Volume 3, Issue 2, February 2013 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Comparative Study of Multiclass classification Techniques for Intrusion Detection

Viralkumar Prajapati* *Department of IT G. H. Patel College of Engg. & Tech., India*

Chirag Nathwani *Department of IT G. H. Patel College of Engg. & Tech., India*

Prof. Deven Agravat *Department of IT G. H. Patel College of Engg. & Tech., India*

Abstract— Nowadays, with the rapid development of the Internet, Internet security is becoming an important problem recently. Everyone wants to protect their data from both the internal and external attacks. In this initiative firewall, encryption and authentication serve as the first line of defense. And Intrusion Detection serves as the second line of defense. Intrusion detection has become an indispensable tool to keep information systems safe and reliable. In this work data mining techniques used for intrusion detection. Most of works in this area are done as binary classification, while intrusion detection is multiclass classification approach. Multiclass classification technique namely AdaBoost, LAD Tree, Naïve Bayes, Decision Tree (C4.5) and Random Forest are studied and compared for detecting the Intrusion. The tool weka is used for evaluating this on the KDD CUP '99 dataset.

Keywords— intrusion detection; data mining; multiclass classification; adaboost; LAD tree, naïve bayes; decision tree; random forest

I. INTRODUCTION

In today's modern world, Computers and Internet have become essential part of human life. Communication between two or more peoples starts from simple chatting to more privacy contents. So always there was a possible for intruders in Internet to tackle the communication between two parties. So to ensure the secure communication in internet, we need a security system to detect the attacks very effectively. Intrusion Detection System (IDS) is a model which provides alarm, if any intrusion (attacks) happens in a network. Intrusion Detection Systems is a combination of software and hardware that attempts to perform intrusion detection. Intrusion detection is a process of gathering intrusion related knowledge occurring in the process of monitoring the events and analyzing them for sign or intrusion.

There are mainly two types of Intrusion detection system. One is Network based and other one is Host based intrusion detection system. In Network Intrusion detection system (NIDS), IDS model uses mainly IP package information collected by the network hardware such as switches and routers to detect the possibility of attacks. For each network, an IDS model was placed in a way such that any network traffic that can comes or goes outside of network can pass through this IDS model. In Host intrusion detection system (HIDS), IDS Model uses audit data that are collected from the target host machine. Misuse and Anomaly detection are the basic approaches for detecting network intrusions. Misuse detection was based on storing all features of known attacks in a database and thereby checking database for detecting attacks. This Misuse intrusion detection system has high detection rates in the case of well-known attacks, but fails to detect new unknown attacks. Also the time taken for detecting intrusions was more due to checking network traffic with the audit data stored in our system. And also it needs continuous updating as like antivirus software. Anomaly detection was based on detecting behavioral change of users in the network. If the behavioral of network user changes, then it alarms that there was a possibility of attacks in the network.

Data Mining plays a vital role in modeling Intrusion Detection System. Data Mining is an approach used to extract a hidden knowledge from the vast amount of data. J.W. Han et.al, describes about various techniques in data mining [6]. In data mining there are many areas like classification, clustering for extract hidden knowledge.

To model Intrusion detection system, classification algorithms or clustering algorithms are used. Classification algorithms in data Mining was mainly used in Intrusion Detection System to classify attacks (intrusions) from normal things happening in networks. Classification algorithms are supervised learning approach, whereas Clustering algorithms are unsupervised learning approach, because it does not require class labels for the prediction purpose. Classification techniques are mainly categories in two. One is binary classification and other is multiclass classification. Binary classification technique classifies the members of a given set of objects into two groups on the basis of whether they have some property or not, while multiclass classification technique classifies instances into more than two classes. Some classification algorithms naturally permit the use of more than two classes; others are by nature binary algorithms.

The proposed paper organized as, Section 2 outline about related work. Section 3 explained data mining techniques. Section 4 outline the experiment setup. Result included in Section 5 with concluding conclusion in section 6.

II. RELATED WORK

In the concept of classification, there are mainly two approaches for classification. One is binary classification and other is multiclass classification. Various methods have been proposed in different researches for intrusion detection based system with classification techniques. M. Moorthy[10] explores the features of intrusion detection based on data mining. Author also present survey of some data mining techniques such as Feature selection, Machine learning, Inductive rule generation, Genetic algorithm, Fuzzy logic, Neural network, Immunological based techniques, Support vector machine, Statistical techniques and Hidden markov model [10]. Liang Wang and tong Wang reviews the history of intrusion detection technology, and derive six technical difficulties in the field of intrusion detection systems such as detection precision, load, real-time ability, intrusion tolerance, evidence extraction and learning ability are analyzed and discussed that intrusion detection system is facing the six new challenges: intrusion diversification, architecture standardization, high performance algorithms, security, technology integration and cloud security [8]. Li Hanguang and Ni Yu uses Apriori algorithm which is the classic of association rules in Web-based Intrusion Detection System and applies the rule base generated by the Apriori algorithm to identify a variety of attacks, improves the overall performance of the detection system [9]. R.China represent three data mining techniques namely C5.0 Decision Tree, Ripper Rule, Support Vector Machines, which are studied and evaluate them on KDD Cup '99 [15] standard dataset [12]. Safwan Mawlood Hussein proposed hybrid IDS by integrated Snort with Naive Bayes to enhance system security to detect attacks [14].

Most of work until done in intrusion detection based on binary classification. If attacks (intruders) classified only in to two, we have only information about it is attack or not. But, if we classified these attacks in to more than two classes, it gives us which type of attack it is. So, we can easily take action against those attacks (intruders).

III. CLASSIFICATION TECHNIQUES

Some of data mining techniques which based on multiclass classification are discussed below.

A. AdaBoost

AdaBoost is stands for Adaptive Boosting. It is a machine learning algorithm. AdaBoost formulated by Yoav Freund and Robert Schapire [2]. It is a meta-algorithm, and can be used in conjunction with many other learning algorithms to improve their performance.

"Boosting" is used for improving the performance of any learning algorithm. In theory, boosting can be used to significantly reduce the error of any "weak" learning algorithm that consistently generates classifiers which need only be a little bit better than random guessing. Boosting works by repeatedly running a given weak learning algorithm on various distributions over the training data, and then combining the classifiers produced by the weak learner into a single composite classifier.

AdaBoost is adaptive in the sense that subsequent classifiers built are weaked in favor of those instances misclassified by previous classifiers. AdaBoost is sensitive to noisy data and outliers. The classifiers it uses can be weak, but as long as their performance is not random, they will improve the final model.

AdaBoost generates and calls a new weak classifier in each of a series of rounds t = 1,…,T, For each call, a distribution of weights D_t is updated that indicates the importance of examples in the data set for the classification. On each round, the weights of each incorrectly classified example are increased, and the weights of each correctly classified example are decreased, so the new classifier focuses on the examples which have so far eluded correct classification.

Boosting can be seen as minimization of a convex loss function over a convex set of functions. Specifically, the loss being minimized is the exponential loss

$$\sum_i e^{-y_i f(x_i)} \tag{1}$$

and we are seeking a function

$$f(x) = \sum_t \propto_t h_{t(x)} \tag{2}$$

B. LADTree

Friedmann et al [5], in defining the multiclass context. Namely, that for an instance i and a J class problem, there are J responses y_ij^* each taking values in {-1, 1}; The predicted values, or indicator responses, are represented by the vector Fj(x) which is the sum of the responses of all the ensemble classifiers on instance x over the J classes. The class probability estimate is computed from a generalization of the two-class symmetric logistic transformation to be:

$$P_j(x) = \frac{e^{F_j(x)}}{\sum_{k=1}^{j} e^{F_k(x)}}, \sum_{k=1}^{j} F_k(x) = 0 \tag{3}$$

The LogitBoost algorithm can be fused with the induction of ADTrees in two ways, which will be explained in the following subsections. Infirst, more conservative approach called LT1PC we grow separate trees for each class in parallel. In the second approach called LT, only one tree is grown predicting all class probabilities simultaneously.

C. Naïve Bayes

A naive Bayes classifier is a simple probabilistic classifier. It is based on applying Bayes' theorem with strong (naive) independence assumptions. A more straightforward term for the fundamental probability model would be "independent feature model"

In simple terms, a Naïve Bayes classifier assumes that the presence (or absence) of a particular feature of a class is unrelated to the presence (or absence) of any other feature, given the class variable. For example, a fruit may be considered to be an apple if it is red, round, and about 4" in diameter. A naive Bayes classifier considers all these features to contribute independently to the probability that this fruit is an apple, whether or not they're in fact related to each other or to the existence of the other features.

For some types of probability models, Naïve Bayes classifiers can be trained very efficiently in a supervised learning setting. In many practical applications, parameter estimation for Naïve Bayes models uses the method of maximum probability; in other words, one can work with the Naïve Bayes model without believing in Bayesian probability or using any Bayesian methods.

An advantage of the Naïve Bayes classifier is that it only requires a small amount of training data to estimate the parameters (means and variances of the variables) necessary for classification. Because independent variables are assumed, only the variances of the variables for each class need to be determined and not the entire covariance matrix.

D. Decision Tree

C4.5 is an algorithm used to generate a decision tree developed by Ross Quinlan. [11] The decision trees generated by C4.5 can be used for classification, and for this reason, C4.5 is often referred to as a statistical classifier.

C4.5 builds decision trees from a set of training data, using the concept of information entropy. The training data is a set $S = s_1, s_2, \ldots\ldots$ of already classified samples. Each sample s_i consists of a p-dimensional vector $x_{1,i}, x_{2,i}, \ldots \ldots, x_{p,i}$ where the x_i represent attributes or features of the sample, as well as the class in which s_i falls.

At each node of the tree, C4.5 chooses the attribute of the data that most effectively splits its set of samples into subsets enriched in one class or the other. The splitting criterion is the normalized information gain (difference in entropy). The attribute with the highest normalized information gain is chosen to make the decision.

All the samples in the list belong to the same class. When this happens, it simply creates a leaf node for the decision tree saying to choose that class. None of the features provide any information gain. In this case, C4.5 creates a decision node higher up the tree using the expected value of the class.

Instance of previously-unseen class encountered. Again, C4.5 creates a decision node higher up the tree using the expected value.

E. Random Forest

Random forest is an ensemble classifier that consists of many decision trees and outputs the class that is the mode of the classes output by individual trees. The algorithm for inducing a random forest was developed by Leo Breiman and Adele Cutler, and "Random Forests" is their trademark [1]. The term came from random decision forests that were first proposed by Tin Kam Ho of Bell Labs in 1995. The method combines Breiman's "bagging" idea and the random selection of features, introduced independently by Ho[3][4] and Amit and Geman [13] in order to construct a collection of decision trees with controlled variation.

IV. EXPERIMENT SETUP

A. Dataset

KDD Cup '99 is a standard dataset for intrusion detection. The KDD Cup '99 [15] data set was provided by Stolfo and Lee for the Knowledge Discovery and Data Mining Tools competition (and associated conference) in 1999.The data set consists of 41 different attributes and 39 different attacks. There are three partitions of the KDD Cup '99 data available: a full training set which consist of 4,898,431 instances, a 10% version of this training set which consist of 494021 instances, and a test set which consist of 311,029 instances. The attacks are commonly grouped into 4 classes: Probing, Denial of Service (DoS), User to Root (U2R) and Remote to Local (R2L) [7]. Each class consists of following attacks.

TABLE I
ATTACK TYPES GROUPED TO THEIR RESPECTIVE CLASSES.

Class	Attacks
Probing	ipsweep, mscan, nmap, portsweep, saint, satan
DoS	apahe2, back, land, mailbomb, neptune, pod, prosesstable, smurf, teardrop, udpstorm

Class	Attacks
U2R	buffer_overflow, loadmodule, perl, ps, rootkit, sqlattack, xterm
R2L	ftp_write, guess_passwd, httptunnel, imap, multihop, named, phf, sendmail, snmpgetattack, snmpguess, spy, Warezclient, worm, xlock, xsnoop

In this paper, experiments perform on two different dataset versions one consist of 10,000 data instances and other consist of 25,067 data instances. Distribution of data instances are shown in following table.

TABLE II
PROPORTIONS OF ATTACK CLASSES IN THE KDD CUP '99 DATA SET.

Dataset	Class				
	Normal	***DoS***	***U2R***	***R2L***	***Probing***
Training (full)	972,780	3,883,370	52	1,126	41,102
Training (10%)	97,278	391,458	52	1,126	4,107
Test	60,593	4,166	229,853	70	16,347
Training (10,000)	1,959	7,882	52	24	83
Training (25,067)	3,938	15,848	52	1,124	4,105

B. Measures

The following terms are often used when discussing the performance of IDSs:

True positive (TP): classifying an intrusion as an intrusion.
False positive (FP): incorrectly classifying normal data as an intrusion.
True negative (TN): correctly classifying normal data as normal.
False negative (FN): incorrectly classifying an intrusion as normal.

The performance metrics calculated from these are:

$$True\ positive\ rate\ (TPR) = \frac{TP}{TP + FN} = \frac{\#correct\ intrusions}{\#intrusions} \quad (4)$$

$$False\ positive\ rate\ (FPR) = \frac{FP}{TN + FP} = \frac{\#normal\ as\ intrusions}{\#normal} \quad (5)$$

$$True\ negative\ rate\ (TNR) = \frac{TN}{TN + FP} = \frac{\#correct\ normal}{\#normal} \quad (6)$$

$$False\ negative\ rate(FNR) = \frac{FN}{TP + FN} = \frac{\#intrusions\ as\ normal}{\#intrusions} \quad (7)$$

Two additional performance metrics are also commonly used, referred to as accuracy and precision:

$$Accuracy = \frac{TP+TN}{TP + FP + TN + FN} = \frac{\#correct\ classifications}{\#all\ instances} \quad (8)$$

V. RESULTS

In this paper, experiments perform on datasets with techniques AadaBoost, LADTree, Naïve Bayes, Decision Tree (C4.5) and Random Forest techniques in to WEKA (Waikato Environment for Knowledge Analysis). Experiment first done on to the dataset with 10,000 data instances and then on to the dataset with 25,067 data instances. Both experiments are done using ten cross fold validation.

TABLE III
EXPERIMENT RESULTS ON DATASET WITH 10,000 DATA INSTANCES

Measures	Techniques				
	AdaBoost	***LADTree***	***Naïve Bayes***	***Decision Tree***	***Random Forest***

Measures	Techniques				
	AdaBoost	***LADTree***	***Naïve Bayes***	***Decision Tree***	***Random Forest***
TP Rate	0.974	0.991	0.908	0.997	**0.998**
FP Rate	0.025	0.011	0.009	0.003	**0.002**
Time	1.87	68.26	0.55	2.12	**2**
Accuracy	97.36	99.15	90.81	99.71	**99.85**

Above results illustrate that Random Forest technique is classified instance correctly than other techniques which discussed above. Among five techniques Random Forest techniques gives us better accuracy. If FP Rate is smaller, effectiveness of techniques is better. In above results, FP Rate of Random Forest technique is only 0.002. So, its effectiveness is better than other four techniques. Time taken to build a training set of Naïve Bayes techniques is better than other four techniques. Finally, results illustrate that Random Forest is better approach than other four techniques on dataset which contain 10,000 data instances.

TABLE IV
EXPERIMENT RESULTS ON DATASET WITH 25,067 DATA INSTANCES

Measures	Techniques				
	AdaBoost	***LADTree***	***Naïve Bayes***	***Decision Tree***	***Random Forest***
TP Rate	0.762	0.974	0.982	0.995	**0.999**
FP Rate	0.374	0.011	0.003	0.004	**0**
Training Time	1.3	281.79	0.19	0.97	**1.26**
Accuracy (%)	76.22	97.39	98.24	99.52	**99.86**

Above table shows that Random Forest techniques is classified instance correctly than other four techniques. From five techniques, Random Forest techniques give us better accuracy. In above results, FP Rate of Random Forest technique is only zero. So, its effectiveness is better than other four techniques. Naïve Bayes techniques was take less time for build a training set than other four techniques. Finally, we illustrate that Random Forest is better approach than other four techniques on dataset which contain 25,067 data instances.

VI. CONCLUSIONS

In this paper compare five different techniques of multiclass classification for intrusion detection in terms of accuracy, TP Rate, FP Rate and time taken to build training set. These five techniques include AdaBoost, LADTree, Naïve Bayes, Decision Tree and Random Forest. Results analysis illustrates Random Forest techniques is better than others on both datasets (10,000 and 25,067 data instances). In KDD Cup '99 dataset there are total 39 different attacks are there. So, In future work it challenge to classified dataset in to 39 different classes. By this work, we can know about which type of attack is there, so it is easy to action against intrusion that occurs.

REFERENCES

[1] Breiman, Leo. "Random Forests". Machine Learning 45 (1): 5–32. doi:10.1023/A:1010933404324. 2001.
[2] Freund, Yoav; Schapire and Robert E. A Decision-Theoretic Generalization of on-Line Learning and an Application to Boosting. CiteSeerX: 10.1.1.56.9855. 1995
[3] Ho, Tin Kam. "Random Decision Forest". Proceedings of the 3rd International Conference on Document Analysis and Recognition, Montreal, QC, 14–16 August 1995. pp. 278–282. 1995.
[4] Ho, Tin Kam. "The Random Subspace Method for Constructing Decision Forests". IEEE Transactions on Pattern Analysis and Machine Intelligence 20 (8): 832–844. doi:10.1109/34.709601. 1998.
[5] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. Additive logistic regression: a statistical view of boosting. The Annals of Statistic, 28(2):337{374, 2000.
[6] J.W. Han and M.Kamber, Data Mining: Concepts and Techniques, 2nd edition. Morgan Kaufmann. 2006.
[7] K. Kendall. A Database of Computer Attacks for the Evaluation of Intrusion Detection Systems. Master's thesis, Massachusetts Institute of Technology, 1999.
[8] Liang Wang and Tong Wang. A Trend Survey of Intrusion Detection Technology. IEEE-International Conference on Cyber Technology in Automation, Control and Intelligent Systems. 978-1-4673-1421-3. 2012.
[9] Li Hanguang and Ni Yu. Intrusion Detection Technology Research Based on Apriori Algorithm. Elsevier, International Conference on Applied Physics and Industrial Engineering, Physics Procedia 24 1615–1620. 2011.
[10] M. Moorthy and Dr. S. Sathiyabama. Study of Intrusion Detection using Data Mining. IEEE-International Conference On Advances In Engineering, Science And Management (ICAESM), ISBN: 978-81-909042-2-3. 2012.
[11] Quinlan, J. R. C4.5: Programs for Machine Learning. Morgan Kaufmann Publishers, 1993.
[12] R.China Appala Naidu and P.S.Avadhani. A Comparison of Data Mining Techniques for Intrusion Detection. IEEE-International Conference on Advanced Communication Control and Computing Technologies (ICACCCT), ISBN No. 978-1-4673-2048-1112. 2012.

[13] R. Lippmann, D. Fried, I. Graf, J. Haines, K. Kendall, D. McClung, D. Weber, S. Webster, D. Wyschogrod, R. Cunningham and M. Zissman. Evaluating Intrusion Detection Systems: The 1998 DARPA Off-line Intrusion Detection Evaluation. In Proceedings of the DARPA Information Survivability Conference and Exposition, volume 2, pages 12–26. IEEE Computer Society Press, 2000.

[14] Safwan Mawlood Hussein, Fakariah Hani Mohd Ali and Zolidah Kasiran. Evaluation Effectiveness of Hybrid IDS Using Snort with Naïve Bayes to Detect Attacks. IEEE, ISBN No. 978-1-4673-0734-5/12. 2012.

[15] "KDD cup 1999 Data." Available: http://kdd.ics.uci.edu /databases/kddCUP99/kddCUP99.html

