Network Intrusion Detection System Using Data Mining

Lídio Mauro Lima de Campos, Roberto Célio Limão de Oliveira, and Mauro Roisenberg

Universidade Federal do Pará - UFPA
Av. dos Universitários s/n - Jaderlandia - Castanhal- PA - Brasil Cep: 68746-360
{lidio,limao,mauro}@ufpa.br
http://www.campuscastanhal.ufpa.br

Abstract. The aim of this study is to simulate a network traffic analyzer that is part of an Intrusion Detection System - IDS, the main focus of research is data mining and for this type of application the steps that precede the data mining : data preparation (possibly involving cleaning data, data transformations, selecting subsets of records, data normalization) are considered fundamental for a good performance of the classifiers during the data mining stage. In this context, this paper discusses and presents as a contribution not only the classifiers that were used in the problem of intrusion detection, but also the initial stage of data preparation. Therefore, we tested the performance of three classifiers on the KDDCUP'99 benchmark intrusion detection dataset and selected the best classifiers. We initially tested a Decision Tree and a Neural Network using this dataset, suggesting improvements by reducing the number of attributes from 42 to 27 considering only two classes of detection, normal and intrusion. Finally, we tested the Decision Tree and Bayesian Network classifiers considering five classes of attack: Normal, DOS, U2R, R2L and Probing. The experimental results proved that the algorithms used achieved high detection rates (DR) and significant reduction of false positives (FP) for different types of network intrusions using limited computational resources.

Keywords: Datamining, Network Intrusion Detection System, Decision Tree, Neural Network, Bayesian Network.

1 Introduction

With the enormous growth of computer networks usage and the huge increase in the number of applications running on top of it, network security is becoming increasingly more important. As shown in [2], all computer systems suffer from security vulnerability whose solution is not only technically difficult but also very expensive to be solved by manufacturers. Therefore, the role of Intrusion Detection Systems (IDSs), as special purpose devices to detect anomalies and attacks in the network, has become more and more important. KDDCUP'99 dataset is widely used as one of the few publicly available data sets for network-based

C. Jayne, S. Yue, and L. Iliadis (Eds.): EANN 2012, CCIS 311, pp. 104–113, 2012.

anomaly detection systems. It has been used as the main intrusion detection dataset for both training and testing [1] different Intrusion Detection schemes. However your research shows that there are some inherent problems in the KDDCUP'99 dataset[1] that must be corrected before performing any experiment.

Many researchers are devoted to study methodologies to project (IDSs), [3] employed 21 learned machines (7 learners, namely J48 decision tree learning [4], Naive Bayes [5], NBTree [6], Random Forest [7], Random Tree [8], Multilayer Perceptron [9], and Support Vector Machine (SVM) [10] from the Weka [11] collection to learn the overall behavior of the KDDCUP'99 data set), each trained 3 times with different train sets to label the records of the entire KDD train and test sets, which provided 21 labels for each record. Surprisingly, about 98% of the records in the train set and 86% of the records in the test set were correctly classified with all the 21 learners. Moreover, each dataset record was annotated with a #successfulPrediction value, which was initialized to zero. Once the KDD set had provided the correct label for each record, they compared each record predicted label given by a specific learner with actual label, where #successfulPrediction was incremented by one by one if a match was found. Through this process, the number of learners capable of correctly labeling that given record was calculated. The highest value for #successfulPrediction was 21, which conveys the fact that all learners were able predict that record label. Once conducted a statistical analysis on this data set and proposed a new data set,NSL-KDD,which consists of selected records of the complete KDDCUP99 [13] dataset and does not suffer from any of mentioned shortcomings.

[12]Proposed a new learning algorithm for adaptive network intrusion detection using Naive Bayesian classifier and decision tree, which performs balance detections and keeps false positives at an acceptable level for different types of network attacks, thus eliminating redundant attributes as well as contradictory examples from training data that make the detection model complex. Panda and Patra [14] used Naive Bayes for anomaly detection and achieved detection rate of 95%. Faroun and Boukelif [15] used Neural Networks with K-mean clustering and showed detection rate of 92%. Gaddam and Phoha [16] proposed a method to cascade clustering and decision tree for classifying anomalous and normal data. We used the dataset KDDCUP'99 in our research as proposed by [3] and then we proposed some improvements changes in the dataset through preprocessing to reduce the number of attributes from 42 to 27. Using the modified dataset, a study was conducted on the problem of Intrusion Detection using data mining, Initially, we tested a Decision Tree and a Neural network using this dataset, suggesting improvements in it, by reducing the number of attributes from 42 to 27 and considering only two detection classes normal and intrusion. Following the simulation we discuss some of the improvements in the work of [3]. Then, using the original KDDCUP'99 Dataset [13], we solved the same problem using two classifiers (Decision Tree and Bayesian networks) considering five classes of detection: Normal, DOS,R2L,U2R and Probing. In section 2 we presented the considerations about KDDCUP'99 Dataset are presented, in section 3 the concepts about Intrusion Detection Systems are discussed, in section 4 we presented

the Description of the used algorithms, in section 5 a description and discussion of the experiments, and finally the conclusions on section 6.

2 Considerations about the KDDCUP'99 Dataset

The KDDCUP'99 dataset was used in the 3rd International Knowledge Discovery and Data Mining Tools Competition for building a network intrusion detector. In 1998, DARPA intrusion detection evaluation program, a simulated environment was set up to acquire raw TCP/IP dump data for a local-area network (LAN) by the MIT Lincoln Lab to compare the performance of various intrusion detection methods. The KDDCUP'99 dataset contest uses a version of DARPA'98 dataset[12]. DARPA98 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 [3]. Attack types were divided into 4 main categories as follow: **i. Probing Attack** is an attempt to gather information about a network of computers for the apparent purpose of circumventing its security controls. **ii. Denial of Service (DOS)** Denial of Service (DOS) is a class of attacks where an attacker makes some computing or memory resource too busy or too full to handle legitimate requests, denying legitimate users access to a machine. **iii. User to root (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. **iv. Remote to user (R2L)** This attack happens when an attacker sends packets to a machine over a network that exploits the machines vulnerability to gain local access as a user illegally. There are different types of R2U attacks; the most common attack in this class is done by using social engineering. In the KDDCUP'99 dataset these attacks (DoS, U2R, R2L, and probe) are divided into 22 different attacks types that are tabulated in Table 1. Not only they refer to the specific case of KDDCUP'99 Dataset, there are lots of known computer system attack classifications and taxonomies, some of them have been analyzed in this research [19].

Table 1. Different Types of attacks in KDDCUP'99 Dataset

Attack Classes	22 Types of Attacks
DoS	back,land,neptune,pod,smurt,teardrop
R2L	ftp-write,guess-passwd,imap,multihop,phf,spy,warezclient,warezmaster
U2R	buffer-overflow,perl,loadmodule,rootkit
Probing	ipsweep,nmap,portsweep,satan

2.1 Inherent Problems of KDDCUP'99 DataSet

The total number of records in the original labeled training dataset is 972.781 for Normal, 41.102 for Probe, 3.883.370 for DoS, 52 for U2R, and 1.126 for R2L attack classes. One of the most important deficiencies in the KDD data set is the huge number of redundant records, which causes the learning algorithms to be biased towards the frequent records, and thus prevent them from learning unfrequented records which are usually more harmful to networks such as, U2R and R2L attacks. Besides, the existence of these repeated records in the test set will lead to biased evaluation results by the methods with better detection rates on the frequent records. We addressed this matter by removing all the repeated records on both KDD train and test set, and kept only one copy of each record. Tables 2 and 3 show the statistics of repeated records on the KDD train and test sets, respectively.

Table 2. Statistics of Redundant Records in the KDD Train Set [3]

	Original Records	Distinct Records	Reduction Rate
Attacks	3.925.650	262.178	93.32%
Normal	972.781	812.814	16.44%
Total	4.898.431	1.074.992	78.05%

Table 3. Statistics of Redundant Records in the KDD Test Set [3]

	Original Records	Distinct Records	Reduction Rate
Attacks	250.436	29.378	88.26%
Normal	60.591	47.911	20.92%
Total	311.027	77.289	75.15%

3 Intrusion Detection Overview

Intrusion detection (ID) is a type of security management system for computers and networks. An ID system gathers and analyzes information from various areas within a computer or a network to identify possible security breaches, which include both intrusions (attacks from outside the organization) and misuse (attacks from within the organization). A network based IDS (NIDS) monitor and analyze network traffics, and use multiple sensors for detecting intrusions from internal and external networks [17]. IDS analyze the information gathered by the sensors, and return a synthesis of the input of the sensors to system administrator or intrusion prevention system. System administrator carries out the prescriptions controlled by the IDS. Today, data mining has become an indispensable tool for analyzing the input of the sensors in IDS. Ideally, IDS should have an attack detection rate (DR) of 100% along with false positive (FP) of 0%. Nevertheless, in practice this is really hard to achieve. The most

Table 4. Parameters for performance estimation of IDS[2]

Parameters	Definition
True Positive (TP) or Detection Rate (DR)	Attack occur and alarm raised
False Positive (FP)	No attack but alarm raised
True Negative (TN)	No attack and no alarm
False Negative (FN)	Attack occur but no alarm

important parameters involved in the performance estimation of IDS are shown in Table 4.

Detection rate (DR) and false positive (FP) are used to estimate the performance of IDS [18] which are given as bellow:

$$DR = \frac{Total\_Detected\_Attacks}{Total\_Attacks} * 100 \tag{1}$$

$$FP = \frac{Total\_Misclassified\_Process}{Total\_Normal\_Process} * 100 \tag{2}$$

4 Description of the Used Algorithms

An Artificial Neural Network (ANN) is an information processing paradigm that is inspired by the way biological nervous systems, such as the brain, process information. It is composed of a large number of highly interconnected processing elements (neurons) working in unison to solve specific problems. ANNs, like people, learn by example. An ANN is configured for a specific application, such as pattern recognition or data classification, through a learning process. Learning in biological systems involves adjustments to the synaptic connections that exist between the neurons. This is true of ANNs as well. In this study we use multilayer neural network (MLP) employing backpropagation algorithm.

Decision trees are a classic way to represent information from a machine learning algorithm, and offer a fast and powerful way to express structures in data. The J48-WEKA algorithm used to draw a Decision Tree. The same is a version of an earlier algorithm developed by J. Ross Quinlan, the very popular C4.5.

Bayesian networks (BNs), belong to the family of probabilistic graphical models . A Bayesian network, or belief network, shows conditional probability and causality relationships between variables. The probability of an event occurring given that another event has already occurred is called a conditional probability. The probabilistic model is described qualitatively by a directed acyclic graph. The vertices of the graph, which represent variables, are called nodes. The nodes are represented as circles containing the variable name. The connections between the nodes are called arcs or edges. The edges are drawn as arrows between the nodes, and represent dependence between the variables.

5 Methodology and Experiments

Nowadays data mining has become an indispensable tool for analizing the input of used sensors in IDS. The objective of this research is to simulate a network

traffic analyzer that is part of an IDS, as described in the abstract, to do this we tested the performance of three classifiers by employing the KDDCUP99 dataset and selected the best classifiers based on the parameters described in Table 4. The methodology used in this study used the data mining steps that consists of three stages: (1) the initial exploration - this stage usually starts with data preparation which may involve cleaning data, data transformations, selecting subsets of records and in case of data sets with large numbers of variables ("fields") - performing some preliminary feature selection operations to bring the number of variables to a manageable range (depending on the statistical methods which are being considered) (2) model building or pattern identification with validation/verification - this stage involves considering various models and choosing the best one based on their predictive performance and (3) deployment - that final stage involves using the model selected as best in the previous stage and applying it to new data in order to generate predictions or estimates of the expected outcome.

In the initial experiments, we used the modified KDDCUP'99 dataset, proposed by [3]. However some modifications were initially made by reducing the numbers of attributes from 42 to 27, for the following reasons: using statistics of software "Weka", some attributes that had unique value were eliminated, among them "num_outbound_cmds" and "is_host_login." Additionally, we eliminated attributes with high correlation coefficient, it was considered attributes strongly correlated those with correlation coefficients greater than or equal to 0.8. Our aim was to make the selection of attributes instead of synthesis, reason why we eliminated these attributes[1]. Highly correlated attributes influence each other and bring little information, as a result it is not interesting to maintain them in the data set, and so we used PCA (Principal Components Analysis) available in the "Weka". The following attributes were removed : sensor_rate, same_srv_rate, srv_serror_rate, st_host_srv_serror_rate, rerror_rate, srv_rerror_rate, srv_count. It is important to mention that Data mining is a step in the KDD process that consists of applying data analysis and discovery algorithms that produce a particular enumeration of patterns (or models) over the data. We focus primarily on the statistical approach to model fitting, which tends to be the most widely used basis for practical data mining applications given the typical presence of uncertainty in real world data generating processes, in other words the modifications made to the dataset reflect real world conditions.

Some attributes were selected and normalized: wrong_fragment, count, duration num_failed_logins, num_compromised,dst_host_srv_rerror_rate,num_file_cre ations,num_access_files and dst_host_count, these values were normalized with the values assumed in the interval [0,1]. Normalization is necessary in order to provide the data the same order of magnitude. Without this procedure some quantities could have existed quantities which would be more important than others. .Once the changes were made the "dataset" provided by [16] now has 27 attributes, as a result we obtained some improvements in relation to the performance of the classifiers, decision tree and neural network used in the study of [3] whose simulation results are shown on Table 5. The first classifier used was a

decision tree(algorithm J48 from "Weka") to conduct training and testing. The algorithm J48 is an implementation of the C4.5 algorithm in java. In Table 5, it is clear that by using the test data and decision tree, we obtained a detection rate of 99.4% for normal connections and 91.1% for intrusion and false positives of 8.9% and 6%. During the simulations with a neural network we used the following parameters 23 neurons in the input layer, two in the intermediate layer, an one in the output layer, learning rate 0.3, momentum 0.2, sigmoid activation function for all neurons, 50000 epochs, For the test data we obtained a detection of 95% for normal connections, 92.3% for intrusion and 7.7% of false positives for normal connections and 5% for intrusion. The total number of instances correctly classified by the decision tree, was 95.12% and in the work of [3] was 93.82% by the neural network was 93.47% and in the work of [3] was 92.26%, thus reducing attributes according to the techniques previously shown which favored a better performance of the classifiers with respect to the results obtained by the work of [3]. In the experiments described above we used the "modified dataset" proposed by[3] considering only two classification classes : normal and intrusion. The dataset proposed in [3] is suitable, however the changes made in this study show that the results obtained by the classifiers are better. We made additional experiments, using the dataset proposed by [13], adopting the five classes of attack: Normal, DOS, Probing, R2L and U2R. Some modifications were made in the dataset, before carrying out the next step, data mining : we eliminated the single_valued attributes num_outbound_cmds and is_host_login. After selecting these attributes we normalize the following attributes: wrong_fragment, num_failed _logins, num_compromised, num_file_creations, num_access_files, count, dst_host_count and duration, these values were normalized with the values assumed in the interval [0,1]. Following, there was a balance between the classes, selecting records in a manner inversely proportional to the occurrences in accordance with Table 7. While doing this process, we encountered two invalid records in the KDD test set, number 136.489 and 136.497. These two records contain an invalid value, ICMP, as their service feature. Therefore, we removed them from the KDD test set. After the changes were made in the dataset, two classifiers were used in the simulations , Decision Tree algorithm (J48) and the Bayesian Network whose results are shown on Table 6. The decision tree scored better than the Bayesian network, especially in the classification of instances belonging to the class of R2L attacks, the decision tree correctly classified 95.2% of the records of this class, since the Bayesian network, only managed to correctly classify 69.3%. As for the other classes (Normal, Probe, DOS, U2R), the performance of both classifiers was similar. Comparing the results presented in this study, Table 6, with the work of [12] which used a hybrid system employing the algorithm ID3 with a Naive Bayes classifier, the results of [12] were better, except for the detection of false positive of R2L class. The following values were obtained by [12] for detection rate (DR%) and false positive (FP%): normal 99.72% and 0.06%, probe 99.25% and 0.39% Dos 99.75% and 0.04%, U2R 99.20% and 0.11%, R2L 99.26% and 6.81%. However, the results obtained in the second experiment, Table 6, are acceptable. For R2L class we achieved better

Table 5. Results for test using J48 Decision Tree and Neural Network MLP, given the dataset provided by [3] and the reduction of attributes from 42 to 27

Classifier	Normal	Intrusion
Decision Tree (DR%)	99.4%	91.1%
Decision Tree (FP%)	8.9%	6%
Neural Network (DR%)	95%	92.3%
Neural Network (FP%)	7.7%	5%

Table 6. Results for the test dataset [3], considering five classes

Classifier	Normal	Probe	Dos	R2L	U2R
Decision Tree (DR%)	98.9%	98.3%	99.7%	95.2%	93.9%
Decision Tree (FP%)	0.04%	0.02%	0.03%	0.01%	0.01%
Bayesian Network (DR%)	99.1%	93.5%	98.7%	69.3%	90.03%
Bayesian Network (FP%)	0.13%	0.05%	0.02%	0.14%	0.06%

Table 7. Number of Records by class in the Kddcup99 (10%) Dataset[13], with proposed reductions for train and test

Class attack	Dataset (10%)	Train	Test
Normal	97294	12607	2887
Denial of Service (DOS)	391458	36929	9607
Remote to User (R2L)	1113	911	202
User to Root (U2R)	51	31	21
Probing	4106	1247	293
Total	494022	51816	13020

results, such as 0.01 for false positive (FP%) and [12] obtained 6.81%. while the DoS attack type appears in 79% of the connections, the U2R and R2L attack types only appears in 0.01% and 0.225% of the records respectively. And these attacks types are more difficult to predict and and the more costly if missed.

Some rules extracted from decision tree used in the simulations of the second experiment, using the J48 algorithm, are illustrated below . The first rule associated with attacks of type "probe" corresponds to the detection of open ports and services on a live server used during an attack. The third rule concerns the "scans" performed on multiple hosts looking for open ports (e.g. TCP port = 1433). The rules associated with attacks of type R2L are characteristic of access to e-mail box and operations of download and upload files. The last two rules, which identify the type U2R attacks identify hidden files trying to evade antivirus programs running on the client machine overloading the buffer.

RULE 1 (DOS) - IF((flag=REJ OR flag=RSTO OR flag=SO) AND land LESS THAN 0.5) THEN label = neptune.

RULE 2 (Probe) - IF (count LESS THAN 3.5 AND (service=eco_i OR service=ecr_i) THEN label=ipsweep.

RULE 3 (R2L) - IF (service=pop_3 OR service=telnet) THEN label=guess _passwd.IF((service=pop_3 OR service=telnet) AND (num_failed_logins LESS THAN 0.5) AND (flag!=REJ OR flag!=RSTO) AND (service=http OR service=login)) THEN label=ftp_write.

RULE 4 (U2R) - IF (dst_bytes LESS THAN 665.5) THEN label=rootkit. IF (dst_bytes GREATER THAN 665.5) THEN label=Buffer_overflow

6 Conclusions

In the models proposed in this study some simplifications were considered: "Probing" is not necessarily a type of attack except if the number of iterations exceeds a specific threshold. Similarly a packet that causes a buffer overflow is not necessarily an attack. Traffic collectors such as TCP DUMP that is used in DARPAS'98 are easy to be overwhelmed and drop packets in heavy traffic, were not checked the possibilities of packets dropped. The "dataset" proposed by [3] which consists of selected records of the "dataset" has unique advantages, as it does not include redundant or duplicate records which could bias the results obtained by the classifiers. However, the experiments in this study showed that it is possible to reduce the number of attributes from 42 to 27, improving the performance of the decision tree classifiers and neural network in accordance with the results shown in Table 5 and compared to the work of [3].

In experiments performed with the original dataset [13] and modified according to Table 7 and Table 6, we conclude that the initial stages of the process of knowledge discovery in databases: data selection, pre-processing and transformation are essential for the data mining. The procedures used in the second experiment: attribute selection, data normalization, allowed to obtain satisfactory results shown in Table 6 that are within acceptable standards in accordance with the results presented in [12]. The results obtained by [12] were higher than those shown in Table 6, because the use of hybrid systems proposed by [12]. However [12] did not provide the decision rules obtained by decision tree, what is important in the process of knowledge discovery in database.

Tests were performed with a group of classifiers, where the best classifiers for both cases were the decision tree (J48 algorithm) and Neural Network. The Bayesian network was not a good classifier for five classes of attacks. We believe that the methodology presented in this study may help researchers to compare different methods of intrusion detection.

References

1. Stolfo, S.J., et al.: KDD cup 1999 data set. KDD repository. University of California, Irvine, `http://kdd.ics.uci.edu`
2. Landwehr, C.E., Bull, A.R., McDermott, J.P., Choi, W.S.: A taxonomy of computer program security flaws. ACM Comput. Surv. 26(3), 211–254 (1994)
3. Tavallaee, M., Bagheri, E., Lu, W., Ghorbani, A.: A Detailed Analysis of the KDD CUP 99 Data Set. Submitted to Second IEEE Symposium on Computational Intelligence for Security and Defense Applications, CISDA (2009)

4. Quinlan, J.: C4.5: Programs for Machine Learning. Morgan Kaufmann (1993)
5. John, G., Langley, P.: Estimating continuous distributions in Bayesian classifiers. In: Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence, pp. 338–345 (1995)
6. Kohavi, R.: Scaling up the accuracy of naive-Bayes classifiers: A decision-tree hybrid. In: Proceedings of the Second International Conference on Knowledge Discovery and Data Mining, vol. 7 (1996)
7. Breiman, L.: Random Forests. Machine Learning 45(1), 5–32 (2001)
8. Aldous, D.: The continuum random tree. I. The Annals of Probability, 1–28 (1991)
9. Ruck, D., Rogers, S., Kabrisky, M., Oxley, M., Suter, B.: The multilayer perceptron as an approximation to a Bayes optimaldiscriminant function. IEEE Transactions on Neural Networks 1(4), 296–298 (1990)
10. Chang, C., Lin, C.: LIBSVM: a library for support vector machines (2001), Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`
11. Waikato environment for knowledge analysis (weka) version 3.5.7 (June 2008), `http://www.cs.waikato.ac.nz/ml/weka/`
12. Farid, D.M., Harbi, N., Rahman, M.Z.: Combining naive Bayes and Decision Tree for adaptative Intrusion Detection. International Journal of Network Security & Its Applications (IJNSA) 2(2) (April 2010)
13. KDD Cup 1999 (October 2007), `http://kdd.ics.uci.edu/datasets/kddcup99/kddcup99.html`
14. Panda, M., Patra, M.R.: Network intrusion detection using naive bayes. IJCSNS (2006)
15. Faroun, K.M., Boukelif, A.: Neural network learning improvement using k-means clustering algorithm to detect network intrusions. IJCI (2006)
16. Gaddam, S.R., Phoha, V.V., Balagani, K.S.: 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)
17. Wasniowski, R.: Multi-sensor agent-based intrusion detection system. In: Proc. of the 2nd Annual Conference on Information Security, Kennesaw, Georgia, pp. 100–103 (2005)
18. Chen, R.C., Chen, S.P.: Intrusion detection using a hybrid support vector machine based on entropy and TF-IDF. International Journal of Innovative Computing, Information, and Control (IJICIC) 4(2), 413–424 (2008)
19. Alvarez, G., Petrovic, S.: A new taxonomy of web attacks suitable for efficient encoding. Computers and Security 22(5), 435–449 (2003)