A NOVEL FRAMEWORK FOR INTRUSION DETECTION USING DISTRIBUTED COLLABORATION DETECTION SCHEME IN PACKET HEADER DATA

T.S. Urmila[1] and R. Balasubramanian[2]

[1] Research Scholar, Mother Teresa Women's University, Kodaikannal
[2] Dean, KarpagaVinayagar College of Engineering and Technology, Mathuranthagam

ABSTRACT

In recent years, the number of attacks on the computer network is voluminous. Secure data communication over the network is always under threat of intrusions. To protect from these attacks various intrusion detection techniques have been developed. Anomaly detection system detects the novel attacks based on deviation of the behavior of packets from the normal flow and Signature detection system detects known attacks based on stored signatures. We have proposed a Distributed collaboration detection scheme that combines the advantages of Anomaly and Signature based method by capturing the packets in real time. The uninteresting traffics are filtered by packet filtering and further normalization. The relevant features are selected based on our Correlation based BAT Feature Selection (CBBFS) Algorithm. Our Proposed Efficient Behavioral Prediction (EBP) scheme analyzes the episodes and classifies the attack based on EGSSI. Then Proficient Ordinance Generation (POG) for Inspection of IP Phase labels the IP as trusted or untrusted. Our proposed framework outperforms the results of existing classification algorithms (C4.5, Naive Bayes, PSO, GSA and EDADT) by reducing the rate of false positives.

KEYWORDS

Intrusion Detection, Anomaly Detection, Signature Detection, Distributed Collaboration, Packet Header Data

1. INTRODUCTION

The emerging trends of internet cause so many security issues and it becomes a major challenge to administrators. The attacker forward malicious packets to the target machine thereby steal, modify or corrupt secret information by breaking the security and also evade the Firewalls present in the network. So, we have to develop new security models to safeguard the availability, confidentiality and integrity of information systems. Immediate detection and reporting about the intrusions are essential to protect the system before the attacks spoilt it. Intrusion detection System (IDS) could be a mechanism or that supervises the malicious activities of user and constructs reports to a management station. They operate by analyzing current system events, either at the host level (HIDS), the network level (NIDS), or both. IDS will be classified into two categories; Anomaly and Signature based detection. In Anomaly based detection, the normal user behavior patterns are profiled. The incoming packets which deviate from this normal behavior pattern are identified as intrusions. It can detect novel attack but it is very complicated to develop and decide the normal behavior patterns. In contrast, the Misuse or Signature based IDS have a collection of signatures stored in a database. The incoming packets are compared with the signatures from the database and the matched signatures are identified as intrusions. This signature based IDS are very effective but it identifies only the known attack [4]. Hence we go for hybrid intrusion detection which improved by pre-processing and hybrid detection model.

The rest of the paper is organized as follows: Section II will discuss the Review of the proposed methods, Motivation of this system is described in section III, section IV says the proposed work, and section V says the performance evaluation and section VI ends with the conclusion and future enhancement.

DOI: 10.5121/ijcnc.2017.9408

2. RELATED REVIEW

Many approaches have been studied to detect, prevent and mitigate malicious network traffic. For example, rule-based approaches, such as SNORT, try to apply previously established rules against incoming traffic to detect and identify potential DOS attacks close to the victim's network. To cope with novel attacks, however, IDS tools such as Snort require to be updated with the latest rules. This paper looks at the problem of designing generalized measurement based real-time detection mechanisms. Measurement-based learning have considered traffic dimensions [1], [2], [3], number of flows [5] as probable signals that can be evaluated in order to spot anomalies in network traffic, while we further indulgence the traffic headers such as addresses and port numbers. Work in [6] relies on input data from multiple sources (i.e., all links in a network), while our work focuses on a single link at a time. Some approaches proactively seek methods to suppress the overflow of traffic at the source [7]. Controls based on rate limits have been adopted for reducing the monopolistic consumption of available bandwidth, to diminish the effects of attacks, either at the source or at the destination [8]. The apparent symptoms of bandwidth attack may be sensed through monitoring bit rates [9] and/or packet counts of the traffic flow. Bandwidth accounting mechanisms have been suggested to identify and contain attacks. Packeter [10] and others offer commercial products that can account traffic volume along multiple dimensions and allow policy-based rate control of bandwidth. Pushback mechanisms have been proposed to contain the detected attacks closer to the source. Trace back has been proposed to trace the source of DDoS attacks even when the source addresses may be spoofed by the attacker [11].

Data mining technique has been wide applied within the network intrusion detection system by extracting useful data from sizable amount of network information. During this [12] paper a hybrid model is planned that integrates Anomaly based Intrusion detection technique with Signature based Intrusion detection technique is split into two stages. During this [13] paper we've enforced the signature-based Network intrusion detection exploitation Snort and WinPcap. The paper [14] illustrates the idea of detecting the DoS Attack. There are many methods available to Detect and avoid the DoS attack. This paper mainly deals with the DoS attacks. In this [15] project this pattern matching is based on the regular expression where as these pattern of known attack are stored in the database of Intrusion Detection System. Regular Expressions are often used to describe malicious network pattern. In this [16] work, we identify the intrusion by capturing the real time network traffic by using Snort and perform the detailed analysis on the captured packet using network monitoring tool called Wireshark. It detains the real traffic from the wired or wireless intermediate and carries out the intrusion detection on snort.

3. MOTIVATION

Our approach to detecting anomalies envisions two kinds of detection mechanisms, i.e., investigation and real-time modes. An investigation analysis may exploit many hours of traffic data as a single data set, employing more rigorous, resource-demanding techniques for analyzing traffic. Such an analysis may be useful for detection in engineering purposes, analysis of packets and its usage, understanding header data, etc. On the other hand, real-time analysis would concentrate on analyzing a small window of traffic data with a view to provide a quick and possibly dirty warning of impending/ongoing traffic anomalies. Real-time analysis may rely on less sophisticated analysis because of the resource demands and imminence of attacks. The proliferation of heterogeneous computer networks has serious implications for the intrusion detection problem. Foremost along with these suggestions is the increased prospect for not permitted access that is afforded by the network's connectivity.

- The difficulty is aggravated when dial-up or internetwork access is allowable, as well as when unmonitored hosts (viz. hosts without audit trails) are present.

- The employ of distributed rather than centralized computing possessions also implies reduced organize over those resources.
- The multiple self-determining computers are likely to produce more audit data than a single computer, and this audit data is isolated among the various systems.

4. PROPOSED METHODOLOGY

This paper intended that inspecting and identifying the network intrusion by capturing the network traffic in real time. The network monitoring and packet capturing is performed by our proposed framework "Distributed *Collaboration Detection (DCD)"*. This framework incorporates with the pre-defined rules that permits and monitoring the captured packets header and the status of the network.

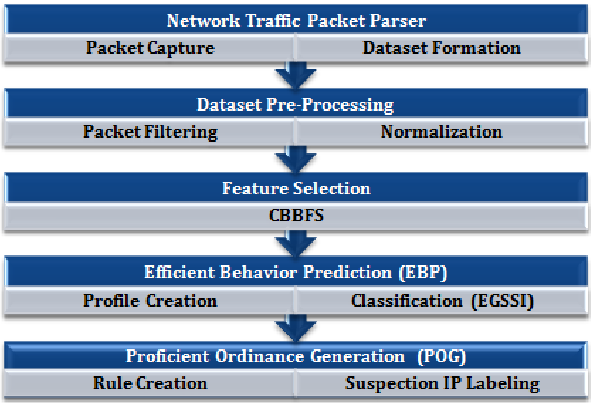


Figure 1. Overall Architectural Design of Intrusion Detection Based on DCD Framework

The packets are pre-processed by filtering based on some criteria and further normalization. Then feature reduction is done by our algorithm which efficiently selects the best features. The reduced features form a feature vector. The Efficient behavioral prediction scheme creates a profile which includes normal and attack behavior of the incoming traffic flow and classifies the suspected packets. The suspected packets are then passed through the set of rules to confirm the intrusion. This hybrid model efficiently identifies the intrusions both profile based and rule based methods. . Figure 1 depicts the proposed architecture of the Intrusion Detection based on Distributed Collaboration Detection (DCD) Framework.

4.1. FORMATION OF PROPOSED NETWORK TRAFFIC PACKET PARSER PHASE

Packet sniffer collects raw binary data from the wire. Typically, this is done by switching the selected network interface into promiscuous mode. Our approach is based on the observation of the numbers of packets collected from the card's interface of network equipment that is likely to have strong patterns of behavior over time. The captured packets are stored in a database in an appropriate format for further analysis.

The objective of this paper is to set up a test network for generating normal as well as attack network traffic and capture the traffic in terms of packet as well as flow modes in isolated environments. The captured traffic will be filtered, pre-processed, analyzed.

4.1.1 Formation Of Proposed Network Traffic Packet Parser Phase

Packet sniffer collects raw binary data from the wire. Typically, this is done by switching the selected network interface into promiscuous mode. Our approach is based on the observation of the numbers of packets collected from the card's interface of network equipment that is likely to have strong patterns of behavior over time. The captured packets are stored in a database in an appropriate format for further analysis.

The objective of this paper is to set up a test network for generating normal as well as attack network traffic and capture the traffic in terms of packet as well as flow modes in isolated environments. The captured traffic will be filtered, pre-processed, analyzed.

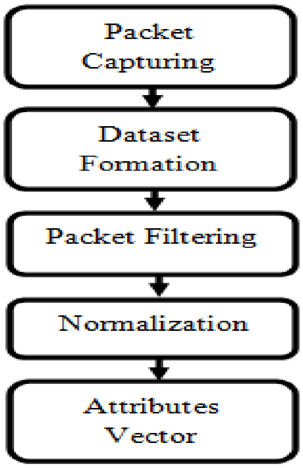


Figure 2. Design of Dataset Pre-Processing

4.2. Design Of Proposed Dataset Pre-Processing Phase

The Pre-processing is significant to enhance the packet processing time and outcome of this process. In this context, we have chosen the progression time window for the packets transaction as 2 minutes. In this time interval, transferred packets rate might be average of '200' packets. With these received and delivered packets the main issue is the process time for the separate transactions. Thus the step of pre-processing deals with reducing the progression time for received packets. In this paper, the *packet filtering* is used to reduce the packet processing time by avoiding the outgoing packets. The packet filtering process steps are one by using the following conditions. Hereby we are eliminating packets based on following conditions,

1. Only IP packets are considered, non-IP packets are filtered
2. Remove packets which has the UDP Protocol and Destination Port > 1023
3. Remove packet which has packet size less than 250bytes.

The non-IP packets like ARP, STP, etc., are filtered because the non-IP packets do not have any content in it. They are just a Broadcasting Packets. Therefore, there is no need to focus on those packets for intrusion and we have removed those packets from further processing.UDP packets with higher port (>1023) is removed because an untrusted user may be interested to formulate applications using these ports. (The first 1024 ports are restricted ports or standard ports like HTTP, FTP, etc.).

Data sets were pre-processed by extorting the IP packet description information to generate attribute vectors. The feature vector has the following format:

Table 1. Packet Dataset Attributes Vector

SIPA(x)	SP	DIPA(x)	DP	PLEN	PT
TIME	TOS	TTL	TCPSEQ	TCPACK	TCPHEADERLEN
TCPFLAG	TCPWINSIZE	TCPCHECKSUM	TCPOPTION	TCPURGPTR	ICMPTYPE
ICMPCODE	ICMPCHECK	UDPLEN	UDPCHECK	SERVICES	

Where SIPA = Source IP address nibble, where x = [1-4]. Four nibbles constitute the full source IP address SP = Source Port number DIPA = Destination IP address nibble, where x = [1-4]. Four nibbles constitute the full destination IP address DP = Destination Port number, PLEN = Packet length in bytes,PT = Protocol type: TCP, UDP or ICMP.

Normalization

The normalization is the process of renovating the filtered network traffic packets into nominal dataset.Ordinal normalization is to rank the continuous value of anattribute and then normalize the rank into [0, 1]. Clearly, ordinal normalization also ranges the values of anattribute into [0, 1]. In this paper, we do not increase the rankif some values of an attribute are the same.

4.3 Proposed Feature Selection using Correlation Based BAT Feature Selection (CBBFS) Algorithm

IDS need to examine very large data with high dimension. Due to this, it suffers from low detection rate and huge computations. Feature selection select the feature subset of original feature that retains the high detection accuracy and improves the training and testing time that helps to build efficient IDS which is pertinent for the real time.

In this work, the CBBFS approach is used to obtain the essential feature (field) selection of the network traffic packets by using the fusion technique of *Correlation Based BAT Feature Selection (CBBFS) Algorithm.*The Correlation Feature Selection (CFS) measure evaluates subsets of features on the basis of the following hypothesis: ”Good feature subsets contain features highly correlated with the classification, yet uncorrelated to each other”. The main characteristics in the BA are based on the echolocation behavior of microbats. As BA uses frequency tuning, it is in fact the first algorithm of its kind in the context of optimization and computational intelligence.The algorithm (CBBFS) mainly consists of two parts for achieving the goal of reducing dimensionality of the original feature space.

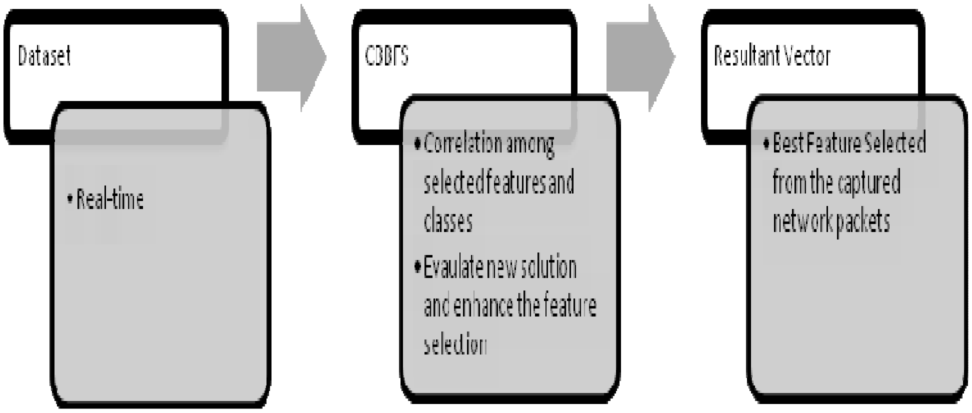

Dataset
•Real-time
CBBFS
•Correlation among selected features and classes
•Evaulate new solution and enhance the feature selection
Resultant Vector
•Best Feature Selected from the captured network packets

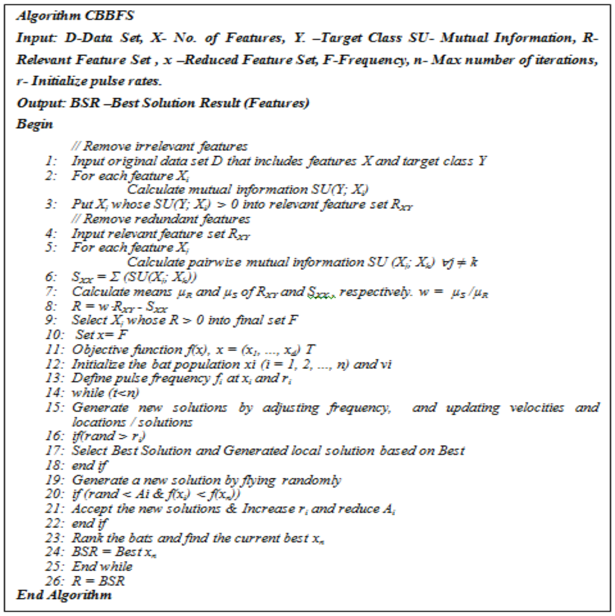
Figure 3. Process Flow of CBBFS

In the first component, the algorithm removes irrelevant features with poor prediction, eliminates redundant features that are inter-correlated with one of more other features and the significant features are selected. Given a data set with a number of input features and a target class, the algorithm firstly calculates the mutual information between features and class. The algorithm starts with calculating the inter-correlated strengths of each pair of features. The total amount of mutual information for each feature is acquired by adding all mutual information measures together that performed on feature-to-feature and feature-to-class to the same level and reach to the same important rank. Finally, the differences of them are computed and we only keep those features whose values are greater than zero, which means the selected features are the most “significant features” that restrain indispensable information of the original feature space.

In the second part, the algorithm represents that each bat is encoded with a velocity v^t_i and a location x^t_i, at iteration t, in a d-dimensional search or solution space. The location can be considered as a solution vector to a problem of interest. Among the n bats in the population, the current best solution x_* found so far can be archived during the iterative search process. The main objective of the bat algorithm can be summarized as follows: It would initialize, generate the initial features and determine the best solution from the features set. Virtual bats are moved in the search space according to updating rules of the bat algorithm and the best feature solution is enhanced by random walks.

Table 2. Feature Selection of Various Methodologies

Data Set / Measures	CFS	BAT	Proposed Methodology (CBBFS)
Real Time	1,8,5,7	4,5,6,8	3,4,5
KDD	5,4,2,1	4,5,6,9	2,4,5



```
Algorithm CBBFS
Input: D-Data Set, X- No. of Features, Y. –Target Class SU- Mutual Information, R-Relevant Feature Set , x –Reduced Feature Set, F-Frequency, n- Max number of iterations, r- Initialize pulse rates.
Output: BSR –Best Solution Result (Features)
Begin
    // Remove irrelevant features
    Input original data set D that includes features X and target class Y
    For each feature $X_i$
        Calculate mutual information $SU(Y; X_i)$
    Put $X_i$ whose $SU(Y; X_i) > 0$ into relevant feature set $R_{XY}$
    // Remove redundant features
    Input relevant feature set $R_{XY}$
    For each feature $X_i$
        Calculate pairwise mutual information $SU (X_j; X_k)\ \forall j \neq k$
    $S_{XX} = \Sigma\ (SU(X_j; X_k))$
    Calculate means $\mu_R$ and $\mu_S$ of $R_{XY}$ and $S_{XX}$, respectively. $w = \mu_S / \mu_R$
    $R = w \cdot R_{XY} - S_{XX}$
    Select $X_i$ whose $R > 0$ into final set F
    Set x= F
    Objective function $f(x)$, $x = (x_1, ..., x_d)\ T$
    Initialize the bat population xi ($i = 1, 2, ..., n$) and vi
    Define pulse frequency $f_i$ at $x_i$ and $r_i$
    while (t<n)
    Generate new solutions by adjusting frequency, and updating velocities and locations / solutions
    if(rand > $r_i$)
    Select Best Solution and Generated local solution based on Best
    end if
    Generate a new solution by flying randomly
    if (rand < Ai & $f(x_i) < f(x_n)$)
    Accept the new solutions & Increase $r_i$ and reduce $A_i$
    end if
    Rank the bats and find the current best $x_n$
    BSR = Best $x_n$
    End while
    R = BSR
End Algorithm
```

Figure 4. Algorithm for CBBFS Feature Selection

The outcome of Essential field selection algorithm is described in the following format

Table 3. Essentially Selected Fields

SORADD	DESADD	DESPORT	PROTOCOL	TCPFLAG	SERVICES

SORADD (Source Address), DESADD (Destination Address) and DESPORT (Destination Port) fields are significant for all the packet transaction. Every packet transaction acquires a protocol for communication. In this context, three various protocols are selected for the communication such as TCP, UDP and ICMP. The TCPFLAG field is the combination of syn, ack, rej flags combination that decides the status of the current packet. SERVICES field contains telnet, ftp, ecr_i, etc.

4.4 Design Of Proposed Efficient Behavioral Prediction (EBP) Phase

In this phase of EBP, the prediction model is presented by profile creation and classification models to predict the attack. The prediction model detects the suspected network IP by the following conceptions.

4.4.1 Conception of Profile Creation for Essential Features

Our Proposed Detection Engine has some procedures to find attacks such as Aggregation and Compute Threshold for profile creation. The process of profile creation is employed into two form namely *Single connection derived feature aggregation (SCD)* and *Multiple connection derived feature aggregation (MCD)*.SCD based aggregation is obtained by single source and destination pair. MCD based aggregation is obtained by considering overall packets grouped based on protocols.

4.4.1.1 Aggregation

Sophisticated attack traffic may not be distinguishable from regular traffic on an individual packet basis and the individual packets cannot be used to identify unusual trends or patterns over time.

So we go for aggregation based on 2 minutes window. The Packet Header Data which is stored in Database within the time limit of 2 minutes are aggregated based on the Formula.

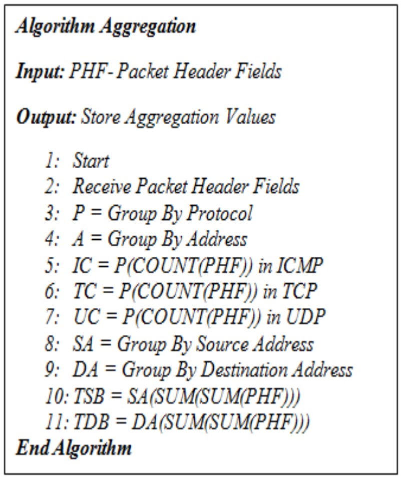
$$A(x) = \sum_{i=1}^{k} x_i \qquad (1)$$

Where A(x) is the aggregation value of x, x is the number of packets based on the protocol (TCP, UDP and ICMP) and k is the total number of packets transaction from every 2 minutes window. This equation (1) calculated the aggregation value based on the traffic flow between individuals. This assessment is further useful to calculate the Threshold value.

Table 4. Resultant Aggregation Fields

ICMP-COUNT	TCP-COUNT	UDP-COUNT	TCP-PACKETS	TOT- PACKETS
SENT-BYTES	RECE- BYTES	A1T	A1C	B1T
B1C	URGPTR	SADD-COUNT	SPORT-COUNT	REL_SER_RATE
DIS_SER_RATE	REJ	S-SEQ	R-SEQ	SSEQ-COUNT

Table 4 depicts the resultant aggregation fields for measuring the performance of packets transaction.



```
Algorithm Aggregation

Input: PHF- Packet Header Fields

Output: Store Aggregation Values

    Start
    Receive Packet Header Fields
    P = Group By Protocol
    A = Group By Address
    IC = P(COUNT(PHF)) in ICMP
    TC = P(COUNT(PHF)) in TCP
    UC = P(COUNT(PHF)) in UDP
    SA = Group By Source Address
    DA = Group By Destination Address
    TSB = SA(SUM(SUM(PHF)))
    TDB = DA(SUM(SUM(PHF)))
End Algorithm
```

Figure 5. Algorithm of Aggregation of Packet Header Fields

Table 5. Aggregation Result (Two Minutes Session Data)

ICMPCOUNT	TCPCOUNT	UDPCOUNT	Send_byte	Rec_byte	A1024	B1024
40	200	60	327	327	1	1
20	100	80	327	327	1	1
0	80	20	622	622	2	1
10	10	30	865	865	1	1
30	150	120	56432	56432	4	1
10	50	50	3074	3074	7	6
25	55	160	881	881	3	3
20	110	70	350	350	0	0

4.4.1.2 Compute Threshold

Threshold is a minimum or maximum value established characteristic of the traffic flow over the network which serves as a benchmark for guidance of which may term for a complete assessment of the situation of the packet flow among the network. If numerical constraints of network traffic, such as mean and standard deviation, are inactive disseminated under given traffic, thresholds of an explicit day could be useful to other days. It is generally known that traffic volume exhibits diurnal periodicity, which can be extracted by approximation coefficients.

After calculating the aggregated values of the traffic flow, we go for calculating the Threshold value to accomplish the next level of prediction. The aggregated values are averaged based on the number of incoming packets to compute the threshold. The Packet flow which crosses this threshold values are suspected to be an Intrusion attack. If there is a drastic change or any flooded traffic in the network, then definitely the heavy load affects the Threshold value. Thus this scheme, computes threshold value for every TCP, UDP and ICMP packets to measure standard measurements in predicting the traffic flow of packet declaration.

$$\lambda(x) = \frac{\sum_{i=1}^{n} A(x_i)}{Count(y)} \qquad (2)$$

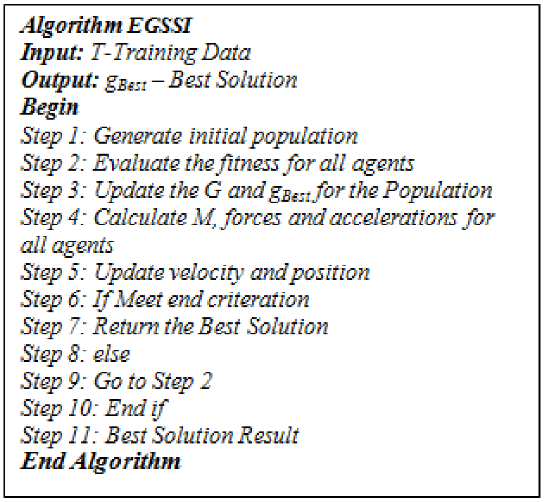
Where λ(x) is the threshold value of various communication protocols (TCP, ICMP and UDP), A(x) is total number of aggregation value count for a particular protocol, n is the number of total aggregation value based on the protocol, y is the total number of value count in the aggregation table, i is the number of iterations based on the total number of values and $x_i$is the sum of aggregation values in the aggregation table based on protocol.

4.4.2. Design of Classification using Enhanced Gravitational Search with Swarm Intelligence (EGSSI) Algorithm

In a practical system, known attack-free training data will not always be available as it is in our Data set. If the training data contains attacks, then the anomalies that it generates will be added to the model of "normal" traffic, so that future attacks of the same type will be missed. We could use a shorter 10 training period to reduce the probability of training on an attack, but this could also reduce the set of allowed values, resulting in more false alarms. We performed experiments to answer two questions. First, how much is performance degraded by using a shorter training period? Second, how much is performance degraded due to the hidden result, where training on one attack type hidden detection of other attacks. To answer the first question, we reduced the training period from 7 days (week 3, days 1-7), to 1 or 5 days. To answer the second question, we ran our framework in on-line mode. For each test day, we used the previous day's traffic for training. In testing, the training data therefore enclosed attacks. The sample training data are shown in Table 6. This depicts the total count of ICMP, TCP and UDP packets, total number of packets and the category of the packet class. In this, the category type with 0 denotes the normal packets and the category type with 1 denotes abnormal packets.

Table 6. Sample Training Data

ICMPCOUNT	TCPCOUNT	UDPCOUNT	No. of packets	Category (Class)
40	200	60	300	0
20	100	80	200	0
0	80	20	100	0
10	110	380	500	1
30	1500	120	1650	1
10	50	500	560	1



```
Algorithm EGSSI
Input: T-Training Data
Output: $g_{Best}$ – Best Solution
Begin
Step 1: Generate initial population
Step 2: Evaluate the fitness for all agents
Step 3: Update the G and $g_{Best}$ for the Population
Step 4: Calculate M, forces and accelerations for all agents
Step 5: Update velocity and position
Step 6: If Meet end criteration
Step 7: Return the Best Solution
Step 8: else
Step 9: Go to Step 2
Step 10: End if
Step 11: Best Solution Result
End Algorithm
```

Figure 6. Algorithm of Aggregation of Packet Header Fields

In this phase, the classification of network IP of the captured packet headers is processed for the inspection of the intrusion detection. The network IP based on the suspected packets with protocol class is categorized as training data is used for the classification. This training data set is instructed to the classifier to test the framework to inspect and classify the network traffic packets. In this paper, the classification metrics and prediction of the suspected IP are obtained by the PSO with GSA. In this paper, we hybridize PSO with GSA using low-level co-evolutionary heterogeneous hybrid, because we combine the functionality of both algorithms. It is co-evolutionary because we do not use both algorithm one after another. In other words, they run in parallel. It is various because there are two different algorithms that are concerned to make final results. The essential idea of EGSSI is to combine the capability of social thinking (g_{best}) in PSO with the local search capability of GSA.

Table 7. Performance Metrics Evaluation for Various Classification Methods

Methodology / Measures	Accuracy	MSE	True Positive	False Negative
C4.5	93.23%	0.8134	0.6023	0.3541
Naive Bayes	87.18%	0.7908	0.4656	0.3785
PSO	98.1%	0.7672	0.5714	0.3333
GSA	90.0%	0.1343	0.5700	0.3333
EDADT	98.12%	0.0134	0.4023	0.3312
Proposed Methodology (EGSSI)	99.2%	0.0014	0.5870	0.3033

The steps of EGSSI are represented in figure 6.

- In EGSSI, at first, all features are randomly initialized.
- Each feature is considered as a candidate solution.
- After initialization, Gravitational force, gravitational constant and resultant forces among agents are calculated.
- After that, the accelerations of particles are evaluated.
- In each step, the best solution so far should be updated.
- Finally, the positions of features are defined and the process of updating normal and abnormal packets in the network traffic.

4.5. Design Of Proposed Proficient Ordinance Generation (POG) For Inspection Of IP Phase

The main objective of rule generation is to ensure that the packet is concerned with any attack. Given a packet that contains a variation of a known attack, there should be some automated way to identify the packet as nearly matching an attack. If a particular statement has a set of conditions matches or against it, conditions in a rule may match some of the conditions in the packet. In this accomplishment, generation of rule in the case of matching network packets against rules involves allowing a packet to generate an aware if:

- The terms in the rule do not all equal, yet most of them do;
- The only conditions that do not equal exactly nearly equal.

As an example, assume a certain rule states that an alert should be generated if a packet is a particular length, on a particular port and contained a certain bit pattern. Using this generation rule a packet matching those criteria, except perhaps on a different port, or with a slightly different bit pattern, would still count as matching, and a (modified) alert would be generated.

Table 8. Rule Table

Protocol	Services	Host_Count	Source_Byte	Custom Fields	Attack
ICMP Type=8	-	-		REL_SER_RATE>1	NMap
	-	1	18		IPSweep
	-	255	1032		Smurf
	-	255	1480	DIS_SER_RATE>0.2	POD
TCP	PRIVATE/CTF	>0		Serror>1	Neptune
	TELNET	255			Phf
	PRIVATE/FTP	255			Multihop
	FTP	>=255		Source&Dest_byte>980	Warzclient
	Telnet/Ftp	=255			Rootkit
	Private/ remote ic	255			Portsweep
	http	-	>54540	REL_SER_RATE >1	Back
	-	-		F=0x14 & 0x19	Man in middle
UDP	Private	-		Packet-size>0	Satan

The table 8 illustrates the rule set for the IDS need to have some stored rules that enable them to detect intrusions. These rules are generally presented in the following form:

if{condition} then{ check attack}

The condition refers the affair in which a particular packet matches a rule in the IDS. The "check attack" can be seen as an action that needs to be performed based on the detected intrusion. This form of rule generation which satisfies condition by employing the various field values that differs on basis of protocol.

For instance, ICMP protocol packets with Type 8 which has the formation of rule like:

if {Protocol, Type, Host_Count, Source_Byte, Custom_Fields} then {Attack}

where *Protocol* defines the protocol type of the traffic flow, *Type* defines the packet type acquired by the protocol, *Source_Byte* describes the count of the source address byte and *Custom_Fields* denotes to the conventional field value that satisfies the other field criteria to identify the attack. For example,

if {ICMP, 8, 255, 1408, Dis_Ser_Rate>0.2} then {POD}

The intact rule generation is depicted in the following hierarchy representation in Figure 7.

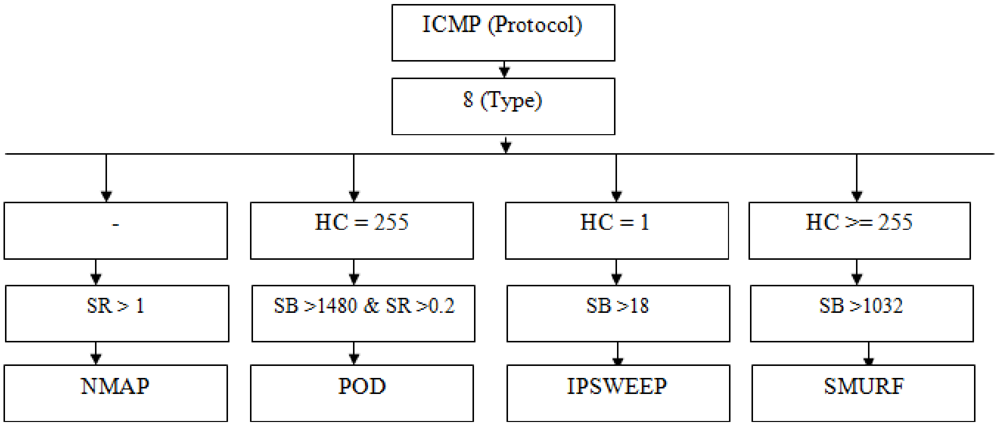


Figure 7. Collaboration Tree

When abnormal is discovered, the engine radios alarm information. If alarming message, which is collected contrary to POG from EBP, arrived the predestination threshold, EBP was sure that POG was the intrude node.

4.6. FORMATION OF SUSPECTED IP LABELLING PHASE

In this phase, the inspected network IP is suspected IP that gratifies the attack rule generated and labelled as trusted and untrusted IP which would reduce the time complexity of the inspection of entire network packets. This IP labelling phase is employed to inspect these labelled IPs' for the attack suspection instead of inspecting the entire network traffic IP that would condense the performance time of the detecting the intrusion.

5. PERFORMANCE EVALUATION

This section described the performance evaluation for the proposed system. Our implementation is in Core-i5 processer, 8 GB Ram, 10 Megabytes of training data and 5 gigabytes of test data in 364 seconds, or 1000 packets per second. The overhead is 23 seconds of CPU time per simulated day, or 0.026% at the simulation rate.

The performance of detection rate for number of packets, detection rate based on the existing methods and number of packets inspected within 5 days is shown as follows.

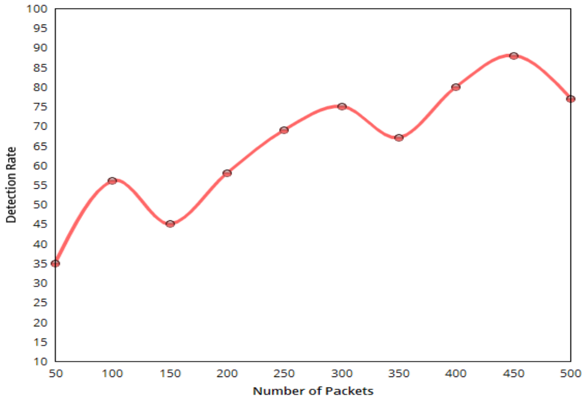


Figure 8. Detection Rate based on Number of Packets

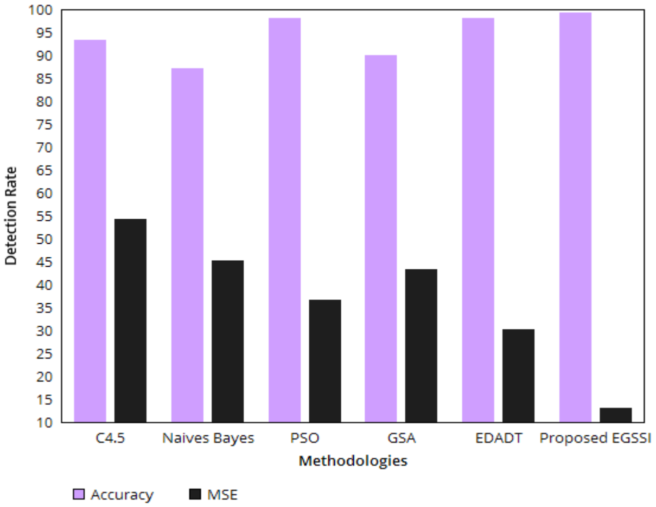


Figure 9. Detection Rate based on Existing Methods (Accuracy and MSE)

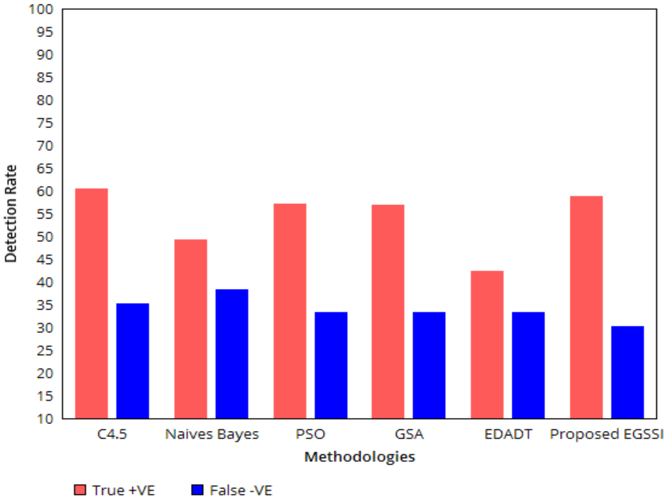


Figure 10. Detection Rate based on Existing Methods (True +VE and False -VE)

Figure 8 depicts the detection rate of the traffic flow for the proposed system. This figure illustrates that the detection rate flows on sensible for various number of packets.

The detection rate based on existing methods is depicted in figure 9 and figure 10. The figure defines that the detection rate of the proposed method provides high detection rate when compared with existing methods C4.5, Navies Bayes, PSO, GSA and EDADT [16]. The performance metrics like Accuracy, Mean Square Error (MSE), True Positive and False Negative are evaluated to measure and obtain the best detection rate. Among these the proposed technique EGSSI provides good detection rate.

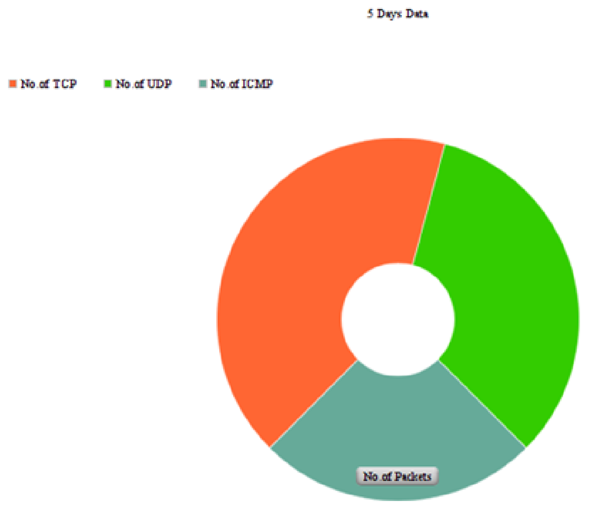


Figure 11. Number of Packets Inspected

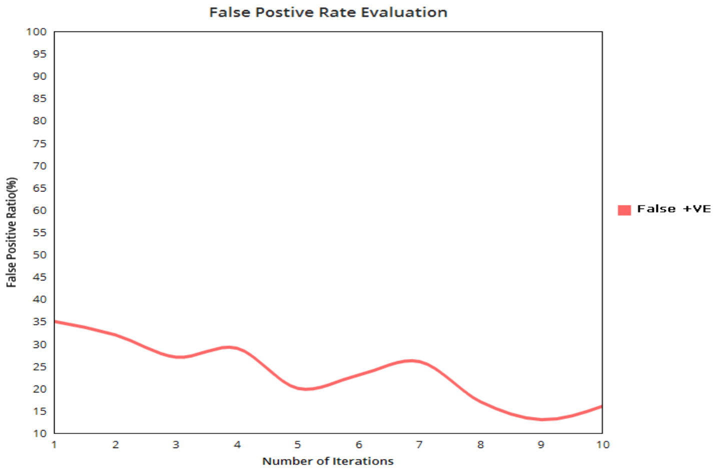


Figure 12. False Positive Rate Evaluation

Figure 11 defines that the number of TCP, UDP and ICMP packets inspected within 5 days. The false alarm rate typically refers to the anticipation of the false positive ratio. Figure 12 depicts the false positive rate evaluation for the proposed methodology DCD. The figure defines that the DCD provides sensibly low rate of false positive by means of the number of iterations.

6. Conclusion And Future Work

We studied the feasibility of analyzing packet header data through DCD (Distributed Collaborative Scheme) Framework for detecting traffic anomalies. This framework filters uninteresting traffic and obtains significant fields that reduce the time complexity. Experimental results illustrate that statistical analysis of aggregate traffic header data based on Single IP and Multiple IP artefact afford an efficient mechanism for the detection of anomalies within a network. We studied the effectiveness of our approach in inquisition and real-time analysis of network traffic.

Future system can be extended by incorporating Data Mining techniques to analyze the information in the packet content which may help in efficient decision making. This work can be extended by incorporating intellect into it in order to expand knowledge by itself by examining the growing traffic and learning new Intrusion patterns.

References

[1] P. Barford et al., “A signal analysis of network traffic anomalies,” in ACM SIGCOMM Internet Measurement Workshop, Nov. 2002

[2] A.Hussein, J. Heidemann, and C. Papadopoulus, “A framework for classifying denial of service attacks,” in ACM SIGCOMM, Aug. 2003.

[3] A. Lakhina, M. Crovella, and C. Diot, “Diagnosing network-wide traffic anomalies,” in ACM SIGCOMM, Sep. 2004

[4] J. Singh, Manisha J. Nene, “A Survey on Machine Learning Techniques for Intrusion Detection Systems”, International Journal of Advanced Research in Computer and Communication Engineering Vol. 2, Issue 11, November 2013.

[5] D. Plonka, “FlowScan: A network traffic flow reporting and visualization tool,” in USENIX LISA 2000, New Orleans, LA, Dec. 2000.

[6] A.Lakhina, M. Crovella, and C. Diot, “Diagnosing network-wide traffic anomalies,” in ACM SIGCOMM, Sep. 2004

[7] J. Mirkovic, G. Prier, and P. Reiher, “Attacking DDoS at the source,” in IEEE Int. Conf. Network Protocols, Nov. 2002

[8] A.Garg and A. L. N. Reddy, “Mitigation of DoS attacks through QoS regulation,” in Proc. IWQOS, May 2002

[9] Y. Zhang, L. Breslau, V. Paxson, and S. Shenker, “On the characteristics and origins of internet flow rates,” in ACM SIGCOMM, Aug. 2002

[10] S. Floyd, S. Bellovin, J. Ioannidis, K. Kompella, R. Mahajan, and V. Paxson, “Pushback messages for controlling aggregates in the network,” IETF Internet draft, work in progress, Jul. 2001.

[11] S. Savage, D. Whetherall, A. Karlin, and T. Anderson, “Practical network support for IP traceback,” in ACM SIGCOMM, 2000

[12] Jaina Patel, Mr. KrunalPanchal, "Effective Intrusion Detection System using Data Mining Technique",Journal of Emerging Technologies and Innovative Research (JETIR),Volume 2, Issue 6,June 2015.

[13] Sagar N. Shah Ms. Purnima Singh, "Signature-Based Network Intrusion Detection System Using SNORT And WINPCAP", International Journal of Engineering Research & Technology (IJERT), Vol. 1 Issue 10, December- 2012

[14] SuchitaPatil, Dr. B.B.Meshram, "Network Intrusion Detection and Prevention techniques for DoSattacks",International Journal of Scientific and Research Publicat ions, Volume 2, Issue 7, July 2012

[15] Mr. Chandrapal U. Chauhan Mrs. V.A.Gulhane, "Signature Based Rule Matching Technique in Network Intrusion Detection System", International Journal of Advanced Research in Computer Science and Software Engineering,Volume 2, Issue 4, April 2012

[16] RashmiHebbar , Mohan K, “Packet Analysis with Network Intrusion Detection System”, International Journal of Science and Research (IJSR), Volume 4 Issue 2, February 2015.

[17] G.V. Nadiammai, M. Hemalatha, "Effective approach toward Intrusion Detection System using data mining techniques", Egyptian Informatics Journal, Vol 15, PP-37-50, 2014

AUTHORS

First T.S. Urmila was born in Madurai, Tamilnadu, India. She received B.Sc Computer science and MCA degree from Madurai Kamaraj University, Madurai. She is pursuing the Ph.D. in Computer Science at Mother Teresa Women's University Kodaikannal. She is having more than 12 years of teaching experience in Computer Science department. Her Research interests include supervised and unsupervised learning, intrusion detection, virus signatures, and pattern matching algorithm. She is also presented and published more than 10 papers in these areas.

Second Dr. R. Balasubramanian has completed his Ph.D. in the year 1999. He is having more than 40 years of teaching experience and Research experience of more than 10 years. Now, he is working as Dean MBA/MCA in KarpagaVinayaga College of Engineering and Technology, Kanchipuram. He has published 14 papers in International Journals and 10 papers in National Journals. His areas of interest are Knowledge management, Network Security and Artificial Intelligence.