Journal of Multidisciplinary Engineering Science and Technology (JMEST)
ISSN: 2458-9403
Vol. 3 Issue 3, March - 2016

# A Layered Approach: Towards Effective Hybrid Intrusion Detection System

**Deepali Gawde**
Department of Computer Engineering
Sinhgad Institute of Technology, Lonavala
Savitribai Phule Pune University, Pune
atharva21deepika@gmail.com

**Thaksen J. Parvat**
Department of Computer Engineering
Sinhgad Institute of Technology, Lonavala
Savitribai Phule Pune University, Pune
pthaksen.sit@sinhgad.edu

***Abstract*—In this paper, a new hybrid intrusion detection system (HIDS) that hierarchically integrates an anomaly detection model and misuse detection model in a decomposition structure is proposed. 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, Naïve Bayes algorithm is created for the decomposed subsets. As a result, each anomaly detection model does not only use the known attack information indirectly, but also builds the profiles of normal behavior very precisely. 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. The experimental results demonstrate that the proposed method is better than the conventional methods in terms of the detection rate for both unknown and known attacks while it maintains accuracy. In addition, the proposed method significantly reduces the high time complexity of the training and testing processes.**

***Keywords—Intrusion Detection, Machine Learning, NSL-KDD dataset, Data Classification***

## I. INTRODUCTION

Today's computer systems need to be designed to prevent illegal access from the outdoors intruders. An unauthorized mechanism designed to access system resources and/or data is called intrusion and designers are called intruders. There are two types of Intruders or malicious activities, Internal Intruders and External Intruders. Internal Intruders attempt to elevate their limited privileges by abusing it. External Intruders attempt to gain unauthorized access to system resources from out-side the target network[1][2].

The vital role of Intrusion Detection Systems (IDSs) is to detect anomalies or malicious activities and attacks in the network and/or for a single host only. The work in the intrusion detection system field can mostly focused on signature-based and misuse-based detection techniques. As a particular application or particular security related portal area Intrusion Detection Systems are designed to protect from the various attacks and computer viruses[4].

However, there are many problems in todays and existing IDSs, such as high false alarm rate, low detection rate over the unknown malicious codes and so on. Many times all the IDSs systems use the signature-based or misuse-based detection techniques to detect the anomalies with all dataset with all attribute. The dataset may be KDD-99, MIT dataset, or new designed ADFA-LD i.e. Linux Dataset is used to detect the anomalies. In this types of datasets there are 41 packet attribute and any traditional IDSs they use the all the attribute to detect the anomalies. Because of this all attribute the performance of the IDS will be low i.e. the time complexity requirement is more[3][5][6].

The reaming paper is organized as following: Section 2 is background and literature review of the IDSs and the IDSs terminologies. Section3 defines the KDD 99 dataset description with all types of categories of attacks in that dataset. Section 4 represents the experiment and result used in this research, while section 5 contains the final conclusion with the feature work[7][8].

## II. BACKGROUND AND LITERATURE REVIEW:

There are two types of classifiers that can be applied to an Intrusion Detection System (IDS). First is Host-based Intrusion Detection System (HIDS) and second is Network-based Intrusion Detection System (NIDS). Host-based systems are used to protect a single host or single system, and to prevent them from malicious activates as well as from the secure on that system. Network-based Intrusion Detection System (NIDS) this type of IDS provide protection by observing network traffic in an attempt to malicious activates[5][6][7].

Intrusion Detection Systems (IDS) can be further differentiating into anomaly-based and/or signature-based. An Anomaly-based IDS detects the malicious activities in the host systems and computer network. The deviation or the unauthorized access from the normal behavior is considered as an attack or disturb that particular system. In an anomaly based IDS detect attacks or malicious activities by comparing the new traffic with the already existing database. Signature based detection system matches the signatures of already known malicious activities that are stored into the database to detect the malicious activities in the host system.

The intrusion detection evaluation of any problem and with its solution usually affects the choice of the suitable intrusion detection system for a particular environment depending on different factors. The false alarm rate (FAR) and the detection rate is calculated from the four instances in the intrusion detection system i.e.) False Positive (FP), False Negative (FN), True Positive (TP) and True Negative (TN). The tradeoff between these two factors (false alarm rate and the detection rate) has been analyzed with the help of the one curve i.e. Receiver Operating Characteristic (ROC) curve[8][9][10].

| | | Predicted | | Total |
|---|---|---|---|---|
| | | Normal | Attacks | |
| Actual | Normal | TN | FP | TN+FP |
| | Attacks | FN | TP | TP+FN |
| Total | | TN+FN | TP+FP | |

*Table No. 1 IDS Confusion Matrix*

There are four classes True Positive (TP), True Negative (TN), False Positive (FP) and False Negative (FN) are counted as predicted and actual classes. They are merged into in the 2x2 confusion matrix as shown in Table 1. Show that there are two columns of “Normal” and “Attack”. Here True Positive (TP) means a legitimate attack or malicious activities which trigger IDS to produce an alarm. True Negative (TN) An event when no attack at that time no detection is made. False Positive (FP) an IDS to produce an alarm when no attack has taken place. False Negative (FN) there is no alarm is raised when an attack has been done. In the machine learning algorithms, there are low false alarm rate as compare to the other IDS systems.

KDD 99 dataset description:

The KDD Cup 99 dataset has been mostly used till 1999 for the detection of the abnormal behavioral in the network or the single host. In this experiment, we use the KDD 99 with 20% dataset in that there are approximately 25192 records with the 41 attributed dataset. For each connection, there are 41 attributes to specify the particular packet is normal or abnormal[1][5].

KDD Cup 99 intrusion detection dataset which are based on the DARPA 98 dataset for researcher for the detection of the intrusion. This dataset is publicly available to the researcher. In this KDD Cup 99 dataset there are 21 different types of malicious activities such as back, buffer_overflow, ftp_write, guss_password, imap, ipsweep, land, loadmodule, multihop, nmap, neptune, phf, pod, portsweep, rootkit, satan, smuf, spy, teardrop, warezwclient, warezmaster in this dataset. And this all attack will be dividing into to four basic characteristics they are Denial of Service (DOS), Remote to Local (R2L), User to Root (U2R), Probe.

In this dataset, the simulated attacks fall in one of the following four types of categories[11][12][13]:

- **Denial of Service (DOS):** Using some services attacker tries to prevent ligaments users
- **Remote to Local (R2L):** On the victim machine there is no attacker's account but tries to access that system.
- **User to Root (U2R):** On the victim machine there is attackers account but tries to access that system with gain super user or administrator privileges.
- **Probe:** Attacker can be access the gain information from the target host.

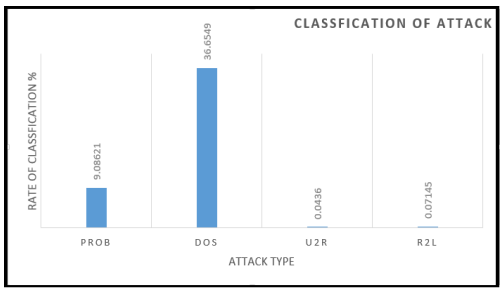


*Fig.1. Classification of Attack*

In the figure 1. Shows the classification of the all the attacks into the four basic attacks. In the KDD Cup 99 20% dataset there are 9.08621% of attacks are Probs, 36.6549% of attacks are DOS, 0.0436% attacks are U2R, and 0.07145 of attacks are R2L. The normal records in the dataset is 54.1441% present in the total dataset.

## III. PROPOSED HYBRID INTRUSION DETECTION METHOD

In this section, the DT i.e. C4.5 and NB Naïve Bayes (NB) algorithms that are required in order to build the issue detection model and anomaly detection model, respectively, are briefly introduced. Then, the integration of these models is explained and the properties of the proposed hybrid intrusion detection system(HIDS) are discussed[1][2][3].

### *A. Decision tree and C4.5*

A decision tree (DT) is one of the most widely used classification algorithms in data mining. It operates in a divide and conquer (D & C) manner, which recursively partitions the training data set based on its attributes until the stopping conditions are satisfied. The C4.5 consists of nodes, edges, and leaves. The 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 dividing process and the node is labeled as following the class label of the data set. This labeled node is called a leaf node. In this way, the C4.5 recursively partitions the training data set, which creates a tree-like structure. 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 gain is calculated using the following formula[4][5][6]:

$$IG(S,A) = Entropy(j) - \sum_{i=1}^{n} f_s(A_i) * Entropy(S_A) \qquad (1)$$

where Gain(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 SAi is a subset of S containing all items where the value of A is Ai. Here, the entropy is obtained as follows:

$$Entropy(S) = \sum_{j=1}^{m} F_s(j) * \log_2 F_s(j) \qquad (2)$$

Where m is the number of different values of the attribute 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 certain decomposed regions become homogeneous. Then, C4.5 performs the final pruning step. This step 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 regions as the classes of major data belonging to each decomposed region.

### *B. 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 Naive Byes use a probabilistic graph model. The structure of the Naive Byes typically described into directed acyclic graph (DAG). In the Naive Byes, 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.

$$P(Cj)=P(XCj)P(Cj)$$

$$IF(CjX)>(CiX),1\leq i\leq m,i\neq j \qquad (3)$$

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].

$$(Cj)=SjS(4)$$

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

$$(ACj)=(A1Cj)P(A2Cj)\ldots\ldots\ldots..P(AkCj) \qquad (5)$$

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

$$IF(ACj)(Cj)>P(ACj)P(Ci),1\leq i\leq k,i\neq j(6)$$

## IV. PROPOSED HYBRID INTRUSION DETECTION METHOD

The proposed HIDS is as follows. First, C 4.5 model based on the training data set is built. It is well known that a misuse detection model can detect known attacks with a small false positive rate, while it cannot detect unknown attacks well. Because the false positive rate of the C4.5 model is low, the results of the attack detections of the C4.5 model are followed.

Then, Naive Byes algorithm is trained for each normal training data set, which is decomposed by the C4.5 model. The primary reason for decomposing the normal training data set is that the anomaly detection models in the previous hybrid intrusion detection systems have attempted to profile the normal connection patterns using one outlier detection model. However, in reality, there are various normal patterns according to the protocol type (TCP, UDP, ICMP, etc.), service type (HTTP, FTP, SNMP, etc.), and so on.

Although a Naive Byes model is appropriate for creating a nonlinear decision boundary, the Naive Byes model can be very sensitive to the training data set and can increase the false positive rate. In order to alleviate this problem, the normal data set is decomposed into smaller subsets and then Naive Byes model is built for the decomposed subsets. Because the data patterns of each decomposed subset are less complex than those of the whole data set, multiple models for each decomposed data pattern can be less flexible than a single model for the whole data pattern. The training process is described in Table 1.

The training time and testing time of the anomaly detection model are also improved using the tree decomposition. The time complexity of training a 1-class SVM model is in the order of n2, where n is the number of training instances. If each decomposed problem handles r training instances, then the complexity of solving the entire problem is in the order of (n/r) r2 = nr, which is significantly smaller than n2. Although the distribution of the number of data instances in each decomposed region is not uniform, the decomposition can reduce the training time. When it is considered that the training time is a cost for updating the detection model, reducing the training time provides more opportunities to update the detection model. The testing process and diagram of the proposed method are described in Table 2 and Fig. 3, respectively.

The testing time is directly related to the intrusion detection performance. Anomaly detection systems are often designed for offline analyses due to the expensive processing and memory overheads. Hence, the time and memory complexity of the anomaly detection model for testing should be minimized in order to operate the detection model in real time. In this paper, the decomposition concept can contribute to reducing the testing time of the anomaly detection model in real time operations. When a Naive Byes model classifies the test instances, the most time consuming work is the computation of a decision function. The complexity of the decision function is measured using the number of encountered support vectors because the number of support vectors dominates the complexity of the computation. In the proposed method, the test instances are classified by one of the Naive Byes models in a decomposed region[14].

## V. EXPERIMENTS AND RESULTS

In this section, the effectiveness of the proposed method is evaluated carefully through experiments using the NSL-KDD data set, which is a modified version of well-known KDD'99 data set. Because the KDD'99 data set has an inherent problem of a number of redundant instances existing in the training and testing data set, NSL-KDD data set was proposed by removing all redundant instances and reconstituting the data set, which makes it more efficient to have an accurate evaluation of different learning techniques are used to evaluate the performance of the proposed method.

In this paper, the evaluation data set is organized by modifying the KDDTrain+.TXT and KDDTest+.TXT documents in the NSL-KDD data set. Each of these documents consists of traffic records including information of the traffic features and a connection label. Because the connection label specifies the attack type and KDDTest+.TXT contains some attack types that do not exist inKDDTrain+.TXT, it is possible to categorize the attack records in KDDTest+.TXT into known attacks and unknown attacks. However, even though the attack labels are the same, the traffic characteristics between KDDTrain+.TXT and KDDTest+.TXT are not sufficiently similar to satisfy the requirement for known attacks in the proposed context. Hence, in order to clearly distinguish between the known attacks and unknown attacks, the training data set and test data set were organized as follows.

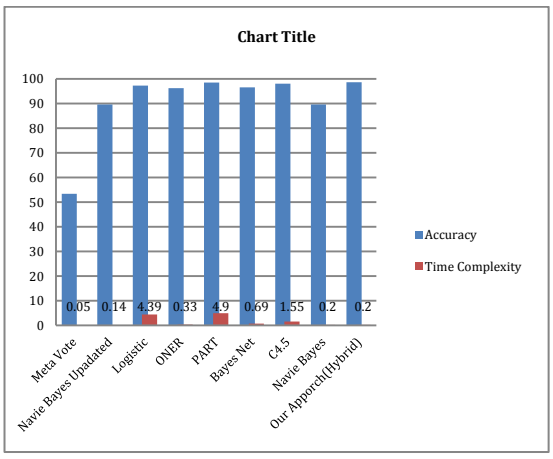


*Fig. 2.Result Comparison*

This experiment demonstrates that the proposed hybrid intrusion detection method is better than the conventional methods in terms of detection performance, training time, and testing time. It should be noted that the proposed method does not require an additional overhead in order to integrate the detection models. It is suggested that the training data set is decomposed before performing the anomaly detection. This decomposition is appropriate when using multi-core/parallel/distributed computing for further increases in speed. In future research, a specific decision tree algorithm that is more appropriate to the proposed hybrid intrusion detection method will be developed. Because the original C4.5 decision tree does not consider clusters in the normal data set, it can divide a well-formed normal cluster during the decomposition processes. This can hinder the well-formed normal cluster that belongs to a single decision boundary in the NB model, which can degrade the profiling ability. In addition, the uneven distribution of data instances impedes the reduction of the training time and testing time. Note that approximately 98% of the training data belonged to subset. Hence, future research will focus on modifying the C4.5 decision tree algorithm to improve the supplement points as mentioned above without losing its ability to detect known attacks.

## VI. CONCLUSION

In this research, a new HIDS that hierarchically integrates a misuse detection model and an anomaly detection model in a decomposition structure was proposed. First, the C4.5 decision tree (DT) was used to create the misuse detection model that is used to decompose the normal training data into smaller subsets. Then, the Naïve Bayes (NB) was used to create an anomaly detection model in each decomposed subset. Throughout the integration, the anomaly detection model can indirectly use the known attack information to enhance its ability when building profiles of normal behavior. The experiments demonstrated that the proposed HIDS could improve the IDS in terms of detection performance for unknown attacks and detection speed. Hence, future work on this research issue will focus on real time HIDS with enhanced attributes.

## REFERENCES

[1] J. Neumann, The Computer and the Brain, Yale University Press, New Haven, CT, USA, 1958.

[2] Michał Woz´niak a, Manuel Grana b, Emilio Corchado, A survey of multiple classifier systems as hybrid systems, Information Fusion 16 (2014) 3–17

[3] A. Newell, Intellectual issues in the history of artificial intelligence, in: F. Machlup, U. Mansfield (Eds.), The Study of Information: Interdisciplinary Messages, John Wiley & Sons Inc., New York, NY, USA, 1983, pp. 187–294.

[4] G.V. Nadiammai, M. Hemalatha, Effective approach toward Intrusion Detection System using data mining techniques, Egyptian Informatics Journal (2014) 15, 37–50

[5] D. Wolpert, The supervised learning no-free-lunch theorems, in: Proceedings of the 6th Online World Conference on Soft Computing in Industrial

[6] Applications, 2001, pp. 25–42.

[7] C.K. Chow, Statistical independence and threshold functions, IEEE Transactions on Electronic Computers EC-14 (1) (1965) 66–68.

[8] L. Shapley, B. Grofman, Optimizing group judgmental accuracy in the presence of interdependencies, Public Choice 43 (3) (1984) 329–333.

[9] B.V. Dasarathy, B.V. Sheela, A composite classifier system design: concepts and methodology, Proceedings of the IEEE 67 (5) (1979) 708–713.

[10] L. Rastrigin, R.H. Erenstein, Method of Collective Recognition, Energoizdat, Moscow, 1981.

[11] L. Hansen, P. Salamon, Neural network ensembles, IEEE Transactions on Pattern Analysis and Machine Intelligence 12 (10) (1990) 993–1001, http://dx.doi.org/10.1109/34.58871.

[12] L. Xu, A. Krzyzak, C. Suen, Methods of combining multiple classifiers and their applications to handwriting recognition, IEEE Transactions on Systems, Manand Cybernetics 22 (3) (1992) 418–435.

[13] K. Tumer, J. Ghosh, Analysis of decision boundaries in linearly combined neural classifiers, Pattern Recognition 29 (2) (1996) 341–348.

[14] T. Ho, J.J. Hull, S. Srihari, Decision combination in multiple classifier systems, IEEE Transactions on Pattern Analysis and Machine Intelligence 16 (1) (1994)66–75.