An Experimental Evaluation of Bayesian Classifiers Applied to Intrusion Detection

Nasir Majeed Mir1*, Sarfraz Khan2, Muheet Ahmed Butt1 and Majid Zaman1

1Department of Computer Sciences, University of Kashmir, Srinagar – 190006, Jammu and Kashmir India; mimasir14@gmail.com, ermuheet@gmail.com, zamanmajid@gmail.com
2Department of MIS, CCBA, Dhofar University, Oman; skhan@du.edu.om

Abstract

Background/Objectives: Security is gaining its importance in today’s highly connected world. In this paper we study the application of Bayesian classifiers to improve Intrusion Detection. Methods/Statistical analysis: We compared three Bayesian classifiers that are/ can be used for Intrusion detection viz., Naïve Bayes, Naïve Bayes Updateable and BayesNet classifiers. These classifiers are tested using a data mining tool called WEKA. The dataset used for the course of our work (to perform comparative/ experimental evaluation) is NSL-KDD dataset. Findings: We performed the experimental evaluation of above mentioned algorithms using NSL KDD dataset. The results proved BayesNet as the better classifier; however, it still requires some improvements. BayesNet had a True Positive Rate of around 95% and False Positives were as low as 4.87% whereas both Naïve Bayes and its updateable version resulted in True Positive Rate of around 80% and False Positive rate of 19.26% which is not good when compared to BayesNet. Similarly, BayesNet had lesser error rates than its counterparts. The evaluation of BayesNet resulted in the Mean Absolute Error of around 5% and Root Mean Squared Error of around 21% while in the case of Naïve Bayes and Naïve Bayes Updateable Mean Absolute Error was around 4times than that of BayesNet and Root Mean Squared Error was twice of the BayesNet’s. Further results and analysis are provided in the sections 7 and 8 respectively. Application/Improvements: The studied classifiers need further improvements e.g., model building time for BayesNet classifier and classification rate for the other two classifiers.

Keywords: Bayes Net, Classification, Data Mining, Flex Bayes, Intrusion Detection, Naïve Bayes, Naïve Bayes Updateable, WEKA

1. Introduction

Intrusion can be defined as a series of actions that attempt to compromise information integrity, information accessibility and resource availability1. There are many classifications of Intrusion Detection Systems based on taxonomy (Network IDS and Host IDS), structure (Centralized and distributed IDS), etc. but the most widely accepted classification is based on approach or technique used2. The technique/approach based classification is generally of two types Anomaly based and Signature Based. In Anomaly based technique, a database called “Profile Database” stores “Normal Behavior” of the user, and whenever the data comes in from the network, it is captured, transformed and checked against the profile database. If it is present then the traffic is termed as normal traffic, otherwise, it is an anomaly and hence an intrusion or attack. Similarly, in Signature based IDS, a database called “signature database” stores signatures of known attacks. Again the same process of capturing, transforming and querying the database takes place but the working is slightly different. In Signature based IDS, if the signature is present in the incoming traffic, then it is an intrusion otherwise the normal data. In both the cases, whenever there is an intrusion, an “Alarm” is raised for the network administrator to act upon. One of the major disadvantages of these techniques is that it does not prove helpful when there is a new profile (legitimate data) in case of Anomaly based IDS and when there is a new attack in case of Signature based IDS i.e., these techniques are not

*Author for correspondence
self-improving. Hence, for the purpose, since we already have large information about profiles and attack signatures, Data Mining is used for extracting useful Information about profiles or signatures.

Data mining is the de-facto process for extraction of useful information from large data sets. The process consists of many subtasks like classification, prediction, data reduction and data exploration. Classification is the most basic form of data analysis within the process of data mining. In Classification, main goal of a classifier (algorithm) is to classify that the specified item belongs to a class. It can be supervised or unsupervised. There are various methods of classification like Decision Tree Induction, Bayesian Classification, Rule Based Classification and Classification by Back propagation. This paper is an attempt to study the efficiency of Bayesian Classifiers in intrusion detection datasets. In the next section we will present the basics of Bayesian classification and also present a brief overview and working of three prominent Bayesian Classifiers viz., Naïve Bayes, Naïve Bayes Updateable and BayesNet Classifiers. In the subsequent sections we will briefly introduce the toolkit (WEKA) and the dataset (NSL KDD test+) used for achieving our objective. Finally in the remaining sections we will discuss our methodology, Experimental setup, results, discussions, and conclusion & future direction of the work.

2. Bayesian Classification

Bayesian classifiers are Probabilistic Statistical classifiers used to predict class membership probabilities and hence are very important for classifying Intrusions. These classifiers are supervised classifers and therefore require training before they are tested for classification of intrusions. The underlying principle of any Bayesian classification model is the Bayes theorem, which has been named after Thomas Bayes. This theorem is used to test a Hypothesis (H), such as whether a given tuple (X) belongs to a specified Class (C) and is given as:

$$P(H|X) = \frac{P(X|H)P(H)}{P(X)}$$

Where,
- \(P(H|X)\) is called a posteriori probability of hypothesis \(H\) conditioned on given tuple \(X\)
- \(P(H)\) is called prior probability of hypothesis \(H\)
- \(P(X|H)\) is called posterior probability of given tuple \(X\) conditioned on hypothesis.

There are many Bayesian classifier models; however, in this paper we will limit our study to three Bayesian classifiers viz., Naïve Bayes, Naïve Bayes Updateable and BayesNet Classifiers. Naïve Bayes and Naïve Bayes updateable algorithms that are studied in this paper are derived from the work carried out by John & Langley whereas BayesNet is based on the work carried out by Remco Bouckaert. These classifiers are discussed as under:

2.1 Naïve Bayes Classifier

This is the simplest of Bayesian classifiers. This classifier works on the assumption of Class Conditional Independence which means the effect of a particular attribute on a given class is independent of the effects of other attributes of the given class. As such, this algorithm is called naïve and since it use Bayes rule to compute the effect of attributes it is called naïve Bayes algorithm. The advantages and applicability of these algorithms have been discussed by several researchers. This algorithm works in the following manner:

2.1.1 Problem

Whether a given tuple belongs to a specified class.

2.1.2 Assumptions

\(D\) is a training set of tuples with each tuple \((x_i, x_j, \ldots, x_n)\), which depict \(n\) measurements \(M_1\) to \(M_n\) which are independent. Also we have \(m\) classes, \(C_1\) through \(C_m\).

2.1.3 Algorithm

This algorithm computes the probability of each \(X\) belonging to Classes \(C_1, C_2, \ldots, C_m\), then using maximum posteriori hypothesis concludes that \(X\) belongs to max probable class which may be mathematically stated as

$$P(C_j|X) > P(C_i|X)$$

Where, \(j = 1, 2 \ldots n\), \(i = j\) and \(C_i\) being class with max probability. Also to compute each \(P(C_j|X)\) we use Bayes rule as stated above. Since, our problem is a maximization problem i.e., we need to find \(P(C_j|X)\) Such that it has the highest probabilities.

- If class probabilities \(C_i\) are not given we assume all the classes are equally likely, otherwise we compute it as \(|C_i,D|/|D|\) where \(|C_i,D|\) is the total no. of training tuples \(D\) that belong to \(C_i\).
Then, as per Bayes rule, We compute $P(X|C_i)$ but as per the assumptions that the variables are independent. Mathematically,

$$P(X|C_i) = \prod_{k=1}^{n} P(x_k | C_i)$$

Where, $x_k$ has the measurement $M_k$ associate with it. Since $M_k$ can have two types of associated measurements, we have two cases:

- **Case 1:** Categorical Measurement In this case each $P(x_k | C_i)$ is the total no. of tuples in class $C_i$ in D having the measurement $M_k$.
- **Case 2:** Continuous-valued Measurement in this case we use the Gaussian distribution measures of standard deviation ($\sigma$) and mean ($\mu$) as under

$$P(x_k | C_i) = g(x_k, \mu_i, \sigma_i)$$

Where,

$$g(x_k, \mu_i, \sigma_i) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x_k-\mu_i)^2}{2\sigma^2}}$$

Finally based on computed $P(X|C_i)$, the tuple $X$ is classified to belong to a particular class $C_i$, such that, $P(X|C_i)P(C_i)$ is the maximum of all the other classes.

### 2.2 Naïve Bayes Updateable Classifier

This classifier is also known as Flexible Bayes or Flex Bayes algorithm. It operates in the same manner as Naïve Bayes classifier, however, in this Classifier, rather than using normal density measures for Continuous-Valued attributes/measurements, Kernel density estimation methods are used. Also, in naïve Bayes classifier Based on the analysis carried out on training data numeric estimator precision is chosen whereas in updateable version of naïve Bayes algorithm, if zero training instances are supplied then 0.1 is used as default numeric estimator precision. Since, this classifier works in the same way as naïve Bayes, the algorithm used for the purpose is same with the difference of calculating “$P(x_k | C_i)$” using kernel density estimation given as under:

$$P(x_k | C_i) = \frac{1}{n} \sum_j g(x_k, \mu_j, \sigma | C_i)$$

Where, $j$ ranges to compute for the training points of $x_k$ in class $C_i$, $\mu_j = x_j$. The above formula is equal to standard kernel density estimation formula as discussed by Remco.

### 2.3 BayesNet Classifier

Both Naïve Bayes and Naïve Bayes updateable operate on an assumption of class conditional independence. However, in practice, there is a possibility of having dependencies among subsets of attributes. BayesNet or Bayesian Belief networks allow for these possibilities of dependences. Although Researchers argue Naïve Bayes as a type of BayesNet \(^1\) i.e., they argue, that, Naïve Bayes classifier is BayesNet without dependencies among attributes, in this study we will treat them as separate entities. Formally, a BayesNet is composed of directed acyclic graph and conditional probability tables for each variable\(^8\). This directed acyclic graph (BayesNet) is a network structure $B$ over $X$, where $X$ is the set of variables $x_1, x_2, ... x_n$ and the network structure $B$ has a set of associated conditional probabilities $B_p$ as shown under:

$$B_p = \{p(x|pa(x)) | x \in X\}$$

Where, $pa(x)$ is the set of parents of $x$ in the network structure $B$. The probability distribution represented by BayesNet is shown below\(^8\),

$$P(X) = \prod_{i=1}^{n} P(x_i | pa(x_i))$$

The algorithm is briefly discussed as under:

#### 2.3.1 Problem

Whether a given tuple belongs to a specified class.

#### 2.3.2 Assumptions

D is a training set of tuples with each tuple ($X$) having $n$ attributes ($x_1, x_2, ... x_n$), with each attributes described by its parents ($pa(x_i)$).

#### 2.3.3 Algorithm

Constructing a BayesNet is a two-step process, first we construct a Bayes Network Structure and then we Learn Conditional Probabilities;

- **Bayes Network Structure:** The various approaches available for learning Bayes Network Structure available in WEKA are Local Score Metrics, Conditional Independence Tests, Global Score Metrics and fixed structure. (Discussing them is beyond the scope of this paper, for further study please see\(^8\))
• **Conditional Probabilities Tables [CPT]:** After constructing BayesNet through any of the models listed above, next step is to construct/estimate the Conditional Probabilities Tables. For this purpose, the available estimators in the WEKA data mining tool are SimpleEstimator, BMAEstimator, MultiNominalEstimator and BayesNetEstimators.

• Finally we use the following inference formula to use Bayesian Network as a classifier,

\[ P(C_j | x) = \prod_{i=1}^{n} P(x_i | pa(x_i)) \]

Where, \( x_i \in X \) and \( (x_i) \) are the parents of \( x_i \).

### 3. WEKA

WEKA is a Data Mining tool and the name “WEKA” is an abbreviation for Waikato Environment for Knowledge Analysis that has been introduced by Waikato University, New Zealand under GNU public License. This data mining tool has the biggest positive of withstanding the notion of time, since its inception in 1997. WEKA is a collection of inter-dependent programs that are bound together by a single user interface. One of the advantages of WEKA is that it is not just a data mining tool but also a Data Visualization and Data Preprocessing tool. Also, it supports multiple dataset formats like csv data files, Json Instance files, libsvm data files, Matlab ASCII files etc., with the default being ARFF Data files making it invariably a strong candidate for Data Mining on heterogeneous types of datasets. This tool does not only allow researchers to use its default programs but it also acts as a framework to modify and develop new programs.

### 4. NSL KDD Dataset

Behind the success of every data mining algorithm there is a dataset, so to choosing a dataset is a necessary step for any comparative analysis of data mining/machine learning algorithms. In our case of Intrusion detection we had the choice to make between multiple datasets like KDD Cup ’99/KDD Cup ’98 but the inherent problems which led us to use NSL KDD dataset as a source dataset for performing the comparative analysis of the classifiers discussed in this paper. This dataset is the work for lessening (if not total elimination) the disadvantages of KDD Cup datasets to some extent. One of the main advantages of NSL KDD dataset is that it includes no/lesser redundancy making our classifier not to be biased. We have used NSL KDD’s Test+ Dataset for the analysis. This dataset contains 22544 instances with each instance spread over 42 attributes. This data set contains records about all four types of intrusions (DOS, Probing, R2L, and U2R) (16). There are basically four types of attributes in this dataset viz., Basic (9 attributes), Content (13 attributes), Traffic (9 attributes) and Host (10 attributes) and the final one being class attribute.

### 5. Methodology

The methodology used for this purpose is quite simple, we took a dataset and put it through different tests for evaluation under the above specified Classifiers using WEKA, then computed the results according to measures like F-Measure, Total Time Taken, Correctly/Incorrectly Classified Instances etc. Finally we put forward our results and conclude with the advantages of one classifier over the other.

### 6. Experimental Setup

During the course of this study we used NSL KDD dataset as a case study dataset to determine the effectiveness of Naïve Bayes, Naïve Bayes Updateable and BayesNet Classifiers on intrusion detection systems. In addition to this we used WEKA as a data mining tool for carrying out the experiments. To check for robustness of the algorithm with the overwhelming data, we did not perform any features selection on the dataset under consideration i.e., we used all the 42 attributes available with NSL KDD dataset. After loading the dataset into WEKA, first step was to preprocess (apply filtering) the dataset and for this we used AllFilter instance of WEKA to keep the dataset unmodified. Then for each algorithm/classifier some of the basic parameters were set like we used Cross validation with 10 folds for measuring performance as a testing option for the classifiers. Also, we used default settings available with WEKA on each classifier and then computed their results.

### 7. Results

One of the most important step of any comparative study are the results of the experiments carried out. Our results were computed on the basis of correctly/Incorrectly
classified instances, Time taken for each classification, Training as well as simulation errors and true positive and false positive rates. The results are summarized and consolidated in the following tables.

The above Table 1 summarizes the results acquired during the course of experiments for different classifiers according to the parameters already set. The results have been consolidated into a single table to facilitate comparative study of the classifiers.

Table 2 shows training/simulation error rates that occurred during the course of experiments. It also shows a Performance measure called F-Measure that is computed on the basis of Precision and Recall as under:

\[ F - \text{Measure} = \frac{2 \cdot (\text{Precision} \cdot \text{Recall})}{\text{Precision} + \text{Recall}} \]

It must also be noted that above statistics are based on the result obtained on NSL KDD’s Test+ dataset containing a total of 22544 instances. In pursuit for improved understanding of the results we have made figures/ charts from the tables (Tables 1 and 2).

Table 1. Results Acquired

<table>
<thead>
<tr>
<th>Classifier</th>
<th>Correctly Classified Instances</th>
<th>Incorrectly Classified instances</th>
<th>Time Taken (Seconds)</th>
<th>True Positive Rate</th>
<th>False Positive Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>BayesNet</td>
<td>21446</td>
<td>1098</td>
<td>1.39</td>
<td>0.95</td>
<td>0.048</td>
</tr>
<tr>
<td>Naïve Bayes</td>
<td>18200</td>
<td>4344</td>
<td>0.48</td>
<td>0.807</td>
<td>0.158</td>
</tr>
<tr>
<td>Naïve Bayes Updateable</td>
<td>18200</td>
<td>4344</td>
<td>0.48</td>
<td>0.807</td>
<td>0.158</td>
</tr>
</tbody>
</table>

Table 2. Training/Simulation Errors and Performance Measure

<table>
<thead>
<tr>
<th>Classifier</th>
<th>Mean Absolute Error</th>
<th>Root Mean Squared Error</th>
<th>Relative Absolute Error (%)</th>
<th>Root Relative Squared Error (%)</th>
<th>F-Measure</th>
</tr>
</thead>
<tbody>
<tr>
<td>BayesNet</td>
<td>0.0505</td>
<td>0.2104</td>
<td>10.2924</td>
<td>42.4921</td>
<td>0.951</td>
</tr>
<tr>
<td>Naïve Bayes</td>
<td>0.1924</td>
<td>0.4371</td>
<td>39.2297</td>
<td>88.2712</td>
<td>0.807</td>
</tr>
<tr>
<td>Naïve Bayes Updateable</td>
<td>0.1924</td>
<td>0.4371</td>
<td>39.2297</td>
<td>88.2712</td>
<td>0.807</td>
</tr>
</tbody>
</table>

8. Discussion

Based on the results obtained in Table 1 and as illustrated in Figure 1 it is evident that BayesNet has better classification rates for successfully classifying the dataset under consideration. BayesNet has a total of 95.13% of success in correctly classifying the instance of intrusions or normal data and only 4.87% of instances have been incorrectly classified instances as intrusions or normal data whereas Naïve Bayes and Naïve Bayes Updateable have classified instances correctly only 80.73% instances and have incorrectly classified instances 19.27 of instances have been incorrectly classified. Table 1 and Figure 2 show the time taken by these classifiers to classify the instances and clearly Naïve Bayes and Naïve Bayes Updateable hold an edge there over BayesNet Classifier. It takes only a split
of a second, precisely 0.48 seconds for Naïve Bayes and Naïve Bayes Updateable classifiers to classify the instances, however for BayesNet it takes around 1.39 seconds i.e., more than twice the time taken by the former classifiers. In addition to this, Table 1 also summarizes the True Positive and False Positive rates of each classifier. This has been illustrated in Figure 3, which shows that BayesNet has better True/ False positive rates than Naïve Bayes and Naïve Bayes Updateable classifiers. It has a True Positive rate of 0.951 and False Positive rate of 0.048 to go with the True Positive rate of 0.807 and False Positive rate of 0.158 for the Naïve Bayes and Naïve Bayes Updateable classifiers. Table 2 summarizes the Training/ Simulation Errors and F- measure (Which is a performance measure) of the classifiers. Specifically, Table 2 and Figure 4 illustrate

![Figure 2. Time Taken(seconds).](image)

![Figure 3. True /False Positive Rates.](image)

![Figure 4. Training /Simulation Errors](image)

![Figure 5. F-measure](image)

Training/ Simulation errors. To allow for generality, the indicators (Mean Absolute and Root Mean Squared Error) used for the purpose of showing errors are commonly used. Also, two more indicators (Relative absolute error and Root Relative squared Error) are shown to further strengthen the argument. And looking at the figure it can be easily observed that BayesNet has lesser error rates than its counterparts. Also, from Table 2 and Figure 5 Bayes Net has better F-measure (0.951) than Naïve Bayes and Naïve Bayes Updateable classifiers (0.807). It must also be noted that Naïve Bayes and Naïve Bayes Updateable classifiers present the same results, this is because normality condition holds in our dataset21.

9. Summary and Conclusion

The main objective of this paper was to see which Bayesian classifier is better amongst BayesNet, Naïve Bayes and Naïve Bayes Updateable classifiers when applied to a dataset for Intrusion Detection Systems. For this purpose, we used NSL KDD’s Test+ Dataset and WEKA as a data mining tool to make a comparative study of the Classifiers. We applied our dataset on each classifier and then summarized our results and based on the results we can clearly state that BayesNet is a better classifier with an accuracy of 95.13% and average error rate being 0.13045, it also has better True/ False positive rate and F-measure.

However, the problem with the BayesNet classifier is the time taken to Build/ execute the model which is twice the Naïve Bayes and Naïve Bayes Updateable classifiers and in intrusion detection system we know that, “it is not only important who gets it right but it is more important that who gets it right in time”. Hence, in future we would work towards improving BayesNet Classifiers so that the model building time can be effectively reduced while not compromising on its performance. Also the effect of feature selection
can be checked on the algorithms. In addition to this, the effect of Naïve Bayes and Naïve Bayes Updateable classifiers can be checked on the datasets where the Independent Assumption holds but normality condition does not.

10. References