IOSR Journal of Engineering
May. 2012, Vol. 2(5) pp: 961-965

A STUDY OF THE MODIFIED KDD 99 DATASET BY USING CLASSIFIER ENSEMBLES

Mohammed J. Alhaddad, Amir Ahmed, Sami M. Halawani, Abdulrahman H. Altalhi, Abdulfattah S. Mashat

Department of Information Technology, Faculty of Computing and Information Technology,King Abdul Aziz University, Jeddah, Saudi Arabia

ABSTRACT

Network security has been an important research area. KDD 99 dataset [1] has been used to analyze various network security methods. However, it has been shown that this dataset has redundant data points that make the analysis bias for these data points. New modified data sets are proposed that overcome these weaknesses. We carried out the experiments with different classifiers on this datasets to study the applicability of different classification methods for this dataset. Naïve Bayes and decision trees and their ensemble methods are used for this study. We used different performance measures in our study. Results suggest that no single classification method is the best for all types of datasets on all type of performance measures. The comparative performance suggests that classifiers based on decision tree performed better than classifiers based on naïve Bayes. Results also suggest that single decision tree is a good classifier for this data as it has reasonable classification accuracy and less training and testing time.

1. INTRODUCTION

Nowadays an increasing number of commercial and public services are offered trough Internet, so that security is becoming one of the key issues [2]. The so-called "attacks" to internet service providers are carried out by exploiting unknown weaknesses or bugs always contained in system and application software. Computer networks are usually protected against attacks by a number of access restriction policies that act as a coarse grain filter. Intrusion detection systems (IDS) are the fine grain filter placed inside the protected network, looking for known or potential threats in network traffic and/or audit data recorded by hosts. Some of the IDS detection techniques are signature based techniques [3]. In these techniques, the signatures of the known attacks are maintained. When a new pattern comes, its signature is compared with these stored signatures to predict whether the given pattern is normal or attacks. This method is fast, however, it can identify only known attacks.

Researchers recently proposed intrusion detection approaches based on data mining algorithms trained on malicious and normal traffic activities [4-8]. It allows designing decision "boundaries" between normal and malicious network traffic. In these methods models are trained on the historical data and these modes are used to predict the type of the new traffic activity. Different classifiers like decision trees, naïve Bayes, neural networks, support vector machines have been used to classify normal and anomalous [4 -8]. Ensembles are a combination of multiple base models the final classification depends on the combined outputs of individual models [9-15]. Classifier ensembles have shown to produce better results than single models provided the classifiers are accurate and diverse. Ensembles perform best when base models are unstable-classifiers whose output undergoes significant changes in generalization with small changes in the training data-decision trees and neural networks are in this class.

During the last decade, anomaly detection has attracted the attention of many researchers to overcome the weakness of signature-based IDSs in detecting novel attacks, and KDDCUP'99 is the mostly widely used data set for the evaluation of these systems. Tavallaee, et al. [16] conducted a statistical analysis on this data set. They found that the dataset has many redundant data points. That makes data mining methods bias for the data points that are repeated. The authors have proposed a new data set, which consists of selected records of the complete KDD data set. This data set is publicly available for researchers through their website and has the following advantages over the original KDD data set:

1. It does not include redundant records in the train set, so the classifiers will not be biased towards more frequent records.
2. There is no duplicate records in the proposed test sets; therefore, the performance of the learners are not biased by the methods which have better detection rates on the frequent records.
3. The number of selected records from each difficulty level group is inversely proportional to the percentage of records in the original KDD data set. As a result, the classification rates of distinct machine learning methods vary in a wider range, which makes it more efficient to have an accurate evaluation of different learning techniques.
4. The number of records in the train and test sets are reasonable, which makes it affordable to run the experiments on the complete set without the need to randomly select a small portion. Consequently, evaluation results of different research works will be consistent and comparable.

In this paper, an approach to intrusion detection in computer networks based onmultiple classifier systems will be studied. This approach is motivated by the observation that generally a combination of classifiers performs better than a single classifier. Hence, we will study the use of classifier ensemble to predict intrusion detection.
In the Section 2, we discuss the classifier methods and the data sets used for the study. The section 3 has experiments and discussion. Section 4 has conclusion and future work.

2. METHODS AND MATERIAL

Decision trees and naïve Bayes are two popular classifiers. In this paper, we carried out experiments with these classifiers and ensembles of these classifiers. In this section, we discuss about different methods that we use in the paper.

2.1. Decision Trees

Decision trees are very popular tools for classification [17-18]. The attractiveness of decision trees is due to the fact that decision trees represent rules. Rules can readily be expressed so that humans can understand them. 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.

2.2. Naive Bayes

A Naive Bayes classifier is a probabilistic classifier [19]. It is based on conditional probabilities computed by using naive Bayes theorem. Despite a strong independence assumption (all attributes are independent), it has shown excellent performance over a variety of datasets.

2.3. Bagging

Bagging (Bootstrap Aggregation) [11] generates different bootstrap training datasets from the original training dataset and uses each of them to train one of the classifiers in the ensemble. 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. Adaboost.M1

Boosting [12] generates a sequence of classifiers with different weight distribution over the training set. In each iteration, the learning algorithm is invoked to minimize the weighted error, and it returns a hypothesis. The weighted error of this hypothesis is computed and applied to update the weight on the training examples. The final classifier is constructed by a weighted vote of the individual classifiers. Each classifier is weighted according to its accuracy on the weighted training set that it has trained on.

2.5. Random Forests

Random Forests are very popular decision tree ensembles [15]. 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, k attributes are selected randomly and the node is split by the best attribute from these k attributes. Due to its robustness of the Random Forests, they are widely used

2.6. Dataset

In this paper, we are using the modified KDD anomaly detection datasets. Tavallaee et al. presented one modified KDD training datasets and two testing datasets [16]. We will call them type1 training dataset and type 2 dataset. The details of these datasets are presented below.

2.6.1 Training dataset

To solve the problems discussed in the Section 1, they removed all the redundant records from the training data. We used this new training dataset for training the classifiers.

2.6.2 Type 1 testing dataset

They first removed the all the redundant records from the testing dataset. They used 21 classifiers to find divide the testing dataset and into 5 groups on the basis of prediction difficulty. They did the same exercise for the testing dataset. A testing dataset is created by selecting data points from each group such that the number of data points selected from each group were inversely proportion the number of data points in that group.

2.6.3 Type 2 testing dataset

In this test dataset, they did not include any of the data points that had been correctly classified by all 21 classifiers. This testing dataset was expected to be the most difficult dataset.

3. EXPERIMENTS AND DISCUSSION

All the experiments were carried out by using WEKA software [20]. We did the experiments with Bagging, AdaBoost.M1 and Random Forests modules. For the Bagging and AdaBoost.M1 modules, we carried out experiments with J48 tree (the implementation of C4.5 tree) and Naive Bayes classifier as the base classifier. 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 and single naïve Bayes classifier. We used following performance measures to compare different classifiers.

3.1. Performance measures

We define various parameters to evaluate the performances of various classification techniques.

Sensitivity = (TP/(TP + FN))x100%

Specificity = (TN/(FP + TN))x100%

Accuracy = ((TP + TN)/(Total number of data points))x100%

TP is the number of true positive (attack is predicted correctly)

The high sensitivity is most desirable as we do not want any attacks go unnoticed.

3.1.1 Area under ROC curve

The ROC curve defines the plots of true positive and false positive for a two class classifier as its discrimination threshold is varied. Area under cures are used to evaluate the performance of the classifier. Higher value means better performance.

3.2. Results

We carried out testing with three types of testing datasets discussed in the last section.

3.2.1 Training data as the testing data

We carried out experiments with the training dataset as the testing datasets. Results are presented in Table 1 and Table 2. As expected, all the classifier performed well for this testing dataset. However, performance of ensembles based on decision trees are better than ensembles based on naïve Bayes classifiers. AdaBoost.M1 with decision trees and Random Forests are the best classifier for this problem. Adaboost.M1 is the best methods among classifiers based on naïve Bayes classifier. Hence, AdaBoost.M1 is the best ensemble method among the ensemble methods studied.

Classifier	Sensitivity	Specificity	Accuracy	ROC
Single	99.8	99.8	99.80	1.000
Random Forests	100.0	100.0	100.00	1.000
Bagging	99.8	99.9	99.99	1.000
AdaBoost.M1	100.0	100.0	100.00	1.000

Table 1 Classification results for training data as testing data for classifiers based on decision tree. Results are presented in % classification accuracy

Single	87.3	91.3	89.43	0.963
Bagging	87.6	91.0	89.39	0.965
AdaBoost.M1	96.0	92.3	94.00	0.988

Table 2 Classification results for training data as testing data for classifiers based on naïve Bayes. Results are presented in % classification accuracy.

Classifier	Sensitivity	Specificity	Accuracy	ROC
Single	68.9	97.2	81.05	0.874
Random Forests	66.5	97.1	79.68	0.940
Bagging	67.4	97.3	80.26	0.922
AdaBoost.M1	65.5	97.2	79.11	0.936

Table 2 Classification results for type 1 testing data for classifiers based on decision trees. Results are presented in % classification accuracy**.**

Classifier	Sensitivity	Specificity	Accuracy	ROC
Single	64.3	92.8	76.56	0.918
Bagging	64.6	92.7	76.71	0.919
AdaBoost.M1	62.5	92.0	75.22	0.884

Table 3 Classification results for type 1 testing data for classifiers based on naïve Bayes. Results are presented in % classification accuracy.

Classifier	Sensitivity	Specificity	Accuracy	ROC
Single	58.8	87.3	63.97	0.771
Random Forests	55.7	87.1	61.35	0.805
Bagging	56.8	87.9	62.46	0.823
AdaBoost M1	54.3	87.2	60.27	0.820

Table 4 Classification results for type 2 testing data for classifiers based on decision tree. Results are presented in % classification accuracy.

Classifier Ensembles	Sensitivity	Specificity	Accuracy	ROC
Single	53.1	67.8	55.77	0.650
Bagging	53.5	67.4	56.04	0.652
AdaBoost.M1	50.4	65.1	53.08	0.671

Table 5 - Classification results for type 2 testing data for classifiers based on naïve Bayes. Results are presented in % classification accuracy.

3.2.2 Type 1 testing data

Results for these experiments are presented in Table 3 and Table 4. The performance of all classifiers all not as good as in the last case (the training dataset as the testing dataset). As some of the attacks in the testing datasets are not exactly the same as in the training dataset, the classifiers have difficulty in predicting these attacks. Decision trees ensembles perform better than naïve Bayes ensembles. The accuracy of the ensembles created by using the AdaBoost.M1 is less than the accuracy of single classifiers. As AdaBoost.M1 has problem in learning the mislabeled training data points, the low accuracy of these ensembles suggests that the training dataset has mislabeled data points.

3.2.3 Type 2 testing data

Results are presented in Table 5 and Table 6. As discussed in the last section, this is the most difficult testing dataset to predict. The performances of all classifiers are quite poor as compared to the other two cases. This is due the fact that the level of difficulty (for prediction) for this dataset is more for this dataset. In this case also, single decision tree has the best accuracy and sensitivity. Whereas, Bagging method has the best the AUC under ROC. Classifiers based on decision trees performed better than classifiers based on naïve Bayes classifiers

3.3. Summary of the results

We observed following points from our experiments;

1. No classifier (among the classifiers we studied) is best for all the performance measure. Hence, one has to decide the performance measure carefully in order to compare different classifiers.
2. Decision tree ensembles perform better than naive Bayes ensembles. Hence, decision tree ensembles should be preferable.
3. In real life, most of the attacks are similar but not exactly same. In these cases (type 1 and type 2 testing datasets), Random Forests and Bagging performed best. Hence, Random Forests and Bagging are the better choice as compared to AdaBoost.M1.
4. The network security datasets are quite large. The training of these ensembles on these datasets take a lot of time, whereas the storage of these ensemble takes a lot of space. As we observed that the performance of single decision tree is quite comparable with decision trees ensembles, one may use the single decision tree if the performance requirements are not very strict (the best performance).

4. CONCLUSION AND FUTURE WORK

Network security is an important research area. Data mining techniques provide important tools to predict the network attacks. KDD 99 data has been used by many researchers to test their proposed data mining techniques. KDD 99 data has redundant data points. A modified data set is presented to overcome this weakness. We carried out a detailed study of this datasets by using different classifier ensembles. We used naïve Bayes and decision trees as the base classifiers of the ensembles. The study suggests that decision trees ensembles performed better than naïve Bayes ensembles. Even the performance of single decision tree is quite competitive. We suggest that a single decision tree is a useful classifier for network security data. In future, we will use other ensemble methods; like multiboosting and Rotation Forests, and other classifiers like support vector machines [19] for our study.

5. ACKNOWLEDGEMENTS

This research is done at King Abdulaziz University and funded by King Abdulaziz University, Deanship of Scientific Research DSR, project number 27- 005 / 430.

REFERENCES

[1] KDD Cup'99 Data http://www.sigkdd.org.

[2] C.P.Pleeger. *Security in Computing 2nd Edition*. Prentice Hall, 1997.

[3] A.Lazarevic, A.Ozgur, L.Ertoz, J.Srivastava, and V.Kumar. *A comparative study of anomaly detection schemes in network intrusion detection*. Proceedings of the SAIM International Conference on Data Mining, 2003.

[4] M.Sabhnani and G.Serpen. *Application of machine learning algorithms to kdd intrusion detection dataset within misuse detection context*. Proceedings of the Intelligent Data Analysis, 2004.

[5] S.R.Gaddam, V.V.Phoha, and K.S.Balagani. *Means+id3 a novel method for supervised anomaly detection by cascading k-means clustering and id3 decision tree learning methods*. IEEE Transactions on Knowledge and Data Engineering, 2007.

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

[7] Shi-Jinn Horng, Ming-Yang Su, Yuan-Hsin Chen, Tzong-Wann Kao, Rong-Jian Chen, Jui-Lin Lai, Citra Dwi Perkasa, *"A novel intrusion detection system based on hierarchical clustering and support vector machines"*, Expert Systems with Applications, Vol: 38, No: 1, pp: 306-313, 2011.

[8] Arman Tajbakhsh, Mohammad Rahmati, Abdolreza Mirzaei, *"Intrusion detection using fuzzy association rules*", Applied Soft Computing, Vol: 9, No: 2, pp: 462-469, 2009.

[9] Hansen, L.K. and P. Salamon, 1990. *Neural network ensembles*. IEEE Trans. Patt. Anal. Mach. Intell., 12: 993-1001, 1990.

[10] Kuncheva, L. I., *Combining pattern classifiers: methods and algorithms*. Wiley-IEEE Press, New York, 2004.

[11] Breiman, L., *Bagging predictors.* Mach. Learn. 24(2):123–140, 1996.

[12] Freund, Y. and R.E. Schapire, 1997. *A decision theoretic generalization of on-line learning and an application to boosting.* J. Comput. Syst. Sci., 55: 119-139. DOI: 10.1006/jcss.1997.1504.

[13] G. I. Webb, Multiboosting: *A Technique for Combining Boosting and Wagging*, Machine Learning 40 (2000), no. 2, 159–196.

[14] [14] T. K. Ho, *The Random Subspace Method for Constructing Decision Forests*, IEEE Transactions on Pattern Analysis and Machine Intelligence 20, no. 8, 832–844, 1998.

[15] Breiman, L., *Random Forests*, Machine Learning 45, no. 1, 5–32. 2001.

[16] Tavallaee M.E, Bagheri W. Lu and Ghorbani A., *"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), pp. 53-58. 2009.

[17] Breiman, L., J. Friedman, R. Olshen and C. Stone, *Classification and Regression Trees*. 1st Edn., Chapman and Hall, London, ISBN- 10: 0412048418, 1984.

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

[19] Bishop, C.M.. *Pattern Recognition and Machine Learning.* 1st Edn., Springer-Verlag, New York, ISBN-10: 9780387310732, 2006.

[20] Witten, I. H., and Frank, E., *Data mining: practical machine learning tools with java implementations*. Morgan Kaufmann, San Francisco, 2000.