VOL. 11, NO. 2, JANUARY 2016 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences


www.arpnjournals.com

MACHINE LEARNING IN NETWORK INTRUSION DETECTION SYSTEM

K. Umamaheswari [1] and S. Janakiraman[2]
[1]Department of computer Science, Bharathiar University, Coimbatore, Tamilnadu, India
[2]Department of Banking Technology, School of Management, Pondicherry University, Pondicherry, India
E-Mail: umamca82@gmail.com

ABSTRACT

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. 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. Due to large volumes of security audit data as well as complex and dynamic properties of intrusion behaviors, optimizing performance of IDS becomes an important open problem that is receiving more and more attention from the research community. In this paper, we evaluate performance of a comprehensive set of classifier algorithms using KDD99 dataset.

Keywords: Data mining, machine learning, classifier, network security, intrusion detection, algorithm selection, KDD database.

INTRODUCTION

Intrusion detection techniques using data mining have attracted more and more interests in recent years. A literature survey that was done by us also indicates a fact that, for intrusion detection, most researchers employed a single algorithm to detect multiple attack categories. In this paper, a comprehensive set of classifier algorithms will be evaluated on the KDD dataset [2]. We will try to detect attacks on the four attack categories: Probe(information gathering), DoS (denial of service), U2R (user to root), R2L (remote to local).

KDD'99 features can be classified into three groups:

1) Basic features: this category encapsulates all the attributes that can be extracted from a TCP/IP connection. Most of these features leading to an implicit delay in detection.

2) Traffic features: this category includes features that are computed with respect to a window interval and is divided into two groups:

a) "same host" features: examine only the connections in the past 2 seconds that have the same destination host as the current connection, and calculate statistics related to protocol behavior, service, etc.

b) "same service" features: examine only the connections in the past 2 seconds that have the same service as the current connection.

3) Content features: unlike most of the DoS and Probing attacks, the R2L and U2R attacks don't have any intrusion frequent sequential patterns. This is because the DoS and Probing attacks involve many connections to some host(s) in a very short period of time; however the R2L and U2R attacks are embedded in the data portions of the packets, and normally involves only a single connection. To detect these kinds of attacks, we need some features to be able to look for suspicious behavior in the data portion, e.g., number of failed login attempts. These features are called content features

The remainder of this paper is organized as follows. We make 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 provide the study of machine learning algorithms in NIDS. Section 4 will detail about our Data set description. Section 5 is detailed about the open source data mining software WEKA. Section 6 about the study and prove the effectiveness of our models. Section 7 about the performance measures for the NIDS. Finally, Section 8 will conclude our study and discuss the future works.

BACKGROUND STUDY

Literature survey shows that, for all practical purposes, most researchers applied a single algorithm to address all four major attack categories with dismal performance in many cases. This leads to our belief that different algorithms will perform differently on different attack categories, and this also is the motivation of our study. The research is usually to find out the normal network behavior style signature and then establish a normal network behavior model. If the flow is different to the normal behavior style collected from the system in the monitoring network, it will be judged as the improper/attack behavior.

MACHINE LEARNING IN NIDS

Machine learning applications involve tasks that can be set up as supervised. In this paper we have concentrated the classification techniques and feature selection such as supervised learning based on KDD cup 99 dataset. Classification is a model finding process that is used for portioning the data into different classes according to some constraints. Different classifier model must be built and evaluated based on the results. The goal of data mining is to produce a model expressed as an executable code which can be used to perform data mining tasks such as classification, prediction or other similar

tasks. Classification plays a main role in intrusion to detect accuracy which increases detection rate and reduce false alarm rate.

DATA SET DESCRIPTION

Since 1999, KDD'99 [3] has been the most wildly used data set for the evaluation of anomaly detection methods. This data set is prepared by Stolfo *et al*. [5] and is built based on the data captured in DARPA'98 IDS evaluation program [6]. DARPA'98 is about 4 gigabytes of compressed raw (binary) tcpdump data of 7 weeks of network traffic, which can be processed into about 5 million connection records, each with about 100 bytes. The two weeks of test data have around 2 million connection records. KDD training dataset consists of approximately 4,900,000 single connection vectors each of which contains 41 features and is labeled as either normal or an attack, with exactly one specific attack type.
The simulated attacks fall in one of the following four categories:
1) Denial of Service Attack (DoS): is an attack in which the attacker makes some computing or memory resource too busy or too full to handle legitimate requests, or denies legitimate users access to a machine.
2) User to Root Attack (U2R): is a class of exploit in which the attacker starts out with access to a normal user account on the system (perhaps gained by sniffing passwords, a dictionary attack, or social engineering) and is able to exploit some vulnerability to gain root access to the system.
3) Remote to Local Attack (R2L): occurs when an attacker who has the ability to send packets to a machine over a network but who does not have an account on that machine exploits some vulnerability to gain local access as a user of that machine.
4) Probing Attack: is an attempt to gather information about a network of computers for the apparent purpose of circumventing its security controls.

WEKA INTERFACE

WEKA (Waikato Environment for knowledge Analysis) is a widely used toolkit for machine learning and data mining that was originally developed at the University of Waikato in New Zealand. It is written in JAVA. The file format for the WEKA is ARFF (Attribute Relation file Format) . It becomes very popular with the academic and industrial researchers. WEKA contains various tools for regression, Classification, Clustering, Association rules, Visualization and data preprocessing. The Weka tool incorporates the four applications within it.(a) Explorer (b) Experiment (c) Knowledge Flow and (d) Simple CLI. For the classification of the dataset, weka explorer is used to generate the result or statistics.

EXPERIMENTAL STUDY

In the testing phase, dataset is evaluated to classify the input data as a normal or attack. The obtained result is then used to compute overall accuracy of the proposed system, used with four different classifiers which are normally used to estimate the rare class prediction.

Table-1. Measures for the different machine learning algorithms.

Classifier	J48	JRIPPER	Random Forest	Random tree
Correctly Classified	98.59	95.15	98.15	98.70
Incorrectly classified	1.49	1.8	1.84	1.2
Precision	0.986	0.95	0.981	0.987
Recall	0.986	0.95	0.981	0.987
F-measure	0.986	0.95	0.981	0.987

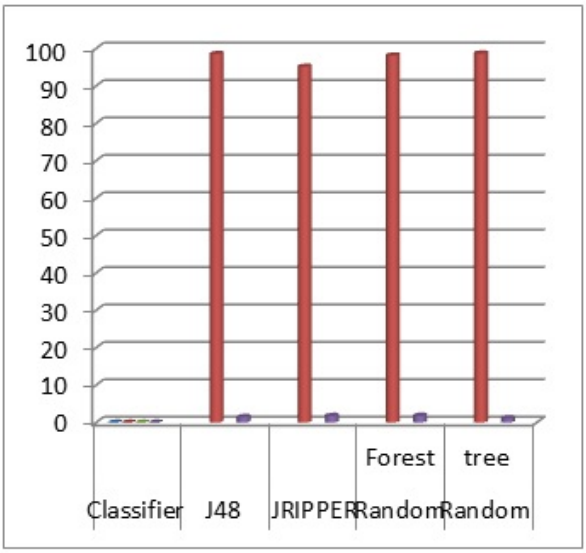

Figure-1. Chart for the correctly and incorrectly classified instances.

J48 Algorithm Results in WEKA

Time taken to build model: 5.03 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances	22228	98.5983 %
Incorrectly Classified Instances	316	1.4017 %
Kappa statistic	0.9714	
Total Cost	316	
Average Cost	0.014	
Mean absolute error	0.017	
Root mean squared error	0.1056	
Relative absolute error	3.4631 %	
Root relative squared error	21.316 %	
Coverage of cases (0.95 level)	99.4899 %	

Mean rel. region size (0.95 level) 51.71 %
Total Number of Instances 22544

=== Detailed Accuracy By Class ===

TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
0.986	0.014	0.982	0.986	0.984	0.971	0.996	0.994	normal
0.986	0.014	0.989	0.986	0.988	0.971	0.996	0.995	anomaly
Weighted Avg. 0.986	0.014	0.986	0.986	0.986	0.971	0.996	0.995	

=== Confusion Matrix ===

a	b	<-- classified as
9574	137	a = normal
179	12654	b = anomaly

PERFORMANCE MEASURES

In machine learning and data mining algorithms, many different measures are used to evaluate the classification models. We use three performance measures: attack detection rate, false alarm rate and accuracy. The detection rate and false positive rate are calculated based on Detection rate and False Alarm Rate.

Attack Detection Rate (ADR) - is the ratio between total numbers of attacks detected by the system to the total number of attacks present in the data set. ADR will show if our proposed model is capable of detecting attacks (thus raising alarms).

$$ADR = \frac{\text{Total No. of correctly classified instances.}}{\text{Total No. of Instances}} * 100$$

• **False Alarm Rate (FAR)** - is the ratio between total number of misclassified instances to the total number of normal instances. The FAR value will show if our model generates many false alarms. For IDS, this value should be as low as possible, otherwise too many false alarms may confuse the administrator.

$$FAR = \frac{\text{Total No. of incorrectly classified instances.}}{\text{Total No. of Instances}} * 100$$

• **Accuracy** - is the ratio between total number of correctly classified instances to the total number of samples from the data set. The accuracy will show if our model is capable of raising proper alarms, when it detects attacks and not generating false alarms when the network traffic is normal. The accuracy of the predictive model is calculated based on precision, recall values of classification matrix.

Precision is the fraction of retrieved instances that are relevant. It is calculated as total number of true positive divided by total number of true positive + total number of false positives.

$$Precision = \frac{\text{True Positives}}{\text{True Positives + False Positives}}$$

Recall is fraction of relevant instances that are retrieved. It is calculated as total number of true positive divided by total number of true positive + total number of false negatives.

$$Recall = \frac{\text{True Positives}}{\text{True Positives + False Negatives}}$$

F-measure is a measure of a test accuracy and is determined by the formula

$$F\text{-measure} = \frac{(2*TP\ Rate * Precision)}{(TP\ rate + Precision)}$$

These measures are calculated by using the confusion matrix. Where,

- TP (true positives) the number of attack records that are correctly classified, thus properly raising alarms.
- FP (false positive) the number of normal records that are incorrectly classified, thus generating false alarms.
- TN (true negatives) the number of normal records that are correctly classified as normal.
- FN (false negative) the number of attack records that are incorrectly classified as normal.

CONCLUSIONS

For the contribution of this paper, firstly we made an on recent studies about network intrusion detection that was evaluated with KDD99 dataset. We then use WEKA to bring out an extensive performance comparison among the most popular classifier algorithms such as Random forest, Random tree, j48 etc. For the future directions, we would like to evaluate our work on another dataset for the network intrusion detection system. Besides, we would also like to make real implementations of our algorithm selection models to practically experiment its effectiveness in WEKA. In this paper, we conclude random tree algorithms produces better accuracy compared with other algorithms. Network Intrusion Detection System is a latest kind of defense technology which is one of the vibrant areas in network security. In recent years many techniques are available for intrusion detection.

REFERENCES

[1] WEKA – Data Mining Machine Learning Software, http://www.cs.waikato.ac.nz/ml/weka/

[2] KDD Cup 1999 Data, http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html

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

[4] Agarwal, R., Joshi, M.V.: PNrule: A New Framework for Learning Classifier Models in Data Mining. Tech. Report, Dept. of Computer Science, University of Minnesota (2000)

[5] Denning D., "An intrusion detection model," IEEE Trans. Software Eng., vol. 13, no. 2, pp. 222–232, Feb. 1987.

[6] Ektefa M., Memar S., "Intrusion Detection Using Data Mining Techniques," IEEE Trans., 2010.

[7] Reddy E., Reddy V., Rajulu P., "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