Intrusion Detection System using Naive Bayes algorithm

Sharmila B S
Dept of ECE
The National Institute of Engineering, Mysuru
sharmilabs@nie.ac.in

Dr. Rohini Nagapadma
Dept of ECE
The National Institute of Engineering, Mysuru
rohini_nagapadma@nie.ac.in

***Abstract*—The growth of internet usage increased the need of security in network which is monitored by Intrusion Detection System (IDS). Using machine learning algorithms is common for implementing any IDS to detect network traffic weather it is normal or attack. Naive Bayes algorithm is one of the popular supervised classification algorithm for categorical dataset which is built on conditional independence of feature assumption. Our experimental research focused on comparison of traditional Naive Bayes algorithm and PCA based implementation using with(sklearn) and without built in python library. Experimental results using PCA based NSL-KDD intrusion detection system indicate better accuracy compared to traditional Naive Bayes in both with and without built in sklearn python library.**

I. Introduction

Intrusion detection system is one of the important software for monitoring network[1] , which is basically identify malicious activity and alarm if any event is detected. To decrease false alarm many machine learning algorithms are applied to explore and analyze large dataset [2]. The aim of this research paper is to implement Naive Bayes supervised classification for NSL KDD dataset using Principal Component Analysis (PCA) to categorize any event is normal or attack. Classification basically identifies the category labels based on features or attributes but attributes for KDD CUP 99 is huge so PCA a feature reduction technique is used to reduce the number of attributes. The KDD CUP 99 [3] dataset is a huge dataset captured using tcpdump packet analyser where simulation was conducted around seven weeks. NSL KDD dataset is cleaned compered to KDD CUP 99. NSL KDD has np duplicate and redundant records[4].

II. Existing Work

Many machine learning algorithms are proposed by different authors for IDS, which focused mainly on accuracy. According to [5], Jajodia et al. implemented Naive Bayes technique used for Audit data analysis and mining results where has 77% of accuracy. Koc et al. modified as Hidden Naive Bayes Binary Classifier [6] which is able to detect traffic is attack or normal. The algorithm was implemented based on entropy minimization discretization which uses minimum entropy heuristic for discretize continuous attributes. [7] Hamed et al. implemented two layer dimension reduction and two-tier classification model which used both PCA and LDA feature extraction methods which results in 76.56% of accuracy. Xiaoyan et al. [8] used Principal Component Analysis to reduce data dimension and improve efficiency but with change dataset the data points might even appear to be outlier which are known as swamping effect and also cannot detect all the actual real deviating observations called as Masking which may results in false alarm [9]. [10] proposed J48 classifier using filter method for attribute selection results in 78% of accuracy with respect to 28 unknown attacks. Anish Halimaa and et al. performed comparative analysis for SVM and Naive Bayes which shows the accuracy of 93.85% and 71% respectively [11]. [12] [13] Proposed intrusion detection framework to enhance the performance using PCA. Here PCA with 17 components gives best result for many machine learning algorithms.

III. Block diagram of the system

Our proposed scheme for Intrusion Detection System broadly divided into five blocks to predict whether a set of network traffic is normal or attack is shown in Figure 1. In our research work KDD99 dataset is parsed and normalized then supervised Naive Bayes classification is applied with(sklearn) and without built in python library. Finally accuracy is compared in both cases.

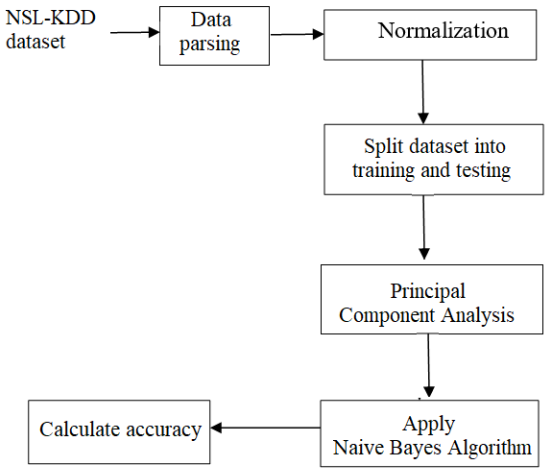

Fig. 1. Block Diagram

978-1-7281-4499-3/19/$31.00 ©2019 IEEE

A. Data Preparation and Normalization

The Dataset selected for our research is 20 percent of NSL-KDD Master Dataset which is improved version of KDDCUP 99 with the advantage of no redundant records and no duplicate records which increases the performance in terms of analysis. Firstly, the selected dataset is raw traffic need to be processed before applying any machine learning algorithms, so below steps are followed for data preparation:

1) The raw dataset 41 column names are appended.
2) The set of 21 attacks are grouped into 4 major attacks shown in below Table I.

TABLE I
ATTACKS IN KDD-DATASET

Attack Group	Attack Types
DOS	Back, Land, Neptune, Pod, Smurf, Teardrop, Mailbomb, Processtable, Udpstorm, Apache2, Worm.
Probe	Satan, IPsweep, Nmap, Portsweep, Mscan, Saint.
R2L	Guess_password, Ftp_write, Imap,Phf, Multihop, Warezmaster, Xlock, Xsnoop, Snmpguess, Snmpgetattack, Httptunnel, Sendmail, Named.
U2R	Buffer_overflow, Loadmodule, Rootkit, Perl, Sqlattack, Xterm, Ps.

3) Dataset we selected for our research has raw data where all 41 attributes have different scales, So complete dataset need to be normalized that is all attributes must reduce to common scale[14]. The normalization is performed by equation[1] below:

$$x_{new} = \frac{x - x_{min}}{x_{max} - x_{min}} \tag{1}$$

4) Once Normalization is completed dataset need to check any missing values in all attributes using library missingo, where features like "is hot login" and "num outbound cmds" were completely zero, so in later part these two features are completed removed from dataset. The below Figure 2 shows the output of finding missing values. Then dataset is divided into 70% train and 30% test dataset.

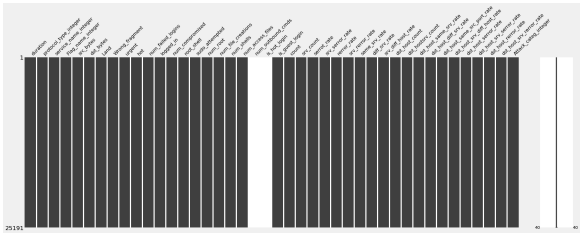

Fig. 2. Missing attributes

5) Finally attribute reduction method PCA is applied for KDD dataset.

B. Principal Component Analysis(PCA)

Principal Component Analysis is a unsupervized technique for dimensionality reduction. It transforms high dimensional vector set to low dimensional vector set which helps in reduction of execution time and improve accuracy. The steps for implementation of PCA is shown below:

1) Consider Normalized dataset
2) Compute mean for all attributes.
3) Compute covariance matrix using inbuilt python function.
4) Decompose covariance matrix as eigen values and eigen vectors.
5) Project data into subspace using below equation where X is original dataset and Y is eigen vectors:

$$P(x) = Y^T * X \tag{2}$$

C. Naive Bayes classification:

Naive Bayes algorithm is a supervised classification algorithm [15] which works on the basis of Bayes theorem having assumption of independence among class values of all attributes. There are three types of Naive Bayes Algorithm as shown below:

1) Gaussian: It is a supervised algorithm suitable for categorical dataset having normal distribution. The probability calculation will be done using below equation:

$$P(a \mid x) = \frac{P(x \mid a)P(a)}{P(x)} \tag{3}$$

where,

a) $P(a \mid x)$ is the posterior probability
b) P(a) is the prior probability of attack.
c) $P(x \mid a)$ is the likelihood which is the probability of predictor given class.
d) P(x) is the prior probability of predictor.

2) Multinomial Naive Bayes: It is a supervised algorithm used for continuous dataset which takes discrete count. For example, calculating number of occurrence of words in text.
3) Bernoulli Naive Bayes: It is binomial model suitable for both continuous and categorical dataset but feature vectors must be binary that is zeros and ones.

So our dataset is categorical model having more then two groups of attacks. So Gaussian Naive Bayes algorithm is selected for our work. The steps followed for calculating probability is shown below:

1) Calculate mean and variance for all attributes.
2) Measure likelihood using below equation[3] below:

$$P(x \mid c) = P(x1 \mid c) * P(x2 \mid c) * .... \tag{4}$$

a) For calculating each term in above likelihood equation it was assumed to be Gaussian, so conditional probability is given by equation below:

$$P(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-(x-\mu)^2/2\sigma^2} \tag{5}$$

3) Calculate prior probability for class P(c) regard-less of data and also prior probability of data p(x) regardless of class.
4) Apply all the calculation in equation [2] to find out posterior probability.
5) Finally class prediction is based on which class has nearest predicted value.

D. Evaluation

The Naive Bayes algorithm is performed in both with and without sklearn library. Figure 3 below shows the accuracy in both cases. Compared to sklearn library the accuracy is more when we implement actual algorithm. To reduce the attributes of dataset and execution time PCA is applied for KDD dataset. The Figure 4 shows the accuracy and also execution time.

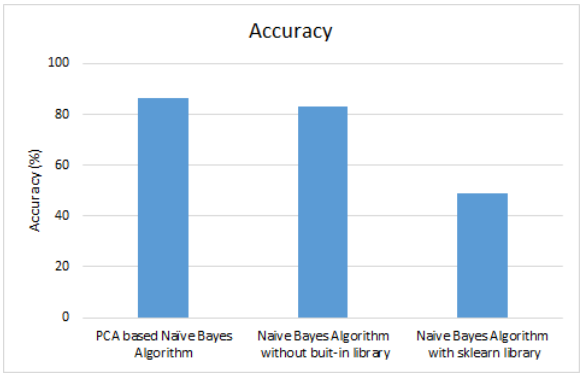

Fig. 3. Accuracy

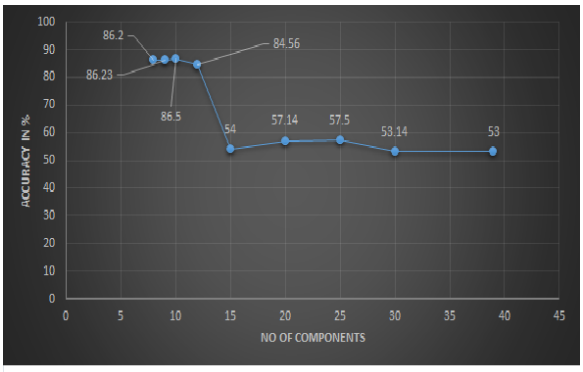

Fig. 4. Naive Bayes with PCA

Most of the machine algorithm performance is analyzed using confusion matrix. The confusion matrix not only provide the details of error prediction but also gives insight type of error being made in terms of sensitivity, specificity and accuracy manually [16] . The below Figure 5 shows the confusion matrix of Naive Bayes algorithm, where the attack were grouped to integers, namely 0 for normal, 1 for DOS, 2 for Probe, 3 for R2L and 4 for U2R

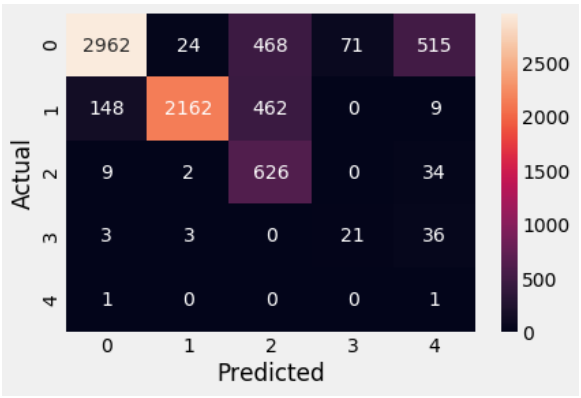

Fig. 5. Confusion matrix

IV. Conclusion

In this paper, traditional and PCA based Naive Bayes algorithm was implemented using python language with and without sklearn library for comparison of accuracy. The Naive Bayesian classifier has advantage of handling missing values by simply omitting the probabilities when performing likelihoods of members in each category. The results showed that compared to traditional Naive Bayes algorithm PCA based algorithm performs better accuracy for 10 principal components and also decreases the execution time. But with increase in number of components the accuracy will decreases.

References

[1] S. Rastegari, P. Hingston, and C.-P. Lam, "Evolving statistical rulesets for network intrusion detection," *Applied soft computing*, vol. 33, pp. 348–359, 2015.

[2] A. Dastanpour, S. Ibrahim, R. Mashinchi, and A. Selamat, "Comparison of genetic algorithm optimization on artificial neural network and support vector machine in intrusion detection system," in *2014 IEEE Conference on Open Systems (ICOS)*, IEEE, 2014, pp. 72–77.

[3] P. Aggarwal and S. K. Sharma, "Analysis of kdd dataset attributes class wise for intrusion detection," *Procedia Computer Science*, vol. 57, pp. 842–851, 2015.

[4] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, "A detailed analysis of the kdd cup 99 data set," in *2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications*, IEEE, 2009, pp. 1–6.

[5] D. B. J. C. S. Jajodia and L. P. N. Wu, "Adam: Detecting intrusions by data mining," in *Workshop on Information Assurance and Security*, vol. 1, 2001, p. 1100.

[6] L. Koc and A. D. Carswell, "Network intrusion detection using a hidden naive bayes binary classifier," *International Journal of Simulation Systems, Science & Technology*, vol. 16, p. 3, 2015.

[7] H. H. Pajouh, R. Javidan, R. Khayami, D. Ali, and K.-K. R. Choo, "A two-layer dimension reduction and two-tier classification model for anomaly-based intrusion detection in iot backbone networks," *IEEE Transactions on Emerging Topics in Computing*, 2016.

[8] L. Koc and A. D. Carswell, "Network intrusion detection using a hidden naive bayes binary classifier," *International Journal of Simulation–Systems, Science & Technology*, vol. 16, p. 3, 2015.

[9] X. Han, L. Xu, M. Ren, and W. Gu, "A naive bayesian network intrusion detection algorithm based on principal component analysis," in *2015 7th International Conference on Information Technology in Medicine and Education (ITME)*, IEEE, 2015, pp. 325–328.

[10] A. R. Onik, N. F. Haq, L. Alam, and T. I. Mamun, "An analytical comparison on filter feature extraction method in data mining using j48 classifier," *International Journal of Computer Applications*, vol. 124, p. 13, 2015.

[11] A. Abubakar and B. Pranggono, "Machine learning based intrusion detection system for software defined networks," in *2017 Seventh International Conference on Emerging Security Technologies (EST)*, IEEE, 2017, pp. 138–143.

[12] B. Subba, S. Biswas, and S. Karmakar, "Enhancing performance of anomaly based intrusion detection systems through dimensionality reduction using principal component analysis," in *2016 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS)*, IEEE, 2016, pp. 1–6.

[13] N. Badr and N. A. Noureldien, "Examining outlier detection performance for principal components analysis method and its robustification methods," *International Journal of Advances in Engineering & Technology*, vol. 6, no. 2, p. 573, 2013.

[14] I. H. Witten, E. Frank, M. A. Hall, and C. J. Pal, *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, 2016.

[15] C. M. Rahman, D. M. Farid, and M. Z. Rahman, "Adaptive intrusion detection based on boosting and naive bayesian classifier," 2011.

[16] J. A. Sidey-Gibbons and C. J. SideyGibbons, "Machine learning in medicine: A practical introduction," *BMC medical research methodology*, vol. 19, no. 1, p. 64, 2019.