International Journal of Intelligent Computing and Technology

IJICT

ISSN: 2457-0249 (Online)

Website: https://ijict.com 2019 (Jan)-Vol.2, Iss.2 Page: 32-46

An Enhanced Method for Reducing the Computational Complexity of Distributed Denial of Service using LAMSTAR Principal Component Analysis

Dr M Thangavel[1], A Rajesh[2], M Parameswari

[1]Professor, [2]Assistant Professor, Department of Computer Applications, Erode Sengunthar Engineering College, Perundurai, Tamilnadu, India

[2]Associate Professor, Department of Computer Science, Erode Arts and Science College, Erode, Tamilnadu, India

thangavelpamu@gmail.com

Article History- Received: Dec 2018; Published: January 2019

ABSTRACT

The security of computer networks plays a strategic role in modern computer systems. Distributed Denial of Services (DDOS) act as the ‘‘second line of defense’’ placed inside a protected network and looking for known or potential threats in network traffic and/or audit data recorded by hosts. The system is developed a Distributed Denial of Service attacks detection method using LAMSTAR neural network to classify observed system activities and compared the performance of LAMSTAR method. LAMSAR method gives better performance at the cost of testing time high Computational complexity and Training time than the other classification techniques. The system is further reduced the dimension of the data using principal component analysis by reducing the Computational Complexity of LAMSTAR DDOS which in turn reduces the testing and training time with almost the same performance

KEYWORDS: Distributed Denial of Service, LAMSTAR, PCA, False alarm rate

INTRODUCTION

Computer security has become a critical issue with the rapid development of business and other transaction systems over the internet. DDoS attack detection is to detect intrusive activities while they are acting on computer network systems. There are two major DDoS attack detection techniques: misuse detection and anomaly detection [1]. Misuse detection discovers attacks based on the patterns extracted from known DDoS attacks. Anomaly detection identifies attacks based on the deviations from the established profiles of normal activities. Activities that exceed thresholds of the deviations are detected as attacks. Misuse detection has low false positive rate, but cannot detect new types of attacks. Anomaly detection can detect unknown attacks, under a basic assumption that attacks deviate from normal behavior. The system developed a Distributed Denial of Service using LAMSTAR neural network to learn patterns of normal and intrusive activities, to classify observed system activities. LAMSAR DDOS gives better performance at the cost of high Computational complexity, Training time and testing time, when compared to other classification techniques. we further reduced the computational complexity of LAMSTAR DDOS by reducing the dimension of the data using principal component analysis which in turn reduces the, training and testing time with almost the same performance. This paper is organized as follows: Section 2 gives some theoretic background about LAMSTAR Neural Network. Section 3 presents the details about the testing and training the Distributed Denial of Service, and the cost matrix used to calculate cost per example. Section 4 gives details about the principal component analysis. In Section 5 we discuss the dimension reduction of data and performance of various algorithms in classifying the data. The paper is finally concluded in section 6 with the most essential points.

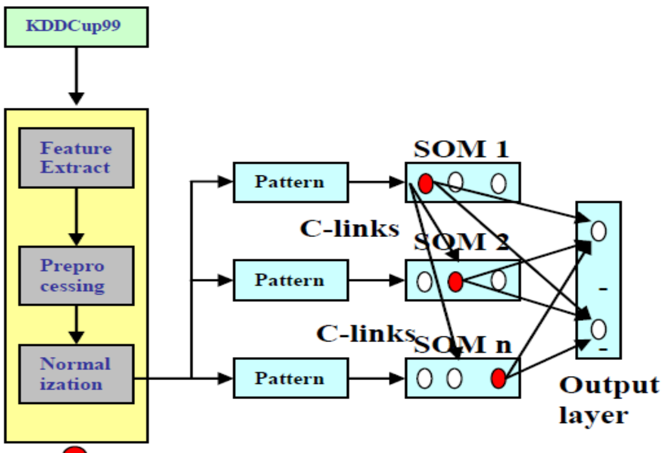
LAMSTAR

A Large-Scale Memory Storage and Retrieval (LAMSTAR) network is proposed in [2] by combining SOM modules and statistical decision tools. LAMSTAR was specifically developed for application to problems involving very large memory that relates to many different categories (attributes) where some data is exact while the other is fuzzy and where for a given problem some categories might be totally missing [3]. Large Scale Memory Storage and Retrieval (LAMSTAR) network research, which targets large-scale memory storage and retrieval problems. This model attempts to imitate the processes of the human central nervous system (CNS) concerning storage and retrieval of patterns and its impressions, and sensed observations including processes of forgetting and recollection. It attempts to achieve this without contradicting findings from physiological and psychological observations in an

input/output manner. Furthermore, it attempts to do so in a computationally efficient manner using tools of neural networks, especially Self-Organizing-Map based (SOM) network modules are combined with statistical decision tools. And its design was guided by trying to find a mechanistic neural network-based model for very general storage and retrieval processes involved. This general approach is related to Minsky's idea that the human brain consists of many agents, and a knowledge link is formed among them whenever the human memorizes an experience. When the knowledge link is subsequently activated, it reactivates the mental agents needed to recreate a mental state similar to the original. The LAMSTAR network employs this general philosophy of linkages between a large number of physically separate modules that represent concepts, such as time, location, patterns, etc., in an explicit algorithmic network.
The LAMSTAR network has been successfully applied in fields of medicine (diagnosis) [4, 5, 6], engineering (automotive fault detection) and multimedia information systems [7]. Whereas the LAMSTAR design addresses large-scale memory retrieval problems, system use LAMSTAR concepts to processes of storage and retrieval, interpolation and extrapolation of input data, and the use of reward-based correlation-links between modules to detect DDoS attacks. In this modified LAMSTAR network, each Kohonen SOM module represents a class of sub-patterns. The model assumes that the input patterns have been separated into sub-patterns before entering the SOM module. The network is thus organized to assign each neuron to a class of neurons (i.e., one SOM module) that best corresponds to the input sub-pattern. This SOM configuration yields very rapid matching with good error tolerance, and is capable of generalization. Arrays of correlation links (C-links) connect the modules using coefficients determined by the statistical correlations between the various patterns considered. A coordinated activation of neurons between the various modules allows the network to recreate (interpolate) complex patterns and make associations.

LAMSTAR DDOS DESIGN

A modified LAMSTAR network used for DDoS attack detection is as shown in Fig. 1. The model reads in KDDCup 99 data sends it first to the feature extraction module which extracts 41 features of the data and sends it to preprocessing module. The preprocessing module converts the 41 features into a standardized numeric representation. Normalization block reads the preprocessed data and normalizes the data into a format required by the SOM's. The normalized input pattern was split into sub patterns (basic features 9, content features 13, traffic 9, and others 10) [8]. Each sub pattern is given to one SOM module. This SOM configuration yields very rapid matching with good error tolerance, and it is capable of generalization.



● Winning Neuron; for clarity not all C-links are shown
Fig. 1 Modified LAMSTAR architecture

Between SOM modules, connections are established using correlation links. The correlation links distribute information between various modules. The training data contains 22 attack patterns and normal patterns. The SOM modules are trained using this pattern. The coordinated activation of neurons between the various modules allows the network to detect DDoS attacks. The input pattern stored as a real vector x given by:

$$X = [\, x1T, \ldots . xiT, \ldots . xmT \,]T \tag{1}$$

To store data concerning the i'th category of the input pattern, each sub-pattern x is channeled to the corresponding i'th SOM module. A winning neuron is determined for each input based on the similarity between the input vector x and i weight vectors w (stored information). For a sub-pattern x, the winning neuron is determined by the minimum Euclidean distance between x and w:

$$||x^i - w^i{}_{\text{winner}} \qquad = min: ||X^i - W^i{}_k \qquad ||\forall \;\; k \tag{2}$$

Where *x* - input vector in i'th SOM module winner -index of the winning neuron

$W^i{}_{\text{winner}}$ - winner weight vector in ith SOM module

k - a number of neurons(stored pattern)in ith SOM module

||-|| - Vector Euclidean distance

$$||X - W\,|| = \sum_{i=1}^{i=n} (\, W_{t-}\, X^t\,)^2$$

where n - dimension of sub vectors x and w.

The SOM module is a Winner-Take-All [9] network where only the neuron with the highest correlation between its input vector and its correspondence weight vector will have a non-zero output. The Winner take all features also involves lateral inhibition such that each neuron has a single positive feedback onto itself and negative feedback connections to all other units.

$$O^{i}{}_{j} = \begin{cases} 0 \\ 1 \text{ for } \|X^{t} - W^{i}\text{winner}\| < \|X^{i} - W^{i}{}_{j}\| \end{cases}$$

¥ winner ≠ 0 otherwise

Where

O_j^i output of neuron j in ith SOM module

w^i Winning weight vector in the ith SOM Module

Winner - index of the winning neuron is the SOM Module. The neuron with the smallest error determined is declared the winner and its weights Wwinner are adjusted using the Hebbian learning law,:

$$w^i\, winner(t+1) = w^1\, winner(t)_0 + \alpha(x^t(t) - w^1\, winner(t)) \quad (4)$$

α - Learning rate a slowly decreasing function of time and initial weights are assumed with random values. The learning rate is updated by, α (t+1) = 0.5α (t).

The adjustment in the LAMSTAR SOM module i weighted according to a pre-assigned variable with neighbourhood function Δ (winner, j).

$$\boldsymbol{W}^{i}\mathrm{j}(\mathrm{t}+1) = \boldsymbol{W}^{i}\mathrm{j}(\mathrm{t}) + \Delta(\mathrm{winner}, \mathrm{j}).\,\alpha(\boldsymbol{x}^{i}(\mathrm{t}) - \boldsymbol{W}^{i}\mathrm{j}(\mathrm{t}) \quad (5)$$

Where j(t+1) - new weight of the neighbor neuron j from winning neuron.

Δ (winner, j) – neighborhood defined as variable.

Training Phase

The training of the SOM modules are done as described below SOM modules are trained with sub-patterns derived from the Attack data. Given an input pattern x and for each x sub-pattern to be stored, the network inspects all weight vectors win the I'th SOM module. If any previously If any previously stored pattern matches the input sub-pattern within a preset tolerance (error ε), the system updates the proper weights or creates a new pattern in the SOM module. The choice of ε‘s value depend on the size of the cluster. The following expression is used to calculate the value of ε

$$\varepsilon = MAXx\varepsilon.cli.dist(x, ci)\ /\ 10 \qquad (6)$$

where ci is the cluster center and cli is the cluster i. It stores the input sub-pattern x as a new pattern, x = w, where index j is the first unused k neuron in i'th SOM module. If there are no more ‘free’ neurons, the system will fail, which means either the preset tolerance has to be increased to include more patterns in the same cluster of already stored patterns and more neurons have to be added on the i ’th SOM module.

Correlation links C-links among SOM modules are created as follows. Individual neurons represent only a limited portion of the information input. Sub-patterns are stored in SOM’s and the correlation links between these sub-patterns are established in such a way that the information’s are distributed between neurons in various SOM modules and correlation links. Even if one neuron fails only a little information information is lost since the information is spread among SOM’s and correlation links. Correlation link coefficient values C-link are determined by evaluation distance minimization to determine winning neurons, where a win activates a count up element associated with each neuron and with its respective input-side link. During training sessions, the values of c links are modified according to the following simple rule (reward)

$$c_{(i,l)}^{(k,j)}\ (new) = c_{(i,l)}^{(k,j)}\ (old) - \beta_{reward}(c_{(i,l)}^{(k,j)}\ (old)\ ) - C_{Max}), for$$
$$c_{(i,l)}^{(k,j)}\ (old) \neq 0, 1\ otherwise \qquad (7)$$

Where $c_{(i,l)}^{(k,j)}$ - Correlation link between k’th neuron in the i’th SOM module and the i’th neuron in j’th SOM.

β reward-reward coefficient, initially value is assumed with some random values. Breward (t+1)=.5. β reward (t).

To keep the link-weights within a reasonable range, whenever the highest of all weights reaches a certain threshold all link- weights to that SOM are uniformly reduced by the same proportion, for example 50% additionally, link-weights are never reduced to zero or the connection between the two neurons will be lost permanently. If the correlation link between two sub-patterns already exists, namely, i result from previous training), the formula of equation 6 updates (increases) the analyzed C-link. If there are no correlations (Ckj = 0), the system creates new system C-link with initial value k" I = 1

Detection Phase

The sub-patterns from input pattern is selected and the correlations with stored sub-patterns in each SOM module is examined. For example, one i'th SOM module could have previously stored source IP address, and will correlate any given input i'th sub-pattern and determine if there is a match or not. The Intruder packet is detected by means of its C- links. Once all the winning neurons are determined, the system obtains all correlation-links coefficient values among all SOM modules. The Output SOM layer (Fig.1) with which all C-links are interconnected, will determine whether the input pattern is an intruder packet or a normal packet.

FEATURE EXTRACTIONS AND PREPROCESSING

The input data to the neural network must be in the range [0 1] or [-1 1]. Hence preprocessing and normalization of data is required. The format data is preprocessed, each of one which is in continuous, discrete and symbolic form, with significantly varying ranges, based on type of neural nets, the input data has different forms and so needs different preprocessing. Some neural nets only accept binary inputs and some can also accept continuous-valued data in preprocessor, after extracting features from each record, each feature is converted from text or symbolic form into numerical form. For converting symbols into numerical form an integer code is assigned to each symbol.

Normalizations

For normalizing feature values, a statistical analysis is performed on the values of each feature based on the existing data and the acceptable minimum value for each feature is determined. According to the maximum values and the following simple formula, normalization of feature values in the range [0, 1] is calculated.

$$If\ (f\ >\ MaxF\ )\ Nf = 1;\ Otherwise\ Nf\ =\ (f\ /\ MaxF) \qquad (8)$$

Cost Matrix

A cost matrix (C) is defined by associating classes as labels for the rows and columns of a square matrix: in the current context for the Attack dataset of DDoS Attack and therefore the matrix has dimensions of 5×5. An entry at row i and column j, C(i,j), represents the nonnegative cost of misclassifying a pattern belonging to class i into class j. Cost matrix values employed for the Attack defined also in [11]. These values were also used for evaluating results of the Attack competition. T magnitude of these values was directly proportional to the impact on the computing platform under attack if a test record was placed in a wrong category. A confusion matrix (CM) is similarly defined in that row and column labels are class names: a 5×5 matrix for the Attack data. An entry at row i and column j, CM (i,j), represents the number of misclassified patterns, which originally belong to class i yet mistakenly identified as a member of class j. Given the cost matrix as predefined in [11] and the confusion matrix obtained subsequent to an empirical testing process, cost per example (CPE) was calculated using the formula,

$$CPE = \frac{1}{N}\sum_{i=1}^{5}\sum_{j=1}^{5} CM(i,j) * C(i,j) \tag{9}$$

Where CM corresponds to confusion matrix, C corresponds to the cost matrix, and N represents the number of patterns tested. A lower value for the cost per example indicates a better classifier model. Comparing performances of classifiers for a given attack category is implemented through the probability of detection along with the false alarm rate, which are widely accepted as standard measures. The confusion matrices in the above, columns correspond to predicted categories, while rows correspond to actual categories.

The software tool LNKnet, which is a publicly available pattern classification software package [12], was used to simulate pattern recognition and machine learning models. The LAMSTAR was simulated using JNNS [13] software tool.

Standard Metrics for Evaluations of DDoS attacks (Attacks)

The system evaluated the performance of various DDOS system based on the detection rate: detecting normal traffic from attack and recognizing the known attack type False alarm rate: misdetection attack [14].

PRINCIPAL COMPONENT ANALYSIS

Principal Component Analysis (PCA) [15,16,17] is one of the most widely used dimensionality reduction techniques for data analysis and compression. Using PCA, patterns in data has been identified and also expressing the data to highlight their similarities and differences. Since patterns in data can be very hard to find in data of high dimensions, PCA is a powerful tool for analyzing data. Once patterns in the data are found, data can be compressed by reducing the number of dimensions without loss of information..

Given the Attack data, each data has 41 features represented by x11x12…x14, x21, and x22….x241 and so on. The data set can be represented by a matrix xnxm. The average observation is defined as

$$\mu = \frac{1}{n}\sum_{i=1}^{n} x_i \quad (10)$$

The deviation from the average is defined as

$$\varphi I = x_i - \mu \quad (11)$$

The sample covariance matrix of the dataset is defined as

$$C=\frac{1}{n}\sum_{i=1}^{n} (X_i - \mu)(X_i - \mu)^T = \frac{1}{n}\sum_{i=1}^{n} \varphi_i \varphi_i^T = \frac{1}{n} A A^T \quad (12)$$

Eigen values and vectors of the sample covariance matrix C usually computed by the Singular Value Decomposition. Suppose (A1,U1), (A2,U2)---- (Am,u.) are m eigen value-eigen vector pairs of the sample covariance matrix C. The k eigenvectors having the largest eigen values are selected. The dimensionality of the subspace k can be determined by

$$\frac{\sum_{i=1}^{k} \lambda i}{\sum_{i=1}^{m} \lambda i} \geq 0 \quad (13)$$

where α is the ratio of the variation in the subspace to the total variation in the original space. A mxk matrix U is formed whose columns consists of the k eigenvectors. The principal components represents the consist of projecting the data onto the k-dimensional subspace according to the following rules.

$$Yi = U^T (X\,i - \mu) = U^T \varphi_i \quad (14)$$

PERFORMANCE RESULT AND DISCUSSION

This study helps to analyze the packet information and filter it based on the available information. It feeds the information in the packet only once when it enters into the first router in the network. The computational burden and comparison of scalability with different techniques is shown below in Figure 2.

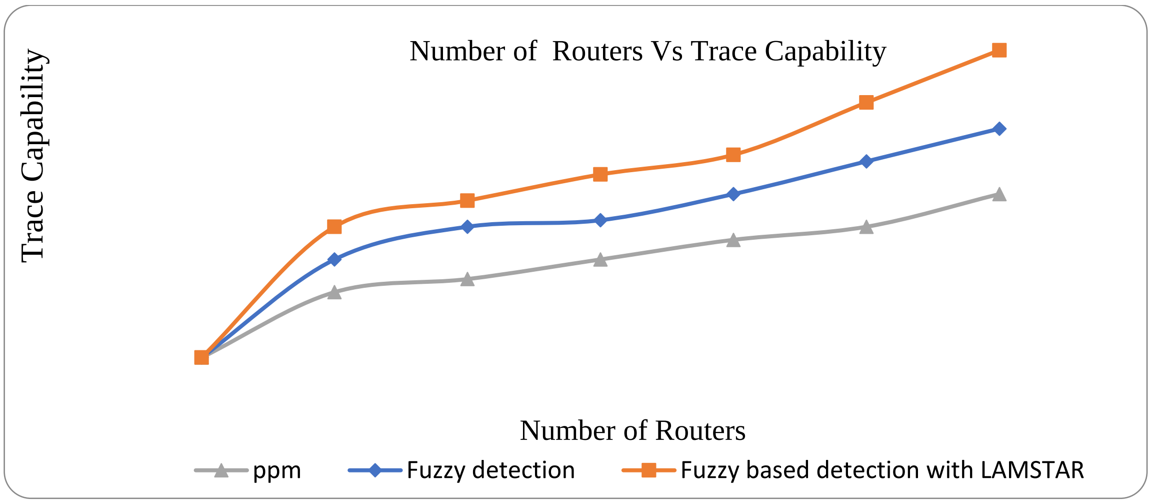


Fig2. Trace Capability Vs Number of routers

As a result, the fuzzy based detection technique using LAMPSTAR network stands best among the various other existing techniques. It utilizes the available path information for tracing the source system and hence the traceability is improved. It enables the router to reduce the overhead in packet forwarding and hence the tracing is easy. The performance comparison based on the number of routers traced is shown in Figure 1. The result shows that the performance of the other existing techniques reduces as the number of routers, when the packet crossed increases.

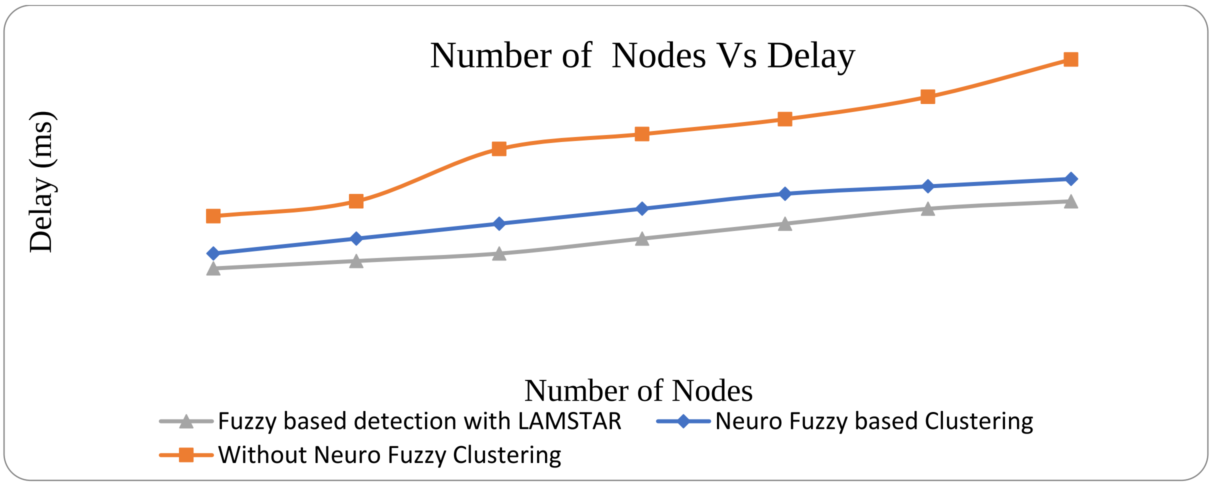


Fig3. Number of nodes Vs Delay

The simulation result based on the nodes variation affecting the propagation delay is depicted in the Figure 3. The propagation delay increases as the number of nodes increased in the anonymous traffic network. It indicates that Neuro-Fuzzy detection with LAMPSTAR technique has better result.

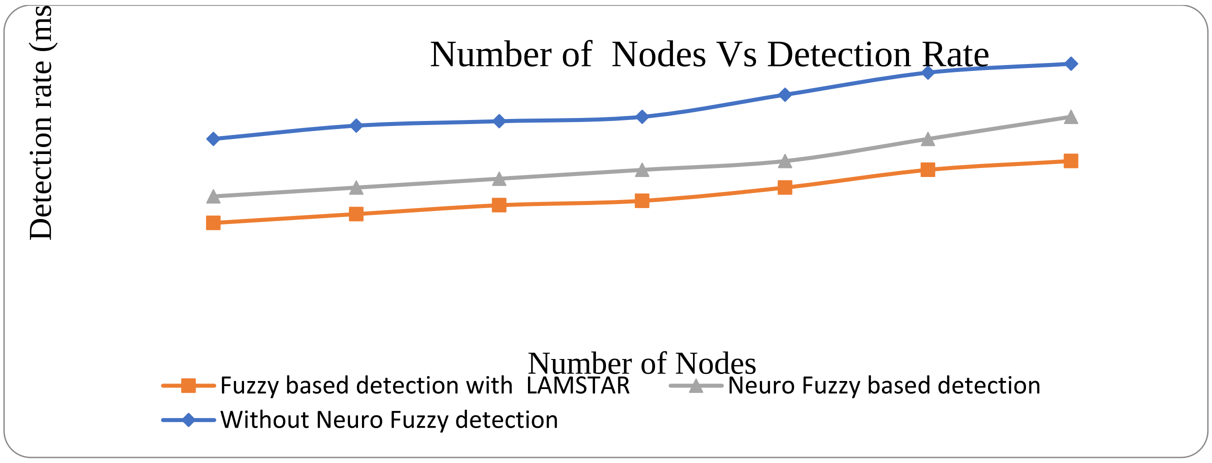


Fig4. Detection rate Vs No. Of Nodes

The performance graph as in Figure 4 shows that the neuro-fuzzy based LAMPSTAR technique detection system has achieved the higher response time for the rate of detection both in the wired and wireless networks. The proposed neuro-fuzzy based LAMPSTAR with clustering technique detection system is higher in performance than the traceback mechanism for DDoS attack detection method. The improvement of the response time is 35 to 45% of the detection in the network traffic data.

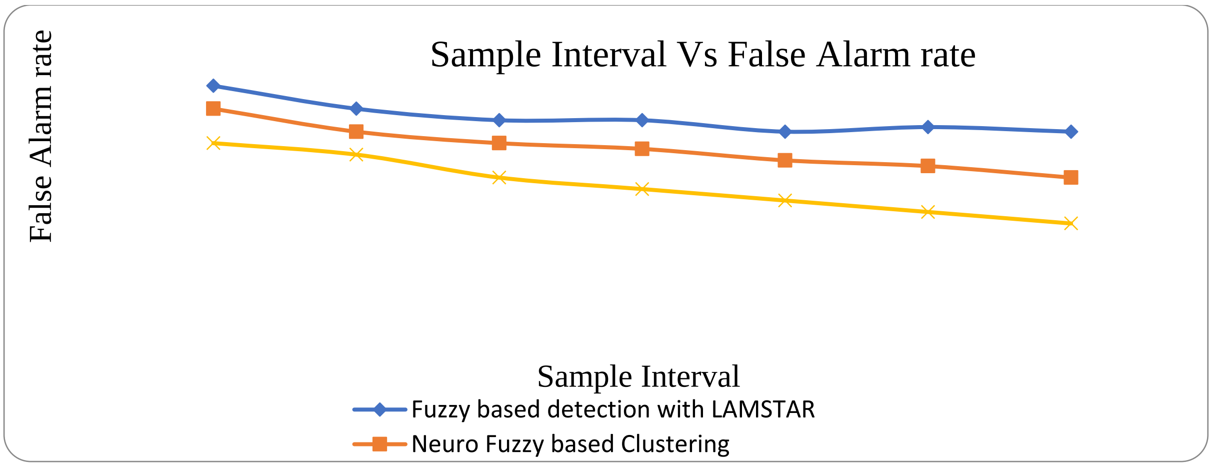


Fig5. Sample Interval Vs False Alarm rate

Figure 5 shows the graphical representation of the number of DDoS attacks that are detected for various experiments conducted for Neuro fuzzy based LAMPSTAR detection system. The experimental results indicate that the detection of the number of DDoS attacks is higher when compared to the other detection system. The turnaround point is fixed. The detection threshold and the value produce less result. False positive and too high values result in false negative. This value is influenced by the number of attributes selected for detection and it is validated for varying attack intensities using the response curve as shown in Figure 5.

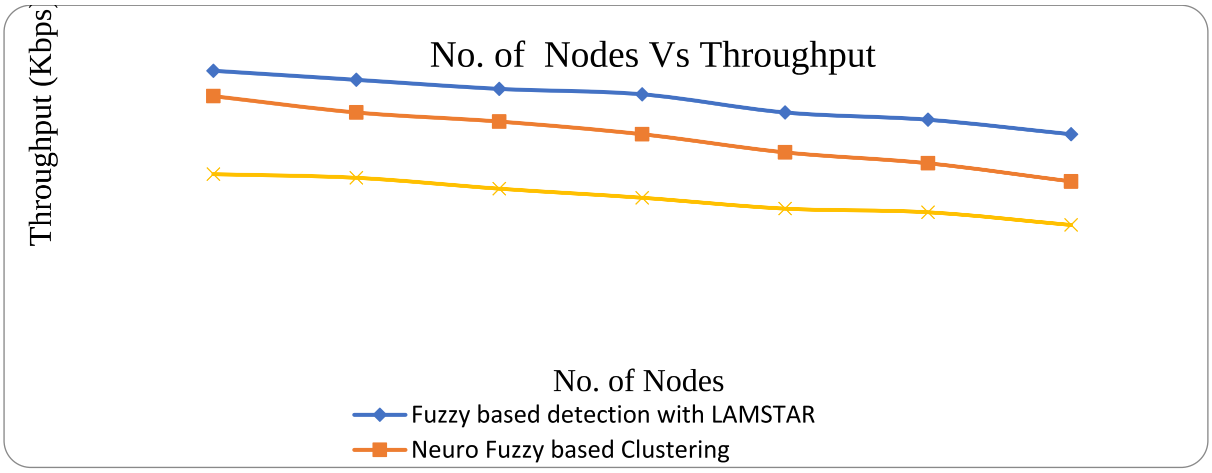


Fig6. No. of Nodes Vs Throughput

The Figure 6 depicts the output of the simulation by varying the number of nodes in the network traffic, which is based on the values shown. There is an appreciable change in the throughput of the data communication. When the number of nodes increases, throughput decreases. Neuro-Fuzzy based LAMPSTAR techniques detection outperforms.

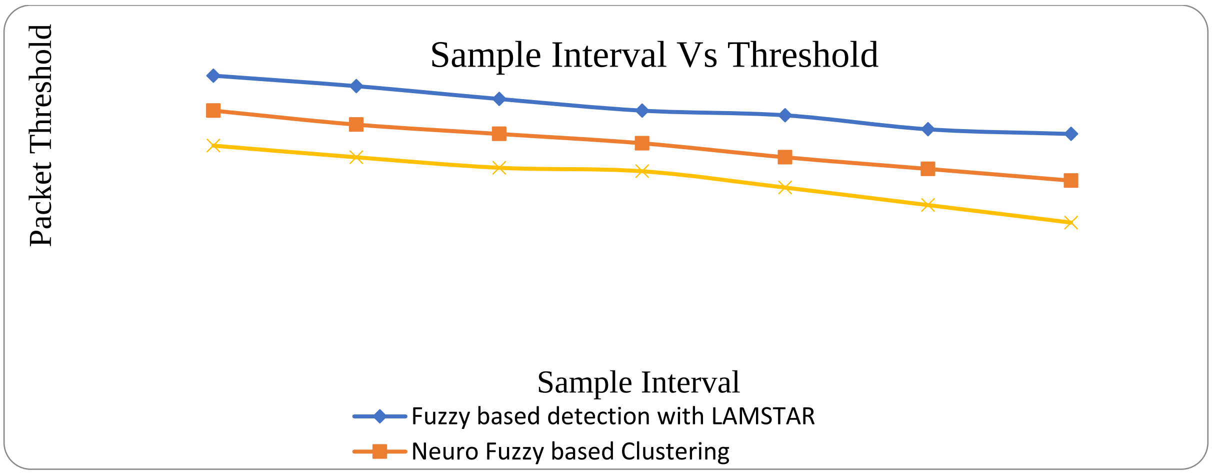


Fig7. Packet Threshold Vs Sample interval

The performance of the detection scheme is related to the value of threshold. Figure 7 shows the relationship between the sample interval and threshold. A false alarm will occur when Sk exceeds threshold without attack. When threshold grows, the false alarm rate decreases as expected. Due to the totally different nature of fuzzy based LAMPSTAR detection and other well known traceback schemes, involving packet marking or packet logging techniques, quantitative comparison of the various schemes is not possible.

CONCLUSION

An approach for detecting network DDoS attacks using LAMSTAR Neural Network is proposed in this paper. The performance of LAMSTAR DDOS evaluated using Attack data and compared with three other methods. Simulation results demonstrated that all the algorithms performed well for detecting attack. It gives a better performance than the other algorithms. The performance of LAMSTAR DDOS is obtained at the cost of high training and testing time due to computational complexity. The computational complexity can be reduced by, training and testing time, Principal Component Analysis was applied to Attack data. Experimental results with other features show significant reduction in training and testing time due to the reduction in computation while keeping the detection rate and false alarm rate almost the same.

REFERENCES

1. Ravinder Kumar (2017), A Review of Network Intrusion Detection System using Machine Learning Algorithms, International Journal of Computer Science and Engineering, 5(12), 94-100
2. Abirami Muralidharan and J Patrick Rousche (2005), Decoding of auditory cortex signals with a LAMSTAR neural network, Neurological Research, 27, 4-10
3. D Graupe and H Kordylewski (1998), A Large Memory Storage and Retrieval Neural Network for Adaptive Retrieval and Diagnosis, International Journal of Software Engineering and Knowledge Engineering, 8, 115-138
4. D Graupe (1997), Principles of Artificial Neural Networks, World Scientific Publishing Co. Pte. Ltd., Singapore, 191-222
5. H Kordylewski (1998), A Large Memory Storage and Retrieval Neural Network for Medical and Engineering Diagnosis/Fault Detection, Doctor of Philosophy’s Thesis, University of Illinois at Chicago, TK- 99999-K629

6. D Graupe and H Kordylewski (1997), A large scale memory (LAMSTAR) neural network for medical diagnosis, Proc. 19th Annual International Conference of the IEEE, 3(30), 1332 – 1335.
7. S K Chang, D Graupe, K Hasegawa and H Kordylewski (1998), An Active Multimedia Information System for Information Retrieval, Discovery and Fusion, International Journal of Software Engineering and Knowledge Engineering, 8, 139-160
8. http://kdd.ics.uci.edu//databases/Attack/Attack.html
9. Teuvo Kohonen (1990), The Self Organizing Map, Proc. IEEE, 78(9), 1464-1480
10. Srilatha Chebrolu, Ajith Abraham and Johnson P.Thomas (2005), Feature deduction and ensemble design of DDoS attack detection systems, Elsevier Journal of Computers & Security, 24(4),295-307
11. Itzhak Levin (2000), KDD-99 Classifier Learning Contest LLSoft’s Results Overview, SIGKDD Explorations, Copyright 2000 ACM SIGKDD, 1(2), 67 -75
12. www.ll.mit.edu/SST/lnknet/
13. www-ra.informatik.uni-tuebingen.de/ software/ JavaNNS/ welcome_e.html.
14. Dae-Ki Kang (2005), Learning Classifiers for Misuse and Anomaly Detection Using a Bag of System Calls Representation, Proc. 6th IEEE Workshop on Information Assurance and Security United States Military Academy, West Point, NY
15. D Nguyen, A Das, G Memik and A Choudhary (2006), Reconfigurable Architecture for Network DDoS attack Detection Using Principal Component Analysis, Proc. ACM/SIGDA 14th international symposium on Field programmable gate arrays, 235-235
16. M L Shyu, S C Chen, K Sarinnapakorn and L Chang (2003), A novel anomaly detection scheme based on principal component classifier, Third IEEE International Conference on Data Mining (ICDM’03), 172–179
17. I T Jolliffe (2002), Principal Component Analysis, Springer Verlag, New York, NY, third edition
18. Jing Gao, Haibin Cheng and Pang Ming Tan (2006), A Novel Framework for Incorporating Labeled Examples into Anomaly Detection . Proc. of the Siam Conference on Data Mining
19. Dima Novikov, Roman V. Yampolskiy and Leon Reznik (2005), Anomaly Detection Based DDoS attack Detection, Third IEEE International Conference on Information Technology: New Generations (ITNG'06,420-425

20. Richard Lippmann (2003), Passive Operating System Identification From TCP/IP Packet Headers, Proc. of the Workshop on Data Mining for Computer Security (DMSEC), Lincoln Laboratory, Massachusetts
21. Liberios Vokorokos, Anton Baley and Martin Chovenac (2006), DDoS attack detection system using self organizing map, Acta Electrotechnica et Informatica, 6(1), 1-6
22. Chaker Katar (2006), Combining Multiple Techniques for DDoS attack Detection, International Journal of Computer Science and Network Security, 6(2B)
23. M Thangavel and P Thangaraj (2011), Efficient Hybrid Network (Wired and Wireless) Intrusion Detection using Statistical Data Streams and Detection of Clustered Alerts, Journal of Computer Science, 7(9), 1318-1324

How to Cite:
Dr M Thangavel, A Rajesh & M Parameswari, "An Enhanced Method for Reducing the Computational Complexity of Distributed Denial of Service using LAMSTAR Principal Component Analysis", International Journal of Intelligent Computing and Technology (IJICT), Vol.2, Iss.2, pp.32-46, 2019