

**Anomaly Detection Framework to Prevent DDoS Attack in Fog  
Empowered IoT Networks**

**A THESIS REPORT**

SUBMITTED IN PARTIAL FULFILLMENT OF THE  
REQUIREMENTS FOR THE AWARD OF THE DEGREE OF  
BACHELOR OF ENGINEERING(B.E.) IN DIVISION OF

**INFORMATION TECHNOLOGY**

SUBMITTED BY

**Gaurav Agrawal 2017UIT2626**

**Tarun 2017UIT2615**

UNDER THE GUIDANCE OF

**Mr. Satish Kumar Singh**



**NETAJI SUBHAS INSTITUTE OF TECHNOLOGY  
UNIVERSITY OF DELHI**

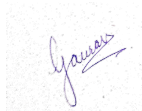
# INDEX

<b>ACKNOWLEDGEMENT</b>	<b>ii</b>
<b>DECLARATION</b>	<b>iii</b>
<b>CERTIFICATE</b>	<b>iv</b>
<b>LIST OF FIGURES</b>	<b>v</b>
<b>ABSTRACT</b>	<b>vii</b>
<b>CHAPTER 1 - INTRODUCTION</b>	<b>1</b>
1.1 CONCEPTS	2
1.2 MOTIVATION	5
1.3 AIM AND OBJECTIVE	6
<b>CHAPTER 2 - DETAILED ANALYSIS</b>	<b>7</b>
2.1 CHALLENGES IN IOT	7
2.2 ADVANTAGES OF FOG COMPUTING	8
2.3 ANOMALY DETECTION	9
2.4 TCP SYN	10
2.5 SMURF ATTACK	11
2.6 RELATED WORK	12
<b>CHAPTER 3 - PROPOSED APPROACH</b>	<b>13</b>
3.1 DATA PREPROCESSING	14
3.2 PCA	16
3.3 CRPS-ES	18
3.4 TRAINING ALGORITHM	19
3.5 TESTING ALGORITHM	20
<b>CHAPTER 4 - EXPERIMENT AND RESULTS</b>	<b>22</b>
4.1 DATASET	22
4.2 TRAINING DATA FOR TCP-SYN ATTACK	24
4.3 TESTING DATA FOR TCP-SYN ATTACK	26
4.4 TRAINING DATA FOR ICMP ATTACK	28
4.5 TESTING DATA FOR ICMP ATTACK	30
<b>CHAPTER 5</b>	<b>32</b>
5.1 CONCLUSION	32
5.2 FUTURE WORKS	32
<b>REFERENCES</b>	<b>33</b>
<b>PLAGIARISM REPORT</b>	<b>35</b>
<b>APPENDIX</b>	<b>39</b>
PCA	39
KERNEL DENSITY ESTIMATION (KDE)	41

# ACKNOWLEDGEMENT

Foremost, our thanks and praises to the lord almighty, for his esteemed blessings throughout the work which led to the completion of our project successfully.

We would sincerely like to express our deep gratitude to our research supervisor, Mr. Satish Kumar Singh, for giving us the opportunity to do this project under his guidance and providing his invaluable support throughout our project. His vision, dynamism, motivation and sincerity deeply inspired us. It was an extreme privilege and honour to work under his guidance. His able and helpful guidance and abilities along with his appreciation of the work and constructive feedback made the process easier.



**Gaurav Agrawal (2017UIT2626)**



**Tarun (2017UIT2615)**

# DECLARATION



**Division of Information technology**

**University of Delhi**


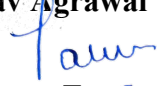
**Delhi-110007, India**

We, Gaurav Agrawal(2017UIT2626), Tarun(2017UIT2615), students of B.E., Division of Information Technology, hereby declare that the Project-Thesis titled “Anomaly detection framework to prevent DDoS attack in Fog empowered IoT networks” which is submitted by us to the Division of Information Technology, Netaji Subhas Institute of Technology, Delhi (University of Delhi) in partial fulfillment of the requirement for the award of the degree of Bachelor of Engineering, is original and not copied from source without proper citation. This work has not previously formed the basis of the award of any degree.

**Place: Delhi**

**(Name and signature of student(s))**

**Date: 31st December 2020**

  
**Gaurav Agrawal**  
  
**Tarun**

# CERTIFICATE



**Division of Information technology**

**University of Delhi**

**Delhi-110007, India**

This is to certify that the work embodied in the Project-Thesis titled “Anomaly detection framework to prevent DDoS attack in Fog empowered IoT networks” has been completed by Gaurav Agrawal(2017UIT2626), Tarun(2017UIT2615), students of B.E., Division of Information Technology, under my guidance towards fulfilment of the requirements for the award of the degree of Bachelor of Engineering. In my best knowledge ,this work is based on original research and has not been submitted in full or in part for any other diploma or degree of any university.

**Place: Delhi**

**(Name and signature of supervisor)**

**Date: 31st December 2020**

**Mr. Satish Kumar Singh**

# LIST OF FIGURES

**Fig 1.1:** IoT as a network of various entities

**Fig 1.2:** Cloud Computing

**Fig 1.3:** Fog Computing

**Fig 1.4:** Cyber Attacks

**Fig 2.1:** TCP SYN Attack

**Fig 2.2:** Smurf Attack

**Fig 3.1:** The Proposed Architecture

**Fig 3.2:** Difference in the average time between the packets given the number of packets is same

**Fig 3.3:** Principal Components in PCA

**Fig 3.4 :** Flowchart for CRPS-ES

**Fig 4.1 :** Snapshot of DARPA99 dataset using Wireshark tool.

**Fig 4.2 :** First Feature of TCP SYN packets in training dataset

**Fig 4.3 :** Second Feature of TCP SYN packets in training dataset

**Fig 4.4 :** CRPS-ES values for training data after applying PC

**Fig 4.5 :** Distribution of the CRPS-ES calculated using KDE

**Fig 4.6 :** First Feature of TCP SYN packets in testing dataset

**Fig 4.7 :** Second Feature of TCP SYN packets in testing dataset

**Fig 4.8 :** CRPS-ES values for testing data after applying PCA

**Fig 4.9 :** First Feature of ICMP Echo reply packets in training dataset

**Fig 4.10 :** Second Feature of ICMP Echo reply packets in training dataset

**Fig 4.11:** CRPS-ES values for training data after applying PCA

**Fig 4.12 :** Distribution of the CRPS-ES calculated using KDE

**Fig 4.13 :** First Feature of ICMP Echo reply packets in testing dataset

**Fig 4.14 :** Second Feature of ICMP Echo reply packets in testing dataset

**Fig 4.15 :** CRPS-ES values for testing data after applying PCA

# ABSTRACT

Internet of things or in short IoT is a network of interconnected entities such as computing devices, mechanical machines, digital gadgets etc. These entities work with each other by either collecting or producing some data or by processing some data in order to provide a smart adaptable environment for the users. In smart home environments, all the home appliances communicate with each other to provide better living experience for e.g. a thermostat can inform the air conditioner to regulate the room's temperature.

The ease of use of IoT is backed by a strong Cloud based infrastructure which allows the sensory IoT devices to perform their specific functions. An important feature of cloud is its reliability and security where the latter must be dealt with proper care. Cloud centric systems are susceptible to Denial of Service (DoS) attacks wherein the cloud server is subjected to an overwhelming number of incoming requests by a malicious device. If the same attack is carried out by a network of devices such as IoT devices then it becomes a Distributed DoS (DDoS) attack. A DDoS attack may render the server useless for a long period of time causing the services to crash due to extensive load.

In this project we will try to introduce the concept of fog computing and try to explain its importance in a 3-tier architecture. We have proposed an anomaly detection architecture for IoT networks where the detection actually happens on the fog layer. The algorithm is based on the CRPS metric which is a single variable algorithm which is the case in most statistical algorithms. Therefore, we have proposed a way to use multiple variables and shown why it is required in a heterogeneous network like IoT. In the end we have demonstrated the working of the algorithm in DARPA-99 dataset developed by MIT which is a dataset based on Wireless networks and perfectly suits our case. In the end we have proposed some future works which we can achieve if we continue with this project in the next semester.



# CHAPTER 1 - INTRODUCTION

Internet of things or in short IoT is a network of interconnected entities such as computing devices, mechanical machines, digital gadgets etc. These entities work with each other by either collecting or producing some data or by processing some data in order to provide a smart adaptable environment for the users. In smart home environments, all the home appliances communicate with each other to provide better living experience for e.g. a thermostat can inform the air conditioner to regulate the room's temperature. There are many such use cases of IoT networks such as smart traffic systems, smart security networks etc. It is estimated that there will be more than 21 billion IoT devices by 2025. We'll be using apps to control the LEDs, door locks and other home appliances in our houses. Our cars will be able to communicate with each other to avoid road accidents. Cities will have a huge network of interconnected security cameras, alert systems etc.

But with this level of connectivity comes a great risk of security. Devices connected to an IoT network are very prone to cyberattacks. Therefore, there is a need to develop security mechanisms, attack detection systems, attack mitigation systems, secure IoT devices and connections etc. in order to make next generation IoT networks more secure and reliable. Many researchers have proposed such attack detection and mitigation techniques working at different layers of the network architecture and preventing the systems from various cyber attacks. A cyberattack over a network creates anomalous behavior in the network traffic which means the flow of packets is different than usual and this fact can be exploited to detect attacks using various anomaly detection algorithms. Many anomaly detection techniques based on artificial intelligence, neural networks, statistics etc have been used and these techniques have shown promising results against many attacks such as phishing attacks, DDoS attacks, sinkhole attacks etc. In this work we'll propose a similar but efficient anomaly detection technique to prevent DDoS (Distributed Denial of Service) attacks in fog-empowered IoT networks.

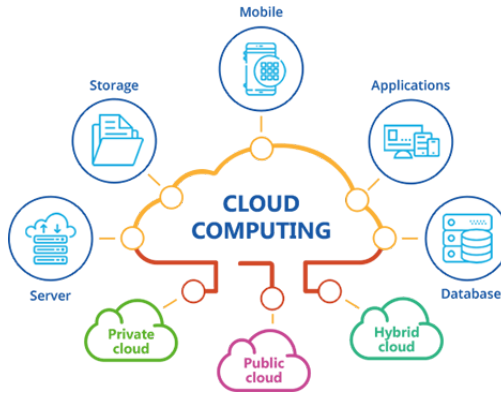
## 1.1 CONCEPTS

1. **IoT (Internet of Things)** : As discussed in the Introduction, IoT is a network of interconnected entities working in coordination with each other to make smarter decisions. The exact definition of IoT has evolved over time due to the introduction of multiple technologies like real-time analytics, blockchain, machine learning, big data etc. in the field of computer networks. The purpose of IoT networks is basically to exchange data collected by local devices or things with other systems over the internet.



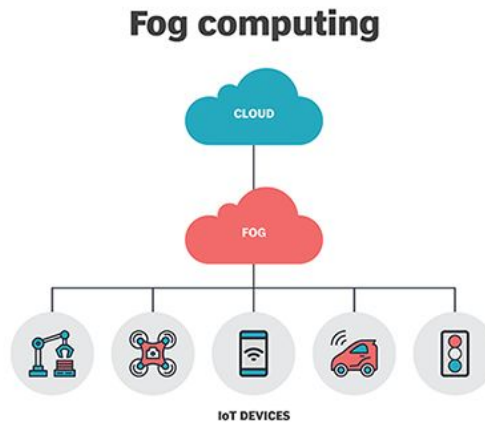
**Fig 1.1: IoT as a network of various entities**

2. **Cloud Computing** : Cloud Computing is defined as the on-demand availability of computer resources such as storage, processing power etc with the help of data centers available over the internet. The resources are owned and managed by an organisation and any user using the internet can share these resources after getting the required access. The term “Cloud” is used to reference these data centers only. Cloud Computing helps many small and large enterprises in managing their business by providing a secure centralised hub for all the computing requirements such as data management, data analytics etc..



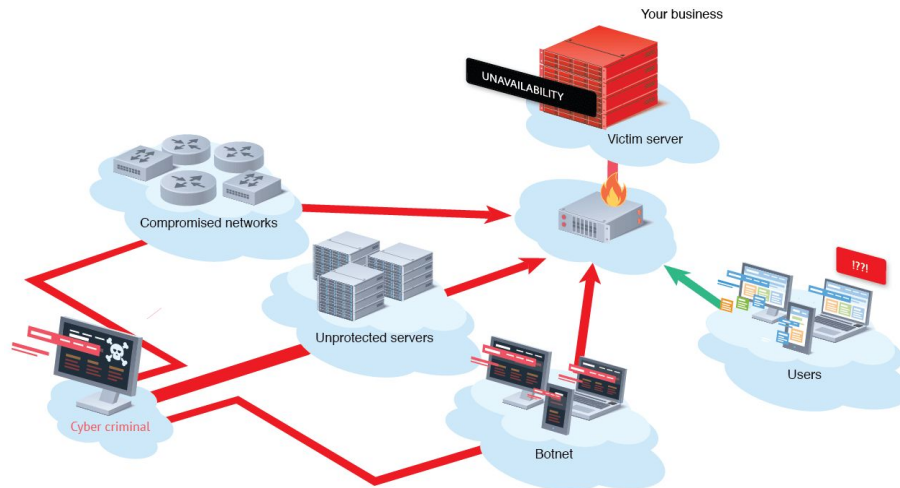
**Fig 1.2: Cloud Computing**

3. **Fog Computing** : Fog Computing is a network architecture where rather than passing every sensory information to the centralised servers, data collected can be processed in local hubs also known as fog nodes located near to the sensory nodes which are collecting the data. This addition of local servers helps the network in taking real-time decisions and prevent the overhead caused by communication with the central cloud servers. Fog nodes act as a middle layer which can process urgent requests and then pass the information or data to the cloud for some high level analytics and data processing [2].



**Fig 1.3: Fog Computing**

4. **Cyber Attacks** : A cyber attack is an attack launched by red hat hackers or cybercriminals with an intention to damage or compromise a computer system or a network using one or more computers. A cyber attack can leak private data, damage servers, cause financial harm, stage another attack etc. There are many types of attacks that can be performed but attacks like denial of service attacks, phishing attacks, sinkhole attacks etc. are most commonly used[3].



**Fig 1.4: Cyber Attacks**

## 1.2 MOTIVATION

IoT works on two layer architecture or a three layer architecture and is vulnerable to various cyber security attacks. One such attack is Distributed denial of services(DDoS). DDoS attack is an attempt to disrupt targeted servers in a malicious way. DDoS attack takes help of some compromised nodes and floods the targeted server with requests ultimately leading to the breakdown of the server. At a higher standard, DDoS attacks clog the destination server with unexpected traffic. DDoS attacks are a little less complicated than any other cyber Security attacks. There are 3 categories to DDoS attack a) Volume based attacks b) Protocol attacks c) Application based attacks. Common types of DDoS attack include SYN flood attack, UDP attack, ICMP attack etc.

Recently, it has been seen that IoT devices have been a major force to drive DDoS attacks. This is a threat that has not been diminished. Security in IoT has been a major talk as it opens up many avenues for attacks to take place. It is anticipated that around 20.4 million devices are due to be deployed by 2020, and it is safe to say that the scale of DDoS attack utilizing this vulnerability could have serious consequences. So, it is important that researchers work on a security technology that is well baked from the start.

Some further observation that motivated us to work on DDoS is the use of MultiVariate statistical algorithms in detection of DDoS attack. Considering multiple variables rather than single variables can bring out some interesting results.

## 1.3 AIM AND OBJECTIVE

With respect to the interest and motivation presented earlier our aim with this project is to propose an anomaly detection Model for fog empowered networks. With the help of fog nodes present in the network we can detect anomalies locally. It also eases the task of detecting the malicious node for future research works. We will discuss more about this in further sections.

We want to use statistics based algorithms for the purpose of anomaly detection(DDoS Detection). These algorithms are usually used for time series based datasets where packets tend to possess a trend in their incoming pattern. In a statistics based algorithm to detect the DOS and DDOS attacks every new traffic network measure is compared to the reference attack free traffic distribution. In simple words trends are analyzed and on the basis of the trends detected anomalies are spotted.

In most of the recent works, usually one feature for example : number of data packets per unit time, is used for the detection of DDoS attacks for time series based datasets. Our aim is to use more than one parameter by analyzing the dataset which will help in detection of anomalies. The IoT traffic is heterogeneous in nature so using one feature may not be a correct criteria always for judging the anomaly. Therefore, in further sections we will discuss the valuable parameters that can be extracted from the dataset and how we can detect anomalies by implementing statistics based algorithms. Our algorithm uses dimensionality reduction which will be used for extracting important information from features generated and the algorithm is based on Continuous ranked probability score(CRPS) score. CRPS is used in probabilistic forecasting. We will be using python for implementing the algorithm on a dataset and present the output.

# **CHAPTER 2 - DETAILED ANALYSIS**

## **2.1 CHALLENGES IN IOT**

Recently, the source code of a famous mirai malware was released, According to the experts the malware scans the web for IoT devices that havent been secured properly and infect them, as it's easier to hack them. After the devices have been compromised they are used as a part of a botnet that's directed to launch DDoS attacks with malicious intention. All the IoT devices at a global stores its information in the cloud, compromising the IoT devices means hackers can cause discrepancy in the information stored in the cloud which might cause a loss of large information. The problem is that the experts believe that the cyber security attacks on IoT are nowhere stopping here. With lack of security the attacks will increase in a vast number.

Being one of the hottest technologies in recent times, IoT should come with a great responsibility of security. Although it has numerous benefits, these devices are easily hacked which gives it a negative image in the market. IoT should be using new hardware and software which are being generated daily, eradicating the old technologies.

The main challenge with this technology is to detect anomalies in real time and check if the system is affected or not as early as possible since the system is monitoring multiple devices at all times[12]. Further examples of Security can be home security, autonomous vehicles security, data management using autonomous systems etc. DDoS attacks and cyber extortion are dangers looming on the normally brilliant horizon of the IoT. Enterprises and consumers alike should take steps to ensure the security of their networks, so that these common IoT exploits don't take them down.

## 2.2 ADVANTAGES OF FOG COMPUTING

As discussed in section 1.2, Fog Computing includes a 3-layered network architecture where the topmost layer is cloud layer which consists of centralised huge data centers and servers, middle layer is called fog layer where small local servers are present really close to the bottommost layer of sensory nodes to support low latency real time processing. Nodes in the fog layer are called fog nodes. These fog nodes receive data from a small environment and process that data to send desired results with very low propagation time delay. After sending the required response back, the data can then be sent to the cloud servers for further processing and analysis. Usually a group of smart homes or a small network of roads or a production line in a factory can have its own fog server node to collect and process data coming from a bunch of sensory nodes such as cameras, thermostats, pressure gauges etc.

Apart from providing low latency, real-time data processing and distributed network of processing units, addition of fog layer also provides security benefits. Because the data traffic coming from the sensory nodes passes through the fog nodes first, many security checks can be performed to prevent anomalous data or traffic from reaching cloud servers. This mechanism helps in avoiding cyber attacks like DoS or DDoS attacks to a great extent by alarming the network whenever a node tries to flood the server by sending a large amount of packets (SYN packets or Echo reply packets) in a very short amount of time[3]. On getting the alarm, traffic coming from a particular local network under the connection the fog node which raised the alarm can be dumped or blocked to keep the servers safe. The sensors also known as edge nodes are not very secure because of their simple design and application which makes them vulnerable to a lot of cyber attacks. Cyber criminals use these kinds of compromised nodes as bots to launch various kinds of distributed attacks usually on single points of failures in the networks. But the non-centralised nature of the fog empowered networks prevents these edge nodes from getting exposed in the public network and therefore improves security of the network.



## 2.3 ANOMALY DETECTION

Anomaly detection can be defined as the process of identifying abnormal or unexpected patterns within a data set. Abnormal or unexpected patterns are basically just data points which differ from the normal trend of the data set as a whole. For e.g. in water transportation pipelines, a particular range of pressure is important to be maintained and anything above or below that range can cause critical situations. In this case if the pressure shoots very high suddenly or maybe falls unexpectedly then it can be considered as an abnormal behaviour or an anomaly as compared to a normal scenario. These kinds of anomalies are introduced in the system either by accident or by human mistake. But there are also some cases where this is done by someone intentionally. For eg in cyber attacks, an attacker tries to compromise a node or some part of a network which creates anomalous behaviour in network traffic or communication.

As it can be inferred from the above mentioned examples that an anomaly is generally not desired. It is a rare situation but it can still lead to critical scenarios. So there is a need for anomaly detection mechanisms which can detect these abnormal behaviours in the system and warn the concerned authorities to take necessary actions. Researchers have proposed many effective anomaly detection techniques based on machine learning, neural networks, statistical analysis etc. These anomaly detection algorithms usually train on unsupervised datasets and try to find the pattern or we can say distribution of the normal trend then any new data point is compared to this pattern that the algorithm has extracted out. If the new data fits in the trend and follows the similar distribution then the algorithm classifies it as normal data otherwise it is classified as an anomaly. The comparison of the data point to the distribution can be based on various metrics. One such metric is called continuous ranked probability score (CRPS) which is used for probabilistic forecasting. We will discuss this metric in detail in next section.

As mentioned earlier there are 3 types of DDoD attack. They are :

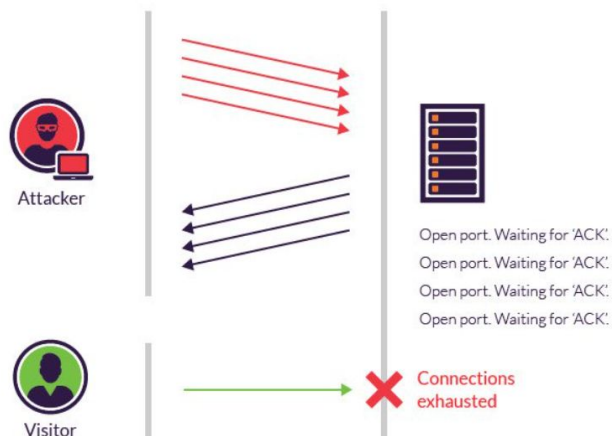
- a) Volume Based Attack
- b) Protocol based
- c) Application layer.

Here we are going to focus on 2 attacks that are included in the protocol based attacks. Let's discuss both in detail.

## 2.4 TCP SYN

TCP SYN is a type of DDoS (Distributed Denial Of Services) that exploits the TCP three way handshaking for attacking the targeted resource. It sends a large number of TCP requests that damps the server causing network saturation.

In TCP SYN a hacker sends a fake and large number of SYN packets to every port on the targeted server using fake IP addresses. The server is unaware of the malicious intent and starts responding to every request and replies them with an ACK packet. This causes an overloading on the server due to very high resource usage.



**Fig 2.1: TCP SYN Attack**

As the IoT devices are exposed to the internet and do not have a proper security can easily be compromised by which hacker can enable a TCP SYN attack to the cloud with some malicious intent.

## 2.5 SMURF ATTACK

It is a type of DDoS attack similar to ping floods carried out by sending a large number of ICMP echo packets.

A smurf attack scenario can be broken down as :

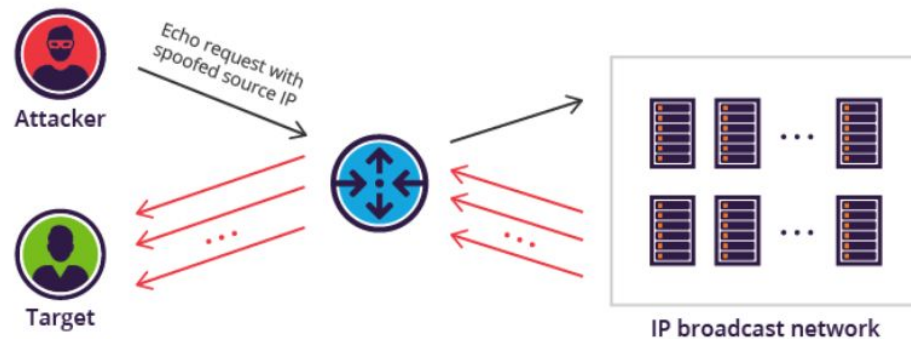
Smurf malware is used to generate a fake Echo request containing a spoofed source IP, which is actually the target server address.

The request is sent to an intermediate IP broadcast network.

The request is transmitted to all of the network hosts on the network.

Each host sends an ICMP response to the spoofed source address.

Similar to TCP SYN attacks they are very common in IoT for the same reasons.



**Fig 2.2: Smurf Attack**

## 2.6 RELATED WORK

In the past, many researchers have proposed various DDoS detection and mitigation techniques. Machine learning algorithms and Artificial neural networks are very popular and useful choices for detecting such protocol based attacks in wireless networks.

In [10], Wang et.al. used a semi supervised clustering algorithm to detect DDoS attacks. In this method, 3 features are selected by analysing the characteristics of the attack to form a detection feature vector. Then a multi-feature based constrained k-means algorithm is used which improves the convergence speed and the accuracy of the detection mechanism.

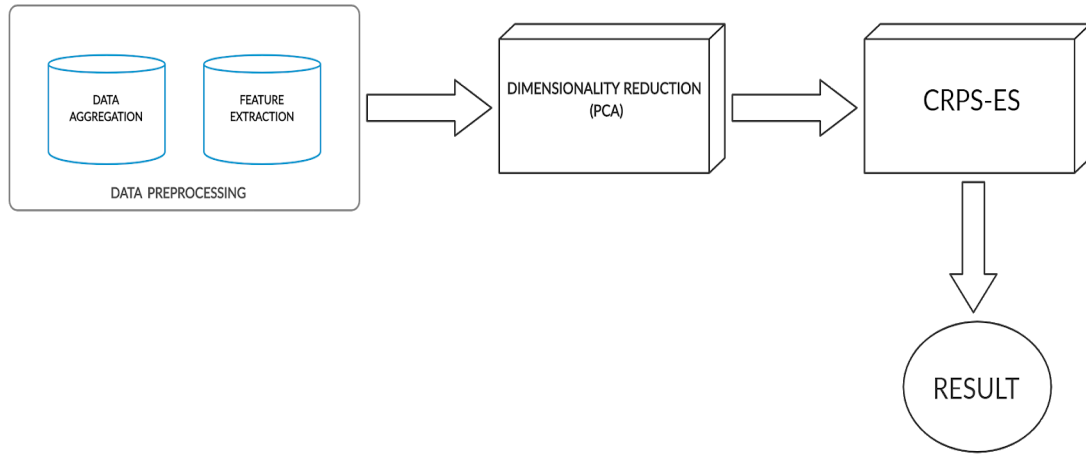
In [11], Sumathi et.al. proposed a DDoS detection mechanism using neural networks. Their model evaluates the network traffic using a deep learning classifier based on cost minimization strategy for publicly available datasets. They used detection accuracy, cost per sample, average delay, packet loss, overhead, packet delivery ratio and throughput as evaluation metrics for the performance analysis. The proposed algorithm works really well on the accuracy metrics but use of deep learning based methods can be computationally heavy for small edge devices.

In [4], Kadri et.al. explains a special statistical technique that can be used to detect DDoS attack based on a metric called CRPS. The paper demonstrates how CRPS can be used for probabilistic forecasting of DDoS attack. The paper also demonstrates how it can be used for real time detection in time series based environments. The proposed algorithm uses a single variable for detection of DDoS attack which might not always be a correct method of detecting DDoS attack in such heterogeneous networks like IoT.

We will try to propose an anomaly detection model based on CRPS for Fog-empowered networks with an attempt to show the importance of multiple features and demonstrate how we can use them to detect anomaly in a given traffic.

## CHAPTER 3 - PROPOSED APPROACH

In this work, we are proposing a 3-step anomaly detection framework to detect DDoS attacks in fog networks. Fig(3.1) shows the steps involved in the process starting from gathering data from the network traffic to finally detecting the attack. In our approach, we are using a statistical algorithm which works on features extracted from the incoming traffic. These features are first passed in a PCA component before putting the values in the algorithm. This is done because the algorithm used is single variate.



**Fig 3.1: The Proposed Architecture**

First step of the process is called the data preprocessing step where packets are bundled into time bound windows and then features like number of packets, average time between the packets, number of source IPs etc. can be extracted depending on the network state and requirements. Second step is a Principal Component Analysis(PCA) step where these features are transformed into principal components and the component with the maximum variance along its axis is selected as the main variable to be passed onto the next step.

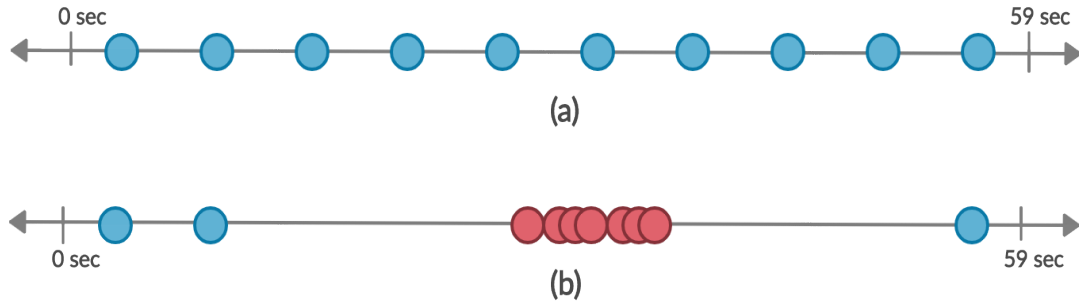
The final step is the detection of anomalies present in the dataset. This step uses a statistical metric called CRPS which is used to calculate the difference in the distribution of testing data and training data. This difference can indicate the measure of anomaly in the network traffic,

### 3.1 DATA PREPROCESSING

The proposed mechanism deals with time series data which means real-time sequential flow of packets moving from a few source ip addresses to some other destination ip addresses in a network. When these packets pass through the monitoring fog nodes, traffic information is gathered usually for network analysis, security check-ups, traffic behaviour monitoring etc. In our proposed architecture, packet information such as time-stamp, source IP, destination IP, type, packet length etc. is collected. These packets i.e the collected data entries, are then bundled in the groups or we can say windows of 60 seconds each based on the time-stamps. This is called the Data-Aggregation step. The window size can be changed depending on the network traffic.

In developing anomaly detection mechanisms for issues like DDoS attacks, extracting valuable features from the network or the traffic that can be exploited to detect suspicious behaviour of edge nodes or unexpected incoming network traffic is a challenging task. In most of the recently proposed time-series based DDoS detection mechanisms, the number of packets passing through the monitoring point within a fixed time limit or a time window, is used as a feature to detect attacks like TCP-SYN attacks or smurf attacks etc. This can give a rough idea about the network traffic and its behaviour but this feature is not sufficient enough to detect all the hidden patterns in the data. There can be many cases where the packet flow per time window is not exceptionally high but there are a few nodes which are consuming abnormal amount of energy from the network or if in case of DDoS attacks particularly, there can be few instances where some amount of packets were transmitted from a bunch of source node to a target destination node in a very short amount of time or even almost at the same time. Now this amount of packet flow can be detected as slightly anomalous if the total packet count within the time window is significantly large otherwise it will be considered as normal traffic. Whereas these impulses of packets from a group of different networks can accumulate and flood the target node.

Therefore, anomaly detection mechanisms based on multiple variables are more reliable and accurate. In this work, Along with the number of packets passing in the network within a time window, which is a common feature used by many researchers, the average time between the subsequent packers within the time window is used as an another parameter to tackle the above discussed issue of many nodes flooding the target by sending very high frequency of packets in a very short period of time. This another feature is important because this provides a new perspective to the situation. Suppose there is a window of say 60 seconds, in one case there are 100 packets passing through the network within these 60 seconds and the flow of packets is uniformly distributed throughout this time window, and in second case, there are no packets within first 30 seconds and then suddenly there are 100 packets within 1 second, from 30th to the 31st second and then again no packets till 60th second. Both these cases are showing a very different behaviour though the number of packets is the same. The only difference is the average time between the packets that can project that the flow in the second case is anomalous.



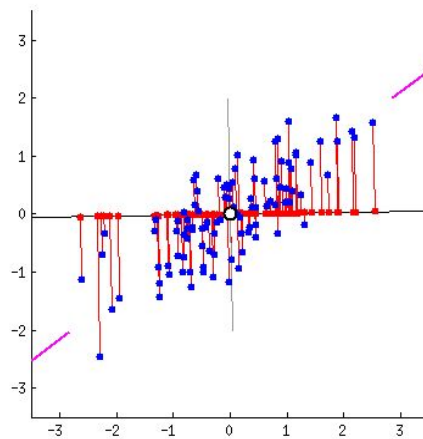
**Fig 3.2: Difference in the average time between the packets given the number of packets is same**

The calculation of both the above discussed features for every time window is performed in the Feature Extraction step which also concludes the Data Preprocessing. After that a trend based on the combined effect of the extracted features is calculated in the dimensionality reduction component which uses principal component analysis or PCA. We have discussed more about this in the next section.

## 3.2 PCA

Principal Component Analysis is a dimensionality reduction method that is often used in reducing dimensionalities in datasets by transforming a dataset with many variables into a smaller number of variables which stores most of the information of the original dataset. There can be many reasons to use PCA in your dataset, due to a large dataset you might want to reduce the number of variables or to remove unnecessary information from your dataset which can help you increase accuracy. PCA is performed by transforming the existing variables into a new set of variables which are known as principal components. These components are orthogonal to each other.

As there are many principal components, the component with the maximum variance is selected as the first principal component. Selecting the component with maximum variance means we are trying to retain the maximum variation in the dataset along an axis. As you can see in the figure below, the first component will be the line that aligns with the pink line. It is the line on which projections are most spread out. Mathematically speaking it is the line that maximizes variance(average of the square of the distances from the red dots(projected points) to the origin). Now, the second component will be the component with second highest variance with a condition that it is uncorrelated to the previous one.



**Fig 3.3: Principal Components**



For applying PCA certain conditions have to be met. 1) Data should be standardized 2) There should be some correlation between variables. Having no correlation at all will have no effect on the process and having an excessive correlation means both the variables can be considered the same. So, keeping these factors in mind PCA should be applied.

As discussed in the previous section(Pre processing) we will be using two variables for our algorithm. We have used PCA to convert the two variables into a single variable to meet our requirements. PCA will try to extract only the useful information from the two variables and convert it to a single variable. Time series based anomaly detection algorithms usually consider a single variable but using PCA we can produce a single variable which is a combination of variations along different metrics which can obviously be considered a better choice for anomaly detection considering the fact that the IoT traffic is heterogeneous in nature.

As discussed earlier there are certain conditions that have to be met to apply PCA. So it is very important to show the correlation between different variables that we have considered in the previous section. We will be using graphical results to explain the correlation in the Evaluation section.

### 3.3 CRPS-ES

CRPS or continuous rank probability score is a statistical metric which is commonly used in probabilistic forecasting models. In our work, we have selected this metric because CRPS can be used to quantify the dissimilarity between a new observation and the attack free traffic distribution [5].

This makes CRPS a better suited algorithm for real time detection as compared to other similar metrics like KL divergence or even  $\chi^2$ (chi square) which requires the whole data beforehand to compute distributions of training and testing data. At the time of testing, the difference between the CRPS values of the incoming data points and the attack free normal data can identify the abnormal behaviour of the network traffic [4].

To enhance the detection efficiency of CRPS we apply Exponential smoothing to it (CRPS-ES). This is done for the inclusion of previous and current information in the decision process which helps it in uncovering even small anomalies and makes the statistic less sensitive towards noise.

Usually the distribution of data is assumed to be gaussian but here we try to figure out a more realistic distribution for the data using the Kernel Density Estimation (KDE) [13]. CRPS-ES considers a non-parametric decision threshold which is computed by analyzing the flow of the underlying distribution. The major plus point of CRPS-ES metric is that it considers the past data in the detection which makes it sensitive to apparent attacks. It is more suitable for real time detection as the new traffic that might contain anomalies is compared to the attack free traffic.

### 3.4 TRAINING ALGORITHM

**Step-1.** In this phase we will have a preprocessed data with PCA applied to it. We will use the CRPS metric on this data. CRPS is helpful in quantifying the deviations from attack free data which help us in detecting anomalies. For a observation  $x$ , the CRPS value is calculated as :

$$CRPS(F, x) = \int_{-\infty}^{\infty} (F(y) - 1\{y \geq x\})^2 dy \quad (3.1)$$

Where  $1\{y \geq x\}$  :

$$1(x) = \begin{cases} 0, & x < 0 \\ 1, & x \geq 0 \end{cases}$$

It should be noted that when the traffic is attack free its distribution is gaussian. So, the CRPS for function  $F$  as gaussian will be :

$$CRPS(\mathcal{N}(\mu, \sigma^2), x) = \sigma \left[ \frac{x - \mu}{\sigma} (2\Phi\left(\frac{x - \mu}{\sigma}\right) - 1) + 2\phi\left(\frac{x - \mu}{\sigma}\right) - \frac{1}{\sqrt{\pi}} \right] \quad (3.2)$$

**Step-2.** Now the exponential smoothing is applied. This is done for the inclusion of previous and current information in the decision process which helps it in uncovering even small anomalies and makes the statistic less sensitive towards noise. The mathematical formula is calculated as :

$$z_t^{CRPS} = v d_t + (1 - v) z_{t-1}^{CRPS} \quad (3.3)$$

Where,  $z(t)$  is the calculated value of the current data point,  $z(t-1)$  is the calculated value of previous data ,  $d(t)$  is the observation of the current data point and  $v$  is the forgetting parameter.

**Step-3.** At this step, we have the calculated CRPS-ES values for the training data. Now, a non-parametric threshold ( $th$ ) value needs to be calculated to detect the anomalies in the testing phase. This threshold should be able to accurately justify the results i.e. if a value is crossing the threshold, it must differ from the underlying distribution of the training CRPS-ES statistic. This distribution is calculated using Kernel Density Estimation. The threshold is calculated as the  $(1-\alpha)$ th quantile of the calculated distribution.

### 3.5 TESTING ALGORITHM

From the training algorithm we have a threshold calculated which will be used in the Testing phase. Here, the data is not attack free. We will try to detect the TCP SYN attack.

**Step-1.** In this phase we will have a preprocessed data with PCA applied to it similar to the training phase. CRPS is applied similar to the training phase.

**Step-2.** Then Exponential smoothing is applied.

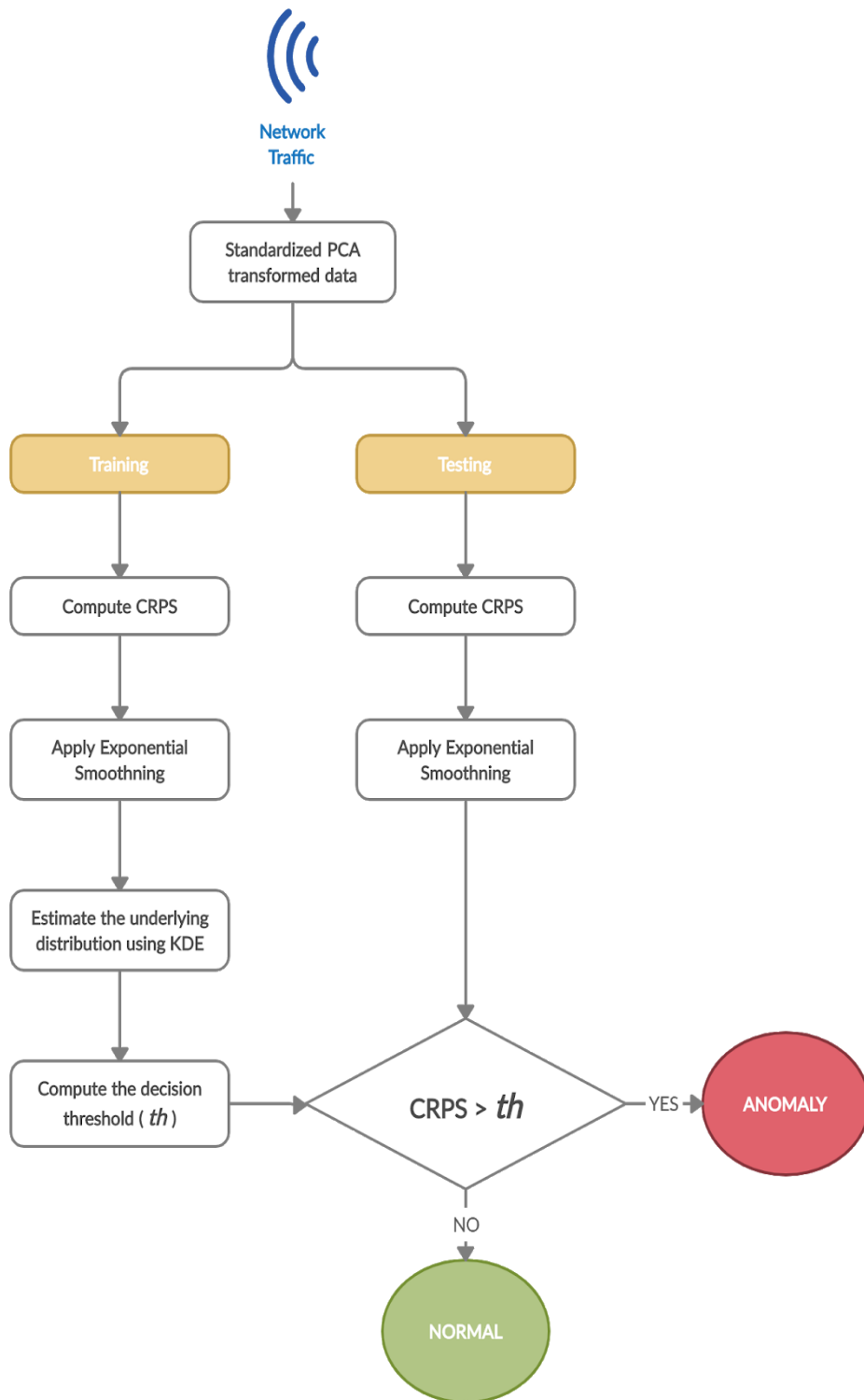
**Step-3.** The output of step 2 is compared to the threshold obtained from the training phase.

**If,** Case 1 :  $CRPS\_ES > th$

Anomaly present

**Else,** Case 2 :  $CRPS\_ES < th$

Normal traffic



**Fig 3.4 : Flowchart for CRPS-ES**

# CHAPTER 4 - EXPERIMENT AND RESULTS

## 4.1 DATASET

We have used a very popular dataset used for Wireless Sensor Networks to test our algorithm which is DARPA99 dataset. It is an anomaly detection evaluation dataset generated in 1999 by Lincoln Laboratory at MIT.

This dataset consists of data packets passing through a network over a period of 5 weeks. The first 3 weeks are a part of the training dataset with various labelled information and many different types of evaluations and observations most of which are out of the scope of this work. The last 2 weeks are for testing where the many cyber attacks like TCP-SYN, ICMP etc were launched on the network under observation.

No.	Time	Source	Destination	Protocol	Length	Info
99753	5707.537587	172.16.115.5	204.96.59.199	TCP	60	5879 → 80 [ACK] Seq=1
99754	5707.538471	172.16.115.5	204.96.59.199	HTTP	356	GET / HTTP/1.0
99755	5707.539211	172.16.113.105	197.182.91.233	SMTP	139	S: 220 swallow.eyrie.
99756	5707.551938	197.182.91.233	172.16.113.105	TCP	60	8331 → 25 [ACK] Seq=1
99757	5707.553466	204.96.59.199	172.16.115.5	TCP	60	80 → 5879 [ACK] Seq=1
99758	5707.559858	204.96.59.199	172.16.115.5	TCP	1514	80 → 5879 [ACK] Seq=1
99759	5707.561090	204.96.59.199	172.16.115.5	TCP	1514	80 → 5879 [ACK] Seq=1
99760	5707.571635	172.16.115.5	204.96.59.199	TCP	60	5879 → 80 [ACK] Seq=3
99761	5707.576397	204.96.59.199	172.16.115.5	TCP	1514	80 → 5879 [ACK] Seq=2
99762	5707.577620	204.96.59.199	172.16.115.5	TCP	1514	80 → 5879 [ACK] Seq=4
99763	5707.578980	204.96.59.199	172.16.115.5	TCP	1514	80 → 5879 [ACK] Seq=5
99764	5707.579248	197.182.91.233	172.16.113.105	SMTP	77	C: EHLO mars.avocado.
99765	5707.581722	172.16.115.5	204.96.59.199	TCP	60	5879 → 80 [ACK] Seq=3
99766	5707.583049	172.16.113.105	197.182.91.233	SMTP	80	S: 500 Command unreco
99767	5707.583439	197.182.91.233	172.16.113.105	SMTP	77	C: HELO mars.avocado.
99768	5707.584287	172.16.113.105	197.182.91.233	SMTP	99	S: 250 (mars.avocado.
99769	5707.584591	197.182.91.233	172.16.113.105	SMTP	92	C: MAIL From:<felinaj
99770	5707.586239	204.96.59.199	172.16.115.5	TCP	1514	80 → 5879 [ACK] Seq=7
99771	5707.586795	204.96.59.199	172.16.115.5	TCP	710	80 → 5879 [PSH, ACK]
99772	5707.586938	172.16.113.105	197.182.91.233	SMTP	99	S: 250 <felinaj@mars.
99773	5707.587050	204.96.59.199	172.16.115.5	HTTP	60	HTTP/1.0 200 OK (tex
99774	5707.587286	197.182.91.233	172.16.113.105	SMTP	95	C: RCPT To:<matthewf@

> Frame 99762: 1514 bytes on wire (12112 bits), 1514 bytes captured (12112 bits)  
> Ethernet II, Src: 3Com\_de:54:36 (00:60:97:de:54:36), Dst: Cisco\_38:46:32 (00:10:7b:38:46:32)  
> Internet Protocol Version 4, Src: 204.96.59.199, Dst: 172.16.115.5  
> Transmission Control Protocol, Src Port: 80, Dst Port: 5879, Seq: 4381, Ack: 303, Len: 1460

0000 00 10 7b 38 46 32 00 60 97 de 54 36 08 00 45 00 ..{BF2...T6..E..  
0010 05 dc 87 ae 00 00 00 06 c6 30 cc 60 3b c7 ac 10 .....@..0';...  
0020 73 05 00 50 16 f7 1b 9f ce 4f f1 bd b2 6e 50 10 s...P....O...nP..  
0030 7f e0 f3 96 00 00 44 45 52 3d 30 20 4e 41 54 55 .....DE R=0 NATU..  
0040 52 41 4c 53 49 5a 45 46 4c 41 47 3d 33 3e 3c 2f RALSIZEF LAG=3></..  
0050 41 3e 3c 42 52 3e 0a 20 09 20 20 20 20 20 20 3c A><BR>...<..  
0060 41 20 48 52 45 46 3d 68 74 74 70 3a 2f 2f 77 69 A.HREF=h ttp://wi..  
0070 72 65 2e 61 70 2e 6f 72 67 2f 3f 46 52 4f 4e 54 re.ap.or g/?FRONT

Fig 4.1 : Snapshot of DARPA99 dataset using Wireshark tool.

Wireshark is a tool used for analyzing data packets sent from one IP address to another. We have used Wireshark to extract out important information from the DARPA 99 dataset and use it for our experiment purpose.

As depicted in above snapshot packets in DARPA 99 contain a timestamp, Source IP, destination IP, Protocol used, Length of the message and some information about. For a particular type of attack the packets with the same type of protocol used are filtered out. For eg : For, TCP SYN attack packets with TCP protocol are filtered out. Wireshark provides an inline command facility to filter out packets.

Here for our experiment we have used Week 1 Day 1 as an attack free traffic as there is no attack on Day 1 Week 1. For detection purposes(testing data) we have used Week 5 Day 1 as it contains a TCP SYN attack initiated once for a duration of 6 minute 51 second and for ICMP Week 4 Day 1 is used it has 2 attacks for 1s each.

**Command for TCP-SYN used in wireshark :**

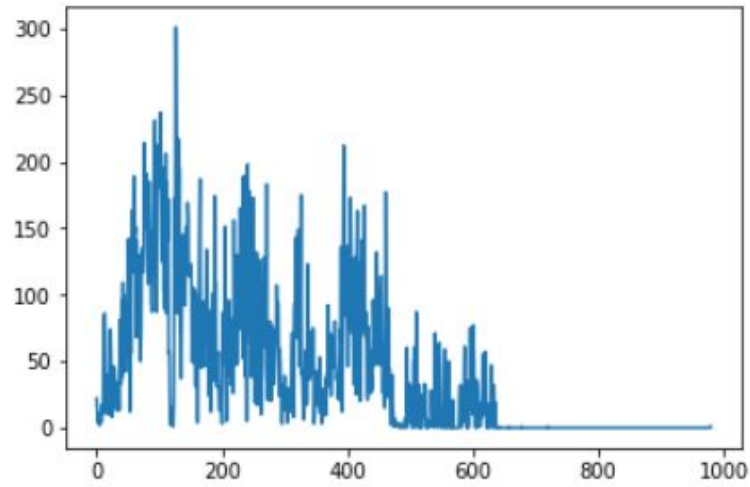
```
tcp.flags.syn==1 && tcp.flags.ack==0
```

**Command for ICMP packets :**

```
icmp.type == 0
```

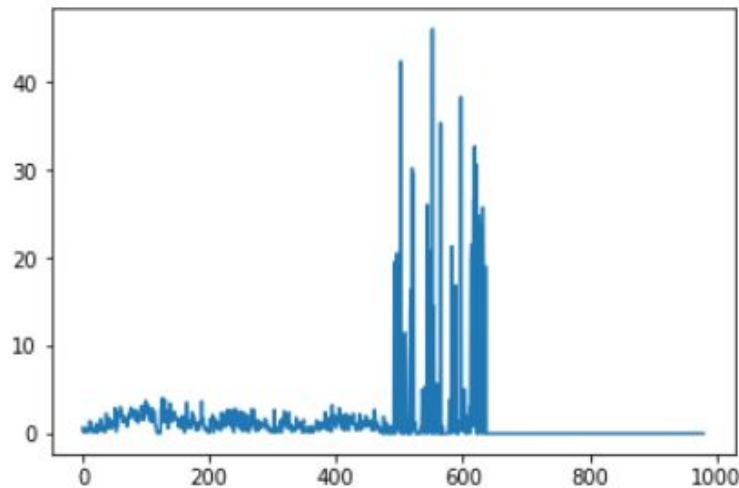
## 4.2 TRAINING DATA FOR TCP-SYN ATTACK

Week 1 Day 1 of DARPA 99 dataset



**Fig 4.2 : First Feature**

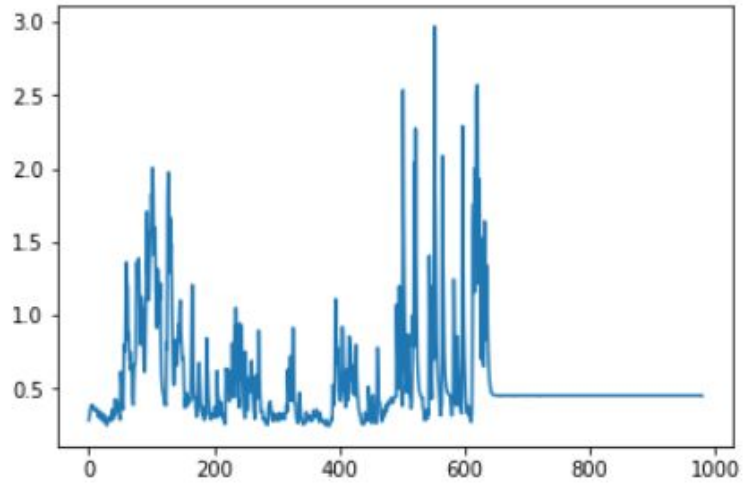
(This figure shows the number of packets passing through the network per 75 second window. Here x-axis is the index of windows and y-axis is the first feature values)



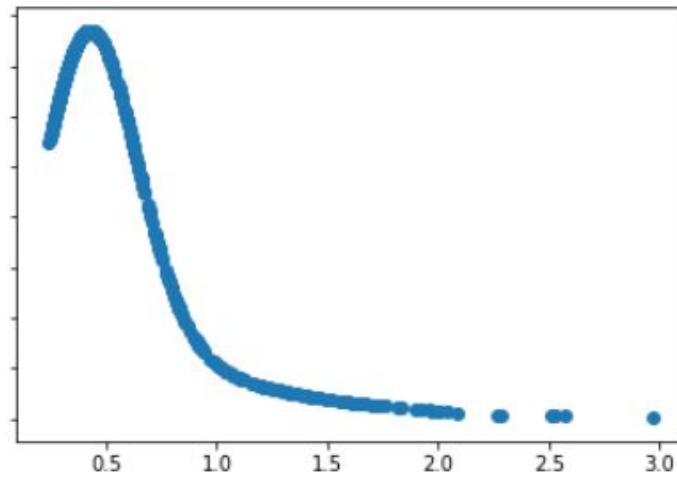
**Fig 4.3 : Second Feature**

(This figure shows an inverse function of average time between the packets within the 75 second window. Here x-axis is the index of windows and y-axis is the second feature values)





**Fig 4.4 : CRPS-ES values for training data after applying PCA**  
 (Here x-axis is the index of windows and y-axis is the CRPS-ES values)

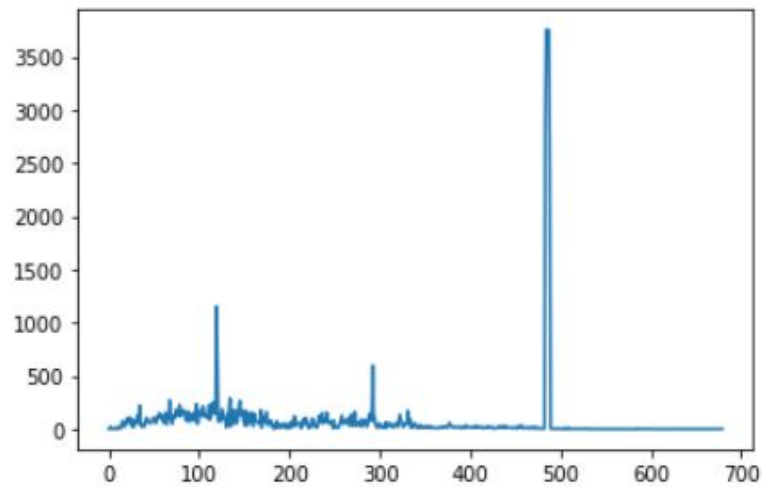


**Fig 4.5 : Distribution of the CRPS-ES calculated using KDE**

This can be observed from the Fig(4.5) that the distribution is left skewed. This means that the trend of the attack free traffic is falling on the left hand side and if the incoming testing data is falling on the right corner or the right edge of the plane then it can be considered as an anomaly to the attack free data. We have taken our threshold as  $(1-\alpha)$ th quantile of this distribution which means that values less than this threshold or the we can say values falling under the graph are considered as normal data.

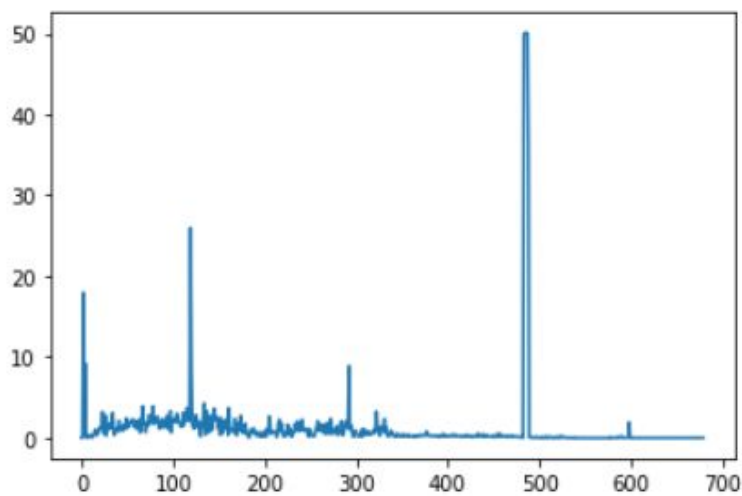
### 4.3 TESTING DATA FOR TCP-SYN ATTACK

Week 5 Day 1 of DARPA 99 dataset (One attack for 6 minute 51 second)



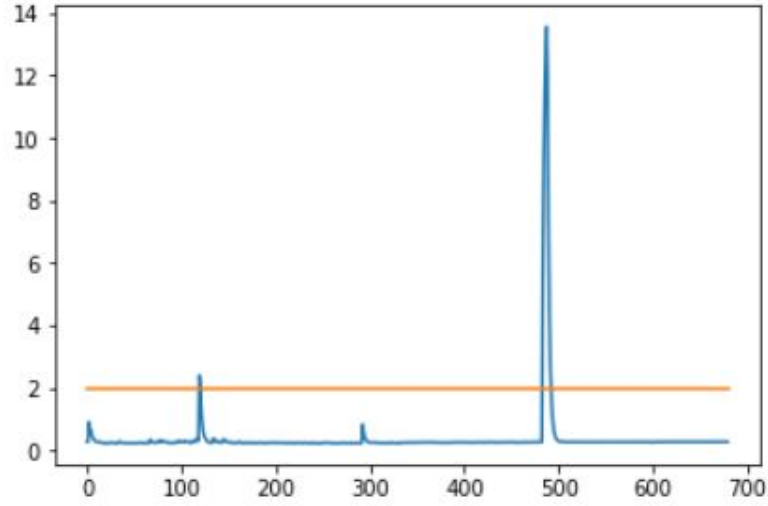
**Fig 4.6 : First Feature**

(This figure shows the number of packets passing through the network per 75 second window. Here x-axis is the index of windows and y-axis is the first feature values.)



**Fig 4.7 : Second Feature**

(This figure shows an inverse function of average time between the packets within the 75 second window. Here x-axis is the index of windows and y-axis is the second feature values)

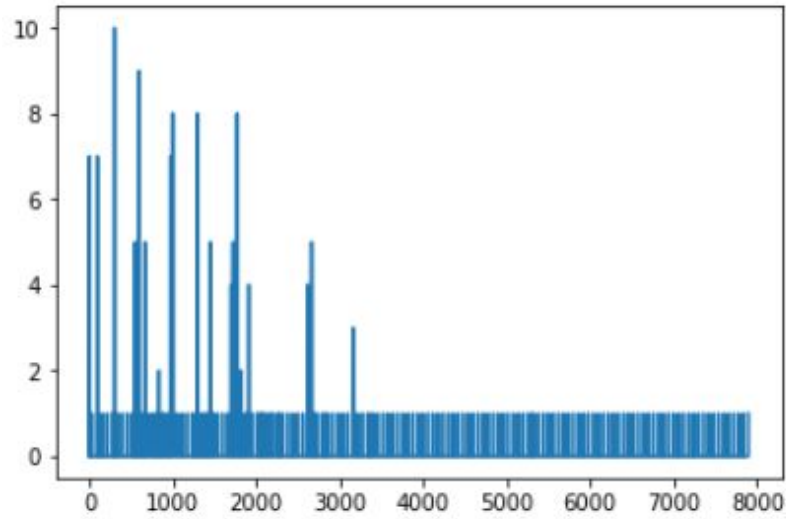


**Fig 4.8 : CRPS-ES values for testing data after applying PCA**

(The orange line represents the threshold CRPS-ES value ( $th = 1.97, (1-\alpha) = 0.99$ ).

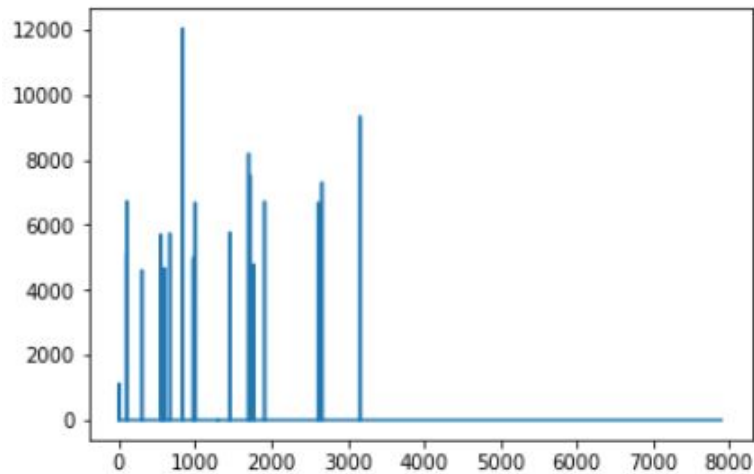
The peaks in the figure are the attack points with CRPS-ES values  $> th$ . It can be observed in the figure that all the attacks are successfully detected by the proposed algorithm. Here x-axis is the index of windows and y-axis is the CRPS-ES values)

## 4.4 TRAINING DATA FOR ICMP ATTACK



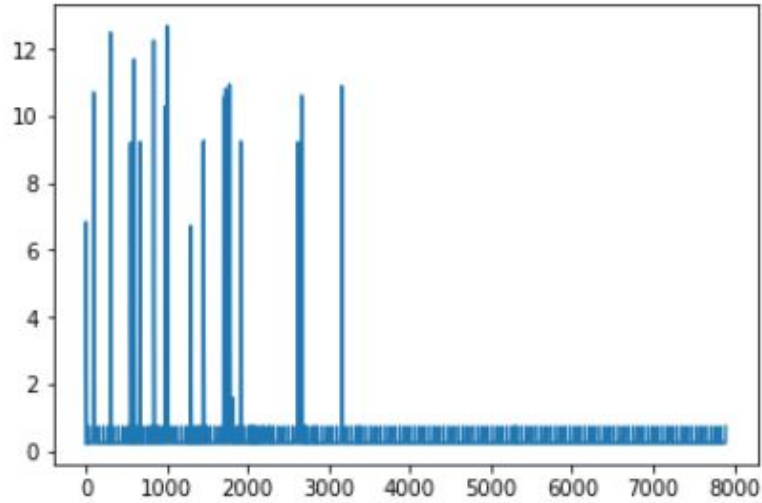
**Fig 4.9 : First Feature**

(This figure shows the number of packets passing through the network per 75 second window. Here x-axis is the index of windows and y-axis is the first feature values)

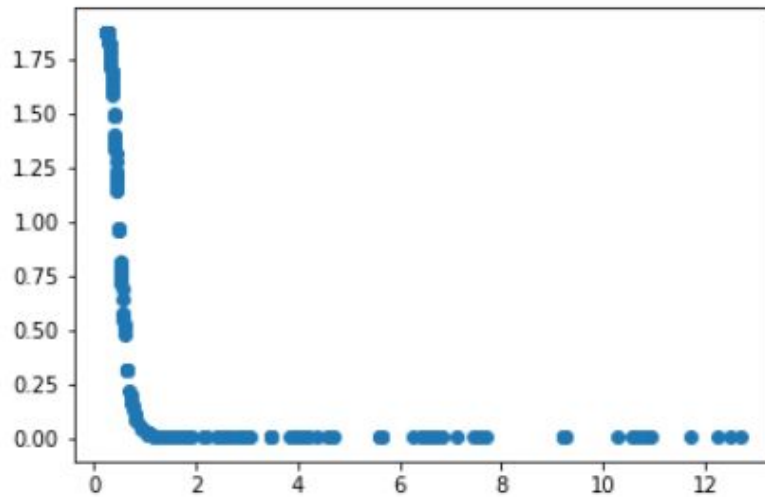


**Fig 4.10 : Second Feature**

(This figure shows an inverse function of average time between the packets within the 75 second window. Here x-axis is the index of windows and y-axis is the second feature values)



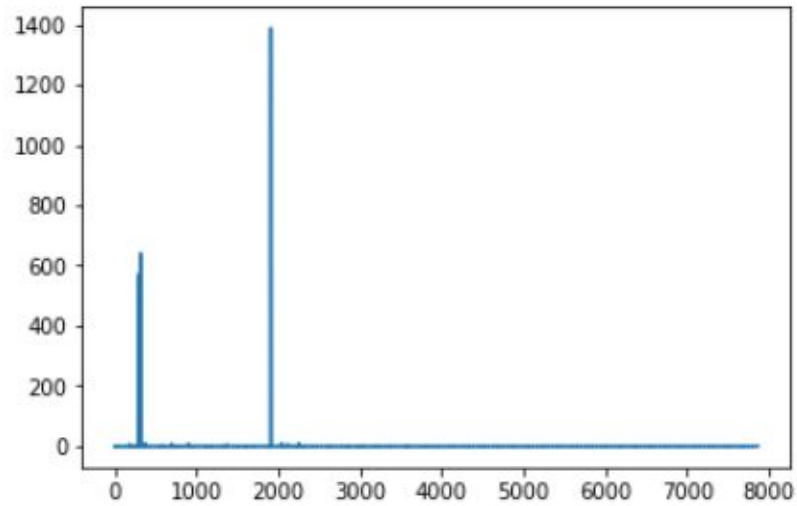
**Fig 4.11: CRPS-ES values for training data after applying PCA**  
(Here x-axis is the index of windows and y-axis is the CRPS-ES values)



**Fig 4.12 : Distribution of the CRPS-ES calculated using KDE**

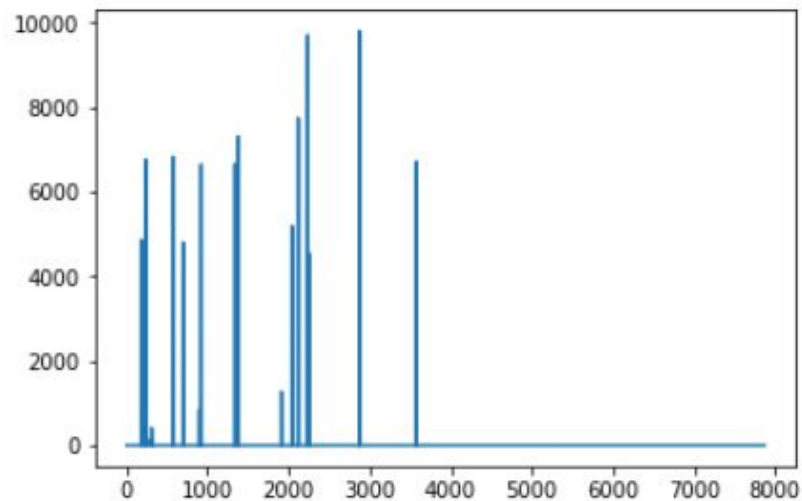
This can be observed from the Fig(4.12) that the distribution is left skewed. This means that the trend of the attack free traffic is falling on the left hand side and if the incoming testing data is falling on the right corner or the right edge of the plane then it can be considered as an anomaly to the attack free data. We have taken our threshold as  $(1-\alpha)$ th quantile of this distribution which means that values less than this threshold or the we can say values falling under the graph are considered as normal data.

## 4.5 TESTING DATA FOR ICMP ATTACK



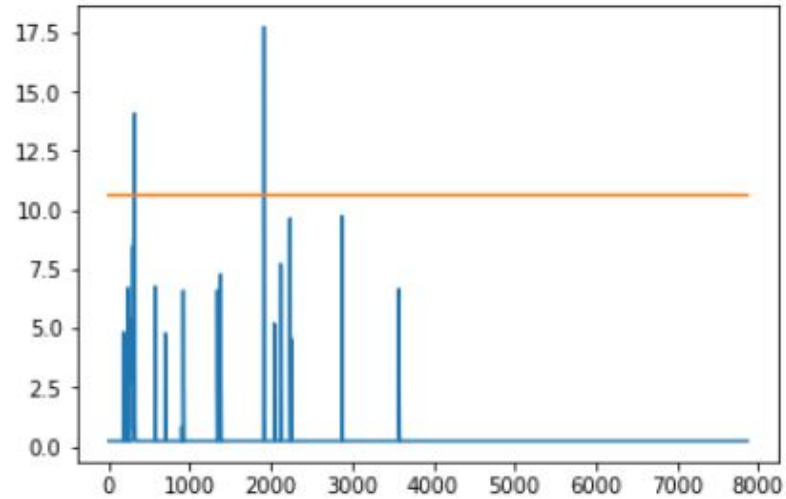
**Fig 4.13 : First Feature**

(This figure shows the number of packets passing through the network per 75 second window. Here x-axis is the index of windows and y-axis is the first feature values)



**Fig 4.14 : Second Feature**

(This figure shows an inverse function of average time between the packets within the 75 second window. Here x-axis is the index of windows and y-axis is the second feature values)



**Fig 4.15 : CRPS-ES values for testing data after applying PCA**

(The orange line represents the threshold CRPS-ES value ( $th = 10.62, (1-\alpha) = 0.99$ ). The peaks in the figure are the attack points with CRPS-ES values  $> th$ . It can be observed in the figure that all the attacks are successfully detected by the proposed algorithm. Here x-axis is the index of windows and y-axis is the CRPS-ES values)

# CHAPTER 5

## 5.1 CONCLUSION

We were able to detect all the TCP-SYN and ICMP attacks present in the dataset used using fast statistical algorithms as compared to deep learning mechanisms which are computationally very expensive and are not feasible for edge and fog nodes. We were also able to show the importance of fog computing in the anomaly detection frameworks as it provides local monitoring of the network traffic which can help us in detection of various attacks at the lowest level. Our aim here was to highlight the importance of multiple features in time series based algorithms. We have demonstrated why we should use multiple features and how we can extract multiple features from the network. Use of multiple features plays an important role in detecting hidden attacks and exposing various unknown trends present in the data.

## 5.2 FUTURE WORKS

In future works, this architecture can be made more accurate and useful by,

- Including more network parameters such as energy consumption, number of source IPs etc, as features in the anomaly detection model.
- Using multivariate CRPS components because this can increase the capability of extracting network features and using them in analysing the data trends by manifolds.
- Identification of the malicious nodes present in the network. In this work we have detected the anomaly present in the data but finding the source of the anomaly data is still a very challenging task that we would like to tackle in our future works.
- Exploring the capabilities of Fog computing and figuring out how we can leverage it to enhance the efficiency of our model to the maximum.



## REFERENCES

1. Stojmenovic, I., Wen, S., Huang, X., and Luan, H. (2016) An overview of Fog computing and its security issues. *Concurrency Computat.: Pract. Exper.*, 28: 2991– 3005. doi: 10.1002/cpe.3485.
2. Shanhe Yi, Cheng Li, and Qun Li. 2015. A Survey of Fog Computing: Concepts, Applications and Issues. In *Proceedings of the 2015 Workshop on Mobile Big Data (Mobidata '15)*. Association for Computing Machinery, New York, NY, USA, 37–42. DOI:<https://doi.org/10.1145/2757384.2757397>
3. V. Hassija, V. Chamola, V. Saxena, D. Jain, P. Goyal and B. Sikdar, "A Survey on IoT Security: Application Areas, Security Threats, and Solution Architectures," in *IEEE Access*, vol. 7, pp. 82721-82743, 2019, doi: 10.1109/ACCESS.2019.2924045.
4. Harrou F, Bouhaddou B, Sun Y, Kadri B (2018) Detecting cyberattacks using a CRPS-based monitoring approach. 2018 IEEE Symposium Series on Computational Intelligence (SSCI). Available: <http://dx.doi.org/10.1109/SSCI.2018.8628797>.
5. Tilmann Gneiting & Adrian E Raftery (2007) Strictly Proper Scoring Rules, Prediction, and Estimation, *Journal of the American Statistical Association*, 102:477, 359-378, DOI: 10.1198/016214506000001437
6. Shanhe Yi, Cheng Li, and Qun Li. 2015. A Survey of Fog Computing: Concepts, Applications and Issues. In *Proceedings of the 2015 Workshop on Mobile Big Data (Mobidata '15)*. Association for Computing Machinery, New York, NY, USA, 37–42. DOI:<https://doi.org/10.1145/2757384.2757397>
7. Yi S., Qin Z., Li Q. (2015) Security and Privacy Issues of Fog Computing: A Survey. In: Xu K., Zhu H. (eds) *Wireless Algorithms, Systems, and Applications. WASA 2015. Lecture Notes in Computer Science*, vol 9204. Springer, Cham. [https://doi.org/10.1007/978-3-319-21837-3\\_67](https://doi.org/10.1007/978-3-319-21837-3_67)

8. Stojmenovic, I., Wen, S., Huang, X., and Luan, H. (2016) An overview of Fog computing and its security issues. *Concurrency Computat.: Pract. Exper.*, 28: 2991– 3005. doi: 10.1002/cpe.3485.
9. L. Feinstein, D. Schnackenberg, R. Balupari and D. Kindred, "Statistical approaches to DDoS attack detection and response," *Proceedings DARPA Information Survivability Conference and Exposition*, Washington, DC, USA, 2003, pp. 303-314 vol.1, doi: 10.1109/DISCEX.2003.1194894.
10. Yonghao Gu, Yongfei Wang, Zhen Yang, Fei Xiong, Yimu Gao, "Multiple-Features-Based Semi Supervised Clustering DDoS Detection Method", *Mathematical Problems in Engineering*, vol. 2017, Article ID 5202836, 10 pages, 2017. <https://doi.org/10.1155/2017/5202836>
11. Sumathi, S., Karthikeyan, N. Detection of distributed denial of service using deep learning neural networks. *J Ambient Intell Human Comput* (2020). <https://doi.org/10.1007/s12652-020-02144-2>
12. Madakam, S., Ramaswamy, R. and Tripathi, S. (2015) Internet of Things (IoT): A Literature Review. *Journal of Computer and Communications*, 3, 164-173. <http://dx.doi.org/10.4236/jcc.2015.35021>
13. Weglarczyk, Stanislaw. (2018). Kernel density estimation and its application. *ITM Web of Conferences*. 23. 00037. 10.1051/itmconf/20182300037.

# PLAGIARISM REPORT

2nd time

## ORIGINALITY REPORT

% <b>9</b>	% <b>3</b>	% <b>5</b>	% <b>4</b>
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

## PRIMARY SOURCES

<b>1</b>	Submitted to Netaji Subhas Institute of Technology Student Paper	% <b>3</b>
<b>2</b>	Benamar Bouyeddou, Benamar Kadri, Fouzi Harrou, Ying Sun. "DDOS-attacks detection using an efficient measurement-based statistical mechanism", Engineering Science and Technology, an International Journal, 2020 Publication	% <b>2</b>
<b>3</b>	blog.trendmicro.com Internet Source	% <b>1</b>
<b>4</b>	S. Sumathi, N. Karthikeyan. "Detection of distributed denial of service using deep learning neural network", Journal of Ambient Intelligence and Humanized Computing, 2020 Publication	<% <b>1</b>
<b>5</b>	Submitted to University of Northumbria at Newcastle Student Paper	<% <b>1</b>
<b>6</b>	"Handbook of Computer Networks and Cyber	

	Security", Springer Science and Business Media LLC, 2020 Publication	<% 1
7	Submitted to Georgia State University Student Paper	<% 1
8	<a href="https://scholarcommons.usf.edu">scholarcommons.usf.edu</a> Internet Source	<% 1
9	"Information and Communication Technology for Intelligent Systems", Springer Science and Business Media LLC, 2021 Publication	<% 1
10	Yonghao Gu, Yongfei Wang, Zhen Yang, Fei Xiong, Yimu Gao. "Multiple-Features-Based Semisupervised Clustering DDoS Detection Method", Mathematical Problems in Engineering, 2017 Publication	<% 1
11	<a href="http://www.hindawi.com">www.hindawi.com</a> Internet Source	<% 1
12	Submitted to University College London Student Paper	<% 1
13	<a href="https://pdfs.semanticscholar.org">pdfs.semanticscholar.org</a> Internet Source	<% 1
14	"Cognitive Computing for Big Data Systems Over IoT", Springer Science and Business	<% 1

## Media LLC, 2018

Publication

15	<a href="http://www.ijitee.org">www.ijitee.org</a> Internet Source	<% 1
16	<a href="https://medium.com">medium.com</a> Internet Source	<% 1
17	<a href="https://link.springer.com">link.springer.com</a> Internet Source	<% 1
18	Mohammad Irfan Bala, Mohammad Ahsan Chishti. "Survey of applications, challenges and opportunities in fog computing", International Journal of Pervasive Computing and Communications, 2019 Publication	<% 1
19	Jyoti Grover, Ashish Jain, Sunita Singhal, Anju Yadav. "Chapter 65 Real-Time VANET Applications Using Fog Computing", Springer Science and Business Media LLC, 2018 Publication	<% 1
20	Rakesh Kumar Jha, Puja, Haneet Kour, Manoj Kumar, Shubha Jain. "Layer based security in Narrow Band Internet of Things (NB-IoT)", Computer Networks, 2020 Publication	<% 1
21	<a href="http://itu.diva-portal.org">itu.diva-portal.org</a> Internet Source	<% 1

22 lists.smc.org.in  
Internet Source

<%1

23 "Internet of Things, Smart Spaces, and Next  
Generation Networks and Systems", Springer  
Science and Business Media LLC, 2019  
Publication

<%1

EXCLUDE QUOTES OFF  
EXCLUDE ON  
BIBLIOGRAPHY

EXCLUDE MATCHES OFF

# APPENDIX

## PCA

Principal Component Analysis, or PCA, is a dimensionality-reduction method that is often used to reduce the dimensionality of large data sets, by transforming a large set of variables into a smaller one that still contains most of the information in the large set.

### STEP 1: STANDARDIZATION

The aim of this step is to standardize the range of the continuous initial variables so that each one of them contributes equally to the analysis.

$$z = \frac{\textit{value} - \textit{mean}}{\textit{standard deviation}}$$

### STEP 2: COVARIANCE MATRIX COMPUTATION

The aim of this step is to understand how the variables of the input data set are varying from the mean with respect to each other, or in other words, to see if there is any relationship between them. Because sometimes, variables are highly correlated in such a way that they contain redundant information. So, in order to identify these correlations, we compute the covariance matrix.

$$\begin{bmatrix} Cov(x, x) & Cov(x, y) & Cov(x, z) \\ Cov(y, x) & Cov(y, y) & Cov(y, z) \\ Cov(z, x) & Cov(z, y) & Cov(z, z) \end{bmatrix}$$

### STEP 3: COMPUTE THE EIGENVECTORS AND EIGENVALUES OF THE COVARIANCE MATRIX TO IDENTIFY THE PRINCIPAL COMPONENTS

Eigenvectors and eigenvalues are the linear algebra concepts that we need to compute from the covariance matrix in order to determine the *principal components* of the data. We have already discussed principal components in section 3.2. always come in pairs, so that every eigenvector has an eigenvalue. And their number is equal to the number of dimensions of the data. For example, for a 3-dimensional data set, there are 3 variables, therefore there are 3 eigenvectors with 3 corresponding eigenvalues. eigenvectors of the Covariance matrix are actually *the directions of the axes where there is the most variance* (most information) and that we call Principal Components. And eigenvalues are simply the coefficients attached to eigenvectors, which give the *amount of variance carried in each Principal Component*.

$$v_1 = \begin{bmatrix} 0.6778736 \\ 0.7351785 \end{bmatrix} \quad \lambda_1 = 1.284028$$
$$v_2 = \begin{bmatrix} -0.7351785 \\ 0.6778736 \end{bmatrix} \quad \lambda_2 = 0.04908323$$

### STEP 4: FEATURE VECTOR

So, the feature vector is simply a matrix that has as columns the eigenvectors of the components that we decide to keep.

### LAST STEP: RECAST THE DATA ALONG THE PRINCIPAL COMPONENTS AXES

The aim is to use the feature vector formed using the eigenvectors of the covariance matrix, to reorient the data from the original axes to the ones represented by the principal components (hence the name Principal Components Analysis). This can be done by multiplying the transpose of the original data set by the transpose of the feature vector.



## KERNEL DENSITY ESTIMATION (KDE)

Kernel Density estimation in statistics is a non-parametric way to estimate the PDF(probability Density Function) of a random variable. The bandwidth of the kernel estimates the smoothness if the bandwidth is small and the estimate may include bumps and reveals little about the underlying distribution if the bandwidth is large.

Let  $(x_1, x_2, \dots, x_n)$  be a univariate independent and identically distributed sample drawn from some distribution with an unknown density  $f$  at any point  $x$ . We try to figure out this function  $f$ . Its KDE is :

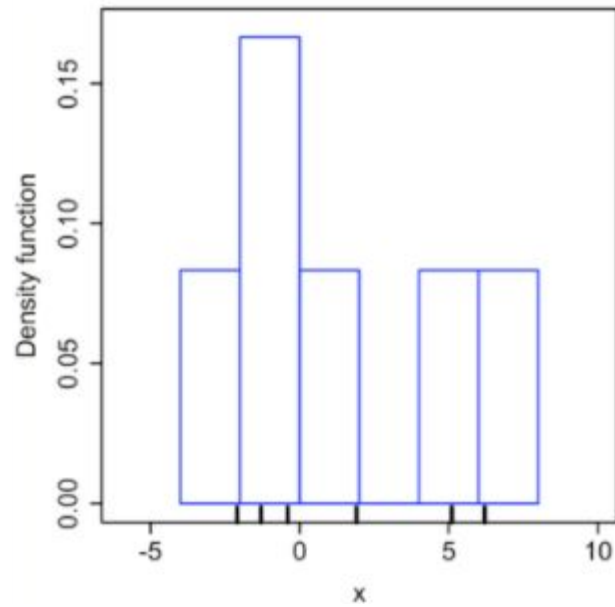
$$\hat{f}_h(x) = \frac{1}{n} \sum_{i=1}^n K_h(x - x_i) = \frac{1}{nh} \sum_{i=1}^n K\left(\frac{x - x_i}{h}\right)$$

Where  $K$  is the kernel which is a non negative function and  $h > 0$  is a smoothing parameter also called the bandwidth. A range of kernel functions is used for the estimation such as uniform,triangular,biweight and triweight etc. Due to its convenient mathematical properties, the normal kernel is often used.

KDE is closely related to histograms. A six point example can be used to understand the difference between Histograms and KDE.

Sample	1	2	3	4	5	6
Value	-2.1	-1.3	-0.4	1.9	5.1	6.2

The below figure demonstrates a histogram, the x-axis is divided into six bins with a width of 2 each and whenever a data point falls in the stack its height is raised by 1/12th of the previous size of the stack. As we can see some stacks have a large height as a large number of points lie in that area.



The below figure demonstrates KDE distribution. In the x axis at every point  $x_i$  a normal curve (red lines) of standard deviation 2.25 is taken as mean. Then the kernel is summed up to create a distribution.

