Indian Journal of Science and Technology, Vol 9(13), DOI: 10.17485/ijst/2016/v9i13/87983, April 2016
ISSN (Print) : 0974-6846
ISSN (Online) : 0974-5645

Anomaly based Malicious Traffic Identification using Kernel Extreme Machine Learning (KELM) Classifier and Kernel Principal Component Analysis (KPCA)

Lekha Jayabalan* and Padmavathi Ganapathi

Department of Computer Science, Avinashilingam Institute for Home Science and Higher Education for Women, Coimbatore – 641008, Tamil Nadu, India; lekhaphdscholar@gmail.com, ganapathi.padmavathi@gmail.com

Abstract

Objectives: The rapid growth of new vulnerabilities causes the network by Denial of Service attack (DoS). The DoS attack causes traffic flow in network. Therefore it increases the difficulties to detect the DoS attack in traffic by means of misuse detection. The behavior patterns are analyzed in anomaly detection to identify the attack. **Methods:** In detection of unknown worms anomaly detection is more comfortable than misuse detection. In this paper, hybrid optimization and extreme machine learning classifier is proposed for anomaly detection. This approach detects the DoS attack by analyzing the profiles of traffic patterns. **Findings:** Kernel Principal Component Analysis (KPCA) is adopted in this approach to extract the feature from the dataset. A short time window is utilized to gather all features from packet headers. Extreme learning machine based HGAPSO is used to classify the unknown attack. **Improvement:** Thus the proposed system is implemented as real-time. Performance evaluation shows that this approach provides 1.016s time consumption and 95 % accuracy than existing approach during detection of DoS in network traffic.

Keywords: Dos, ELM, MLBG, Optimization

1. Introduction

In network, Intrusion Detection System (IDS) is a hardware that supervises the system activities for malicious activities or policy violations. The major category of IDS are Network Based (NIDS) and Host based (HIDS) intrusion detection systems[1,2]. NIDS supervise traffic from all devices on the network[3] by placing at a strategic point or points in the network. Passing traffic is analyzed on the whole subnet that works in a promiscuous mode, and coincide with traffic that is passed on the subnets to the library of known attacks. HIDS supervise the inbound and outbound packets and alert the user or administrator if suspicious activity is detected[4] by running it on individual hosts or devices on the network. DoS also attack the database systems by corrupting the data, which can propagate quickly to other parts of the database system through valid users[5]. NIDSs are classified into two main classes: misuse detection and anomaly detection[6,7]. Misuse detection is used to detect known attacks by analyzing the rules. The rule updates are occurred sequentially and frequently released by NIDS vendors. Misuse detection is difficult if new vulnerabilities are arrived. The normal behavior patterns are analyzed in anomaly detection to detect the worms. In detecting the unknown worm or novel attack, anomaly detection method is more suitable technique than the misuse detection. However it may generate too many false alarms.

Denial-of-Service (DoS) attack makes the resource unavailable to its users. DoS attacks are the major threat in computer networks. NSFOCUS DoS Threat Report 2013 provides an analysis report that for every hour approximately 28 DoS attacks occur over the network. Due to this attack the resources across the network become unavailable to the planned user. The NSFOCUS Thread response and research team observed this analysis based on 244,703 DoS incidents. This attack is produced by flooding attacks by means of injecting malicious traffic in network[8].

*Author for correspondence

DDoS is a type of attack for disabling normal service in the network. This attack access to a network server that is blocked for legal costumer[9]. Misuse based detection scheme is difficult to find this type of malicious traffic generated in the network[10]. Hence, in this paper anomaly based detection is used for detecting the flooding attack in real time.

In this paper, hybrid optimization and extreme learning machine classifier based anomaly detection method has been proposed. Principal component analysis is used to extract the features from the dataset. A short time window is utilized to gather all features from packet headers. Extreme learning machine based hybrid optimization (HGAPSO) is used to classify the unknown attack. Thus, presented approach is implemented as real-time. Various detection mechanisms have been proposed by different authors to protect the network from malicious traffic flow like DoS attack. Different Graph Based Anomaly Detection Techniques are discussed in[11]. Some of the techniques proposed for detection of DoS attack from traffic flow is discussed below.

In[12] proposed a Naive Bayesian classifier methodology for the detection of DoS attacks. The IDS is one of the Dos detection systems that use the Naïve Bayesian methodology. This approach utilize the naïve Bayesian classifier to provide the variation by combining data pre-processing and feature selection model used in naïve Bayesian classifier to provide the performance variation. DoS provide huge loss in data in the network area. Therefore it is a major issue and that have be rectified by the technology several techniques are there to fix DoS attack but low accuracy level in detection of the DoS attack . The proposed method is proved that the accuracy and performance is improved by using the combination of data preprocessing and feature selection.

In[13] presented a DoS attack detection system based on multivariate correlation analysis which is powered by the triangle-area based on both MCA and anomaly-based detection technique. Geometrical correlation is mined in this technique that hides in two different features of separate in each network traffic record. This approach offers accurate performance for network traffic characteristics. The latter method promotes the system to differentiate the attack (both known and unknown DoS attacks) from authentic network traffic.

In[14] found DDoS attack possess higher similarity when compared with flash crowd flows under present controls of botnet size and organization. This approach utilized the flow correlation coefficient as a metric to calculate the similarity among suspicious flows in order to differentiate DDoS attacks from genuine flash crowds. The authors proved the feasibility of the proposed detection method and proved the better performance of the discrimination method in the current botnet size and organization.

In[15] presented a method to identify Dos attacks based on Computer vision technology. Several methods are available to detect the dos in the network. The previous detecting mechanism is worked based on machine learning approach and analysis of statistical approach. The new procedure for DoS detection work is separated into two modules first module is converting the traffic record as a images and the second module is considering the attack detection as computer vision. The converted image part of traffic record is the input for the proposed system. This conversion is proposed by one of the dissimilar measure called Earth Mover's Distance (EMD). The author have used the KDD Cup 99 data set and ISCX 2012 IDS data set which is collected from real experiments involving a DoS attack.

In[16] proposed a packet filtering scheme based on router that provides many filters during the reduction of quantity of filtering routers by tuning it. Filter aggregation to an integer linear programming problem and design an incremental, efficient updating filter aggregation algorithm, therefore the attack flows can be timely filtered in the progress of undesired-flow identification. Based on the real-world Internet topologies the authors implemented this work on the emulated DoS scheme.

In[17] an Artificial Immune System (AIS) is implemented to proposed a IDS to identify the DDoS attacks. An algorithm based on anomaly and signature-based detection is used in this approach which mapped to AIS known as "Generation of Detector (Genetic Algorithm)". To identify the intrusion an attack is detected at each time and a new generation is appended to the detectors dataset. This algorithm increases detection rate of intrusions and reduces the false positives.

In[18] a novel scalable detection method with Support Vector Machines is presented for network based anomalies. Dynamically Growing Self-Organizing Tree (DGSOT) is a clustering algorithm that deals with large datasets in order to improve the training time of SVM.

Table 1 shows the different methods proposed to detect the traffic which affects the network. This observation gives the clear idea that, traffic misbehaviors are found through monitoring. Most of the authors were experimented for off-line detection and they couldn't

Table 1. Comparison of existing detection methods

Year	Authors	Techniques Used	Detection mechanisms	Parameters used for evaluation
2014	Lu Ning , Su Sen ,Jing Maohua and Han Jian	router based packet filtering	Filter aggregation to an integer linear programming problem. Then, design an efficient incremental updating filter aggregation algorithm, in which the attack flows can be timely filtered in the progress of undesired-flow identification.	Throughputs, number of filters and Collateral Damage Ratio
2014	Tan and Zhiyuan (Thomas)	Computer vision technology	Initially converting the traffic record as a images and the then consider the attack detection as computer vision. The converted image part of traffic record is the input for the proposed system. This conversion is proposed by one of the dissimilar measure called Earth Mover's Distance (EMD).	high accuracy, the minimization of the system burden, and extendibility for system deployment
2014	Zhiyuan Tan, ArunaJamdagni, Xiangjian He, Priyadarsi Nanda, and Ren Ping Liu	MCA-based DoS attack detection system	uses multivariate correlation analysis (MCA) for accurate network traffic characterization by extracting the geometrical correlations between network traffic features	detection accuracy,time and False-Positive Rate
2013	Katkar, V.D, Kulkarni, S.V	Naive Bayesian classifier	evaluates variation in performance	Better accuracy and performance.
2012	S. Yu, W. Zhou, W. Jia, S. Guo, Y. Xiang, and F. Tang	flow similarity-based algorithm	flow correlation coefficient is used as a metric to measure the similarity among suspicious flows to differentiate DDoS attacks from genuine flash crowds	Delay,noise, number of merged flows.

meet the demands of real-time for NIDSs due to the 41 features inKDDCUP99 were from connections only not packets. To rectify these issues, the proposed work identifies and monitors the traffic flow.

2. Proposed Methodology

We proposed an improved LBG (MLBG) methodology in this work. The MLBG algorithm is clustering algorithm which applied in order to reduce the item in the sampling dataset. Then feature extraction if performed by the Kernel Principal Component Analysis (KPCA) from the sampling dataset. The optimal weight vector is calculated by the hybrid optimization technique. The distance calculation is forced by the weight vector W that promotes the modified Extreme learning machine classification technique which is used to train the dataset and classify them into malicious and non-malicious type of network traffic. The proposed methodology flow diagram of is given below in Figure 1.

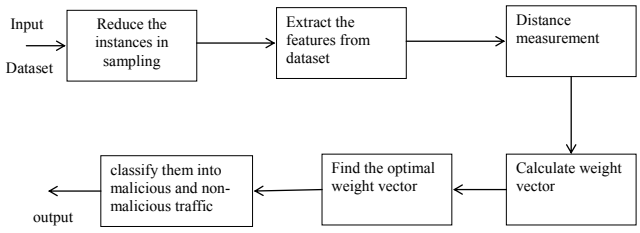

Figure 1. Block diagram of the proposed methodology.

2.1 Modified LBG Algorithm (MLBG)

MLBG is based on simple nearest neighbor algorithm is shown in Figure 2. The finite sequence of iterations in MLBG produces a new quantizer at every step, with total number of distortion. This distortion should be less than or equal to previous one. A set of input vectors s is considered as input for MLBG which is defined as $s=\{x_i \in Rd \mid i=1,2,...,n\}$ then it provides a characteristic subset of vectors which is defined as $c=\{c_j \in Rd \mid j=1,2,...K\}$. This subset is generated with a user specific $K << n$ as output according to the similarity measure. The convergence of LBG algorithm depends on the initial codebook C, the distortion D_k, and threshold, provide a maximum number of iterations to guarantee the convergence.

Figure 3 describes the flow chart of the proposed approach. The objective of this work is to determine the

1. Input training vectors $s=\{x_i \in Rd \mid i=1,2,...,n\}$.
2. Initiate a codebook $C=\{c_j \in Rd \mid j=1,2,...,K\}$.
3. Set D0 = 0 and let k = 0.
4. Classify the n training vectors into K partition cells according to

   $R_i=\{x : d(x,y_i) \le d(x,y_j), \forall j \in I\}$

   Which satisfy condition givenbelow

   $\hat{X}=Q(X) only \quad if \quad d(x,y_i) \le d(x,y_j), \quad \forall j.$
5. Update cluster centers y_i, i = 1, 2, · · · , K

   $y_i = E[X \mid X \in R_i]$

   $y_i = cent(R_i)$
6. Set m ← m + 1 and compute the mean quantization distortion

   $D_m = \frac{1}{N}\sum_{i=1}^{N} D_i$

   and define, for each cell, a utility measure U_i, as the value of distortion D_i of the i^{th} cell, normalized with respect to the mean value D_m:

   $U_i = \frac{D_i}{D_m}$
7. If $d(x,y_i) \le d(x,y_j), \quad \forall j$ repeat steps 4 ~ 6.
8. Output the codebook $c=\{y_i \in Rd \mid j=1,2,....,K\}$

Figure 2. Modified LBG algorithm (MLBG).

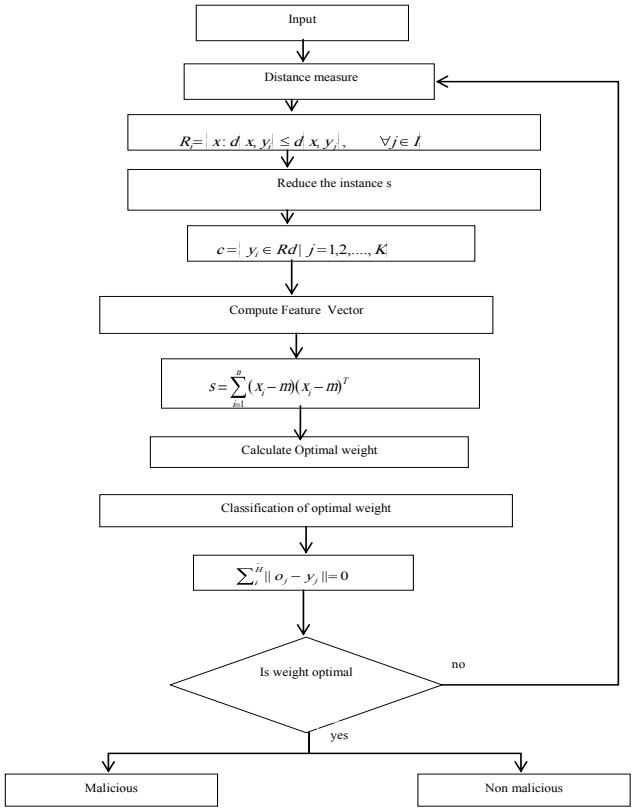

Figure 3. Flow chart of proposed methodology

Input: dataset samples
Output: unknown attacks
Begin:
For each data set
Calculate distance measure

$dist(X,Y)=\sqrt{(x_1-y_1)^2+(x_2-y_2)^2+....,+(x_n-y_n)^2}$

Reduce the instances
For each training vector do
Set D0=0 and k=0
For each training vector do
Set D0=0 and k=0
Classify traing vector

$R_i=\{x : d(x,y_i) \le d(x,y_i), \quad \forall j \in I\}$

Compute mean

$D_m = \frac{1}{N}\sum_{i-1}^{N} D_i$

Output the codebook

$c=\{y_i \in RdJay = 1,2,....K\},$

End for
Compute feature vector

$s=\sum_{i-1}^{n}(x_i-m)(x_i-m)^T$

End for
End for
For each feature do
Calculate weight vector
For each weight vector do

$N=\{(x_i,y_i), x_i \in R^m, \quad i=1,2,....N\}$

For each set
Do hidden layer output matrix h(x).

$f(x)=\sum_{i=1}^{n}\beta_i h_i(x)$

x_j

Compute optimum weight
End for
If weight is optimal
Then
Classify unknown attacks

$\sum_{i}^{H}\|o_j-y_j\|=0$

else
Repeat the step from distance measure
End for

Figure 4. Pseudo code for the proposed approach

malicious traffic in an efficient manner. Initially 90% of the data set is considered as input and remaining 10% is taken for testing. The clustering is performed by the modified LBG algorithm. From the cluster the appropriate features are extracted by the KPCM techniques. Finally the optimistic weight vector is computed by the HGAPSO. The malicious traffic is determined by classifying the weight vector.

2.2 Kernel Principal Component Analysis (KPCA)

The PCA is enhanced by a kernel function which is known as KPCA. Instead of original linear operation of PCA the kernel method reproduce the kernel hilber space by a non-linear mapping. For the feature space $\rho(x_1 ... x_l)$. The equivalent system equation for the KPCA is

$$\tau(\rho(x_k).V) = (\rho(x_k).CV) \text{ for all } k = 1 .... l$$

Where τ eigen values and V is is eigen vector. The procedure of the KPCA is described in the following steps

1. Calculate the covariance matrix S from the input data.

$$C = \frac{1}{l}\sum_{j=1}^{l} \rho(x_j)\rho(x_j)^T$$

2. Compute the Eigen values and eigen vectors of S where eigen vector should be greater than 0 and eigen value should belongs to the feature space F.
3. Normalize the eigen value that belongs to the non-zero eigenvalues corresponding to the vectors in F.

$$1 = \sum_{i,j=1}^{l} \sigma_i^k \sigma_j^k (\rho(x_j).\rho(x_j)) = (\sigma^k.K\sigma^k) = \tau_k(\sigma^k.\sigma^k)$$

Where σ coefficient.

4. Finally kernel function is applied for dot products without performing the actual map (ρ).

$$(V^k.\rho(x)) = \sum_{i=1}^{l} \sigma_i^k (\rho(x_i).\rho(\rho x))$$

The Pseudo code for proposed approach is given in the Figure 4. The list of extracted feature are given in Table 2.

2.3 Hybrid Genetic Algorithm and Particle Swarm Optimization (HGAPSO)

The features extracted have different weight. Some features have priority than other features. In order to gain an optimal weight vector which is defined as $W = [w_1, w_2, ..., w_n], 1 \le i \le n$ where w_i represents the weight of feature. The distance calculation will be influenced by the weight vector W and promote the classifier.

Duration - length (number of seconds) of the connection
sport - Source Port
DPort - Destination Port
Protocol - Type of the protocol, e.g. ARP,IP,UDP,ICMP,TCP,IGMP
Flag - normal or error status of the connection
Urg_Flag - number of urgent packets
hot - number of ``hot" indicators
Wrong_Fragment - number of ``wrong" fragments
Total_length - Size Of the File
error_count - % of connections that have ``SYN" errors
Sequence_number - transmission sequence number Of protocol
Total_packet_number - Total Number Of packets Transfered
Filename - Filename

Figure 5. Attributes in Dataset.

2.3.1 Genetic Algorithm (GA)

GA is a powerful and heuristic search technique that obtained optimistic solution in certain search problems and imitates the natural selection. In a population, GA performs based on the genetic structure and some of the behavior of chromosomes. The GA starts to execute with a set of solutions which is known as population. A new population is formed by taking a solution of another population that should be better than the old one. The solution which is taken as input to form the new solution is based on their fitness. The feasible solution will be provided by performing the mutation, selection and crossover and it is explained below.

2.3.2 Particle Swarm Optimization (PSO)

PSO is a recent and powerful optimization technique. The proper solution is achieved by routinely utilizing this algorithm to improve the solution. PSO optimizes a search problem with a possible solutions. This method follows the natural behavior of bird, fish etc. similar to their natural behavior. Here the population of solution is taken as particles and operates these particles to move around a search space by applying a mathematical formula with particle position and velocity. The working procedure of the PSO is starting to execute by initializing a population of particles with random position and velocity. The initial population is taken from an n-dimensional space. A fitness function is calculated for each

Table 2. List of Extracted Features

No.	Protocol	Features
1	IP	S.IP slots hit
2	TCP	S.IP+S.port slots hit
3	TCP	S.IP+D.port slots hit
4	TCP	S.IP+SYN count
5	TCP	S.IP+URG_Flag+URG_data count
6	TCP	S.IP+ACK_Flag+ACK count
7	IP	S.IP+ARP count
8	IP	D.IP slots hit
9	IP	Headerlength!=20 count
10	IP	MF_Flag count
11	IP	IPTotallength>1400||<40) &&TTL==64 count
12	IP	checksum_error count
13	TCP	S.port slots hit
14	TCP	D.port slots hit
15	TCP	S.port count
16	TCP	D.port count
17	TCP	Sequence_number¼¼0 count
18	TCP	SYN count
19	TCP	URG_Flag+URG_data count
20	TCP	ACK_Flag+ACK count
21	TCP	Checksum_error count
22	TCP	Same_length_interval count
23	TCP	Port(20)+length(>1400) count
24	UDP	S.port count
25	UDP	D.port count
26	UDP	checksum_error count
27	UDP	Same_length_interval count
28	ICMP	Type error count
29	ICMP	checksum_error count
30	ICMP	Length>1000 count
31	IGMP	Type error count
32	IGMP	checksum_error count
33	IGMP	Length>1000 count
34	ARP	Size error count
35		Total packet number
36	PV	Variance of payload packet length for time interval
37	PX	Number of packets exchanged for time interval

particle and compared with pbest. If better fitness value is achieved, then it is added to the pbest. A global solution can be computed among the pbest through the modified position and velocity of the particle. This process executes routinely to obtain a feasible solution. The PSO provides optimistic solution by having low memory consumption, low CPU resource.

2.4 HGAPSO

PSO and GA algorithm shares many similarities. Both techniques start with a group of random variable, and both apply a fitness value evaluation of population. The population and search will be updated for the optimum solution in a repeated manner. A combination of PSO and GA search abilities is given in one algorithm in this work with four operator enhancement, selection, crossover, and mutation. The description of these operators is given in the following sections.

2.5 Enhancement

The best performing fitness values are marked after the calculation of fitness values for every individual in the population where these individuals are noticed as elites. In this part we first enhance the elite instead of regenerating them directly to the next generation. In this operation individuals become suitable to the environment after receiving the required information from the society. The enhanced elites are used as parents so the created progeny will obtain a better performance than original elites. This enhancement is performed by the velocity and position update procedures in PSO.

The stability condition in PSO is defined by

$$C_1 r_1 + C_2 r_2 > 0$$
$$\frac{C_1 r_1 + C_2 r_2}{2} - \omega < 1$$
$$\omega < 1$$

Knowing that $\omega \in [0,1]$, the following parameter selection heuristic was derived

$$0 < C_1 + C_2 < 4$$
$$\frac{C_1 + C_2}{2} - 1 < \omega < 1$$

The parameters used in the above conditions are considered based on necessary searching ability to solve each problem. The newly generated elites corresponds to both groups of the previous generation, i.e., the

enhanced elites. Consider the parent of previous generation and produce the elites by setting t_{iv} to zero, and t_{ip} is set to t_{ix}, i.e., the newly generated individual itself. Otherwise, t_{ip} records the best solution of individual i evolved so far.

2.6 Selection

Selection procedure is used to choose the individuals to the next generation that depend upon the fitness of each individual. The fitness value determination is performed by objective function. The transformation between the populations is made by allowing the individuals to pass on the gene by giving higher preference to the better individuals. The enhanced elite produced by the PSO is then applied for GA operation in the proposed work. The tournament selection procedure is used in this work to select the parent for the next GA operation. The enhanced elites are selected randomly, and the fitness value of these elites is compared. The elites with better fitness values are placed in the mating pool.

2.7 Cross over

Crossover provides a solution by taking more than one parent solution as input and gives a child solution. The selections of parents are performed randomly from the mating pool and crossover procedure is performed to create a next generation. The children spreading over the parent solutions is performed by Simulated Binary Crossover (SBX) that is maintained by distribution index, c_{η}. The arbitrary neighboring region can be searched by Appling the SBX operator, the required diversity is maintained among the feasible parent solutions.

$$c1 = x - 1/2\,\beta(p2 - p1)$$
$$c2 = x + 1/2\,\beta(p2 - p1)$$
$$where \quad x = 1/2\beta(p2 + p1), \quad p2 > p1, c = x$$

2.8 Mutation

Mutation is the final operation in genetic algorithm, the objective of this operator to provide the diversity of the population and to maintain a genetic diversity. The final genetic operator is mutation. In order to maintain the population's diversity mutation creates a novel genetic material in the population. It is not necessary to apply the mutation for all population. According to the fitness value the mutation probability (m_p) is allocated to the individual.

$$P_{mi} = 0.5 \times \left[\frac{F_{\max} - F_i}{F_{\max} - F_{ave}}\right] \qquad if \quad F_i \geq F_{ave}$$
$$P_{mi} = \left[\frac{F_{ave} - F_i}{F_{\max} - F_{ave}}\right] \qquad if \quad F_i < F_{ave}$$

F_i denotes the fitness value of the individual i, The $F_{\max}$ denotes the highest fitness value whereas F_{ave} refers the average fitness values of the population in each generation. A random number in the range [0, 1] is created after assigning the P_{mi}, for each generation. The individuals having a P_{mi} greater than this number is mutated. Variable dependent random mutation is applied in this approach. Vicinity of the parent solution with a uniform probability distribution is created as given below.

$$x_i^{(1,t+1)} = x_i^{(1,t)} + (r_i - 0.5)\Delta_i$$

r_i refers the random number in a range of [0, 1]. Δ_i is denoted as user defined maximum perturbation that is allowed in the ith decision variable(x_i). At each generation Δ_i, x_i is computed in two ways one is by applying the average of that variable and second method is by applying the difference between its maximum and minimum in the population, i.e.

$$\Delta_i = 0.5 \times \left(\max\left(x_i\right) - \min\left(x_i\right)\right)$$
$$\Delta_i = \left(0.025 \sim 0.075\right) \times ave\left(x_i\right)$$

The progeny and the enhanced elites from PSO generates the new population after applying the GA operators. Then elites for the next generation is selected by evaluating the fitness of those elites.

2.9 Kernel Extreme Learning Machine (KELM)

KELM provides a better performance for classification and regression problems, which is applied in this approach to improve the classification accuracy. KELM is a heuristic algorithm for the Single Hidden Layer Feed Forward Network (SLFN). With a fast learning speed he KELM improves the classification accuracy.

The output function g(x) of the KELM is written

$$g(x) = [k(x, x_1), \ldots k(x, x_N)]\left(\frac{1}{c} + M_c\right)^{-1} T$$

Where M_c are Mercer's conditions used to identify the feature mapping h(x).

It randomly chooses hidden nodes ($\tilde{N}$), input weights (w_i), hidden layer biases (b_i) and output weights (o_j) of SLFNs using the least square method. The SLFN with $\tilde{N}$ hidden neurons and activation function f(x), Number of training set (N), Input features set X [x_1, x_2, x_n]

Input: Training set and testings e as $N = \{(x_i, y_j), x_i \varepsilon R^n, y_i \varepsilon R^m, i = 1, ..., N\}$, weights ($w_i$), biases b_i, Kernel function f(x), and hidden neuron $\tilde{N}$,

Step 1: Hidden layer node $\tilde{N}$, assign input weights w_i **randomly** and hidden layer biases b_i, (i = 1, 2, …$\tilde{N}$)

Step 2: Calculate the hidden layer output matrix h(x).

$$g(x) = [k(x, x_1), ...k(x, x_N)]\left(\frac{1}{c} + M_c\right)^{-1} T$$

Step 3: For hidden layer, output weight $\tilde{N}$ can be calculated using sigmoid function $sg(x) = \frac{1}{H} g(x_i), g(x_j)$

Step 4: To solve linear equation $H\beta = T$

3. Experiment and Result

Dataset consists of 207 attack program which is collected from the website VX Heaven Virus Collection. This collection is taken from kaspersky lab in http://academic-torrents.com/details. Dataset capacity is 44.59GB which contains 47,880,726,923 bytes of data. The dataset contains various attributes.

1.48 GB of whole dataset has been used in this experiment. Due to the problems in the operating system version and missing of some specific DLL files, only 84 flooding programs have been used in Windows XP platform for experimentation. The list of 84 attack programs is shown in Table 3. Three types of dataset used in this experiment. They are sampling dataset, training and testing dataset. Attributes used in data set is shown in Figure 5. The sampling dataset is considered for calculating the distance in classification phase. The classifier and optimization technique trains the dataset. Finally the test dataset was used to evaluate the performance of the classifier.

Table 3. Attack Programs

Backdoor.Perl.IRCBot.AC	Backdoor.PHP.WebShell.AE	Backdoor.IRC.Kelebek.A;	Backdoor.IRC.Kelebek.AG
Backdoor.IRC.Mox.B	Backdoor.IRC.Flood.Q;	Backdoor.IRC.Kelebek.AA;	Backdoor.PHP.Rst.I
Backdoor.PHP.Agent.U;	Backdoor.Perl.RShell.B;	Backdoor.IRC.Kelebek.AB;	Backdoor.PHP.Agent.AJ;
Backdoor.PHP.Agent.AK;	Backdoor.PHP.Agent.AS;	Backdoor.IRC.Kelebek.AC;	Backdoor.PHP.Agent.BO;
Backdoor.PHP.Agent.U;	Backdoor.PHP.Agent.W;	Backdoor.IRC.Kelebek.AJ;	Backdoor.IRC.Watal.C;
Backdoor.IRC.Watal.A	Backdoor.ASP.Ace.DI;	Backdoor.IRC.Kelebek.C;	Backdoor.ASP.Ace.E;
Backdoor.ASP.Ace.FR;	Backdoor.ASP.Ace.I;	Backdoor.IRC.Kelebek.F;	Backdoor.ASP.Ace.AC;
Backdoor.ASP.Ace.AJ;	Backdoor.ASP.Ace.AR;	Backdoor.IRC.Kelebek.H;	Backdoor.ASP.Ace.B;
Backdoor.ASP.Ace.BA;	Backdoor.ASP.Ace.BF;	Backdoor.IRC.Kelebek.I;	Backdoor.Perl.Small.E;
Backdoor.IRC.Kelebek.AG;	Backdoor.IRC.Kelebek.G;	Backdoor.IRC.Kelebek.J;	Backdoor.IRC.Kelebek.G;
Backdoor.IRC.Kelebek.L;	Backdoor.IRC.Kelebek.O;	Backdoor.IRC.Kelebek.Q;	Backdoor.IRC.Kelebek.R;
Backdoor.IRC.Kelebek.U;	Backdoor.IRC.Kelebek.W;	Backdoor.IRC.Kelebek.X;	Backdoor.IRC.Zapchast.I;
Backdoor.IRC.Zapchast.ZWQX;	Backdoor.IRC.Zapchast.A	Backdoor.IRC.Zapchast.B;	Backdoor.IRC.Zapchast.E;
Backdoor.IRC.Zapchast.F;	Backdoor.IRC.Zapchast.G;	Backdoor.IRC.Zapchast.I;	Backdoor.IRC.Zapchast.J;
Backdoor.IRC.Zapchast.K;	Backdoor.IRC.Zapchast.L	Backdoor.IRC.Zapchast.ZWQX;	Backdoor.IRC.Cloner.J;
Backdoor.IRC.Cloner.N;	Backdoor.IRC.Cloner.110;	Backdoor.IRC.Cloner.20;	Backdoor.IRC.Cloner.30;
Backdoor.IRC.Cloner.AB;	Backdoor.IRC.Cloner.AE;	Backdoor.IRC.Cloner.AF;	Backdoor.IRC.Cloner.AH;
Backdoor.IRC.Cloner.C;	Backdoor.IRC.Cloner.E;	Backdoor.IRC.Cloner.F;	Backdoor.IRC.Cloner.G;
Backdoor.IRC.Cloner.H;	Backdoor.IRC.Cloner.J;	Backdoor.IRC.Cloner.K;	Backdoor.IRC.Cloner.L;
Backdoor.IRC.Cloner.N;	Backdoor.IRC.Cloner.O;	Backdoor.IRC.AH;	Backdoor.IRC.Cloner.J;

3.1 Performance Metrics

The proposed methodology can be evaluated by performance metrics such as time and accuracy. ELM classification time for the proposed methodology has been given in the Table 4 and Figure 6. Formula for execution time is given below.

Execution Time = starting time of sampling reduction process-ending time of classifier

From the Figure 6 and Table 4, it is clearly understood that the proposed classifier take less execution time than existing method. ELM and KNN classification training accuracy without clustering for proposed methodology and existing method has been given in the Table 5 and Figure 7. It is represented by percentage (%) It is also clearly understood that the proposed classifier takes high training accuracy than existing method.

ELM and KNN classification training accuracy with clustering for proposed methodology and existing method has been given in the Table 6 and Figure 8. It is represented by percentage (%).

From the Figure 8 and Table 6, it is clearly understood that the proposed classifier takes high training accuracy than the existing method. ELM and KNN classification testing accuracy without clustering for proposed methodology and existing method has been given in the Table 7 and Figure 9. It is represented by percentage (%)

From the Figure 9 and Table 7, it is clearly understood that the proposed classifier takes high testing accuracy than the existing method. ELM and KNN classification testing accuracy with clustering for proposed methodology and existing method has been given in the Table 8 and Figure 10. It is represented by percentage (%)

From the Figure 10 and Table 8, it is clearly understood that the proposed classifier takes high testing accuracy than existing method

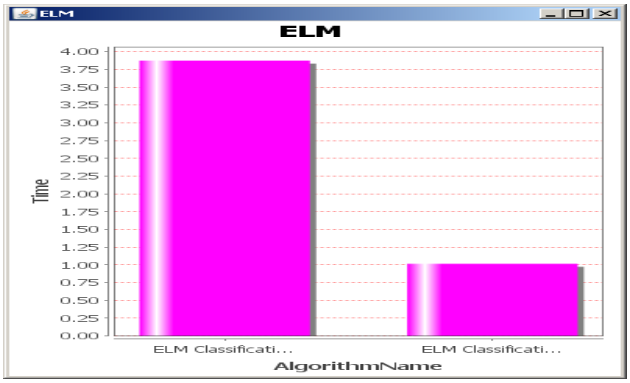

Figure 6. ELM Classification time

Table 4. kELM Classification time

KELM Classifier	Time in seconds
Testing time	1.016
Training time	3.875

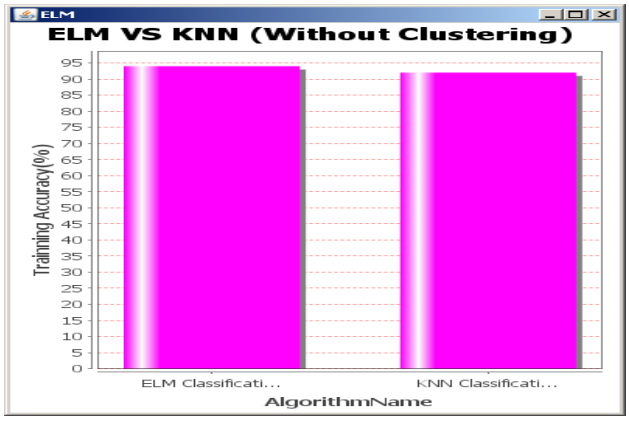

Figure 7. Training accuracy for classifiers.

Table 5. Training accuracy for classifiers

Classifier	Accuracy (%)
KELM training accuracy	94
KNN training accuracy	92

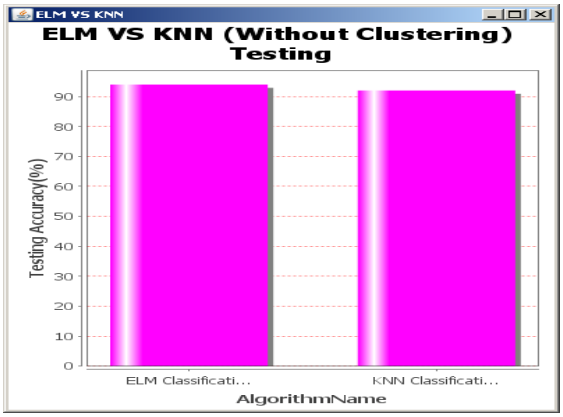

Figure 8. Training accuracy for classifiers with clustering.

Table 6. Training accuracy for classifiers with clustering

Classifier	Accuracy (%)
KELM training accuracy	95
KNN training accuracy	93

Table 7. Testing accuracy for classifiers

Classifier	Accuracy (%)
KELM testing accuracy	94
KNN testing accuracy	92

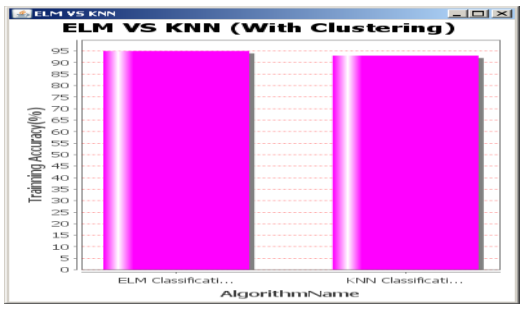

Figure 9. Testing accuracy for classifiers.

Table 8. Testing accuracy for classifiers with clustering

Classifier	Accuracy (%)
KELM testing accuracy	95
KNN testing accuracy	93

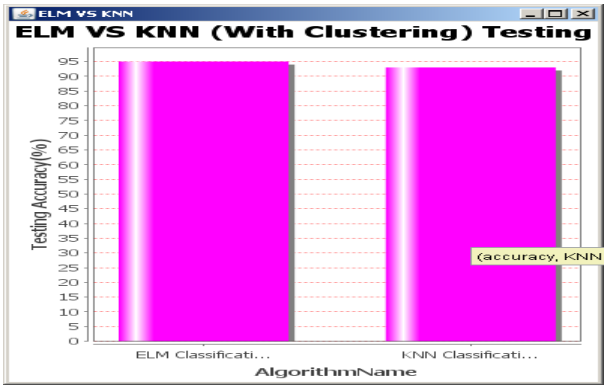

Figure 10. Testing accuracy for classifiers with clustering.

4. Conclusion

DoS attack makes resource unavailable to its users and thus it is a major threat in current computer networks. This paper proposed hybrid optimization and extreme learning machine classifier based anomaly detection to detect the malicious traffic flow in network. The training and testing time of classifier is reduced by using the MLBG. Kernel Principal Component Analysis (KPCA) is used to extract the feature from the dataset. A short time window is utilized to gather all features from packet headers. Extreme learning machine based hybrid optimization (HGAPSO) is used to classify the unknown attack. Likewise the proposed approach is implemented as real-time.

The proposed method gives better detection accuracy in training and testing and time computation for detection of attacks compared to that of existing approach.

5. References

1. Visumathi J, Shunmuganathan KL. A computational intelligence for evaluation of intrusion detection system. Indian Journal of Science and Technology. 2011 Jan 14; (1): 40–5.
2. Sundaram A. An introduction to intrusion detection. Crossroads. 1996 Apr 1; 2(4): 3–7.
3. Saravanan C, Shivsankar MV, TamijeSelvy P, Anto S. An optimized feature selection for intrusion detection using layered conditional random fields with MAFS. Int J Mob Netw Commun Telematics. 2012; 2(3): 79–91.
4. Saboori E, Parsazad S, Sanatkhani Y. Automatic firewall rules generator for anomaly detection systems with Apriori algorithm. In Advanced Computer Theory and Engineering (ICACTE). 2010 3rd International Conference on IEEE; Chengdu. 2010 Aug 20-22; 6: 6–57.
5. Javidi MM, Rafsanjani MK, Hashemi S, Sohrabi M. An overview of anomaly based database intrusion detection systems. Indian Journal of Science and Technology. 2012 Oct 1; 5(10): 3550–9.
6. Mahmood DY, Hussein MA. Feature Based Unsupervised Intrusion Detection. World Academy of Science, Engineering and Technology. International Journal of Computer, Electrical, Automation, Control and Information Engineering. 2014 Nov 4; 8(9): 1633–7.
7. Rajeswari LP, Arputharaj K. An active rule approach for network intrusion detection with enhanced C4. 5 Algorithm. International Journal of Communications, Network and System Sciences. 2008 Nov 1; 1(4): 314.
8. Chang RK. Defending against flooding-based distributed denial-of-service attacks: a tutorial. Communications Magazine. IEEE; 2002 Oct; 40(10): 42–51.
9. Sharifi AM, Amirgholipour SK, Alirezanejad M, Aski BS, Ghiami M. Availability challenge of cloud system under DDOS attack. Indian Journal of Science and Technology. 2012 Jun 1; 5(6): 2933–7.
10. Su MY. Using clustering to improve the KNN-based classifiers for online anomaly network traffic identification. Journal of Network and Computer Applications. 2011 Mar 31; 34(2): 722–30.

11. Sarma DS, Sarma SS. A Survey on Different Graph Based Anomaly Detection Techniques. Indian Journal of Science and Technology. 2015 Nov 11; 8(1).
12. Katkar VD, Kulkarni SV. Experiments on detection of Denial of Service attacks using Naive Bayesian classifier. Green Computing, Communication and Conservation of Energy (ICGCE). 2013 International Conference on IEEE; 2013 Dec 12. p. 725–30.
13. Tan Z, Jamdagni A, He X, Nanda P, Liu RP. A system for denial-of-service attack detection based on multivariate correlation analysis. Parallel and Distributed Systems, IEEE Transactions on. 2014 Feb; 25(2): 447–56.
14. Yu S, Zhou W, Jia W, Guo S, Xiang Y, Tang F. Discriminating DDoS attacks from flash crowds using flow correlation coefficient. Parallel and Distributed Systems, IEEE Transactions on. 2012 Jun; 23(6): 1073–80.
15. Tan Z, Jamdagni A, He X, Nanda P, Liu RP, Hu J. Detection of denial-of-service attacks based on computer vision techniques. Computers, IEEE Transactions on. 2015 Sep 1; 64(9): 2519–33.
16. Yu J, Lee H, Kim MS, Park D. Traffic flooding attack detection with SNMP MIB using SVM. Computer Communications. 2008 Nov 20; 31(17): 4212– 9.
17. Uddin M, Alsaqour R, Abdelhaq M. Intrusion detection system to detect DDoS attack in gnutella hybrid P2P network. Indian Journal of Science and Technology. 2013 Feb 1; 6(2): 4045–57.
18. Renjit JA, Shunmuganathan KL. Network based anomaly intrusion detection system using SVM. Indian Journal of Science and Technology. 2011 Sep 1; 4(9): 1105–8.