Performance Analysis of Network Intrusion Detection Schemes using Apache Spark

Manish Kulariya, PriyankaSaraf, Raushan Ranjan, Govind P. Gupta

***Abstract*—Fast and efficient network intrusion detection is a very challenging issue as the size of network traffic has become increasingly big and complex. A real time intrusion detection system should be able to process large size of network traffic data as quickly as possible in order to prevent intrusion in the communication system as early as possible. In this paper, we have employed five machine learning algorithms such as Logistic regression, Support vector machines, Random forest, Gradient Boosted Decision trees & Naive Bayes for detecting the attack traffic. For processing and detecting the attack traffic as fast as possible, we have used Apache Spark, a big data processing tool for detecting and analysis of intrusion in the communication network traffic. Performance comparison of intrusion detection schemes are evaluated in terms of training time, prediction time, accuracy, sensitivity and specificity on a real time KDD'99 data set.**

***Index Terms*—Intrusion Detection, Network Traffic Analysis, Classification, Apache Spark**

I. INTRODUCTION

As the use of computers and internet is increasing rapidly, so the security concerns are also increasing with the time. Various intrusion detection systems are getting developed and deployed for detecting the attack or malicious traffic [1]. An intrusion detection system (IDS) is a system which is used to monitors network or system activities for detecting malicious activities or policy violations and produces alerts messages to the controlling station [2]. There are various other security methods like information encryption, access control and intrusion prevention but these are not able to detect every type of attack and new attacks [3]. Thus IDS are very important aspect of network security. Intrusion detection consists of two main components: anomaly detection and misuse detection. Anomaly detection component is used to discover attacks based on the traffic patterns that deviate from the normal behavior. However, misuse detection is used to search attacks based on the patterns extracted from known intrusion patterns [1-3].

Manish Kulariya, Final year Student in Department of Information Technology at National Institute of Technology, Raipur,(e-mail: kulariya.manish@gmail.com).

PriyankaSaraf, Final year Student in Department of Information Technology at National Institute of Technology, Raipur (e-mail: priyanka.saraf30@gmail.com).

RaushanRanjan, Final year Student in Department of Information Technology at National Institute of Technology, Raipur,(e-mail: dasraushan.nitrr@gmail.com).

Govind P. Gupta is with Department of Information Technology with the National Institute of Technology, Raipur (corresponding author) e-mail: gpgupta.it@nitrr.ac.in.

As the time changes, a lot of new attacks are being generated or the current attacks configurations are being changed, so a smart intrusion detection system is a need of the hour. Adaptively and fast detection is a very important aspect of any intrusion detection scheme, as a lot of new attacks are being generated on a day-to-day basis. A method should be adaptive so that it can detect new attacks as well.

Data mining schemes plays a vital role in intrusion detection [3-4]. It helps to extract malicious pattern from data that help to understand the nature of normal or any specific kind of attack. It helps to build various classifiers to detect different kinds of attack. As the numbers of users of Internet are increasing rapidly, the size of generated network traffic data is also increasing at a large scale. The conventional tools and techniques are not appropriate to use as they take a lot of time and may not able to handle the load. So there is a strong need to use a tool which is able to handle the increased load and gives better results in less time.

Apache spark [5-6] is such a tool specially designed for big data processing problem. It can analyze huge amount of data in a very less time. In this paper, we have employed five machine learning algorithms such as Logistic regression [7], Support vector machines [7], Random forest [7], Gradient Boosted Decision trees [7] and Naive Bayes [7] for detecting the attack traffic. For processing and detecting the attack traffic as fast as possible, we have used Apache Spark [5-6], a big data processing tool for detecting and analysis of intrusion in the communication network traffic. Performance comparison of intrusion detection schemes are evaluated in terms of training time, prediction time, accuracy, sensitivity and specificity. We have used a real time network traffic data set, called KDD'99[8] dataset for comparison and analysis of intrusion detection schemes.

This paper is organized as follows: Section II presents the related work. In Section III, brief descriptions of the various classifiers are given. In Section IV, we give a comparative study of these methods based on various metrics. And finally Section V carries the conclusion.

II. RELATED WORK

Intrusion detection system has been a very popular area in the field of cyber security of the network systems [1-3]. In [1], Huang et al. present analysis of attack strategy for designing a large scale distributed intrusion detection system. Tan Z et al. [2] proposed a collaborative intrusion detection scheme for enhancing big data security. In [3], Susan et al. proposed the use of fuzzy data mining and genetic algorithm for detection of intrusion. In [9], Zhang et al. discussed the application of

978-1-5090-0396-9/16/$31.00 ©2016 IEEE

unsupervised outlier detection scheme for network intrusion detection. Lazarevic et al. [10] studied a comparative analysis of anomaly detection schemes for detections of network intrusion. In [11], a survey of all kinds of network attacks is summarized by Hoque and its fellow researchers. G. V. Nadiammai et. al. [12] presented a comparative study of all rule based and some function based classifiers for detecting intrusion and provides results in terms of accuracy, sensitivity, specificity, time and error.

In [13], Kalyani et. al. discussed the comparison of classification techniques such as Naive Bayes, J48, OneR, PART and RBF Network algorithm for detection of intrusion in network traffic. In [14], Himadri et al. discussed a comparative study of classification techniques for detection of intrusion and they used ten classification algorithms namely J48, BayesNet, Logistic, SGD, IBK, JRip, PART, Random Forest, Random Tree and REPTree for detecting the intrusion in the network traffic and present their performance analysis. In this comparative analysis, they have used a very low size instance of data set i.e. in thousands records. All the above comparisons do not focus of fast detection of attack traffic, they only focus on accuracy. However, in this paper, we have taken big data size where number records used for processing are of big size i.e. around5 millions. In this paper, we not only focus on selecting an efficient and fast scheme but also measure its accuracy, scalability.

III. DESCRIPTION OF CLASSIFICATION TECHNIQUES

A. Logistic Regression

Logistic regression [7] is widely used to predict a binary response. It is considered a linear method with the loss function in the formula given by the logistic loss by:

$$L(w; x, y) := \log(1+\exp(-yw^T x)) \quad (1)$$

The algorithm outputs a binary logistic regression model for binary classification problems. If a data point is given, denoted by x, the model can make predictions by application of thefollowing logistic formula:

$$f(z) = 1/(1+e^{-z}) \quad (2)$$

Where $z = w^T x$. By default, when $(w^T x)>0.5$, the outcome is positive, else negative. The output of the logistic regression model calculated from the formula f(z), has a probabilistic interpretation (the probability that is positive).The logistic function's formula intakes an input with values ranging from negative infinity to positive infinity, but always outputs values lying in between zero and one and therefore is interpreted as probability measure.

B. Support Vector Machine

A standard method for classification of tasks on a large scale is linear SVM [7, 15-16]. This is a linear function with the loss in the formula given as the hinge loss:

$$L(w; x, y) := \max\{0, 1-yw^T x\} \quad (3)$$

By default, linear Support Vector Machines [7] are trained L2 regularization. SVMs construct a hyperplane or sets of hyperplanes in either a high or an infinite-dimensional space, that are be used for classifications, regressions, or similar tasks. The linear Support Vector Machines algorithm outputs a SVM model structure. For a new data node, denoted as x, the model predicts based on the value of $w^T x$. By convention, if $w^T x \geq 0$ then the outcome is said to be positive, else negative.

C. Naïve Bayes

Naive classifiers [7] dependent on Bayes classifiers. Applying Bayes' family based on simple probabilistic theorem having strong assumptions of independence between the features in machine learning. Classifiers of Naive Bayes [7] are scalable, and require a great number of parameters that are linear variables in a learning task. This classifier presents a simple multiclass classification algorithm with a simple assumption of independence among every pair of feature. It can be trained really efficiently. In a single iteration of the training data, the algorithm computes the conditional probability distribution of each feature of the given label, and then Bayes' theorem is applied to find the distribution of conditional probability of label for an observation and it is used for prediction.

D. Random Forest

Ensembles of decision trees are Random forests [7]. One of the most successful machine learning models are random forests whether classification or regression. This algorithm combines a number of decision trees so that it can reduce the over fitting risk. Same like decision trees, the random forests can handle the categorical features, can be extended to the multiclass classification settings, no requirement of feature scaling, and can capture non-linear aspects and feature interactions. What random forests do is train a set of decision trees separately because the training is done in parallel. The random forests inject randomness inside the training process so that each decision tree is different from the other. Predictions from each tree are combined which reduces the variance of the predictions, and thus improves the performance on the given test data.

E. Gradient Boosted Decision Tree

This scheme is an ensemble of decision trees. It is a machine learning algorithm which iteratively constructs an ensemble of weak decision tree learners through boosting techniques. In order to minimize the loss function, GBDT iteratively train decision trees. In each iteration, GBDT scheme uses the current ensemble to predict the label of each training instance and then compares the prediction with the true label. This scheme re-labeled the dataset in order to put more emphasis on training instances with poor predictions.

In this way, in the next iteration, the decision tree will help to correct the previous mistakes.

IV. BIG DATA PROCESSING TOOL: APACHE SPARK

A. Apache Spark [5-6]

Apache Spark [5-6] is an open source cluster computing platform designed for big data processing. In Comparison to the other big data processing tools such as Hadoop and Storm, it uses multi-staged in-memory processing scheme which results in 100 times faster processing than map-reduce processing. It also supports multiple languages like Java, Scala, or Python and provides a user friendly API and shells in Python, Scala, Java and SQL for handling jobs and writing queries. Spark can run Hadoop clusters and can access and process any Hadoop data sources [6]. The main core of Apache Spark [6] contains some basic functionality such as component of task scheduling, memory management, fault recovery and interaction with storage systems. RDDs (resilient distributed data sets) are the main programming abstraction of Apache Spark. It represents a set of things which is distributed across many computing nodes for processing data in parallel.

V. PERFORMANCE EVALUATION

In this section, we have presented the performance evaluation of the intrusion detection schemes in terms of training time, prediction time, accuracy, sensitivity and specificity. For testing the system, we have used a real time network traffic data: KDD'99 data set. Experiments are performed using a latest big data processing tool, called Apache Spark. The machine learning library of Spark, called MLib, is used. We have used five well known classifier such as Logistic regression, Support vector machines, Random forest, Gradient Boosted Decision trees & Naive Bayes for designing the IDS.

A. KDD-99 Data Set

For performance evaluation of the intrusion detection scheme, we have used KDD'99 dataset [8, 17]. This is a most widely used dataset for the evaluation of intrusion detection schemes. This dataset was prepared by Stolfo et al. [8] and was based on the experimental data collected during DARPA'98 Intrusion Detection evaluation program. KDD'99 training data set consists of approximately 4898431 instances and testing dataset consists of 311029 connection instances. Each single connection instance contains 41 features and every connection instance is labeled as either normal or an attack [8, 17].

B. Performance Matrices

For comparison of the intrusion detection schemes, we have used five performance matrices are used in this paper such as accuracy, sensitivity, specificity, training time and prediction time. Accuracy, sensitivity and specificity are calculated using true positive measure, false positive measure, true negative measure and false positive measure.

- *True Positive Measures:* Attacks correctly identified as attacks.
- *False Positive measures*: Normal incorrectly identified as attacks.
- *True Negative Measures:* Normal correctly identified as normal.
- *False Negative Measures:* Attack incorrectly identified as normal.

a) *Accuracy:* It is the most important measure of any classification scheme. It describes how accurately a scheme can detect connections as normal or attack.

$$\text{Accuracy}= \frac{TP+TN}{TP+TN+FP+FN} \quad (4)$$

b) *Sensitivity:* It is also called true positive rate. It is used to measures the proportion of positives that are correctly identified as such.

$$\text{Sensitivity}= \frac{TP}{TP+FN} \quad (5)$$

c) *Specificity:* It is also called true negative rate. It is used to measures the proportion of negatives that are correctly identified as such.

$$\text{Specificity}= \frac{TN}{TN+FP} \quad (6)$$

d) *Training Time:* Time taken to train a classifier

e) *Prediction Time:* This describes how much time a particular algorithm has taken to predict whole data set as normal or attack.

C. Result Analysis

In this section, we discuss the result analysis of intrusion detection scheme which is evaluated on KDD'99 datasets.

i. Performance analysis in terms of Accuracy, Sensitivity and Specificity

In this experiment, we have compared the performance of five intrusion detection schemes which is designed based on well known classifiers such as Logistic regression, Support vector machines, Random forest, Gradient Boosted Decision trees & Naive Bayes for detecting the attack traffic. Results are listed in Table I.

As results are listed in Table I, it is observed that Naïve Bayes based scheme perform better than all the remaining schemes in terms of sensitivity. This is due to fact that it gets 99% sensitivity followed by SVM with 92%. Logistic Regression and GB Tree have almost same sensitivity with values 90.02% and 89.23% to be precise. Random Forest is the least ranked amongst all the classifiers in terms of sensitivity.

TABLE I
COMPARISON OF DIFFERENT INTRUSION DETECTION METHODS ON VARIOUS METRICS

Method	Accuracy	Sensitivity	Specificity	Training Time	Prediction Time
Logistic Regression	91.64	90.02	98.31	341.665	21.230
SVM	92.13	92.17	91.16	561.044	26.369
Naive Bayes	91.45	99.41	58.56	96.871	22.025
Random Forest	91.66	89.79	99.38	175.739	20.326
GB Tree	91.29	89.23	99.8	354.771	23.276

From the Table I, it is observed that specificity for the Random Forest and GB Tree based schemes are almost same with approximately 99% specificity. However, specificity for Logistic Regression and SVM based scheme are approximately 98% and 91% respectively. Among the all schemes, GB tree based scheme perform better in terms of specificity. However, Naïve Bayes based scheme perform poor in terms of specificity amongst all the schemes. The accuracy of SVM based scheme is better than all with 92.13%. However, accuracy of the GB tree based scheme is lower among the all schemes.

ii. Performance Analysis in terms of Training Time

Fig. 1. Performance Analysis of different intrusion detection methods in terms of Training Time.

As shown in Fig. 1, Naïve Bayes is fastest of all classifier methods to train, with a training time of just 96 seconds. On the other hand, SVM is most costly classification method used among all the five. SVM took a total of 561 seconds to train. Random Forest is second most time efficient method as it took 175.739 seconds followed by GBTree and Logistic regression took almost 350 seconds.

iii. Performance analysis in terms of Prediction Time

It is observed from the Fig.2.that Random Forest took the least time approximately 20 seconds and thus is the fastest detection scheme of all. Whereas SVM took highest time to predict with the maximum time of 26.369 seconds and thus become the slowest scheme of all the detection methods. Logistic Regression ranks the third fastest scheme with approximately 21 seconds to predict, followed by Naïve Bayes with 22 seconds and GB Tree with approximately 23 seconds.

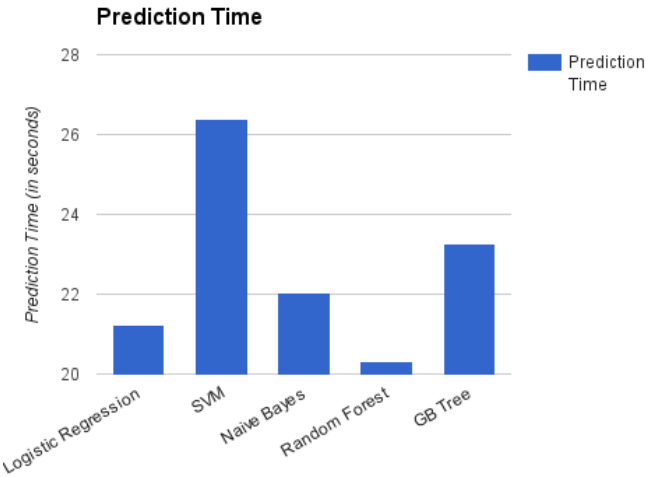

Fig. 2. Performance Analysis of different intrusion detection methods in terms of Prediction Time.

VI. CONCLUSION

The objective of this experiment was to find algorithms which can accurately classify the records and at the same time take less time to predict. Experiment was done using KDD'99 dataset using Apache spark. Random Forest gave the best performance with respect to all measures accuracy, sensitivity and specificity. It displayed approximately 99% specificity, 89% sensitivity with 91% accuracy in just a matter of 175 seconds of training time. Although Naïve Bayes has the least training time amongst all the methods, it lacks behind in specificity by a huge difference as compared to all the classifiers. Rest all algorithms; Logistic Regression, GB Tree and SVM give results almost similar to all the measures, that is their values are nearly the same in terms of specificity, sensitivity and accuracy. However, their training time differs by a significant amount, thus placing Logistic Regression at the second position, GB Tree at the third and SVM at the fourth. Naive Bayes has the last rank amongst all because of the significant drop in its specificity percentage. Our future work will be focused on enhancing the accuracy and other measures of these particular algorithms using different technologies.

REFERENCES

[1] Huang M-Y, Jasper RJ, Wicks TM, "A large scale distributed intrusion detection framework based on attack strategy analysis", Computer Network, pp. 65–75, 1999.

[2] Tan Z, Nagar UT, He X, Nanda P, Liu RP, Wang S, Hu J.,"Enhancing big data security with collaborative intrusion detection", *IEEE Cloud Computer*, pp. 27–33, 2014.
[3] Susan M. Bridges, and Rayford B. Vaughn, "Fuzzy Data Mining and Genetic Algorithms Applied to Intrusion Detection", Proceedings of the National Information Systems Security Conference (NISSC), Baltimore, MD, October, 2000.
[4] Ho C-Y et. al., "Statistical analysis of false positives and false negatives from real traffic with intrusion detection/prevention systems", *IEEE Communication Magazine*, pp. 46–54, 2012.
[5] Apache Spark™ - Lightning-Fast Cluster Computing, http://spark.apache.org/
[6] Holden Karau et al." Learning Spark", Published by O'Reilly Media, Inc, 2015.
[7] C.C. Aggarwal, Data Classification: Algorithms and Applications, CRC Press,2014.
[8] KDD Cup 1999 dataset. Avaliable online: http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html.
[9] J. Zhang, M. Zulkernine. "Anomaly based network intrusion detection with unsupervised outlier detection". *Proc. of the IEEE ICC* 2006. pp. 2388-2393.
[10] A. Lazarevic, L. Ertoz, A. Ozgur, J. Srivastava & V. Kumar, "A Comparative Study of Anomaly Detection Schemes in Network Intrusion Detection", Proceedings of Third SIAM Conference on Data Mining, San Francisco, May 2003.
[11] Hoque N et. al. ," Network attacks: taxonomy, tools and systems. Journal of Network & Computer Application, pp. 7-24, 2014.
[12] G. V. Nadiammai and M. Hemalatha, "Perspective analysis of machine learning algorithms for detecting network intrusions," *IEEE Third International Conference on Computing Communication & Networking Technologies (ICCCNT)*, Coimbatore, India, 2012, pp. 1-7.
[13] G. Kalyani and A. J. Lakshmi, "Performance Assessment of Different Classification Techniques for Intrusion Detection," IOSR Journal of Computer Engineering (IOSRJCE), vol. 7, no. 5, pp. 25-29, 2012.
[14] HimadriChauhan, Vipin Kumar, SumitPundir and Emmanuel S. Pilli,"A Comparative Study of Classification Techniques for Intrusion Detection", In. Proceedings of International Symposium on Computer and Business Intelligent, pp.40–43, 2013.
[15] H.A. Nguyen and D. Choi, "Application of Data Mining to Network Intrusion Detection: Classifier Selection Model," Lecture Note in Computer Science, pp. 399–408, 2008.
[16] Cristianini N, Shawe-Taylor J," An introduction to support vector machines: andother kernel-based learning methods", Cambridge University Press; 2000.
[17] R. Lippman R et al. "The 1999 DARPA off-line intrusion detection evaluation", Journal of Computer Networks, 2000, 34(4):579-595.