TRIBHUVAN UNIVERSITY

INSTITUTE OF ENGINEERING

CENTRAL CAMPUS, PULCHOWK

THESIS NO.: 071MSCS664

An Entropy-based Detection for Tracing DDoS Attack Packets using Clustering with Machine Learning

By

Sanil Maharjan

A THESIS

SUBMITTED TO THE DEPARMENT OF ELECTRONICS AND COMPUTER ENGINEERING IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF MASTER OF SCIENCE IN COMPUTER

DEPARMENT OF ELECTRONICS AND COMPUTER ENGINEERING

LALITPUR, NEPAL

NOVEMBER 2018

ACKNOWLEDGEMENT

I would like to express my sincere gratitude to **Babu Ram Dawadi**, my thesis supervisor for the constant guidance with his insightful ideas and valuable suggestions and encouragement during the period of thesis.

I would like to express my sincere thanks and indebtedness to our Head of Department **Associate Prof. Dr. Surendra Shrestha, Prof. Dr. Subarna Shakya, Prof. Dr. Sashidhar Ram Joshi, Associate Prof. Dr. Dibakar Raj Pant** and **Associate Prof. Dr. Sanjeeb Pandey** for their encouragement and precious guidance.

I would also like to thank **Dr. Aman Shakya**, Program Coordinator of Master's Degree, for his constant focus on research activity, choosing Thesis Topic and cooperation to give out the best. Last but not least I would like to thank everyone who directly or indirectly helped me to make this final report successful.

Last but not least, I want to thank Vianet Communications Pvt. Ltd. for providing real ISP traffic dataset that helps to validate the work.

COPYRIGHT

APPROVAL PAGE

The undersigned certify that they have read and recommended to the Department of Electronics and Computer Engineering for acceptance, a thesis entitled "**An Entropy-based Detection Method for Tracing DDoS Attack Packets using Clustering with Machine Learning",** submitted by **Sanil Maharjan** in partial fulfillment of the requirement for the award of the degree of **"Master of Science in Computer System and Knowledge Engineering"**.

.....

Supervisor: Babu Ram Dawadi

Lecturer, Department of Electronics and Computer Engineering

Institute of Engineering, Pulchowk Campus

.....

External Examiner: Adesh Khadka

IT Director, Ministry of Finance

Government of Nepal

.....

Committee Chairperson: Dr. Aman Shakya

Program Coordinator

Department of Electronics and Computer Engineering

Date: ……...

DEPARTMENTAL ACCEPTANCE

The thesis entitled "**An Entropy-based Detection Method for Tracing DDoS Attack Packets using Clustering with Machine Learning",** submitted by **Sanil Maharjan** in partial fulfillment of the requirement for the award of the degree of **"Master of Science in Computer System and Knowledge Engineering"** has been accepted as a bonafide record of work independently carried out by him in the department.

.....

Associate Prof. Dr. Surendra Shrestha

Head of the Department

Department of Electronics and Computer Engineering,

Pulchowk Campus,

Institute of Engineering,

Tribhuvan University,

Nepal

ABSTRACT

The importance of internet in everyone's life is similar like oxygen. Today it is all about online reputation, internet marketing, online business, online degrees, social media presence and internet banking. Therefore, the availability of internet is very critical for the socio economic growth. One of the serious issue in the current Internet, is the denial-of-service (DoS) attack that prevent the legitimate users from serving by servers.

The main step for stopping DDoS attacks is to detect attacks and generate alarm so that necessary precaution need to be taken. The challenging part for network security is detection of DDoS attacks. In this thesis, an approach is made aims at detecting DDoS attacks in network using Entropy based detection algorithm. The proposed model is being developed in intention to bridge the system complexities acquired in detection by advanced techniques like machine learning, deep neural network with the traditional approach such as Clustering without compromising in accuracies as they possessed. Therefore, the entropy based technique hybrid with machine learning algorithm, K-Nearest Neighbors (KNN) is adapted in which entropy is calculated not only with singular parameter but also with the parameters like source IP, source port, destination IP and destination port with respect to time widows of 1sec, 5sec, 10sec and 15sec and compared with the threshold for each respective parameter. The detection threshold is determined using unsupervised data mining algorithm which is dynamic in nature. For this, k-means clustering algorithm is used since it is much faster than other clustering algorithms. To reduce false alarm and classifying the attacks, K-Nearest Neighbors (KNN) algorithm is used which maximizes the accuracy against clustering alone. The network traffic profiling is also maintained so as entropies against all feature parameters as mentioned earlier which helps in determining flow pattern. Moreover, packet count per second and average packet length per second are also calculated for adding attribute on precision detection. Since, the uses of bandwidth in the network during attacks get significantly higher than the normal the traffic flow. Therefore, the bandwidth is being monitored so closely throughout the testing period.

KEYWORDS: DoS, DDoS, Entropy, Bandwidth, KNN

LIST OF FIGURES

ABBREVIATION

DoS	Denial of Service
DDoS	Distributed Denial of Service
CPU	Central Processing Unit
NTP	Network Time Protocol
DNS	Domain Name System
UDP	User Datagram Protocol
TCP	Transmission Control Protocol
HTTP	Hyper Text Transfer Protocol
SYN	Synchronization
KNN	K-nearest Neighbors
ACK	Acknowledge
TP	True Positive
FN	False Negative
FP	False Positive
TN	True Negative
ADR	Attack Detection Rate
IDS	Intrusion Detection System
IP	Internet Protocol

LIST OF TABLES

TABLE OF CONTENT

CHAPTER 1: INTRODUCTION

1.1 Background

Distributed Denial of Service (DDoS) attacks are the attacks committed by attackers with a huge amount of request packets to targets (victims) by using amounts of compromised computers (zombies), which rapidly exhaust available resources of target systems and intentionally disrupt network services. The DDoS attack aims to consume resources, including bandwidth, memory and CPU computing capacity, until exhausted to prevent legal traffic obtaining these resources.

Generally, in DDoS attack, the attacker begins by exploiting a vulnerability in the system and making it the DDoS master. The master system identifies other vulnerable systems and gains control over them by either infecting the systems with malware or through bypassing the authentication controls. A computer under the control of an intruder is known as bot or zombie. The attacker creates comand-and-control server to command the network of bots, called botnet. Therefore, the DDoS attacks is deployed by utilizing multiple compromised computer systems as sources of attack.

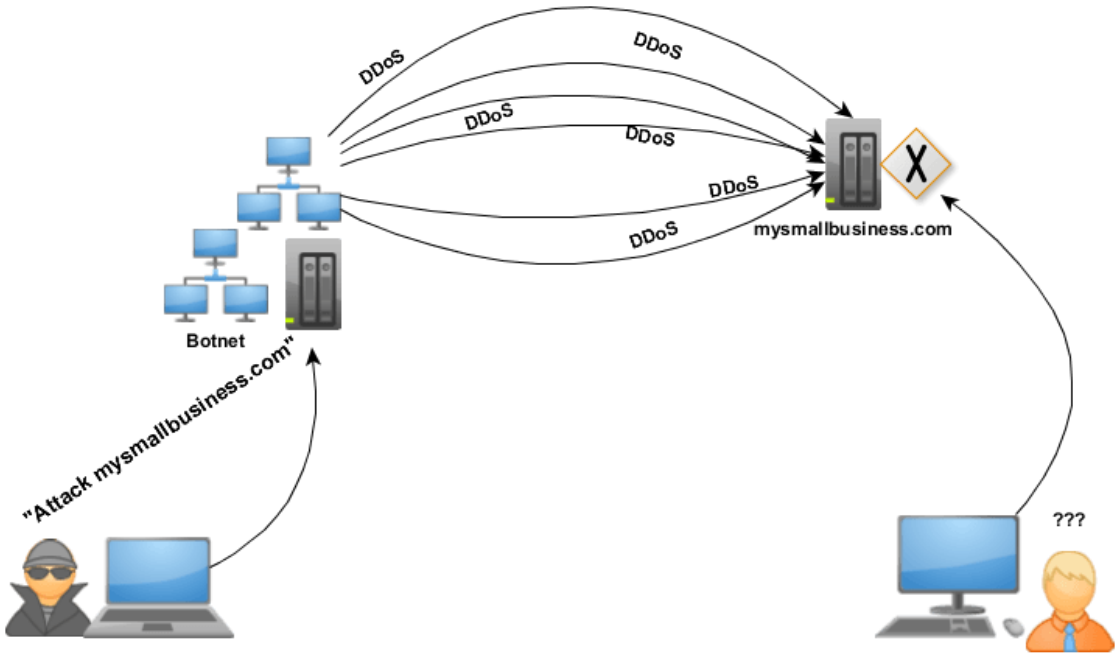

Figure 1: DDos attack Overview [9]

From above figure1: it is clearly seen that the attacker controls the bot through handlers and aligned all to attack at same instant with common goal to disrupt the services offered by mymaillbusiness.com which results the legitimate users get service denied thus suffered.

There are three categories where DDoS attacks fall into:

1. Volumetric Attacks

The attacks where massive amount of traffic used to saturate the bandwidth of the target. It is easy to generate by simply employing any amplication techniques. Examples: NTP Amplication, DNS Amplification, UDP Flood, TCP Flood

2. Protocol Attacks

The attacker get target in-accessible by exploiting a weakness in the Layer 3 and Layer 4 in protocol stack. Examples: Syn Flood, Ping of Death

3. Application Attacks

In this attacks, attackers exploit a weakness in the Layer 7 of protocol stack. It is done by establishing a connection with the target and then exhaust the server's resources by monopolizing the processes. Example: HTTP Flood, Attack on DNS Services.

ENTROPY

In information theory, entropy is a measure of the uncertainty in a random variable which quantifies the expected value of the information contained in a message. The formula for entropy was introduced by Claude E. Shannon in his 1948 paper "A Mathematical Theory of communication" and defined as

$$\text{Entropy (H)} = -\sum_{i=1}^{n} \text{p(x}i\text{)log2 p(x}i\text{)} \dots\dots\dots (1)$$

Where,

$x_1, x_2, x_3 \ldots\ldots, x_n$ represent random variables from an information source that has (n) different values and the probability of (x_i) appears in information sample identified by (x_i). The value of entropy should be greater or equal to zero i.e. $H \geq 0$ and probability range must be in $0 \leq P(x_i) \leq 1$. If the entropy (H) is close to zero ($H \rightarrow 0$), it refers to higher similarity in the sample and lower level of uncertainty and if ($H>0$), it refers to lower similarity in the sample and higher level of uncertainty.

CLUSTERING

Clustering is an unsupervised data mining technique that attempts grouping of a particular set of objects based on their characteristics, aggregating them according to their similarities. There are multiple clustering methods such as K-means or Hierarchical Clustering. Often, a measure of distance from point to point is used to find which category a point should belong to as with K-means and for this thesis, k-means is chosen for clustering since it is fast, robust and easier to understand. The objective function for the K-means clustering algorithm is the squared error function which is defined as:

$$J(V) = \sum_{i=1}^{c} \sum_{j=1}^{ci} ((xi - vj))^2 \ldots\ldots\ldots\ldots (2)$$

Where,

$||x_i - v_j||2$ = the Euclidean distance between x_i and v_j

ci = the number of data points in i^{th} cluster

c = the number of cluster centers

1.2 Problem Definition

DDoS attackers send a huge amount of traffic or by using an army of zombies that mimicked the normal packets from regular users. Even after the attack has been stopped, these compromised users still suffer from performance degradation or even the denial of service. ICMP (Internet Control Message Protocol) flood and SYN flood are the most common attacks. ICMP flood is a type of DDoS attack in which attackers send a large number of ICMP echo-reply packets to victims. This attack exhausts not only the resources of victims but also the entire network. SYN flood utilizes the defect of the TCP three-way-handshake to maliciously increase the number of the half-open connections making system resources occupied so that the legitimate users cannot obtain services from servers. Another damage is that an application or a protocol on the victim forced to be freeze or reboot by sending a few malformed packets.

1.3 Objective

1) To detect DDoS attacks based on entropy along with clustering algorithm integration.

2) To introduce dynamic threshold parameter as key measuring parameter determined by K-means algorithm.

3) To introduce KNN algorithm for classifying the types of DDoS attacks.

1.4 Scope

Basically, there are lots of DDoS attacking mechanism, it is difficult to include all sorts of attacking technique in the research. Therefore, to narrow down this thesis, it dealt with TCP flood and UDP Flood packets to address volumetric based DDoS attacks. Similarly, SYN Flood known as half open connection attack and Ping of Death, that is the over sizing payload for ping packets are also covered to include protocol based DDoS Attacks.

CHAPTER 2: LITERATURE REVIEW

Most of Researchers have already been proposed different entropy based detection techniques in order to mitigate cyber threats launched via DDoS attacks. The researchers named Jaswinder Singh, Monika Sachadeva and Krishan Kumar analyzed Source IP based entropy using anomaly detection algorithm [1]. Jisa David and Ciza Thomas developed the model of detection using entropy based approach which is on flow based. The working mechanism was comparing the difference in entropy of flow count at each instant and mean value of entropy at same interval with the adaptive threshold [2].

Wesam Bhaya and Mehdi Ebady Manaa introduced unsupervised data mining technique known as Clustering Using Representative (CURE) to detect the DDoS attack in the network flow [3]. Xi Qin, Tongge Xu and Chao Wang made an overall assessment on flow based entropy by modeling the pattern using clustering [5].

The machine learning techniques like Navies Bayes, K-Nearest Neighbors (KNN), Fuzzy c-means, K-means, Support Vector Machines (SVM) are examined using the features based on information gain and Chi-Square by Manjula and Anitha [13].

4.1 Proposed Model

The proposed model is being developed in intention to bridge the system complexities acquired in detection by advanced techniques like machine learning, deep neural network with the traditional approach such as Clustering without compromising in accuracies as they possessed. Therefore, the entropy based technique hybrid with machine learning algorithm, K-Nearest Neighbors (KNN) is adapted in which entropy is calculated not only with singular parameter but also with the parameters like source IP, source port, destination IP and destination port with respect to time widows of 1sec, 5sec, 10sec and 15sec and compared with the threshold for each respective parameter. The detection threshold is determined using unsupervised data mining algorithm which is dynamic in nature. For this, k-means clustering algorithm is used since it is much faster than other clustering algorithms. To reduce false alarm and classifying the attacks, K-Nearest Neighbors (KNN) algorithm is used which maximizes the accuracy against clustering alone. The network traffic profiling is also maintained so as entropies against all feature parameters as mentioned earlier which helps in determining flow pattern. Moreover, packet count per second and average packet length per second are also calculated for adding attribute on precision detection. Since, the uses of bandwidth in the network during attacks get significantly higher than the normal the traffic flow. Therefore, the bandwidth is being monitored so closely throughout the testing period.

CHAPTER 3: METHODOLOGY

A proactive detection process of DDoS attacks is described in figure 2 showing overview of whole system.

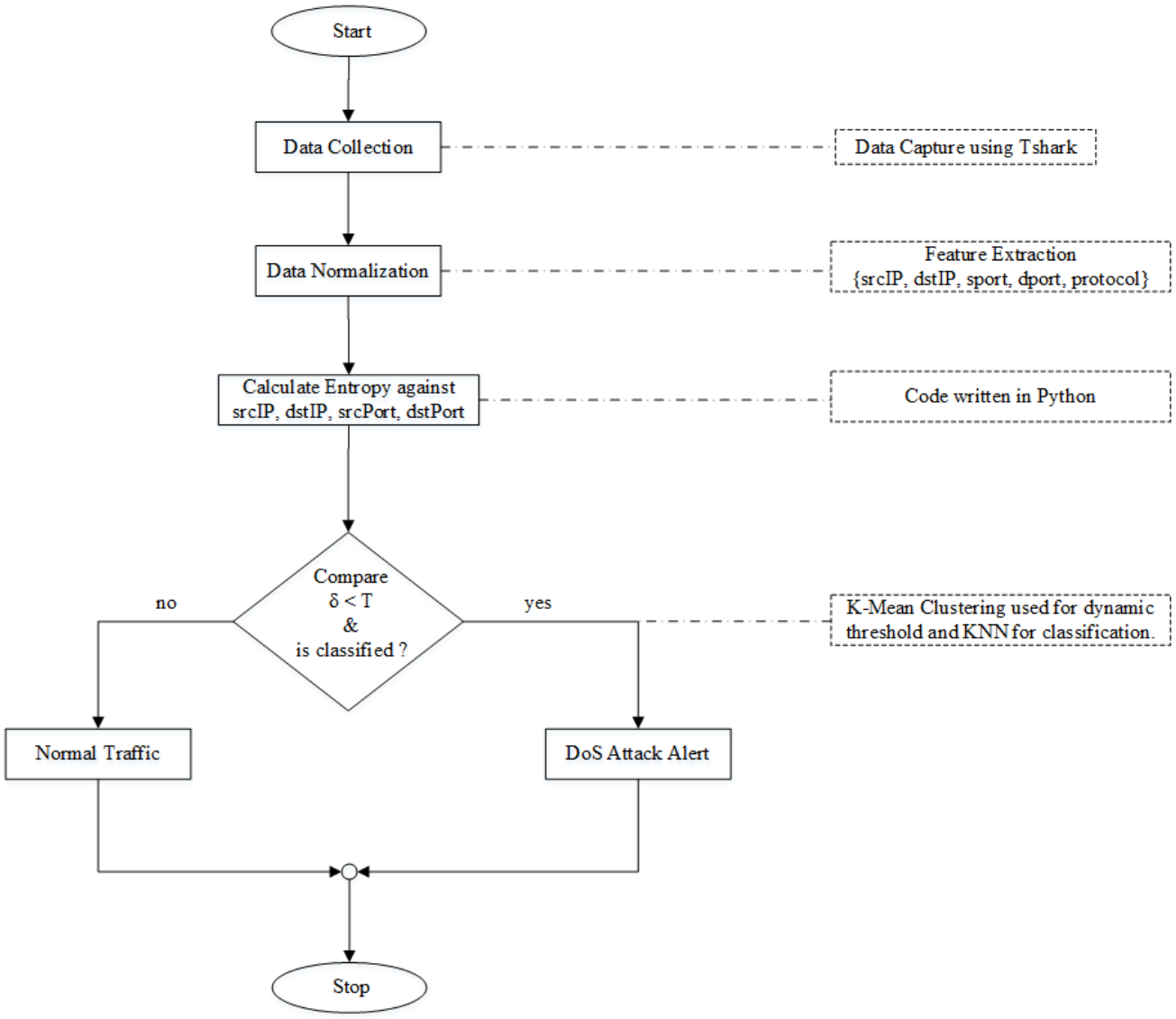

Figure 2: Work Flowchart of Proposed Model

In figure 2, the symbols δ and T denotes profiling threshold history of network traffic and dynamic threshold calculated using clustering respectively. In this system, threshold history profiles and classification done using KNN with known attacks or signatures play a key role in determination of the attacks. In implementation of KNN, the dataset is split in a standard ratio of 70:30 for training and test data and the best value of k is determined as shown in figure 47 which is k=10, used throughout the work. Hence, all the newly incoming unlabeled test dataset is classified or predicted against trained dataset.

3.1 Data Collection

For any research, data collection is the preliminary steps. Here, data is generated in real time as in the form of pcap file format. For this, T-shark command in Linux is used for all the network traffic capture in pcap file format.

3.2 Data Normalization

Thus captured traffic from data collection is raw and not formatted for further processing. Therefore, normalization is done for proper formatted data from the unformatted data by processing feature extraction so that only useful information is extracted.

3.3 Entropy Calculation

Entropy Calculation is done with features parameter like source IP, source port, destination IP and destination port in against time for each parameters. With this, it is clearly visualized whether there is any misbehavior happening in the network traffic or not.

3.4 Threshold Comparison

Determining threshold plays a crucial role in detecting DDoS attack. To minimizing false positive, K-mean clustering algorithm is used rather than just averaging the total values obtained by the calculation thus normalized data entropy with respect to feature extraction. In this process, three random centroids are initially introduced. With this algorithm, the centroids are so adjusted until they get conversed which includes all scatter points within defined number of centroids. Thus obtained final required value of centroids and the centroids got from similar fashion with entropy value of normal traffic are used to set the threshold value by which other network traffics get compared. The threshold value changes periodically and thus it is dynamical in nature.

3.5 Clustering Workflow

The following flowchart diagram shows how the K-means algorithm is carried out.

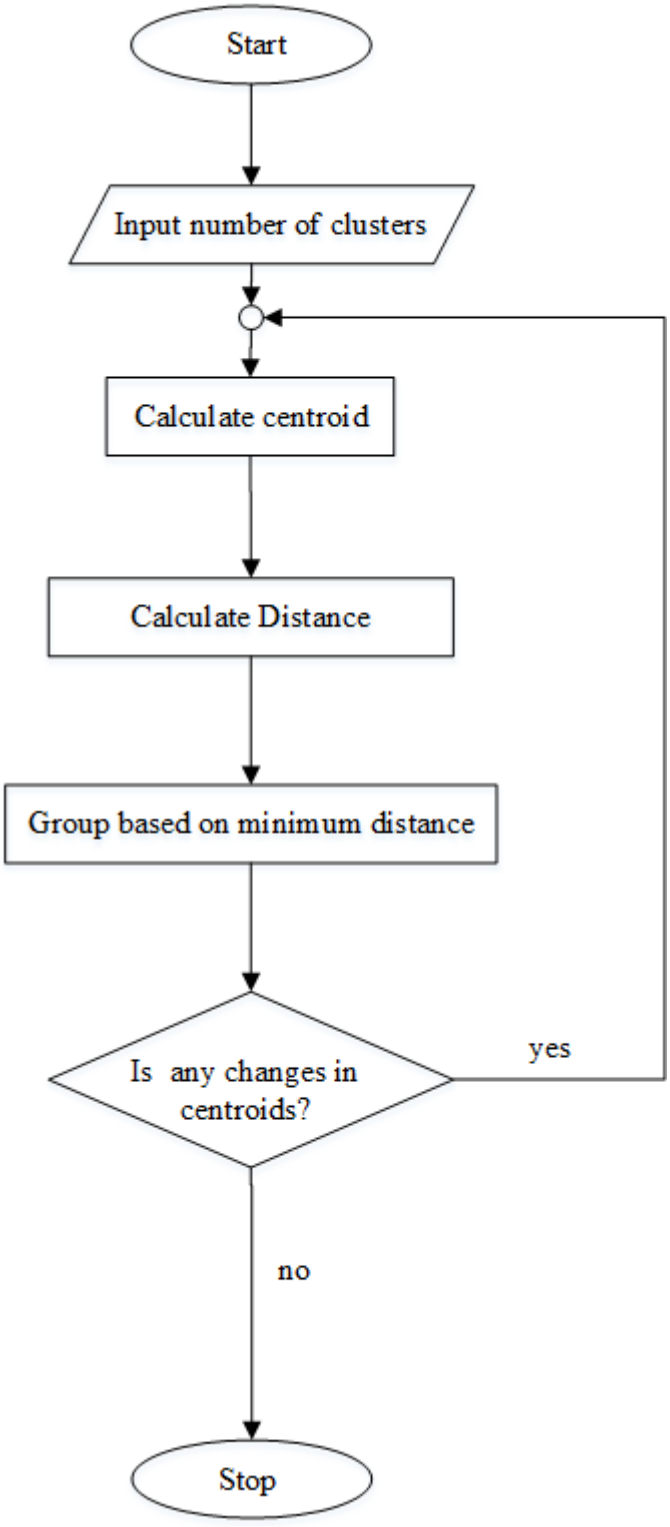

Figure 3: Flowchart about how clustering is done

The above figure 3: illustrates how the K-means clustering is carried out and aims to partition n observations into k clusters.

There are mainly 3 steps involve in the algorithm:

- Initialization – K initial "means" (centroids) are generated at random
- Assignment – K clusters are created by associating each observation with the nearest centroid
- Update – The centroid of the clusters becomes the new mean

Assignment and Update are repeated iteratively until convergence. The end result is that the sum of squared errors is minimized between points and their respective centroids.

3.6 K-Nearest Neighbor (KNN) Algorithm

KNN is a non-parametric supervised learning technique used to classify the data point to a given category with the help of training set. It can be simply defined as the technique embraced to captures information of all training cases and classifies new cases based on a similarity.

Predictions are made for a new instance by searching through the entire training set for the K most similar cases i.e neighbors and summarizing the output variable for those K cases. In classification this is the mode or the most common class value.

Algorithm

Step1: Start
Step2: Read the value of K
Step3: Load the dataset
Step4: Split the dataset into train and testing purpose
Step5: Euclidean distance calculation function
Step6: Prediction of classes for records with unknown labels
Step7: Stop

3.6 DDoS Attack Scenario

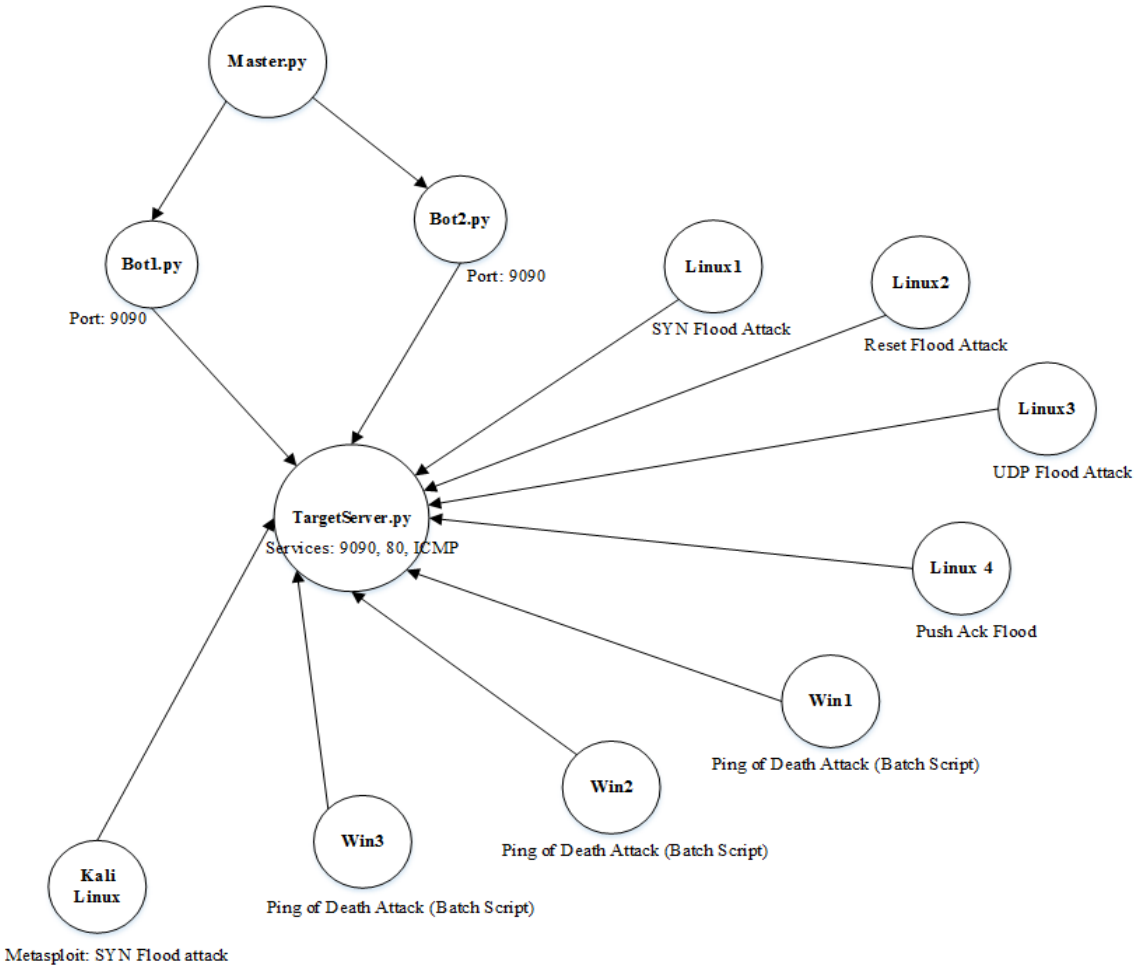

Figure 4: Attack Plan Diagram

There is the popular DDoS Attack Dataset known for The CAIDA "DDoS Attack 2007" Dataset which is now subject to the terms of the IMPACT Acceptable Use Agreement. Hence, the mail has sent for approval to get downloaded but they had rejected the request by saying your location is not DHS-approved location, therefore they were unable to extend a user account. Therefore, in the purpose of same pattern of standard dataset as in CAIDA generation in their respective standard format pcap file, it is decided to have dataset in real time.

To collect real time DDoS attack traffic, the above setup is done where there are lots of botnet which is automated as well as manually run at sync time. In automatic bot, there is the master which controls other bots i.e. bot1 and bot2 in this case which created a socket on specified port and starts listening for the master command, when and how long to attack the targeted server. There are other manually run bots like linux1, linux2, linux3, win1, win2, win3 and kali Linux.

Windows machines are used for ping of Death Attack which is executed with batch script, whereas Linux machines are used for different type of Flood attack like SYN Flood, UDP flood, Push ACK Flood, Reset Flood. Kali Linux is used to SYN Flood attack with Metasploit using DoS attack module.

3.7 Attacks Committed

The following types of DoS attacks are carried out during the research as follows:

- **Ping of Death**

  Ping of Death is one of DoS attack in which an attacker tries to crash or freeze the targeted server or computer by sending continuous oversized packets using ping command.

  This type of attack is commonly known as a Ping Flood attack.

- **SYN Attack**

  SYN Attack is the type of DDoS attack where the attacker sends SYN Flood which exploits the normal TCP three – way handshake, making half open connection. This attacks consume almost whole the resources on targeted server which ultimately render it unresponsive for further request.

  The main mechanism under this attack is the attacker or the malicious client either does not send ACK signal or if the IP is spoofed never receives the SYN-ACK signal. Thus the server under this type of attack will wait for acknowledgement of its SYN-ACK packet which make connection opened.

- **UDP Flood**

  UDP Flood is another type of DoS attack in which the attacker sends a larger number of UDP packets to the targeted server making overwhelmed so that the server gets exhausted. Therefore, it is unresponsive to other legitimate clients.

- **SYN FIN Flood**

  Generally, in TCP-SYN session, it is required to exchange of RST or FIN packets between the requester and the host i.e. server. During the FIN Flood, the targeted server gets overwhelmed with fake RST or FIN packet that have no any connection

to any TCP-SYN session for service in server. In this process, the victim server has to compensate with lots of system resources wasted which ultimately its services to other client gets partially unavailable.

- **Push ACK Flood**

  When the request gets connected with a server, the client can ask for ACK flag for confirmation or it can forcibly make server to process the information in the packet by setting the PUSH flag. Thus, the victim server gets attacked by ACK flood with fake ACK packets that do not belong to any of the sessions in server. Hence, valid traffic is prevented from getting response from server. This technique is called a PUSH or ACK Flood.

- **Reset Flood**

  Reset Flood is also known as forged TCP resets. In this type of attack, it tempers and terminated the internet connection by sending forged TCP reset packet.

3.8 Performance Parameters:

Following parameters will be calculated while training and testing of MLP.

- **True Positive (TP):** Situation in which a predefined rule or signature is matched then it acknowledges as an attack and an alarm is generated.

- **False Positive (FP):** Situation in which the normal traffic identifies as threat i.e. signature mismatched.

- **True Negative (TN):** Situation in which the normal traffic does not cause the signature to raise a detection alarm.

- **False Negative (FN):** Situation in which a signature is not fired even an attack is detected.

- **Attack Detection Rate (ADR)**: The detection rate is defined as the number of intrusion instances detected by the system (True Positive) divided by the total number of intrusion instances present in the test set.

  ADR **=** **(**Total detected attacks / Total attacks) * 100 % ----------(3)

- **Recall Rate:** Recall rate measures the proportion of actual positives which are correctly identified.

  Recall Rate = TP/ (TP + FN) ----------(4)

- **Precision Rate:** Precision rate is the ratio of true positives to combined true and false positives.

  Precision Rate = TP/ (TP + FP) ----------(5)

3.9 Tools:

Python:

Python is a general-purpose programming language that is becoming more and more popular for doing data science. Most of companies worldwide are using Python to harvest insights from their data and get a competitive edge. In this thesis, python is being used as based coding platform for doing clustering, classification and rendered different graphs.

Vim text editor:

Vim is a highly configurable text editor for efficiently creating and changing any kind of text. It is included as "vi" with most of linux and UNIX like systems. Since the entire thesis is done on linux platform, therefore vim text editor is used throughout the project.

Wireshark:

Wireshark is an open source tool for profiling network traffic and analyzing packet so that this tool is more often used as a network analyzer and packet sniffing and capturing using wireshark terminal command **T-shark** in this thesis.

Snort:

Snort is an open-source security software product that looks at network traffic in real time and logs packets to perform detailed analysis used to facilitate security and authentication efforts. In this thesis, it is used as to validate whether the proposed detecting model is effective enough to generate alarm to indicate any DDoS attacks in the system. There are two flavors of snort available, host-based and network-based. But in this thesis, network-based IDS has embraced as it is suitable standard validation tool to valid the result obtained from the research

Kali Linux:

Kali Linux is a Debian-based Linux distribution aimed at advanced Penetration Testing and Security Auditing. Kali contains several hundred tools which are geared towards various information security tasks, such as Penetration Testing, Security research, Computer Forensics and Reverse Engineering. Kali Linux is developed, funded and maintained by Offensive Security, a leading information security training company.

In this thesis, it is used to do different types of attacks like Ping Flood, SNY Flood, SYN FIN etc as during attacking simulation environment setup.

CHAPTER 4: RESULT, ANALYSIS AND COMPARISON

4.1 Results and Analysis

4.1.1 Data Collection (Network Traffics Capturing)

```
13:37:43.389836 IP 192.168.200.8.9090 > 192.168.200.153.36146: Flags [R], seq 892221000, win 0, length 0
13:37:43.389839 IP 192.168.200.152.36136 > 192.168.200.8.9090: Flags [.], seq 6204343:6205791, ack 2, win 229, options [nop,nop,TS val
777464 ecr 2644842247], length 1448
13:37:43.389848 IP 192.168.200.8.9090 > 192.168.200.152.36136: Flags [R], seq 3452498373, win 0, length 0
13:37:43.390500 IP 192.168.200.152.36136 > 192.168.200.8.9090: Flags [.], seq 6205791:6207239, ack 2, win 229, options [nop,nop,TS val
777465 ecr 2644842247], length 1448
13:37:43.390540 IP 192.168.200.8.9090 > 192.168.200.152.36136: Flags [R], seq 3452498373, win 0, length 0
13:37:43.390623 IP 192.168.200.152.36136 > 192.168.200.8.9090: Flags [.], seq 6207239:6208687, ack 2, win 229, options [nop,nop,TS val
777465 ecr 2644842247], length 1448
13:37:43.390638 IP 192.168.200.8.9090 > 192.168.200.152.36136: Flags [R], seq 3452498373, win 0, length 0
13:37:43.390746 IP 192.168.200.153.36146 > 192.168.200.8.9090: Flags [.], seq 6124145:6125593, ack 2, win 229, options [nop,nop,TS val
830232 ecr 1219614811], length 1448
```

Figure 5: Raw Network Traffic

The above figure 5 shows the unformatted pattern of raw data collected through tshark command which is the data feed for the thesis.

Traffic	Captured	Displayed	Displayed %	Marked	Marked %
Packets	5829051	5829051	100.000%	0	0.000%
Between first and last packet	407.571 sec				
Avg. packets/sec	14301.919				
Avg. packet size	445 bytes				
Bytes	2592434140	2592434140	100.000%	0	0.000%
Avg. bytes/sec	6360689.396				
Avg. MBit/sec	50.886				

Figure 6: Summary of Captured Traffic

4.1.2 Data Normalization

frame.number	frame.time	ip.src	ip.dst	tcp.srcport	tcp.dstport	frame.protocols	frame.len	ip.proto
1	Jul 24, 2018 13:37:07.111205814 +0545	192.168.200.8	8.39.54.93	54072	443	eth:ethertype:ip:tcp:ssl	119	6
2	Jul 24, 2018 13:37:07.422598098 +0545	8.39.54.93	192.168.200.8	443	54072	eth:ethertype:ip:tcp:ssl	119	6
3	Jul 24, 2018 13:37:07.422690467 +0545	192.168.200.8	8.39.54.93	54072	443	eth:ethertype:ip:tcp	66	6
4	Jul 24, 2018 13:37:09.548894229 +0545	192.168.200.10	255.255.255.255			eth:ethertype:ip:udp:data	82	17
5	Jul 24, 2018 13:37:12.451018374 +0545	192.168.200.1	239.255.255.250			eth:ethertype:ip:udp:ssdp	313	17
6	Jul 24, 2018 13:37:12.451629312 +0545	192.168.200.8	192.168.200.10			eth:ethertype:ip:data	1514	1
7	Jul 24, 2018 13:37:12.452252731 +0545	192.168.200.151	192.168.200.8	43964	80	eth:ethertype:ip:tcp	60	6
8	Jul 24, 2018 13:37:12.452906714 +0545	192.168.200.151	192.168.200.8	43965	80	eth:ethertype:ip:tcp	60	6
9	Jul 24, 2018 13:37:12.453500666 +0545	192.168.200.151	192.168.200.8	43966	80	eth:ethertype:ip:tcp	60	6
10	Jul 24, 2018 13:37:12.454131376 +0545	192.168.200.151	192.168.200.8	43967	80	eth:ethertype:ip:tcp	60	6
11	Jul 24, 2018 13:37:12.454776434 +0545	192.168.200.151	192.168.200.8	43968	80	eth:ethertype:ip:tcp	60	6
12	Jul 24, 2018 13:37:12.455597025 +0545	192.168.200.151	192.168.200.8	43969	80	eth:ethertype:ip:tcp	60	6
13	Jul 24, 2018 13:37:12.456247816 +0545	192.168.200.151	192.168.200.8	43970	80	eth:ethertype:ip:tcp	60	6
14	Jul 24, 2018 13:37:12.456852156 +0545	192.168.200.151	192.168.200.8	43971	80	eth:ethertype:ip:tcp	60	6
15	Jul 24, 2018 13:37:12.457447045 +0545	192.168.200.151	192.168.200.8	43972	80	eth:ethertype:ip:tcp	60	6
16	Jul 24, 2018 13:37:12.458034357 +0545	192.168.200.151	192.168.200.8	43973	80	eth:ethertype:ip:tcp	60	6

Figure 7: Normalized Network Traffic

The "frame number" field is the packet index for captured network traffic, whereas "frame. time" field holds the time that the packet appeared since from the first packet of captured traffic. The "ip.src" and "ip.dst" fields contain the source and destination IP values, respectively. Similarly, "tcp.srcport" and "tcp.dstport" fields contain the source port and destination port values, respectively. "The "protocol" field holds the protocol name of the network packet. The "frame.len" field holds the packet size in bytes. Last, "ip.proto" field contain the protocol type by numeric values.

4.1.3 Entropy Calculation

4.1.3.1 Entropy Against Source IP

During the DDoS attack, there will be so many packets with different source IP addresses because of traffics comes with different sources. Therefore, it is one of the important parameter to be calculated for detection process.

The following results show the difference between normal traffic entropy and attacked one below:

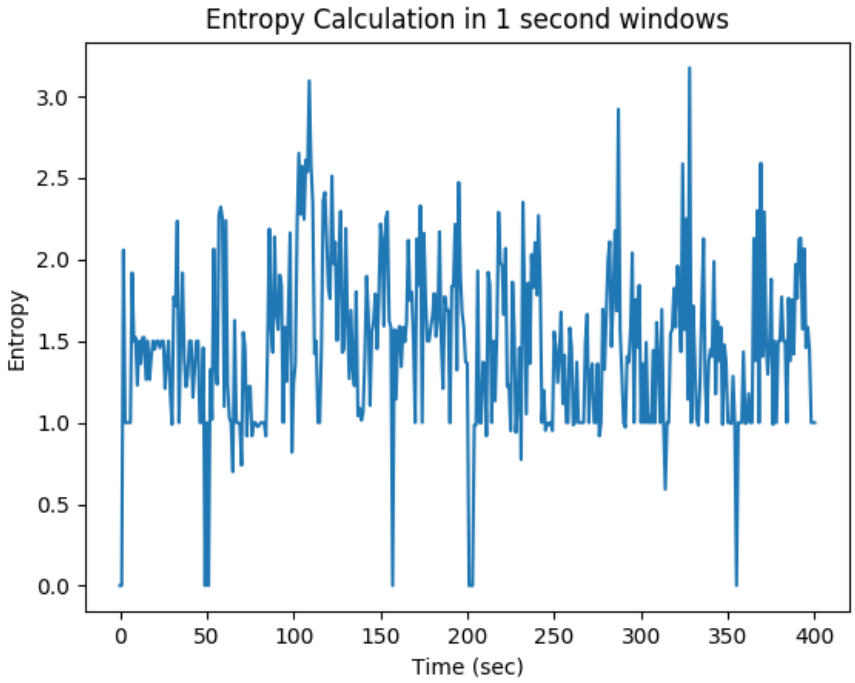

Figure 8: Normal traffic entropy behavior in 1 sec window

It is clearly seen from the figure 8 above that during normal traffic there is no vast changes in entropy, the variation is almost steady.

As per below figures 9-11, there is contrast with normal traffic entropy noted large variation in entropy. For the clarity in visualization, the complete scenario is taken 400 seconds with 1, 5, 10 and 15 seconds windows time. It is seen that attacks committed 115-270 seconds since there is significant variation in entropy values.

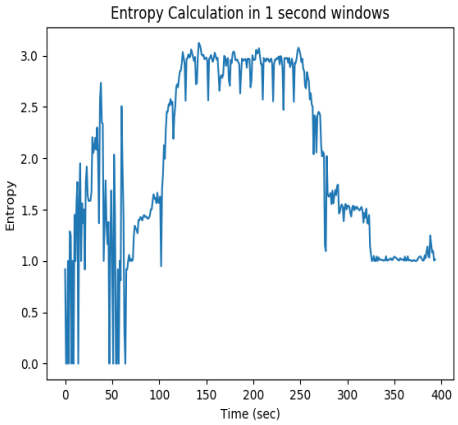

Figure 9: Entropy in 1 sec window

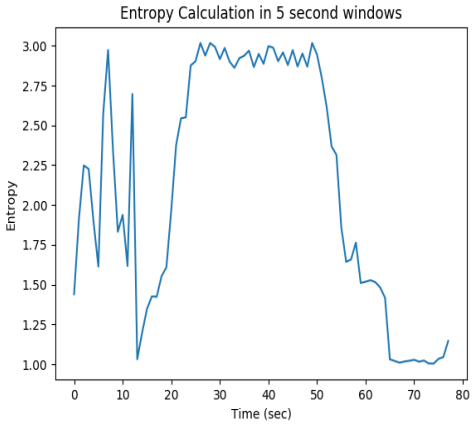

Figure 10: Entropy in 5 sec window

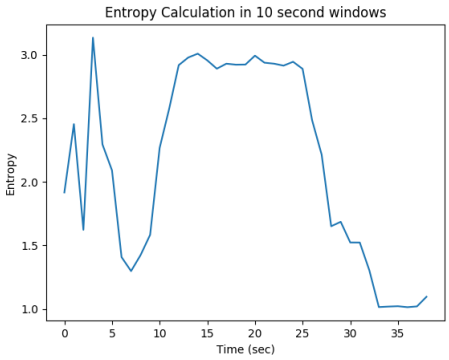

Figure 11: Entropy in 10 sec window

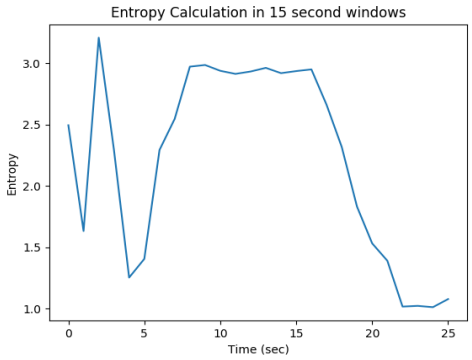

Figure 12: Entropy in 15 sec window

4.1.3.2 Entropy Against Destination IP

During attack, there will be so many packets concentrate towards same destination IP addresses. In this case, unique destination IP addresses converges into a small value. Therefore, the uncertainty will increase during attack for entropy against destination IP addresses.

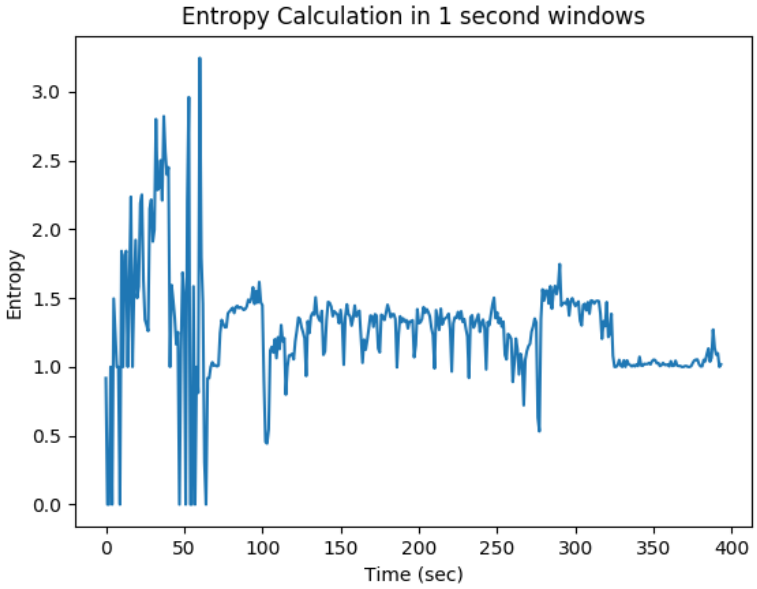

Figure 13: Entropy against destination IP per sec

4.1.3.3 Entropy Against Source Port

The measurement of entropy value of source ports is a good parameter to determine whether system is in attack or not. Normally, incoming traffic is distributed on some specific source ports. But somehow, there will be so many ports number tends to be used. Therefore, it's the certainty so entropy will increase.

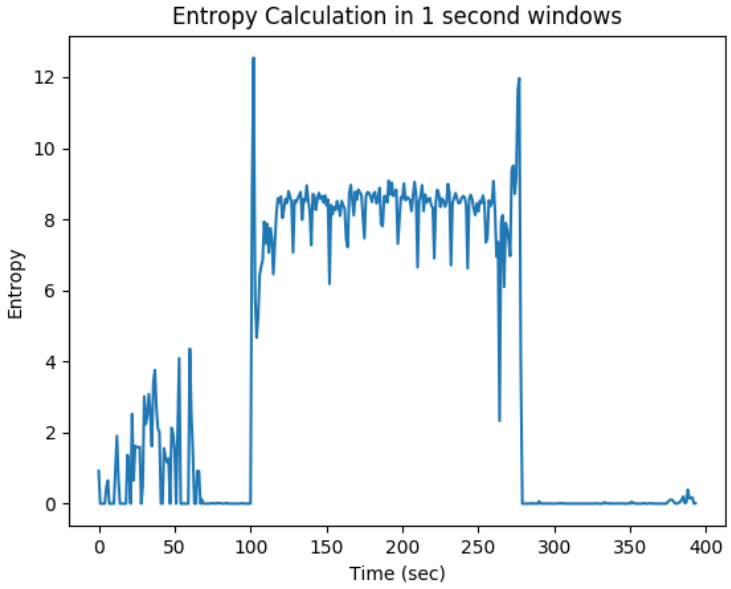

Figure 14: Entropy against Source Port per sec

4.1.3.14. Entropy Against Destination Port

The Server on normal condition, it servers only in a few ports. If the certain port is not opened, it replies with "unreachable port" warning packet. However, during attacks the attacker send too many packets in too many different ports to make server busier so as other legitimate clients are prevented from reply. It is cleared that during attack the entropy continuously increased.

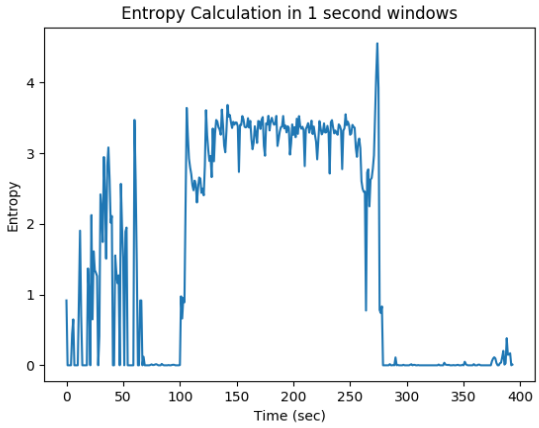

Figure 15: Entropy against Destination Port per sec

4.1.4 Packet Per Second

The number of packets flow within certain time calculations plays the significant role in detecting dos attacks. During an attack, the packets count significantly increased within the time frame. In this case, it is calculated with 1 sec time windows. With reference to figure 16, the packets count increased and maintained for 115-270 seconds. Hence, during this period it can be said system is on attack.

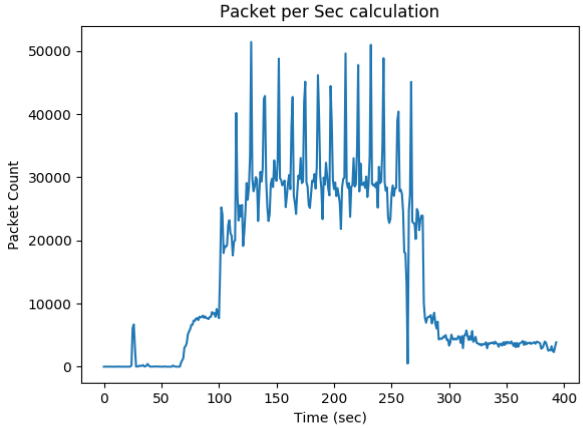

Figure 16: Packet Vs sec

4.1.5 Average Packet Per Second

The content of packets has usually very similar sizes during attack. Therefore, the average packet length gets converged to that value. From figure 17, it is clearly observed that during 115-270 seconds the average packet stay stabled. Therefore, it can be concluded that the system in attack.

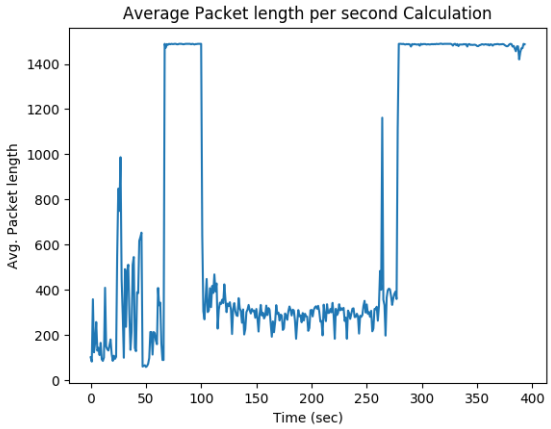

Figure 17: Average packet Vs sec

4.1.6 Bandwidth

The DDoS attack can be visualized with bandwidth monitoring since the bandwidth consumed drastically during the attack as compare than in normal traffic flow. It can be observed with respect to figure 18, that during attack period, bandwidth consumption remains constant for 115-270 seconds.

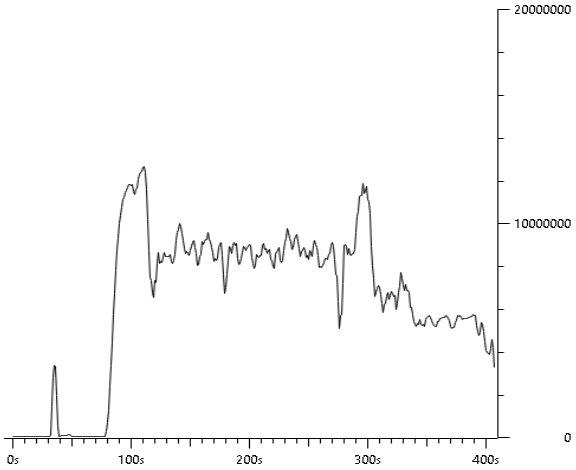

Figure 18: Bandwidth (Bytes/sec)

4.1.7 Threshold Determination

Determining threshold plays a crucial role in detecting DDoS attack. To minimizing false positive, K-mean clustering algorithm is used rather than just averaging the total values. In this process, three random centroids are initially introduced. With this algorithm, the centroids are so adjusted until they get conversed which includes all scatter points within defined number of centroids. Thus obtained final required value of centroids and the centroids got from similar fashion with entropy value of normal traffic are used to set the threshold value by which other network traffics get compared. The threshold value changes periodically and thus it is dynamical in nature.

The following outputs described difference in normal and attacked traffic clustering.

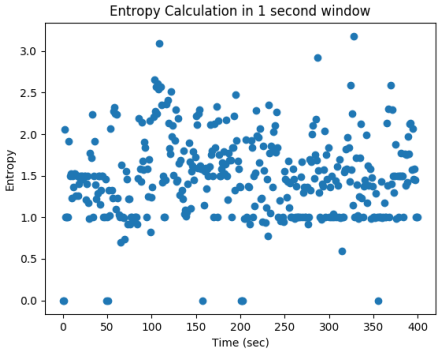

Figure 19: Normal Traffic Scatter plot

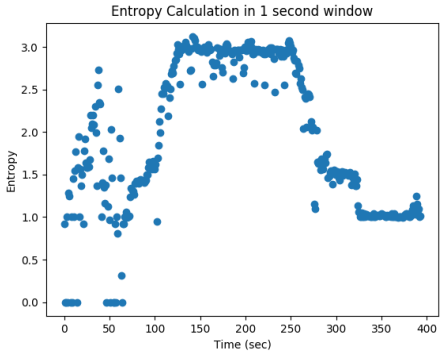

Figure 20: Attacked Traffic Scatter plot

The above figures 19-20 show the scattering graph of both normal traffic and the traffic under attack and visualize the difference on it.

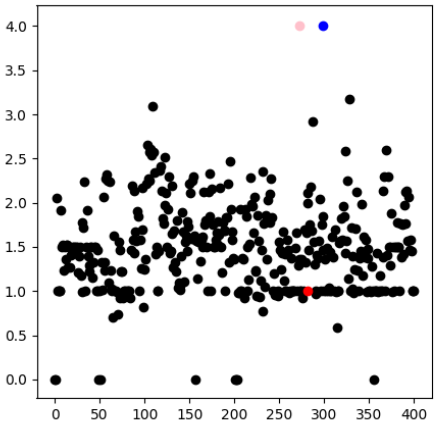

Figure 21: Normal Traffic Random Centre

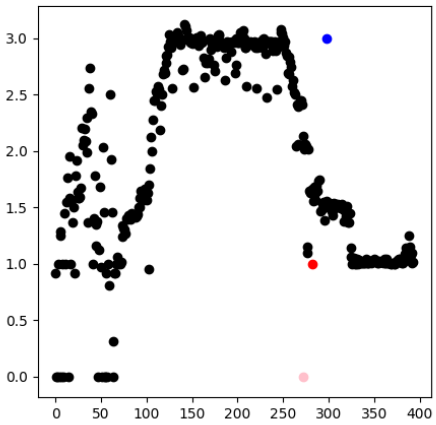

Figure 22: Attacked Traffic Random Centre

The above figures 21-22 describes about the random centre assignment as initialization of clustering process.

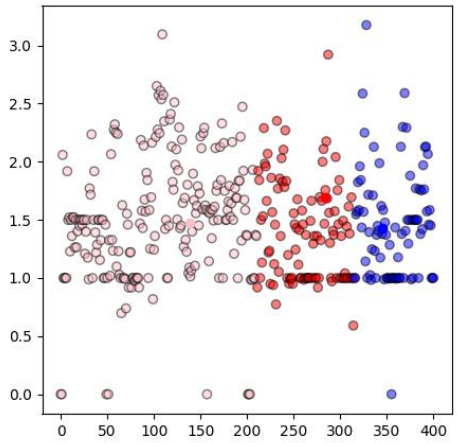

Figure 23: Normal Traffic Adjusting centre

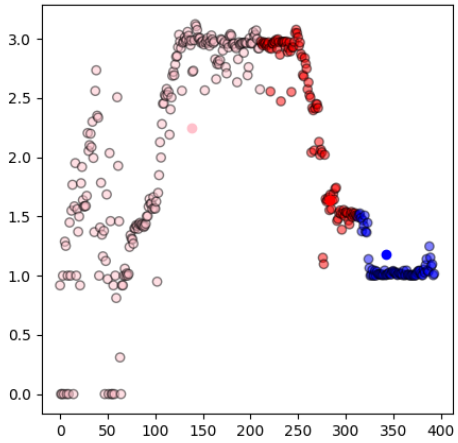

Figure 24: Attacked Traffic Adjusting centre

The figures 23-24 illustrate about cluster adjustment along with adaptive centre until it all the cluster get converged.

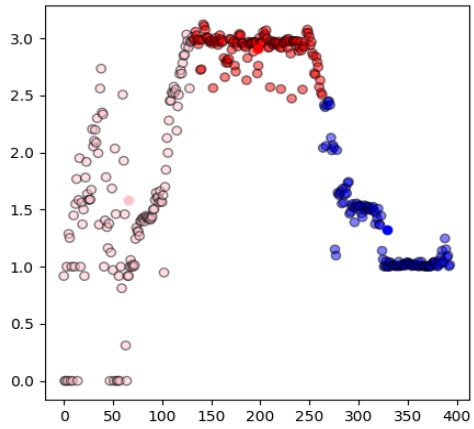

Figure 25: Final Clusters of Normal traffic

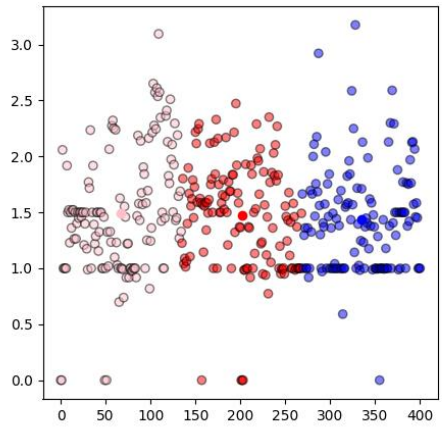

Figure 26: Final Clusters of Attacked one

The above figures 25-16 visualize the final clustered dataset of both normal traffic as well as traffic under attacked.

The following snapshots show the final converged centroid of normal and attacked traffic respectively.

```
root@pop-os:~/thesis/code/final# ./entropy_cluster.py
{1: [201.5, 1.469555412787095], 2: [67.0, 1.4895223477394606], 3: [334.5, 1.4385950550194218]}
```

Figure 27: Final Converged Centroids of normal traffic

```
root@pop-os:~/thesis/code/final# ./entropy_cluster.py
{1: [197.5, 2.912230959879262], 2: [65.5, 1.5857674294333115], 3: [328.5, 1.3240440633690194]}
```

Figure 28 Final Converged Centroids of attacked traffic

It is clearly seen from figure 27 that there is no vast difference in all of the 3 centroids with respect to entropy coordinate. This verifies there is no such significant changes in network traffic, hence can be said it is normal.

But it is not the case with figure 28 since there is vast difference in entropy coordinate in 1st array with 2nd and 3rd array. The proposed model is developed such a way that it averages all three centroids of entropy coordinate only of normal traffic with respect to figure 27 and declared as threshold value. Moreover, the lowest two values are also averaged and thus get value also play important in decision making. That is the decision making criteria is the threshold set by normal traffic and threshold determined by later averaged value. Now, all the traffic's entropy will be compared to that criteria, if the traffic is rated beyond the criteria with 15-20 continuous counts then it generates detection alert, otherwise it marked as normal traffic.

4.1.8 System Monitoring during Attacks

The following monitoring graphs illustrate what changes occurred in network during attacks.

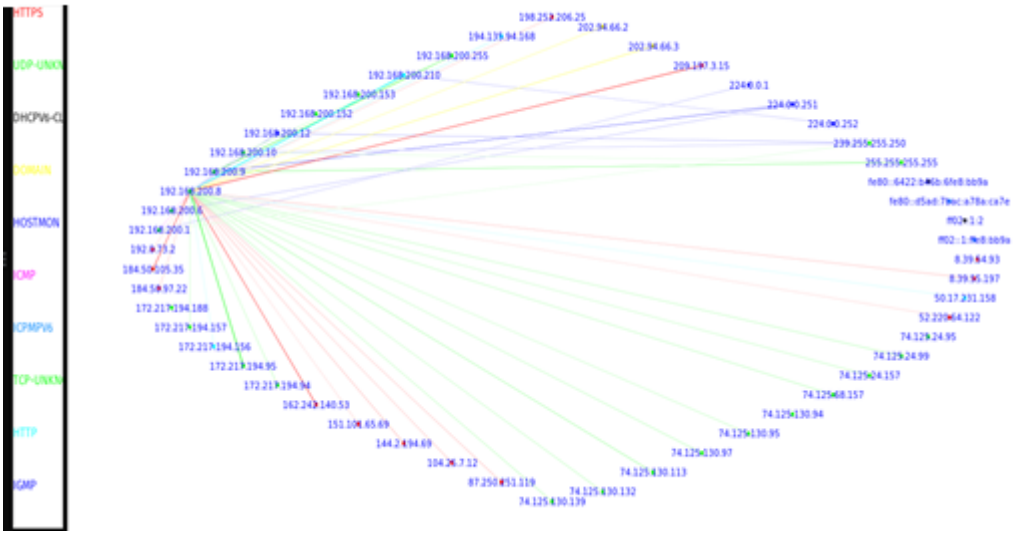

Figure 29: Network Wheel Graph

Fig 21. shows network activity graphically so that it is clearly visualized which hosts are actually talking to the server. It also gives quickest method to identify the system's roles

by different colored lines connecting them. Each color represents the different protocols and the thickness of the line represents the amount of traffic between the source and destination. With this graph, it is very quickest way to identify the top talkers on the network.

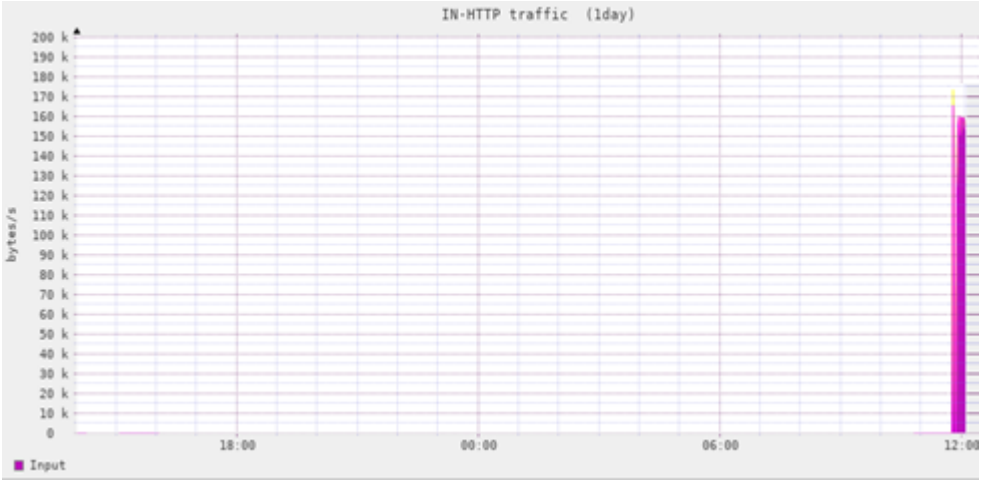

Figure 30: HTTP Traffic in targeted server

In the figure 30, the http traffic get suddenly overwhelmed at 12:00pm and get back into normal after few minutes. During that time, the traffic rises up exponentially which is definitely due to some abnormal causes in the network. Hence, it can be identified the system under attacks.

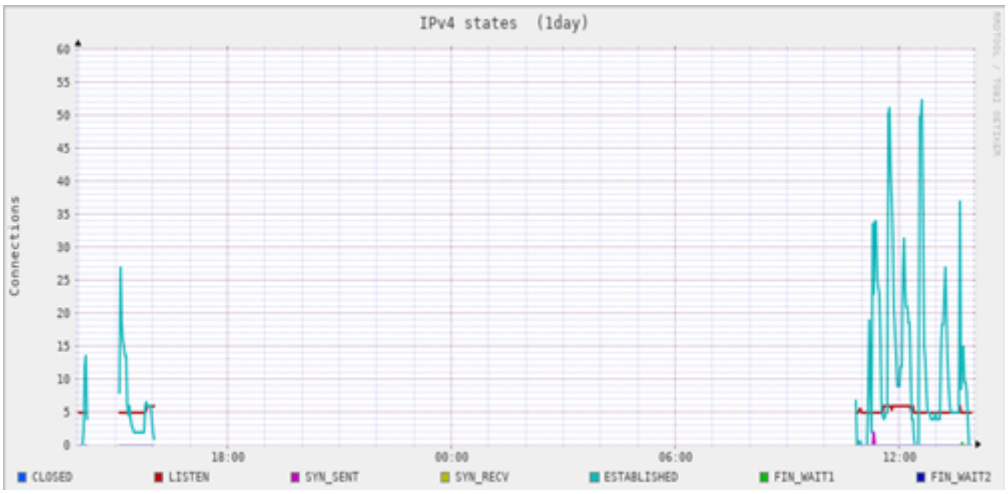

Figure 31: Connections (IPv4) in targeted server

The above figure 31 shows the total half open connections, syn_recv signal send by server but get no respond, FIN_WAITS signal in IPv4 TCP connections which ultimately made system resources consumed through the attacking period.

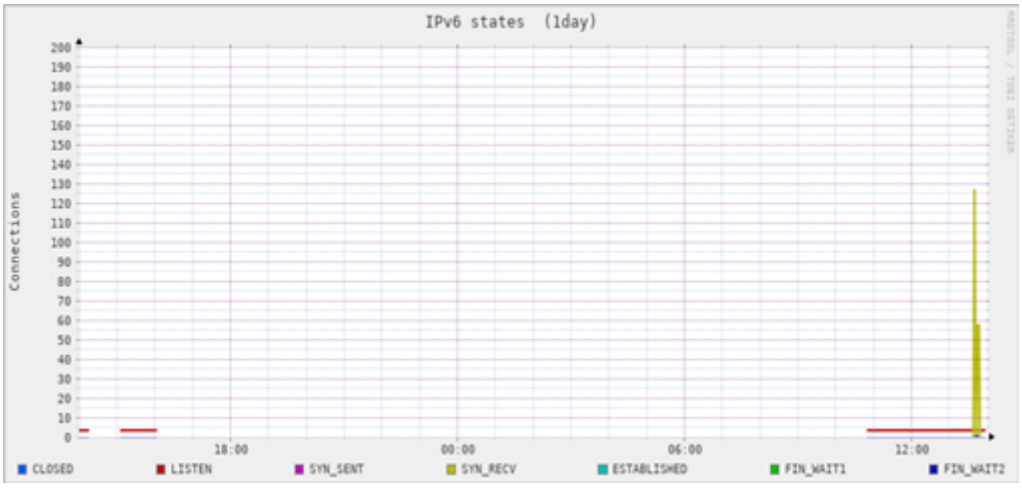

Figure 32: Connections (IPv6) in targeted server

The above figure 32 shows the total half open connections, syn_recv signal send by server but get no respond in establishing IPv6 TCP connections which ultimately made system resources consumed through the attacking period.

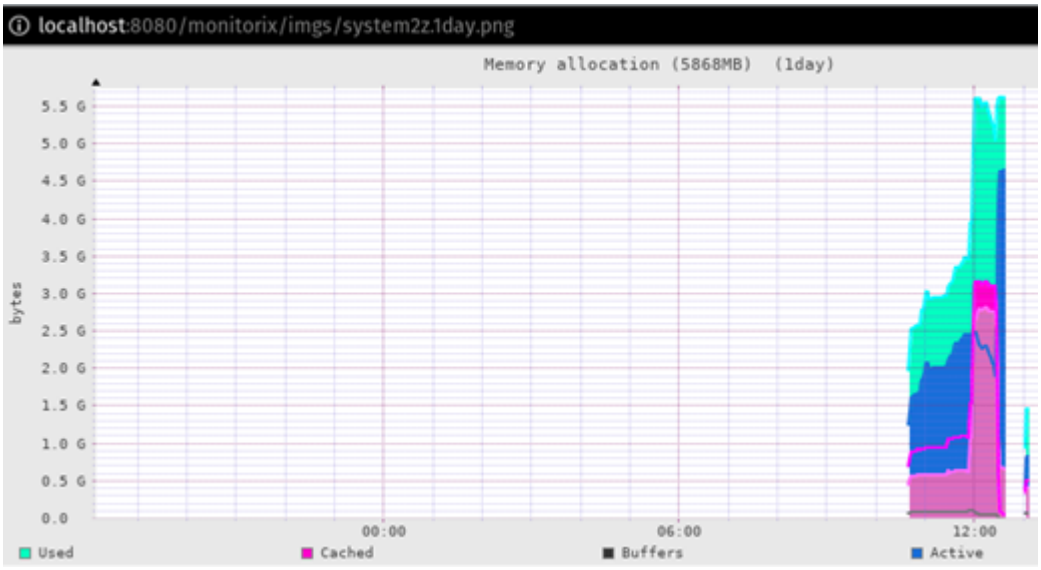

Figure 33: Resources used during attacks

The figure 33 above self explains how the memory resource consumption jump up suddenly higher than 5.5G during DDoS attack.

Figure 34: CPU used during attacks

The Fig 26. Shows how the CPU usage hikes up during DDoS attacks.

```
tcp        0      0 192.168.200.151:80      176.33.101.101:11452    SYN_RECV
tcp        0      0 192.168.200.151:80      176.33.101.101:12156    SYN_RECV
tcp        0      0 192.168.200.151:80      176.33.101.101:13139    SYN_RECV
tcp        0      0 192.168.200.151:80      176.33.101.101:13405    SYN_RECV
tcp        0      0 192.168.200.151:80      176.33.101.101:14117    SYN_RECV
tcp        0      0 192.168.200.151:80      176.33.101.101:14141    SYN_RECV
tcp        0      0 192.168.200.151:80      176.33.101.101:14163    SYN_RECV
tcp        0      0 192.168.200.151:80      176.33.101.101:14885    SYN_RECV
tcp6       0      0 192.168.200.151:80      192.168.200.5:48818     FIN_WAIT2
tcp6       0      0 192.168.200.151:80      192.168.200.5:48820     FIN_WAIT2
tcp6       0      0 192.168.200.151:80      192.168.200.5:48848     FIN_WAIT2
tcp6       0      0 192.168.200.151:80      192.168.200.5:48910     FIN_WAIT2
tcp6       0      0 192.168.200.151:80      192.168.200.5:48916     FIN_WAIT2
tcp6       0      0 192.168.200.151:80      192.168.200.5:48918     FIN_WAIT2
tcp6       0      0 192.168.200.151:80      192.168.200.5:48920     FIN_WAIT2
tcp6       0      0 192.168.200.151:80      192.168.200.5:50988     TIME_WAIT
tcp6       0      0 192.168.200.151:80      192.168.200.5:50990     TIME_WAIT
tcp6       0      0 192.168.200.151:80      192.168.200.5:51010     TIME_WAIT
tcp6       0      0 192.168.200.151:80      192.168.200.5:51012     TIME_WAIT
```

Figure 35: Netstat Results

With reference of figure 35 above, FIN_WAIT means the socket is closed, and the connection is shutting down whereas SYN_RECV is a connection request has been received from the network. And TIME_WAIT means the socket is waiting after close to handle packets still in the network.

The number of FIN_WAIT, SYN_RECV, TIME_WAIT connection should be pretty low, preferably less than 5. On Dos attack incidents, the number jumps to pretty high.

4.2 Result Validation between Normal Traffic and Attacked Traffic

The following graphs illustrate how the attacked traffic pattern get differed against normal traffic and hence can be identify the system is on attack at particular time.

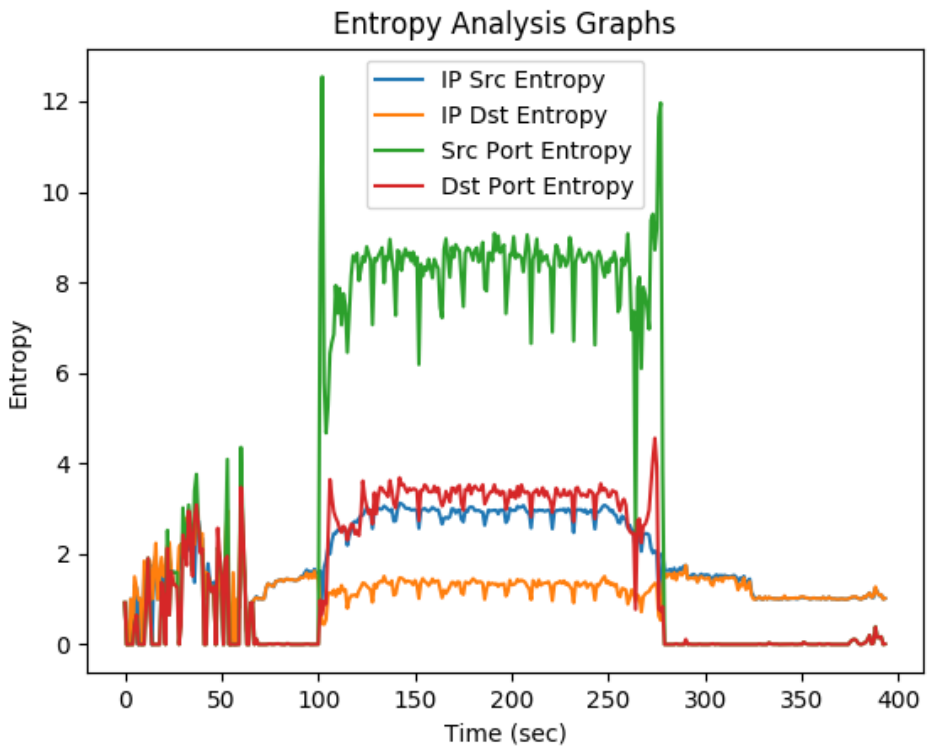

Figure 36: Entropy Analysis Graph of Attacked Traffic

It is seen as per above figure 36 that the entropies of different measuring parameter like IP Source entropy, IP Destination entropy, Source Port entropy and Destination Port entropy against targeted system, get converged at certain time slap of 116– 70 seconds irrespective to the entropy values of each entities. Hence, it can be concluded as the system is under attack at that time frame.

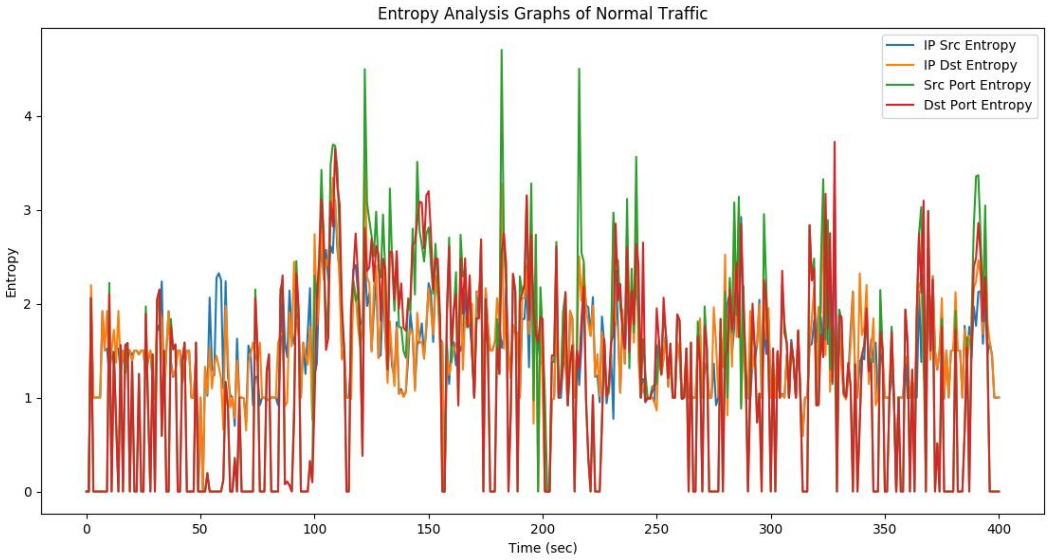

Figure 37: Entropy Analysis Graph of Normal Traffic

From the above figure 37, it is clearly seen that the entropies of different measuring parameter like IP Source entropy, IP Destination entropy, Source Port entropy and Destination Port entropy against targeted system, are intercepted throughout the given time frame with no anomalies behavior in the network.

As it is compared Figure 36 verses Figure 37, it is cleared that no anomalies behavior or any significant changes occurred throughout the given time slap in the network if it is taken in the case of Normal traffic but during the DDoS attacks there is the significant changes seen with all the measuring parameters and get conversed at attacking period irrespectively to the values of entropies of each entity.

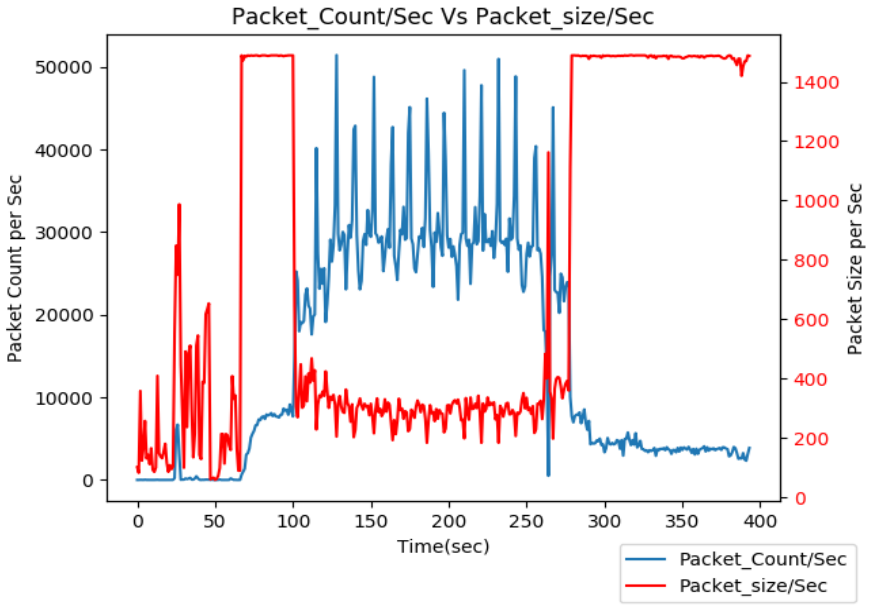

Figure 38: Packet count Vs Avg Packet size per sec graph of Attacked Traffic

The figure 38 above shows the analytical graph between the packet count per second and average packet size per second in a single graph. It pointed that there is same pattern of traffic flows with both the determined parameters in particular time period as per figure 38 irrespectible to their value, thus during that time period it can be considered as system is under attack.

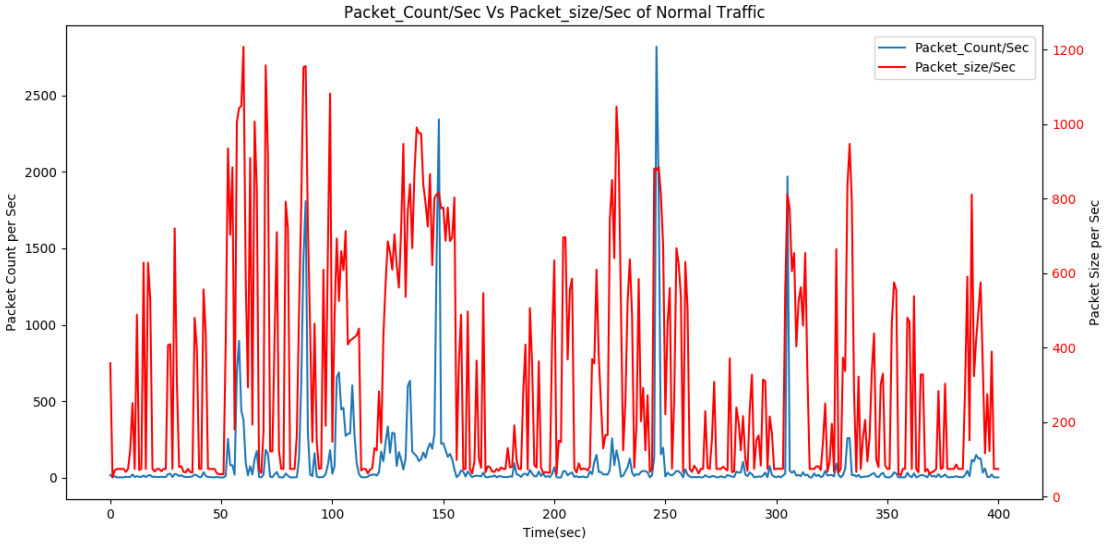

Figure 39: Packet count Vs Avg Packet size per sec graph of Normal Traffic

With reference to figure 39, it is seen that there are no significant changes noticed in pattern of traffic flows with both the determined parameters in the case of normal traffic throughout the given time period.

On comparison Figure 38 with Figure 39, it is clearly visualized that there is no anomalies behavior or any significant changes occurred throughout the given time slap in the network as in the case of Normal traffic but during the DDoS attacks there is the significant changes seen with all the measuring parameters and get conversed at attacking period irrespectively to their respective values.

4.3 Results and Evaluation against realtime DDoS

In order to evaluate the proposed model in real time world, it has been collected the recent DDoS attacked dataset encountered at one of the leading ISP, Vianet Communication Pvt. Ltd. The following are the results obtained through the proposed detections mechanism as:

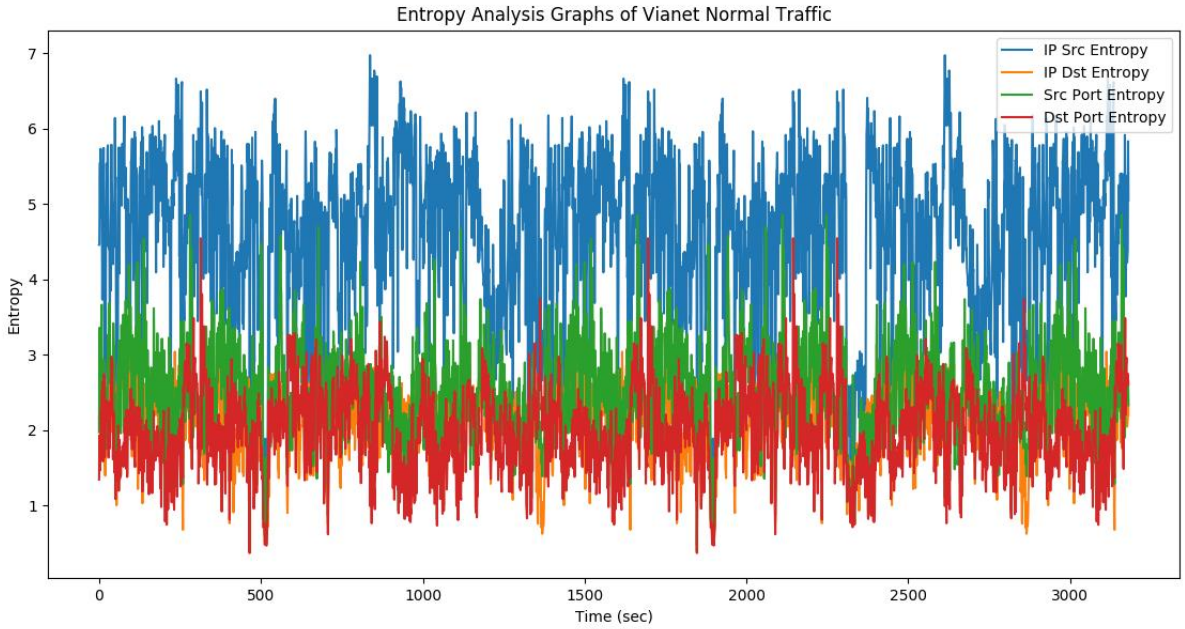

Figure 40: Entropies Analysis of normal traffic (Vianet) with 1 sec window

In figure 40, the entropies of all the feature attributes aligned with same pattern through the observed period. There is no anomalies behavior being encounter as per graph. Thus, it can be said the traffic is normal in nature.

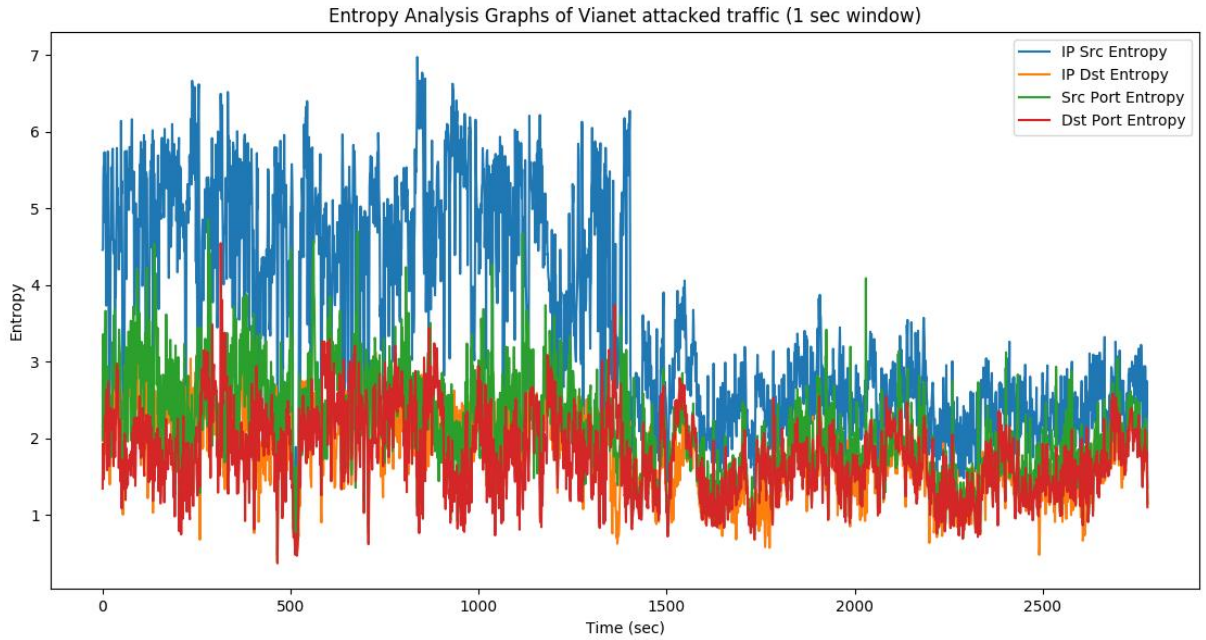

Figure 41: Entropies Analysis in attacked traffic (Vianet) with 1 sec window

It is known that when normal traffic flow, there will be high value in Source IP entropy as seen in above figure 41, but during attacked the values drop drastically from 5 to 2.5. Moreover, there is notably fluctuation encountered with other performance metric parameters at attacking period unlike the normal traffic analysis graph as in figure 40.

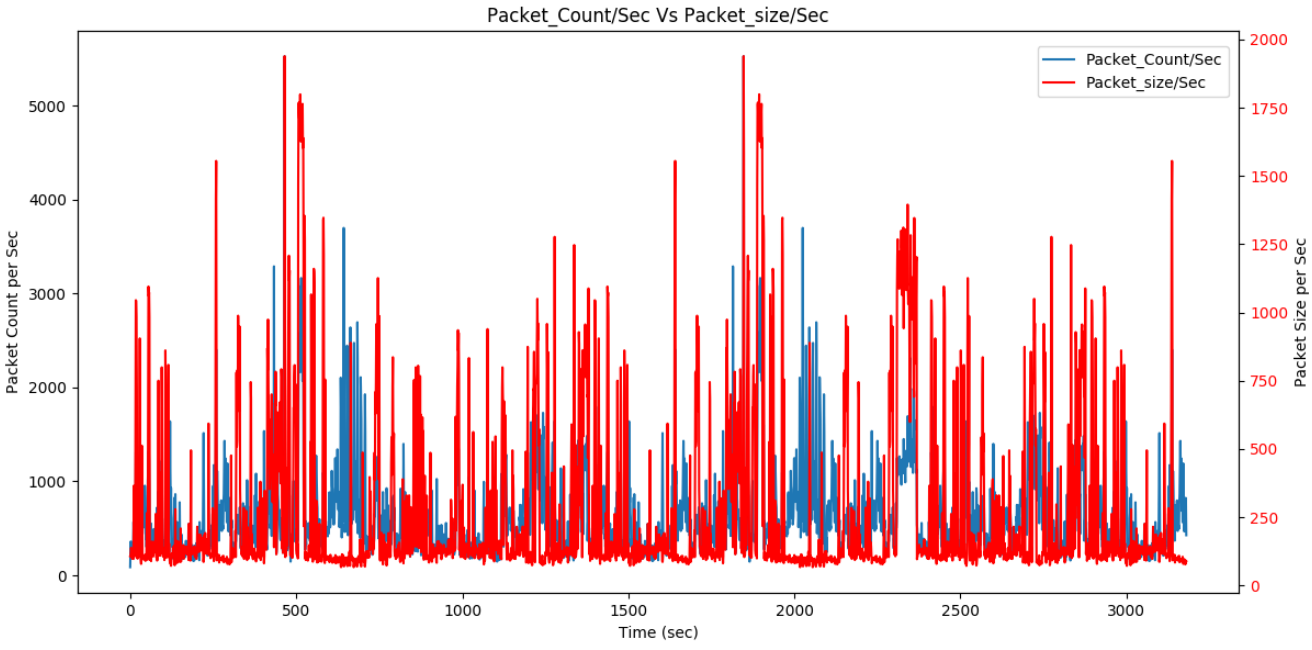

Figure 42: Packet count Vs Avg Packet size per sec in normal traffic (Vianet)

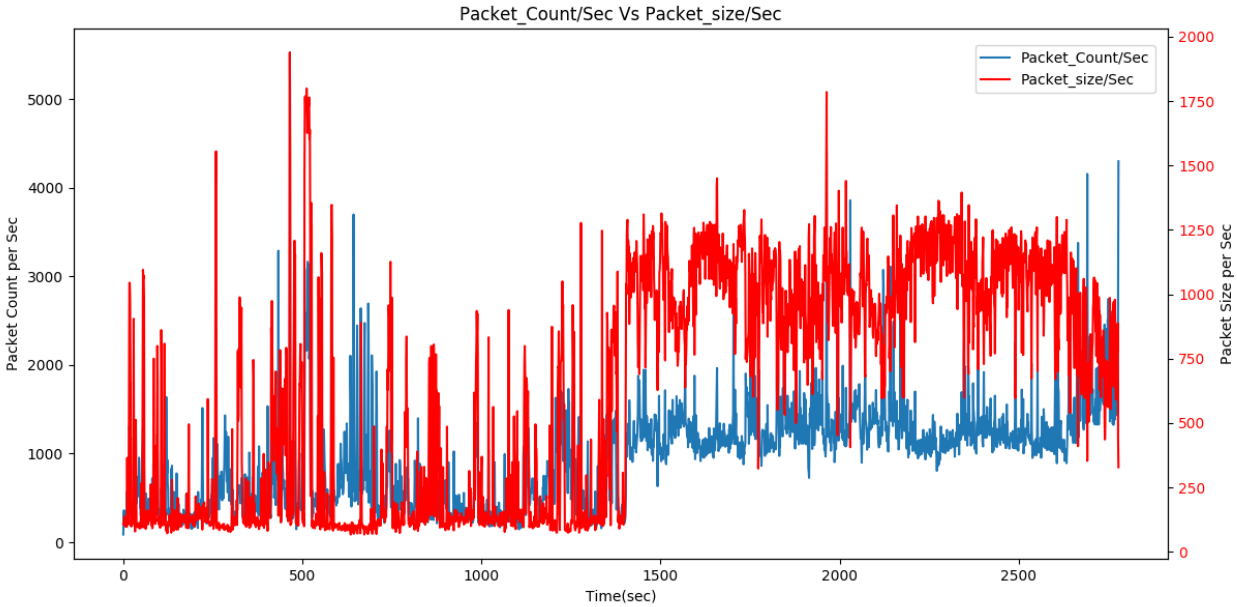

Figure 43: Packet count Vs Avg Packet size per sec in attacked traffic (Vianet)

When the network traffic is normal, there is constant stream of traffic but the significant changes is seen during some abnormal encountered in the system at the time of 1400sec, both in packet count and average packet size as compared figure 43 with figure 42.

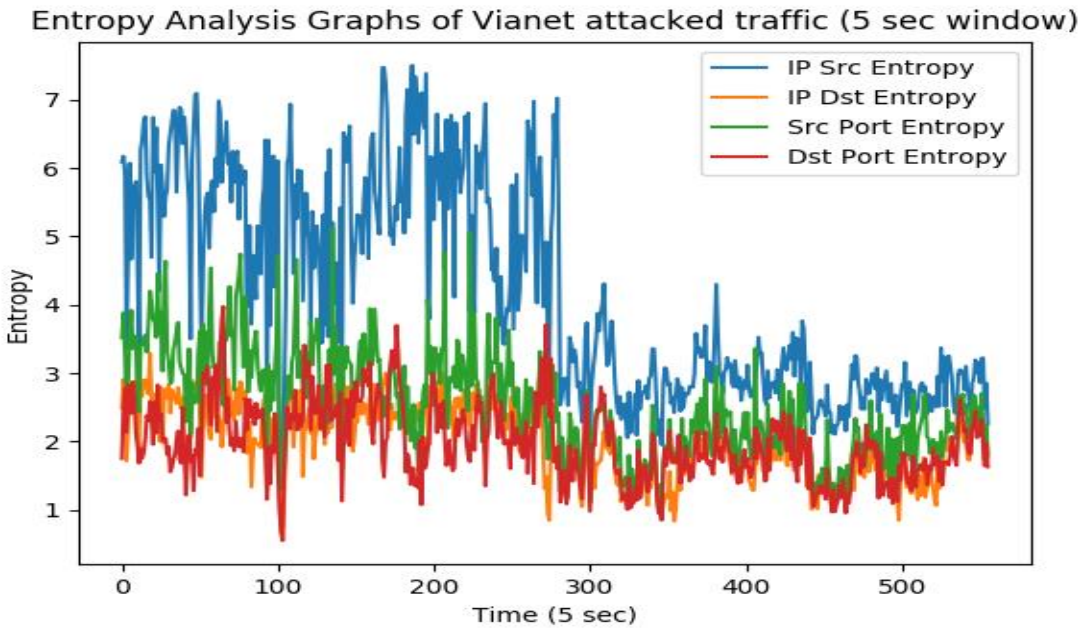

Figure 44: Entropies Analysis Graph in 5 sec windows

It is more cleared when the graph being plotted in 5 sec windows and it can be identified more precisely. From figure 44, it is observed during (5x280) 1400 sec, an attacker has hit the server with DDoS attacks.

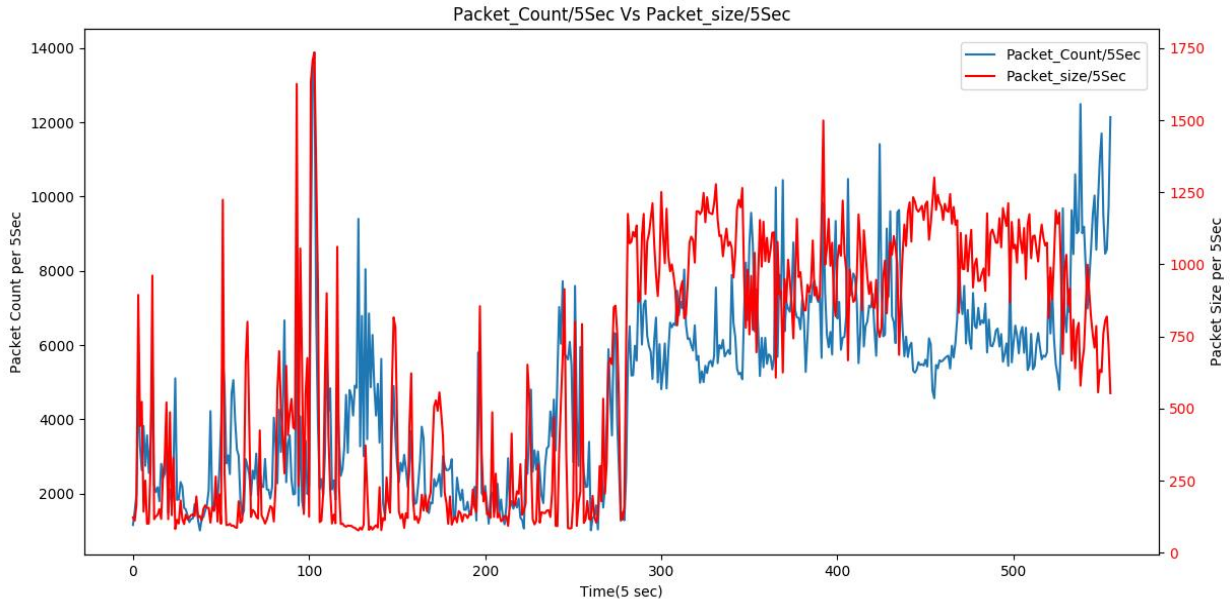

Figure 45: Packet count Vs Avg Packet size per sec graph in 5 sec windows

For more precise visualization, the graph between Packet count Vs Avg. Packet size per sec is taken in 5 sec of window time. Thus, it is clearly observable how and when the graph rises up which helps in identifying the behavior of traffic profiling.

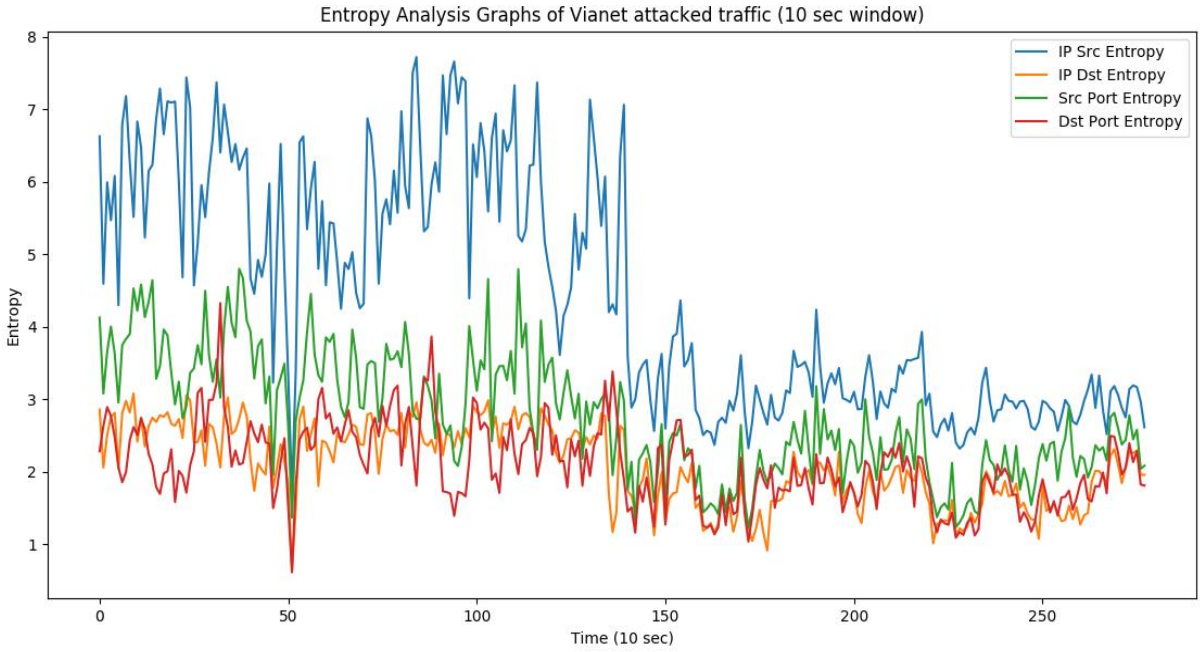

Figure 46: Entropies Analysis Graph in 10 sec windows

The figure 46 shows the entropies analysis graph taken in 10 sec of window time and provision with clear view about how the variation goes changes in 1400secs.

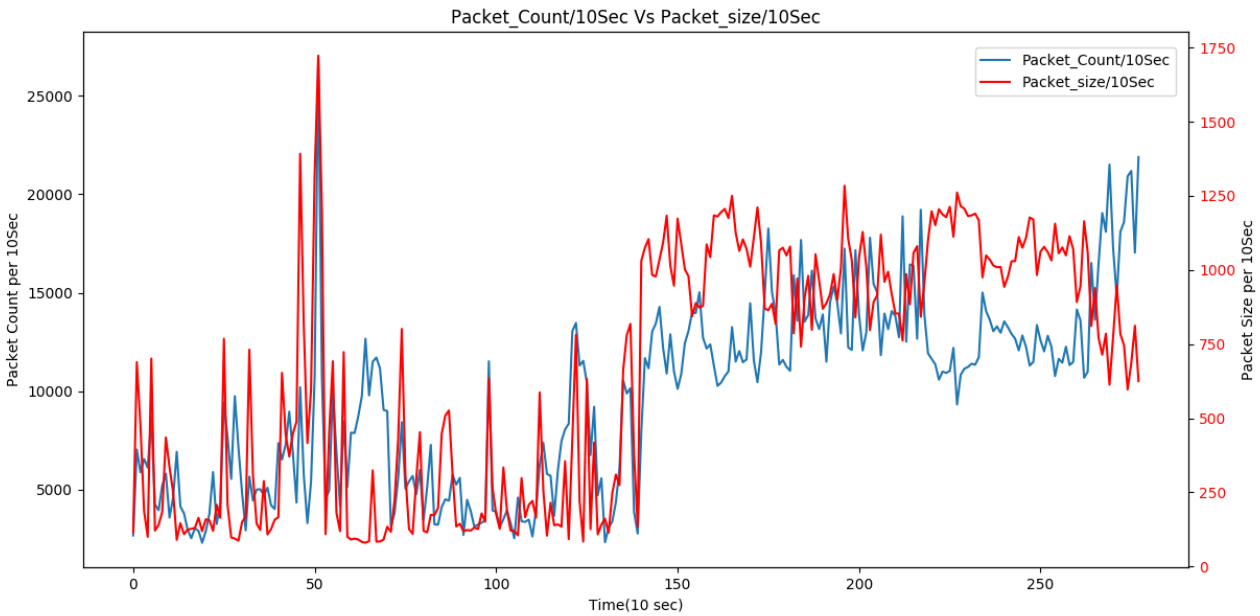

Figure 47: Packet count Vs Avg Packet size per sec graph in 10 sec windows

For more precise visualization, the figure 47 above is being plotted in 10 sec of window time and it is clearly portrayed that during 1400 sec period, there is significant rise in packet counts as well as average packet size. Hence, it can be said that during that instant, the system is on attacked.

4.5 Setting up benchmark for evaluation

There will be always need some benchmark or standard tested value against which the proposed model can be evaluated and validated. For this, snort is used as it is most accepted free and open source Network Intrusion Detection System and has ability to perform real-time traffic analysis and packet logging on Internet Protocol (IP) networks. Snort performs protocol analysis, content searching, and content matching by rules.

There are two types of rules, default rule and custom rule. Snort has set of rules to detect older type of attack so custom rules can also be written to detect existing attacks.

Some additional custom rules are written to detected more precisely as below:

- **Rule to detect TCP Flood**

  Activate tcp any any ->$HOME_NET !80 (flags:S; activates:1; msg:"SYN_FLOOD"; sid:1000002; threshold:type threshold, track by_src, count 10, seconds 1;)

  dynamic tcp any any -> $HOME_NET !80 (activated_by:1; sid:1000003; count: 1000;)

Description: When the packets with SYN flag set come through default port 80 and if the count is more than 20 per second, the alert is raised and the descendant of rule get fired and gives alert when the count exceeds 1000 per second.

- **Rule to detect ICMP Flood**

  activate icmp any any -> $HOME_NET any (activates:3; msg:"ICMP_FLOOD"; sid:1000004; threshold:type limit, track by_src, count 100, seconds 1;)

  dynamic icmp any any -> $HOME_NET any (activated_by:3; sid:1000005; count: 1000;)

  Description: If there are more than 100 ping requests per second then alert is fired otherwise the second rule generates an alert if the count exceeds 1000 per second.

- **Rule to detect UDP Flood**

  activate udp $EXTERNAL_NET any -> $HOME_NET any (activates:1; msg:"UDP_FLOOD"; sid:10000010; threshold:type threshold, track by_src, count 1000, seconds 1;)

  dynamic tcp $HOME_NET any -> $EXTERNAL_NET any (activated_by:1; flags: R+; sameip; sid:10000013; count: 1000;)

  Description: There is an alert if the packet exceeds 100 UDP packets per second and the descendant gives an alert when the victim sends the packets with the reset flag set.

- **Rule to detect HTTP Flood**

  alert tcp $EXTERNAL_NET any -> $HOME_NET 80 (msg:”GET HTML Request”; sid:1000022; threshold:type threshold, track by_src, count 1000, seconds 1;)

  Description: An alert generated when server receives 1000 HTTP request/sec.

- **Rule to detect Session Flood**

alert tcp any any -> any any (flow: established, to_server; msg: “ACK Flood”; sid:1000033; threshold:type threshold, track by_src, count 100, seconds 1;)

Description: This rules fired an alert if the client establishes 100 connections within one second.

Alert Generated by Snort

The following alert are generated while the DDoS rule, whether the default or custom rules get matched.

[**] [1:1000004:0] ICMP_FLOOD [**]
07/24-13:43:53.928968 192.168.200.5 -> 192.168.200.8
ICMP TTL:128 TOS:0x0 ID:959 IpLen:20 DgmLen:65528
Type:8 Code:0 ID:1 Seq:6598 ECHO

[**] [1:621:7] SCAN FIN [**]
[Classification: Attempted Information Leak] [Priority: 2]
07/24-13:41:04.772610 192.168.200.151:12611 -> 192.168.200.8:80
TCP TTL:64 TOS:0x0 ID:16226 IpLen:20 DgmLen:40
*******F Seq: 0x63138536 Ack: 0x16636A05 Win: 0x200 TcpLen: 20

[**] [1:624:7] SCAN SYN FIN [**]
[Classification: Attempted Information Leak] [Priority: 2]
07/24-13:41:04.760680 192.168.200.152:6444 -> 192.168.200.8:80
TCP TTL:64 TOS:0x0 ID:846 IpLen:20 DgmLen:40
******SF Seq: 0x65D5AF50 Ack: 0x2C0F1207 Win: 0x200 TcpLen: 20

[**] [1:1000033:0] “ACK Flood” [**]
07/24-13:37:41.194232 192.168.200.153:36146 -> 192.168.200.8:9090
TCP TTL:64 TOS:0x0 ID:52649 IpLen:20 DgmLen:781 DF
AP Seq: 0x1587D101 Ack: 0x352E3647 Win: 0xE5 TcpLen: 32
TCP Options (3) => NOP NOP TS: 828036 1219612615

[**] [1:1000004:0] ICMP_FLOOD [**]

[Priority: 0]

07/24-13:37:38.389392 192.168.200.1 -> 192.168.200.8

ICMP TTL:128 TOS:0x0 ID:0 IpLen:20 DgmLen:56

Type:11 Code:0 TTL EXCEEDED IN TRANSIT

192.168.200.8:60172 -> 198.252.206.25:5355

TCP TTL:0 TOS:0x0 ID:2690 IpLen:20 DgmLen:60 DF

Seq: 0xA5CDE4E7

Estimation of right value of K

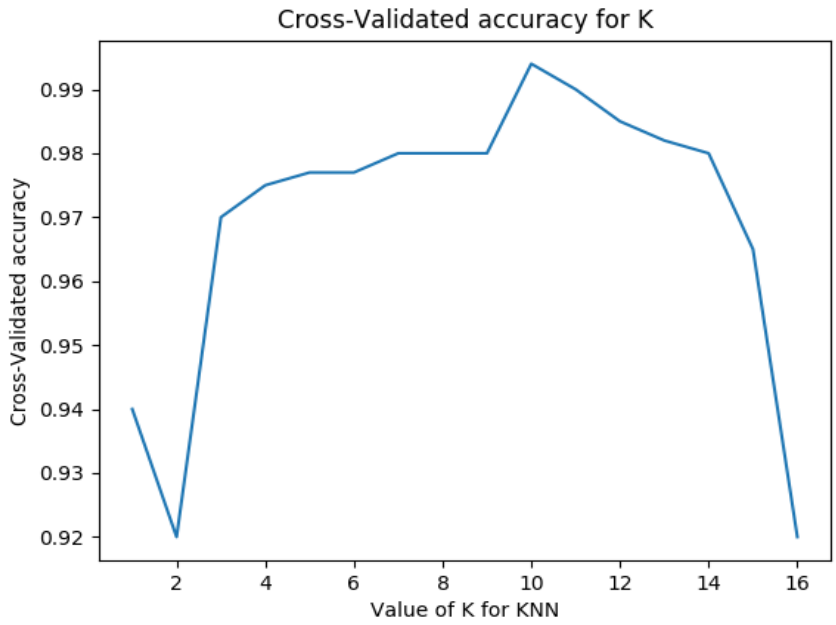

Figure 48: Cross-Validated Accuracy Check for K

From above figure 48, it is clearly seen that the maximum cross-validated accuracy occurs at k=10. The general shape of the curve is upside down yield which is quite typical when examining the model complexity and accuracy. That means low the values of k, low bias but high variance i.e. the 1-nearest Neighbor classifier is the most complex nearest neighbor model and has most jagged decision boundary and is most likely to over-fit. But when the values of k are high, it has high bias but low variance, thus most likely to under-fit. Therefore, the best value is the middle of k which is 10 as per above figure 48 and also value of k=10 is generally recommended that has been shown experimentally to produce the best out-of-sample estimate.

4.6 Performance Evaluations

Since there is no availability of latest dataset to validate the result with, thus the standard worldwide acceptable IDS system known as snort is taken as benchmark system. There is an assumption that the detection by snort is 100% true. For evaluation purpose, the latest real time DDoS dataset of almost an hour of attack provided by one the leading ISP, Vianet is being used. The simulated traffic is used to trained the system and maintaining profile history of the network traffic. The total captured packets of simulated attack are 5829051 observed for 7 minutes and that of Vianet, real time dataset is 2700359 of packets being hit for an hour. The detection and classification of DDoS attacks are tabulated in packets wise below:

Table 1: Evaluation Results for Simulated DDoS Attacks (Snort)

Attack	**TP**	**FP**	**FN**	**Recall**	**Precision**
ICMP Flood	3462900	0	0	100 %	100%
SYN Attack	537407	0	0	1000%	100%
ACK Flood	9800	0	0	100%	100%
SYN FIN Flood	529537	0	0	100%	100%
Total	4539644	0	0	100%	100%

Table 2: Evaluation Results for Vianet DDoS Attacks (Snort)

Attack	**TP**	**FP**	**FN**	**Recall**	**Precision**
ICMP Flood	338400	0	0	100 %	100%

The table 1 and table 2 shows all the detection by snort is 100% correct and acts as benchmark for both the simulated traffic and the real time DDoS dataset.

Table 3: Evaluation Results for Simulated DDoS Attacks (Proposed Model)

Attack	**TP**	**FP**	**FN**	**Recall**	**Precision**

ICMP Flood	3436755	6024	20121	99.41 %	99.82%
SYN Attack	526083	5503	5821	98.90%	98.96%
ACK Flood	9354	211	235	97.54%	97.79%
SYN FIN Flood	523331	2996	3210	99.39%	99.43%
Total	4495523	14734	29387	99.35%	99.67%

Table 4: Evaluation Results for Vianet DDoS Attacks (Proposed Model)

Attack	**TP**	**FP**	**FN**	**Recall**	**Precision**
ICMP Flood	328375	4923	5102	99.47%	99.52%

In table 3 and 4, all the encountered DDoS attacks are tabulated showing the number of packets detected as ICMP, ACK, SYN FIN and SYN Flood respectively and calculated recall and precision against respective attacks using proposed model.

Table 5: Result Validation of Simulated DDoS Traffic

SN	**Algorithm**	**Detection Rate**	**Recall**	**Precision**	**Time**
1	Snort	100%	100%	100%	3:56
2	Proposed Model	99.02%	99.35%	99.67%	3:59

Table 6: Result Validation of Vianet Traffic

SN	**Algorithm**	**Detection Rate**	**Recall**	**Precision**	**Time**
1	Snort	100%	100%	100%	2:50
2	Proposed Model	99.13%	99.47%	99.52%	2:55

The tables 5 and 6 illustrate that the proposed model is very closest to the benchmark IDS in all aspects like detection rate, recall percentage and precision. And also the execution complexity is negligibly difference as compared to snort.

4.2 Comparison

The graphical representation of obtained results shown in below Figure 49 and 50.

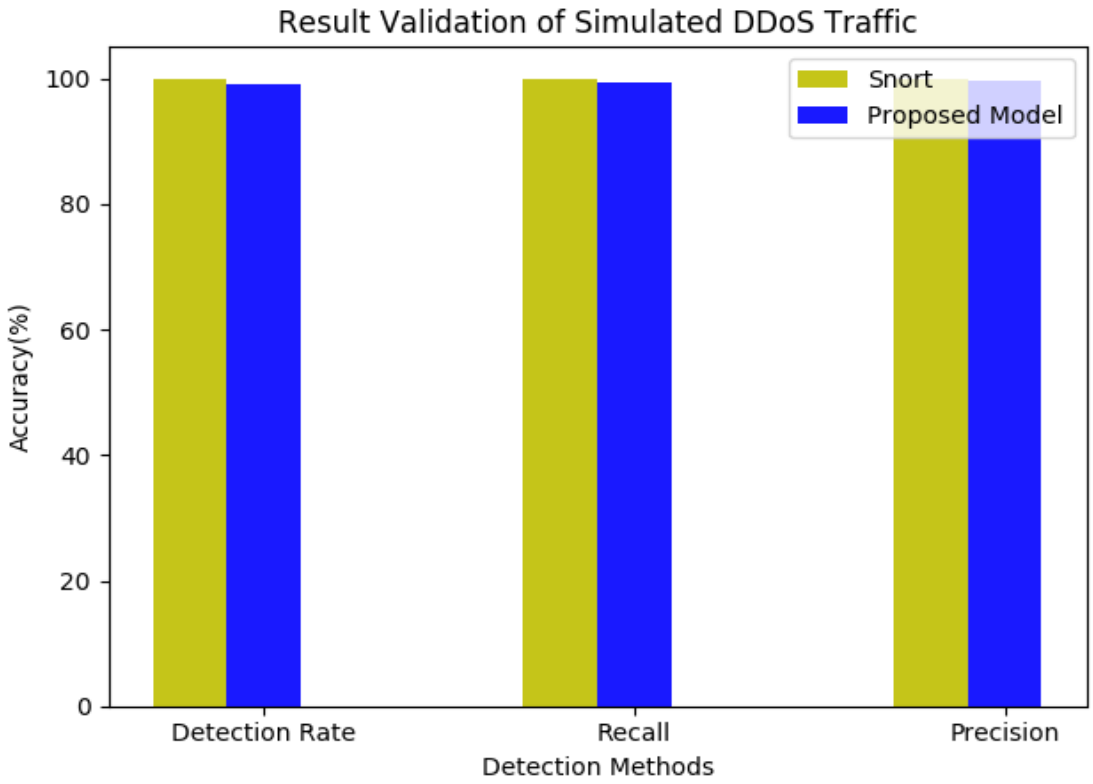

Figure 49: Accuracies comparison between Snort Vs Proposed Model in Simulated Attack

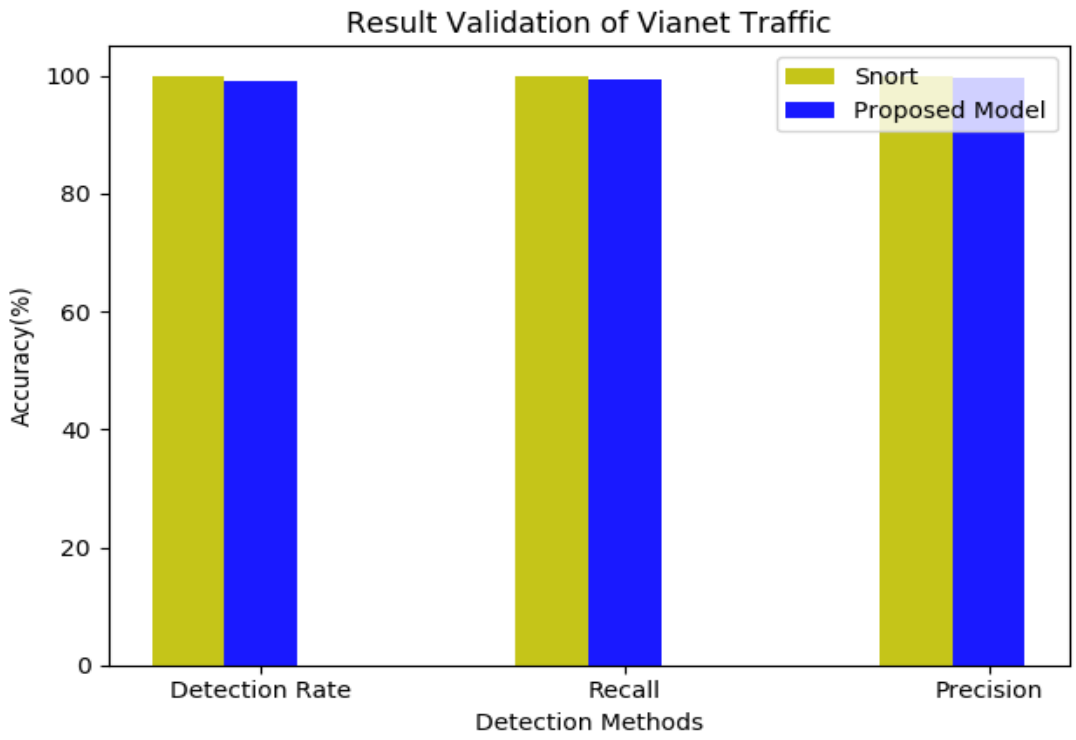

Figure 50: Accuracies comparison between Snort Vs Proposed Model in real time Vianet Attack

From figure 49 and 50, it is clearly visualized that the proposed model is very closest to the benchmark IDS, Snort in all aspects of accuracy measurement parameters like detection rate, recall percentage and precision in accordance with no significant difference in execution complexity.

CHAPTER 5: CONCLUSION

The proposed detection approach hybridizes the traditional and latest detection trend of using machine learning, results in good accuracy with no heavy difference in system execution complexity. The introduction of multiple features extraction attributes rather than relying on single features like Source IP alone of network traffic, helps to reduce the false alarm. The key terms dynamic threshold and its history profiling makes the detection efficient both on high as well as in low-rate traffic flow.

CHAPTER 6: LIMITATION AND FUTURE ENHANCEMENT

Although an entropy based approach minimizes the false positive rate but its complexities rises as per high traffic rate. This approach is difficult in classifying the type of attacks as we increase the size of cluster number, the result gets worst. Therefore, it is required to set accurate value of K in order to get it work effectively. Moreover, there is slightly misdetection on low-rate traffic as there is no significant changes in entropies.

Since the system mostly tested and validated in this research is with IPv4 based server and in the legacy network based system only. Thus, this approach can also be implemented with the emerging worldwide IPv6 network as well as in Software Defined Network (SDN). It can also be tested over Software Defined IPv6 (SoDIP6) networks as future works.

REFERENCE

[1] Jaswinder Singh, Monika Sachdeva and Krishan Kuman, 2013, “Detection of DDoS attacks using source IP based entropy”, International Journal of Computer Science Engineering and Information Technology Research, Vol. 3.

[2] Jisa David and Ciza Thomas, 2015, “DDoS attack detection using Fast Entropy approach on flow based Network Traffic” ISBCC.

[3] Wesam Bhaya and Mehdi Ebady Manaa, 2017, “DDoS attack detection approach using an efficient cluster analysis in large data scale”, NTICT.

[4] Suratose Tritilanunt, Suphannee Sivakorn and Ausanee Siripornpisan, 2016, “Entropy-based Input-Output traffic mode Detetion scheme for DDoS attacks”.

[5] Xi Qin, Tongge Xu and Chao Wang, 2015, “DDoS attack detection using Flow Entropy and Clustering Technique”, ICCIS.

[6] Luo, J.T., Yang, X.L., Wang, J., Xu, J., Sun, J. and Long, K.P. (2014) On a Mathematical Model for Low-Rate Shrew DDoS. IEEE Transactions on Information Forensics and Security, 9, 1069-1083.

[7] Yihua Liao, V. RaoVemuri, 2002, "Use of K-Nearest Neighbor classifier for intrusion detection", Computers & Security, Vol 21, No 5, pp 439-448

[8] Thwe Thwe Oo, Thandar Phyu, 2013, “DDoS Detection System based on a Combined Data mining Approach”, 4th International Conference on Science and Engineering.

[9] Distributed Denial-of-Service system overview diagram. [online]. Available: https://javapipe.com/images/distributed-denial-of-service-attack.png.

[10] Akash Mittal, Prof. Ajit Kumar Shrivastava and Dr.Manish Manoria, November 2011, “A Review of DDOS Attack and its Countermeasures in TCP Based Networks”, International Journal of Computer Science & Engineering Survey (IJCSES) Vol .2, No.4.

[11] Mitko Bogdanoski, Tomislav Shuminoski, Aleksandar Risteski, June 2013, Analysis of SYN Flood DOS attack, I.J. Computer network and information security, 8, 1-11.

[12] Anukool Lakhina, Mark Crovella, Christophe Diot, 2005, "Mining Anomalies Using Traffic Feature Distributions”, ACM.

[13] M. Suresh and R. Anitha, 2011, “Evaluating Machine Learning Algorithms for Detecting DDoS Attacks,” in International Conference on Network Security and Applications, pp. 441-452.

[14] W. Bhaya and M.E. Manaa, 2014, “Review Clustering Mechanisms of Distributed Denial of Service Attacks,”Journal of Computer Science, vol.10, no. 10, pp.2037-2046.

APPENDIX

Annex I

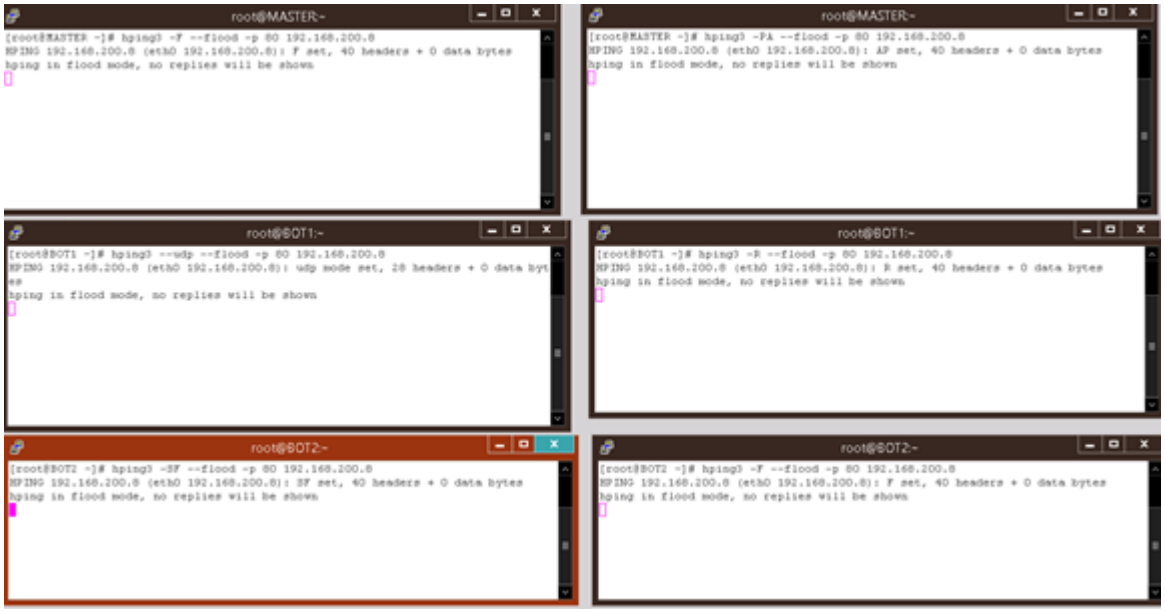

Figure 51: Snapshots of attempting attack with SYN, Push ACK, FIN and RESET Floods

The figure 51 show that how the hping3 command is used for different type of DoS attacks like SYN Attack, Push ACK, FIN Flood and RESET Floods to the targeted system.

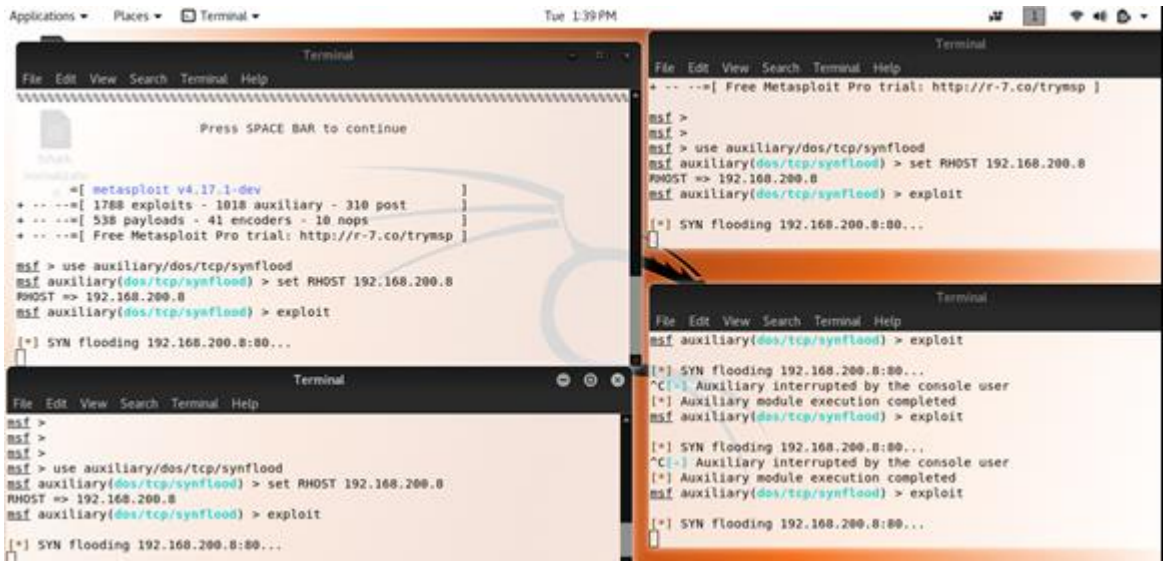

Figure 52: Snapshots of attempting attack with metasploit in kali linux

The above figure 52 illustrate how does kali linux executes SYN floods attack using the module called metasploit

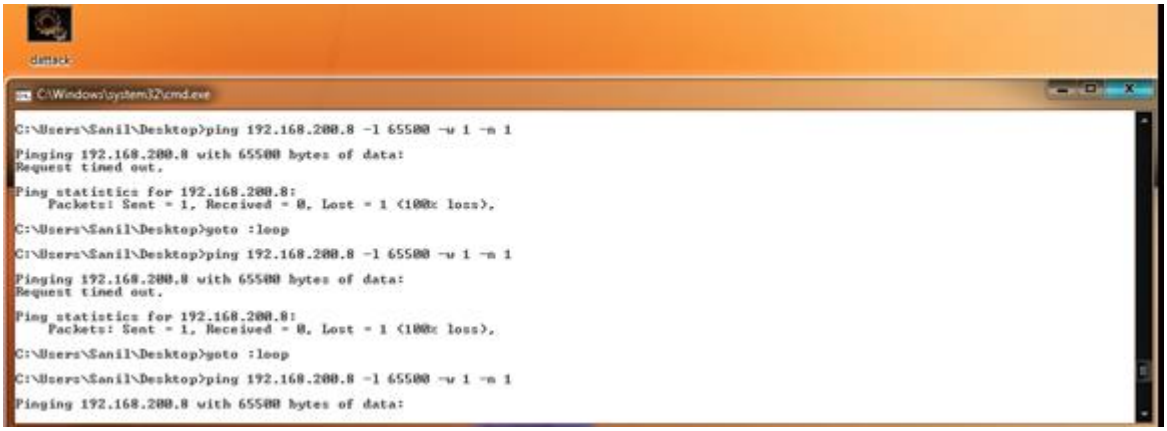

Figure 53: Snapshots of attempting attack with Ping of Deaths with windows OS

The above figure 53 shows the ping of Death batch script runs infinity loop to crash, destabilize, or freeze the targeted computer or service by sending malformed or oversized packets using a simple ping command.

Annex II

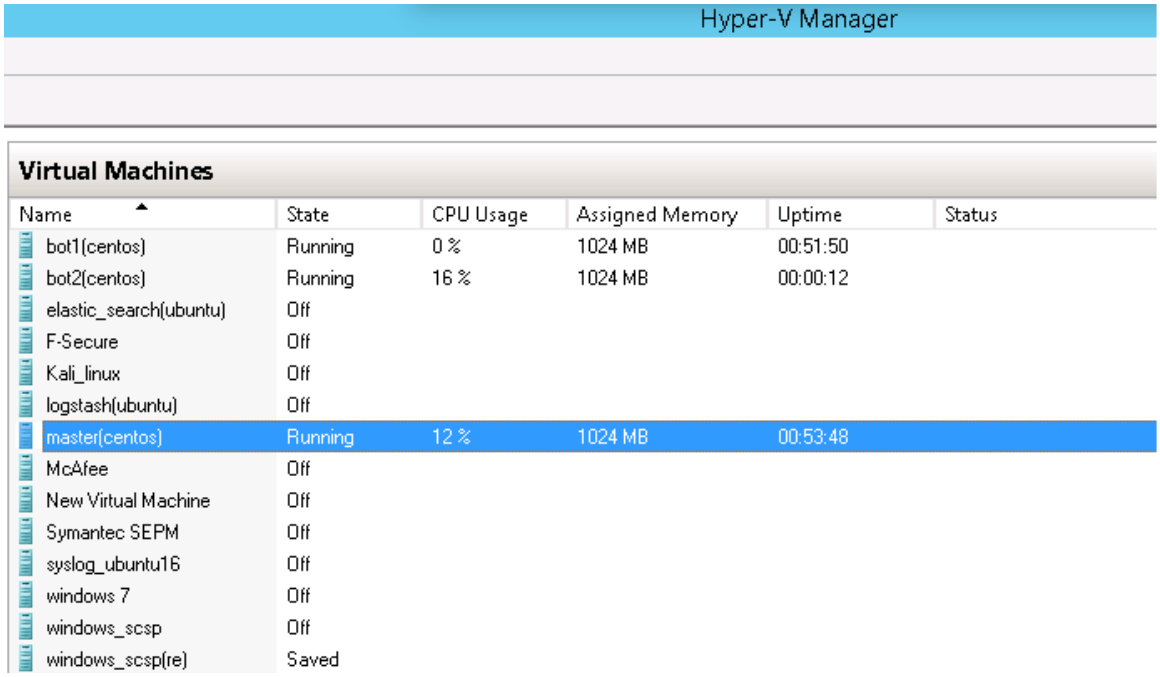

Figure 54: Snapshots of Hyper-V for Botnets and Master

With reference to above figure 54, there are three virtual machines running for the purpose of demonstration, how the botnets are created and take command by master to trigger for attacks.

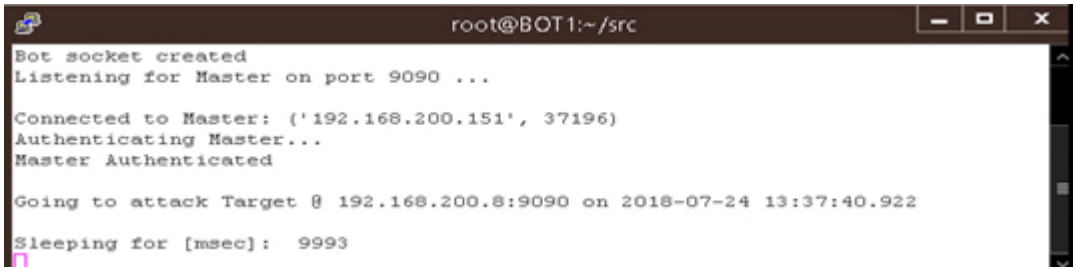

Figure 55: Snapshots of Initializing Bots

The above figure 55 glimpse about the initializing for bot or zombie whose primarily motto is waiting for the master or handler's command to execute the attack.

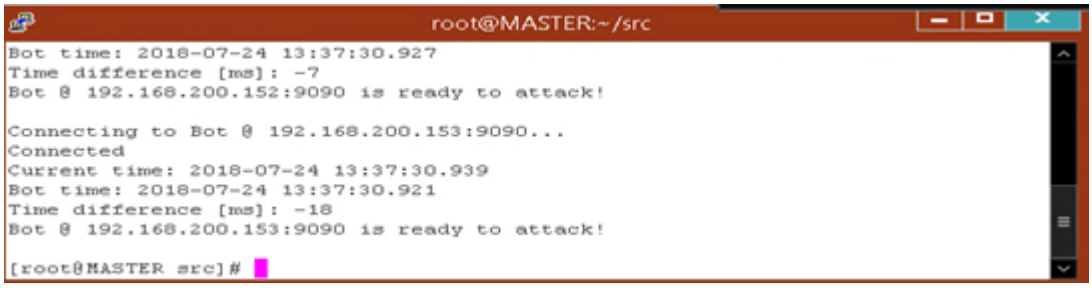

Figure 56: Snapshots of Master taking control of other bots

Accordance to figure 56, master takes control over the active botnets and commands to all bots to fire in given fixed time.

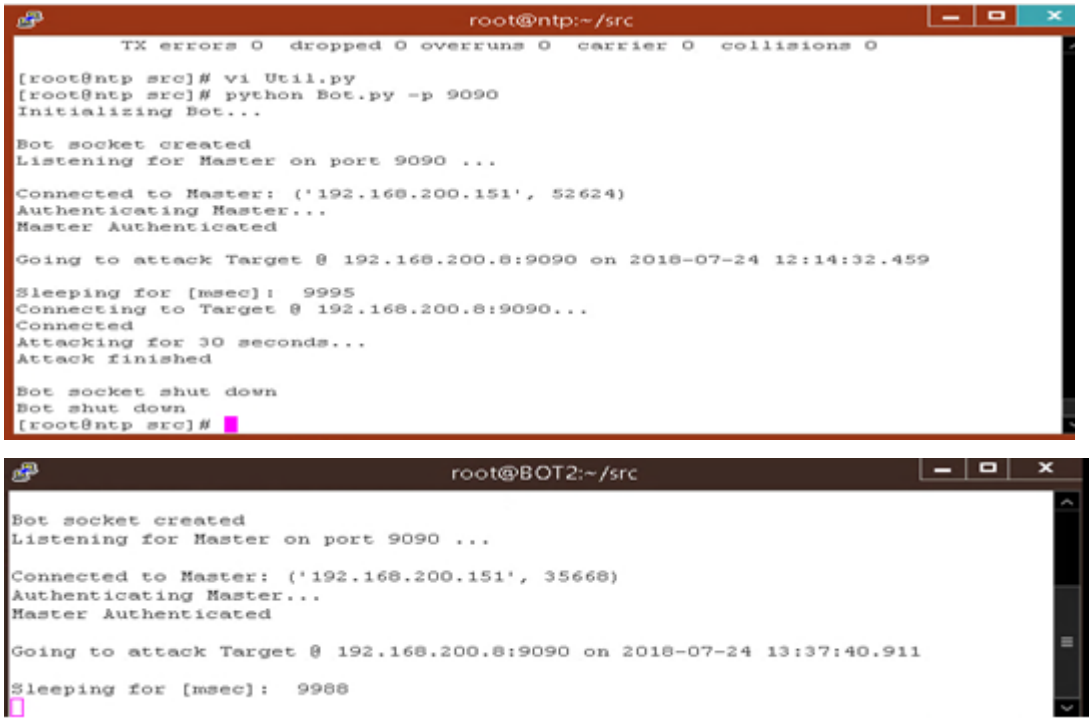

Figure 57: Snapshots of Bots are controlled by master

The figure 57 shows how the master controlled the bots with certain ports i.e 9090 and makes them to execute at the same time.

Annex III

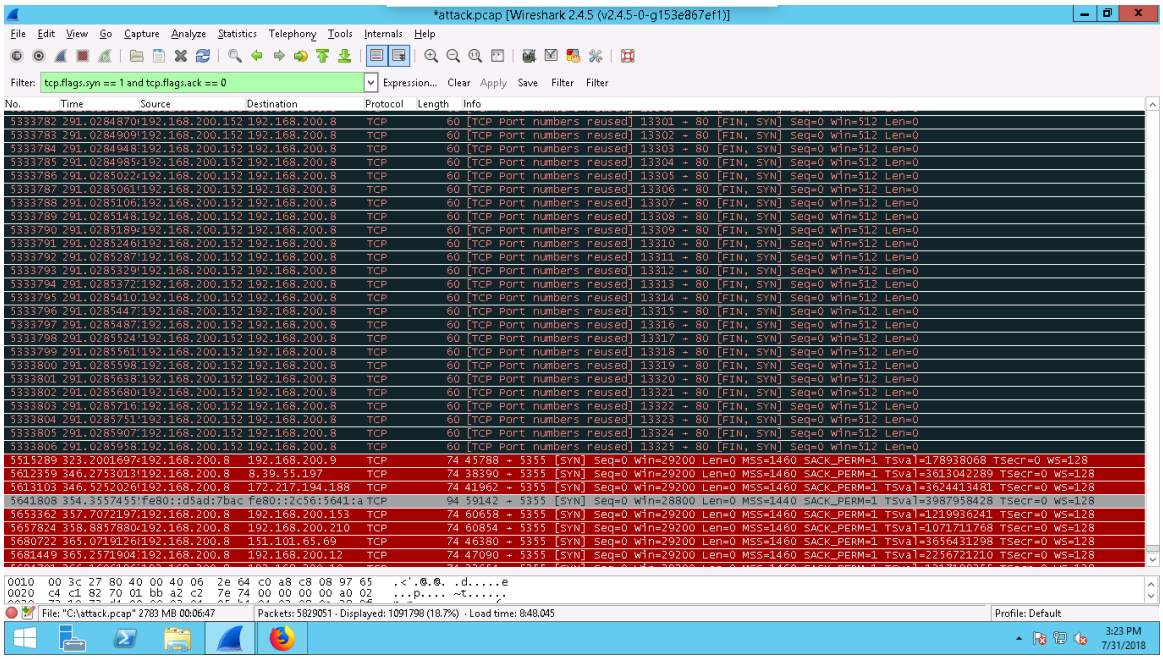

Figure 58: Snapshots of wireshark visualized SYN attacks

With reference to figure 58, it is seen that wireshark is one of the good tool to identify whether there is the SYN attacks or not in the network.

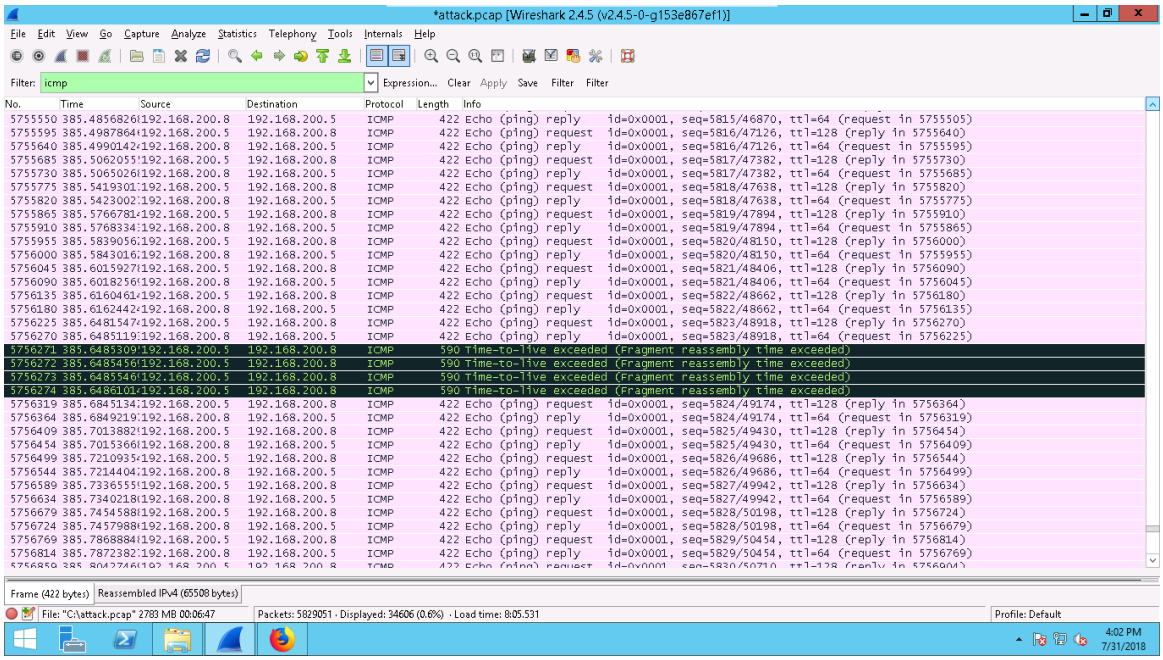

Figure 59: Snapshots of wireshark visualized packets fragmented

The above figure 59 shows how wireshark can be used for visualizing whether there are bad fragmented packets in the network aimed for making system busy for nothing.