Detection of Application Layer DDoS Attack by Modeling User Behavior Using Logistic Regression

Satyajit Yadav and S. Selvakumar

Department of Computer Science and Engineering, National Institute of Technology,

Tiruchirappalli, Tamil Nadu, India

satyajityadav13@gmail.com

ssk@nitt.edu

***Abstract*— DDoS attack has been a threat to network security since a decade and it will continue to be so in the near future also. Now a days application layer DDoS attack poses a major challenge to webservers. The main objective of web server is to offer an uninterrupted application layer services to its benign users. But, the application layer ddos attack blocks the services of the web server to its legitimate clients which can cause immense financial losses. Moreover, it requires very less amount of resources to perform the application layer ddos attack. The solutions available to detect application layer ddos attack, detect only limited number of application layer ddos attacks. The solutions that detect all types of application layer ddos attacks have huge complexity. To find an effective solution for the detection of application layer ddos attack the normal user browsing behavior has to be modeled in such a way that normal user and attacker can be differentiated. In this paper, we propose a method using feature construction and logistic regression to model normal web user browsing behavior to detect application layer ddos attacks. The performance of the proposed method was evaluated in terms of the metrics such as total accuracy, false positive rate, and detection rate. Comparison of the proposed solution with the existing methods reveals that the proposed method performs better than the existing methods.**

***Keywords*—DDoS, Application Layer DDoS Attack, Feature Construction, Logistic Regression, User Behavior.**

I. INTRODUCTION

Distributed Denial of Service (DDoS) attack is one of the major security threats to the web security, where lots of zombie machines flood the web server with massive packets. It is reported in [1] that there has been a 51 percent increase of application layer attacks in a year from Q4, 2013 to Q4, 2014 and 16 percent increase in 3 months from Q3, 2014 over Q4, 2014. The seriousness of the application layer ddos attack has been brought up in one blog report of [2]. In this report, it is mentioned that one of their clients had suffered from the application layer ddos attack. The attack had been carried out in multi-fold, viz., first transport layer ddos attack had been carried out to get access on to the web server and then application layer ddos attack was launched. As lot of transport layer ddos attack detection solutions are available, transport layer ddos attack had been prevented within a few hours. But as only a few number of application layer ddos attack detection solutions are available it took a week to prevent it. Moreover, the administrator may not be able to detect such attack as it consumes very less amount of resources. DDoS attack can be carried out in each layer of the protocol stack, be it OSI or TCP/IP. For instance ARP flooding, ICMP flooding, TCP/UDP flooding, and HTTP flooding attacks are carried out in MAC, Network, Transport, and Application layer respectively. MAC, Network, and Transport layer attacks are lunched exploiting the vulnerabilities in the protocol stack or by IP spoofing technique. But to get access to application layer services and to perform an application layer ddos attack the user has to make a legal connection with the web server.

The Fig. 1, depicts the launching of the application layer ddos attack. An attacker can perform application layer ddos attack either by inserting some malicious code (Bot) on the normal user machine or by finding some back door to send HTTP traffic to the web server. To perform application layer ddos attack bot uses the client IP address to get access to the web server.

Following types of attacks are possible in the application layer ddos attack [3]:

1. Session flooding: it sends a session connection request at a huge rate than benign user.
2. Request flooding: in one session it makes a large number of requests than benign user.
3. Asymmetric attack: it makes requests with very high workload such as downloading of big files or response to some database intensive query.

In this paper, our focus is on three types of attacks. The rest of this paper is structured as follows:

In Section II, related work is described. In Section III, Motivation for this research work is presented. In Section IV, our proposed technique to detect the application layer ddos attack is explained. In Section V, experiments conducted and performance evaluation are presented. Section VI concludes our research work.

II. RELATED WORK

The application layer ddos attack detection method can be classified into the following two types: A. Sequential pattern recognition. B. Statistical pattern recognition.

A. Sequential Pattern Recognition

In [4, 5] Hidden Semi Markov Model (HsMM) for anomaly based application layer ddos attack detection is proposed. It uses HsMM to calculate similarity measures between normal traffic and attack traffic. HsMM is used to model normal web

978-1-4673-7231-2/15/$31.00 ©2015 IEEE

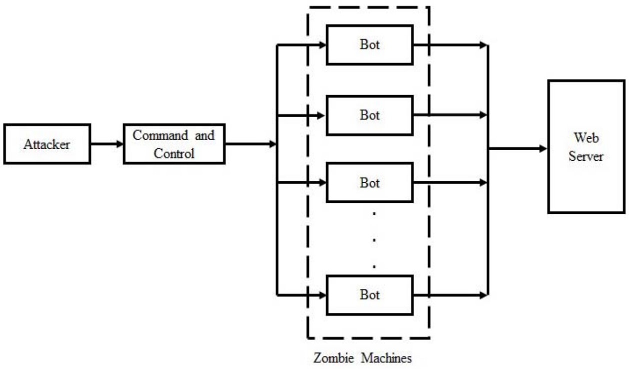

Fig. 1 Launching of application layer DDoS attack

user browsing behavior. It models the different sequences of request for normal client. It assumes that normal user accesses web page in some sequential order whereas attacker will not follow such sequential access. This assumption need not be true always and complexity of HsMM algorithm is very high. In [6] dynamic access matrix and HsMM are introduced to detect application layer ddos attacks based on anomaly detection. The web page popularity model is used to detect application layer ddos attack. Still the complexity of HsMM algorithm is high. In [7] an extended random walk graph model is used to describe user browsing behavior and to establish legitimate pattern of browsing sequence. Then it finds the similarity between predicted and observed page sequence by using Jacobi coefficient, based on which legality of users are decided. The drawback of application layer ddos attack detection using sequential pattern recognition may be summarized as follows:

1) One of the limitations in sequential pattern recognition is its inability to handle dynamic website, in which pages are added or removed frequently. Then the ddos attack detection would not work as it has to be trained afresh. For instance in NEWS website which is dynamic, it is hard to find the sequence of web pages request. Further it consumes a lot of time and resources.

2) The algorithm has to find all the possible sequences of the web page request that can be accessed by web users.

3) If number of clients are more (during finding the sequences at the time of training) then the number of sequences that has been generated by the client is huge.

4) As algorithm complexity is more, it could not be used on a web server which is under attack.

B. Statistical Pattern Recognition

1) *Puzzle Based*: CAPTCHA based authentication mechanism to prevent application layer ddos attack has been proposed in [8]. An authentication server maintained to differentiate normal user and attacker, sends CAPTCHA puzzles to the user and based on the response for puzzles, authentication is provided to the user. But the major problem is authentication server can become a point of ddos attack and solving CAPTCHA puzzles can cause huge overhead for benign user. The drawback of authentication server in CAPTCHA based authentication mechanism was overcome in [9] by introducing a hardware based mechanism, Sentinel, to prevent application layer ddos attack. Sentinel is a network processor which is used for mitigating bot based application layer request flooding attacks as per Botz4Sale. The Sentinel authenticates clients by sending CAPTCHA. If upcoming network traffic is above average load of network traffic then upcoming network traffic is redirected for authentication. Authenticated clients only get cookies and unauthenticated clients are blocked.

2) *History Based:* In [3] Trust management helmet, a signature based method to detect application layer ddos attack has been introduced. Trust is assigned to a client based on visiting history of client IP address. If the client is frequently accessing the website then the higher trust value is assigned to it. Trust value is divided into five stages of increasing order, viz., very low, low, medium, high, and very high. If the user performs some malicious activity its trust value gets reduced. As trust value is assigned based on visiting history of IP address attacker can get easy access to web server just by visiting before performing the attack. Connection score based method proposed in [10] assigns score based on the history and statistical analysis. Statistical analysis had been done during normal condition. The resources are realized for a connection based on the connection score, viz., higher the score better is the resources assigned. This method suffers from the drawback of attacker getting easy access to the web server by just visiting the web server before performing the attack.

3) *Clustering Based:* K-Means clustering has been introduced in [11] for the detection and offense mechanisms to protect legitimate sessions. But it consumes resources during the implementation. Hierarchical clustering used in [12] extracts the following four features, viz., average size of the object requested in a session, request rate, average popularity of object in session, and average transition probability. These extracted features are used to detect attack if it falls out of cluster. This method can cause a lot of false alarms.

In [13] DDoS Shield and resilient scheduler to counter application layer ddos attack has been proposed. DDoS shield was used to find suspicion assignment metrics for each session or every client. Suspicion assignment metrics get updated on upcoming requests. Suspicion metric for each session is found out using the attributes such as session inter arrival time, request inter arrival time, and session arrival time. Then the resilient scheduler schedules requests based on suspicion metrics with least suspicious first. The attributes used for suspicion assignment metrics are not sufficient to determine deviation between normal and attacker. In [14] a generalized entropy metrics and information distance metrics have been proposed to detect low rate application layer ddos attack. This is a router based solution and requires control of all routers in the network. In [15] a Group-Testing based approach is proposed. It is deployed on the back end of the server. It creates virtual servers. A set of group client requests are distributed to each virtual server. It identifies an attacker by testing the client in the group using a dynamic threshold.

Creating a virtual server can cause a large overhead on the server and it also increases response time. The drawbacks of application layer ddos attack detection using statistical pattern recognition may be summarized as follows:

1) Features available to determine application layer ddos attacks are very less.

2) The available features are not sufficient to determine all three types of application layer ddos attack, viz., Session Flooding, Request Flooding, and Asymmetric Attack.

III. Motivation

From the literature survey it is found that all existing solutions of application layer ddos attack detection may be broadly classified into sequential and statistical pattern recognition. But both of these suffer from drawbacks. In sequential pattern recognition, all URL request sequences made by normal user have to be determined prior to modeling the normal user browsing behavior. These are used to differentiate between normal user and attacker. The complexity of the algorithm is high if the number of normal users is very high. In statistical pattern recognition, the number of features is not sufficient enough to correctly classify all application layer ddos attacks. The existing solutions cannot detect all types of application layer ddos attacks as they use very less features to model user behavior. This motivated us to construct a new feature set from the original feature set, which can increase the accuracy of classification of all types of application layer ddos attack detection, and to use logistic regression. Logistic regression is used for modeling normal user behavior since it has the property of, its dependent variables being in dichotomous nature, viz., taking binary values either 0 or 1.

IV. Proposed Method

A. Introduction

In this paper, an anomaly based statistical pattern recognition method is proposed to detect application layer ddos attacks. Fig. 2, depicts the block schematic diagram of the proposed method.

From the web server log (Attack and Normal), after preprocessing, an application layer ddos attack dataset is constructed. From the constructed dataset features are extracted. From the extracted feature set some new characteristics are built. Then combining extracted and constructed features a new application layer ddos attack dataset is generated. The new application layer ddos attack dataset is split into two sub datasets, viz., training dataset and testing dataset. This has been done for cross validation to avoid over fitting of results. In this paper, 10 fold cross validation is used. In training phase, the algorithm is trained to detect application layer ddos attack and in testing phase, the algorithm is tested against upcoming traffic.

B. Dataset

There are too many standard datasets available for network/transport layer ddos attack detection. But there is no such standard dataset available for application layer ddos attack detection and those available, viz., ClarkNet, NASA, SDSC contain only normal traffic. These datasets are also very old. Moreover some of these datasets are not publicly available. Hence we have created application layer ddos attack dataset in our laboratory, the details of which are as follows:

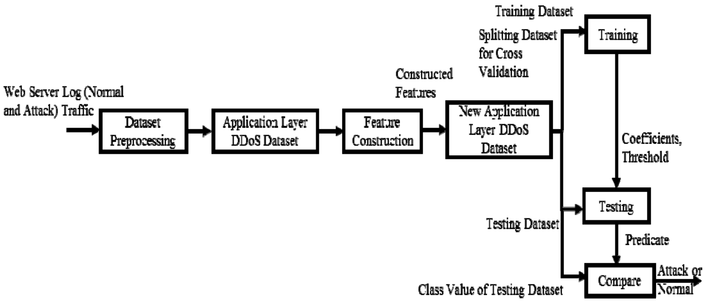
Web Server Log (Normal and Attack) Traffic
Dataset Preprocessing
Application Layer DDoS Dataset
Feature Construction
Constructed Features
New Application Layer DDoS Dataset
Training Dataset
Splitting Dataset for Cross Validation
Training
Coefficients, Threshold
Testing
Testing Dataset
Predicate
Compare
Class Value of Testing Dataset
Attack or Normal

Fig.2 Block schematic diagram of proposed method

1) *Normal Dataset:* In this paper, NIT Trichy website, nitt.edu web server log has been used as normal traffic. The web server logs from 1 Feb. 2015 to 8 Feb. 2015 contain all real world traffic coming from outside world to nitt.edu website.

2) *Attack Dataset:* Attack dataset has been created by performing an attack on the same website, nitt.edu, in a testbed environment as shown in Fig. 3. A web server has been created for nitt.edu website and used it for offline browsing. Then around 100 PCs are connected to web server through a switch. For performing an attack, two Bots, viz., a) Java LOIC (Low Orbit Ion Cannon) and b) Golden Eye Master have been used. All incoming traffic towards web server have been captured using traffic capture software.

a) Java LOIC: it is an open software based on windows platform. It provides GUI and gives freedom for the attacker to choose a specific URL to perform the application ddos attack. Traffic generated by Java LOIC does not contain 'refer' and 'user agent' fields.

b) Golden Eye Master: it is a python code which is run through the command prompt. It sends random URLs to the web server to perform application layer ddos attack. Traffic generated by this bot contains 'refer' and 'user agent' fields.

Also normal dataset in Testbed has been generated using the same experimental setup as shown in Fig. 3. Table 1 shows the datasets generated for use in this paper.

C. Dataset Preprocessing

1) *Feature Extraction:* A Two types of application layer ddos datasets (both normal and attack) have been created to validate the proposed application layer ddos attack detection method. One, which contains the real time NIT Trichy (website nitt.edu) web server log as normal dataset and attack dataset generated on the same website in offline mode, say D1 (as real time attack cannot be performed on an up and running website). Second, normal dataset generated in a Testbed environment and attack dataset in the same environment, say D2. Fig. 4, shows the block schematic diagram of dataset preprocessing for normal and for attack dataset.

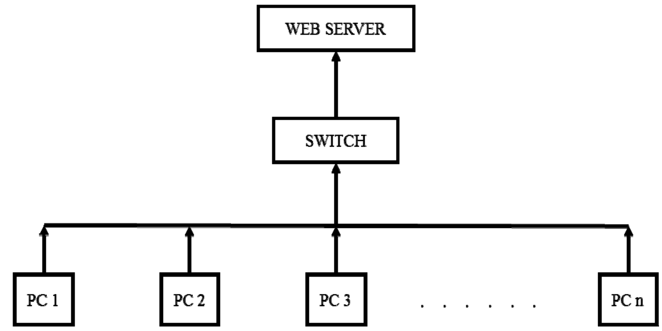

Fig. 3 Experimental setup for attack traffic generation

TABLE 1. Summary of the Datasets Used in the Paper

Type of Attack	Real Dataset D1 Normal(webserver log)+Attack(testbed)	Testbed Dataset D2 Normal+Attack(both testbed)
1) Request Flooding	D_{11}	D_{21}
2) Session Flooding	D_{12}	D_{22}
3)Asymmetric Attack	D_{13}	D_{23}

Both the datasets contain typically the following types of instances:

123.201.214.50 - - [01/Feb/2015:11:23:26 +0530] "GET /prm/ShowResult.htmHTTP/1.1" 200 345 http://www.nitt.edu/ Mozilla/5.0 (Windows NT 6.1) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/40.0.2214.93 Safari/537.36

From both the datasets the following features are extracted:

1) Src_IP_Addr: it (123.201.214.50) is the source IP address of the user.

2) Timestamp: it (01/Feb/2015:11:23:26) is the date and time at which request is made.

3) Timezone: it (+0530) is the time zone in which the web server is located.

4) URL: it (GET /prm/ShowResult.htm HTTP/1.1) represents the page that has been requested by the web user and the method "GET" or "POST".

5) Response_code: it is the response (200) generated by a web server to the request made by user.

6) No_of_Byte_Sent: it (345 bytes) is the number of data bytes sent from the web server to the user for a given request.

7) Refer: this (http://www.nitt.edu/) field indicates from which previous page the user has come to this page.

8) User_agent: this (Mozilla/5.0 (Windows NT 6.1) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/40.0.2214.93 Safari/537.36) field indicates the browsing information about the user. All these 8 features are extracted from both normal and attack datasets. The three features, viz., URL, Refer, and User_agent, extracted are in string format and these features are used to check only their presence or absence, as these features could cause a huge impact on detection of application layer ddos attacks.

2) *Feature Construction:* Features available through a feature extraction process are very less and some of the features are in string format. So, it is difficult to determine all three types of application layer ddos attacks using these features. Hence some more features are also constructed which are as follows:

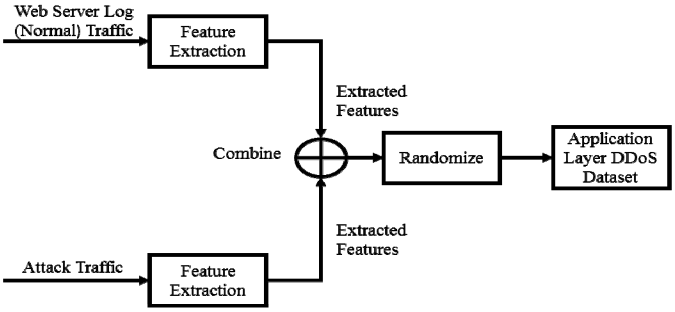

Fig. 4 Block schematic diagram of dataset preprocessing

1) Freq_of_HTTP_Req: it measures the number of HTTP requests made by a user in a particular timeslot. This parameter is used in HTTP request flooding attacks detection.

2) Freq_of _diff_HTTP_Req: it measures how many unique URL requests have been sent by users.

3) Freq_of_200_Resp: it measures how many requests by the user were successful.

4) Freq_of_302_Resp: it measures how many requests of users have been redirected.

5) Freq_of_404_Resp: it measures how many invalid requests have been made by users.

6) Sess_duration: it is the session duration between the first request made by an user to the last request in a particular timeslot.

7) Avg_no_of_req_Per_unit_time: it measures the average number of requests by an IP in a session, which is calculated as follows:

Avg_no_of_req_Per_unit_time = No. of requests / session duration

8) Perc_diff_Req: it measures the percentage of different requests made in a particular timeslot, which is calculated as follows:

Perc_diff_Req = (Freq_of_diff_HTTP_Req / Freq_of_HTTP_ Req) *100

9) Perc_Succ_Resp: it measures the percentage of successful responses received by an user, which is calculated as follows:

Perc_Succ_Resp = (Freq_of_200_Resp / Freq_of_HTTP_Req) *100

By combining all these 9 constructed features with the 8 extracted features, the application layer ddos attack dataset with a total of 17 features is created.

D. Methodology

In this paper, Principle Component Analysis has been used to find effective parameters (feature) of these 17 features. The output of the PCA is input to Logistic Regression which is used to compare the upcoming traffic for classifying into normal and attack traffic.

1) *Principle Component Analysis (PCA):* It is the simplest statistical procedure which uses orthogonal transformation to convert a set of correlated variables (features) into a set of values that are linearly uncorrelated called principle

component. The following steps are used to reduce data from n dimensions to k dimensions:

1) Compute the covariance matrix:

$$\Sigma = \frac{1}{m}\sum_{i=1}^{n}(x^i)(x^i)^T$$

Where (x^i) is feature set and $(x^i)^T$ is the transpose of feature set.

2) Compute eigenvector of covariance matrix Σ:

$$[U, S, V] = SVD(\Sigma)$$

Where SVD is Singular Value Decomposition, and U is [n x n] matrix.

3) After reducing U to k, the k dimensions (feature) are used in Step 6 of Algorithm-1.

2) *Logistic Regression:* Logistic regression is one of the most popular classification methods. It has been used for modeling user behavior. The application layer ddos attack can be determined by using the effective features from these 17 features. In this paper, logistic regression is used as it is suitable for modeling normal user web browsing behavior.

The logistic regression can be modeled as follows: If there are n independent variables/features $x_1, x_2, \dots x_n$ then the joint action of these variables, viz., the conditional probability of attack detection is given by

$$P = P(y{=}1|\ x_1, x_2, \dots x_n) \text{ and}$$

Logistic regression as $p = \frac{e^y}{1+e^y}$

Where, $y = \gamma_0+\gamma_1x_1+\gamma_2x_2+ \dots +\gamma_nx_n$, γ_i is the coefficient, and x_1, $x_2, \dots x_n$ are features, these are used in Step 7 of Algorithm-1.

E. Algorithm

Two algorithms, one for creating effective feature set, computing coefficients, computing threshold, and the other for predicting class value have been proposed in this paper.

1) *Training Dataset: Algorithm 1:*

```
Input: Normal dataset, Attack dataset.
Output: effective feature set, $\gamma^T$, threshold $\tau$.
// Initialize parameter for training
Step 1. Assign class values for normal as well as attack
        dataset.
Step 2. Combine normal and attack dataset and generate
        application layer ddos attack dataset.
Step 3. Randomize the whole application layer ddos traffic.
Step 4. Extract the features from the dataset.
For each instance
Step 5. Construct 9 features from the extracted feature set.
Step 6. Select effective parameter using PCA.
Step 7. Compute $\gamma^T$, threshold $\tau$, and output.
End
```

2) *Testing Dataset: Algorithm 2:*

```
Input: $\gamma^T$, threshold $\tau$, effective features form training
        dataset, upcoming traffic.
Output: predicted class value.
// Initialize parameter
Step 1. Extract the features from the dataset.
For each instance
Step 2. Construct effective feature from the extracted
        feature set.
Step 3. Find the probability (p) for instance using $\frac{e^y}{1+e^y}$
        Where, $y = \gamma_0+\gamma_1x_1+\gamma_2x_2+ \dots +\gamma_nx_n$
Step 4. Compare it with threshold $\tau$
        if $p > \tau$
        Predicted value = attack
        else
        Predicted value = normal
End
```

V. Experimental Result

A. Introduction

This section discusses the different experiments carried out on the application layer ddos attack detection and the analysis of the result. All of these experiments were carried out in Java and in Machine Learning tool Weka (Waikato environment for knowledge analysis) over Intel Core 2 Quad CPU running at 3.0 GHz and 4GB RAM. The metrics, Total Accuracy (TA), Detection Rate (DR), and False Positive Rate (FPR) were used as performance metrics.

B. Experiment-1: Detection of request flooding attack

The Real dataset D_{11} and Testbed dataset D_{21} were input to Algorithm-1 and Algorithm-2. The three metrics computed are listed in Table 2. As these two datasets contain only request flooding attacks, the three metrics were the indication of detection of request flooding attack.

C. Experiment-2: Detection of session flooding attack

The Real dataset D_{12} and Testbed dataset D_{22} were input to Algorithm-1 and Algorithm-2. The three metrics computed are listed in Table 3. As these two datasets contain only session flooding attacks, the three metrics were the indication of detection of session flooding attack.

D. Experiment-3: Detection of asymmetric attack

The Real dataset D_{13} and Testbed dataset D_{23} were input to Algorithm-1 and Algorithm-2. The three metrics computed are listed in Table 4. As these two datasets contain only asymmetric attacks, the three metrics were the indication of detection of asymmetric attack.

E. Discussion on the result

From Tables 2, 3, and 4 it is seen that the proposed method detects the three types of application layer ddos attacks, such as Request Flooding, Session Flooding, and Asymmetric Attack present in the Dataset $\{D_{11}, D_{21}\}$, $\{D_{12}, D_{22}\}$, and $\{D_{13}, D_{23}\}$ respectively. From this six values average was calculated and it worked out to be 1.41%, 98.47%, and 98.64% for False Positive Rate, Total Accuracy, and Detection Rate respectively. These average values are used to compare the performance of proposed method with the existing methods such as HsMM, Random Walk Graph, and Hierarchical

TABLE 2. Detection of Request Flooding Attack on D_{11} and D_{21} Datasets

Dataset	FPR (%)	TA (%)	DR (%)
D_{11}	1.7	98.05	98.79
D_{21}	1.66	98.55	98.79

TABLE 3. Detection of Session Flooding Attack on D_{12} and D_{22} Datasets

Dataset	FPR (%)	TA (%)	DR (%)
D_{12}	0.9	98.64	98.79
D_{22}	1.2	98.61	98.51

TABLE 4. Detection of Asymmetric Attack on D_{13} and D_{23} Datasets

Dataset	FPR (%)	TA (%)	DR (%)
D_{13}	1.5	98.40	98.15
D_{23}	1.5	98.58	98.81

TABLE 5. Comparison of Proposed Method with Existing Solutions

Methods	FPR (%)	DR (%)
HsMM	4.50	86.70
Random Walk Graph	2.10	96.00
Hierarchical Clustering	3.87	91.53
Proposed Method	1.41	98.64

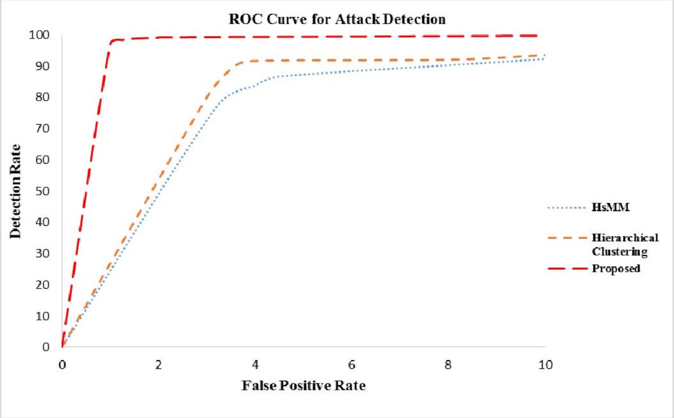

Fig. 5 Comparison ROC curve for Detection Rate vs False Positive Rate

Clustering as shown in Table 5. From the Table 5, it is evident that the performance of the proposed method is better than the existing methods. Further the Receiver Operating Characteristic (ROC) curve for the proposed method and the existing solutions for Detection Rate vs False Positive Rate are shown in Fig. 5. From the Fig. 5 it can be seen that the detection rate of the proposed method is really high as compared to existing solutions of application layer ddos attack detection.

VI. Conclusions

In this paper, a model was proposed using Logistic Regression for modelling the normal user browsing behaviour for detecting the application layer ddos attack traffic, if any, from the upcoming traffic. To model user behaviour, different features were constructed to differentiate between attacker and normal user. The optimal or near optimal features were selected from newly constructed feature set and used along with existing feature set. A normal traffic collected from web server log and attack traffic generated in Testbed environment were used to validate our approach. The proposed method was tested for different real world and generated datasets. From the experimental results it is evident that the proposed method effectively classifies the attack traffic from the normal traffic with an average Detection Rate of 98.64% and an average False Positive Rate of 1.41%.

References

[1] [Online]. Available: https://blogs.akamai.com/2015/01/q4-2014-state-of-the-internet---security-report-some-numbers.html [Accessed on 12 May 2015].

[2] [Online]. Available https://www.incapsula.com/blog/funded-persistent-multi-vector-ddos-attack.html [Accessed on 12 May 2015].

[3] J. Yu, C. Fang, L. Lu, and Z. Li, "Mitigating Application Layer Distributed Denial of Service Attacks via Effective Trust Management," *IET Comm.,* vol. 4, no. 16, pp. 1952-1962 Nov. 2010.

[4] Yi Xie and Shun-Zheng Yu, "A Novel Model for Detecting Application Layer DDoS Attacks," *Proceedings of the First International Multi-Symposiums on Computer and Computational Sciences (IMSCCS'06)* pp. 56 - 63 20-24 June 2006 Hanzhou, Zhejiang.

[5] Yi Xie and Shun-Zheng Yu, "A Large-Scale Hidden Semi-Markov Model for Anomaly Detection on User Browsing Behaviors," *IEEE/ACM Transactions on Networking,* Vol. 17, No. 1, pp.54 - 65 February 2009.

[6] Yi Xie and Shun-Zheng Yu, "Monitoring the Application-Layer DDoS Attacks for Popular Websites," *IEEE/ACM Transactions on Networking,* Vol. 17, No. 1, pp. 15 – 25 February 2009.

[7] Chuan Xu, Guofeng Zhao, Gaogang Xie, Shui Yu, "Detection on Application Layer DDoS using Random Walk Model," *IEEE ICC 2014 - Communication and Information Systems Security Symposium* pp. 707 – 712 10-14 June 2014 Sydney, NSW.

[8] Srikanth Kandula, Dina Katabi, Matthias Jacob, Arthur Berger, "Botz4Sale: Surviving Organized DDoS Attacks That Mimic Flash Crowds," *Proceeding NSDI'05 Proceedings of the 2nd conference on Symposium on Networked Systems Design & Implementation* - Volume 2 pp. 287-300 2005-05-02 CA USA.

[9] Peter Djalaliev, Muhammad Jamshed, Nicholas Farnan and Jos´e Brustoloni, "Sentinel: Hardware-Accelerated Mitigation of Bot-Based DDoS Attacks," in *Proceedings of 17th International Conference on Computer Communications and Networks, ICCCN'08,* pp. 1 - 8 St. Thomas, US Virgin Islands.

[10] Hakem Beitollahi, Geert Deconinck, "ConnectionScore: a statistical technique to resist application-layer DDoS attacks," *Journal of Ambient Intelligence and Humanized Computing*, Vol 5 pp. 425 – 442 June 2014 Springer-Verlag Berlin Heidelberg.

[11] Jie Yu, Zhoujun Li, Huowang Chen, Xiaoming Chen, "A Detection and Offense Mechanism to Defend Against Application Layer DDoS Attacks," *Third International Conference on Networking and Services (ICNS'07)* pp. 54 - 59 19-25 June 2007 Athens.

[12] Chengxu Ye, Kesong Zheng, "Detection of Application Layer Distributed Denial of Service," *2011 International Conference on Computer Science and Network Technology* pp. 310 - 314 24-26 Dec. 2011 Harbin.

[13] Supranamaya Ranjan, Ram Swaminathan, Mustafa Uysal, Antonio Nucci, and Edward Knightly, "DDoS-Shield: DDoS-Resilient Scheduling to Counter Application Layer Attacks," *IEEE/ACM Transactions On Networking,* Vol. 17, No. 1, pp. 26 - 39 February 2009.

[14] Yang Xiang, Ke Li, and Wanlei Zhou, "Low-Rate DDoS Attacks Detection and Traceback by Using New Information Metrics," *IEEE Transactions on Information Forensics and Security,* Vol. 6, No. 2, pp. 426 - 437 June 2011.

[15] Ying Xuan, Incheol Shin, My T. Thai, Taieb Znati, "Detecting Application Denial-of-Service Attacks: A Group-Testing-Based Approach," *IEEE Transactions on Parallel And Distributed Systems,* Vol. 21, No. 8, pp. 1203 – 1216 August 2010.