Indian Journal of Science and Technology, Vol 9(12), DOI: 10.17485/ijst/2016/v9i12/86291, April 2016
ISSN (Print) : 0974-6846
ISSN (Online) : 0974-5645

An Experimental Evaluation of Bayesian Classifiers Applied to Intrusion Detection

Nasir Majeed Mir[1*], Sarfraz Khan[2], Muheet Ahmed Butt[1] and Majid Zaman[1]

[1]Department of Computer Sciences, University of Kashmir, Srinagar – 190006, Jammu and Kashmir India; mirnasir14@gmail.com, ermuheet@gmail.com, zamanmajid@gmail.com
[2]Department 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 it 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 availability[1]. 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 used[2]. 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 their 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[3].

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[4]. 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[5] and hence are very important for classifying Intrusions. These classifiers are supervised classifiers 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) is called prior probability of hypothesis X, and

P(X|H) is called posterior probability of given tuple X conditioned on hypothesis[6].

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[7] whereas BayesNet is based on the work carried out by Remco Bouckaert[8]. 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[9, 10]. 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) having n attributes (x_1,x_2,…,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 …, C_m, then using maximum posteriori hypothesis concludes that X belongs to max probable class which may be mathematically stated as

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

Where, j= 1, 2…n, i! = j and C_i being class with max probability. Also to compute each $P(C_i|X)$ we use Bayes rule as stated above. Since, our problem is a maximization problem i.e., we need to find $P(C_i|X)$ 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[7]. 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 (σ) and mean (μ)[7] as under

$$P(x_k | C_i) = g(x_k, \mu C_i, \sigma C_i)$$

Where,

$$g(x_k, \mu C_i, \sigma C_i) = \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

- 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:

$$P(xk|Ci) = \frac{1}{n} \sum_{j} g(xk, \mu j, \sigma Ci)$$

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[7].

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[11] 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 $x_1, x_2 \ldots x_n$ 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 $(x_1, x_2, \ldots, x_n)$, 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, MultiNomnalEstimator and BayesNetEstimators[8].
- Finally we use the following inference formula to use Bayesian Network as a classifier,

$$P(C_i \mid x) = \prod_{i=1}^{n} P(x_i \mid pa(x_i))$$

Where, $x_i \in X$ and (x_i) are the parents of x_i[8].

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[12]. 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[13]. 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[14]. 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[15,16].

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[17] 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[18]. One of the main advantages of NSL KDD dataset is that it includes no/lesser redundancy making our classifier not to be biased[19]. 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[20].

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:

$$F-Measure=\frac{2(Precision*Recall)}{(Precision+Recall)}$$

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).

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

Figure 1: Correctly/ Incorrectly Classified Instances

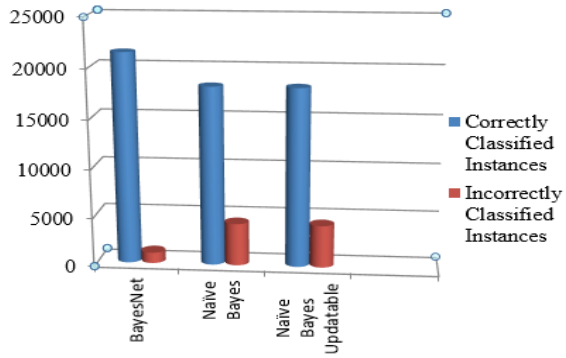

Figure 1. Correctly/Incorrectly Classified Instances

Table 1. *Results Acquired*

Classifier	Correctly Classified Instances	Incorrectly Classified instances	Time Taken (Seconds)	True Positive Rate	False Positive Rate
BayesNet	21446	1098	1.39	0.95	0.048
Naïve Bayes	18200	4344	0.48	0.807	0.158
Naïve Bayes Updateable	18200	4344	0.48	0.807	0.158

Table 2. Training/Simulation Errors and Performance Measure

Classifier	Mean Absolute Error	Root Mean Squared Error	Relative Absolute Error (%)	Root Relative Squared Error (%)	F-Measure
BayesNet	0.0505	0.2104	10.2924	42.4921	0.951
Naïve Bayes	0.1924	0.4371	39.2297	88.2712	0.807
Naïve Bayes Updateable	0.1924	0.4371	39.2297	88.2712	0.807

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

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

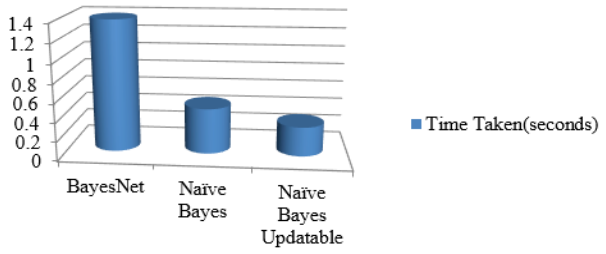

Figure 2. Time Taken(seconds).

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

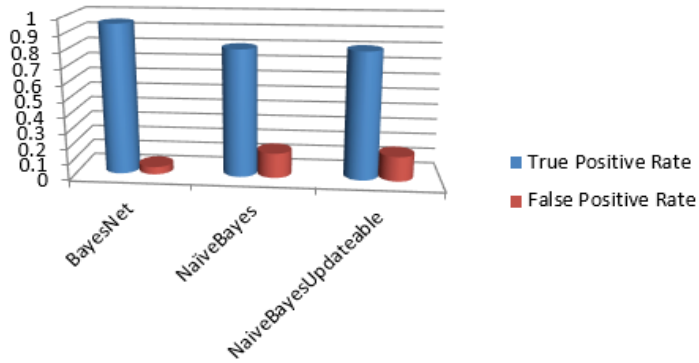

Figure 3. True /False Positive Rates.

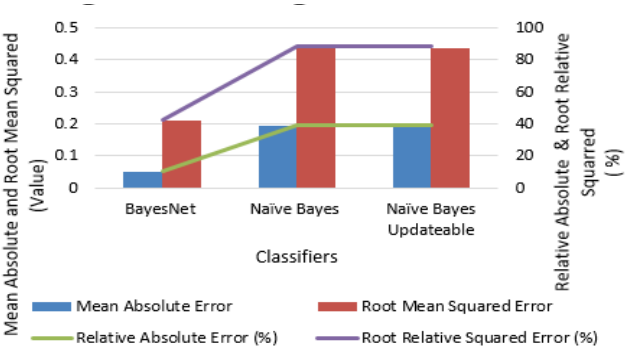

Figure 4. Training /Simulation Errors

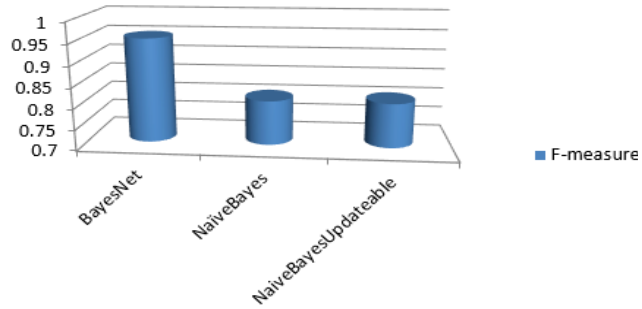

Figure 5. F-measure

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 dataset[21].

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

1. Heady R, Luger GF, Maccabe A, Servilla M. The architecture of a network level intrusion detection system. Department of Computer Science, College of Engineering. University of New Mexico; 1990.
2. Azad C, Jha VK. Data mining in intrusion detection: a comparative study of methods, types and data sets. International Journal of Information Technology and Computer Science (IJITCS). 2013; 5(8): 75.
3. Lee W, Stolfo SJ, Mok KW. A data mining framework for building intrusion detection models. Security and Privacy. Proceedings of the IEEE Symposium on; Oakland: CA ; IEEE; 1999. p. 120–32.
4. Kamber M, Winstone L, Gong W, Cheng S, Han J. Generalization and decision tree induction: efficient classification in data mining. Research Issues in Data Engineering. 1997 Proceedings Seventh International Workshop on; IEEE;1997.
5. Zadrozny B, Elkan C. Obtaining calibrated probability estimates from decision trees and naive Bayesian classifiers. ICML; Citeseer. 2001. p. 1–8.
6. Heckerman D. Bayesian networks for data mining. Data mining and knowledge discovery. 1997; 1(1): 79–119.
7. John GH, Langley P. Estimating continuous distributions in Bayesian classifiers. Proceedings of the Eleventh conference on Uncertainty in artificial intelligence; Morgan Kaufmann Publishers Inc; 1995. p. 338–45.
8. Bouckaert RR. Bayesian network classifiers in weka. Department of Computer Science. University of Waikato; 2004.
9. Chandrashekar Azad, Vijay Kumar Jha. Data Mining Based Hybrid Intrusion Detection System. Indian Journal of Science and Technology. 2014 Jan; 7(6). Doi no: 10.17485/ijst/2014/v7i6/37551
10. Sheela Evangelin Prasad SN, Srinath MV, Murtaza Saadique Basha. Intrusion Detection Systems. Tools and Techniques—An Overview. Indian Journal of Science and Technology. 2015 Dec; 8(35). Doi no:10.17485/ijst/2015/v8i35/80108
11. Amor NB, Benferhat S, Elouedi Z. Naive bayes vs decision trees in intrusion detection systems. Proceedings of the 2004 ACM symposium on Applied computing; ACM; 2004. p. 420–24.
12. Kumar Y, Sahoo G. Analysis of Bayes, neural network and tree classifier of classification technique in data mining using WEKA. 2012.
13. Garner SR. Weka: The waikato environment for knowledge analysis. Proceedings of the New Zealand computer science research students conference; Citeseer.1995. p. 1–8.
14. Holmes G, Donkin A, Witten IH. Weka: A machine learning workbench. Intelligent Information Systems. Proceedings of the 1994 Second Australian and New Zealand Conference on; IEEE; 1994 Nov-Dec 29-2. p. 357–61.
15. Sharma TC, Jain M. WEKA approach for comparative study of classification algorithm. International Journal of Advanced Research in Computer and Communication Engineering. 2013; 2(4):1925–31.
16. Witten IH, Frank E, Trigg LE, Hall MA, Holmes G, Cunningham SJ. Weka: Practical machine learning tools and techniques with Java implementations.1999.
17. McHugh J. Testing intrusion detection systems: a critique of the 1998 and 1999 DARPA intrusion detection system evaluations as performed by Lincoln Laboratory. ACM transactions on Information and system Security. 2000; 3(4):262–94.
18. Tavallaee M, Bagheri E, Lu W, Ghorbani A-A. A detailed analysis of the KDD CUP 99 data set. Proceedings of the Second IEEE Symposium on Computational Intelligence for Security and Defence Applications; 2009.
19. Revathi S, Malathi A. A Detailed Analysis on NSL-KDD Dataset Using Various Machine Learning Techniques for Intrusion Detection. International Journal of Engineering Research and Technology. ESRSA Publications; 2013.
20. Aggarwal P, Sharma SK. Analysis of KDD Dataset Attributes-Class wise for Intrusion Detection. Procedia Computer Science. 2015; 57:842–51.
21. Langley P, Iba W, Thompson K. An analysis of Bayesian classifiers. AAAI; 1992.