Journal of Soft Computing Paradigm (ISSN: 2582-2640)
www.irojournals.com/jscp/

Intrusion Detection for Database Security using a Hidden Naïve Bayes Binary Classifier

M. Deepa[1], J. Dhilipan[2]

[1]Research scholar Department of Computer Applications, SRMIST, Ramapuram Campus, Chennai
[2]Professor & Head, Department of Computer Applications, SRMIST, Ramapuram Campus, Chennai

E-mail: [1]dm8027@srmist.edu.in

Abstract

The Hidden Naive Bayes Binary Classifier is used for Database Security to detect Intruders. Data mining is used a lot in intrusion detection systems to classify normal or anomaly events. This method is a transparent, effective, and widely used mining method based on the idea of conditional attribute independence. HNB classifier is a more advanced version of Naive Bayes classifier algorithm and is efficiently used for intrusion attacks. It keeps the simplicity and efficiency of Naive Bayes, but loosens the independence condition. In the tests, it is proved that this binary classifier model can be used to solve the intrusion detection problem.

Keywords : Data mining, intrusion detection, machine learning, classifier

1. Introduction

Intrusion detection has been a major threat in Database environment. Even though there are many changes in working environments inside and outside the database which provides high level of security for the data, still the intrusion attack continues resulting in loss of data, misinterpretation of data etc.[1]. Classification (usage of ML classifier) is a technique that is used to tell the difference between malicious and normal events. The labels (like "malicious " or "normal") of database event records (instances) that have attributes. Learned classifier models are based on the concept of learning and then building a classifier model based on that learning. The class labels are created for new data instances by the classifier model built. It has all been used to solve the intrusion detection problem. Data mining method "Hidden Naive Bayes (HNB)" is used to create a binary classifier for categorising database events as "normal" or "attack" [3][4]. In data mining, the Naive Bayes (NB) method is a straightforward method to look for patterns in data.

DOI: https://doi.org/10.36548/jscp.2022.2.001
Received: 23.04.2022, received in revised form: 10.05.2022, accepted: 20.05.2022, published: 30.05.2022

The Nave Bayes method is called the HNB method because it is an extended version of the NB method. Use the HNB multiclass classifier model to detect database intrusions[5], and the results are good when compared to traditional database security models [12, 13]. Roopa have proposed an Intrusion Detection and Prevention System for C3I system called SMIML-IDPS (Smart Multi Instance Multi-Label Intrusion Detection and Prevention System). It acts as a decision-making unit in the centralized server for investigating the clients as normal or intruder.

This experimental simulation study shows how this HNB model helps with the intrusion problem on databases. A well-known dataset for intrusion detection called the KDD'99 dataset was used to model and test our claim. This is the data set for intrusion detection. A classifier function "Naïve Bayes Classifier" predicts a different class for each object in a dataset, so it can figure out which one is suitable.

The research methodology section talks about the intrusion detection model and how the research is performed. The Experiments and Results section shows how the experiment is set up and the results obtained. HNB binary classifier results are compared to those from standard NB classifiers in the same part of the text. In the last part of this paper, a quick summary of the research findings and what they mean for the future are summarized.

2. Methodology

2.1 Classifiers by Naive Bayes

It is the most simple type of classifier. These classifiers are simple, which is based on the assumption that attributes aren't linked to each other.

When a Bayesian classifier looks at a dataset called D (E1, E2,..., Et), it looks at the feature set of A (a1, a2,...). It then looks at the class set of C (c1, c2,..., cn) and maps it into the class set of D.

$$P(E \mid c) = P(a1, a2, \ldots \ldots an \backslash c) = |\prod\_(i = 1)\text{^}n \blacksquare n|P(ai|c) \quad (1)$$

$$c(E) = \arg\max \ \llbracket P(c)P(a1, a2, \ldots an|c) \rrbracket \quad (2)$$

Figure 1 shows how the NB and HNB structures work together, as well as how they look. The NB classifier takes into account that the attributes of a class are independent of each other (2).

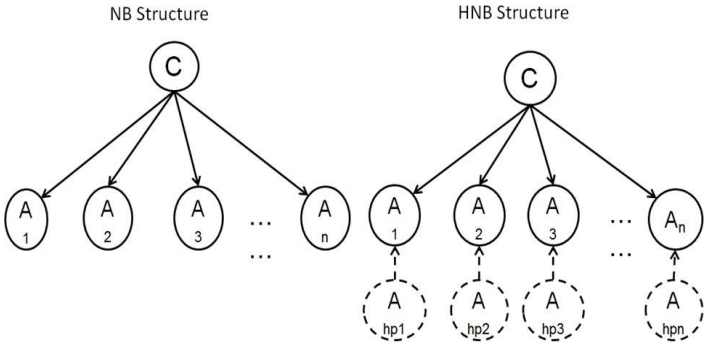

Figure 1. NB and HNB Structures

When attributes are thought to be conditionally independent, NB classifier is identified with P(c) and P(ai|c) and make an NB classifier. After all, the NB classifier is easy to use and quick to use. It also has the same level of accuracy as other popular methods, like classification trees and neural networks [14].

$$c(E) = \arg\max \ [\![P(c) \prod(i = 1)\text{^}n \blacksquare P(ai|c)]\!] \quad (3)$$

This is an even more advanced version of the NB classifier called the HNB classifier. As shown in Figure 1, it is based on the addition of an extra layer to represent a parent for each attribute [10,11] .

The hidden parent (Ahpi) sums the weights of all oattributes (Ai) into one. The joint distribution is referred to as (4)

$$\mathrm{P(A1, \ldots., An \,|\, C) = P(C) \prod n \blacksquare P(Ai|Anpi|C)} \quad (4)$$

The NB classifier does well on datasets where the conditional independence assumption is true.

The HNB Classifier can be expressed as:

$$\mathrm{P(Ai \,|\, Anpi..C) = P(C)(\sum_{i=1}^{j=1} P\ wij * p(Ai|Aj.C)} \quad (5)$$

Table 1. Analysis of KDD'99 Dataset

Class	**No. of rec. in Training data**	**Distribution of training data (%)**	**No. of rec. in test data**	**Distribution of test data (%)**
Normal	97387	19.59	60493	19.78
Attack	386473	80.41	250346	80.22
Total	483860	100.00	310839	100.00

In Audit Data Analysis and Mining, Bayesian classifier is used for intrusion detection. This study is one of the first to use the model (ADAM). ADAM uses pseudo-Bayes estimators to figure out how likely it is that new attacks will happen. These probabilities are used to build an NB classifier that can be used to classify normal and attack events even if you don't know about new attacks.

2.2 HNB Classifiers

The HNB Classifier can be expressed as:

$$c(E) = \arg\max P(c) \prod_{i=1}^{n} P(ai|anpc) \quad (6)$$

$$P(ai|anpi..,c) = P(.c)\sum_{i-1,j<>i}^{n} Wij * P(ai|aj,c) \quad (7)$$

In the last two decades, a lot of research has been done on how to make NB models less dependent on each other.

Weights Wij can be calculated in a number of ways. One way to get the weights Wij is to use the conditional Mutual Information (CMI).The attributes Ai and Aj are used to get the weight of P(Aj|Ai, C) as given in equation 8.

$$Wij = \frac{Ip\ (Aij,Ai|C)}{\sum_{j=1,j<>i}^{n} Lp(Ai,Aj|C} \quad (8)$$

This is how Wij can be calculated. Here, the CMI is Ip(Ai,Aj|C).

$$lp(Ai,Aj|C) = \sum_{ainic} P(ai, aj, c) \log \frac{P(ai,aj|c)}{P(ai|c)P(aj|c)} \quad (9)$$

Class	No. of rec. in Training data	Distribution of training data (%)	No. of rec. in test data	Distribution of test data (%)
Normal	97387	19.59	60493	19.78
Attack	386473	80.41	250346	80.22
Total	483860	100.00	310839	100.00

TCP dumps, which are records of network traffic connections, are in the dataset. It also has training and test data. A part of KDD'99 dataset is used which has the tagged records.

As a point of difference, the datasets for training and testing are different when it comes to number of records and class distributions.

A recent study used an HNB-based multiclass classifier to detect intrusions, and the results were good because the HNB multiclass classifier has been shown to be very good at classifying things [10,12], this paper looks into using the HNB model binary classifier to solve the anomalies. The results are compared to the results of the normal NB classifier, and the simulation results are displayed.

Data from the DARPA offline intrusion detection dataset from 1998 was used in this study. A sample from the MIT Lincoln Laboratory that is a part of this dataset is used. Despite its flaws, the KDD'99 dataset is known as a classic. This dataset was implemented as it is the complete public dataset that can be found and is still used to test and evaluate intrusion detection systems. Normal connection events are shown in 20% of the records. Each connection record has 34 continuous features, 7 discontinuous features, and a total of 41 features. A discretization method must be used to turn continuous attributes into discrete counterparts. This is because the model is made up of discrete attributes.

The discretization is supposed to assist classifier models perform better on huge datasets , such as the KDD'99 dataset the EMD approach is better for the NB classifier method because of how well it worked on the tested dataset.

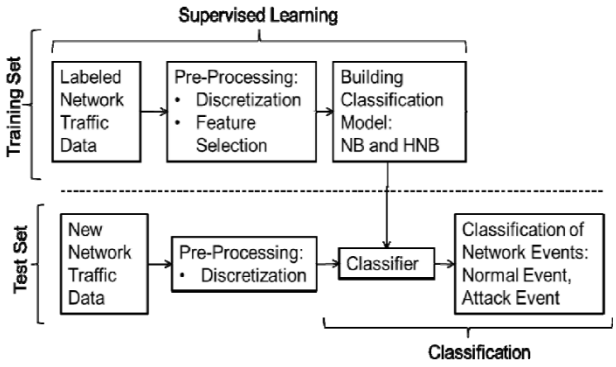

Figure 2. Research framework

Because of a dataset's large feature set, huge data dimensionality and feature interdependence, any data mining model faces significant challenges. Making sure to pick the appropriate features and cut down the irrelevant features is important for speedy processing and accurate predictions. This is the same thing that happens in most of the data mining techniques. As shown in Figure 2 of our research framework, a common approach to challenges is the selection of feature. The feature selection filter utilises consistency-based (CONS), and it works well with the KDD'99 dataset when used with this method.

The CONS method uses an inconsistency criterion to figure out dimensionally reduced data. In each round, it picks a random group, and the one that is the most consistent is saved. In this study, accuracy and error rate are calculated. The receiver operating characteristics (ROC) curve is also determined to measure the efficiency of the model. These techniques are widely used to summarise and compare performance of various classifiers. Accuracy is the percentage of things that are appropriately categorised. People make mistakes when they categorise things in a dataset. This is called the error rate.

Table II shows the results. It is easy to see how well a binary classification method works by looking at a "confusion matrix." The normal and attack columns are for the class that is supposed to be there. The normal and attack rows are for the class that is there. The corresponding counts is obtained from the intersection of rows and columns which can be used to get a quick overview of the algorithm's performance. The false positive (FP) count of the model generates false alarms, whereas the true positive (TP) count generates attacks that are accurately recognised.

Table 2. Confusion Matrix - Binary Classifier

Confusion matrix		**Predicted class**	
		Normal	**Attack**
Actual class	**Normal**	True Negative (TN)	False Positive (FP)
	Attack	False Negative (FN)	True Positive (TP)

Accuracy is defined as:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \quad (10)$$

Error rate is defined as:

$$\text{Error Rate} = 1 - \text{Accuracy} \quad (11)$$

This method is based on computing the AUC [25]. If AUC value is close to one, it's widely thought that it will be more accurate.

3. Experiments and Results

In this section, four experiments and the results obtained are summarized. The simulation studies are carried out with the KNIME tool, which is a Java-based open source machine learning software [26] and was based on the research framework shown in Figure 2. First, a supervised discretize () method is used to make the continuous variables that make up 10% of the KDD'99 dataset into separate parts.

A pre-processed dataset is used to develop the classifier models for NB and HNB after discretization and feature selection. As part of the supervised learning phase, the 10-fold cross-validation approach is used on the training dataset to generate the models. The cross-validation method is a common one that works well when there are a lot of different things to look at. The method is based on fact of randomly dividing a dataset into ten disjoint sections of roughly equal size. The subsets is utilised as the test data in each of the ten runs, and the other nine datasets are used as training sets to develop a model. In order to figure out how accurate the classifier is, you take the average of the estimates for each run.

Table 3. Confusion Matrix for NB Binary Classifier

Confusion matrix		**Predicted class**	
		Normal	**Attack**
Actual class	**Normal**	58967	775
	Attack	26273	224119

Table 4. Confusion Matrix for HNB Binary Classifier

Confusion matrix		**Predicted class**	
		Normal	**Attack**
Actual class	**Normal**	58967	775
	Attack	19453	2230963

The classifier models that are made from the results of the FP are then used. It's most commonly used in cost-benefit analysis. An example of how the method is used: In Tables III and IV, the confusion matrices summarise the performance with correct predicted or incorrectly predicted instance counts. As shown in these two confusion matrix tables, the

number of attacks correctly classified by the HNB model is significantly greater as compared to the number of correctly classified by the NB model.

Table 5. Test Results for the Classifier Performance

Model	Accuracy	Error rate	AUC*
NB	0.9219	0.0681	0.9400
HNB	0.9340	0.0660	0.9790

Table V summarises the experiment's outcomes for accuracy, error rate, and AUC using the test dataset (Area under the ROC curve). The accuracy of the HNB model is 0.9430, the error rate is 0.0670, and the AUC value is 0.989, as shown in these data. According to the results, the accuracy of the classic NB model is 0.9219, the error rate is 0.0681, and the AUC value is 0.9300.

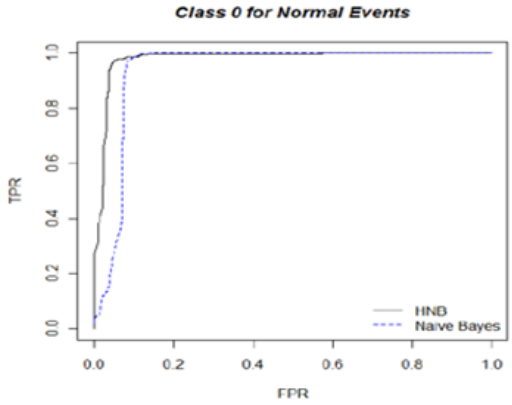

Figure 3. ROC graph for detection of normal events

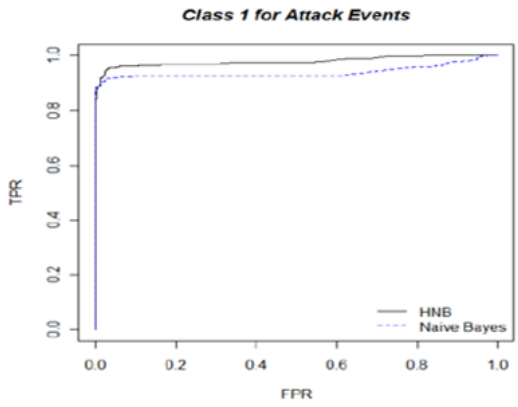

Figure 4. ROC graph for attack of detection events

People who use the HNB binary classifier-based model always have a better AUC than people who use the NB binary classifier-based model, which is shown in the figures 3 and 4. The HNB model outperforms the NB-based model on all aspects utilising various datasets [10] and the KDD'99 dataset [12].

4. Conclusion

In this work, the growing necessity to classify network attack events using data mining methodologies has been highlighted. The Naive Bayes classifier model is simple and widely used mining method based on the assumption of attribute independence. A classifier model with Hidden Naive Bayes (HNB) approach has been provided to remove the naïve assumption while preserving its simplicity and efficiency. The well-known KDD'99 intrusion detection dataset has been used to see how well the novel classifier algorithm works on the hard database intrusion detection problem. It has been found that, the HNB binary classification model is better than the classic NB model when implemented with CONS filtering and EMD discretization and Feature selection methods.

References

[1] P. Kabiri and A. A. Ghorbani, "Research on intrusion detection and response: A survey," International Journal of Network Security, vol. 1, pp. 84-102, 2005.

[2] T. F. Lunt, "Real-time intrusion detection," in COMPCON Spring '89. Thirty-Fourth IEEE Computer Society International Conference: Intellectual Leverage, Digest of Papers., 1989, pp. 348-353.

[3] Dr.Roopa M ,Dr.Selvakumarraja “Secured Intrusion Detection System using Tristate Algorithm”, International Journal of Applied Engineering Research, ISSN 0973-4562 Vol. 10 No.20 (2015), 0973-4562.

[4] Dr.Roopa M ,Dr.Selvakumarraja “An intelligent network algorithm for enhanced security in a mobile ad hoc network”, International Journal of Networking and Virtual Organisations, Vol. 17, Nos. 2/3, 2017,1470-9503.

[5] Dr.Roopa M ,Dr.Selvakumarraja “Implementation of MIML Technique in Intrusion and Detection for Network layer attacks in MANET”, International journal of Research in Dynamic research and control systems, Vol.10,No.6, 895-901,2018.

[6] J. Cannady, "The application of artificial neural networks to misuse detection: initial results," in Proceedings of the Recent Advances in Intrusion Detection ’98 Conference, Louvain-la-Neuve, Belgium, 1998, pp. 31-47.

[7] R. P. Lippmann, Cunningham, R. K., "Improving intrusion detection performance using keyword selection and neural networks," Computer Networks, vol. 34, pp. 597-603, 2000.

[8] M. Panda and M. R. Patra, "Network intrusion detection using naive Bayes," International Journal of Computer Science and Network Security, vol. 7, pp. 258-263, 2007.

[9] B. Gao, HY. Ma, and YH. Yang "HMMs (Hidden Markov Models) based on anomaly intrusion detection method," in International Conference on Machine Learning and Cybernetics, 2002, 2002, pp. 381-385.

[10] L. Jiang, Z. Harry, and C. Zhihua, "A Novel Bayes Model: Hidden Naive Bayes," Knowledge and Data Engineering, IEEE Transactionson, vol. 21, pp. 1361-1371, 2009.

[11] J. Yaguang, Songnian, Y., Yafeng, Z., "A novel Naive Bayes model: Packaged Hidden Naive Bayes," in Information Technology and Artificial Intelligence Conference (ITAIC), 2011 6th IEEE Joint International, 2011, pp. 484-487.

[12] L. Koc, T. A. Mazzuchi, and S. Sarkani, "A network intrusion detection system based on a Hidden Naïve Bayes multiclass classifier," Expert Systems with Applications, vol. 39, pp. 13492- 13500, 2012.

[13] KDD-Cup. (1999, July 29, 2011). KDD Cup 1999 Data. Available: http://kdd.ics.uci.edu/databases/kddcup99/kddc up99.html

[14] P. Langley, W. Iba, and K. Thompson, "An analysis of Bayesian classifiers," in Proceedings of the tenth national conference on artificial intelligence, San Jose, California, 1992, pp. 223- 228.