NETWORK INTRUSION DETECTION USING NAÏVE BAYES

Mrutyunjaya Panda[1] and Manas Ranjan Patra[2]

[1]Department of E &TC Engineering, G.I.E.T., Gunupur, India
[2]Department of Computer Science, Berhampur University, Berhampur, India

Summary
With the tremendous growth of network-based services and sensitive information on networks, network security is getting more and more importance than ever. Intrusion poses a serious security risk in a network environment. The ever growing new intrusion types posses a serious problem for their detection. The human labelling of the available network audit data instances is usually tedious, time consuming and expensive. In this paper, we apply one of the efficient data mining algorithms called naïve bayes for anomaly based network intrusion detection. Experimental results on the KDD cup'99 data set show the novelty of our approach in detecting network intrusion. It is observed that the proposed technique performs better in terms of false positive rate, cost, and computational time when applied to KDD'99 data sets compared to a back propagation neural network based approach.

Keywords:
Network Security, Intrusion Detection, Data Mining, Naive Bayes classifier, ROC.

1. INTRODUCTION

With the tremendous growth of network-based services and sensitive information on networks, network security is becoming more and more importance than ever before. Intrusion detection techniques are the last line of defences against computer attacks behind secure network architecture design, firewalls, and personal screening. Despite the plethora of intrusion prevention techniques available, attacks against computer systems are still successful. Thus, intrusion detection systems (IDSs) play a vital role in network security. Symantec in a recent report[1] uncovered that the number of fishing attacks targeted at stealing confidential information such as credit card numbers, passwords, and other financial information are on the rise, going from 9 million attacks in June2004 to over 33 millions in less than a year.
One solution to this is the use of network intrusion detection systems (NIDS), that detect attacks by observing various network activities. It is therefore crucial that such systems are accurate in identifying attacks, quick to train and generate as few false positives as possible. This paper presents the scope and status of our research in anomaly detection. This paper gives a comparative study of several anomaly detection schemes for identifying novel network intrusion detections. We present experimental results on KDDCup'99 data set. Experimental results have demonstrated that our naïve bayes classifier model is much more efficient in the detection of network intrusions, compared to the neural network based classification techniques. Section 2 describes IDS in general. section 3 presents an overview of frequently occurring network attacks, and section 4 discusses related research done so far. Section 5 describes our proposed method and section 6 presents the experimental results. Finally, section 7 provides the concluding remarks and future scope of the work.

2. INTRUSION DETECTION

An Intrusion Detection System (IDS) inspects the activities in a system for suspicious behaviour or patterns that may indicate system attack or misuse. There are two main categories of intrusion detection techniques; Anomaly detection [2] and Misuse detection. The former analyses the information gathered and compares it to a defined baseline of what is seen as "normal" service behaviour, so it has the ability to learn how to detect network attacks that are currently unknown. Misuse Detection is based on signatures for known attacks, so it is only as good as the database of attack signatures that it uses for comparison. Misuse detection has low false positive rate, but cannot detect novel attacks. However, anomaly detection can detect unknown attacks, but has high false positive rate.

In this paper, we review the performance of classifiers when trained to identify signatures of specific attacks. These attacks are discussed in more detail in the following section.

3. NETWORKING ATTACKS

The simulated attacks were classified, according to the actions and goals of the attacker. Each attack type falls into one of the following four main categories [3]:

Manuscript received December 5, 2007
Manuscript revised December 20, 2007

- Denials-of Service (DoS) attacks have the goal of limiting or denying services provided to the user, computer or network. A common tactic is to severely overload the targeted system. (e.g. apache, smurf, Neptune, Ping of death, back, mailbomb, udpstorm, SYNflood, etc.).

- Probing or Surveillance attacks have the goal of gaining knowledge of the existence or configuration of a computer system or network. Port Scans or sweeping of a given IP-address range typically fall in this category. (e.g. saint, portsweep, mscan, nmap, etc.).
- User-to-Root (U2R) attacks have the goal of gaining root or super-user access on a particular computer or system on which the attacker previously had user level access. These are attempts by a non-privileged user to gain administrative privileges (e.g. Perl, xterm, etc.).
- Remote-to-Local(R2L) attack is an attack in which a user sends packets to a machine over the internet, which the user does not have access to in order to expose the machine vulnerabilities and exploit privileges which a local user would have on the computer (e.g. xclock, dictionary, guest_password, phf, sendmail, xsnoop, etc.).

4. RELATED WORK

ADAM (Audit Data Analysis and Mining) [4] is an intrusion detector built to detect intrusions using data mining techniques. It first absorbs training data known to be free of attacks. Next, it uses an algorithm to group attacks, unknown behaviour, and false alarms. ADAM has several useful capabilities, namely;

- ✓ Classifying an item as a known attack
- ✓ Classifying an item as a normal event,
- ✓ Classifying an item as an unknown attack,
- ✓ Match audit trial data to the rules it gives rise to.

IDDM (Intrusion Detection using Data Mining Technique) [5] is a real-time NIDS for misuse and anomaly detection. It applies association rules, meta rules, and characteristic rules. It employs data mining to produce description of network data and uses this information for deviation analysis.
MADAM ID (Mining Audit Data for Automated Models for Intrusion Detection) [6] is one of the best known data mining projects in intrusion detection. It is an off-line IDS to produce anomaly and misuse intrusion detection models. Association rules and frequent episodes are applied in MADAM ID to replace hand-coded intrusion patterns and profiles with the learned rules.

In [7], the authors propose a method of intrusion detection using an evolving fuzzy neural network. This type of learning algorithm combines artificial neural network (ANN) and fuzzy Inference systems (FIS), as well as evolutionary algorithms. They create an algorithm that uses fuzzy rules and allow new neurons to be created in order to accomplish this. They use Snort to gather data for training the algorithm and then compare their technique with that of an augmented neural network.

In [8], a statistical neural network classifier for anomaly detection is developed, which can identify UDP flood attacks. Comparing different neural network classifiers, the back propagation neural network (BPN) has shown to be more efficient in developing IDS [9]. In [9], the author uses the back propagation method by Sample Query and Attribute Query for the Intrusion Detection, whereby analysing and identifying the most important components of training data. It could reduce processing time, storage requirement, etc.

In [10], Axellson wrote a well-known paper that uses the Bayesian rule of conditional probability to point out that implication of the base-rate fallacy for intrusion detection. In [11], a behaviour model is introduced that uses Bayesian techniques to obtain model parameters with maximal a-posteriori probabilities. Their work is similar to our, to the extent that Bayesian statistics are employed. However, the difference lies in that; we use naïve bayes for our model.

5. THE PROPOSED METHOD

The Naïve Bayes method is based on the work of Thomas Bayes (1702-1761). In Bayesian classification, we have a hypothesis that the given data belongs to a particular class. We then calculate the probability for the hypothesis to be true. This is among the most practical approaches for certain types of problems. The approach requires only one scan of the whole data. Also, if at some stage there are additional training data, then each training example can incrementally increase/decrease the probability that a hypothesis is correct. Thus, a Bayesian network is used to model a domain containing uncertainty {12, 13].
Consider the following example where a farmer has a bottle of milk that can be either infected or clean. She also has a test that determines with a high probability whether the milk is infected or not (i.e. the outcome of the test is either positive or negative). This situation can be represented with two random variables, infected and positive. The variable infected is true when the milk is

actually infected and false otherwise. The variable positive is true when the test claims that the milk is infected and false when the outcome of the test is negative. Note that, it is possible that the milk is clean when the test data has a positive outcome and vice-versa.

Naïve Bayes:
The naïve Bayes model is a heavily simplified Bayesian probability model [14]. In this model, consider the probability of an end result given several related evidence variables. The probability of end result is encoded in the model along with the probability of the evidence variables occurring given that the end result occurs. The probability of an evidence variable given that the end result occurs is assumed to be independent of the probability of other evidence variables given that end results occur. Now we will consider the alarm example using a naïve Bayes classifier. Assume that we have a set of examples that monitor some attributes such as whether it is raining, whether an earthquake has occurred etc. Lets assume that we also know, using the monitor, about the behaviour of the alarm under these conditions. In addition, having knowledge of these attributes, we record whether or not a theft actually occurred. We will consider the category of whether a theft occurred or not as the class for the naïve Bayes classifier. This is the knowledge that we are interested in. The other attributes will be considered as knowledge that may give us evidence that the theft has occurred.
Figure1 below shows the framework for a Naïve Bayesian model to perform intrusion detection.
The naïve Bayes classifier operates on a strong independence assumption [14]. This means that the probability of one attribute does not affect the probability of the other. Given a series of n attributes, the naïve Bayes classifier makes 2n! independent assumptions. Nevertheless, the results of the naïve Bayes classifier are often correct. The work reported in [15] examines the circumstances under which the naïve bayes classifier performs well and why. It states that the error is a result of three factors: training data noise, bias, and variance. Training data noise can only be minimised by choosing good training data. The training data must be divided into various groups by the machine learning algorithm. Bias is the error due to groupings in the training data being very large. Variance is the error due to those groupings being too small.

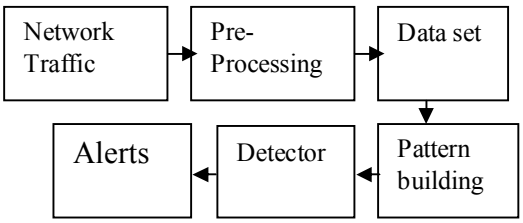

Figure1. The framework of the Intrusion Detection Model.

In the training phase, the naïve bayes algorithm calculates the probabilities of a theft given a particular attribute and then stores this probability. This is repeated for each attribute, and the amount of time taken to calculate the relevant probabilities for each attribute. In the testing phase, the amount of time taken to calculate the probability of the given class for each example in the worst case is proportional to n, the number of attributes. However, in worst case, the time taken for testing phase is same as that for the training phase.

6. EXPERIMENT AND RESULTS

In this section, we summarize our experimental results to detect network intrusion detections using the naïve Bayes algorithm over KDDCup'99 data set. We first describe the data set used in this experiment and then discuss the results obtained. Finally, we evaluate our approach and compare the results with the results obtained by other researchers using BPN algorithms and with the best result of the KDD'99contest.

Dataset and pre processing
Under the sponsorships of Defence Advanced Research projects Agency (DARPA) and Air force Research Laboratory (AFRL), MIT Lincoln Laboratory has collected and distributed the datasets for the evaluation of computer network intrusion detection systems [16, 17]. DARPA dataset is the most popular data set used to test and evaluate a large number of IDSs. The KDD'99 dataset is a subset of DARPA dataset prepared by Sal Stolfo and Wenke Lee [18]. The data set was pre-processed by extracting 41 features from the tcpdump data in the 1998 DARPA data set. The KDD'99 dataset can be used without further time-consuming pre-processing and different IDS can be compared with each other by working on the same dataset. Therefore, we carry out our experiment on 10% of the KDD'99 dataset, which contains 65,525 connections.

For our experiments, we choose the naïve Bayes Classifier in WEKA (Waikato Environment for

Knowledge Analysis) [19]: with full training set and 10-fold cross validation for the testing purposes. In 10-fold cross-validation, the available data is randomly divided into 10 disjoint subsets of approximately equal size. One of the subsets is then used as the test set and the remaining 9 sets are used for building the classifier. The test set is then used to estimate the accuracy. This is done repeatedly 10 times so that each subset is used as a test subset once. The accuracy estimates is then the mean of the estimates for each of the classifiers. Cross-validation has been tested extensively and has been found to generally work well when sufficient data is available. A value of 10 for this has been found to be adequate and accurate. Finally, the ROC (Receiver Operating Characteristic) curve is obtained as a measure of performance analysis of our approach, using MATLAB7.0. The experiment is carried out using a machine with Intel Pentium4 processor, 2.8GHz speed, and 512MB RAM.

Evaluation and Discussion:

We carried out the experiment over 10% KDDCup'99 data set. We evaluate the performance of our system by the detection rate and the false positive rate.

The detection rate is the number of attacks detected by the system divided by the number of attacks in the data set. The false positive rate is the number of normal connections that are misclassified as attacks divided by the number of normal connections in the data set.

Next, we calculate the error rate, which is an estimate of the true error rate and is expected to be a good estimate, if the number of test data is large and representative of the population. It is defined as follows:
Error Rate = (Total test data ─ total correctly classified data) / Total test data.
A "Confusion Matrix" is sometimes used to represent the result of testing, as shown in Table 1.The Advantage of using this matrix is that it not only tells us how many got misclassified but also what misclassifications occurred.

Table1: Experimental result in Confusion Matrix

Predicted / Actual	**Probe**	**DoS**	**U2R**	**R2L**
Probe				
Our results	756	4	1	27
BPNSQAQ	2523	81	509	8
BPN	564	181	0	0
DoS				
Our results	0	23349	19	297
BPNSQAQ	564	227080	126	0
BPN	25	222153	0	0
U2R				
Our results	1	1	38	2
BPNSQAQ	25	0	83	8
BPN	0	0	0	0
R2L				
Our results	0	1	5	54
BPNSQAQ	14	479	147	6660
BPN	4	2	0	0
FPR				
Our results	0.0014	0.26	0.000163	0.00025
BPNSQAQ	0.242	0.009	0.908	0.235
BPN	0.0896	0.0042	0.0	0.0
Precision Rate				
Our results	96%	99%	90.47%	90%
BPNSQAQ	75.8%	99.1%	9.2%	76.5%
BPN	91.1%	99.6%	00.0%	00.0%
Recall Rate				
Our results	99.8%	99.5%	60.3%	14.2%
BPNSQAQ	60.7%	98.8%	36.4%	41.2%
BPN	60.3%	96.7%	00.0%	00.0%
FNR				
Our results	0.13% 85.8%	0.02%		39.68%
BPNSQAQ	39.4% 58.6%	1.20%		63.6%
BPN	39.6% 100.0%	03.4%		100.0%

We plot a ROC (Receiver Operating Characteristic) curve which is often used to measure performance of IDS. The ROC curve is a plot of the detection rate against the false positive rate, which is shown in fig2.

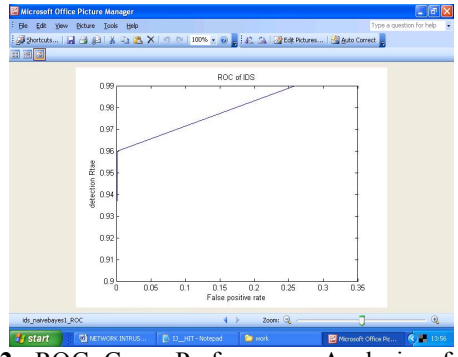

Fig2. ROC Curve-Performance Analysis of Intrusion Detection using Naive Bayes

Next, we build a cost matrix as in table 2. The cost matrix can be used to measure the damage of mis-classification [18].

Let $\mathbf{M_{ij}}$ denote the number of samples in class misclassified as class j
$\mathbf{C_{ij}}$ indicate the corresponding cost in the cost matrix.
N be the total number of samples.

Then the cost that indicates the average damage of misclassification for each connection is computed as:

$$\text{Cost}=\sum M_{ij} \times C_{ij}/N$$

The Cost-Performance comparison on the KDD'99 dataset is shown in table3.

Table 2. Cost Matrix

	Normal	Probe	DoS	U2R	R2L
Normal	0	1	2	2	2
Probe	1	0	2	2	2
DoS	2	1	0	2	2
U2R	3	2	2	0	2
R2L	4	2	2	2	0

Table 3. Performance Comparison on the KDD'99 dataset

Experiments	Overall Error Rate	Cost	Time in Seconds
Best KDD Result	7.29%	0.2331	Not Provided
Ours	5.1%	0.16	1.89

In [20] which uses NN using K-means clustering shows that the detection rate and execution run time in detecting intrusion is 92% and 28m21s. However, in our case, the detection rate is 95%, with an error rate of 5%. Moreover, it performs faster which takes only 1.89 seconds to build the model. However, in comparison to BPN, our approach generates more false positives, but, it is efficient, cost effective and takes less time.

7. CONCLUSION AND FUTURE WORK

In this paper, we have proposed a framework of NIDS based on Naïve Bayes algorithm. The framework builds the patterns of the network services over data sets labelled by the services. With the built patterns, the framework detects attacks in the datasets using the naïve Bayes Classifier algorithm. Compared to the Neural network based approach, our approach achieve higher detection rate, less time consuming and has low cost factor. However, it generates somewhat more false positives.

As a naïve Bayesian network is a restricted network that has only two layers and assumes complete independence between the information nodes. This poses a limitation to this research work. In order to alleviate this problem so as to reduce the false positives, active platform or event based classification may be thought of using Bayesian network. We continue our work in this direction in order to build an efficient intrusion detection model.

REFERENCES

[1] "Symantec-Internet Security threat report highlights (Symantec.com)", http://www.prdomain.com/companies/Symantec/newreleases/Symantec_internet_205032.htm

[2] R.Durst, T.champion, B.witten, E.Miller, and L.Spagnuolo, "Testing and evaluating computer intrusion detection system" communications of ACM, Vol.42, no.7, pp 53-61, 1999.

[3] A.Sung & S.Mukkamala, "Identifying important features for intrusion detection using SVM and neural networks," in symposium on application and the Internet, pp 209-216, 2003.

[4] D.Barbara, J.Couto, S.Jajodia, and N.Wu, "ADAM: A test bed for exploring the use of data mining in intrusion detection", SIGMOD, vol30, no.4, pp 15-24, 2001.

[5] Tomas Abraham, "IDDM: INTRUSION Detection using Data Mining Techniques", Technical report DSTO electronics and surveillance research laboratory, Salisbury, Australia, May2001.

[6] Wenke Lee and Salvatore J.Stolfo, "A Framework for constructing features and models for intrusion detection systems", ACM transactions on Information and system security (TISSEC), vol.3, Issue 4, Nov 2000.

[7] S.chavan, K.Shah, N.Dave, S.Mukherjee, A.Abraham, and S.Sanyal, "Adaptive neuro-fuzzy Intrusion detection syatems", ITCC, Vol 1, 2004

[8] Z. Zhang, J. Li, C.N. Manikapoulos, J.Jorgenson, J.ucles, "HIDE: A hierarchical network intrusion detection system using statistical pre-processing and neural network classification", IEEE workshop proceedings on Information assurance and security, 2001, pp.85-90.

[9] Roy-I Chang, Liang-Bin Lai, et al, "Intrusion detection by back propagation network with sample query and attribute query", International Journal of computational Intelligence Research,Vol..3, no.1, 2007, pp 6-10.

[10] S. Axelsson, "The base rate fallacy and its implications for the difficulty of Intrusion detection", Proc. of 6th.ACM conference on computer and communication security 1999.

[11] R.Puttini, Z.marrakchi, and L. Me, "Bayesian classification model for Real time intrusion detection", Proc. of 22nd. International workshop on Bayesian inference and maximum entropy methods in science and engineering, 2002.

[12] P.Jenson, "Bayesian networks and decision graphs", Springer, New-york, USA, 2001.

[13] J.Pearl, "Probabilistic reasoning in intelligent system", Networks of plausible inference, Morgan Kaufmann 1997.

[14] S.J.Russel, and Norvig, "Artificial Intelligence: A modern approach (International edition), Pearson US imports & PHIPES, Nov 2002.

[15] P.Domingos, and M.J. Pizzani, "On the optimality of the simple Bayesian classifier under zero-one loss", m/c learning, Vol.29, no2-3, pp 103-130, 1997.

[16] M.Mahoney and P. chan, "An analysis of the 1999 DARPA/Lincoln laboratory evaluation data for network anomaly detection", Proc.of Recent Advances in intrusion detection (RAID)-2003, Pittsburg, USA, Sept. 2003.

[17] MIT Lincoln Laboratory, DARPA Intrusion detection Evaluation, http://www.ii.mit.edu.

[18] Charles Elkan, "Results of the KDD'99 classifier learning", SIGKDD Explorating192), pp. 63-64, 2000.
[19] WEKA: software machine learning, the University of Waikato, Hamilton, New-Zealand.
[20] K.M.Faroun, A.Boukelif, "Neural network learning improvement using K-means clustering algorithm to detect network intrusions", April17, 2006, http://www.dcc.ufla.br/infocomp/artigos/v5.3/art04.pdf

AUTHORS:

Mrutyunjaya Panda holds a Master Degree in Engineering and is presently working as an Assistant Professor in the Department of Electronics & Tele Commn. Engineering, Gandhi Institute of Engineering and Technology, Gunupur, India. He has 10 years of teaching experience. Currently, he is pursuing Doctoral research in Computer Science. He has about 7 publications to his credit. His research interests include Data Mining, Network Security, Intrusion Detection and Soft Computing.

Dr. Manas Ranjan Patra holds a Ph.D. degree in Computer Science from the Central University of Hyderabad and is presently working as a Reader in the Department of Computer Science, Berhampur University, India. He has worked in the International Institute for Software Technology, Macao as a United Nations Fellow during 2000. He has 20 years of experience in teaching and research in different areas of Computer Science. He has about 60 international and national publications to his credit. His research areas include Software Engineering, Data Mining, Intrusion Detection, Artificial Intelligence, and e-business. He has presented papers and chaired technical sessions in many International conferences. He is a member of number of professional bodies. He has executed visiting assignments to many Institutions and Universities.