Decision Tree: A Machine Learning for Intrusion Detection

Shilpashree. S, S. C. Lingareddy, Nayana G Bhat, Sunil Kumar G

Abstract— The Intrusion is a major threat to unauthorized data or legal network using the legitimate user identity or any of the back doors and vulnerabilities in the network. IDS mechanisms are developed to detect the intrusions at various levels. The objective of the research work is to improve the Intrusion Detection System performance by applying machine learning techniques based on decision trees for detection and classification of attacks. The methodology adapted will process the datasets in three stages. The experimentation is conducted on KDDCUP99 data sets based on number of features. The Bayesian three modes are analyzed for different sized data sets based upon total number of attacks. The time consumed by the classifier to build the model is analyzed and the accuracy is done.

Keywords— Intrusion Detection System, Machine Learning, Deep Learning, Decision Tree

I. INTRODUCTION

Recently, digital content has increased dramatically in all areas, thus increased fears of digital attacks as well. From this standpoint, systems and mechanisms have emerged to track and protect organizations, institutions, and individuals. The intrusion detection system (IDS) is one of the most important systems to protect and track intrusion in computer networks [1]. IDS work mechanism is summarized in three points: monitoring the network, verifying happenings in the network, and reporting events that do not match with the security policies of the network administrator. The IDS system is characterized by that it incorporates the methods of detecting and identifying intrusive and non-intrusive network packets. At present time, human analysts analyze system logs for intrusion detection systems in order to distinguish between intrusive and non-intrusive network traffic [2]. Therefore, we note that most of these systems do not overlook the role of the human factor and depend on him in most data analysis operations.

The intrusion detection can be classified into two major types [1]. First one is signature based, which is interested in analyzing network traffic for a sequence of bytes or packet sequences and the second one is the anomaly-based detection system. One of the disadvantages and the observations that can be discovered in this type is signatures are comparatively fair easier to develop and understand. Thus, this leads to knowing the behavior to be determined in the network. Another disadvantage of a Signature-based type detection is reliance on the approach of detecting attacks that have the only signature, but unable to detect any other unregistered attacks in the IDS. The anomaly network intrusion detection is a major part of network security, Where the behavior of anomaly can be represented as the normal use of data. One of the challenges that must be overcome in an intrusion detection system based on anomaly detection is how to classify the normal and abnormal activities so that we can distinguish between the processes effectively [4]. Recently most of the effective intrusion detection systems rely on the machine learning system, where the machine learning mechanisms are very functional so that it provides high possibility of detecting the intrusions in the network. The main reason for the ability of the machine learning system in the detection of intrusions is that it uses supports vector machines, neural networks, and all these possibilities are based on decision trees that have efficient significant schemes in anomaly detection systems. Thus, it enhances the classification performance and accelerate the speed of processes.

Machine learning gives systems the capacity to naturally take in and enhance as a matter of fact without being expressly customized. Machine learning centers around the advancement of computer programs that can get to information and utilize it learn for themselves [4]. A decision tree is defined as a tree-like diagram that contains a tree trunk representing the internal nodes to express a test of a particular characteristic, the branches representing the result of this test, and the leaves representing the nodes as a class mark.

The purpose of classification in the approach of the decision tree is to frame the data so that it contains both the root node and the leaf node. Decision trees can examine information and recognize critical qualities in the system that illustrate the malicious activities. This, in turn, increases the value of some security frameworks by checking the arrangement of intrusion identification information. It can perceive patterns and examples that promotes to check, the advancement of attack signatures, and different activities of checking. What distinguishes the utilizing of the decision trees method from other methods is that the decision tree gives a rich arrangement of rules, which are straightforward, and can be easily integrated with the technologies that are real-time [1].

Revised Manuscript Received on April 12, 2019.

Shilpashree. S, Department of Computer Science and Engineering Sri Venkateshwara College of Engineering Bengaluru, Karnataka, India. (shilpatani@gmail.com)

S. C. Lingareddy, Department of Computer Science and Engineering Sri Venkateshwara College of Engineering Bengaluru, Karnataka India. (sclingareddy@gmail.com)

Nayana G Bhat, Department of Computational Engineering, CIIRC Jyothy Institute of Technology Bengaluru, Karnataka India. (sunainabhatm@gmail.com)

Sunil Kumar G, Department of Computer Science and Engineering, Vijaya Vittala Institute of Technology, Bengaluru, Karnataka India. (gsuneel.k@gmail.com)

II. RELATED WORK

Both Bajaj and Arora discussed in their research paper [3] the different distinctive selection methods such as information gain, gain ratio, and correlation-based feature selection, where they selected 33 features out of 41 then classified these features for comparing the results. This algorithm is called the Simple Cart Algorithm (SCA) and the results that are obtained gives the highest accuracy approximately equal to 66.77%, whereas the classification result of the C4.5 decision tree is 65.65% only .One of the research that is similar to the classification of select features is the paper of Alazab et al. [2], where they have used information gain and decision tree to detect both the old and the new attacks. The research paper [4] discussed the problems of NIDS techniques and then proposed a method called NDAE for unsupervised feature learning. The second step in the proposed method is that it builds a novel model of classification which was built from stacked NDAEs and the RF classification algorithm. The proposed model has been implemented and results evaluated by using Tensor-Flow, where evaluations have utilized the benchmark KDD Cup '99 and NSL-KDD datasets and achieved very satisfactory results. The results of the mechanism of the researchers in this research are that the method provides high levels of accuracy and recall together with reduced training time where the proposed algorithm was compared with DBN technique.

Improving network performance is a challenge that plays an important role in increasing productivity, so in [6] it has been reviewed that various methodologies of implementing DL schemes for the wireless networks. In a nutshell, (1) DL/DRL is very useful for intelligent wireless network management and the reason for that is its ability to recognize the pattern of the human brain. Therefore, with the continuous development in the performance of hardware components of wireless products, the implementation of human brain pattern recognition has become possible and easier. (2) It plays important roles in the multiple protocols layers. Therefore, the physical, MAC and routing layers are where the implementation of the applications is done. This makes the network more intelligent and able to recognize the changes that take place in the entire topology and link conditions, thus, it help to create more convenient protocol parameter controls.(3) The possibility to integrate with different wireless network schemes, used today, such as CRNs, SDNs, etc., and the goal is the ability to implement both centralized or distributed resource allocation and traffic balancing functions. In addition, it cannot be overlooked. Some important things that are listed in this article where, it opens ten research issues in front of researchers to solve them such as network swarming, CRN spectrum handoff, SDN flow table update, dew/fog computing security, etc.Qian Mao et al [7] reviewed comprehensive paper, where they summarized the importance of adopting machine learning algorithms in wireless sensor networks, where it was illustrate that those algorithms should be flexible with the limited resources of the network and deal with the diversity of learning themes and the different patterns, which will suit the problem at hand. Therefore, the researchers must focus on some issues that are still open such as developing lightweight and distributed message passing techniques, online learning algorithms, and hierarchical clustering patterns. Thus, machine learning will enable managing the diverse resources that support to solve the problems of the wireless sensor networks.

Mohammad Abu Alsheikh et al in [8] have proposed hybrid KNN and Neural Network based multilevel classification model. In this model, KNN was utilized as a classifier for anomaly detection with two classes, namely, 'normal' and 'abnormal'. The next step is using a neural network and the goal is to detect a specific type of attack in 'abnormal' class. For instance, the NSL-KDD dataset was used. Implementation steps can be listed as follows:

(i) All the features of the dataset were used for classification.

(ii)The classification is performed on 25 selected features.

Rough Set Theory and Information Gain selected samples separately. What characterizes this classification model is that the Information Gain that contains 25 features of the NSLKDD dataset gives better results compared with 25 features with Rough Set Theory as well as 41 features of NSL-KDD dataset. The authors in [5] designed a multi-layer hybrid machine learning IDS. Where PCA has been used for selecting the attribute, in the first layer of the IDS has selected 22 features. With regard to the next layer was used GA to generate detectors that can distinguish between normal and abnormal behaviour. Finally, in the third layer, the classification was done using several classifiers. The results show that Naive Bayes has good accuracy for two types of attacks, which are User-to-Root (U2R) and Remote-to-Local (R2L) attacks. Nevertheless, the decision tree gives a higher accuracy of up to 82% for Denial-of-Service attacks and 65% of probe attacks.

III. IMPLEMENTATION

The purpose of decision tree approach of ML is to build a decision tree to validate the incoming traffic depending on an available data set to empower it to group the new cases accurately. Out of many approaches available for the construction of the decision tree, the CRT (Classification and Regression Trees) is of interest. In this paper, we make use of CART for our Intrusion Detection System. The following formula is required for this approach.

Let the dataset be S = {(a1, b1), (a2, b2), . . ., (a_N, b_N)}, where $a_i = (a_i^{(1)}, a_i^{(2)}, \ldots, a_i^{(n)})^T$, $i = 1, 2, \ldots, N$,

a_i is the instance of the input and indicates a record for network packet. There are n features in a_i. The variable N represents the amount of packet records included in the dataset S.

$b_i \in \{0, 1, 2, \ldots, M-1\}$is the class tag that means the output of every record detected.

For the purpose of evaluation, we have to introduce the *Score_indicator* and the *Computation_time*. The *Computation_time* of the Intrusion Detection System algorithm is represented by *t*. The time *t* gives the build time and the time taken for detection by the proposed approach. Expecting that we characterize an example dataset as both normal and abnormal, there are four instances of the

classification. As appeared in Table I, that is, the Positive True, Positive, Negative True, and Negative False. True implies that the classification is right while False implies that the classification isn't right. Positive implies that the classifier is separated into ordinary examples and Negative implies that the classifier is partitioned into unusual examples:

Sl.No	Instances	Description
i)	Positive True (PT)	Ordinary case is distinguished accurately.
ii)	Positive False (PF)	Unusual case is mistakenly classified as ordinary.
iii)	Negative False (NF)	Ordinary case is misclassified as unusual one.
iv)	Negative True (NT)	Unusual case is distinguished accurately.

Table I: Dataset Classification Instances

Exactness E indicates the extent of applicable occurrences among the detected examples. E can be gotten by the accompanying equation:

$$E = \frac{PT}{PT+PF}$$

Let the T indicates to the extent of significant occurrences that have been identified over the aggregate sum of relevant examples. T can be gotten by the formula given as:

$$T = \frac{PT}{PT+NF}$$

As a matter of fact, markers of E and R are now and again conflicting, and hence *score Indicator* is the regular assessment pointer.

The Score_indicator is calculated by taking average of E and T that is obtained from the formula given below:

$$\text{Score\_indicator} = \frac{(\theta^2+1)E*T}{\theta^2(E*T)}$$

when, Θ = 1, the *score indicator* will become as:

$$\text{Score\_indicator} = \frac{2ET}{E+T}$$

The approach for building the system is depicted in the fig.1. The data set is given as input to the working model. The working model consists of three stages: the preprocessing stage, normalization stage and decision tree building. The input dataset is usually consisting of strings and numbers. As the value of string cannot be compared directly, we are required to digitize the string by making use of string manipulation operation, and this is done by preprocessing stage. This process is explained as shown in pseudo code 1.

Pseudocode 1: Data set Preprocessing
Read: *Data set DS*
for x *= 1 to N do*
for y *= 1 to N*
(St, n)= compute(processedDS)
end for
end for
for z *= 1 to ndo*
processedDS(St) =D(j)
end for
return *processedDS*

We initially navigate through the input data set *D*S and discover every one of the strings *S*t in data set *D*S and acquire the comparing sections n by utilizing compute () operation. Also, we call the supplant operation to supplant *S*t with random number j. At the end the processed dataset is retuned. And this data set is used as input to the next stage 2.

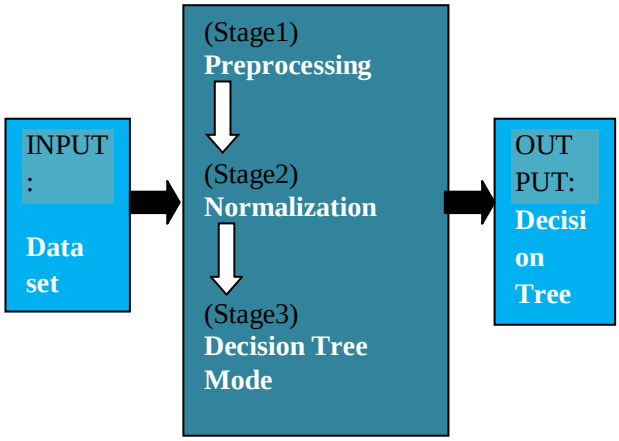

Fig 1: An approach for IDS using Machine Learning.

The processed data set may not contain uniformity. There may be smaller number of column sections that may play crucial role. Therefore, its needed to perform the normalization of processed data before its given to detection algorithm. The need of the stage 2, normalization is to provide the characteristic data shrink. This process is explained as shown in pseudo code 2.

Pseudocode 2: Normalization of Processed data set
Read: *preprocessed Data set*
DS1, DS2 = splitDataSet (processed DS, k%)
A_ set = normalize (DS1)
B_set = normalize (DS2)
return *A_ set, B_set*

The training dataset *D*S1 is selected randomly k% from the *processed DS* as the training dataset *D*S1 and the balance as training data set *D*S2 which is equals to (1 − k%). The results *A_set* and *B_set* are obtained by making use of normalize operation. These will be used as input to the next stage 3.

In decision tree mode, we build the decision tree by making use of training data set *A_set* as input to CRT_fn() function and then obtain the output of the checking data set *B_set*. This process is explained as shown in pseudo code 3.

Pseudocode 3: decision tree
Input: *A_ set, B_set*
dtmode= CRT_fn(A_ set)
X= dtmode (B_set)
(score_indicator, t) = generate ()
return *score_indicator, computation_time t*

The *dtmode*, decision tree mode is obtained by making use of the function *CRT_fn* by applying the Classification and Regression Trees methodology. This *dtmode* is applied on *B_set to produce X, and the score_indicator* and *computation_time t* are generated using *generate()* operation.

IV. EXPERIMENTATION AND RESULT ANALYSIS

The experimental analysis is done by evaluating our intrusion detection system on premier data set KDDCUP99.The implementation is done using Python on a windows platform. This research work requires an extensive number of legitimate test information. Information accumulation can be gotten through some capturing devices, such as Libdump, TCPdump and Wireshark, and afterward association record is created as the information hotspot for system. In this investigation, we utilize KDDCUP99 dataset for our test. The dataset is, 9-week data of network gathered from a reenacted LAN of Air Force that belongs to US. The dataset consists of two sorts, the previous one is 20% dataset termed as kddcup.data.20precent.correceted and the second one is the full dataset termed as kddcup.fulldata.corrected. Every dataset recorded in KDDCUP99 consists of a fixed forty-one characteristics and a class label. The four types of attacks, namely U2R (unauthorized access to local super user), R2L (unauthorized access from a remote machine), DOS (denial-of-service) and probing (surveillance and other probing) have more detailed separations. The experimentation completes the detection of all the four types of attacks. The Table II gives the details of subclasses of attacks belonging to these four major types of attacks. There are 27 types of subclasses of attacks listed. The Table III. Demonstrates the total number of kddcup 99 instances

Classification of attacks	**Subclasses**
DOS (Denial of service attack)	land, back, pod, neptune, teardrop, smurf,mailbomb, apache2
R2L (Illegal access from remote machines)	imap, multihop, phf, spy, warezmaster, Xlock, warezclient, snmpgetattack, ftp_ write
U2R (Unauthorized access of ordinary users) To privileges of administrator)	buffer_overflow, loadmodule, perl, rootkit,guess_passwd
Probing (Monitoring and other detection activities)	Ipsweep,nmap, portsweep,satan,saint

TABLE II: CLASSIFICATION OF ATTACKS.

Attack Type	**Total Training Instances**	**Total Testing Instances**
Dos	1807	3168
U2R	70	105
R2L	216	2338
Probing	1391	2579
Normal	7500	10000
Total	10984	18200

TABLE III. TOTAL NUMBER OF KDDCUP 99 INSTANCES

The experiment results are compared from the *score_indicator* and *computation_time*. To ensure that all types of attacks are covered, the dataset is randomly divided based percentage as training and test data sets. Hence, the result of this, forms three models such as BernoulliNB, MultinomialNB and GaussianNB in Naive Bayesian. This makes the best approach for our Intrusion detection system. We perform three group experiments for every method: (i) 27 types of attacks over 20% of data set (ii) 8 types of attacks over full data set (iii) 27 types of attacks over full data set.

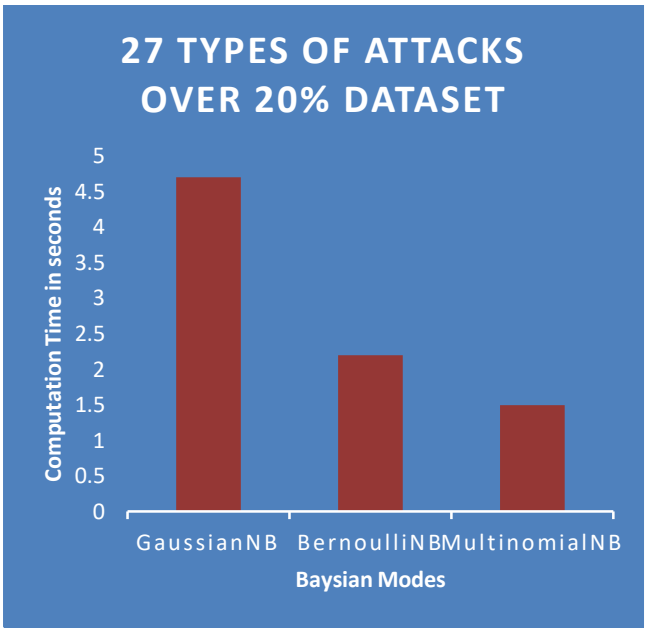

Fig.2. Graphical analysis for 27 types of attacks over 20% of the data set.

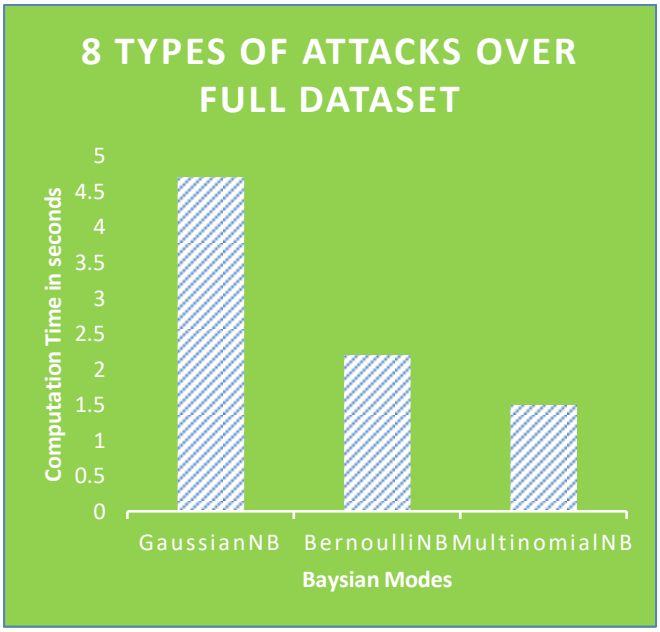

Fig.3. Graphical analysis for 8 types of attacks over full data set.

The Fig. 2 shows the graphical analysis for three Baysian modes. The computation time contrast results are demonstrated. The MultinomialNB gets the least computation time among all the testcases, followed by BernoulliNB, and GaussianNB is the last one. Accordingly, the Fig.3 and Fig.4 gives the graphical analysis for 8 types of attacks over full data set and 27 types of attacks overfull data set respectively. The computation time for all three Baysian modes are analyzed accordingly.

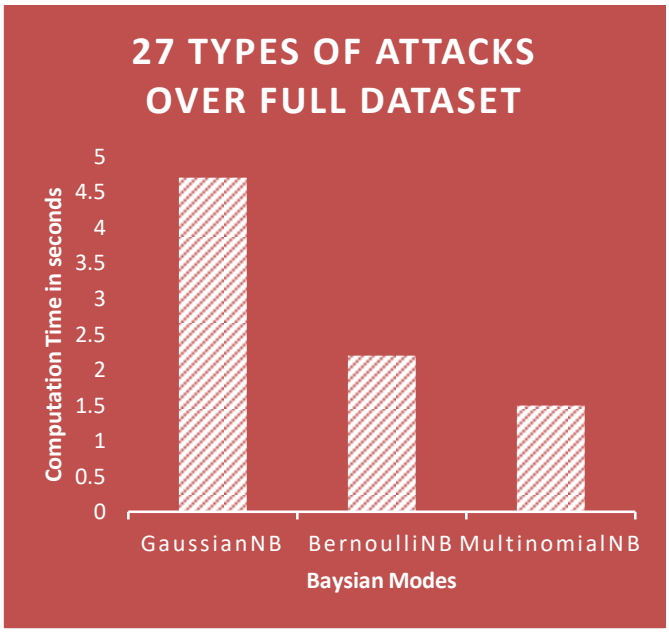

Fig.4. Graphical analysis for 27 types of attacks over full data set.

V. CONCLUSIONS

The attacks are generally present in the networks. The decision tree is employed to help the system administrator to conclude about approaching traffic. i.e., regardless of whether the incoming information is malicious or not by building a model that isolates noxious and non-vindictive traffic. In this work, we built a system based on the decision tree, different strategies are also contrasted with this approach. Both 20% dataset as well as the full data set is tried, and the experiment results demonstrate that our framework is sufficiently powerful. In the future, we will take part in the investigation of the IDS for different sorts of attacks.

REFERENCES

1. J. Markey, Using Decision Tree Analysis for Intrusion Detection: A How-To Guide, SANS Institute InfoSec Reading Room, June, 2011.J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.
2. A. Alazab, M. Hobbs, J. Abawajy, and M. Alazab, Using Feature Selection for Intrusion Detection System, International Symposium on Communications and Information Technologies, 2012.
3. K. Bajaj, and A. Arora, Improving the Intrusion Detection using Discriminative Machine Learning Approach and Improve the Time Complexity by Data Mining Feature Selection Methods, International Journal of Computer Science, vol. 76, Aug, 2013.R. Nicole, “Title of paper with only first word capitalized,” J. Name Stand. Abbrev., in press.
4. A.S.A. Aziz, A.E. Hassanien, S. El-Ola Hanafy,M.F. Tolba, Multi-layer hybrid machine learning techniques for anomalies detection and classification approach, 13th International Conference on Hybrid Intelligent Systems (HIS), 2013, IEEE.M. Young, The Technical Writer’s Handbook. Mill Valley, CA: University Science, 1989.
5. P. Ghosh, C. Debnath, D. Metia, and Dr. R. Dutta, An Efficient Hybrid Multilevel Intrusion Detection System in Cloud Environment, IOSR Journal of Computer Engineering (IOSR-JCE) e-ISSN: 2278-0661, p-ISSN: 2278-8727, Volume 16, Issue 4, Ver. VII (Jul –Aug. 2014), PP 16-26.
6. Nathan Shone, Tran Nguyen Ngoc, Vu DinhPhai, Qi Shi "A Deep Learning Approach to Network Intrusion Detection" in IEEE TRANSACTIONS ON EMERGING TOPICS IN COMPUTATIONAL INTELLIGENCE, NOVEMBER 2017, Manuscript submitted 30 June 2017.
7. Qian Mao, Student Member, IEEE, Fei Hu, Member, IEEE, and Qi Hao, Member, IEEE, "Deep Learning for Intelligent Wireless Networks: A Comprehensive Survey" JOURNAL OF LATEX CLASS FILES, VOL. 14, NO. 8, JANUARY 2018.
8. Mohammad Abu Alsheikh, Shaowei Lin, Dusit Niyato1 and Hwee-Pink Tan, "Machine Learning in Wireless Sensor Networks:” IEEE Communications Surveys & Tutorials (Volume: 16, Issue: 4, Fourth quarter 2014), 24 April 2014, pp.1996 – 2018.