A ROLE OF DATA MINING ANALYSIS TO IDENTIFY SUSPICIOUS ACTIVITY ALERT SYSTEM

Anil Lamba[1], Satinderjeet Singh[2], Natasha Dutta[3], Sivakumar Sai Rela Muni[4]
[1,2,3,4]Department of Computer Science, Charisma University, Turks and Caicos Islands

Abstract: With the tremendous growth of the usage of computers over network and development in application running on various platforms captures the attention toward network security. This paradigm exploits security vulnerabilities on all computer systems that are technically difficult and expensive to solve. Hence intrusion is used as a key to compromise the integrity, availability and confidentiality of a computer resource. The Intrusion Detection System (IDS) plays a vital role in detecting anomalies and attacks in the network. In this work, data mining concept is integrated with an IDS to identify the relevant, hidden data of interest for the user effectively and with less execution time. Four issues such as Classification of Data, High Level of Human Interaction, Lack of Labeled Data, and Effectiveness of Distributed Denial of Service Attack are being solved using the proposed algorithms like EDADT algorithm, Hybrid IDS model, Semi Supervised Approach and Varying HOPERAA Algorithm respectively. Our proposed algorithm has been tested using KDD Cup dataset. All the proposed algorithm shows better accuracy and reduced false alarm rate when compared with existing algorithms.

KEYWORDS: Anomaly based algorithm; Classification algorithms; Data communication; Denial of service attack; Intrusion detection; Cyber Security; Cloud Security; Network ; Cyber; Cyber Threats; Threat Analysis ; Information Security; Data security.

Citation: Anil Lamba, 2014."A ROLE OF DATA MINING ANALYSIS TO IDENTIFY SUSPICIOUS ACTIVITY ALERT SYSTEM", International Journal for Technological Research in Engineering, Volume 2 Issue 3, pp.5814-5825, 2347-4718

I. INTRODUCTION

In this modern world intrusion occurs in a fraction of seconds. Intruders cleverly use the modified version of command and thereby erasing their footprints in audit and log files. Successful IDS intellectually differentiate both intrusive and nonintrusive records. IDS was first introduced by James Anderson in the year 1980 [1]. Most of the existing systems have security breaches that make them easily vulnerable and could not be solved. Moreover substantial research has been going on intrusion detection technology which is still considered as immature and not a perfect tool against intrusion. It has also become a most priority and challenging tasks for network administrators and security experts. So it cannot be replaced by more secure systems. Data mining based IDS can efficiently identify these data of user interest and also predicts the results that can be utilized in the future. Data mining or knowledge discovery in databases has gained a great deal of attention in IT industry as well as in the society. Data mining has been involved to analyze the useful information from large volumes of data that are noisy, fuzzy and dynamic.

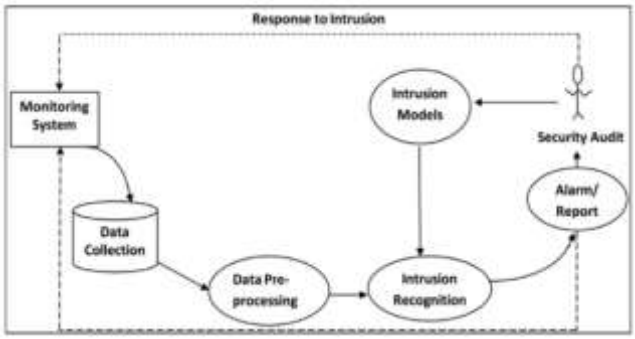

Figure 1 Overall structure of Intrusion Detection System.

Fig. 1 illustrates the overall architecture of IDS. It has been placed centrally to capture all the incoming packets that are transmitted over the network. Data are collected and send for preprocessing to remove the noise; irrelevant and missing attributes are replaced. Then the preprocessed data are analyzed and classified according to their severity measures. If the record is normal, then it does not require any more change or else it send for report generation to raise alarms. Based on the state of the data, alarms are raised to make the administrator to handle the situation in advance. The attack is modeled so as to enable the classification of network data. All the above process continues as soon as the transmission starts.

Ektefa [2] compared C4.5 & SVM to show the performance of both algorithm and FAR values too. Among these two C4.5 works better compared to other. Since the performances of a classifier are often evaluated by an error rate and it does not suit the complex real problems, in particular multiclass. Holden [3] has proposed hybrid PSO algorithm that can deal with nominal attributes without going for the both conversion and nominal attribute values. To overcome the drawback (features) that the PSO/ACO algorithm lacks. The proposed method shows simple rule set efficiently to increase in accuracy. Likewise Ardjani [4] applied SVM with PSO as (PSOSVM) to optimize the performance of SVM. 10Fold cross validation is done to estimate the accuracy. It utilizes the advantage of minimum structural risk with global optimizing features. The result shows better accuracy with high execution time. Since there [5] is an

existence of multidimensional data set, it is necessary to extract the features and also to remove the redundant and inconsistent features that affects classification. Based on this, information gain and genetic algorithm has been combined to select the significant features. This method shows better accuracy when features are selected than individually applied. Panda [6] uses two class classification method in terms of normal or attack. The combination of J48 and RBF shows more error prone and RMSE rate. Compared to this, Nested Dichotomies and random forest method show 0.06% error with a 99% detection rate. Petrussenko Denis [7] involves LERAD to detect attacks in network packets and TCP flow to capture a new form of attack patterns. He found LERAD performs well using a DARPA data set with high detection rate. In [8] Mahoney used anomaly detection methods like PHAD, ALAD, LERAD to model the application layer, data link layer and so on. Out of that LERAD performs well. SNORT [9] is tested on IDEVAL data set and attacks are tabled daily nearly for one week. SNORT, SNORT with PHAD, SNORT with NETAD and SNORT is also combined with PHAD and NETAD as a preprocessor. As a result, SNORT + PHAD + NETAD detects nearly 146 attacks from 201 attacks respectively.

Unsupervised method [10] uses a huge set of data as pre-labeled training data and produces less accuracy. To overcome this issue, a semi supervised algorithm is used. Fuzzy Connectedness based Clustering [11] approach is evaluated using both Euclidean distance and statistical properties of clusters. It facilitates the discovery of any shape and detects not only known but also its variants. ChingHao et al. [12] proposed a co-training framework to leverage unlabeled data to improve intrusion detection. This framework provides lower error rate than single view method and thereby incorporating an active learning method to enhance the performance. In [13] the semi supervised learning mechanism is used to build an alter filter to reduce the false alarm ratio and provides high detection rate. Where the features of both supervised and semi supervised learning are same in nature.

A TriTraining SVM algorithm is used to improve the accuracy rate and speed [14]. Monowar H. Bhuyan [15] present a tree based clustering technique to find clusters among intrusion detection data set without using any labeled data. The data set can be labeled using cluster labeling technique based on a Tree CLUS algorithm. It works faster for the numeric and a mixed category of network data.

Partially Observable Markov Decision Process [16] involves the combination of both misuse and anomaly detection to determine the cost function. Semi Supervised strategy is applied to three different SVM classifiers and to the same PSVM classifiers.

Distributed Denial of Service Attacks [17] has been achieved tremendous growth in recent years since it is difficult to solve. It has been obtained through two ways. First by filtering the legitimate traffic from malicious traffic and second bydegrading the performance of legitimate traffic gradually. Denial of capability attack is one of the major causes for the existence of DDoS attacks. DDoS attacks can be prevented by denial of the capability approach by Sink tree model [18] representing quota assigned to each domain on the network.

Distributed Denial of Service Attacks are not only suited for the specified target machine but also compromises the whole network. Based on this perspective proactive algorithm has proposed by Zhang et al. [19]. The network is divided into a set of clusters. Packets need permission to enter, exit or pass through other clusters. High speed traffic [20]is monitored by IP prefix based aggregation method to detect DDoS related anomalies facilitated in a streaming fashion. In [21]the author presents the synchronous communication based on acknowledgment based and in the presence of fixed clock drift. The client clock relates to that of the server. But any acknowledgment loss and faster or slower clock drift of client enables the adversary to cause a direct attack on it.

Preprocessing of network data [22]consumes time and hard for network administrator to solve. Even though the preprocessing is made successfully, Classification of Data and labeling of unlabeled data seems to be a challenging task. Based on this, four issues such as Classification of Data, High Level of Preprocessing, Semi Supervised Approach and Mitigating Distributed Denial of Service Attack has been solved using proposed approaches that help IDS to efficiently identify the attacks accurately with reduced false alarm rate.

This paper is organized as follows. Section 2includes the motivation of the work. Section 3describes the problem statement. Section 4explains the methodology of the proposed work. Section 5describes the data set used and its features in detail. Section 6includes the details of performance evaluation based on the experimental study. Section 7refers to conclusion and future enhancement.

II. MOTIVATION

This research focuses on solving the issues in intrusion detection communities that can help the administrator to make preprocessing, classification, labeling of data and to mitigate the outcome of Distributed Denial of Service Attacks. Since, the network administrator feels difficult to preprocess the data. Due to the overwhelming growth of attacks which makes the task hard, attacks can be identified only after it happens. To overcome this situation, frequent updating of profiles is needed. Reduced workload of administrator increases the detection of attacks. Data mining includes many different algorithms to accomplish the desired tasks. All of these algorithms aims to fit a model to the prescribed data and even analyzes the data and simulate a model which is closest to the data being analyzed.

III. PROBLEM STATEMENT

Data mining approaches have been implemented by many authors to solve the detection problem. This implies that we are close to the solution. Since pattern signature approach is currently utilized only by network administrators. The fact is that the existing works deal with the subset of problem that are needed for achieving intrusion detection and not others.

To solve the above issues the following solutions were made,

To solve the problem of Classification of Data, an enhanced data adapted decision tree algorithm is proposed. This algorithm works different normal decision tree algorithm. It efficiently classifies the data into normal and attack without any misclassification.

To minimize the workload of network administrator, SNORT is combined with anomaly based approaches. Using this technique, a Hybrid IDS model is proposed. This technique automatically classifies the data based on the predefined rules within it. The problem of implementing supervised and unsupervised method can be solved by using Semi Supervised Approach where with small amount of labeled data, the large amount of unlabeled data can be labeled.

Distributed Denial of Service Attack can be greatly reduced using varying clock drift, with the help of varying clock drift in network based application, the adversary finds difficult to access the port that has been used by the legitimate client. At the same time, any client can communicate with the server for longer time intervals without any interruption. Hence the proposed approaches address the issues and efficiently identifies any kind of attack. All these four solutions have been discussed in the following section in detail.

IV. METHODOLOGY

Some of the open issues have been taken to detect attacks over the network. To achieve this, the framework of the proposed methods has been discussed below. The entire framework of the proposed methodology in Intrusion Detection System is described in Fig. 2.

4.1. Framework of proposed EDADT (Efficient Data Adapted Decision Tree) algorithm

The pseudo code of the proposed EDADT algorithm shown in Fig. 3 utilizesthe hybrid PSO technique to identify the localand global best values for n number of iterations to obtain the optimal solution.

The best solution is obtained by calculating the average value and by finding the exact efficient features from the given training data set. For each attribute a, select all unique values of a to find the unique values belong to the same class label.

If n unique values belong to the same class label, split them into m intervals, and m must be less than n. If the unique values belong to different c class label, check whether the probability of the value belongs to same class.

If it is found then change the class label of values with the class label of highest probability. Split the unique values as c interval then repeat checking of unique values in the class label for all values in the data set. Find out the normalized information gain for each attribute and decision node forms a best attribute with the highest normalized information gain. Sublists are generated using best attributes and those nodes forms the child nodes. These processes continue until the data set converges. At last, train the EDADT model.

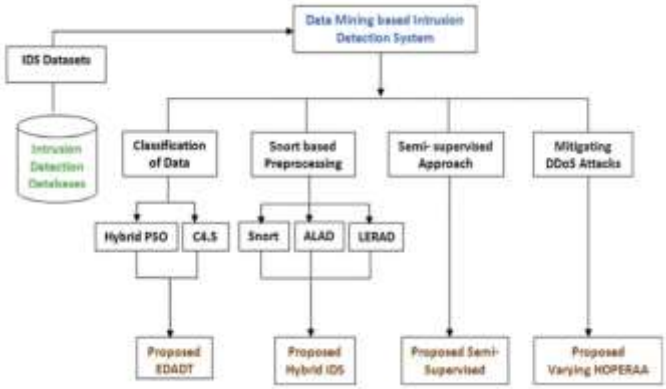

Figure 2 Entire framework of proposed methodology in Intrusion Detection System.

Finding best attributes using information gain
d=dataset
att =attribute
n= set of unique values
m= regular _intervals
$c_{1,2,\ldots,p}$= class label, where c_1, c_2, c_p are same, different class label & child node
d_n= decision node
b_a = best attribute
if ($n \varepsilon c_1$) **then**
split m
else if ($n \varepsilon c_2$)
Highest highest probability ε c
c= c (highest probability)
split c
update split m
m<n
end
until termination condition is met
end
d_n = att+ highest normalized information gain
recurse b_a
C_p=b_a
Repeat
until all b_a found
end

Figure 3 Pseudocode of proposed EDADT algorithm.

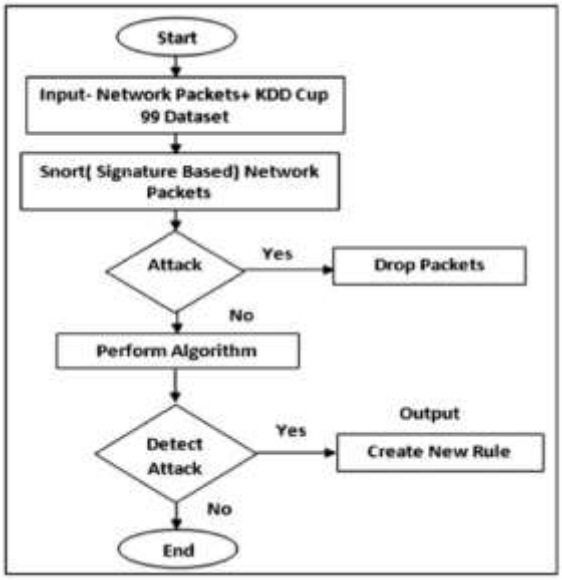

Figure 4 Framework of proposed Hybrid Intrusion Detection System.

4.2. Framework of proposed Hybrid Intrusion Detection System

SNORT shown in Fig. 4is installed to capture the network packets in real time and also KDD Cup 99 dataset is used. SNORT is a signature based method because it detects the attack based on the set of rules that are predefined within the SNORT. If any attack data is found, it automatically drops the packet otherwise the particular record is considered as a normal one. Since SNORT detects only profile based attacks some of the anomaly based approaches such as Packet Header Anomaly Detection, Network Traffic Anomaly Detector, Application Layer Anomaly Detector, Learning Rules for Anomaly Detection have been used to perform better prediction. Based on the new attack, it is updated in the profile as a new rule. By using this approach any kind of attacks can also be found.

4.3. Framework of proposed Semi Supervised Approach

In this approach, the dataset is divided into training and testing data. First, training data includes both the labeled data and unlabeled data. Using the labeled data the unlabeled data can be labeled. Hence this kind of approach is said to be Semi Supervised Approach. The labeled training data are applied to the SVM classifier and the model is generated. Then, change the SVM parameters by applying the Radial Basis Function kernel function and generate the model for each tuning process. Apply the training unlabeled data to SVM model as test data and results are generated for all models. Check majority voting for all models. Drop records which do not satisfy the voting results. Include the changed label as predicted labels. Randomly generate 1000 data points to find the vector distance between each support vector and the data points. This process enables the most confidential data. Provide these new data instances with the trained labeled data. At last, include the unlabeled test data to the classifier and check the accuracy rate and its corresponding false alarm rate respectively. This approach has been successfully tested using KDD Cup 99 data set and the results obtained have been compared with the existing algorithms as shown in Fig. 5.

4.4. Framework of proposed varying HOPERAA (Hopping Period Alignment and Adjustment) algorithm

To mitigate the Effectiveness of Distributed Denial of Service Attack Varying HOPERAA Algorithm is proposed. Based on the previous work done by Zhang Fu, et al. [22], in our approach, a variable clock drift method is proposed to avoid the client waiting time for server and at the same time message loss is avoided greatly. It includes three aspects,

Contact initiation part. Data transmission part.Varying HOPERAA Algorithm.

4.4.1. Contact initiation part

This activity takes place on the client side to initiate a request to the server for connection and further communication. The server divides the range of port numbers into intervals and ports are evenly split for every interval and these ports changes every s time unit. Port sequence is only known by the client and server which is independent from other clients. Suppose if any message is lost then the port remains open which can be disabled by the adversary and starts accessing the server by pretending as the legitimate client. To overcome this issue, the pseudorandom function and index value can be issued by the server to intimate the client to choose the next set of ports through this the bandwidth would be maintained.

Once the server receives the contact initiation message it sends the varying clock value of the client and the server's clock value is t1 . . . tn which denotes the arriving time of the same message at different rates. Then,

Vhc ðt1Þ ¼ flow rate ðLÞ; medium rate ðMÞ; high rate ðHÞg ð1Þ

It would be stored by the client to estimate the variable clock drift. The client waits for the acknowledgment from the server for a specific period of time say 2l + L. After that, it chooses another interval and starts sending messages. It may take any a number of trials to get access to the server. In our proposed algorithm, the number of trails made by the client in contact initiation part has been minimized so as to improve the reliability of the application.

Once the message is received, the server waits for the port to open. As soon as the port is open it sends the acknowledgment with pseudorandom function, index value and varying time and t1, t2 to the client.

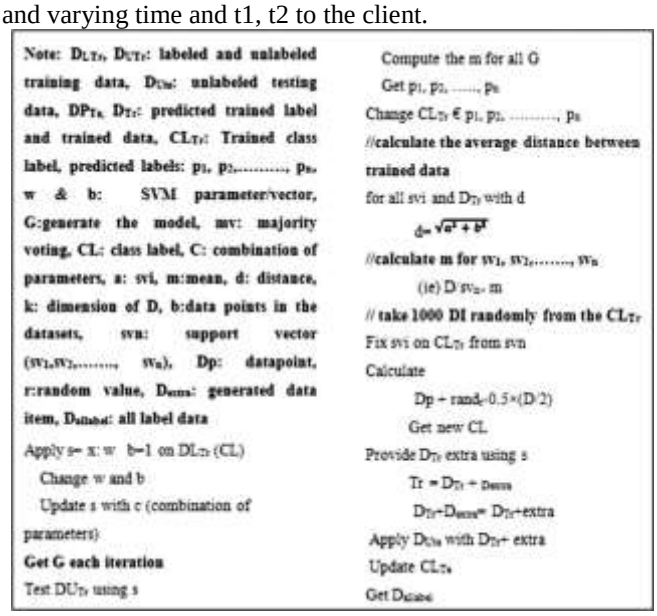

Figure 5 Pseudocode of SemiSupervised Approach

4.4.2. Data transmission part

In this part, the client starts to send messages to sever s. Once the client receives a reply from the client during contact initiation part it got the port hopping sequence. According to the state of the message the port sequence varies. The opening time of the port is L + l time unit where L is greater than l. As soon as the port pi opens when it reaches ltime units, the new subsequent port pi + 1 opens simultaneously like that the port grows according to the message length. In our algorithm the message delivery latency has been improved for better performance. The timer

at client c will be placed zero when it receives the reply message from the server.

4.4.3. Varying HOPERAA Algorithm

The Varying HOPERAA Algorithm has been illustrated using Fig. 6 in which Lc does not drift apart from the server. Since the varying clock drift is maintained depending on the length of the message. In this section, client c has a variable clock drift Vhc related to the server. The server maintains threshold value to estimate the nature of the message. If the client sends data continuously it has taken as high rate, if messages send moderately it will be taken as medium rate and not frequently then it is taken as low rate. The server keeps the part of a HOPERAA algorithm to estimate the clock drift and also evaluates the state of message consequently. At the maximum the client runs Varying HOPERAA one time to estimate the clock drift but in case of fixed clock drift the client runs HOPERAA over three times to know the clock drift is slower or faster than the server. Through the proposed Varying HOPERAA Algorithm growth the growth of intervals is extremely reduced.

V. DATA SET DESCRIPTION

Attacks can be described as

Dos attack – It is a kind of attack where the attacker makes processing time of the resources and memory busy so as to avoid legitimate user from accessing those resources.

U2R attack – Here the attacker sniffs the password or makes some kind of attack to access the particular host in a network as a legitimate user. They can even promote some vulnerability to gain the root access of the system.

R2L attack – Here the attacker sends a message to the host in a network over remote system and makes some vulnerability.

Probe attack – Attacker will scan the network to gather information and would make some violation in the future.

KDD Cup 99 data set [23] contains 23 attack types and their names are shown in Table 1 and its features are grouped as,

1. Basic features

It encompasses all the attributes of TCP/IP connection and leads to delay in detection.

2. Traffic features

It is evaluated in accordance with window interval & two features as same host and same service.

(a) Same host feature

It examines the number of connections for the past 2 s that too from the same destination host. In other words, the probability of connections will be done in a specific time interval.

(b) Same service feature

It examines the number of connections in a particular time interval that too posses same service.

3. Content features

Dos & probe attack have frequent intrusion sequential patterns than the R2L & U2R. Because these two attacks include many connections to several hosts at a particular time period whereas R2L and U2R perform only a single connection. To detect these types of attacks, domain knowledge is important to access the data portion of the TCP packets. Ex. Failed login, etc. these features are called as content features.

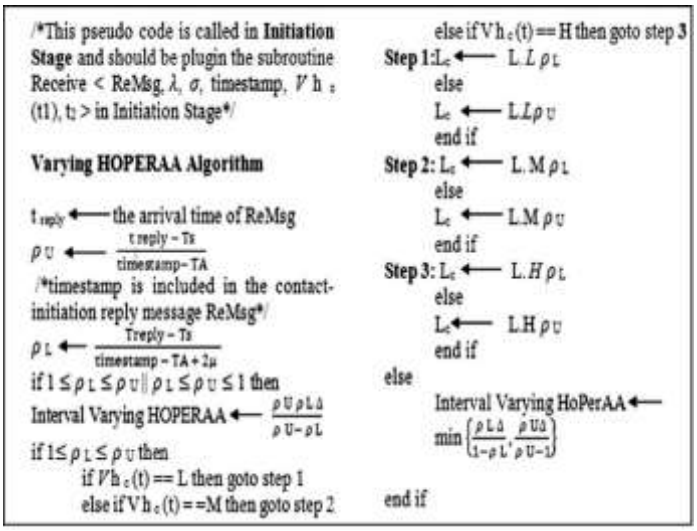

Figure 6 Pseudocode of Varying HOPERAA Algorithm.

Table 1 Name of the attacks classified under 4 groups.

Denial of service	Back, land, neptune, pod, smurf, teardrop
Probes	Satan, ipsweep, nmap, port sweep
Remote to local	ftp_write, imap, guess_passwd, phf, spy, warezclient, multihop, warezmaster
User to root	Buffer_overflow, load module, Perl, root kit

VI. RESULTS AND DISCUSSIONS

KDD Cup 99 data set has been used in this research of which 60% is treated as training data and 40% is considered as testing data. The proposed framework has been implemented in MatLab10 and Java using data mining techniques. Performance of four proposed methods such as, below, has been evaluated in terms of accuracy and FAR values as shown below.

- Classification of network data using EDADT algorithm.
- Proposed Hybrid IDS.
- Performance of SemiSupervised Approach for IDS and,
- Mitigating DDoS attacks using Varying Clock Drift Mechanism.

Table 1 Name of the attacks classified under 4 groups.

Denial of service	Back, land, neptune, pod, smurf, teardrop
Probes	Satan, ipsweep, nmap, port sweep
Remote to local	ftp_write, imap, guess_passwd, phf, spy, warezclient, multihop,

	warezmaster
User to root	Buffer_overflow, load module, Perl, root kit

Table 2 Reduced feature set for each attack after discriminate analysis.

Attack name	Related features
Back	27, 28, 31, 36, 38, 39, 40, 41
Buffer overflow	14, 16, 36, 37
ftp_write	1, 5, 6, 9, 10, 12, 13, 16, 17, 19, 22, 31, 34, 35, 36, 37
guess_pwd	27, 28, 31, 38, 39, 40, 41
Imap	8, 12, 24, 25, 28, 38
ipsweep	5, 27, 28, 31, 34, 35, 36, 37, 40, 41
Land	1, 6, 15, 17, 34, 35, 38, 39
load module	12, 23, 25, 36, 38, 40, 41
multihop	1, 5, 6, 10, 12, 17, 22, 31, 34, 35, 36, 40
Neptune	29, 30, 34, 35
Nmap	5, 25, 26, 34, 35, 36, 38, 39
Normal	31, 34, 35, 36, 37
Perl	1, 5, 34
Phf	5, 10, 15, 24
Pod	34, 35, 36
port sweep	1, 25, 26, 27, 28, 29, 30, 34, 35, 36, 38, 39, 40, 41
root kit	9, 10, 11, 12, 13, 16, 17, 341
Satan	1, 6, 12, 27, 29, 30, 34, 35, 36, 39, 40
Smurf	1, 5, 6, 12, 37
Spy	12, 15, 17, 18, 19, 34, 39
teardrop	5, 26, 30, 34
warezclient	5, 15, 28
warezmaster	1, 34

6.1. Classification of network data using EDADT algorithm

Data mining technology to Intrusion Detection Systems can mine the features of new and unknown attacks well, which is a maximal help to the dynamic defense of Intrusion Detection System. This work is performed using Machine learning tool with 5000 records of KDD Cup 99 data set to analyze the effectiveness between our proposed method and the traditional algorithms. The performance of the various algorithms measured in terms of accuracy, Sensitivity, Specificity and false alarm rate. Table 2represents the related features for a particular attack. For all 23 attacks, the related features are calculated by enabling the threshold value. If the attribute satisfies the specified constraints then the attribute is chosen as the related features of particular attack.

Table 3 representsthe rule structure for the KDD Cup 99data set. Using this rule structure the data set can be easily classified in the future. If any new type of attack is found it can also be added in the in this profile for better classification results. Fig. 7represents the overview of EDADT algorithm. The advantage of using this algorithm is that it splits the record into the attack or normal without leaving any data as unclassified. Moreover the data set can be classified as normal or attack record efficiently.

Since many data mining algorithms are available this is an innovative and efficient work toward IDS to detect the ongoing attacks over a large network. However, this algorithm reduces the space occupied by the dataset. So it would be useful for the network administrator/experts to avoid the delay between the arrival and the detection time of the attacks respectively. It also produces a less false alarm rate and computation time in real time. Performance of various existing algorithms is compared with the proposed EDADT algorithm for Intrusion Detection Systems. Information gain is applied to improve the accuracy of Intrusion Detection System through feature extraction. Then, select the training and testing data set. After a EDADT model is generated the test data set is applied on it to evaluate the Detection Rate (DR) and false alarm rate (FAR) values. Hence, compare the values obtained in the previous step and finally represent the results by measures of performance of the model. Table 4 represents the accuracy, sensitivity and specificity values for C4.5, SVM, C4.5 + ACO, SVM + ACO, C4.5 + PSO, SVM + PSO and Improved EDADT algorithms. The ROC curve is shown for a graphical plot of sensitivity and specificity.

Based on values obtained, the accuracy of C4.5 is 93.23%, the accuracy of SVM is 87.18%, the accuracy of C4.5 + ACO is 95.06%, the accuracy of SVM + ACO is 90.82%, the accuracy of C4.5 + PSO is 95.37%, the accuracy of SVM + PSO is 91.57% and the accuracy of Improved EDADT is 98.12%. Finally, an improved EDADT took highest accuracy percentage when compared to all six classification based algorithms. Fig. 8 specifies the corresponding chart for the result obtained in Table 4. Fig. 9 illustrates the build time of C4.5, SVM, C4.5 + ACO, SVM + ACO, C4.5 + PSO, SVM + PSO and Improved EDADT algorithms. C4.5 + PSO take more time to build the model. SVM + ACO takes less time than the proposed algorithm but provide less accuracy percentage than the other. However, the improved EDADT takes less time when compared to C4.5 + PSO and SVM + PSO and provides better accuracy in terms of all existing algorithms.

Fig. 10 shows the performance of existing and proposed EDADT algorithms based on false alarm rate (FAR). Thus the proposed EDADT Algorithm effectively detects attack with less computational time and false alarm rate.

6.2. Proposed Hybrid Intrusion Detection System

The misuse based and anomaly based approach has been taken for the study. Hybrid IDS is developed to overcome the human interaction toward preprocessing. Most of the evaluation of intrusion detection is based on proprietary data and results are not reproducible. To solve this problem, KDD Cup 99 (2009) has been used. Public data availability is one of the major issues during evaluation of Intrusion Detection System. Mixed data set (real time + simulated) has been used for this study. Out of 500 data instances, 320 instances involved in the training phase and remaining 180 instances are taken for testing phase. Each data include the source IP

address, destination IP address, state of the packet and so on. Data can be analyzed with the help of the snort rules that are predefined within it and anomaly score for each packet. Under anomaly based approach, we have four types of statistical methods like Packet Header Anomaly Detection (PHAD), Network Traffic Anomaly Detector (NETAD), Application Layer Anomaly Detector (ALAD), Learning Rules for Anomaly Detection (LERAD) respectively. Analysis is done based on the scenarios given below:
Based on SNORT.
Based on SNORT + PHAD.
Based on SNORT + PHAD + ALAD. Based on SNORT + ALAD + LERAD.

6.2.1. Performance of SNORT
SNORT is tested on real time traffic and simulated data set (one week data including attack) and attacks detected are listed day by day. The files have been downloaded from and LAN network. Attack detected on a daily basis is shown in Fig. 11. SNORT has detected 77 attacks out of 180 attacks without adding any anomaly based approaches.

6.2.2. Performance of SNORT + PHAD
Attacks detected by SNORT and PHAD on their own and results in the Hybrid Intrusion Detection System are shown in Fig. 12. It is understood that after adding PHAD with Snort it detects better than before. The number of attacks detected by SNORT increased from 77 to 105 in SNORT + PHAD.

Table 3 Rule structure for KDD Cup 99 dataset.

Rule no.	Attack description	Attack type
1	protocol = ICMP, service = ecr_i, src_byte = 1032,	Smurf
	flag = SF, host_count = 255	
2	protocol = tcp, service = private or ctf, flag = SO or SF,	Neptune
	serror_rate = 1, srv_serror_rate = 1	
3	protocol = ICMP, service = SF or SH, src_byte = 8,	Nmap
	same_srv_rate = 1, srv_diff_host_rate = 1	
4	protocol = tcp, service = http, flag = SF or RSTFR,	Back
	src_byte = 54540, dst_byte = 7300 or 8314, same_srv_rate = 1, srv_count P 5	
5	protocol = UDP, service = private, flag = SF, src_byte = 1,	Satan
	dst_host_count = 255, dst_host_same_src_port_rate = 1	
6	protocol = UDP, service = SF, src_byte = 28, wrong fragment = 3,	teardrop
	dst_host_count = 255	
7	protocol = icmp, service = eco_i, flag = SF, src_byte = 18,	ipsweep
	count = 1, dst_host_count = 1	
8	protocol = TCP, service = Private or remote_ic, dst_host_count = 255,	portsweep
	dst_host_srv_count = 1	
9	duration = 26 or 134, protocol = tcp, service = FTP or login, flag = SF,	ftp_write
	logged_in = 1	
10	protocol = tcp, service = telnet, flag = RSTO, src_byte = 125 or 126, dst_byte = 179,	guess_passwd
	hot = 1, num_failed_login = 1	
11	service = imap4, count 6 4, dst_host_same_srv_rate = 1,	Imap
	dst_host_srv_count< = 1	
12	service = tcp, flag = telnet or ftp_data, flag = SF, dst_host_srv_count 6 3,	Multihop
	dst_host_same_src_port_rate = 1	
13	duration = 377or 299, service = tcp, flag = telnet, dst_host_count = 255,	Spy
	dst_host_diff_srv_rate = 0.01	
14	protocol = tcp, service = ftp, flag = SF, src_byte> 980, dst_byte P 1202,	warezclient
	hot P 3, dst_host_count = 255	
15	protocol = tcp, service = telnet or ftp_data, flag = SF, loggin_in = 1,	Buffer_overflow
	dst_host_same_srv_rate = 1	
16	duration P 2, protocol = tcp, service = ftp or ftp_data, flag = SF,	warezmaster
	dst_host_count> 2, dst_host_srv_count P 1	
17	protocol = tcp, service = telnet, flag = SF, dst_host_count = 1,	load module
	dst_host_same_src_port_rate = 1	
18	duration P 25, protocol = tcp, service = telnet, flag = SF, logged_in = 1,	Perl
	dst_host_srv_count 6 2, dst_host_diff_srv_rate 6 0.07	
19	protocol = tcp, service = telnet or ftp, flag = SF, dst_host_count = 255,	root kit
	dst_host_diff_srv_rate = 0.02	
20	protocol = tcp, service =	Land

	finger, flag = SO, land = 1, srv_count = 2,	
	dst_host_srv_serror_rate P 0.17	
21	protocol = ICMP, service = ecr_i, flag = SF, src_byte = 1480,	Pod
	wrong_fragment = 1, dst_host_count = 255, dst_host_diff_srv_rate = 0.02	
22	protocol = tcp, service = telnet, flag = SF, dst_host_count = 255,	Phf
	dst_host_serror rate = 0.02	

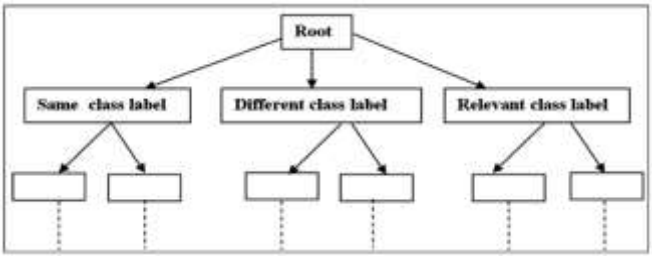

Figure 7 Overview of proposed EDADT algorithm.

Table 4 Performance of proposed EDADT vs. existing algorithms.

Algorithms	Sensitivity (%)	Specificity (%)	Accuracy (%)	FAR (%)
C4.5	86.57	82.00	93.23	3.56
SVM	83.82	64.28	87.18	3.2
C4.5 + ACO	89.26	85.42	95.08	0.87
SVM + ACO	87.42	67.95	90.82	2.42
C4.5 + PSO	92.51	88.39	95.37	0.72
SVM + PSO	90.06	70.88	91.57	1.94
Proposed EDADT	96.86	92.30	98.12	0.18

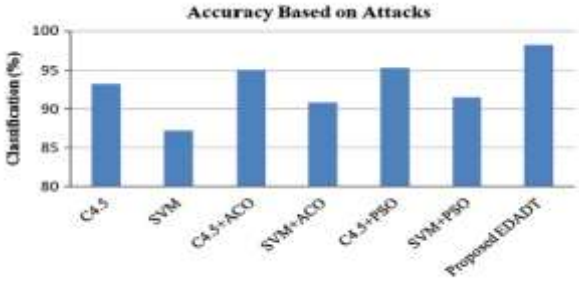

Figure 8 Results of EDADT algorithm vs. existing algorithms.

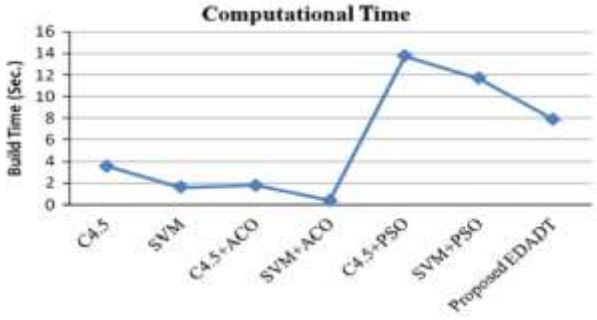

Figure 9 Computational time taken for the existing and proposed EDADT algorithms.

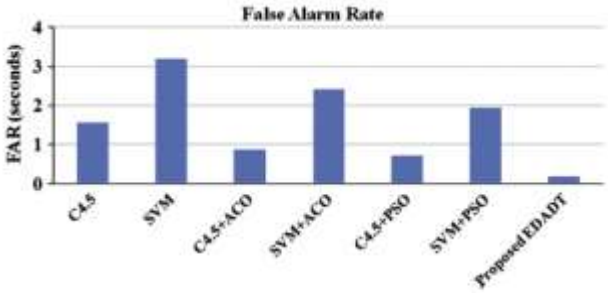

Figure 10 False alarm rate of proposed EDADT algorithm vs. existing algorithms.

6.2.3. Performance of SNORT + PHAD + ALAD

When PHAD and ALAD are added to the snort it detects more attacks than before. It is clearly shown in the graph Fig. 13. The number of attacks increases while adding PHAD and ALAD with SNORT the IDS becomes powerful. The number of attacks detected by SNORT + PHAD increased from 105 to 124 in SNORT + PHAD + ALAD version of IDS. The main reason is Snort detects the attacks based on rule definition files but PHAD and ALAD detect using packet header and network protocol.

6.2.4. Proposed Hybrid IDS (SNORT + ALAD + LERAD)

Attacks detected by SNORT + ALAD + LERAD on their own and results in the Hybrid Intrusion Detection System (SNORT + ALAD + LERAD) is shown in Fig. 14. After adding SNORT + ALAD + LERAD, the IDS gives better results when compare with other methods. The number of attacks detected by SNORT + PHAD + ALAD increased from 124 to 149 in SNORT + ALAD + LERAD (Hybrid IDS) version of the IDS.

6.3. Performance of SemiSupervised Approach for IDS

It is very hard for IDS to collect and analyze the data. Based on this issue, rule based technique has been applied. But if there is any little change in the data then the rule seems to be meaningless. To accomplish this task, we go for SemiSupervised Approach. In supervised approach labeled data can be taken for training phase and unlabeled data have been taken for testing phase. Usually the network data are unlabeled. It needs the security experts to label the unlabeled data which is expensive and time consuming. Because supervised approach needs the formal labeling of data to analyze whether the testing data are attacked or a normal one. But it is not realistic in real time. So SemiSupervised Approach is considered as most significant one. It requires only a small quantity of labeled data with large amount of unlabeled data. This method is done on the assumption. By analyzing the distance between the data points labeling is done. These data points are considered as most confident data. In turn, these data are taken as training data and corresponding testing data are applied to label the unlabeled one. The parameters of SVM are tuned between 0 and 1. This process continues till the bias value is same for many trials. Totally 5000 datasets are taken in this approach. Training phase includes both the labeled and unlabeled data

together and testing with unlabeled data.

Based on the predicted label, the accuracy will be calculated. The four classes of the data set namely Dos, Probe, R2L and U2R label have been labeled as 1, 2, 3 and 4 respectively. In Fig. 15, the proposed SemiSupervised Approach is compared with the existing algorithms like Reduced Support Vector Machine, SemiSupervised clustering algorithm (PCKCM) and Fuzzy Connectedness based Clustering. Among these, the proposed SemiSupervised Approach shows 98.88% in terms of accuracy and Fig. 16 illustrates the false alarm rate out of 6 existing methods such as RSVM, one step Markov, order 10 Markov, Markov chain + drift, PCKCM and FCC. In these comparisons, the proposed semi supervised approach shows 0.5% false alarm rate respectively.

6.4. Mitigating DDoS attacks using Varying Clock Drift Mechanism

By performing DoS attack the adversary captures zombies of machine in the network and explores attack by flooding packets to make the server busy, which are harder to deflect and promotes congestion thereby the victim resources will be flooded with request from hundreds and thousands of multiple sources. Port hopping mechanism is carried out by. In the author presents the synchronous communication based on acknowledgment in the presence of fixed clock drifts, maintained by Hopping Period Alignment and Adjustment (HOPERAA) algorithm where the client clock relates to that of the server. But any acknowledgment loss and faster or slower clock drift of client enables the adversary to cause a direct attack on it. In the existing work the interval of HOPERAA grows gradually over time and opens ports seems to be easy cause for the adversary to enable possible attacks. Moreover, the servers need not to worry about synchronization of clock during multiparty communication. To overcome this issue, the Varying HOPERAA Algorithm has been proposed to adjust the deviation in terms of hopping between client and server with varying clock drift. The message overhead between client and server & its average value is observed to evaluate the performance of the proposed algorithm.

Based on the previous work we also made three experiments to validate the efficiency of the proposed algorithm. It includes,

– The average number of contact initiation trails that the client need to communicate with the server.

– The growth of Varying HOPERAA execution intervals is greatly reduced that are made to estimate the client's clock drift.

– Computation ratio is calculated to prove the efficiency of the server's receiving capability.

– Due to the variable clock drift the message loss do not cause severe damage.

Lemma. Suppose when we use server's clock as reference clock and the client sends the request message at time t1 with timestamp hc (t1) and the message received by the server at time t2 and reply to the message with variable clock drift time at time t4 as found in Fig. 22. The client sets the clock to Vs1 and executes the Varying HOPERAA Algorithm to adjust the hopping period and receives the port sequence randomly. At time t5, the client searches, finds the port and starts sending the message to the specific sequence. In time t7 the server analyzes the state of the message and sends a reply at t7 and this process continues till t8 respectively. Then we have,

Vhcðt9Þ Vhcðt4Þ 6 p 6
Vhcðt9Þ Vhcðt4Þ 2 Þ

ðt8Þ ðt5Þ þ 2l c ðt8Þ ðt5Þ ð

From this lemma, it can be proved that the message transfer delay can be avoided by selecting the appropriate port based on the length of the data. Thus, the proposed Varying HOPERAA can reduces message transfer delay and also decreases the execution time to fractions of seconds as shown in Fig. 17.

Fig. 18(a and b) indicates the average number of the contact–initiation part in various time unit say for example 1000 and 5000 ms. For 1000 ms the number of trails is estimated likewise for 5000 ms. Compared to the existing results, the proposed algorithm provides 3.6 trails for 9000 ports at 1000 ms and 3.2 trails at 5000 ms. Even if we have 10,000–50,000 ports the average number of trails would be reduced from existing algorithm. The message can be of 30, 40, 50 byte and so on. The average time spent in this part will be less than 3 s.

Fig. 19 shows the length of the HOPERAA execution interval is estimated with the hopping period of the server. At D = 0.3 L and pc = {0.7, 0.9, 1.1, 1.3} the Varying HOPER AA execution intervals are greatly reduced at every stage and after 10 executions the client keeps sending data within few minutes. At pc = 0.7 the execution interval seems to be 14 ms and if pc = 0.9 the execution interval seems to be is 13 ms, and if pc = 1.1 the execution interval seems to 10 ms and at pc = 1.3 the HOPERAA execution interval is over 9 ms approximately. If the value of the pc increases then the number of executions reduces. After one Varying HOPERAA execution, the client will know whether the clock rate is faster or slower than the server because it receives the varying clock drift details from the server according to the state of the messages. Hence the hopping strategy is reduced within short time sequences respectively.

Fig. 20(a) and (b) shows the receiving capability of server with various clock drift of the client is taken into account. The clock drift of the client pc is {0.6, 0.7, 0.8, 1.1, 1.2, 1.3, 1.4, 1.5} D = {0.1 L, 0.2 L, 0.3 L} and l = 40 and 100 ms. In this simulation scenario the client executes Varying HOPERAA Algorithm 10 times and server keeps the state of the message in the form of threshold value to issue the range of port numbers in accordance with the message sending ratio. Thus, the Varying HOPERAA algorithm shows 99% as receiving capacity at the 100 ms rate and does not show much deviation for 40 ms rate as shown in existing algorithm. In the previous work, only security issues in terms

of contact

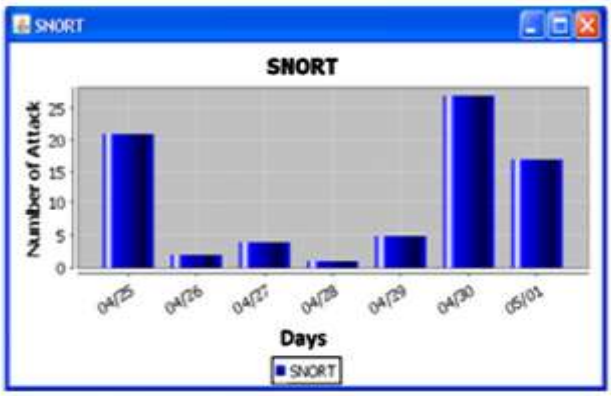

Figure 11 Attacks detected by SNORT on daily basis.

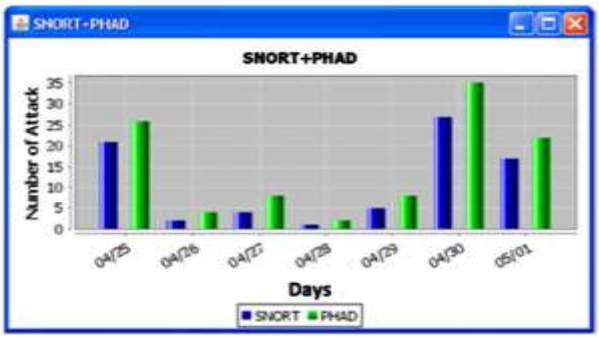

Figure 12 Attacks detected by SNORT + PHAD on daily basis.

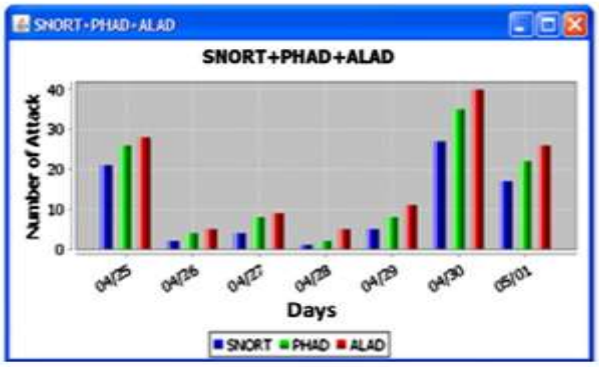

Figure 13 Attacks detected by SNORT + PHAD + ALAD on daily basis.

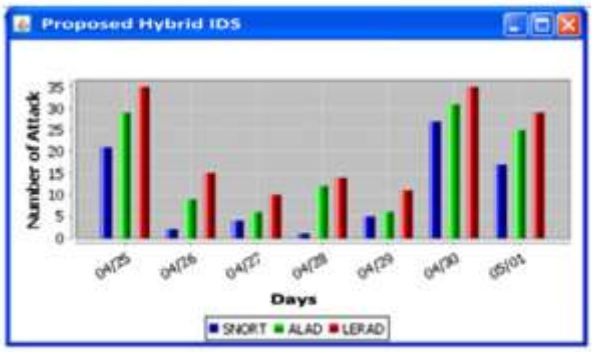

Figure 14 Attacks detected by proposed IDS on daily basis.

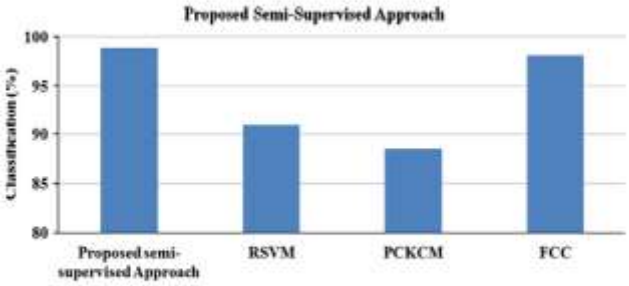

Figure 15 Performance of proposed Semi–Supervised Approach.

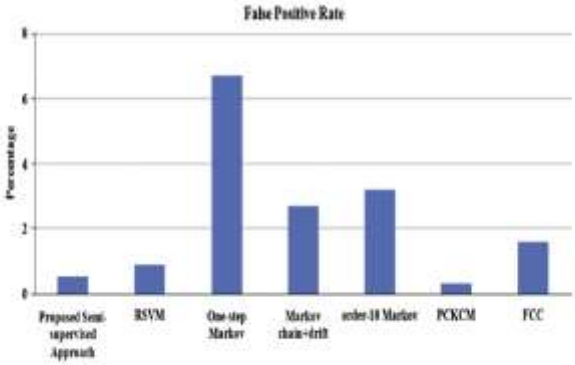

Figure 16 Performance of false alarm rate of proposed Semi-Supervised Approach vs. existing methods.

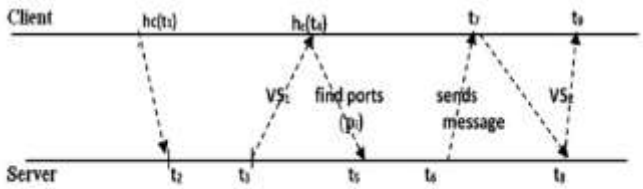

Figure 17 Client and server communication in execution of Varying HOPERAA Algorithm.

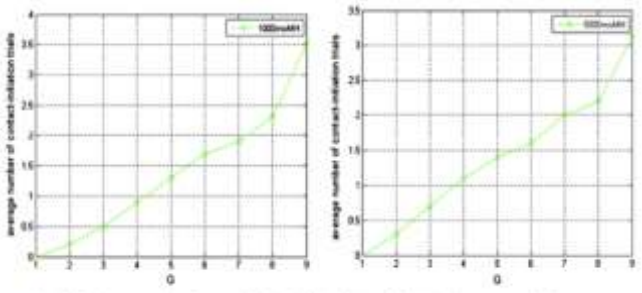

Figure 18 Average number of contacts-initiation trial in (a) 1000 and (b) 5000 ms in one contact initiation part.

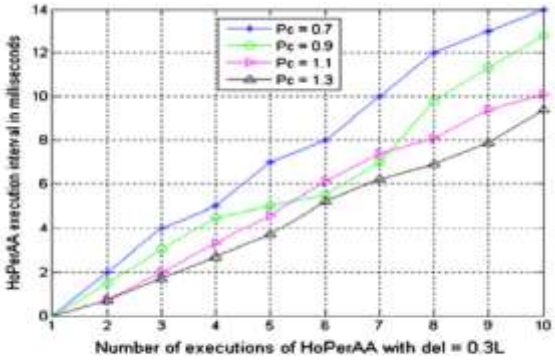

Figure 19 Length of Varying HOPERAA execution intervals vs. number of Varying HOPERAA executions.

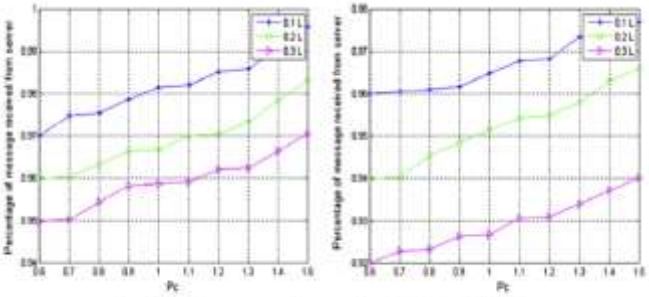

Figure 20 Receiving percentage of the server with μ is set to (a) 40 and (b) 100 ms.

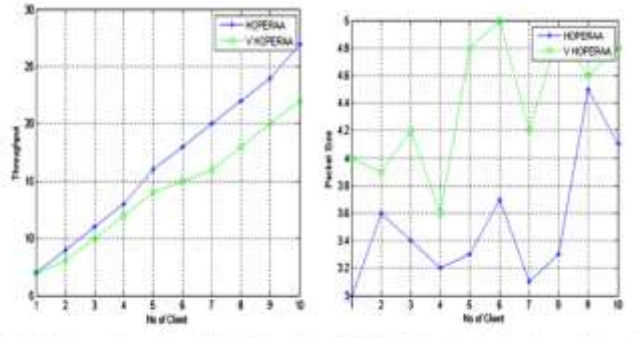

Figure 21 Performance of proposed varying HOPERAA and existing HOPERAA algorithm in terms of throughput, packet size total and number of clients.

List of Notation	
C	Client C
$h_c(t)$	Clock value of client C when server's clock value is it
$f\psi$	the pseudorandom function to generate the hopping sequence(s)
L	is the length of the server's hopping period.
Q	the maximum number of ports that the adversary can attack simultaneously
Δ	is maximum allowed value of the deviation between the hopping times of the server and the client
μ	The maximum message delivery latency
L_c	the length of the hopping period of client C
ρ_c	Clock drift of client C
ρ_L	The lower bound of the client's clock drifts
ρ_U	The upper bound of the client's clock drifts
λ	The seed used by the pseudorandom function to generate the hopping sequence
σ	The integer used by the pseudorandom function to generate a port number of a specific index in the hopping sequence
$Vh_c(t)$	Variable clock value of client C when server's clock value is t So $Vh_c(t)$ = (low rate (L), medium rate (M), Higher rate (H))
$V\rho_c$	Variable Clock drift of client C
$L\rho_L$	The lower bound of the client's clock drifts in low rate
$L\rho_U$	The upper bound of the client's clock drifts in low rate
$M\rho_L$	The lower bound of the client's clock drifts in medium rate
$M\rho_U$	The upper bound of the client's clock drifts in medium rate
$H\rho_L$	The lower bound of the client's clock drifts in higher rate
$H\rho_U$	The upper bound of the client's clock drifts in higher rate

Figure 22 List of notation used in proposed Varying HOPERAA Algorithm.

initiation trials for milliseconds, and sever's receiving capacity are taken into account. Apart from that, in our work the performance measures also evaluated using throughput and packet size metrics. If number of client increases then throughput seems to be increasing gradually. The comparison has been made with the existing HOPERAA and proposed varying HOPERAA algorithm. The result shows that the throughput increases as 5, 8, 11, 13, 16, 18, 20, 22, 24 and 27 for 10 clients and corresponding packet size also shown in Fig. 21 respectively.

VII. CONCLUSION AND FUTURE WORK

Based on the first issue, the proposed EDADT algorithm reduces the actual size of the dataset and helps the administrator to analyze the ongoing attacks efficiently with less false alarm rate respectively. It shows 19.4% better than C4.5, 18.8% better than SVM, 19.6% better than C4.5 + ACO, 19.2% better than SVM + ACO, 19.7% better than C4.5 + PSO, 19.3% better than SVM + PSO in terms of accuracy. Based on the second issue, the proposed Hybrid IDS performs well by detecting 149 attacks out of 180 (83%) attacks after training in one week attack free traffic data. This approach helps to overcome the human interaction toward preprocessing. Regarding third issue, the proposed SemiSupervised approach is 18.1% better than RSVM, 18.9% better than PCKCM, 19.9% better than the FCC. To solve the overwhelming problem of supervised and unsupervised methods, the semi supervised approach has been carried out. Finally, based on the mitigation of DDoS attack scenario, the port hopping concept is used depending upon the message length. Hence the message loss is greatly reduced and it does not create severe damage if happens. Both the security and performance measures with a variable clock drift mechanism have been evaluated. With the help of varying clock drift, the client can easily communicate with the server with minimum contact initiation trails and the improved maximum delivery latency has been achieved. Varying HOPERAA algorithm shows 99% as receiving

capacity at 100 ms and slightly deviates for 40 ms rate with better throughput capacity. Our experimental results proved that the proposed algorithms solves the above mentioned issues and detects the attacks in an effective manner compared with other existing works. Thus, it will pave the way for an effective means of intrusion detection with better accuracy and reduced false alarm rates.

In the future, two issues such as lack of resource consumption information and lack of model adjustment information techniques have been deployed to achieve an automatic Intrusion Detection System.

REFERENCES

[1] Anderson JP. Computer security threat monitoring and surveillance. In: Technical report. Fort Washington, Pennsylvania:James P Anderson co.; 1980.

[2] MohammadrezaEktefa, Sara Memar, Fatimah Sidi, Lilly SurianiAffendey. Intrusion detection using data mining techniques. In: International conference on information retrieval and knowledge management; 2010. p. 200–4.

[3] Holden Alex Nicholas, Freitas A. A hybrid PSO/ACO algorithmfor discovering classification rules in data mining. J ArtifEvolAppl 2008;2:1–11.

[4] Ardjani Fatima, SadouniKaddour. Optimization of SVM multiclass by particle swarm (PSOSVM). J Mod EducComputSci2010;2:32–8.

[5] Sethuramalingam S. Hybrid feature selection for network intrusion. Int J ComputSciEng 2011;3(5):1773–9.

[6] Mrutyunaya Panda, Ajith Abraham, ManasRanjan Patra. A hybrid intelligent approach for network intrusion detection. In: International conference on communication technology and system design, procedia engineering, vol. 30; 2011. p. 1–9.

[7] Petrussenko Denis. Incrementally learning rules for anomalydetection. CS200902. Florida: Florida Institute of TechnologyMelbourne; 2009.

[8] Matthew Vincent Mahoney. A machine learning approach to detecting attacks by identifying anomalies in network traffic. TRCS200313, Melbourne, Florida; 2003.

[9] Mahoney Matthew V, Chan Philip K. PHAD: packet headeranomaly detection for identifying hostile network traffic. In:Technical report CS200104. Florida Institute of Technology; 2001.

[10] Gao Xiang, Wang Min. Applying semi supervised cluster algorithm for anomaly detection. In: IEEE international symposium on information processing, no. 3; 2010. p. 43–5.

[11] Qiang Wang, Vasileios Megalooikonomou. A clustering algorithm for intrusion detection. In: International conference on data mining, intrusion detection, information assurance, and data networks, security, 5(12), 2005, p. 31–8.

[12] ChingHao, HahnMing L, Devi P, Tsuhan C, SiYu H. Semisupervisedcotraining and active learning based approach for multiview intrusion detection. In: ACM symposium on applied computing, no. 9; 2009. p. 2042–7.

[13] ChienYi Chiu, YuhJye Lee, ChienChung Chang. Semisupervised learning for false alarm reduction. In: Industrial conference on data mining, no. 10; 2010. p. 595–605.

[14] Li Jimin, Zhang Wei, KunLun Li. A novel semi supervised SVMbased on tritraining for intrusion detection. J Comput 2010;5(4):638–45.

[15] Monowar H. Bhuyan, Bhattacharyya DK, Kalita JK. An effective unsupervised network anomaly detection method. In: International conference on advances in computing, communications and informatics, no. 1; 2012. p. 533–9.

[16] Lane T. A decisiontheoretic, semi supervised model for intrusion detection. In: International conference on machine learning and data mining for computer security; 2006. p. 157–77.

[17] Zhang Fu, Marina Papatriantafilou, PhilippasTsigas. Offthewall: lightweight distributed filtering to mitigate distributed denial of service attacks. In: IEEE international symposium on reliable distributed systems, no. 31; 2012. p. 207–12.

[18] Zhang Fu. Marina Papatriantafilou, PhilippasTsigas, Wei Wei. Mitigating denial of capability attacks using sink tree based quota allocation. In: ACM symposium on applied computing, no. 25; 2010. p. 713–18.

[19] Zhang Fu. Marina Papatriantafilou, PhilippasTsigas. CluB: a cluster based framework for mitigating distributed denial of service attacks. In: ACM symposium on applied computing, no. 26; 2011. p. 520–27.

[20] Vincenzo Gulisano, Zhang Fu, Mar CallauZori, Ricardo Jim EnezPeris, Marina Papatriantafilou, Marta PatinoMartınez. STONE: a streambased DDoS defense framework. In: Technical report no. 201207, ISSN 1652926X, Chalmers University of Technology; 2012.

[21] Zhang Fu, Marina PapatriantaFilou, PhilippasTsigas. Mitigating distributed denial of service attacks in multiparty applicationsin the presence of clock drifts. IEEE Trans Depend SecureComput 2012;9(3):401–13.

[22] Catania Carlos A, Garino Carlos Garcı´a. Automatic networkintrusion detection: current techniques and open issues. ElsevierComputElectrEng 2012;38(5):1062–72.

[23] KDD Cup99 intrusion Detection Dataset. Available from: <http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html>.

