Application of Feature Selection Methods and Ensembles on Network Security Dataset

Neeraj Bisht
Birla Institute of Applied Sciences,Bhimtal, India

Amir Ahmad
King Abdul Aziz University, Rabigh, Saudi Arabia

Shilpi Bisht
Birla Institute of Applied Sciences,Bhimtal, India

ABSTRACT
Generally intrusion detection systems (IDS) use all the data features to classify normal and anomaly packet. It has been observed in the studies that some of the data features may be redundant or are less important in this classification process. Authors have studied NSL KDD dataset with different feature selected from Gain Ratio and Chi- Square feature selection methods and carried out the experiments with single Decision Tree and then applied ensemble with Random Forests and Decision Tree with Bagging. Results show that significant feature selection is very important in the design of a lightweight and efficient intrusion detection system. Random Forests are better than Single Decision Tree and Decision Tree with Bagging for the current dataset. Performance of Gain Ratio is better than Chi square feature selection method for this dataset.

Keywords
Network security, NSL KDD, classifier, ensembles, Decision trees, Random Forests, Chi Square, Gain Ratio.

1. INTRODUCTION
The rapid advancement of internet is a boon to the society but it has also created various security related issues. The major challenge of the network administrator is to detect the policy variations. Intrusion detection systems (IDS) are the fine grain filter placed inside the protected network, looking for known or potential fears in network traffic and/or audit data recorded by hosts. Recently the researchers have given intrusion detection approaches which are based on data mining algorithms trained on malicious and normal traffic activities [1, 2, 3, 4, 5]. It helps in deciding the "boundaries" between normal and malicious network traffic. These models are trained on the historical data and are used to predict the type of the new traffic activity.

There are so many factors which affect the success of machine learning on a given task. The demonstration and quality of the example data is most important. Theoretically, having more features should result in more selective power. However, practical experience with machine learning algorithms has shown that this is not always the case. Feature subset selection is the process of identifying and removing as much irrelevant and redundant information as possible. This reduces the dimensionality of the data and may allow learning algorithms to operate faster and more effectively. In some cases, accuracy on future classification can be improved; in others, the result is a more compact, easily interpreted representation of the target concept.

Different classification techniques like Decision Trees, Naïve Bayes, Neural Networks, Support Vector Machines are used to classify normal and anomalous data [1, 2, 3, 4, 5]. Ensembles are a blend of several base models and the final categorization depends on the collective outputs of individual models [6, 7]. Classifier ensembles have shown to construct better results than a single model, provided the classifiers are precise and diverse.

A number of different methods have been proposed to build decision tree ensembles. Randomization is introduced to build different decision trees. Bagging[8, 9] bring in randomization by manipulating the training data supplied to each classifier. Breiman [10] combines Random Subspaces technique with Bagging to create Random Forests. To build a tree, it uses a bootstrap imitation of the training sample, then during the tree budding phase, at each node the best split is selected from a random subset of size K of candidate features.

In this paper, we propose the methods of intrusion detection in computer networks based on feature selection and ensemble technique. This approach is motivated by the observation that generally a combination of classifiers performs better than a single classifier. Feature selection further improves the predictive performance of the model.

This paper is divided into four sections. Data set used for the study, the classifier and feature selection methods are discussed in Section 2. Section 3 deals with the experiments and discussions. Section 4 includes conclusion and future work.

2. MATERIAL AND METHODS
Decision Tree is an admired classifier. In this paper, the experiments are carried out with this classifiers and ensembles of this classifier on all the 41 features of dataset [16], 20 features of dataset selected by Chi Square feature selection method and 20 features of dataset selected by Gain Ratio feature selection method. In this section, dataset and different methods that are used in this paper are discussed briefly.

2.1 Dataset
Modified KDD CUP 99 anomaly detection dataset is used in this paper. Tavallaee et al. has given one modified KDD training datasets and two testing datasets [11]. The above dataset consists of total 41 attributes for each connection record and one class label. The class label is either anomaly or normal.

Three tables are taken from the dataset and named Training dataset, Type 1 testing dataset and Type 2 testing dataset. The details of these datasets are given in the next subsection.

Table 1: Network data features

duration	urgent	file creations	serror_rate	dst_host_srv_cou	dst_host__srv_re
protocol type	hot	shells	srv_serror_rate	dst_host_same_s	
Service	Failed_login	access files	rerror_rate	dst_host_diff_srv	
src bytes	logged in	outbound cmds	srv_rerror_rate	dst_host_same_s	
dst bytes	compromised	hot login	same_srv_rate	dst_host_srv_diff	
flag	root shell	guest login	diff_srv_rate	dst_host_serror_r	
land	su_attempted	count	srv_diff_host_rat	dst_host_srv_ser	
wrong_fragment	num_root	srv_count	dst_host_count	dst_host_rerror_r	

2.1.1 Training Dataset
This is obtained by removing all the redundant records from the original KDD Cup'99 training dataset. We have used this new training dataset for training the classifiers.

2.1.2 Type 1 Testing Dataset
This dataset is created by removing all the redundant records from the testing dataset, after that 21 classifiers are used to divide the testing dataset and into 5 groups on the basis of prediction difficulty. A new testing dataset is created by selecting data points from each group such that the number of data points selected from each group were inversely proportional to the number of data points in that group.

2.1.3 Type 2 testing dataset
Any data point which is correctly classified by all 21 classifiers is not included in this dataset. This testing dataset was expected to be the most difficult dataset.

2.2 Decision Trees
Decision trees are extremely popular tools for classification [12, 13]. The beauty of decision trees is due to the fact that decision trees symbolize rules. Rules can readily be expressed so that humans can understand them. A Decision Tree is in the form of a tree structure, where each node is either a *leaf node* (it indicates the value of the target class of examples) or a *decision node* (it specifies some test to be carried out on a single attribute-value), with two or more than two branches and each branch has a sub-tree. A decision tree can be used to classify an example by starting at the root of the tree and moving through it until a leaf node, which provides the rules for classification of the example.

2.3 Bagging
Bagging (Bootstrap Aggregation) [8] generates different bootstrap training datasets from the original training dataset and uses each of them to train one of the classifiers in the ensemble. For example, to create a training set of *N* data points, it selects one point from the training dataset, *N* times without replacement. Each point has equal probability of selection. In one training dataset, some of the points get selected more than once, whereas some of them are not selected at all. Different training datasets are created by this process. When different classifiers of the ensemble are trained on different training datasets, diverse classifiers are created. Bagging does more to reduce the variance part of the error of the base classifier than the bias part of the error.

2.4 Random Forests
Random Forests are very popular decision tree ensembles [10]. It combines Bagging with random subspace. For each Decision Tree, a dataset is created by Bagging procedure. During the tree growing phase, at each node, n attributes are selected randomly and the node is split by the best attribute from these n attributes. Breiman showed that Random Forests are quite competitive to Adaboost. However, Random Forests can handle mislabeled data points better than Adaboost can. Due to its robustness of the Random Forests, they are widely used.

2.5 Gain Ratio
A decision tree is a simple structure where non-terminal nodes represent tests on one or more attributes and terminal nodes reflect decision outcomes. The information gain measure is used to select the test attribute at each node of the decision tree. The information gain measure prefers to select attributes having a large number of values. The basic decision tree induction algorithm ID3 [14] was enhanced by C4.5 [9, 13]. C4.5 a successor of ID3, uses an extension of information gain known as gain ratio, which attempts to overcome this bias. Gain ratio takes number and size of branches into account when choosing an attribute. It corrects the information gain by taking the intrinsic information of a split into account. Intrinsic information is entropy of distribution of instances into branches (i.e. how much info do we need to tell which branch an instance belongs to). Value of attribute decreases as intrinsic information gets larger [15].

2.6 Chi- Square
Chi-Square (χ^2) is one of the commonly used methods of feature selection. The χ^2 method evaluates features individually by measuring their chi-squared statistic with respect to the classes. For a numeric attribute, the method requires its range to be discretized into several intervals. The χ^2 value of an attribute is defined as:

$$\chi^2 = \sum_{i=1}^{m}\sum_{j=1}^{n}\frac{(O_{ij} - E_{ij})^2}{E_{ij}}$$

Where, m is the number of intervals, n the number of classes,

O_{ij} is the number of samples in the ith interval, jth class,

$E_{ij} = \frac{R_i * C_j}{N}$,is the expected frequency of O_{ij}

R_i the number of samples in the ith interval,

C_j the number of samples in the jth class, N the total number of samples

Degree of freedom of Chi Square Test is (m-1)(n-1).

3. EXPERIMENTS AND DISCUSSION

All the experiments were carried out by using WEKA software[16]. Experiments are performed with single Decision Tree, Bagging and Random Forests modules. For Bagging experiments are carried out with J48 tree (the implementation of C4.5 tree). As the training dataset was large, the size of the ensembles was set to 10. All the other default parameters were used in the experiments. We also carried out experiment with single J48 tree. Following performance measures are used to compare different classifiers.

3.1 Performance Measures

These are the various parameters define to evaluate the performances of classification techniques.

(1) Time = Time to build model with training data.

(2) Accuracy= $\frac{TP+TN}{n}\times 100$ where, n= Total number of data points.

(3) F-Measure is given by,

$F(r,p)=\frac{2rp}{r+p}$, where, $r=\frac{TP}{TP+FN}\times 100$, called Recall or Senstivity, and $p=\frac{TP}{TP+FP}\times 100$, called Precision.

(4) TP is the number of true positive (Attack is predicted correctly.

(5) TN is the number of true negative (Normal is predicted correctly).

(6) FP is the number of false negative (Normal is predicted as Attack).

(7) FN is the number of false negative (Attack is predicted as Normal)

3.2 Results

Results of the experiments are disused in the following section:

3.2.1 Single Decision Tree

It is clear from the results shown in table 2 that the accuracy and F- Measure is almost same for all the feature variable sets for training data. Model building time is greatly reduced if we compare 20 features extracted from Chi Square test with complete 41 features set. It is further reduced for the 20 features selected from Gain ratio test.

For Test 1 and Test 2 data sets accuracy and F- Measure of complete dataset is better than features selected from Chi Square and Gain Ratio feature selection methods. The performance of Gain Ratio is better than Chi Square for Test1 and Test 2 data.

3.2.2 Random Forest

Results for Random Forest in table 3 show 100% Accuracy and F- Measure for all the three feature sets for training data. Model building time is reduced for 20 features as compared with 41 features. Accuracy and F-Measure of Gain Ratio is batter then other two for Test 1 and Test 2 data. Complete set of 41 features performed slightly better than 20 features selected from Chi Square test.

3.2.3 Decision Tree with Bagging

Results for Decision Tree with Bagging are showing equal Accuracy and F-Measure for all the three feature sets for training data. Model building time for 41 variables is almost double if compared with 20 variables selected from Chi Square test. This time is further reduced for the features selected from Gain Ratio test. Accuracy and F-Measure of reduced features is less than complete features for Test 1 and Test 2 data. Feature Set selected from Gain ratio has performed better than the Feature set selected from Chi Square test for Test 1 and Test 2 data.

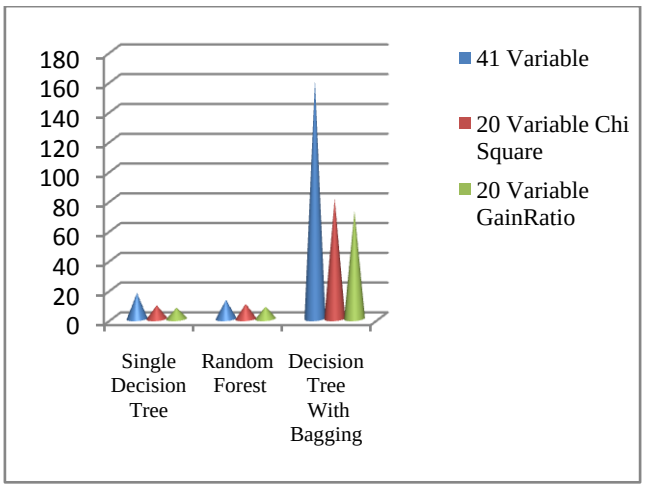

Fig 1: Model building time in seconds with training data

4. CONCLUSION AND FUTURE WORK:

No feature selection and classification method (among the feature selection methods and classifiers studied) is best for all the performance measure. Hence, one has to decide the performance measure carefully in order to compare different classifiers and feature selection methods. Random Forests are better than Single Decision Tree and Decision Tree with Bagging for the current dataset. Performance of Gain Ratio is better than Chi square feature selection method for this dataset. Performance of features selected from Gain ratio is better than the performance of complete feature set for random Forest. Model building time is greatly reduced when we reduced features in the dataset. Dataset with features selected from Gain ratio feature selection method took lowest model building time.

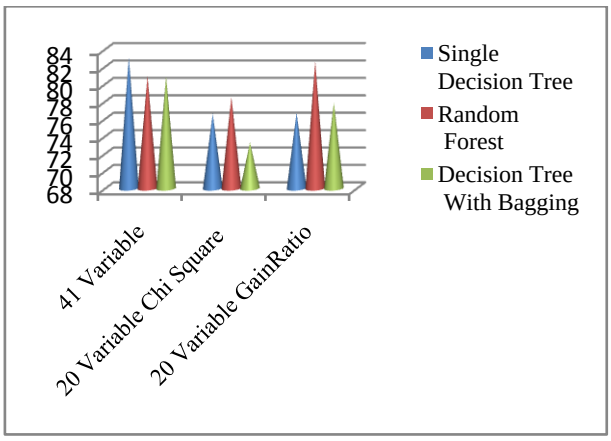

Fig 2: F-Measure for test 1 data

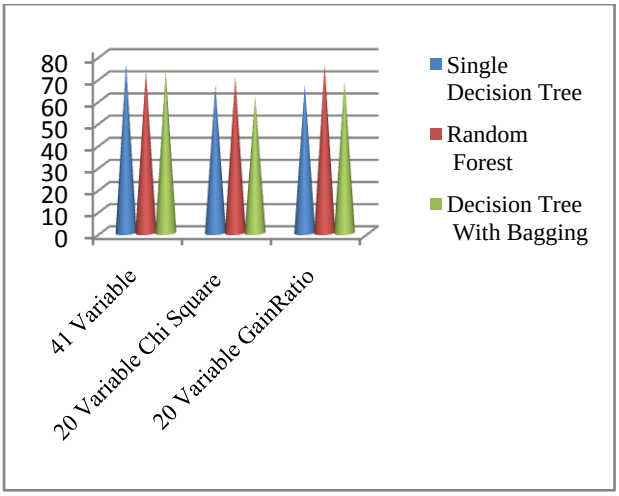

Fig 3: F-Measure for test 2 data

The network security datasets are quite large. The training of ensembles on these datasets takes a lot of time. As we observed that the performance of the same dataset with reduced features is quite good, even better sometime, one may use reduced feature set if the performance requirements are not very strict (the best performance).

In future, we will use other feature selection methods ensemble methods and other classifiers and compare the results for our study.

Table 2: Results For Single Decision Tree

	Training Data			Test 1 Data		Test 2 Data	
No of Features / Feature Selection Method	Time	Accuracy	F-Measure	Accuracy	F-Measure	Accuracy	F-Measure
41 Variable	17.36	99.8	99.8	82.3	83.1	68.1	76.6
20 Variable Chi Square	9.13	99.7	99.7	77.9	76.6	58	66.8
20 Variable GainRatio	7.44	99.8	99.8	78	76.7	58.1	66.9

Table 3: Results For Random Forest

	Training Data			Test 1 Data		Test 2 Data	
No of Features / Feature Selection Method	Time	Accuracy	F-Measure	Accuracy	F-Measure	Accuracy	F-Measure
41 Variable	12.84	100	100	81.3	80.9	64.6	73.3
20 Variable Chi Square	10.09	100	100	78.6	78.5	59.8	70.4
20 Variable GainRatio	8.2	100	100	82.6	82.6	67.2	76.1

Table 4: Results For Decision Tree with Bagging

	Training Data			Test 1 Data		Test 2 Data	
No of Features / Feature Selection Method	Time	Accuracy	F-Measure	Accuracy	F-Measure	Accuracy	F-Measure
41 Variable	160.56	99.8	99.8	81.3	80.9	64.5	73.3
20 Variable Chi Square	81	99.8	99.8	75.5	73.3	53.4	61.8
20 Variable GainRatio	72.67	99.8	99.8	79	77.9	60.1	68.8

5. REFERENCES

[1] Amor, N.B., Benferhat, S., and Elouedi, Z. 2004. Naïve Bayes vs. Decision Trees in Intrusion Detection Systems, Proceedings of ACM Symposium on Applied Computing, Nicosia, Cyprus.

[2] Gaddam, S.R., Phoha, V.V., and Balagani, K.S. 2007. Means+id3 a novel method for supervised anomaly detection by cascading k-means clustering and id3 decision tree learning methods, IEEE Trans Knowl and Data Engg , 19:(3), 345-354.

[3] Horng, S.J., Su, M.Y., Chen, Y.H., Kao, T.W., Chen R.J., Lai, J.L., and Perkasa, C.D. 2011. A novel intrusion detection system based on hierarchical clustering and support vector machines, Expert Syst with Appl, 38:(1), 306-313.

[4] Sabhnani, M., and Serpen, G. 2003. Application of machine learning algorithms to KDD intrusion detection dataset within misuse detection context, Proceedings of Conference on Machine Learning Models, Technology and Application, 209-215, MLMTA.

[5] Tajbakhsh, A., Rahmati, M., and Mirzaei, A. 2009 Intrusion detection using fuzzy association rules, Appl Soft Comput , 9:(2), 462-469.

[6] Hansen, L.K., and Salamon, P. 1990. Neural network ensembles, IEEE Trans Patt Anal Mach Intel, 12, 993-1001.

[7] Kuncheva, L.I. 2004. Combining pattern classifiers: Methods and Algorithms, Wiley-IEEE Press, New York.

[8] Breiman, L. 1996. Bagging predictors, Machine Learning, 24:(2), 123–140.

[9] Quinlan, J.R. 1996. Bagging, Boosting and C4.5, In Proc. 13th National Conf. Back Propagation Intelligence (AAAI'96), Portland, 725-730.

[10] Breiman, L. 2001. Random Forests, Machine Learning, 45:(1), 5–32.

[11] Tavallaee, M.E., Bagheri, W.L., and Ghorbani, A. 2009. A Detailed Analysis of the KDD CUP 99 Data Set, Proceedings of the Second IEEE Symposium on Computational Intelligence for Security and Defense Applications (CISDA), Piscataway, NJ, USA, 53-58.

[12] Breiman, L., Friedman, J.H., Olshen, R., and Stone, C. 1984. Classification and Regression Trees, Chapman and Hall, London.

[13] Quinlan, J.R. 1993. C4.5: Programs for machine learning, Morgan Kaufmann, San Mateo.

[14] Quinlan, J.R. 1986. Induction of Decision Trees, Machine Learning1, Kluwer Academic Publishers, Boston, 81-106

[15] Han, J. and Kamber, M. 2001. Data Mining Concepts and Techniques. Morgan Kaufmann.

[16] Witten, I.H., and Frank, E. 2000. Data Mining: Practical Machine Learning Tools with Java Implementations, Morgan Kaufmann, San Francisco.