of test sample is compared with the classified result from the model[8,9].

4. Classification Algorithms Used

Classifiers such as OneR, ZeroR, BayesNet, NaiveBayes, IBk, Adaboost, Meta bagging, J48, Random forest and Simple cart are used in this paper. A brief note about the various classifiers used in this paper is given below:

4.1 OneR Classifier

OneR classifier short for 'One Rule' is a simple, yet accurate, classification algorithm. It generates one rule for each predictor in the data, and then selects the rule with the smallest total error as its "one rule". OneR produces rules only slightly less accurate than state of the art classification algorithms but produces rules that are simple for humans to interpret.

4.2 ZeroR Classifier

ZeroR is the simplest classification method which relies on the target and ignores all predictors. ZeroR classifier simply predicts the majority category (class). Although there is no predictability power in ZeroR, it is useful for determining a baseline performance as a benchmark for other classification methods.

4.3 BayesNet Classifier

A Bayesian Network, Bayes Network, Bayesian Model or Probabilistic directed acyclic graphical model is a probabilistic graphical model that represents a set of random variables and their conditional dependencies via a Directed Acyclic Graph (DAG).

4.4 NaiveBayes Classifier

The Naive Bayesian classifier is based on Bayes' theorem with independence assumptions between predictors. A Naive Bayesian model is easy to build, with no complicated iterative parameter estimation which makes it particularly useful for very large datasets. Despite its simplicity, the Naive Bayesian classifier often does surprisingly well and is widely used because it often outperforms more sophisticated classification methods[9].

4.5 IBk Classifier

The k-Nearest Neighbour (k-NN) is a method for classification of objects based on the closest training examples in the feature space. k-NN is a type of instance based learning or lazy learning. The k-NN is one of the simplest of machine learning algorithms. An object is classified by a majority vote of its neighbours, with the object being

assigned to the class most common amongst its k nearest neighbours (k is a positive integer, typically small). If k = 1, then the object is simply assigned to the class of its nearest neighbour[10].

4.6 Adaboost Classifier

Bagging and Boosting are Meta algorithms that pool decisions from multiple classifiers. This algorithm iteratively learns from weak classifiers. The final result is the weighted sum of the results of weak classifiers.

4.7 Meta Bagging Classifier

Bagging generates bootstrap samples of the training data. Then it trains a classifier or a regression function using each bootstrap sample. For classification purpose, the majority vote on the classification results is taken. The average on the predicted values is taken for regression. The advantage of bagging is that it reduces variation and it improves performance for unstable classifiers which vary significantly with small changes in the dataset.

4.8 J48 Classifier

J48 is slightly modified C4.5 in WEKA. The C4.5 algorithm generates a classification-decision tree for the given data-set by recursive partitioning of data. The decision is grown using Depth-first strategy. The algorithm considers all the possible tests that can split the data set and selects a test that gives the best information gain. For each discrete attribute, one test with outcomes as many as the number of distinct values of the attribute is considered. For each continuous attribute, binary tests involving every distinct values of the attribute are considered. In order to gather the entropy gain of all these binary tests efficiently, the training data set belonging to the node in consideration is sorted for the values of the continuous attribute and the entropy gains of the binary cut based on each distinct values are calculated in one scan of the sorted data. This process is repeated for each continuous attributes.

4.9 Random Forest Classifier

Random forests are an ensemble learning method for classification, regression and other tasks that operate by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees.

4.10 Simple Cart Classifier

Simple Cart (Classification and Regression tree) is a classification technique that generates the binary decision tree. Since output is a binary tree, it generates only two children. Entropy is used to choose the best splitting attribute. Simple Cart handles the missing data by ignoring that record.

5. Introduction to Weka Tool

WEKA stands for Waikato Environment for Knowledge Learning. It was developed by the University of Waikato, New Zealand. Weka is a collection of machine learning algorithms for data mining tasks. The algorithms can either be directly applied to the dataset or called from java code. Weka contains tools for data pre-processing, classification, regression, clustering, association rules and visualization[11]. It is well suited for developing new machine learning schemes. The dataset used in Weka is to be in the .ARFF format. This type of file consists of a header which describes the attribute types and a data section which is a comma separated list of data.

WEKA tool comprises of four buttons namely, Explorer, Experimenter, Knowledge Flow and Simple CLI. Explorer is an environment for exploring data with WEKA. Experimenter is an environment for performing experiments and conducting statistical tests between learning schemes. Knowledge Flow is an environment which supports essentially the same functions as the Explorer but with a drag-and-drop interface. One advantage is that it supports incremental learning. Simple CLI 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[11].

6. Experiments Conducted

The above mentioned 10 classifiers are applied to the KDD Cup 20% training dataset comprising of 25192 instances through an experiment type of 10-fold cross validation. The experiment exactly took six hours to complete. The tests were configured with Paired T Tester (corrected) and the test of significance was taken as 0.05. The comparison fields Percent_correct, fmeasure, irrecall, irprecision and auc (area under roc) were taken for evaluation. Tests were also performed for ranking and summary. Apart

from ranking, all the classifiers are compared with each of the other classifiers as the baseline classifier. This revealed which classifier outperforms the other classifiers and also whether the classification is statistically significant or not. Different results were outputted and all these result sets are taken for interpretation of results and analysis.

7. Experiment's Results

First, the ranking of classification algorithms based on different performance measures is tabulated as shown in Table 11 and the corresponding chart is shown in Figure 11.

The percentage correct for each of the 10 classifiers as shown in the dataset row of the result is 53.39 for ZeroR, 96.27 for OneR, 96.55 for BayesNet, 89.54 for NaiveBayes, 99.47 for IBk, 94.44 for AdaBoost, 99.63 for Bagging, 99.6 for J48, 99.74 for Random Forest and 99.67 for SimpleCart.

7.1 Analysis of ZeroR as Baseline Classifier

The results show that all the classifiers are statistically better than the baseline classifier ZeroR at the significance level specified 0.05. It is also found that all the classifiers are better than ZeroR once and never equivalent to or worse than ZeroR. (1/0/0). ZeroR with different comparison fields is shown in Table 1 and Figure 1.

Table 1. Results of ZeroR with different comparison fields

Comparison Field	Value
Percent_correct	0.53
Fmeasure	0.70
Irprecision	0.53
Irrecall	1.00
AUC	0.50

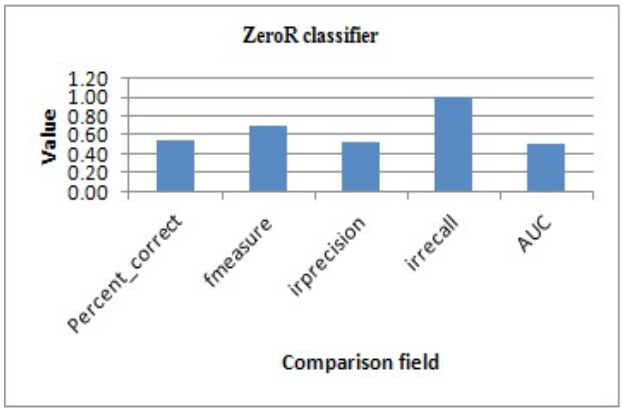

Figure 1. ZeroR classifier.

7.2 Analysis of OneR as Baseline Classifier

The results show that the classifiers IBk, Bagging, J48, Random Forest and SimpleCart are statistically better than the baseline classifier OneR at the significance level specified 0.05. The classifiers ZeroR, NaiveBayes, AdaBoost are statistically worse than OneR classifier. It is also observed that there is no statistical difference between OneR and BayesNet classifier. It is also shown from the results that the classifiers IBk, Bagging, J48, Random Forest and SimpleCart are better than OneR once and never equivalent to or worse than OneR. (1/0/0). The classifiers ZeroR, NaiveBayes and AdaBoost are not better than OneR. (0/0/1). OneR with different comparison fields is shown in Table 2 and Figure 2.

Table 2. Results of OneR with different comparison fields

Comparison Field	Value
Percent_correct	0.96
Fmeasure	0.96
Irprecision	0.99
Irrecall	0.94
AUC	0.96

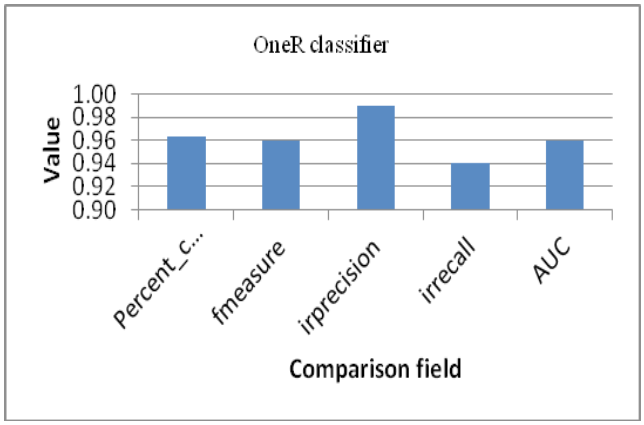

Figure 2. OneR classifier.

7.3 Analysis of BayesNet as Baseline Classifier

The results show that the classifiers IBk, Bagging, J48, Random Forest and SimpleCart are statistically better than the baseline classifier BayesNet at the significance level specified 0.05.

The classifiers ZeroR, NaiveBayes, AdaBoost are statistically worse than BayesNet classifier. It is also observed that there is no statistical difference between BayesNet and OneR classifier. It is also shown from the results that the classifiers IBk, Bagging, J48, Random Forest and SimpleCart are better than BayesNet once and never equivalent to or worse than BayesNet. (1/0/0). The classifiers ZeroR, NaiveBayes and AdaBoost are not better than BayesNet. (0/0/1). BayesNet with different comparison fields is shown in Table 3 and Figure 3.

Table 3. Results of BayesNet with different comparison fields

Comparison Field	Value
Percent_correct	0.97
Fmeasure	0.97
Irprecision	0.95
Irrecall	0.99
AUC	1.00

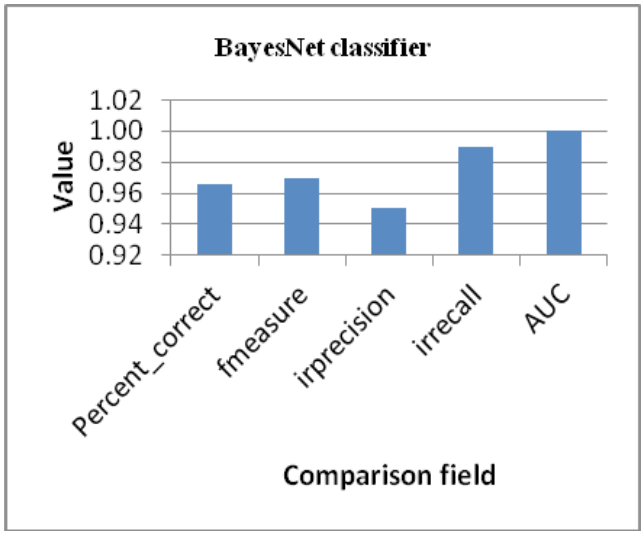

Figure 3. Bayes Net classifier.

7.4 Analysis of NaiveBayes as Baseline Classifier

The results show that all the classifiers except ZeroR are statistically better than the baseline classifer NaiveBayes at the significance level specified 0.05. It is also found that all the classifiers except ZeroR are better than NaiveBayes once and never equivalent to or worse than NaiveBayes. (1/0/0) ZeroR is not better than NaiveBayes. (0/0/1). NaiveBayes with different comparison fields is shown in Table 4 and Figure 4.

Table 4. Results of NaiveBayes with different comparison fields

Comparison Field	Value
Percent_correct	0.90
Fmeasure	0.90
Irprecision	0.89
Irrecall	0.91
AUC	0.97

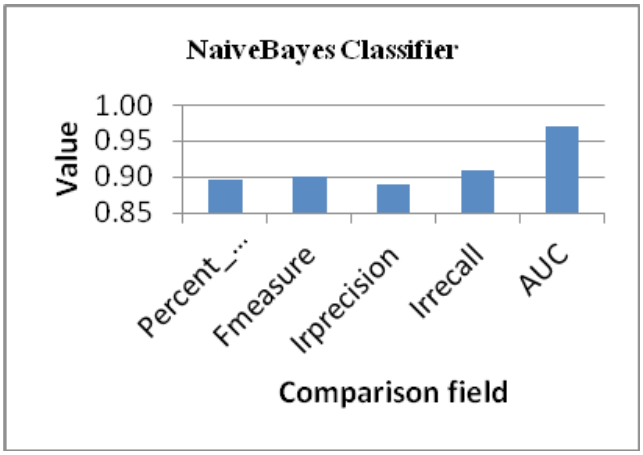

Figure 4. NaiveBayes classifier.

7.5 Analysis of IBk as Baseline Classifier

The results show that the classifiers Bagging, J48, Random Forest and SimpleCart are statistically better than the baseline classifier IBk at the significance level specified 0.05.

The classifiers ZeroR, OneR, BayesNet, NaiveBayes and AdaBoost are statistically worse than IBk classifier. It

is also shown from the results that the classifiers Bagging, J48, Random Forest and Simple Cart are better than IBk once and never equivalent to or worse than IBk. (1/0/0). The classifiers ZeroR, OneR, BayesNet, NaiveBayes and AdaBoost are not better than BayesNet. (0/0/1). IBk with different comparison fields is shown in Table 5 and Figure 5.

Table 5. Results of IBk with different comparison fields

Comparison Field	Value
Percent_correct	0.99
Fmeasure	1.00
Irprecision	0.99
Irrecall	1.00
AUC	0.99

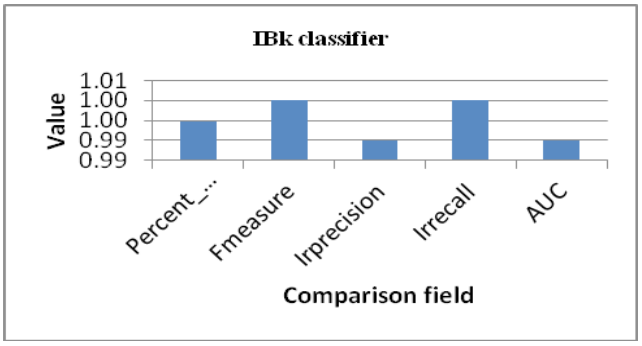

Figure 5. IBk classifier.

7.6 Analysis of AdaBoost as Baseline Classifier

The results show that the classifiers OneR, BayesNet, IBk, Bagging, J48, Random Forest and SimpleCart are statistically better than the baseline classifier AdaBoost at the significance level specified 0.05.

The classifiers ZeroR and NaiveBayes are statistically worse than AdaBoost classifier. It is also shown from the results that the classifiers OneR, BayesNet, IBk, Bagging, J48, Random Forest and Simple Cart are better than AdaBoost once and never equivalent to or worse than AdaBoost. (1/0/0). The classifiers ZeroR and NaiveBayes are not better than AdaBoost. (0/0/1). AdaBoost with different comparison fields is shown in Table 6 and Figure 6.

Table 6. Results of AdaBoost with different comparison fields

Comparison Field	Value
Percent_correct	0.94
Fmeasure	0.95
Irprecision	0.94
Irrecall	0.96
AUC	0.99

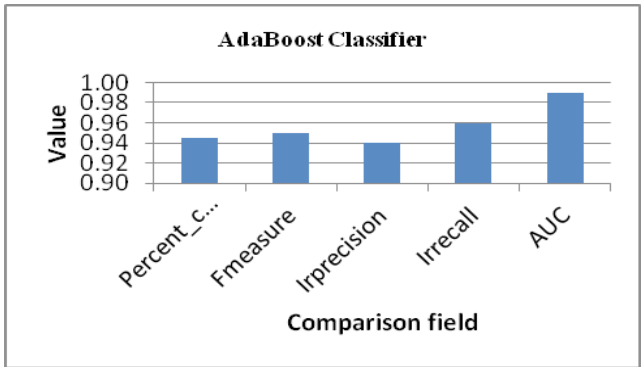

Figure 6. AdaBoost classifier.

7.7 Analysis of Bagging as Baseline Classifier

The results show that the classifier Random Forest is statistically better than the baseline classifier bagging at the significance level specified 0.05.

The classifiers ZeroR, OneR, BayesNet, NaiveBayes, IBk, AdaBoost are statistically worse than Bagging classifier. It is also observed that there is no statistical difference between J48 and SimpleCart and Bagging classifier. It is also shown from the results that the classifier Random Forest is better than Bagging once and never equivalent to or worse than Bagging. (1/0/0). The classifiers ZeroR, OneR, BayesNet, NaiveBayes, IBk, AdaBoost are not better than Bagging. (0/0/1). Bagging with different comparison fields is shown in Table 7 and Figure 7.

Table 7. Results of Bagging with different comparison fields

Comparison Field	Value
Percent_correct	0.99
Fmeasure	1.00
Irprecision	1.00
Irrecall	1.00
AUC	1.00

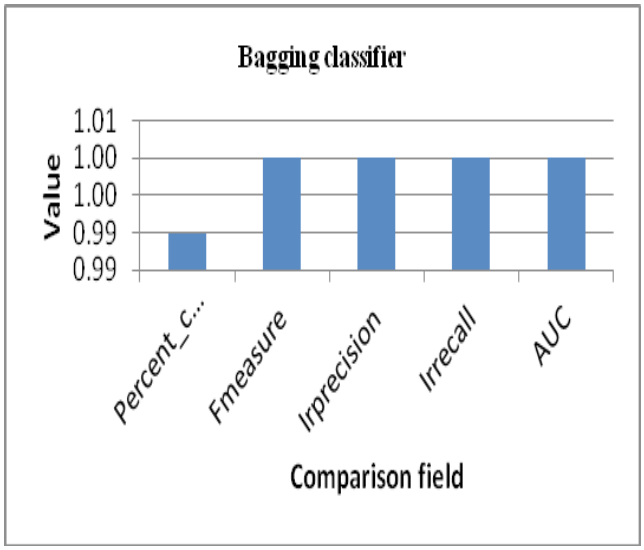

Figure 7. Bagging classifier.

7.8 Analysis of J48 as Baseline Classifier

The results show that the classifier Random Forest is statistically better than the baseline classifier J48 at the significance level specified 0.05.

The classifiers ZeroR, OneR, BayesNet, NaiveBayes, IBk, AdaBoost are statistically worse than J48 classifier. It is also observed that there is no statistical difference between Bagging and Simple Cart when compared to the baseline classifier J48. It is also shown from the results that the classifier Random Forest is better than J48 once and never equivalent to or worse than J48. (1/0/0). The classifiers ZeroR, OneR, BayesNet, NaiveBayes, IBk, AdaBoost are not better than J48. (0/0/1). J48 with different comparison fields is shown in Table 8 and Figure 8.

Table 8. Results of J48 with different comparison fields

Comparison Field	Value
Percent_correct	0.99
Fmeasure	1.00
Irprecision	1.00
Irrecall	1.00
AUC	1.00

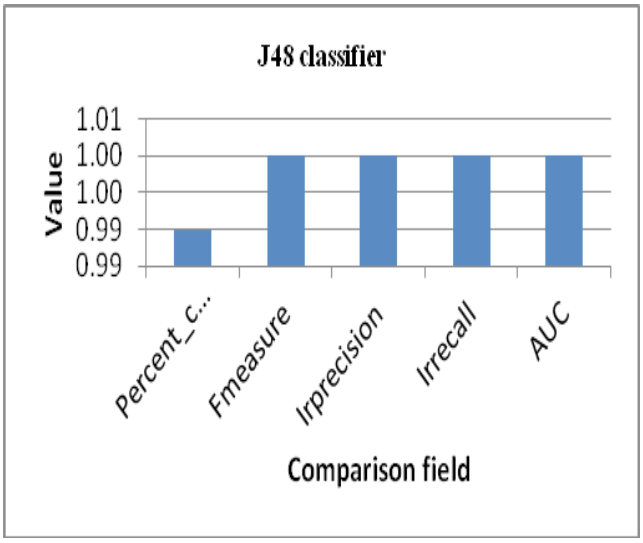

Figure 8. J48 classifier.

7.9 Analysis of Random Forest as Baseline Classifier

The results show that none of the classifiers are statistically better than the baseline classifier Random Forest at the significance level specified 0.05.

The classifiers ZeroR, OneR, BayesNet, NaiveBayes, IBk, AdaBoost, Bagging, J48 are statistically worse than Random Forest classifier. It is also observed that there is no statistical difference between SimpleCart and Random Forest. It is also shown from the results that the classifiers ZeroR, OneR, BayesNet, NaiveBayes, IBk, AdaBoost, Bagging and J48 are not better than Random Forest. (0/0/1). Random forest with different comparison fields is shown in Table 9 and Figure 9.

Table 9. Results of Random Forest with different comparison fields

Comparison Field	Value
Percent_correct	1.00
Fmeasure	1.00
Irprecision	1.00
Irrecall	1.00
AUC	1.00

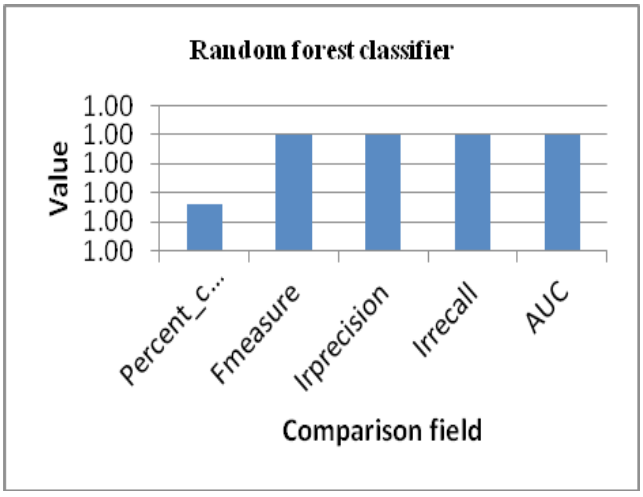

Figure 9. RandomForest classifier.

Table 10. Results of SimpleCart with different comparison fields

Comparison Field	Value
Percent_correct	1.00
Fmeasure	1.00
Irprecision	1.00
Irrecall	1.00
AUC	1.00

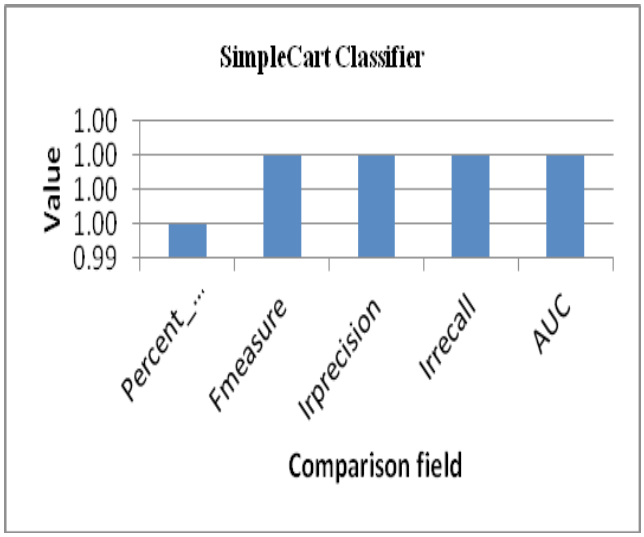

Figure 10. SimpleCart classifier.

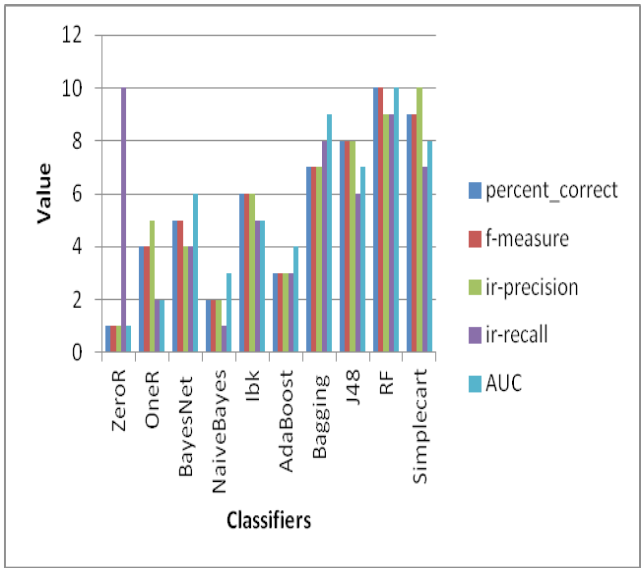

Figure 11. Ranking of the classifiers in terms of different comparison fields.

7.10 E Analysis of SimpleCart as Baseline Classifier

The results show that none of the classifiers are statistically better than the baseline classifier Simple Cart at the significance level specified 0.05.

The classifiers ZeroR, OneR, BayesNet, NaiveBayes, IBk and AdaBoost are statistically worse than Simple Cart classifier. It is also observed that there is no statistical difference exists between Bagging, J48, RandomForest and

Table 11. Ranking of the classifiers in terms of different comparison fields

Comparison Field	ZeroR	OneR	BayesNet	NaiveBayes	Ibk	AdaBoost	Bagging	J48	RF	Simplecart
percent_ correct	1	4	5	2	6	3	7	8	10	9
f-measure	1	4	5	2	6	3	7	8	10	9
ir-precision	1	5	4	2	6	3	7	8	9	10
ir-recall	10	2	4	1	5	3	8	6	9	7
AUC	1	2	6	3	5	4	9	7	10	8

SimpleCart classifier. It is also shown from the results that the classifiers ZeroR, OneR, BayesNet, NaiveBayes, IBk and AdaBoost are not better than Simple Cart. (0/0/1). SimpleCart with different comparison fields is shown in Table 10 and Figure 10.

8. Conclusion

The ranking of the classifiers in terms of different comparison fields are given in Table 11 and Figure 11.

As per the results obtained by the Weka Experimenter with the 10 classifiers on the KDD 20% training dataset, it has been analysed that Random forest classifier works best with the comparison fields percent_correct, fmeasure and AUC (Area under ROC). Simplecart classifier ranks next to Random forest classifer with the comparison fields percent_correct and fmeasure. Simple cart classifier outperforms all other classifiers with respect to the comparison field irprecision. ZeroR is found to be the worst classifier in terms of all the comparison fields other than irrecall. With irrecall as the comparison field, ZeroR ranks first when compared to all the other classifiers.

J48 classifier stands next to Simple cart classifier with the comparison fields percent_correct, fmeasure and irprecision. In this work, the ten different classifiers are applied on the KDD20% training dataset with different comparison fields using the Weka tool experimenter.

Thus it has been found that with the dataset that is taken for experiment, further detailed study could be restricted only with the five classifiers Random Forest, Simplecart, J48, Bagging and IBk. This will definitely reduce computational time and increase the efficiency of classification of data set. Also the reason for the ZeroR classifier's performance with irrecall comparison field has to be studied.

9. References

1. Adetunmbi AO, Falaki SO, Adewale OS, Alese BK. Network Intrusion Detection based on Rough Set and k-Nearest Neighbour. International Journal of Computing and ICT Research. 2008; 2(1):60–6. Available from: http://www.ijcir.org/volume1number2/article7.pdf
2. Ranjan R, Sahoo G. A new clustering approach for anomaly intrusion detection. International Journal of Data Mining and Knowledge Management Process. 2014 Mar; 4(2):29–38.
3. Azad C, Jha VK. Data Mining based Hybrid Intrusion Detection System. Indian Journal of Science and Technology. 2014 Jun; 7(6):781–9.
4. Khor K-C, Ting C-Y, Amnuaisuk S-P. From Feature Selection to Building of Bayesian Classifiers: A Network Intrusion Detection Perspective. American Journal of Applied Sciences 2009; 6(11):1948–59.
5. Lee W, Stolfo SJ, Mok KW. Algorithms for Mining System Audit Data. Proc KDD; 1999.
6. Ghali NI. Feature Selection for Effective Anomaly Based Intrusion Detection. International Journal of Computer Science and Network Security. 2009 Mar; 9(3):285–9.
7. KDD CUP 1999 DATASET: Available from: http://kdd.ics.uci.edu/databases/kddcup99/

8. SANS Institute InfoSec Reading Room. Understanding Intrusion Detection Systems; 2001.
9. Wu X, Kumar V, Ross Quinlan RJ, Ghosh J, Yang Q, Motoda H, McLachlan GJ, Ng A, Liu B, Yu PS, Zhou Z-H, Steinbach M, Hand DJ, Steinberg D. Top 10 algorithms in data mining. London: Springer-Verlag; 2008. p. 1–3. DOI: 10.1007/s10115-007-0114-2
10. Venkata Lakshmi S, Edwin Prabakaran T. Application of k-Nearest Neighbour Classification Method for Intrusion Detection in Network Data. International Journal of Computer Applications (0975-8887); 2014 Jul; 97(7):34–7.
11. Weka Manual. Available from: http://www.ittc.ku.edu/~nivisid/WEKA_MANUAL.pdf