Int. J. Advanced Networking and Applications
Volume: 6 Issue: 5 Pages: 2455-2459 (2015) ISSN: 0975-0290

Detection of malicious attacks by Meta classification algorithms

G.Michael
Department of Computer Science and Engineering, Bharath University, Chennai, India
Email: micgeo270479@gmail.com
A.Kumaravel
Dean , School of Computing, Bharath University, Chennai, India
Email:drkumaravel@gmail.com
A.Chandrasekar
Professor, St.Joseph College of Engineering, Chennai, India
Email: drchandru@gmail.com

----------ABSTRACT----------

We address the problem of malicious node detection in a network based on the characteristics in the behavior of the network. This issue brings out a challenging set of research papers in the recent contributing a critical component to secure the network. This type of work evolves with many changes in the solution strategies. In this work, we propose carefully the learning models with cautious selection of attributes, selection of parameter thresholds and number of iterations. In this research, appropriate approach to evaluate the performance of a set of meta classifier algorithms (Ad Boost, Attribute selected classifier, Bagging, Classification via Regression, Filtered classifier, logit Boost, multiclass classifier). The ratio between training and testing data is made such way that compatibility of data patterns in both the sets are same. Hence we consider a set of supervised machine learning schemes with meta classifiers were applied on the selected dataset to predict the attack risk of the network environment . The trained models were then used for predicting the risk of the attacks in a web server environment or by any network administrator or any Security Experts. The Prediction Accuracy of the Classifiers was evaluated using 10-fold Cross Validation and the results have been compared to obtain the accuracy.

Keywords: Meta Classifier, Data mining, Decision Trees, Decision rules, Malicious detection, KDD dataset, Machine learning, Network security

Date of Submission: November 17, 2014 Date of Acceptance: January 13, 2015

I. INTRODUCTION

Even though there are three primary reasons like software bugs called vulnerabilities, lapses in administration and leaving systems to default configuration, the networks are usually attacked by injected bugs for creating various types of anomalies. Hence the research for detecting the anomalies becomes important to provide security to the organization's private resources from the Internet and protect critical systems. In this paper we consider such malicious attacks for predicting the intruders and their intentions. To defend against various cyber attacks and computer viruses, lots of computer security techniques have been intensively studied in the last decade, namely firewalls, In particular the authors of [1, 3] suggested for Malicious Network detection system (MNDS) using machine learning algorithms. Various researchers like communication engineers , Economists, forecasters, and statisticians have long worked with the idea that patterns in data can be sought automatically, identified, validated, and used for prediction. An IDS is a device that is placed inside a protected network to monitor what occurs within the network. The major objective of intrusion detection systems is :

To accurately detect anomalous network behavior or misuse of resources.
To Sort out the true attacks from false alarms.
To notify the Network administrators of the activity.

Many organizations now use intrusion detection systems to help them determine if their systems have been compromised (Carnegie Mellon University, 2001). Intrusion detection techniques using data mining as an important application area to analyze the huge volumes of audit data and realizing performance the optimization of detection rules. Different researchers propose different algorithms in different categories, from rules [8] to decision trees [6.7], from rule based models [8] to functions studying. The detection efficiencies therefore are becoming better and better than ever before.

However, to the best of our knowledge, a considerable comparison among these classification methods to pick out the best ones that suite the job of intrusion detection. A literature survey that was done by us also indicates a fact

Volume: 6 Issue: 5 Pages: 2455-2459 (2015) ISSN: 0975-0290

that, for intrusion detection, most researchers employed a single algorithm to detect multiple attack categories with depressing performance. Identifying attack category specific algorithms offers a promising research direction for improving intrusion detection performance.

In this paper, a comprehensive set of classifier algorithms will be evaluated on the KDD dataset [2,7,10]. The attacks will be detected on the four attack categories: Probe (information gathering), DoS (deny of service), U2R (user to root), R2L (remote to local. The model for classifier algorithm for the best performing algorithms for each attack category is proposed.

The remainder of this paper is organized as follows. A quick and up-to-date literature survey on attempts for designing Intrusion Detection Systems using the KDD dataset in Section 2. Section 3 will explain in detail about our simulation study (classifiers, evaluation setup and performance comparison).

Two models will be proposed in Section 4, to prove the effectiveness of our models; implementing issues will also be discussed here. Finally, Section 5 will conclude our study and discuss the future works. Section 6 concludes with the References.

II. RELATED WORK

This novelty in malicious behavior detection approach was employed to detect attack categories in the KDD dataset. The technique has achieved the detection rate of 96.71% of DoS, 99.17% of Probe, 93.57% of U2R and 31.17% of R2L respectively. However, due to the fact that no FP was reported by the research scientists a nearly impossible detection rate [5] of 93.57% of U2R category.

Adaptive intrusion dection was proposed by Xin Xu et al [4] based on machine learning algorithms. Multi-class Support Vector Machines (SVMs) is applied to classifier construction in IDSs and the performance of SVMs is evaluated on the KDD99 dataset. Promising results were given: 76.7%, 81.2%, 21.4% and 11.2% detection rate for DoS, Probe, U2R, and R2L respectively while FP is maintained at the relatively low level of average 0.6% for the four categories.

However, our proposed study can only use a very small set of data (10,000 randomly sampled records) comparing to the huge original dataset (5 million audit records). Yang Li and Li Guo [7] though realized the deficiencies of KDD dataset, developed a supervised Malicious Network detection method based on Transductive Confidence Machines for K-Nearest Neighbors (TCM-KNN) machine learning algorithm and active learning based training data selection method. The new method is evaluated on a subset of KDD dataset by random sampling 49,402 audit records for the training phase and 12,350 records for the testing phase. An average TP of 99.6% and FP of 0.1% was reported but no further information about the exact detection rate of each attack categories was presented by the authors.

Literature survey showed that, for all practical purposes, most of the researchers applied a single algorithm to address all four major attack categories. This motivated us to our assumption that different algorithms would perform with different predictions on different attack categories may yield a good performance and high prediction, comparatively.

III. EMPIRICAL STUDY

In order to verify the effectiveness of different classifiers algorithms for the field of intrusion detection, Nsl-KDD [7] dataset has been used to make relevant experiments *step-by-step.*

1) Initially, in order to build the experiment evaluation environment with major steps:
   a) Environment setup
   b) Data preprocessing
   c) Choosing the data mining Software.
2) Secondly, a comprehensive set of most popular classifier algorithms were selected to represent a wide variety of categories like Decision rules and Decision trees..
3) An overview of how specific values of the algorithms were identified as well as their Detection performance will be studied.

4) Finally, the performance Comparison between seven selected Classifiers will be achieved.

3.1 Evaluation Setup

All experiments were performed in a computer with the configurations Intel(R) Core(TM) 2 CPU 2.13GHz, 2 GB RAM, and the operation system platform is Microsoft Windows 7. An open source machine learning package – Weka (the latest Windows version: Weka 3.7.1). Weka is an environment in which a set of machine learning giving in the mining procedures for pre processing, classifying, clustering associating, and visualising mining. This empirical study, however, only deals with a subset of classifier algorithms.

All the machine learning technique [4] that will be used in this paper are implemented inWeka so that they will be easily and fairly compared to each other. The dataset to be used in our experiments in Nsl-KDD labeled dataset. The main reason to use this dataset is that the relevant data that can easily be shared with other researchers, allowing all kinds of techniques to be easily compared in the same baseline.

The Nsl-KDD data-set might have been criticized for its potential problems [7], but the fact is that it is the most widespread dataset that is used by many researchers and it is among the few comprehensive datasets that can be shared in malicious detection nowadays. Like the test

Volume: 6 Issue: 5 Pages: 2455-2459 (2015) ISSN: 0975-0290

dataset, the Nsl-KDD dataset contains one type of normal data and different types of attacks that are broadly categorized in two groups of normal and Anomaly. Table 1 shows the Distribution of Classes in the actual training data for classifiers evaluation and the percentage of malicious attacks is displayed using Pie chart in the Fig. 1.This type of binary class context can be extended to multiclass as in [5].

Table 1. Distribution of Classes in the actual training set

Category of attacks (Class)	Number of records	Percentage of Class Occurrences
Normal	67342	53%
anomaly	58629	47%
Total	125971	100%

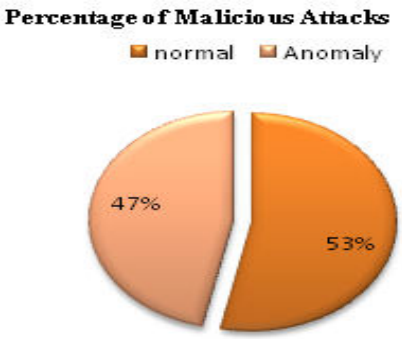

Figure 1. Percentage of Malicious Attacks

The packet information in the original TCP dump files were summarized into connections. This process resulting in 42 features for each connection, and one final feature for classifying as a category. Therefore, each instance of data consists of 42 features and each instance of them can be directly mapped and discussed in classifiers algorithms. Due to the huge number of audit data records in the original Nsl-KDD dataset, 125971 instances have been extracted as datasets for our experiments.

3.2 Classifier Algorithms

3.2.1 AdaBoost.M1

If one has a multiclass classification problem and wants to boost a multiclass base classifier AdaBoost.M1 is a well known and widely applicated boosting algorithm. However AdaBoost.M1 does not work, if the base classifier is too weak. We show, that with a modification of only one line of AdaBoost.M1 one can make it usable for weak base classifiers, too. AdaBoost.M1 is experimented in this study with the Parameters as seen in Weka tool terms Classifier = DecisionStump, numIterations=10, Seed=1, and WeightThreshold=100.

3.2.2 Attribute selected classifier

Dimensionality of training and test data is reduced by attribute selection before being passed on to a classifier. Here the base classifiers are used. Various search methods are used during the attribute selection phase before this classifier is invoked. Attribute selected classifier is experimented in this study with the weka tool Parameters: Classifier=J48, and search=BestFirst.

3.2.3 Bagging

Bagging means bootstrap aggregating. it creates from the original training set, several (sub-)datasets are created by randomly sampling instances with replacement. This is to better guess the population from a given sample. The output from this model are usually combined by majority voting (for nominal target class) or averaging (for numerical prediction). Bagging is experimented in this study with the Parameters: bagsizepercent=100,Classifier=REPTree,numExecutionSlots=1,numIterations=10, and Seed=1.

3.2.4 Classification Via Regression

This Classifier does classification using regression methods. This is binarized and one regression model is built for each class value. Classification Via Regression is experimented in this study with the Parameters: Classifier=M5P,and min Num Instances = 4.0.

3.2.5 Filtered classifier

This Classifier is used with many type of filters keeping the structure of the training data and testing data same. Filtered classifier is experimented in this study with the Parameters :Classifier=J48,a kind of decision tree algorithm [9] with the parameters ,filter = Discretize,confidenceFacto r= 0.25; Number of Folds =3,Seed=1.

3.2.6 Logitboost

This algorithm is an extension of Adaboost algorithm. It replaces the exponential loss of Adaboost algorithm to conditional Bernoulli likelihood loss. This Class is used for performing additive logistic regression. This classifier uses a regression scheme as the base learner, and can handle multiclass problems. Logitboost is experimented in this study with the Parameters:Classifier=Decision Stump,numFolds=0,numIterations=10.

3.2.7 Multiclass classifier

Multiclass or multinomial classifier is use for classifying instances into more than two classes. Error correction codes are accommodated with this classifier for obtaining more accuracy. Multiclass classifier is experimented in this study with the Parameters:Classifier=Logistic,randomwidthFactor=2.0,Seed=1.

3.3 Performance Comparison

Best performing instances of all the seven classifiers

Volume: 6 Issue: 5 Pages: 2455-2459 (2015) ISSN: 0975-0290

selected in Section 3.2 were evaluated on the KDD dataset. Simulation results are given in the Table 2&3. To compare the classifiers, TP (True positive) and FP (False Positive) and Prediction Accuracy for each algorithm are considered in Figure 2 These parameters will be the most important criteria for the classifier to be considered as the best algorithm for the given attack category.

Table 2 Precision, Recall, F-measure for meta classifiers

Algorithm	class	Precision	Recall	F-Measure
AdaBoostM1	anomaly	0.953	0.928	0.940
Attribute Selected Classifier	anomaly	0.997	0.991	0.994
Classification Via Regression	anomaly	0.998	0.997	0.998
Bagging	anomaly	0.999	0.998	0.998
Filtered Classifier	anomaly	0.996	0.997	0.997
LogitBoost	anomaly	0.975	0.962	0.969
MultiClass Classifier	anomaly	0.969	0.968	0.968

Table 3 TP Rate, FP Rate, ROC Area for meta classifiers

Algorithm	Class	TP Rate	FP Rate	ROC Area
AdaBoostM1	normal	0.960	0.072	0.988
Attribute Selected Classifier	normal	0.998	0.009	0.999
Classification Via Regression	normal	0.999	0.003	1.000
Bagging	normal	0.999	0.002	1.000
Filtered Classifier	normal	0.997	0.003	0.998
LogitBoost	normal	0.979	0.038	0.996
MultiClass Classifier	normal	0.973	0.032	0.989

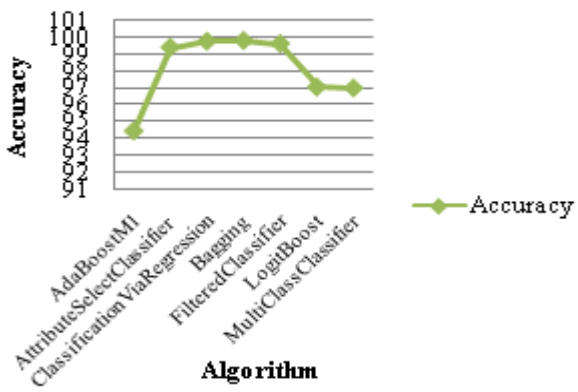

Figure 2. Accuracy vs Metaclassifiers

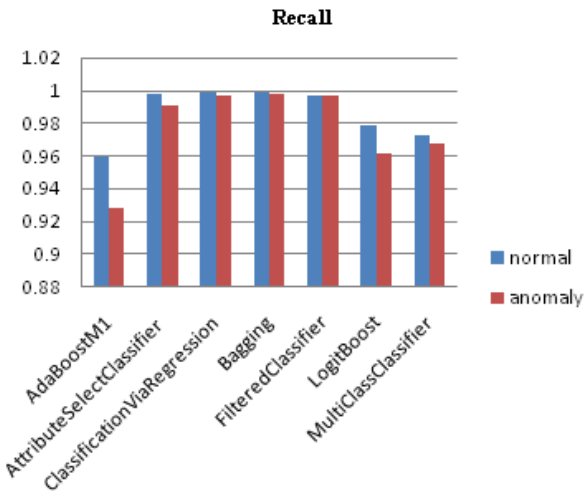

Figure 3. Recall vs Metaclassifiers

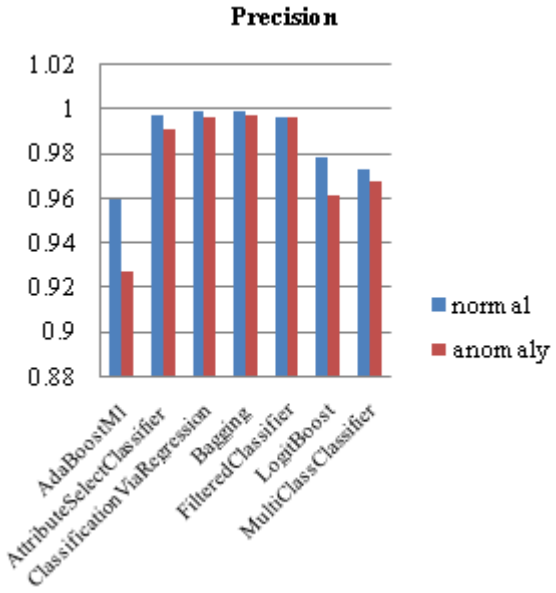

Figure 4. Precision vs Metaclassifiers

Precision for normal class is higher than the anomoly class as seen in the figure 4.

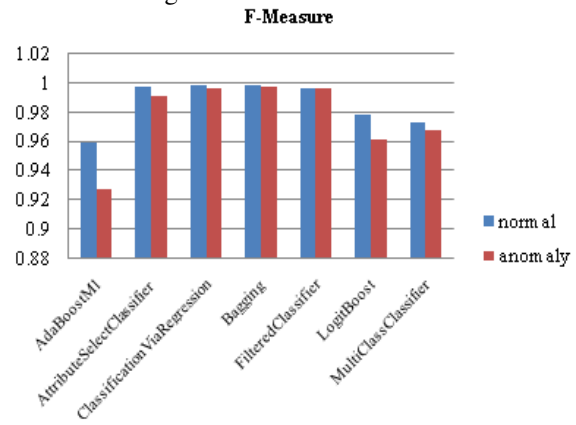

Figure 5. Accuracy vs Metaclassifiers

Volume: 6 Issue: 5 Pages: 2455-2459 (2015) ISSN: 0975-0290

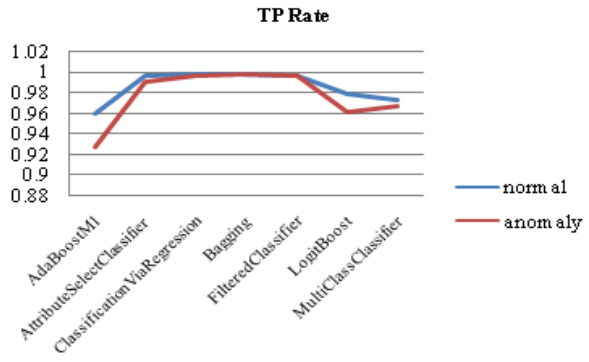

Figure 6. TP Rate vs Metaclassifiers

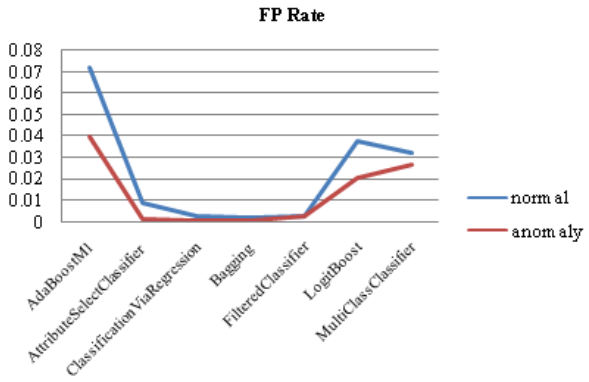

Figure 7. FP Rate vs Metaclassifiers

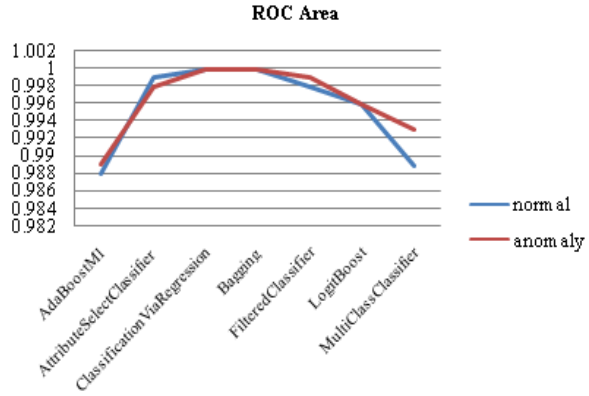

Figure 8. ROC Area vs Metaclassifiers

IV. CONCLUSION

In the research paper, supervised machine learning schemes with meta classifiers were applied on the selected dataset assessment data to predict the attack risk of the network environment and the performance of the learning methods were evaluated (figures 3-8) based on their predictive accuracy and ease of learning. The results indicate that the Bagging Meta Classifier outperforms in prediction than the other meta classifiers. Moreover the measures like TP ,FP Rates, F-measure and ROC Area were found to be higher for normal class professional or the Administrators to assess the risk of the attacks.

REFERENCES

[1]. Witten, I.H., Frank, E.: Data Mining: Practical Machine Learning Tools and Techniques, 2nd edn. Morgan Kaufmann, San Francisco (2005).

[2]. Tavallaee M.E, Bagheri W. Lu 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), pp. 53-58.

[3]. Xu, X.: Adaptive Intrusion Detection Based on Machine Learning: Feature Extraction, Classifier Construction and Sequential Pattern Prediction. International Journal of Web Services Practices 2(1-2), 49–58 (2006).

[4]. Li, Y., Guo, L.: An Active Learning Based TCM-KNN Algorithm for Supervised Malicious Network node detection. In: 26th Computers & Security, pp. 459–467 (October 2007)

[5]. Quinlan, J.: C4.5: Programs for Machine Learning. Morgan Kaufmann, San Mateo (1993).

[6]. "Nsl-KDD data set for network-based intrusion detection systems." Available on: http://nsl.cs.unb.ca/NSL-KDD .

[7]. Panda M. and Patra M.R (2008), "A Comparative study of Data Mining Algorithms for Network Intrusion Detection", Proceedings of the 1st Conference on Emerging Trends in Engineering and Technology, pp. 504-507, IEEE Computer
Society, USA.

[8]. Langley P. and Simon H. A (1995), "Applications of machine learning and rule induction", Communications of the ACM, Vol.38, No. 11, pp. 55–64.

[9]. Amor N.B, Benferhat S. and Elouedi Z (2004), "Naïve Bayes vs. Decision Trees in Intrusion Detection Systems", Proceedings of 2004, ACM Symposium on Applied Computing, pp. 420-424.

[10]. G.MeeraGandhi, Kumaravel Appavoo, S.K.Srivatsa," Effective Network Intrusion Detection using Classifiers Decision Trees and Decision rules", Int. J. Advanced Networking and Applications Volume: 02, Issue: 03, Pages: 686-692 (2010).