# Detection and Analysis of DDOS Attack at Application Layer Using Naïve Bayes Classifier

**Article** · October 2019

CITATIONS
9

READS
393

**3 authors**, including:

Maha Dev
Chandigarh University
**1** PUBLICATION **9** CITATIONS
SEE PROFILE

Vinod Kumar
Gurukula Kangri Vishwavidyalaya
**17** PUBLICATIONS **115** CITATIONS
SEE PROFILE

**International Journal of Computer Engineering & Technology (IJCET)**
Volume 9, Issue 3, May-June 2018, pp. 208–217, Article IJCET_09_03_025
Available online at http://www.iaeme.com/ijcet/issues.asp?JType=IJCET&VType=9&IType=3
Journal Impact Factor (2016): 9.3590(Calculated by GISI) www.jifactor.com
ISSN Print: 0976-6367 and ISSN Online: 0976–6375


# DETECTION AND ANALYSIS OF DDOS ATTACK AT APPLICATION LAYER USING NAÏVE BAYES CLASSIFIER

**Mahadev**
Research Scholar, Dept. of Computer Science,
Gurukula Kangri Viswavidyalaya, Haridwar, Uttrakhand, India.

**Vinod Kumar**
Professor, Dept. of Computer Science,
Gurukula Kangri Viswavidyalaya, Haridwar, Uttrakhand, India

**Himani Sharma**
Research Scholar, Dept. of Computer Science,
Gurukula Kangri Viswavidyalaya, Haridwar, Uttrakhand, India

**ABSTRACT**

*Most of the services provided through internet becomes vulnerable because of DDoS attacks. It is very difficult to detect these attacks at application layer because time to time these attacks change its characteristics to avoid detection using present DDoS attack detection techniques. So, it is compulsory to understand the characteristics of these attacks before mitigation. A NSDA (Network Security against DDoS Attack) model is proposed here which generates new features i.e. difference of two consecutive times of requests per IP address and Bpt denoting similarity and dissimilarity in byte size (BS) from the log file to efficiently detect these attacks at the application layer. In this model, preprocessing is performed using java programming and Weka 3.8 machine learning tool. Re-sampling method SMOTE (Synthetic Minority Oversampling Technique), RANDOMIZE, RESAMPLE of Weka is used to convert the main dataset into a training set, cross-validation set, and test set. A naive bayes classification in Weka 3.8 is for analyzing and detection of a DDoS attack. A website www.wielson.com is designed to collect pure data set of DDoS attacks so that good quality of analysis can be achieved. DDoS attack testing tool Zombies is used to perform HTTP attack.*

**Key words**: NSDA (Network Security against DDoS Attack) Model, DDoS, HTTP, Weka, Application Layer, log file.

**Cite this Article:** Mahadev, Vinod Kumar and Himani Sharma, Detection and Analysis of DDOS Attack at Application Layer Using Naïve Bayes Classifier. *International Journal of Computer Engineering & Technology*, 9(3), 2018, pp. 208–217.
http://www.iaeme.com/ijcet/issues.asp?JType=IJCET&VType=9&IType=3

# 1. INTRODUCTION

This is an era where information is everywhere. Consumption and production of information are increasing very fast. The Internet is playing a major role for uploading and downloading this information by most of the people in the world. The Internet was basically invented for easy circulation of information but security issues were not in the mind of the designers at the time of designing. However, important services are also provided through the internet. More than one billion people are taking services of internet. Businessmen use internet to sale their products and services online and political person persuades people through social media. Disruptions of online services using vulnerabilities of the internet through malpractice create trouble to the life of those people who are dependent on these services for basic needs. The DDoS attack is one such malpractice that disrupts the internet services [1]. In the World Wide Web, websites can be accessed through hyperlinks but these hyperlinks can create a problem when DDoS attacks are performed using these fake links which may not exist. DDoS attack tools with several advanced techniques are easily available on the internet. These tools cause potential damage to resources, finance and brand image of the victims of the DDoS attack that is performed using vulnerabilities of protocols on the network layer, transport layer, and application layer. App-DDoS attacks are very complex to detect because attacker creates legitimate like environment for communicating with victims. To stop this attack, understanding of characteristics of DDoS attack as well as the structure of websites is necessary because mitigation and detection techniques need to know the ranks of the web pages of particular websites on the basis of popularity. This rank can change page sequences when any modification occurs in the structure and contents of this website.

Analysis and detection of modern DDoS attacks are attempted in this paper using Bayesian classification with Weka tools. Log files created by Apache 2 server are used for this purpose. There are several DDoS tools available on internet that generate attack with uniqueness after installation which is difficult to detect so it becomes necessary to understand the behavior of this kind of advanced attack [2]. Here we facilitate the understanding of characteristics of this attack using the log files.

Some types of DDoS attacks target application layer. A few classes of these attacks are mentioned below.

- HTTP Flood Attack
- DNS flood
- DNS Amplification attack
- SNMP Attack

## 1.1. HTTP Flood Attack

Assaulter misuses the real HTTP GET or POST requests to assault a web application or server. These are volumetric assaults, often utilizing a botnet "zombie armed force" a gathering of Internet-connected systems, every one of which has been malevolently under the control of unauthorized access, generally with the help of malware like Trojan Horses. HTTP floods don't utilize distorted packets, spoofing or reflection methods, and utilize less bandwidth than different assaults to cut down the targeted site or server. In this attack, the structure of the targeted website is studied thoroughly and each assault is designed to be successful. HTTP attack becomes harder to recognize in this way. The assault is best when it forces the server or application to distribute most of the possible resources because of every single request. In this way, the attacker will overwhelm the server with several undetected malicious requests demanding high processing power to overrun the server.

For this reason, HTTP flood assaults utilizing POST requests have a tendency to be the most resource oriented from the aggressor's point of view; as POST requests may incorporate parameters that trigger high processing of server side. HTTP GET-based assaults are easier to make and would be more destructive if the botnet is used.

### 1.2. DNS flood

DNS flood is a sort of Distributed Denial of Service (DDoS) assault in which the aggressor targets one or more Domain Name System (DNS) servers related to a given zone, trying to prevent DNS to resolve the problem of updating of resource records of that zone and its sub-zones [3]. DNS floods are symmetrical in nature. These assaults make an effort to deplete server-side resources like CPU and memory etc with a UDP flood created by scripts running on multiple botnet machines. A DNS flood attack utilizes a method of UDP flood because UDP protocol is considered to perform the issue name resolution task on DNS server.

### 1.3. DNS Amplification Attack

DNS amplification makes the DNS server its own weapon to solve the purpose of DDoS attack with requests of small size primarily and then converting them into heavy load towards the targeted server using the unprotected mechanism of DNS server. DNS amplification exploits approachable DNS and makes a flood of a large number of packets in a reflective manner. In this attack, several amplification techniques are used to increase the size of UDP packets in large quantity. This attack is so powerful to make any highly protected internet services fail to achieve its objectives.

### 1.4. SNMP amplification attack

This attack is similar to the earlier version of DNS Amplification attack that uses SNMP (Simple Network Management protocol) which is used for collection of data related to hubs, server, routers and switches etc.

## 2. RELATED WORKS

After searching available literature, very limited works have been noticed to be done on the DDoS attack using the log files. Naïve Bayes classifier [4] is deployed to classify packets into normal and attacked traffic. Several features and characteristics of packets in CAIDA dataset are used for analysis. CAIDA dataset is very old to tackle current version of App-DDoS attack. Payload feature in the dataset is removed so it becomes very complex to detect today's version of DDoS attack.

Sparse vector decomposition and rhythm matching (SVD-RM) [5] method used a log file of Clarknet dataset through Weka and Matlab and claims to clearly find a discrepancy between attackers and normal user in the dataset. SVD-RM doesn't mention the separation of log records into parameters like IP address, time, referer, byte size etc. The Clarknet dataset is not updated with the recent trends of DDoS attack.

RPV (Real-time Frequency Vector) [6] model with the real-time characterization of traffic identifies App-DDoS attack including flash crowd by analyzing the entropy of this attack. A defense infrastructure is also designed consisting of a detection system, traffic filter and head-end sensor. It was claimed that this filter has the ability to mitigate the App-DDoS attack and allow a normal user to access the server. Structure of web site and size of packets are not considered in the parameters. So, it is suspicious that this model will correctly differentiate between App-DDoS attack and legitimate users. Most of the recent researches work on old dataset [7] which may not be of much use in the present scenario of DDoS attack with the sophisticated mechanism.

A model based on information gain and machine learning technique [8] named as random forest classifier, achieves the objective of detection of Denial of service attack. Optimal features are selected from available 41 features from the NSL-KDD dataset using information gain. Weka tool is deployed to implement this model and to get desired results. But NSl-KDD dataset doesn't provide updated data as compared to today's scenario and this model doesn't provide information about selected features which are used for detecting this attack.

A statistical one dimensional access matrix (ODAM) [9] model is proposed to check behavior of DDoS attack and which proposes an algorithm to detect this attack at application layer. This method differentiates DDoS attack and flash crowd in busy time and decreases false positive rate at significant level.

## 3. THE PROPOSED METHOD

Here a new model named as Network Security against DDoS Attack (NSDA) is developed for detection of DDoS attack and identification of the normal user. The following Data flow diagram represents operational and procedural sequence of this model.

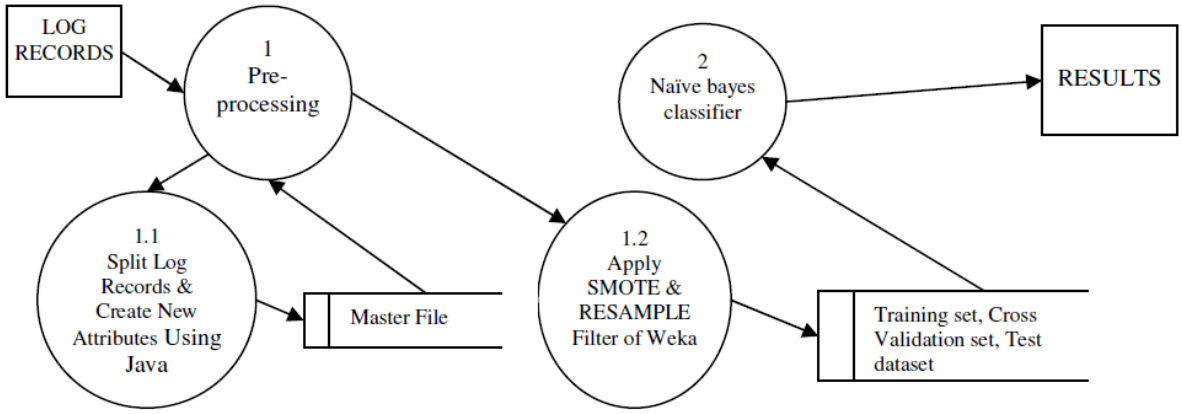


**Figure 1** Network Security against DDoS Attack (NSDA)

### 3.1. Strategy for Collection of Data

Log records of normal data as well as attacked data are collected on different dates which are on 26/3/2018, 28/3/2018 and 29/3/2018. A web site is created for collection of attacked data and normal data. For classification, web site is attacked with DDoS attacking tool zambie and is also made open to group of DDoS attackers on facebook. For collection of normal data, this site is shared with social networking sites, mobile applications through students of Gurukula Kangri Vishwavidyalaya and Invertis University. Performance checking website check-host.net [9] shown that during DDoS attack website became unavailable to all over the world.

### 3.2. Preprocessing

Apply regular expression in Java to split each field of log record into columns.

- Erase all the records of status of 404 (error code of server in response when webpage is not found) and Create two new features DT (Difference of two consecutive times of request per IP address) and Bpt (denote similarity and dissimilarity in byte size (BS)).

$$\mathrm{DT}_{i+1} = \begin{cases} d_i - d_{i+1} & if \;\; IP_i = IP_{i+1} \\ 0 & if \;\; IP_i \neq IP_{i+1} \end{cases} \quad (1)$$

$$\mathrm{Bpt}_{i+1} = \begin{cases} 0 & \text{if } \mathrm{BS_i} \neq \mathrm{BS_{i+1}} \text{ or } \mathrm{DT}_{i+1} \geq 1800 \\ \mathrm{Bpt}_i + 10 & \text{if } \mathrm{BS_i} = \mathrm{BS_{i+1}} \text{ and } \mathrm{DT}_{i+1} \leq 1800 \end{cases} \quad (2)$$

Where condition $DT_{i+1} \geq 1800$ suggests if difference of time of two consecutive requests of the same IP is greater than 30 min (end of session) or webpage of two consecutive requests of same IP is differ then set $Bpt_{I+1} = 0$. If byte size of two consecutive requests is equal and difference of time of two consecutive requests of the same IP is less than 30 min than increase $Bpt_{I+1}$ by ten to show the similarity for the request of same webpage.

- Store all features (IP address, time, method, requested page, protocol, bytes send, referer, user agent) of access log from apache server in Mysql database and then converted into Weka compatible csv file.
- Attribute Byte size (size of the webpage in bytes to be sent to a user in response) from list of attributes of log records and created attributes DT and Bpt are selected for data mining through Weka.
- Process 1.2 resamples a dataset by applying the SMOTE (Synthetic Minority Oversampling Technique), if dataset of one class is less than dataset of other class. This technique increases the numbers of minority instances in a dataset in some percentage and make it balanced to avoid the experience of underperformance and overfitting. This supervised filter of Weka data mining tool for preprocessing dataset give better result than without resampling of dataset. To avoid the condition of overfitting during cross-validation of data after balancing the data set using SMOTE technique, there is a need to randomize dataset with RANDOMIZE function which is an unsupervised filter of Weka.

The dataset should be separated into three groups that is training set, cross-validation set and testing set without any duplicate instances in any dataset to make a good model. Preprocessing technique of Weka (RESAMPLE) produces three random samples from a population as mentioned in Table 1. Population or dataset split up into three sets with a random selection of instances that is first 70% for the training set, second 15% for the cross-validation set, and third 15% for the testing set as per the rule of research methodology as shown in Table 1. Where alphabets in example is showing tuples of dataset

**Table 1** Breaking up of dataset into three

| Dataset | Original | Training | Cross | Test |
|---|---|---|---|---|
| Split % | 100% | 70% | 15% | 15% |
| Ex. Split of data | {a,b,c,d,e,f,g,h,i,j,k,l} | {a,d,e,g,h,j,l} | {c,k,l} | {b,d,f} |

## 3.3. Classification Using Model

In Process 2 of NSDA, Naïve Bayes classifier generates one sub model for every sample (training, cross-validation, and testing) using naïve Bayes classification and produce results as given in different tables in coming sections.

## 3.4. Formula for Naïve Bayes Classification

Bayes rule is used to calculate the probability of each class given the observed value of the predictive parameter. X is a vector set of parameter values.

$$p(G|X) = \frac{p(G)p(X|G)}{p(X)} \quad (3)$$

$$p(X|G) = \prod_{i} p(X_i|G) \quad (4)$$

$$p(X|G) = p(X_1|G) \times p(X_2|G) \times p(X_3|G) \times \ldots\ldots\ldots p(X_n|G) \quad (5)$$

Where $X = (X_1, X_2, X_3 \ldots\ldots X_n)$

G denotes class (n: normal user, d: DDoS attack) of each record of the log.

n = total number of selected parameters

Using Gaussian probability density function, Conditional probability of a record of a given class (n, d) will be computed as

$$p(X|G) = \frac{1}{\sqrt{2\pi\sigma_G}} e^{-\frac{(X-\mu_G)^2}{2\sigma_G{}^2}} \qquad (6)$$

Mean and standard deviation of the dataset will be computed prior to being used in conditional probability as

$$\text{mean } (\mu) = \frac{\sum_{i=1}^{n} X_i}{n}$$

and

$$\text{standard deviation } (\sigma) = \sqrt{\frac{\sum_{i=1}^{n}(X_i - \mu)}{n}}$$

Confusion matrix is shown in table 2 which is the basis for checking accuracy and credibility of the proposed model.

**Table 2** Confusion Matrix

| Dataset | d | n | Outcome |
|---|---|---|---|
| Training | 3944 | 22 | d |
| | 38 | 1784 | n |
| Cross validation | 861 | 11 | d |
| | 3 | 366 | n |
| Testing | 835 | 11 | d |
| | 2 | 391 | n |

Table 3 represents results of naïve classifier model for training, cross-validation and testing dataset in terms of class, mean, standard deviation, weighted sum, and precision for all selected attributes.

**Table 3** Results produced by naïve bayes classifier

| Data set | Attribute | Class | mean | std. dev. | W. sum | Precision |
|---|---|---|---|---|---|---|
| Training | BS | d (0.69) | 1535.28 | 257.70 | 3966 | 1546.20 |
| | | n(0.31) | 6286.62 | 12750.88 | 1822 | 1546.20 |
| | DT | d (0.69) | 0.12 | 26.32 | 3966 | 157.93 |
| | | n(0.31) | 8.75 | 753.77 | 1822 | 157.93 |
| | Bpt | d (0.69) | 3912.64 | 4599.17 | 3966 | 11.79 |
| | | n(0.31) | 0.45 | 3.18 | 1822 | 11.79 |
| Cross validation | BS | d (0.7) | 0.00 | 1083.89 | 872 | 6503.31 |
| | | n (0.3) | 8177.61 | 46818.44 | 369 | 6503.31 |
| | DT | d (0.7) | 0.47 | 10.94 | 872 | 45.30 |
| | | n (0.3) | 3.81 | 135.30 | 369 | 45.30 |
| | Bpt | d (0.7) | 3941.66 | 4649.61 | 872 | 31.88 |
| | | n (0.3) | 0.17 | 5.31 | 369 | 31.88 |
| Testing | BS | d (0.68) | 1674.36 | 140.03 | 846 | 840.16 |
| | | n (0.32) | 5701.53 | 7350.77 | 393 | 840.16 |
| | DT | d (0.68) | 0.00 | 62.49 | 846 | 374.91 |
| | | n (0.32) | 97.31 | 1345.47 | 393 | 374.91 |
| | Bpt | d (0.68) | 3920.81 | 4674.89 | 846 | 32.56 |
| | | n (0.32) | 0.17 | 5.43 | 393 | 32.56 |

Naïve Bayes classifier takes 0.09 sec to build a basic model and 0.19 sec. to build test model on the training set. Further it takes 0.01 sec. to build basic model and .02 sec to build test model on the supplied test set.

Table 4 indicates that naïve Bayes classifier gives efficient and reliable results with selected attributes for the training set, cross-validation set, and testing set. The low Error rate of mean absolute error (average difference between predicted values and actual values), root mean square error (difference between sample values (sample and population)) are good measures of accuracy to make a comparison between different models based on errors, relative absolute error (RAE) as equation (7) and root relative square error (RRSE) as equation (8) show the accuracy of the model based on classification.

$$\text{RAE} = \frac{|p_{r1} - a_1| + \cdots + |p_{rn} - a_n|}{|\bar{a}_1 - a_1| + \cdots + |\bar{a}_n - a_n|} \qquad (7)$$

$$\text{RRSE} = \frac{(p_{r1} - a_1)^2 + \cdots + (p_{rn} - a_n)^2}{(\bar{a}_1 - a_1)^2 + \cdots + (\bar{a}_n - a_n)^2} \qquad (8)$$

Where $a_1, a_2, \ldots . a_n$ are actual target values and $p_{r1}, p_{r2} \ldots \ldots p_{rn}$ predicted target values.

Correctly classified instances are near to 99% for all dataset and kappa statistic is near to 1 which proves that naïve Bayes classifier works excellently to differentiate DDoS attack data and normal request data in the given log file. A few numbers of instances are ignored as an unknown class in the given dataset in modeling. This has not affected the results characterization being very small in number.

**Table 4** Efficiency and Accuracy of Model in Terms of Highly Correct Results and Low Error Rate

| | **Training** | | **Cross Validation** | | **Testing** | |
|---|---|---|---|---|---|---|
| Correctly Classified Instances | **5728** | **98.96%** | **1227** | **98.87%** | **1226** | **98.95%** |
| Incorrectly Classified Instances | 60 | 1.04% | 14 | 1.13% | 13 | 1.05% |
| Kappa statistic | 0.98 | | 0.97 | | 0.98 | |
| Mean absolute error | 0.05 | | 0.05 | | 0.01 | |
| Root mean squared error | 0.14 | | 0.14 | | 0.08 | |
| Relative Absolute error % | 10.79 | | 11.19 | | 2.58 | |
| Root relative squared error % | 30.58 | | 31.49 | | 18.06 | |
| Total Number of Instances | 5788 | | 1241 | | 1239 | |
| Ignored Class Unknown Instances | 10 | | 2 | | 4 | |

Kappa = (observed accuracy - expected accuracy)/(1 - expected accuracy) based on confusion matrix

Table 5 describes accuracy level on different measures of quality. Measured values in this table prove that the DDoS attacks are detected with negligible error separating legitimate requests. True positive (TP) rate, precision, recall and F-score are all about 99%, ROC (Receiver operating characteristic), PRC (Precision recall curve) area are all more than 99% for all kind of datasets, Matthews correlation coefficient (MCC) is between 97% and 98% and False positive (FP) rate is near to 1% for training, cross-validation and test set which shows the precision of data set and accuracy of the model upon which decision is taken for the identification of DDoS attack.

**Table 5** Detailed average accuracy for both classes (normal request and attack)

| Quality Measures | Training | Cross validation | Testing |
|---|---|---|---|
| TP Rate | 0.99 | 0.989 | 0.99 |
| FP Rate | 0.016 | 0.009 | 0.008 |
| Precision | 0.99 | 0.989 | 0.99 |
| Recall | 0.99 | 0.989 | 0.99 |
| F-Measure | 0.99 | 0.989 | 0.99 |
| MCC | 0.976 | 0.973 | 0.976 |
| ROC Area | 0.996 | 0.997 | 0.999 |
| PRC Area | 0.995 | 0.996 | 0.998 |

**Figure 2** to **Figure 4** show the area under ROC for all category of the dataset of good users.

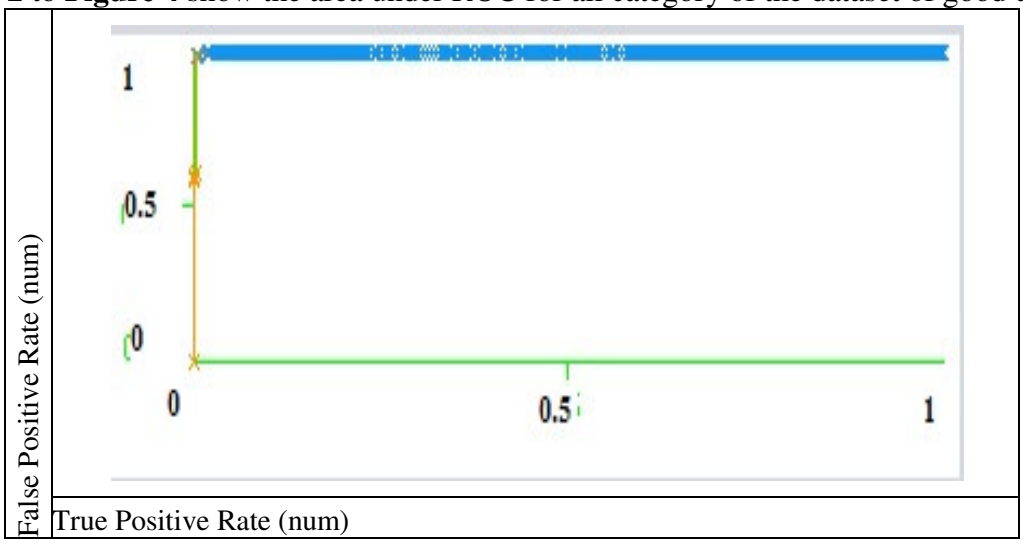


**Figure 2** Area under ROC =0.9964 for Training set of good user

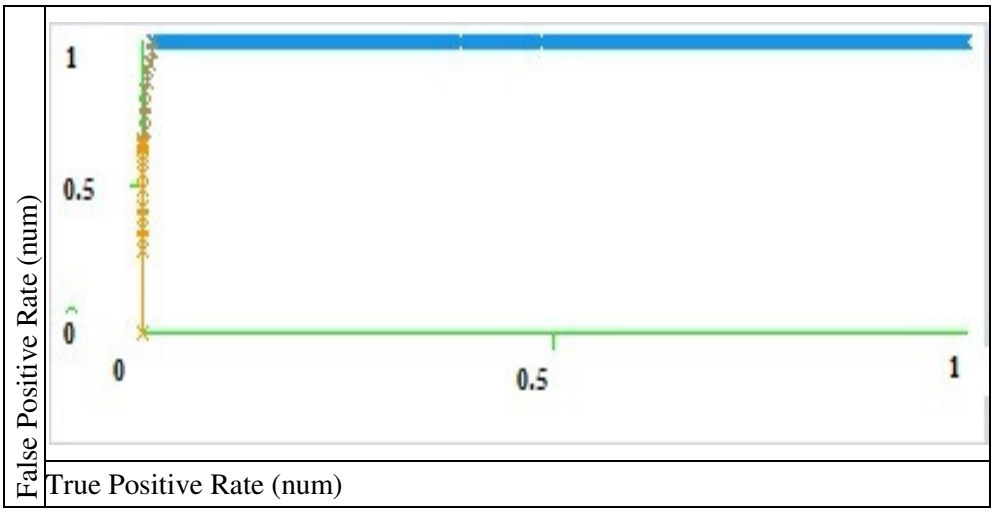


**Figure 3** Area under ROC =0.9969 for cross validation set of good user

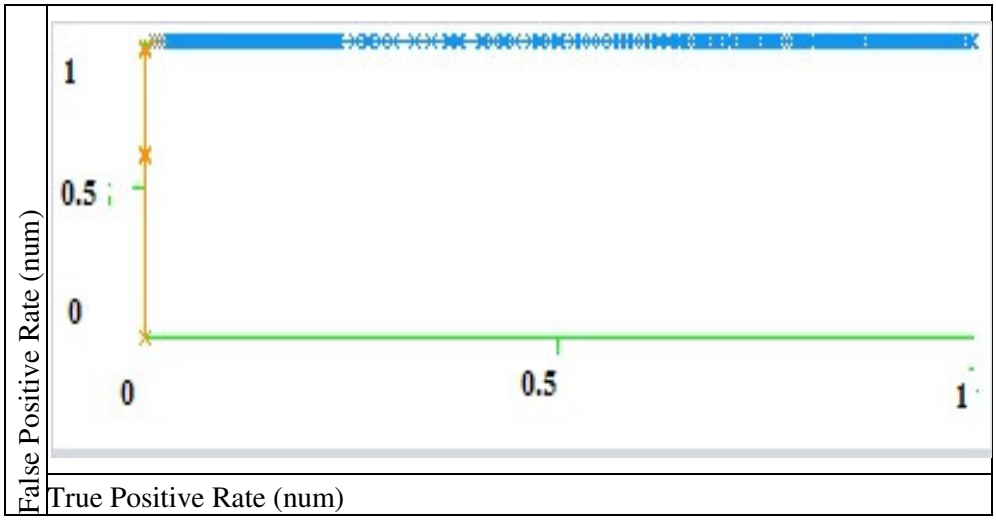


**Figure 4** Area under ROC =0.9985 for testing set of good user

## 4. CONCLUSION

Naïve Bayes classifier gives 99% accuracy in differentiating DDoS attack and legitimate request in the log file. The selected attributes with newly created attributes play the major role for high accuracy. False positive rate and true negative rate are very low in this model.

This model may be used for identification of bot participating in DDoS attack so that those IP addresses could be blocked. Other suspicious datasets of the different websites may be used in this model for detection of a DDoS attack. The attribute of log records and newly created attributes based on the pattern of particular attributes of log records may be used in other classification and clustering methods.

## REFERENCES

[1] Kenig, R., Manor, D., Gadot, Z. and Trauner, D., "DDoS Survival Handbook", pp.1-56, 2013.

[2] Mahadev, Kumar, V. and Kumar, K., "Classification of DDoS Attack Tool and Its Handling Techniques and Strategy at Application Layer", IEEE International Conference on Advances in Computing Communication and Automation (ICACCA), Oct 2016.

[3] Douligeris, C. and Mitrokotsa, A., "DDoS attacks and defense mechanisms: classification and state-of-the-art", Computer Networks, 44( 5), pp. 643-666, April 2004.

[4] Singh, N. A., Singh, K. J. and De, T., "Distributed denial of service attack detection using naive Bayes classifier through info gain feature selection", Proc. Int. Conf. Inform. Anal., pp. 54:1-54:9, 2016

[5] Liao, Q.; Li, H.; Kang, S.; Liu, C.: Application layer DDoS attack detection using cluster with label based on sparse vector decomposition and rhythm matching. Security and Communication Networks, 8(17), pp. 3111–3120 (2015)

[6] Zhou, W.; Jia, W.; Wen, S.; Xiang, Y.; Zhou, W. Detection and defense of application-layer {DDoS} attacks in backbone web traffic. Future Generation Computer Systems, 2014, 38, pp.36–46

[7] Raj Kumar, P. A. and Selvakumar, S., "Distributed denial of service attack detection using an ensemble of neural classifier", Computer Communications, 34(11), pp.1328-1341, Julu 2011

[8] Agrawal, S. and Rajput, R. S., "Denial of Services Attack Detection using Random Forest Classifier with Information Gain", International Journal of Engineering Development and Research, 5(3), pp. 929-938.

[9] Mahadev and Kumar, V., "Behaviour Analysis of DDoS Attack and Its Detection", International Journal of Computer Sciences and Engineering, 5(5), pp. 139-144, 2018.