Journal of Multidisciplinary Engineering Science and Technology (JMEST)
ISSN: 3159-0040
Vol. 2 Issue 4, April - 2015

An Intrusion Detection System, (IDS) with Machine Learning (ML) Model Combining Hybrid Classifiers

Arjunwadkar Narayan M.
Computer Engineering
Sinhgad Institute of Technology, Lonavala
S.P.Pune University (M.S) INDIA 410401
narjunwadkar@gmail.com

Thaksen J. Parvat
Computer Engineering
Sinhgad Institute of Technology, Lonavala
S.P.Pune University (M.S) INDIA 410401
pthaksen.sit@sinhgad.edu

***Abstract*— An Intrusion Detection System (IDS) with Machine Learning (ML) model Combining Hybrid Classifiers i.e. Naïve Byes classifier and C 4.5 classifier is proposed for intrusion detection. In the proposed model, a multi-layer Hybrid Classifier is adopted to estimate whether the action is an attack or normal data. First, a misuse detection model is built based on the C4.5 decision tree algorithm and then the normal training data is decomposed into smaller subsets using the model. Next, multiple one-class Naïve Byes algorithm models are created for the decomposed subsets Hybrid Classifier is used as a preprocessor of Intrusion Detection System to reduce the dimension of feature vectors and shorten training time. In order to reduce or minimize the noise caused by feature differences and improve the performance of Intrusion Detection System. The proposed hybrid intrusion detection method was evaluated by conducting experiments with the NSL-KDD data set, which is a modified version of well-known KDD Cup 99 data set.**

Keywords—Intusion Detectio, Machine Learning, Security, Hybrid Classifiers

I. INTRODUCTION

An intrusion detection system (IDS) or Network intrusion detection system (NIDS) has been developed that is capable of detecting all types of network anomalies or attacks in the available environments. The IDS is placed inside the network that it protects, and it collects network packets promiscuously in the same manner as a Packet Analyzer or network sniffer. The IDS detects malicious network activities by analyzing the collected packets, alarms to system administrator, and blocks attack connections in order to prevent further damage from attacks. It also connects to the firewall as a fundamental technology for network security. Intrusion detection algorithms are categorized into two methods: misuse detection and anomaly detection.[1][3][6]Misuse detection algorithms detect attacks based on the known attack signatures. They are useful in detecting known attacks with minimum errors. However, they cannot detect newly created attacks that do not have similar properties to the known attacks. In contrast, anomaly detection algorithms analyze normal traffic and profile normal traffic patterns.[12][13] The anomaly detection method is based on the hypothesis that the attacker behavior differs to that of a typical user. They classify traffic as an attack if the characteristics of the traffic are far from those of normal traffic patterns. Anomaly detection algorithms can be useful for new attack patterns. They are not as effective as misuse detection models in the detection rate for known attacks and false positive rates, which is a ratio of misclassified normal traffic.

In order to resolve the disadvantages of these two conventional intrusion detection methods, hybrid intrusion detection methods that combine the misuse detection method and the anomaly detection method have also been proposed. Because none of the misuse and anomaly detection methods are better than any other, a hybrid intrusion detection system uses both the misuse detection method and anomaly detection method. The detection performance of the hybrid intrusion detection system depends on the combination of these two different detection methods. Most hybrid detection systems independently train a misuse detection model and an anomaly detection model, and then simply aggregate the results of the detection models. For example, hybrid intrusion detection systems regard a traffic connection as an attack if at least one of the two models classifies the traffic connection as an attack. In this case, the detection rate will be improved but the IDS will still have a high false positive rate. In contrast, if the hybrid method regards a traffic connection as an attack only if both models classify the connection as an attack, false alarms will be reduced but it may overlook many attack connections.[1][3]

II. INFORMATION GAIN AND MACHINE LEARNING TECHNIQUES

In this section, the C4.5 and classes NaïveByes classifier that are required in order to build the hybrid intrusion detection model, are briefly introduced. Then, the integration of these models is explained, and the properties of the proposed hybrid intrusion detection method are discussed.

A. Decision tree and C4.5

A decision tree (DT) or C4.5 is one of the most widely used classification algorithms in data mining. It operates in a divide and conquer manner, which recursively partitions the training data set based on its

attributes until the stopping conditions are satisfied.[1][4][14] The C4.5 consists of nodes, edges, and leaves. A C4.5 node has its corresponding data set; this specifies the attribute to best divide the data set into its classes. Each node has several edges that specify possible values or value ranges of the selected attributes on the node. The data set of the node is divided into subsets according to the specifications of the edges, and the C4.5 creates a child node for each data subset and repeats the dividing process. When the node satisfies the stopping rules because it contains homogeneous data sets or no future distinguishing attributes can be determined, the C4.5 terminates the demarcation process and the node is labeled as following the class name of the data set.[1]This labeled node is called a leaf node. In this way, the C4.5 recursive partitions the training data set, which creates a tree-like structure.

The latest public domain implementation of Quinlan's model is C4.5. The primary issue of the decision tree algorithms is to locate the attribute that best divides the data into their corresponding classes. C4.5 builds decision trees from training data sets using the concept of information entropy. That is; it is based on the highest gain of each attribute. The Information gain is calculated using the following formula:

$$IG(S,A) = Entropy(S) - \sum_{i=1}^{n} \frac{|Si|}{|S|} * Entropy(Si) \quad (1)$$

Where Information Gain (IG)(S, A) is the gain of set S after a split over the A attribute; Entropy(S) is the information entropy of set S; n is the number of different values of attribute A in S; A is the proportion of items possessing Ai as the value for A in S; Ai is the ith possible value of A; and Si is a subset of S containing all items where the value of attribute A is Ai. Here, the entropy is obtained as follows:

$$Entropy(j) = \sum_{j=1}^{n} fs(j) - \log_2 fs(j) \quad (2)$$

Where n is the number of different values of the feature in S (entropy is computed for one chosen attribute) and fs(j) is the proportion of the value j in the set S.

After the tree is created by maximizing the gain, the C4.5 model decomposes the data space such that individual decomposed regions become homogeneous. Then, C4.5 performs the final pruning step. This action reduces the classification errors caused by specializations in the training set; thus, it makes the tree more general. In this study, the C4.5 is used to train the misuse detection model in the hybrid intrusion detection system. Both normal and attack data are used to train the model: C4.5 divides the data into decomposed regions and labels the parts as the classes of major data belonging to each decomposed region.[1]

B. Naive Bayes classifier

Many classifiers can be computing a set of probability distribution functions and in this one of the class whose probability is maximum. [11] In the structural relationship and the /or casual dependencies between the random variables of any problem, the NaïveByes use a probabilistic graph model. The structure of the NaïveByes typically described into directed acyclic graph (DAG). In the NaïveByes, classifier node is represented system variable, and link is nothing but the connection between two system variables. [2][11]

There are many recent IDSs researches exploits Bayesian theory to classify network traffic as normal or as attack events

$$p(C_j\ X) = \frac{p(X\ C_j)p(C_j)}{P(X)}$$

$$IFF\ p(C_j) > P(C_iX), 1 \le i \le m, i \ne j \quad (3)$$

Where Class C j € given set of m Classes {C1, C2, C3... … … … … Cm} X is an unknown data sample and P(X) is constant for each one category. It is sufficient to determine only the numerator term be-cause of P(X) is constant for every 'X'; therefore, Naive Bayes classifier determines Eq.(2)to allocate a sample of unknown X to class Cj.

$$P(C_j) = P(X\ C_j)P(C_j)$$

$$IFF\ P(C_jX) > P(C_iX), 1 \le i \le m, i \ne j \quad (4)$$

To apply a Naive Bayes classifier in IDS; Priori probability P(Cj) can be determined using the training data set in Eq. (3), and if the sample has many attributes, then P(Cj) can be determined using Eq.(4) [2].

$$P(C_j) = \frac{S_j}{S} \quad (5)$$

Where Sj is the training sample size in the class Cj, and Sis the total number of the training samples.

$$P(A\ C_j) = P(A_1C_j)P(A_2C_j) \dots\dots\dots..P(A_kC_j) \quad (6)$$

Where A is the set of attributes {A1, A2... … … … … Ak}, in the IDS A, is the values of the set of features that characterize the network traffics.

Eq.(5)is used to classify records A in the test data set or in the online traffic.

Record A € Cj

$$IFF\ P(A\ C_j)P(C_j) > P(A\ C_j)P(C_i), 1 \le i \le k, i \ne j (7)$$

III. PROPOSED HYBRID INTRUSION DETECTION METHOD

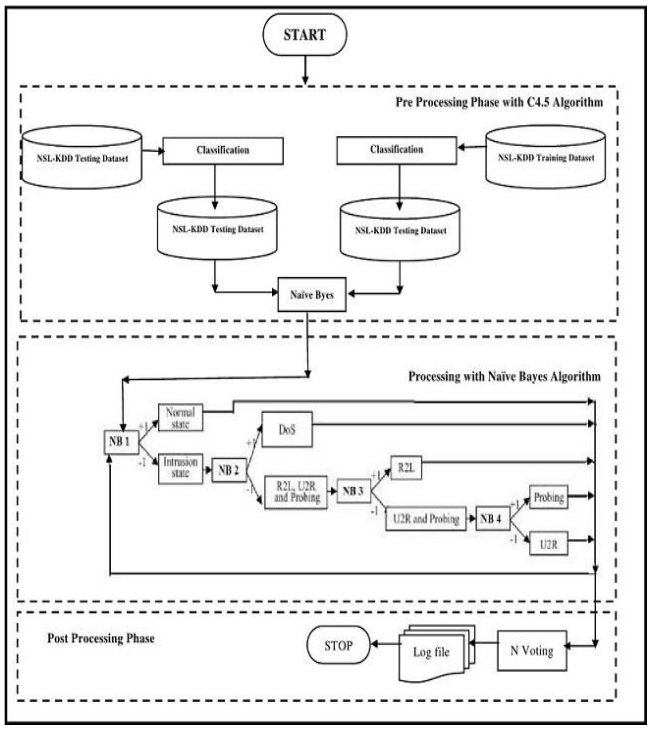

Fig.1 Purposed Intrusion detection system

In this section, Intrusion detection system process, shown in Fig.1, in the purpose Intrusion detection system there are three phases of processing pre-processing phase, the classification phase, and the post-processing stage.

A. Preprocessing Phase

In this phase, the C4.5 algorithm is used to classify the result. In this classification phase consist of two processes is be done i.e. training and testing the dataset. In the training phase, we train the selected machine learning algorithm. Based on this training dataset we classify the records into the normal or abnormal activity. The next phase is the testing phase in this the test the each model with untrained dataset.

The information gain value of each attribute represents the relevance of the attribute to the output class. Parameters of information gain are X and Y, where X defines individual or special features such as number of UCP or ICMP or TCP packets, number of source ports, and Y defines class groups which are Normal data, Probe attack and Denial of Attack(DoS) U2R and R2L. The experimental results from information gain indicate that we have to consider all 12 attribute of the network data for the intrusion classification.

B. Processing with Naïve Bayes algorithm

Multi-Naïve Bayes classifiers are applied to intrusion detection because of multi-types existing in the network. Naïve Bayes classification algorithm needs only k−1 two-class Naïve Bayes classifiers for a case of k classes, while 'One-against-all' Naïve Bayes classification algorithm needs k two-class Naïve Bayes classifiers where each one is trained with all the samples and 'One-against-one' Naïve Bayes classification algorithm needs k (k−1)/2 two-class Naïve Bayes classifiers where each one is trained on data from two classes [5]. Apparently fewer two-class classifiers help to expedite the rate of training and recognition. Thus, 'Binary tree' Naïve Bayes classification algorithm is adopted to construct detection model for intrusion detection. Based on the characteristics of different intrusion detection types, four Naïve Bayes classifiers are developed to identify the five states: normal state (NS) and the four intrusion state or attack state Denial of Service (DoS), Remote to User (R2L), User to Root (U2R), and Probing. With all the training samples of the five states, NB1 is trained to separate the normal state from the intrusion state or attack state. When input of NB1 is a sample representing the normal state, output of NB1 is set to +1; otherwise −1. NB2 is trained to separate the DoS from the other intrusion states. When the input of NB2 is a sample representing DoS, the output of NB2 is set to +1; otherwise −1. NB3 is trained to separate R2L from U2R and Probing. When the input of NB3 is a sample representing the R2L, the output of NB3 is set to +1; otherwise −1. NB4 is trained to separate Probing from U2R. When the input of NB4 is a sample representing Probing, the output of NB4 is set to +1; otherwise −1. Thus, the multilayer Naïve Bayes classifier is obtained. The fundamental principle of intrusion detection model based on C4.5 and Naïve Bayes is shown in fig 1.[5][10]

C. Post processing Phase

The post-processing phase is used to minimize false alarm rate from the result of classification. We propose to use a majority voting algorithm i.e. N voting algorithm for every five consecutive detection results for each pair of IP Addresses (source and destination pair) to determine if the result is normal network activity or an intrusion. To do this, we group the log data from the classification phase into five non-identical records. In each group, if there are at least 3 out of 5 records which are reported to be the same attack type, then this group of data is considered the intrusion. Otherwise, the data is considered as normal network activity. The post-processing or the final phase can help to minimize false alarms from the system. Therefore, the IDS with the post-processing phase can increase the detection accuracy of the IDS.[5]

IV. EXPERIMENTS AND PERFORMANCE EVALUATION

This paper takes the KDD _CUP99 data [8][14] for the experiments. The datasets can be divided into five categories of attack which are denial of service (DoS), unauthorized access from remote machine (Remote to Local, R2L), unauthorized access to local supervisor privileges (User to Root, U2R) and Probe and normal state, Each network record contains 41 attributes or features and in this experiment we use the 13 attribute for the

experiments. This 13 attribute selected by using the information gain (IG) of each attribute. [8][9]

A. Experiment and Result

Table I. Training process for NSL-KDD99

Training process of the proposed hybrid intrusion detection method	
Step 1	Prepare a training data set consisting of normal data and known attack data
Step 2	Build a Classification model using a C4.5 algorithm based on the training data set
Step 3	Decompose the normal training data into subsets according to the C4.5
Step 4	For each normal node of the decision tree, build detection model using the Naïve Bayes classifier based on a normal data subset for the node.

Table II. Testing process for NSL-KDD99

Testing process of the proposed hybrid intrusion detection method	
Step 1	Sense an incoming dataset (NSL-KDD99)
Step 2	Check the dataset trained using the C4.5 is a known attack or unknown attack
Step 3	If the C4.5 classifies the attack as a known attack, then go to Step 6; else go to Step 4
Step 4	Check unknown attack with a trained Naïve Bayes classifier for the corresponding node or attribute, in order to verify if the connection is an unknown attack
Step 5	If the Naïve Bayes classifier detects the attack is Unknown, then; update the Training dataset else, continue
Step 6	Wait for the next incoming attribute

In this section, we selected specimen from the subset of KDD99 dataset to form the training and testing set. There are some accuracy and time complexity indicators for the IDS as follows: True positive (TP), False Positive (FP), True Negative (TN) and False Negative (FN), where TP represents if there is an attack then IDS gives the alarm, FP indicates that the intrusion deportmen is judged as normal, FN indicates that the normal behavior is wrongly thought as intrusion, and TN represents the intrusion is correctly detected.[15]

The final result shows in the fig.2 in that different type of classifiers are used. In this, most of these classifiers haves less accuracy, as well as the time complexity, is more. In this result, the classier Byes Net shows 96.56% accuracy and time complexity is 0.63ms The second classier Nave Byes shows 89.59% accuracy and time complexity is very low i.e. 0.2ms. Third classifier Decision Table shows 98.97% accuracy and time complexity 11.55ms which are very high time complexity. The C4.5 or J48 have high accuracy i.e. 99.55%, and time complexity is 1.55ms. The remaining classifier shows the moderate accuracy as well as the more time complexity. The final result shows the combination of classifiers such as C4.5 and Nave Byes that shows the accuracy is 99.03% and time complexity 0.1ms which are better than all other classifiers.

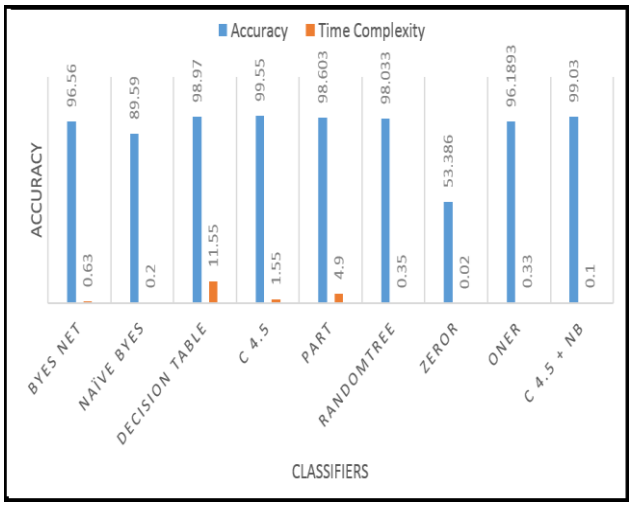

Fig. 2 Comparison of Different Classifiers with C4.5 and Naïve Bayes Classifiers

V. CONCLUSIONS

In this paper, we presented offline intrusion detection system (IDS) model which can be used with existing well-known machine learning algorithms. Our model consists of three phases: the pre-processing phase with C4.5 algorithm, the classification phase with the Naive Byes algorithm, and the post-processing phase in this N-Voting are used to maximize the accuracy level.

We considered and evaluated various machine learning algorithms which are Naive Byes, C4.5 or J48, PART, Random Tree, ZeroR, OneR. The experimental results showed that the Decision Tree i.e. C4.5 and Naive Byes algorithm both gives the higher accuracy and low time complexity. Thus, we developed a new IDS with hybrid machine learning classifiers. As a future work, the proposed IDS can be used in the IDS running phase by installing it on a network to protect this system against real-time attacks.

REFERENCES

[1] Gisung Kim a, Seungmin Lee b, S Kim "A novel hybrid intrusion detection method integrating anomaly detection with misuse detection" (2014) 1690/1700

[2] Mohamed M. Abd-Eldayem "A proposed HTTP service based IDS" Egyptian Informatics Journal _(2014)15, 13/24.

[3] Chih-Fong Tsaia, Yu-Feng Hsub, Chia-Ying Linc, Wei-Yang Lind, “Intrusion detection by machine learning: A review” (2009) 11994 12000

[4] Siva S. Sivatha Sindhu a, S. Geethab, A. Kannan,“Decision tree based lightweight intrusion detection using a wrapper approach” 39 (2012) 129–141

[5] Privet Sangkatsanee a, Naruemon Wattanapongsakorn a, Chalermpol Charn-sripinyo b, “Practical real-time intrusion detection using machine learning approach”(2011) 2227-2235

[6] Hui Lu, Jinhua Xu, "Three-level Hybrid Intrusion Detection System" 978-1-4244-4994-1/09/$25.00 ©2009 IEEE

[7] P. G. Jeya, M. Ravichndran, C.S. Ravichandran, "Efficient Classifier for R2L and U2R Attacks" International Journal of Computer Applications(0975-8887) Vol. No. 45 No-21 May-2012

[8] KDD Cup_99 [online]. Available on: http://kdd.ics.uci.edu/databases/kddcup99/ kddcup99.html, 2007.

[9] Weka 3: Data Mining Software in Java, University of Waikato, New Zealand.[Online]http://www.cs.waikato.ac.nz/ml/weka

[10] Bernhard Pfahringer, "Winning the KDD99 Classification Cup: Bagged Boosting" (2000)

[11] T. J. Parvat, P. Chandra, Arjunwadkar N. M., "Intrusion Detection with Single And Hybrid Machine Learning_Classifiers" 978-93-5107-300-0 Des-14.

[12] M. Shetty, N.M.Shekokar, "Data Mining Techniques for Real-Time Intrusion Detection Systems" April-2012 1ISSN 2229-5518

[13] Su-Yun Wu, Ester Yen., "Data mining-based intrusion detection" 36(2009) 5605-5612

[14] Joong-Hee Leet, Jong-Hyouk Leet, and. Tai-Myoung Chung "Effective Value of Decision Tree with KDD 99 Intrusion Detection Datasets for Intrusion Detection System" ISBN 978-89-5519-136-3 Feb. 17-20, 2008 ICACT 2008

[15] Neminath Hubballi a, Vinoth Suryanarayanan "False alarm minimization techniques in signature-based intrusion detection systems: A survey" 49 (2014) 1–17