ISSN : 0976-8491 (Online) | ISSN : 2229-4333 (Print)

Efficiency of Wireless Intrusion Detection Systems using Bayesian Classification Model

[1]D. Srinivasa Reddy, [2]Y. V. AdiSatyanarayana, [3]P. P. S. Naik, [4]B. Subba Reddy
[1,2,3,4]Dept. of CSE, Dr. Samuel George Institute of Engineering and Technology, Markapur, Prakasam, AP, India

Abstract
Network intrusion detection systems have become a standard component in security infrastructures. 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 poses a serious problem for their detection. Network intrusion detection systems have become a standard component in security infrastructures.. 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
Intrusion Detection Systems, Wireless Networks, Feature Selection

I. Introduction

Network Intrusion Detection Systems (NIDS) have become a standard component in security infrastructures as they allow network administrators to detect policy violations. These policy violations range the gamut from external attackers trying to gain unauthorized access (which can usually be protected against through the rest of the security infrastructure) to insiders abusing their access (which often times is not easy to protect against). Detecting such violations is a necessary step in taking corrective action, such as blocking the o ender (by blocking their machine at the parameter, or freezing their account), by reporting them (to their ISP or supervisor), or taking legal action against them. Alternatively, detecting policy violations allows administrators to identify areas where their defences need improvement, such as by identifying a previously unknown vulnerability, a system that wasn't properly patched, or a user that needs further education against social engineering attacks.

Security of network systems is becoming increasingly important as more and more sensitive information is being stored and manipulated online. Intrusion Detection Systems (IDSs) have thus become a critical technology to help protect these systems

Feature selection is the most critical step in building intrusion detection models. During this step, the set of attributes or features that deemed to be the most effective attributes are extracted in order to construct suitable detection algorithms (detectors). A key problem that many researchers face is how to choose the optimal set of features since not all features are relevant to the learning algorithm, and in some cases, irrelevant and redundant features can introduce noisy data that distracts the learning algorithm and therefore severely degrade the accuracy of the detector and cause slow training and testing process. Feature selection was proven to have a significant impact on the performance of the classifiers. Experiments in [1] show that feature selection can reduce the building and testing time of a classifier by up to 50%.

A Naïve Bayes classifier [4] is simple probabilistic classifier based on applying Bayes theorem with strong independence assumptions. A more descriptive term for the underlying probability model would be independent "feature model".

In simple terms, a Naive 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. Even if these features depend on each other or upon the existence of the other features, a Naive Bayes classifier considers all of these properties to independently contribute to the probability that this fruit is an apple.

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]:

1. 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.).
2. 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.).
3. 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.).
4. Remote-to-Local(R2L) attack is an attack inwhich 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.).

II. Exising System

Artificial Neural Networks (ANN) is a computational model that mimics the properties of biological neurons. A neuron, which is the base of an ANN, is described by a state, synapses, a combination function and a transfer function. The state of the neuron, which is a Boolean or real value, is the output of the neuron. Each neuron is connected to other neurons via synapses. Synapses are associated with weights that are used by the combination function to achieve a pre-computation, generally a weighted sum, of the inputs. Activation function, called also transfer function, computes, from the output of the combination function, the output of the neuron.

An artificial neural network is composed of a set of neurons grouped in layers that are connected by synapses. There are three types of layers: input, hidden and output layers. The input

layer is composed of input neurons that receive their values from external devices such as data files or input signals. The hidden layer which is an intermediary layer that contains neurons with the same combination and transfer functions. The output layer provides the output of the computation to the external applications.

A. Perceptron

Perceptron, fig. 1, is the simplest form of a neural network. It's used for classification of linearly separable problems. It consists of a single neuron with adjustable weights of the synapses. Even though the intrusion detection problem is not linearly separable; we use the perceptron architecture as reference to measure the performance of the other two types of classifiers

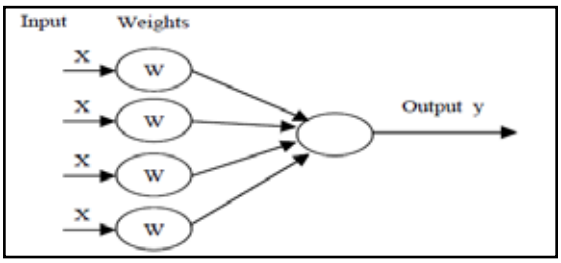

Fig. 1: Perceptron

B. Multi-Layer Back Propagation Perceptrons

The multi-layer back-propagation perceptrons architecture, fig. 2, is an organization of neurons in n successive layers (n>=3). The synapses link the neurons of a layer to all neurons of the following layer. Error propagation is done in the opposite direction of the information flow. We note that we use one hidden layer composed of 8 neurons.

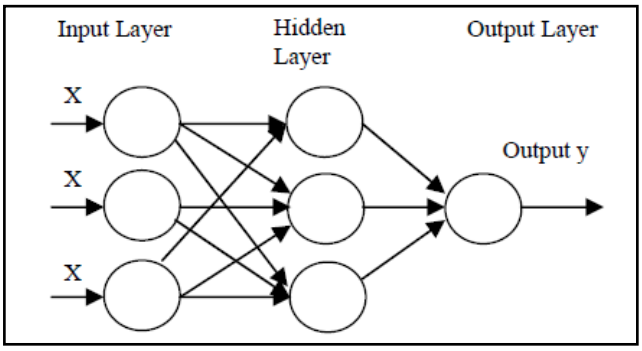

Fig. 2: Multi Layer Back Propagation Perceptrons

C. Hybrid Multi-Layer Perceptrons

Hybrid Multi-Layer Perceptrons architecture, fig. 3, is the superposition of perceptron with multi-layer back-propagation perceptrons networks. This type of network is capable of identifying linear and non linear correlation between the input and output vectors [7]. We used this type of architecture with 8 neurons in the hidden layer.

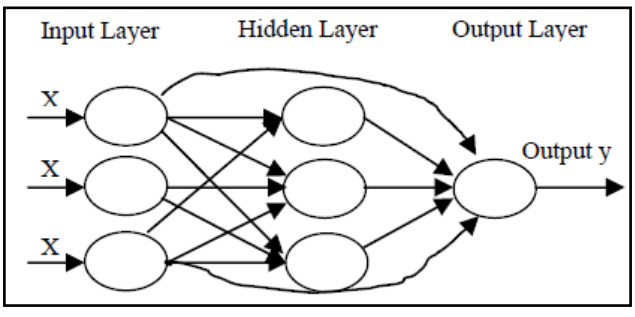

Fig. 3: Hybrid Multilayer Perceptrons

In [5] which use 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.

Naïve Bayes classifier 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.

III. Proposed System

In [2-3], we presented a complete framework to select the best set of MAC layer features that efficiently characterize normal traffic and distinguish it from abnormal traffic containing intrusions specific to wireless networks. Our framework uses a hybrid approach for feature selection that combines the filter and wrapper models [4]. In this approach, we rank the features using an independent measure: the information gain ratio. The Naïve Bayes classifier's predictive accuracy is used to reach an optimal set of features that maximizes the accuracy of detection of wireless attacks. To train the classifier, we first collected network traffic that contains four known wireless intrusions, which are de-authentication, duration, fragmentation, and chopchop attacks [5-6]. As shown in Table 1, the selection algorithm voted 8 features as the best set of features that maximizes the accuracy of the Naïve Bayes classifier.

Table 1: List of Optimal Set Features

Feature	Description
IsWepValid	Indicate if WEP ICV check is successful
Duration Range	Indicate if duration value is low(<5ms). Average (between 5-5ms), or high(>20ms)
MoreFragment	Indiacate whether a frame is non final fragment or not
ToDS	Indicate if a frame is destined to the distribution system
WEP	Indicate if the frame is processed by the WEP protocol
Type	Indicate the type of the frame
Subtype	Indicate the subtype of he frame

A. Datasets

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 [8-9].

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

The naïve Bayes model is a heavily simplified Bayesian probability model [6]. 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 behavior 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 [6]. 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

[7] 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 minimized 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.

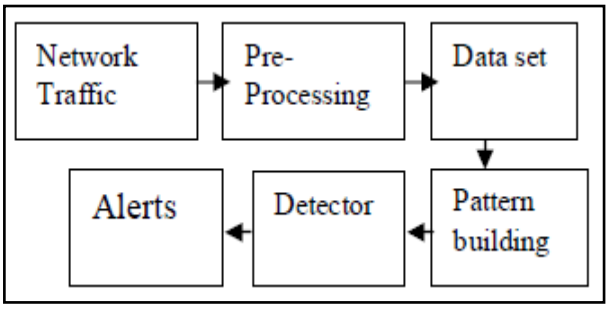

Fig. 4: The Frame Work of the Intrusion Detection Model

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.

IV. Experiment and Results

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.

A. 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 [9-10]. 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 [12]. The data set was preprocessed 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 preprocessing 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) [13]: 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.

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 dataset. The false positive rate is the number of normal connections that are misclassified as attacks divided by the number of normal connections in the dataset. 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.

Table 2: Experimental Result in Confusion Matrix

Predicted Actual	Prob	DoS	U2R	R2L
Prob Our result BPNSQ BPN	756 2523 564	4 81 181	1 509 0	27 8 0
DoS Our result BPNSQ BPN	0 564 25	23349 227048 222153	19 126 0	2 234 23
U2R Our results BPNSQ BPN	1 25 0	1 0 0	38 83 0	2 8 0
R2L Our results BPNSQ BPN	0 14 4	1 479 2	5 147 0	54 6660 0

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

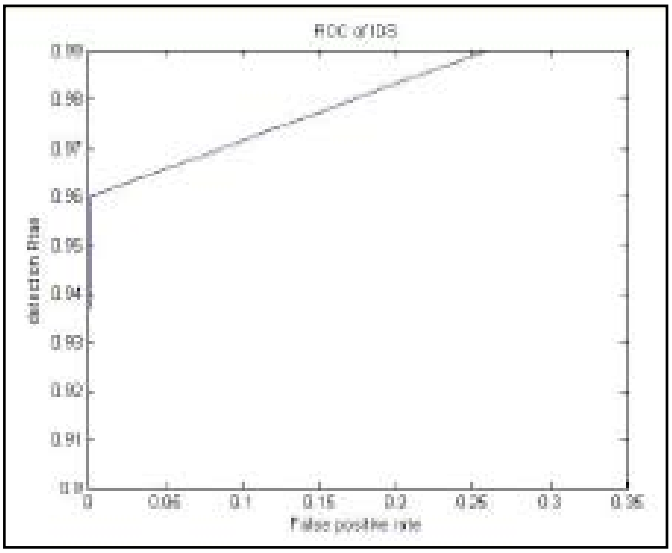

Fig. 5: ROC Curve-Performance Analysis of Intrusion Detection using Naive Bayes

Next, we build a cost matrix as in Table 2. The cost matrix can be used to measure the damage of misclassification [18].

Let Mij denote the number of samples in class misclassified as class j ,Cij 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:

Cost=Σ Mij× Cij/N

The Cost-Performance comparison on the KDD'99 dataset is shown in Table 3.

Table 3: Cost Matrix

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

Table 4: 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 [5] 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.

V. Conclusion

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] Y. Chen, Y. Li, X. Cheng, L. Guo,"Survey and Taxonomy of Feature Selection Algorithms in Intrusion DetectionSystem", Inscrypt 2006.

[2] M. Guennoun, A. Lbekkouri, K. El-Khatib,"Selecting the Best Set of Features for Efficient Intrusion Detection in Wireless Networks", 3rd International IEEE Conference on Information and Communication Technologies: From Theory to Applications, 2008.

[3] M. Guennoun, A. Lbekkouri, K. El-Khatib,"Optimizing the Feature Set of Wireless Intrusion Detection Systems", International Journal of Computer Science and Network Security, Vol. 8, No. 10, October 2008.

[4] H. Liu, H. Motoda,"Feature Selection for Knowledge Discovery and Data Mining", Boston: Kluwer Academic, 1998.

[5] K. M. Faroun, A.Boukelif,"Neural network learning improvement using K-means clustering algorithm to detect network intrusions", April17, 2006.

[6] S.J.Russel, Norvig,"Artificial Intelligence: A modern

approach (International edition)", Pearson US imports & PHIPES, Nov 2002.
[7] P.Domingos, M.J. Pizzani,"On the optimality of the simple Bayesian classifier under zero-one loss", m/c learning, Vol. 29, No. 2-3, pp 103-130, 1997
[8] P.Jenson,"Bayesian networks and decision graphs", Springer, New-york, USA, 2001.
[9] J.Pearl,"Probabilistic reasoning in intelligent system", Networks of plausible inference, Morgan Kaufmann 1997.
[10] M.Mahoney, P. chan,"An analysis of the 1999b DARPA/ Lincoln laboratory evaluation data for network anomaly detection", Proc.of Recent Advances in intrusion detection (RAID)-2003, Pittsburg, USA, Sept. 2003.
[11] MIT Lincoln Laboratory,"DARPA Intrusion detection Evaluation", [Online] Available: http://www.ii.mit.edu.
[12] Charles Elkan,"Results of the KDD'99 classifier learning", SIGKDD Explorating192), pp. 63-64, 2000.
[13] "WEKA: Software machine learning", the University of Waikato, Hamilton, New-Zealand.

D. Srinivasulu Reddy was born in Brahmanakraka, Nellore Dt, Andhra Pradesh, India. He received B.Tech in C.S.E from Bapatla Eng College,ANU University, Guntur, Andhra Pradesh, India. Presently, he is pursuing M.Tech in C.S.E from Dr Samuel George Institute of Engineering and Technology, Markapur, Prakasam Dist, Andhra Pradesh, India. Him Research interest includes Computer Networks.

YV Adisatyanarayana was born in Yerragondapalem, Prakasam Dt, Andhra Pradesh, India. He received B.E in E C E from MADRAS University, Madras, tamilnadu, India. And He completed his M.Tech C S E from JNTU Anatapur, Andhra Pradesh, India. Presently, he is doing PhD in J N T U Kakinada, Andhra Pradesh, India. Currently he is working as Associate professor in CSE department.

P.P.S.Naik was born in Markapur, Prakasam Dt, Andhra Pradesh, India. He received B.Tech in CS E from Dr Samuel George Institute of Engineering and Technology, Markapur, Prakasam Dist,India. And He completed his M.Tech C S E from JNTU kakinada, Andhra Pradesh, India. Presently, he is doing PhD in J N T U Kakinada, Andhra Pradesh, India. Currently he is working as HOD in CSE Department.

B. Subb Reddy was born in Guntur, Guntur Dt, Andhra Pradesh, India. He received B.Tech in CSE from Bapatla Eng College,India And He completed his M.Tech C S E from JNTU kakinada, Andhra Pradesh, India. Presently, he is doing PhD in J N T U Kakinada, Andhra Pradesh, India. Currently he is working as HOD in IT Department.