# Detecting Malicious Cloud Bandwidth Consumption using Machine Learning

Chidananda Murthy P. [#1], A.S. Manjunatha [*2], Anku Jaiswal [#3], Madhu B.R. [#4]

[#1] Research Scholar, Jain University, Bengaluru, Karnataka, India.
chidananda.murthy@gmail.com

[*2] Manvish eTech Pvt. Ltd., Bengaluru, Karnataka, India.
asmanju@manvish.com

[#3] M.Tech Student, SET, Jain University, Bengaluru, Karnataka, India.
jaiswalaku@gmail.com

[#4] Research Scholar, Jain University, Bengaluru, Karnataka, India.
br.madhu@jainuniversity.ac.in

***Abstract*— One of the most difficult and unsolved issues in network is the security issue, because of continuous evolving nature of both threats and the measures used to detect and avoid threats. Among different types of attacks, one of the most vulnerable attacks in network security are bots that consume the resources maliciously and exhaust them. Malicious Cloud Bandwidth Consumption (MCBC) attack is a new type of attack, where the aim of the attacker is to consume the bandwidth maliciously, in turn causing the financial burden to the cloud service host. MCBC is generally vulnerable to the internet based web services in public cloud. MCBC mainly aims at frequently consuming the bandwidth in a slow manner, hence affecting the pay-as-you-go utility model, causing the consumer in the form of monetary loss. Unlike DDOS attack which is short lived and makes the resource unavailable to the user, MCBC attack is a long term attack which slowly attacks the target for an extended period and remains undetectable. As this attack does not affect the availability issue immediately, it is not discussed much as DDOS attack. This paper discuss about how machine learning technique can be used to detect the MCBC attack in the form of request per second, any traffic violating this range are classified as MCBC attack. The proposed system consists of using semi supervised machine learning which uses labeled network traffic for building model and unlabeled traffic to classify using the built model.**

**Keyword-** Network Security, Machine learning, MCBC attack, Supervised learning

## I. Introduction

Network security is affected by different types of attack such as DDOS, zero day attack, http attacks. Different types of IDS have been developed to prevent these attacks. DDOS attack which is one of the most vulnerable attacks mainly aims at making the resources unavailable to the user. The network traffic generated by the attacker is in such a huge amount that it makes the resources unavailable to the legitimate user. Different techniques have been developed to prevent this attack. DDOS attack is easily detectable as the number of traffic is generally high and the behavior of DDOS traffic is similar to the normal traffic. A new type of attack which generally affects the utility model of the internet facing web services is the MCBC attack. Unlike DDOS attack which attack the target for a short period and hence making a huge loss in the form of resources or money, MCBC attack clients targets the utility model for an extended period and hence affecting the consumer in the form of monetary loss. As the web services are hosted on the cloud server, the consumer has to pay on the basis of usage. For each data sent and received from the web services, certain amount is charged to the consumer. Although the loss is not so high if calculated for a small period but as the day increases the amount to be paid also increases. The behaviour of MCBC attack is different from DDOS attack as MCBC request mingles with normal behaviour. In the paper we have proposed a technique which is used to detect MCBC attack. A threshold is given to different types of attacks such as the request per second for normal traffic is 10-19, request per second for MCBC traffic is 20-50 and any traffic above 50 is detected as DDOS. Machine learning technique with supervised and semi-supervised learning is used to detect the network traffic as MCBC. Further the monetary loss for one month period is calculated and shown.

## II. RELATED WORK

Cloud Computing is a technology that is based on the utility pricing model which is pay as you go service for the resources consumed. As we pay for gas and electricity, similarly cloud consumer has to pay for the resources consumed such as storage, bandwidth [1]. Unlike other attack such as DDOS which is vulnerable to the cloud utility model and is monitored by Cloud Service Provider (CSP), CSPs do not monitor attacks which affect cloud consumer application; hence cloud consumer has to take action to prevent such attacks [2]. The main aim of MCBC attack is not to affect the resources utilization by consumer but to slowly affect the utility model. The nature of MCBC attack is subtle and goes undetectable [3]. Much cloud computing adopter has utilized different services such as search engines, application hosting and web hosting [4]. Due to pay as you go service it is easy for consumer to utilize the services. Many high CSP such as Google, Amazon has gone through the loss of availability due to different attacks such as DDOS [5][6]. Current detection techniques mainly focus on excessive HTTP request over a short period of time , hence MCBC attack goes undetected[7].The dataset used in this paper is from Honeynet dataset. This website is attacked by various clients over a period of one month. So many methods have been developed to detect MCBC attack. Machine learning technique which uses supervised and semi- supervised learning is used to detect MCBC attack in the paper. Machine learning tool called WEKA is used to create a model which is used to detect MCBC attack [8].

## III. MCBC ATTACK

The main aim of attacker is to frequently consume the resources such that it does not affect the availability of resources but affect the utility model in a slow manner. Unlike DDOS attack this attack is not for short period and does not massively affect the user. This type is attack is extended to a long period so that the attacker may benefit for a long period and the consumer is affected in the form of monetary loss. The attack scenario of MCBC attack consists of attacker and a normal user. The attacker generates request in such a way that it is blended with normal request and remains undetectable. Such type of attack does not affect the consumer in the form of resource availability but incurs a monetary loss. As most of the web services in cloud are based on pay as you go model this attack directly affect the consumer in form of monetary loss at the end of month. A threshold is given to detect this type of attack. Any normal user can only generate a maximum request of 10-19 per second. Any request between 20 to 50 can be considered as MCBC attack if it is continued for a particular day. For the proposed technique request per second is considered as main criteria to detect the attack.

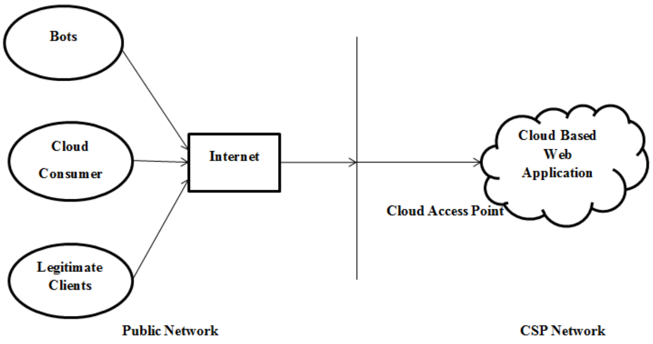


Fig 1: MCBC Attack Scenario

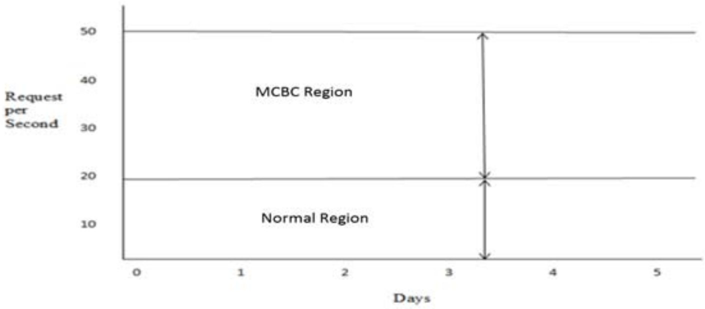


Fig 2 : MCBC Attack Region

## IV. Dataset Description

The dataset taken in this paper is a log file from Honeynet. Honeynet is a network setup to invite attacks so that attacker's activities can be observed and data can be used for improving the network security. The dataset is collected for one month from Feb 1 to Feb 27.The dataset consist of 28 fields [9][10].

### *A. Dataset Analysis and Pre-processing*

The log file collected for one month consist of different information and is analysed for the paperwork. Highest traffic was originated from the following IP addresses. The Algorithm for pre-processing of dataset is as follows:

```
Step1: Convert log file to CSV file.
Step2: Reduction of attributes.
Step3: Adding an attributes called request/second.
Step4: CountNumber of request for each second.
Step5: Add an attribute called class.
Step6: If req/sec <19 name the class as Normal
Else if 50<req/sec<19 name the class as MCBC
```

## V. Normal and MCBC Client Behaviour

Normal client who request a particular site only generate a request of 5 to 10 per second. Sometimes the request may be as high as equal to 19. If a request/sec is more than 20 then, the traffic is considered to be generated by a bot or some intruder. An intruder can generate a traffic of 1000 request/sec for DDOS attack with the help of thousands bot under the control of a botmaster. As MCBC attacker mingle with the behavior of normal traffic it can be considered as 20 to 50 request/sec.

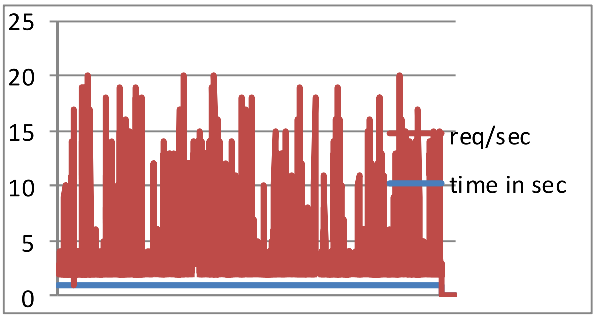


Fig 3: Normal Behavior Graph from Honeynet Dataset

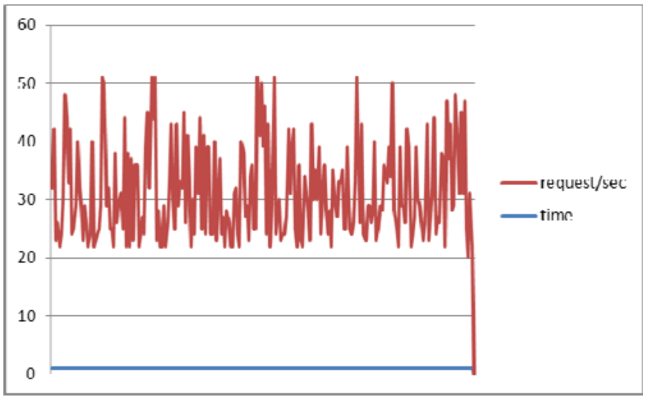


Fig 4: MCBC Behavior Graph from Honeynet Dataset

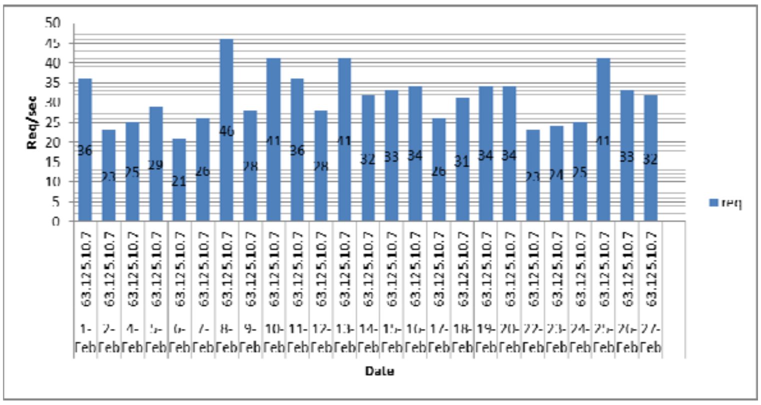
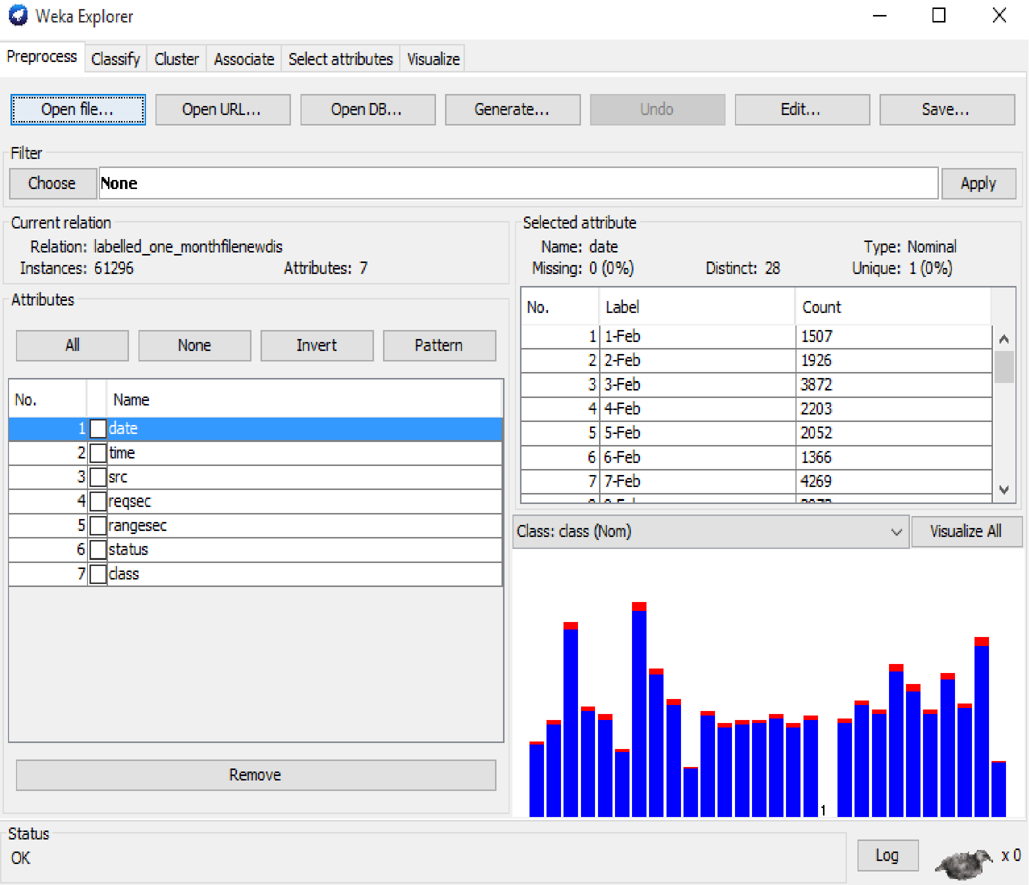


Fig 5: IP Address that fall in MCBC Region for 1 Month

## VI. EXPERIMENTAL SETUP

The experimental setup consists of using machine learning tool called WEKA with a Pentaho environment. WEKA is an open source machine learning tool with data mining algorithm which is used to create model. Preprocessed dataset with class which is labeled as normal or MCBC is used to create a model in WEKA.The model created in WEKA can be integrated with WekaScoring in Pentaho to find the new traffic which does not have a class and the class can be predicted as normal or MCBC based on the range of request/sec



Filter: None

Current relation
Relation: labelled_one_monthfilenewdis
Instances: 61296 Attributes: 7

Attributes

| No. | Name |
|---|---|
| 1 | □ date |
| 2 | □ time |
| 3 | □ src |
| 4 | □ reqsec |
| 5 | □ rangesec |
| 6 | □ status |
| 7 | □ class |

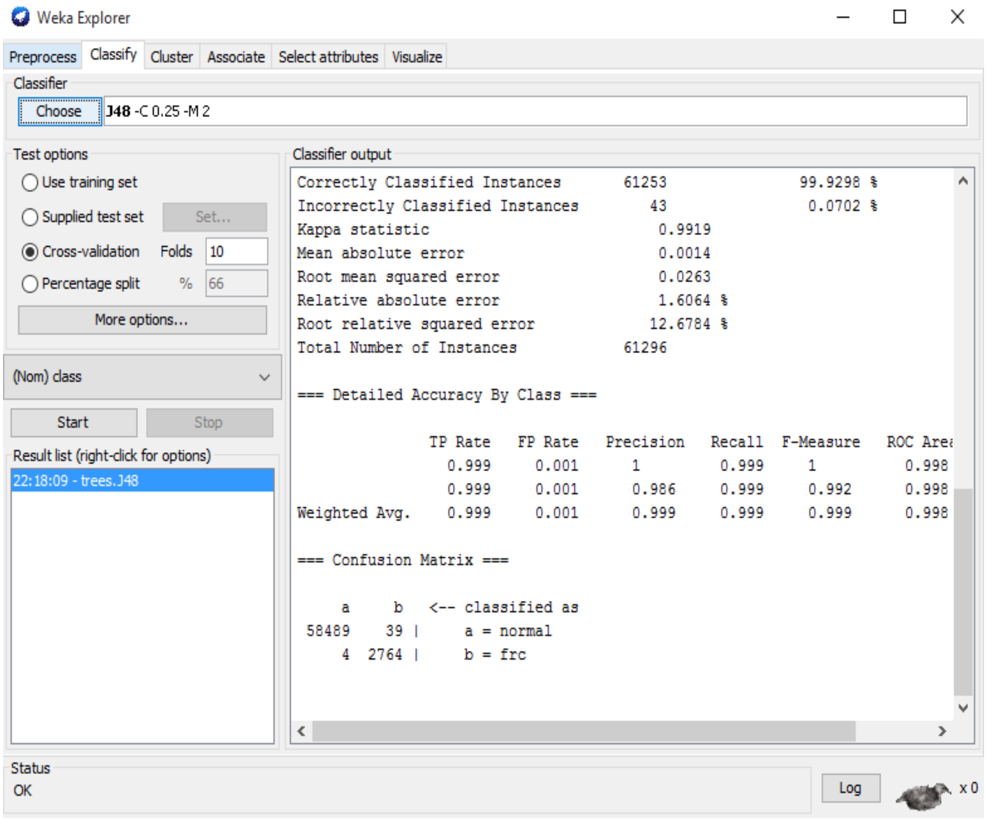
Selected attribute
Name: date Type: Nominal
Missing: 0 (0%) Distinct: 28 Unique: 1 (0%)

| No. | Label | Count |
|---|---|---|
| 1 | 1-Feb | 1507 |
| 2 | 2-Feb | 1926 |
| 3 | 3-Feb | 3872 |
| 4 | 4-Feb | 2203 |
| 5 | 5-Feb | 2052 |
| 6 | 6-Feb | 1366 |
| 7 | 7-Feb | 4269 |

Class: class (Nom)

Status
OK

Fig.6: Data Represented in WEKA



J48 -C 0.25 -M 2

Folds 10

% 66

(Nom) class

22:18:09 - trees.J48

Status: OK

```
Correctly Classified Instances       61253               99.9298 %
Incorrectly Classified Instances        43                0.0702 %
Kappa statistic                          0.9919
Mean absolute error                      0.0014
Root mean squared error                  0.0263
Relative absolute error                  1.6064 %
Root relative squared error             12.6784 %
Total Number of Instances            61296

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall   F-Measure   ROC Area
                 0.999     0.001      1          0.999     1           0.998
                 0.999     0.001      0.986      0.999     0.992       0.998
Weighted Avg.    0.999     0.001      0.999      0.999     0.999       0.998

=== Confusion Matrix ===

     a     b   <-- classified as
 58489    39 |     a = normal
     4  2764 |     b = frc
```

Fig 7: Model Creation using Machine Learning

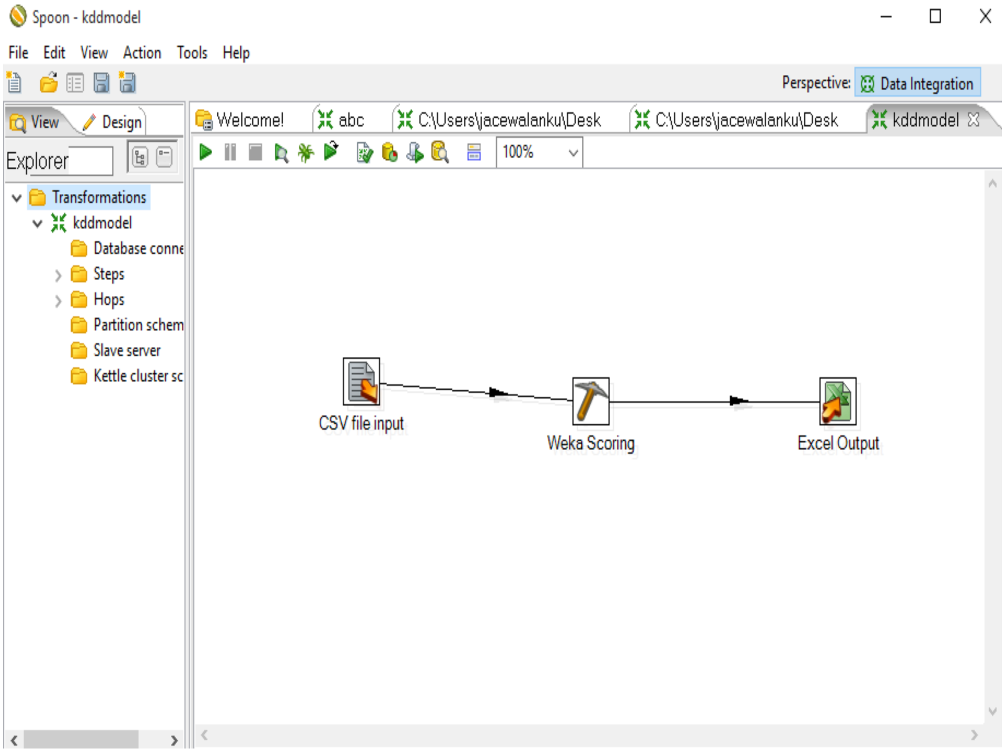
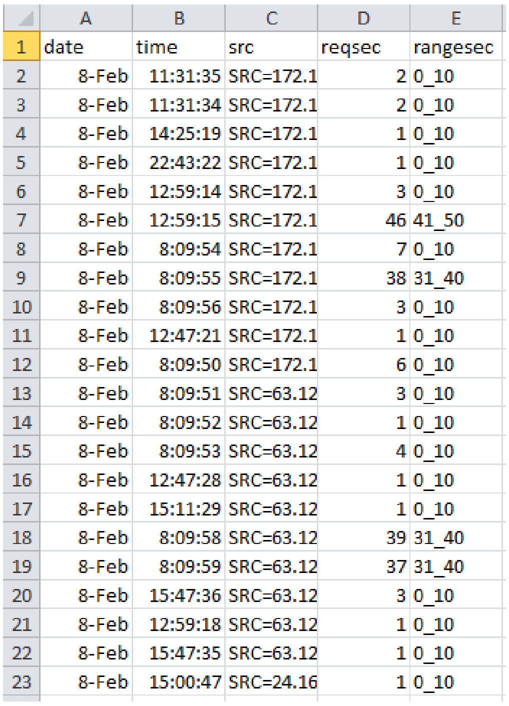


Fig 8: Experimental Setup for Proposed System

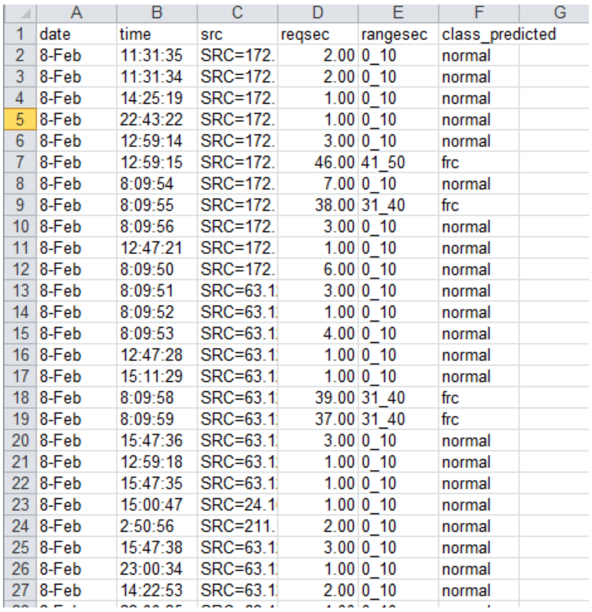
## VII. RESULT ANALYSIS

From the experiment it is observed that the new traffic given is automatically labeled as normal or MCBC based on the range of request/sec. The traffic having request/sec between 1 to 19 is labeled as normal and the traffic having request/sec between 20 to 50 is labeled as MCBC automatically by the model that we developed in experiment. As observed, figure 10 consist of unlabeled data which is given to the model and figure 11 consist of the data which is labeled as normal or MCBC by the model.



| | A | B | C | D | E |
|---|---|---|---|---|---|
| 1 | date | time | src | reqsec | rangesec |
| 2 | 8-Feb | 11:31:35 | SRC=172.1 | 2 | 0_10 |
| 3 | 8-Feb | 11:31:34 | SRC=172.1 | 2 | 0_10 |
| 4 | 8-Feb | 14:25:19 | SRC=172.1 | 1 | 0_10 |
| 5 | 8-Feb | 22:43:22 | SRC=172.1 | 1 | 0_10 |
| 6 | 8-Feb | 12:59:14 | SRC=172.1 | 3 | 0_10 |
| 7 | 8-Feb | 12:59:15 | SRC=172.1 | 46 | 41_50 |
| 8 | 8-Feb | 8:09:54 | SRC=172.1 | 7 | 0_10 |
| 9 | 8-Feb | 8:09:55 | SRC=172.1 | 38 | 31_40 |
| 10 | 8-Feb | 8:09:56 | SRC=172.1 | 3 | 0_10 |
| 11 | 8-Feb | 12:47:21 | SRC=172.1 | 1 | 0_10 |
| 12 | 8-Feb | 8:09:50 | SRC=172.1 | 6 | 0_10 |
| 13 | 8-Feb | 8:09:51 | SRC=63.12 | 3 | 0_10 |
| 14 | 8-Feb | 8:09:52 | SRC=63.12 | 1 | 0_10 |
| 15 | 8-Feb | 8:09:53 | SRC=63.12 | 4 | 0_10 |
| 16 | 8-Feb | 12:47:28 | SRC=63.12 | 1 | 0_10 |
| 17 | 8-Feb | 15:11:29 | SRC=63.12 | 1 | 0_10 |
| 18 | 8-Feb | 8:09:58 | SRC=63.12 | 39 | 31_40 |
| 19 | 8-Feb | 8:09:59 | SRC=63.12 | 37 | 31_40 |
| 20 | 8-Feb | 15:47:36 | SRC=63.12 | 3 | 0_10 |
| 21 | 8-Feb | 12:59:18 | SRC=63.12 | 1 | 0_10 |
| 22 | 8-Feb | 15:47:35 | SRC=63.12 | 1 | 0_10 |
| 23 | 8-Feb | 15:00:47 | SRC=24.16 | 1 | 0_10 |

Fig 9 : Unlabelled Data



| | A | B | C | D | E | F | G |
|---|---|---|---|---|---|---|---|
| 1 | date | time | src | reqsec | rangesec | class_predicted | |
| 2 | 8-Feb | 11:31:35 | SRC=172. | 2.00 | 0_10 | normal | |
| 3 | 8-Feb | 11:31:34 | SRC=172. | 2.00 | 0_10 | normal | |
| 4 | 8-Feb | 14:25:19 | SRC=172. | 1.00 | 0_10 | normal | |
| 5 | 8-Feb | 22:43:22 | SRC=172. | 1.00 | 0_10 | normal | |
| 6 | 8-Feb | 12:59:14 | SRC=172. | 3.00 | 0_10 | normal | |
| 7 | 8-Feb | 12:59:15 | SRC=172. | 46.00 | 41_50 | frc | |
| 8 | 8-Feb | 8:09:54 | SRC=172. | 7.00 | 0_10 | normal | |
| 9 | 8-Feb | 8:09:55 | SRC=172. | 38.00 | 31_40 | frc | |
| 10 | 8-Feb | 8:09:56 | SRC=172. | 3.00 | 0_10 | normal | |
| 11 | 8-Feb | 12:47:21 | SRC=172. | 1.00 | 0_10 | normal | |
| 12 | 8-Feb | 8:09:50 | SRC=172. | 6.00 | 0_10 | normal | |
| 13 | 8-Feb | 8:09:51 | SRC=63.1 | 3.00 | 0_10 | normal | |
| 14 | 8-Feb | 8:09:52 | SRC=63.1 | 1.00 | 0_10 | normal | |
| 15 | 8-Feb | 8:09:53 | SRC=63.1 | 4.00 | 0_10 | normal | |
| 16 | 8-Feb | 12:47:28 | SRC=63.1 | 1.00 | 0_10 | normal | |
| 17 | 8-Feb | 15:11:29 | SRC=63.1 | 1.00 | 0_10 | normal | |
| 18 | 8-Feb | 8:09:58 | SRC=63.1 | 39.00 | 31_40 | frc | |
| 19 | 8-Feb | 8:09:59 | SRC=63.1 | 37.00 | 31_40 | frc | |
| 20 | 8-Feb | 15:47:36 | SRC=63.1 | 3.00 | 0_10 | normal | |
| 21 | 8-Feb | 12:59:18 | SRC=63.1 | 1.00 | 0_10 | normal | |
| 22 | 8-Feb | 15:47:35 | SRC=63.1 | 1.00 | 0_10 | normal | |
| 23 | 8-Feb | 15:00:47 | SRC=24.1 | 1.00 | 0_10 | normal | |
| 24 | 8-Feb | 2:50:56 | SRC=211. | 2.00 | 0_10 | normal | |
| 25 | 8-Feb | 15:47:38 | SRC=63.1 | 3.00 | 0_10 | normal | |
| 26 | 8-Feb | 23:00:34 | SRC=63.1 | 1.00 | 0_10 | normal | |
| 27 | 8-Feb | 14:22:53 | SRC=63.1 | 2.00 | 0_10 | normal | |

Fig 10 : Output Labelled Data

## VIII. CONCLUSION

The proposed system is able to detect the traffic as normal and MCBC using machine learning technique. The data from the website is preprocessed in the form that can be used by model. The future work can be attack cost calculation and representation of the attack traffic in dashboard form which can be used by the administrator for decision making.

## REFERENCES

[1] Peter Mell and Timothy Grance. The NIST Definition of Cloud Computing (Draft). http://www.nist.gov/itl/cloud/upload/cloud-def-v15.pdf, February 2012.

[2] Charlie Kaufman. What's Different About Security in a Public Cloud? In Proceedings of the 3rd ACM Workshop on Cloud Computing Security Workshop, CCSW '11, pages27–28. ACM, October 2011.

[3] I Sheng Wen, WeijiaJia, Wei Zhou, Wanlei Zhou, and ChuanXu. CALD: SurvivingVarious Application-Layer DDoS Attacks That Mimic Flash Crowd. In 4th InternationalConference on Network and System Security (NSS), pages 247–254, 2010.

[4] Amazon. Case Studies. http://aws.amazon.com/solutions/case-studies/, September 2010.

[5] Cade Metz. DDoS Attack Rains Down on Amazon Cloud. http://www.theregister.co.uk/2009/10/05/amazon_bitbucket_outage/, October 2009.

[6] . G. Oikonomou and J. Mirkovic. Modeling Human Behavior for Defense AgainstFlash-Crowd Attacks. In IEEE International Conference on Communications, pages 1–6, 2009

[7] Jaeyeon Jung, Balachander Krishnamurthy, and Michael Rabinovich. Flash Crowds andmDenial of Service Attacks: Characterization and Implications for CDNs and Web Sites. In Proceedings of the 11th International Conference on World Wide Web, pages 293–304, New York, NY, USA, 2002. ACM.

[8] Chidananda Murthy P., Dr A.S. Manjunatha, Anku Jaiswal, Madhu B.R. "Building Efficient Classifiers For Intrusion Detection With Reduction Of Features", International Journal of Applied Engineering Research ISSN 0973-4562 Volume 11, Number 6 (2016) pp. 4590-4596 © Research India Publications. http://www.ripublication.com.

[9] **http://old.honeynet.org/scans/scan30/ honeynet data download site**

[10] Honeynet Challenge of the month scan 30 Submitted by SabyasachiChakrabartyBasudevSaha