International Journal of Computer Science and Engineering (IJCSE)
ISSN 2278-9960
Vol. 2, Issue 3, July 2013, 131-136
© IASET

International Academy of Science, Engineering and Technology
Connecting Researchers; Nurturing Innovations

INVESTIGATION OF FEATURE SELECTION AND ENSEMBLE METHODS FOR PERFORMANCE IMPROVEMENT OF INTRUSION ATTACK CLASSIFICATION

N. S. CHANDOLIKAR[1] & V. D. NANDAVADEKAR[2]

[1]Reader, Vishwakarma Institute of Technology, Pune, Maharashtra, India

[2]Director - MCA, Sinhgad Institute of Management, Pune, Maharashtra, India

ABSTRACT

The security of a computer system is compromised when an intrusion takes place. The popularization of shared networks and Internet usage demands increases attention on information system security. Importance of Intrusion detection system (IDS) in computer network security well proven. Data mining approach can play very important role in developing intrusion detection system. Classification is identified as an important technique of data mining. This paper investigates the possibility of using ensemble algorithms and feature selection to improve the performance of network intrusion detection systems.

KEYWORDS: Intrusion Detection System, Classification Algorithm, Feature Selection, Ensemble Methods

INTRODUCTION

An intrusion can be defined as "any set of actions that attempt to compromise the integrity, confidentiality or availability of a resource". Intrusion Detection[1] is the unrelenting active attempts in discovering or detecting the presence of intrusive activities.

Intrusion Detection System

Intrusion detection in the internet is an active area of research. As network attacks have increased in number and severity over the past few years, intrusion detection system (IDS) is increasingly becoming a critical component to secure the network. It requires accurate and efficient models for analyzing a large amount of system and network audit data. Intrusion Detection System (IDS) can detect, prevent and more than that IDS react to the attack. Therefore, the main objective of IDS is to at first detect all intrusions at first effectively. This leads to the use of an intelligence technique known as data mining/machine learning. These techniques are used as an alternative to expensive and strenuous human input.

Data Mining

Data mining [1] [2] [3] is concerned with systematized mining of hidden predictive information from large sort of database and repositories. Data mining can be used for solving the problem of network intrusion based security attack. It has Ability to process large amount of data and reduce data and by extracting specific data, with this Easy data summarization and visualization that help the security analysis. Feature selection is used for imposing an arbitrary or predefined cutoff on the number of attributes that can be considered when building a model, and also the choice of attributes, meaning that either the analyst or the modeling tool actively selects or discards attributes based on their usefulness for analysis. Ensemble[4] Data Mining Methods, also known as Committee Methods or Model Combiners, are machine learning methods that leverage the power of multiple models to achieve better prediction accuracy than any of the individual models could on their own.

INTRUSION DETECTION DATASETS

KDD Cup ’99 Data Set

The data set used to perform the experiment is taken from KDD Cup ’99[5][6][7], which is widely accepted as a benchmark dataset and referred by many researchers. “10% of KDD Cup’99” from KDD Cup ’99 data set was chosen to evaluate rules and testing data sets to detect intrusion. The entire KDD Cup ’99 data set contains 41 features. Connections are labeled as normal or attacks.

- Training set consists 5 million connections.
- 10% training set - 494,021 connections
- Test set have - 311,029 connections
- Test data has attack types that are not present in the training data .Problem is more realistic
- Train set contains 22 attack types
- Test data contains additional 17 new attack types that belong to one of four main categories.

PROPOSED SYSTEM DESCRIPTION

For the experiment decision tree J48 is chosen. J48 algorithm gives best performance [8] for intrusion attack classification. In this experiment it is investigated that how feature selection and ensemble method affects on performance of intrusion attack classification.

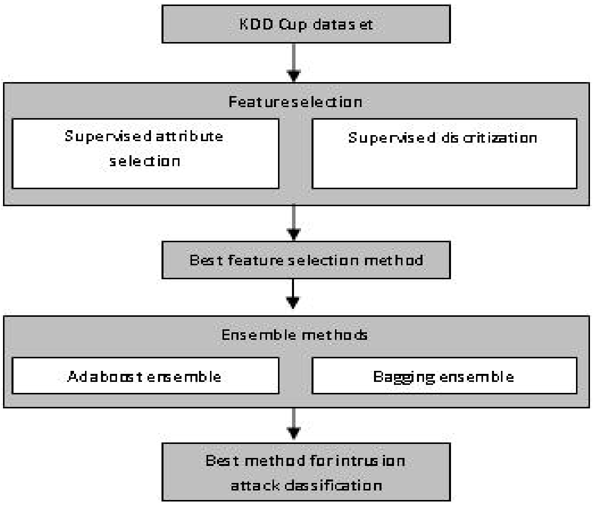
KDD Cup dataset
Featureselection
Supervised attribute selection
Supervised discritization
Best feature selection method
Ensemble methods
Adaboost ensemble
Bagging ensemble
Best method for intrusion attack classification

Figure 1: Proposed Method

The algorithms used in this investigation are briefly described in the following paragraphs

Classification

The goal of classification [9] learning is to develop a model that separates the data into the different classes, with the aim of classifying new examples in the future. Classification [1] [2] data mining technique Classification maps a data item into one of several pre-defined categories. These algorithms normally output "classifiers", for example, in the form of decision trees or rules. An ideal application in intrusion detection will be to gather sufficient "normal" and "abnormal" audit data for a user or a program, then apply a classification algorithm to learn a classifier that will determine (future) audit data as belonging to the normal class or the abnormal class. There are many types of classifiers are available like tree, bayes, function, rule. Basic aim of classifier is predict the appropriate class.

Decision Tree

Decision tree [10] [11] [12] [13] is an important method for data mining, which is mainly used for model classification and prediction. This predictive machine-learning model that decides the target value (dependent variable) of a new sample based on various attribute values of the available data. The internal nodes of a decision tree denote the different attributes; the branches between the nodes tell us the possible values that these attributes can have in the observed samples, while the terminal nodes tell us the final value (classification) of the dependent variable.

- **J48 Algorithm**

The J48 [14] is a Decision tree classifier algorithm. In this algorithm for classification of new item, it first needs to create a decision tree based on the attribute values of the available training data. It discriminate the various instances and identify the attribute for the same. This feature that is able to tell us most about the data instances so that we can classify them the best is said to have the highest information gain. Now, among the possible values of this feature, if there is any value for which there is no ambiguity, that is, for which the data instances falling within its category have the same value for the target variable, then we terminate that branch and assign to it the target value that we have obtained.

Feature Selection

Feature selection [7] is one of the common terms used in data mining. It is used to reduce inputs to a manageable size for processing and analysis. Many tools and techniques are available for the same. Feature selection for intrusion detection is an important factor for the success of intrusion detection system. Supervised discrete filter is used for attribute selection.

Ensemble Methods

Ensemble Methods began about ten years ago as a separate area within machine learning and were motivated by the idea of wanting to leverage the power of multiple models and not just trust one model built on a small training set. An ensemble classifier is a method which uses or combines multiple classifiers to improve robustness as well as to achieve an improved classification performance from any of the constituent classifiers. Furthermore, this technique is more resilient to noise compared to the use of a single classifier. Ensemble learning methods instead generate multiple models. Given a new example, the ensemble passes it to each of its multiple *base* models.

Bagging

*B*ootstrap *Agg*regat*ing* (Bagging) generates multiple bootstrap training sets from the original training set (using sampling with replacement) and uses each of them to generate a classifier for inclusion in the ensemble. This method is usually applied to decision tree algorithms, but it also can be used with other classification algorithms such as naïve bayes, nearest neighbor, rule induction, etc. The bagging technique is very useful for large and high-dimensional data, such as intrusion data sets, where finding a good model or classifier that can work in one step is impossible because of the complexity and scale of the problem.

Boosting

Boosting is a forward stage wise additive model .Boosting, is an ensemble method for boosting the performance of a set of weak classifiers into a strong classifier. This technique can be viewed as a model averaging method and it was originally designed for classification, but it can also be applied to regression. Boosting provides sequential learning of the predictors. The first one learns from the whole data set, while the following learns from training sets based on the performance of the previous one. The misclassified examples are marked and their weights increased so they will have a higher probability of appearing in the training set of the next predictor. It results in different machines being specialized in predicting different areas of the dataset

- **AdaBoost**

In this paper, we select an AdaBoost algorithm, which is one of the most widely used boosting techniques for constructing a strong classifier as a linear combination of weak classifiers. AdaBoost generates a sequence of base models with different weight distributions over the training set.

EXPERIMENTAL SETUP

To assess the effectiveness of the algorithms for proposed intrusion detection, the series of experiments were performed in Weka. The java heap size was set to 1024 MB for weka-3-6. KDD 99 dataset is investigated to identify the relevance of each feature in intrusion detection. To test and evaluate the algorithms we use 10-fold cross validation. This provides a good indication of how well the classifier will perform on unseen data. We used the J48 algorithm available on the Weka collection of machine learning algorithms. J48 is the Weka implementation of the decision tree learner C4.5. This algorithm is chosen for several reasons: this is well-known classification algorithms. It can originate easily understandable rules and are designed to classify into predefined discrete categories (classes).this algorithm gives best performance for intrusion attack classification.

Weka

Weka [14][15] is a collection of machine learning algorithms for data mining tasks. Weka contains tools for data pre-processing, classification, regression, clustering, association rules, and visualization. It is also well-suited for developing new machine learning schemes. WEKA consists of Explorer, Experimenter, Knowledge flow, Simple Command Line Interface, Java interface.

Performance Measurement Terms

- **Correctly Classified Instance**

The correctly and incorrectly classified instances show the percentage of test instances that were correctly and incorrectly classified. The percentage of correctly classified instances is often called accuracy or sample accuracy.

- **Kappa Statistics**

Kappa is a chance-corrected measure of agreement between the classifications and the true classes

- **Mean Absolute Error, Root Mean Squared Error, Relative_Absolute_Error**

The error rates are used for numeric prediction rather than classification. In numeric prediction, predictions aren't just right or wrong, the error has a magnitude, and these measures reflect that.

RESULTS AND DISCUSSIONS

Our ultimate goal is to investigate how a feature selection and ensemble method improves performance of decision tree classifiers. Algorithms are evaluated on the bases of performance measurement terms. For accuracy measurement, Table 1 shows the Performance of j48 with features selection two different mechanisms, supervised attribute selection and discritization. Performance is evaluated based on Correctly classified instance, Incorrectly classified instance, Kappa statistics, Mean absolute error, Root mean squared error.

Table 1: Performance of J48 Algorithm with and without Feature Selection

Sr. No.	Parameter	J48 without Feature Selection	J48 with Feature Selection	J48 with Discritization
1	Correctly classified instance	99.7563	99.742	99.7079
2	Incorrectly classified instance	0.2437	0.258	0.2921
3	Kappa statistics	0.996	0.9957	0.9952
4	Mean absolute error	0.0003	0.0003	0.0004
5	Root mean squared error	0.0141	0.0145	0.0149
6	Relative_absolute_error	0.5621	0.656	0.7006

We can observe from above table that performance of j48 algorithm is approximately same in terms of accuracy. For accuracy measurement, Table 2 shows the Performance of j48 with features selection two different mechanisms, supervised attribute selection and discritization. Performance is evaluated based on time taken to build model.

Table 2: Performance of J48 Algorithm with and without Feature Selection

Sr. No.	Parameter	J48 without Feature Selection	J48 with Feature Selection	J48 with Discritization
1	Time taken to build model:	147.53 seconds	50.74 seconds	94.66 seconds

Based on observation we can say that when J48 decision tree classifier is used with feature selection reduces the time taken to build model. Whereas without feature selection time taken is very high. Table 3 shows the Performance of j48 with and without ensemble methods in terms of Correctly classified instance, Incorrectly classified instance, Kappa statistics, Mean absolute error, Root mean squared error.

Table 3: Performance of J48 Algorithm with and without Ensemble Methods

Sr. No.	Parameter	J48	J48 with Bagging	J48 with Boosting
1	Correctly classified instance	99.742	99.7809	99.8547
2	Incorrectly classified instance	0.258	0.2191	0.1453
3	Kappa statistics	0.9957	0.9964	0.9976
4	Mean absolute error	0.0003	0.0003	0.0001
5	Root mean squared error	0.0145	0.0124	0.0106
6	Relative absolute error	0.656	0.6426	0.2621

We can observe from above table that performance of j48 algorithm is approximately same in terms of accuracy. Table 4 shows the Performance of j48 with and without ensemble methods based on time taken to build model.

Table 4: Performance of J48 Algorithm with and without Ensemble Methods

Sr. No	Parameter	J48	J48 with Bagging	J48 with Boosting
1	Time taken to build model:	50.74 seconds	489.81 seconds	451.52 seconds

Based on observation we can say that when J48 decision tree classifier is used with feature selection reduces the time taken to build model. Whereas ensemble methods like bagging and boosting consume time. Our ultimate goal is to show the impact of feature selection and ensemble methods on the performance of IDS. We find that the feature reduction techniques are able to greatly reduce the feature space. Results show that J48 shows better performance accuracy when applied with feature selection. The J48 algorithm is significantly faster in terms of classification speed and appears to be the best suited for real-time classification tasks.

CONCLUSIONS

Data mining can improve intrusion based security attacks detection system by adding a new level of surveillance to detection of network data indifferences. For this selection of classification algorithm, feature selection and ensemble plays important role for performance improvement. Experiment performed on KDD cup dataset demonstrate that J48 algorithm when used with supervised attribute selection gives best performance for building model for intrusion detection.

REFERENCES

1. Jiawei Han And Micheline Kamber “Data mining concepts and techniques” Morgan Kaufmann publishers .an imprint of Elsevier .ISBN 978-1-55860-901-3. Indian reprint ISBN 978-81-312-0535-8 .
2. Witten IH, Frank E. Data Mining: Practical Machine Learning Tools and Techniques. Second edition, 2005. Morgan Kaufmann.
3. T. Lappas and K. P. , “Data Mining Techniques for (Network) Intrusion Detection System," January 2007.
4. Nikunj C. Oza ,Ensemble Data Mining Methods.
5. H. Güneş Kayacık, A. Nur Zincir-Heywood, Malcolm I. Heywood,. “ Selecting Features for Intrusion Detection: A Feature Relevance Analysis on KDD 99Intrusion Detection Datasets". Dalhousie University, Faculty of Computer Science, http://www.cs. dal. ca /projectx/
6. The KDD Archive. KDD99 cup dataset, 1999. http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html
7. Adetunmbi A.Olusola., Adeola S.Oladele. and Daramola O.Abosede . “Analysis of KDD ’99 Intrusion Detection Dataset for Selection of Relevance Features. Proceedings of the World Congress on Engineering and Computer Science 2010 Vol I WCECS 2010, October 20-22, 2010, San Francisco, USA.
8. N. S. Chandolikar, Dr. V. D. Nandavadekar, “Efficient Algorithm for Intrusion Attack Classification by Analyzing KDD Cup 99”, 978-1-4673-1989-8/12 ©2012 IEEE
9. Kayacik, G. H., Zincir-Heywood, A. N., “Analysis of Three Intrusion Detection System Benchmark Datasets Using Machine Learning Algorithms", Proceedings of the IEEE ISI 2005 Atlanta, USA, May 2005.
10. Ozgur Depren, Murat Topallar, Emin Anarim, M. Kemal Ciliz. “An intelligent intrusion detection system (IDS) for anomaly and misuse detection in computer networks”. Expert Systems with Applications 29 (2005) 713–722Expert Systems with Applications 29 (2005)713722.www.elsevier.com/locate/eswa.
11. Jeff Markey “ Using Decision Tree Analysis for Intrusion Detection: A How-To Guide”. Global Information Assurance Certification Paper , Copyright SANS Institute 2011.
12. E.Kesavulu Reddy, Member IAENG, V.Naveen Reddy, P.Govinda Rajulu,” A Study of Intrusion Detection in Data Mining “,Proceedings of the World Congress on Engineering 2011 Vol III WCE 2011, July 6 - 8, 2011, London, U.K. ISBN: 978-988-19251-5-2 ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)
13. Dewan Md. Farid, Nouria Harbi, Emna Bahri, Mohammad Zahidur Rahman, Chowdhury Mofizur Rahman ,“Attacks Classification in Adaptive Intrusion Detection using Decision Tree”, World Academy of Science, Engineering and Technology 63 2010
14. Weka documentation, http://www.cs.waikato.ac.nz/ml/weka/documentation.html
15. Muamer N. Mohammada,*, Norrozila Sulaimana, Osama Abdulkarim Muhsinb “A Novel Intrusion Detection System by using Intelligent Data Mining in Weka Environment”, Procedia Computer Science www.elsevier.com/locate/procedia WCIT 2010