DOS ATTACK MITIGATION IN IOT

by

Ömer Çel

B.S., Electronics and Communication Engineering, Yıldız Technical University, 2012

Submitted to the Institute for Graduate Studies in
Science and Engineering in partial fulfillment of
the requirements for the degree of
Master of Science

Graduate Program in Electrical Electronics Engineering
Boğaziçi University

2018

ACKNOWLEDGEMENTS

In recent years, IOT applications have entered the world of information rapidly and are becoming more and more active every day. The number of end devices in the IOT has increased exponentially with respect to the previous year, and is becoming as important as safety.

Since it is a new technology, standards have not yet been established in IOT communications and the work on security has been limited. In this study, the attacks on IOT were examined and the perception of these attacks by gateways was studied. Attacks have been realized with the widely used COOJA simulator of the Contiki operating system, which we can run embedded in end devices. From these attacks, due to their statistical properties, the focus is on Hello Flood and Black Hole attacks.

In order to detect attacks, CUSUM, AR Processes, Shannon Entropy, Tsallis Entropy methods have been tried and performance results have been obtained. Besides these algorithms have been run into Raspberry Pi's which are used as an IOT gateway and their behavior has been observed.

ABSTRACT

DOS ATTACK MITIGATION IN IOT

First of all, I would like to thank my thesis advisor, Professor Emin Anarım for his contributions and patience. He was always very helpful and understanding.

Secondly, I want to thank my family for their endless support and motivation during my whole life.

My whole friends have great contributions, especially Göktekin Durusoy, Derya Erhan. I really appreciate their contribution, assistance, friendship.

ÖZET

IOT'DE DOS ATAKLARININ TESPİTİ

Son yıllarda IoT uygulamaları bilişim dünyasına hızla girmiş, her geçen gün daha da aktif olarak kullanılmaya başlanmıştır. IoT'deki uç cihazların sayısı bir önceki yıla nazaran katlanarak artmakta güvenlik olarak da bir o kadar önemli hale gelmektedir.

Yeni bir teknoloji olmasından dolayı henüz IoT haberleşmelerinde standartlar oturmamış ve güvenliğiyle ilgili çalışmalar sınırlı kalmıştır. Bu çalışmada IoT'de yapılan ataklar incelenmiş ve bu atakların ağ geçitleri tarafından algılanması üzerine çalışılmıştır. Uç cihazların içine gömülü olarak çalıştırabileceğimiz Contiki işletim sisteminin oldukça yaygın olarak kullanılan COOJA simulatörü ile ataklar gerçeklenmiştir. Bu ataklardan, istatistiksel özelliklerinden dolayı, Hello Flood ve Black Hole saldırılarının üzerine yoğunlaşılmıştır.

Atakları tesbit etmek için CUSUM, AR Süreçleri, Shannon Entropi, Tsallis Entropi yöntemleri denenmiştir ve performans sonuçları elde edilmiştir.Bunun yan sıra, bu algoritmalar bir IoT ağ geçidi olarak kullanılan Raspberry Pi'ler içine çalıştırılmış ve davranışları gözlemlenmiştir.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS

A	Operator matrix
A_{ip}	IP variables operator matrix
C	Covariance Matrix
c_{ij}	Covariance matrix
$E(x)$	Expected value of x
$f(\varepsilon_1 ... \varepsilon_i)$	Joint probability density function
h	Optimal threshold
H_0	Hypothesis implying a no-change
H_1	Hypothesis implying a change
$H_S(x)$	Shannon Entropy
H_q	Tsallis Entropy
k	Number of parameters estimated by the model
l	Likelihood ratio
L	Likelihood function
N_R	Learning window length
N'_R	Learning window length minus AR order
N_S	Test window length
N'_S	Test window length minus AR order
p	AR order
p_j	Discrete set of probabilities
$p(\varepsilon_{N_R}/\alpha_p)$	Joint likelihood of the residual time series
q	Entropic Parameter
$R(t)$	Learning window
$r_i(t)$	Elements of learning window
$\tilde{r}_i(t)$	Estimation of $r_i(t)$
$r(i)$	Elements of covariance matrix of Yule Walker
$r_x(m)$	Autocorrelation coefficient
R_n	Covariance matrix of Yule Walker

S_j^k	Likelihood ratio for observations from y_i until y_j
s_i	Increment of S_j^k
v	Magnitude of change
y_i	Observations
α	Vector of AR Coefficients
α_R	AR coefficients learning window segment
α_S	AR coefficients test window segment
ε_i	Residual error
σ^2	Variance of segment
σ_R^2	Variance of learning window segment
σ_S^2	Variance of test window segment
σ_P^2	Variance of pooled window segment
η	Likelihood ratio
ψ	Abnormality vector
ϕ	AR coefficient
θ_0	Parameter before change
θ_1	Parameter after change
$\Lambda_1^k(j)$	Likelihood ratio for observations from y_i until y_j
μ	Mean of the unit
$\rho_{\psi_1\psi_2}$	Correlation Coefficient

LIST OF ACRONYMS/ABBREVIATIONS

UDP	User Datagram Protocol
FPR	False Positive Rate
TPR	True Positive Rate
ROC	Receiver Operating Characteristic
DOS	Denial of Service
DDOS	Distributed Denial of Service
IoT	Internet of Things
CUSUM	Cumulative Sum
CPU	Central Processing Unit
CPM	Change-Point Monitoring
IP	Internet Protocol
SYN	Synchronize
SYN-ACK	Synchronize-Acknowledge
TCP	Transmission Control Protocol
ICMP	Internet Control Message Protocol
ICMPv6	Internet Control Message Protocol Version Six
SNMP	Simple Network Management Protocol
TP	True Positive
FN	False Negative
FP	False Positive
TN	True Negative
AR	Autoregressive
MIB	Management Information Base
AIC	Akaike Information Criterion
BIC	Bayesian Information Criterion
RFID	Radio Frequency Identification
WSN	Wireless Sensor Network

1. INTRODUCTION

In the data transmission system, transmitted data is encrypted in a complicated manner. So the interception and deciphering processes are very difficult. Because of these difficulties, attackers change their interest to blocking the network in order to generate delay in service time or they prevent the normal transmission of the data easily.

With growth of IoT, as sensors and mobile devices started to be deployed in health systems and military systems, security have become bigger concern since data become more critical. As stated in [2] and [3] "the mobile edge computing market will be worth 838.6 million USD by 2022" and "edge analytics market will be worth 7.96 Billion USD by 2021". For that reason, it is crucial to detect abnormality in the edge side.

In the following situations, DOS attack may be present in the IoT network :

- Slow network performance unexpectedly
- Unavailability of a particular node

Most common idea is having encryption algorithm over IoT network, however, having a decent firewall should be implemented since attackers may obtain keys.

1.1. Network Security

"To obtain a basic understanding of the performance and behavior of these complex networks, huge amounts of information need to be collected and processed. Often, network performance information is not directly available, and the information obtained must be synthesized to obtain an understanding of the ensemble behavior." [4]

"Using some basic knowledge of the network layout as well as the traffic characteristics at the individual nodes, it is possible to detect network anomalies and performance bottlenecks. The detection of these events can then be used to trigger alarms to the network management system." [4]

2. ANOMALY DETECTION METHODS

2.1. Rule Based Models

"Early work in the area of fault or anomaly detection was based on expert systems. In expert systems, an exhaustive database containing the rules of behavior of the faulty system is used to determine if a fault occurred. Rule-based systems are too slow for real-time applications and are dependent on prior knowledge about the fault conditions on the network" [4].In addition, defining the relevant criteria for different faults will require a set of rules to be developed. In addition, the use of any functional approach scheme, such as back propagation, leads to an increase in computation time and complexity. The number of functions to be learned increases with the number of errors examined [4].

2.2. Pattern Matching Models

"The efficiency of this pattern matching approach depends on the accuracy of the traffic profile generated. Given a new network, it may be necessary to spend a considerable amount of time building traffic profiles. In the face of evolving network topologies and traffic conditions, this method may not scale gracefully" [4].

2.3. Finite State Machines

"Anomaly or fault detection using finite state machines model alarm sequences that occur during and prior to fault events. A probabilistic finite state machine model is built for a known network fault using history data. State machines are designed with the intention of not just detecting an anomaly but also possibly identifying and diagnosing the problem." [4]

2.4. Change Point Detection

"Wang *et al.* present a simple mechanism, called Change-Point Monitoring (CPM), to detect denial of service (DoS) attacks. The core of CPM is based on the inherent network protocol behaviors and is an instance of the Sequential Change Point Detection. To make the detection mechanism insensitive to sites and traffic patterns, a nonparametric Cumulative Sum (CUSUM) method was applied, thus making the detection mechanism robust, more generally applicable, and its deployment much easier." [5]

"To make the detection mechanism insensitive to site and access pattern, a nonparametric Cumulative Sum (CUSUM) method [6] is applied, thus making the detection mechanism much more generally applicable and its deployment much easier." [5]

3. DENIAL OF SERVICE ATTACKS

DOS attacks may create problems not only on the victim side but also on the peripheral side which are in the network branches due to the excessive usage of the bandwidth of a router. [7]

In most cases, the attacker changes his/her own IP address to the other innocent user's IP address which is called IP address spoofing. So identifying the location of the attacker and preventing the network system from the attacker is not easy.

One remark is the difference between the distributed denial-of-service(DDOS) attacks and denial-of-service attacks. The main difference is that in first case, attacks are sent by two or more people, or bots, but in the second case attacks are sent by one person or system. Therefore in distributed denial-of-service attacks, the attacker rules the multiple users which are called as masters and masters also rule multiple innocent users which are called as zombies. So the attack type in distributed-DOS is more complicated than the simple DOS attack case. These type of attacks also consume system resources such as bandwidth on a much higher level compared to the one-to-one DOS attack case [8]. Especially, distributed DOS attacks generates much higher traffic on the targeted system because many number of intended or unintended users consume the resources of the server. The another advantage of using multiple machines in the attack from the view of the attacker is that the harder to turn off the his/her machine. When a server is overloaded with connections, new connections can no longer be accepted.

Under normal conditions, firstly the client sends a SYN(synchronize) message to the server so as to start a TCP connection to a server. Then the server transmits the SYN-ACK(synchronize-acknowledge) message back to the client and finally, the client responds with an ACK(acknowledge) message which means the connection is established. This process is called as the TCP three-way handshake.

The SYN flood attack takes advantage of the TCP three-way handshake. In the SYN flood attack scenario, the attacker sends a synchronize(SYN) message with a spoofed IP address or in other words with a forged sender address. The server tries to send back a SYN-ACK request. However, this SYN-ACK request does not get a response from the attacker which means that the server never obtains the client's ACK request. Therefore resources of the server are left half-open.

In this type of attack, a flood of TCP/SYN packets are sent by the attacker. The server waits for an ACK packet for some time from the client which never comes. So by sending many TCP/SYN packets, the server can become disfunctional. At that point, the server cannot connect to any clients whether they are legitimate or otherwise. Therefore this method prevents the access to the server from the legitimate clients which is the one aim of the DOS attacks.

This problem in the three-way handshake has been used for attacking for years, just like ping of death attack.

Third type of DOS attack is the Internet Control Message Protocol (ICMP) flood attack. In these type of attacks, the attackers sends large number of IP packets with a forged source address which appears on the address of the victim. Generally victim is server. So the bandwidth of the network is quickly used and this causes to prevent the legitimate packets from the server[5].

4. SNMP PROTOCOL

SNMP(Simple Network Management Protocol) is a simple application layer protocol that helps the network administrator to manage the network as the name implies. Essentially, it is designed to simplify the management and control of devices in large networks.The basic principle of SNMP is to send a request and reply to a request that is sent. The UDP protocol is used in the Transport layer. That is, it is not checked whether the submitted information has reached address. The advantage of this is that data transfer is fast because it will avoid unnecessary packet traffic. The network administrator sends the information that the agent application needs to the device it is working with, while the device sends the requested information to the network manager through the manager application.The methods used when these processes are carried out are:

- GET: It is a request in order to get information in devices.
- $GETNEXT$: It is obtained information after GET command.
- SET: Command sent for changes to the device controlled by the administrator.
- $TRAP$: This command is used to notify the manager of the information change automatically when there is a change in the device the agent is running. Unlike the others in this process, no requests are sent from the management system. When a change is detected on the device, a package manager reporting the change is sent without a request from the administrator.
- $GETBULK$: This request is used to display large-scale information more efficiently. So, with this operation, it is possible to have more than one column image on the same table.
- $INFORM$: It is used from one management system to inform other management system.

SNMP also has different versions. These are three, SNMPv1, SNMPv2 and SNMPv3. The operations described above are not used by all versions.

5. DENIAL OF SERVICE ATTACKS IN IOT

5.1. Sybil Attack

In the Sybil attack, the attacker imitates the identities of other nodes in P2MP networks, causing topology and communication to deteriorate. [9]

5.2. Wormhole Attack

"The Wormhole attack is produced by at least two malicious nodes that communicate with each other through a different frequency than the network in which they operate so that communication between them still discreet relative to other nodes." [10] "In Body Area Network, sensors are used to monitor the humans' activities and their actions like health parameters of patients. Therefore, it is essential to secure the privacy of the user's information which is collected by the sensors from the body of the user." [10]

5.3. Sinkhole Attack

"In a Sinkhole attack, the node tries to attract to it the most possible routes to control over most of the data following through the network. The attacker must appear to others as being very attractive by presenting optimal routes." [9] "A thief can put a sensor in the neighborhood of the supermarket, that tells sensors in the supermarket that it is closer to the sink (i.e.,sinkhole attack)." [11]

5.4. Selective Forwarding Attack

Selective Forwarding (Selective transmission) attack occurs when the malicious node tries to route all packets transmitted to a node to remove one of these packages, either randomly or according the importance of data contained in this package. The attacker must have a complete idea of the content of the data following through the

network. [10] “A thief can put a sensor in the neighborhood of the supermarket, that tells sensors in the supermarket that it is closer to the sink (i.e.,sinkhole attack). Hence, all their messages destined to the sink, especially alarms, will be routed through that sensor, which will discard alarm messages (i.e.,selective forwarding attack) and let the thief enter the supermarket.” [11] “In military applications, selective forwarding attacks destroy the transmission packets between the source and base station, and sometimes between the sensor nodes. Malicious nodes refuse to transfer an entire packet. It drops the sensitive information and then forwards the remaining packet.” [12]

5.5. Black Hole Attack

It is variation of Selective Forwarding Attack. Adversary node drops incoming packets. It presents itself as Sink (Base Node). “In military applications, selective forwarding attacks destroy the transmission packets between the source and base station, and sometimes between the sensor nodes. Malicious nodes refuse to transfer an entire packet. It drops the sensitive information and then forwards the remaining packet.” [12]

5.6. Hello Flood Attack

Since sensors and gateways resource constraint devices , with Hello Flood Attack one can easily consume bandwidth and consume their energy. For instance , burglar can consume battery of sensor and get into victims’ house or malicious people may harm sensors that is used in transportation.

Onat and Miri [13] proposed anomaly detection method using a statistical model of nodes’ neighbors used to detect attacks such as node imitation and resource depletion changes. However experimental setup and simulator are not mentioned.

The same authors bring forward a quite similar detection method in [14]. Standard deviation and mean is used as evaluation metrics. However no information is given about experimental setup and simulation.

5.7. Replay Attack

"For instance, an IP temperature sensor located in a remote place can be easily replaced by a computer to obtain illegal information and to launch an attack (e.g. replay attack). Since sensors usually have low (or no) computational power, it is unrealistic to apply encryption techniques, a more suitable approach is to authenticate each sensor and its data." [15]

5.8. ROC Curve Analysis of Detection

Performance evaluation is made by calculating true positive rate (TPR) and false positive rate (FPR).

$$TPR = \frac{TP}{FN + TP} \tag{5.1}$$

TP's account for the estimations are correct and FN's are the cases that although there is an attack but they couldn't be detected.

$$FPR = \frac{FP}{TN + FP} \tag{5.2}$$

FP's are misleading estimates that are considered as attack adversely they are not. TN's are the samples that are defined as legitimate traffic.

6. METHODOLOGY

6.1. AR(1) Estimate

In statistical analysis, an abnormality in the network is modeled as a sudden change in the network of data. Sudden change is defined as any change in the parameters of the time series that occurs with respect to the order of the sampling period. Sudden changes, in other words, abrupt changes in time series data can be modeled using an autoregressive (AR) process [4]. Thottan *et al.* have proposed a method which can detect abrupt changes based on AR(1).

Instead of fetching data using SNMP protocol, variables are obtained through Wireshark as it is mentioned in 6.3.6 Workflow.

In Cooja it is provided only ICMPv6 and UDP packets. ICMPv6 packets in order to define topology and UDP packets as transfer protocol are used. For that reason, they are used as MIB variables.

Due to number of packets in the MIB variables are obtained 1000 ms. In other words, sampling rate was chosen as 1000 ms in each MIB variable. 10 time lags piece wise stationary windows has taken in MIB data. First order AR process was built on them. As it shown in the Figure 6.1, to obtain less correlated residuals, non overlapping windows were used.

As it is mentioned in the section Yule Walker Method and hyphothesis test based on the Generalized Likelihood Ratio Test, variances were estimated using them .

Furthermore, on using Generalized Likelihood Ratio Test and Yule Walker estimations for variance terms, we find;

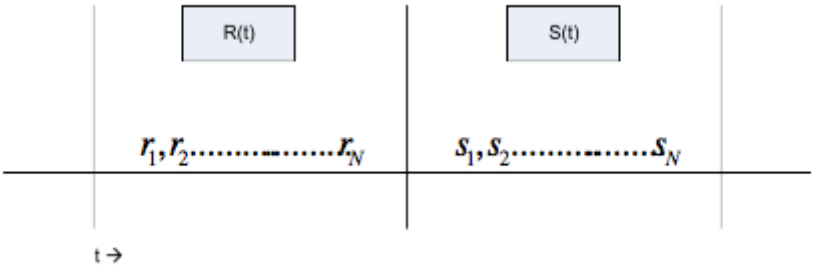

Figure 6.1. Piecewise stationary windows.

$$\eta = \frac{\hat{\sigma}_R^{-N_R}\hat{\sigma}_S^{-N_S}}{\hat{\sigma}_R^{-N_R}\hat{\sigma}_S^{-N_S} + \hat{\sigma}_P^{-N_P}} \tag{6.1}$$

This calculation was made in order to find abnormality vector which is defined;

$$\vec{\psi(t)} = [\eta_1\eta_2....\eta_n] \tag{6.2}$$

Lastly, we need to combine abnormality vectors with operator A matrix and this led to create health vector. In quantum mechanics such operators are frequently used.

$$f(\vec{\psi(t)}) = \vec{\psi(t)}A\vec{\psi(t)}' \tag{6.3}$$

6.1.1. Yule Walker Method

Yule Walker Methods will be used in order to estimate AR parameters. Learning window variance, test window variance and pooled window variance terms σ_R^2, σ_S^2, σ_P^2 are estimated as follows. We define AR(p) model as, $X_t = \sum_{i=1}^{p} \phi_i X_{t-i} + \epsilon$.

Expectation operator is used as,

$$\begin{aligned} &E(X_t X_{t-1}) = \sum_{i=1}^{p} \phi_i X_{t-i} X_{t-1} + \epsilon X_{t-1} \\ &or \\ &R_x(1) = \textstyle\sum_{i=1}^{p} \phi_i R_{i-1} \end{aligned} \tag{6.4}$$

Firstly, writing all the equations together yields;

$$\begin{aligned} r_1 &= \phi_1 r_0 + \phi_2 r_1 + \phi_3 r_2 + ... + \phi_{p-1} r_{p-2} + \phi_p r_{p-1} \\ r_2 &= \phi_1 r_1 + \phi_2 r_0 + \phi_3 r_1 + ... + \phi_{p-1} r_{p-3} + \phi_p r_{p-2} \\ &. \\ &. \\ &. \\ r_{p-1} &= \phi_1 r_{p-2} + \phi_2 r_{p-3} + \phi_3 r_{p-4} + ... + \phi_{p-1} r_0 + \phi_p r_1 \\ r_p &= \phi_1 r_{p-1} + \phi_2 r_{p-2} + \phi_3 r_{p-3} + ... + \phi_{p-1} r_1 + \phi_p r_0 \end{aligned} \tag{6.5}$$

For simplification equation above can be rewritten as,

$$\begin{bmatrix} r_1 \\ r_2 \\ . \\ . \\ . \\ r_{p-1} \\ r_p \end{bmatrix} = \begin{bmatrix} r_0 & r_1 & r_2 & ... & r_{p-2} & r_{p-1} \\ r_1 & r_0 & r_1 & ... & r_{p-3} & r_{p-2} \\ & . & & & & \\ & . & & & & \\ & . & & & & \\ r_{p-2} & r_{p-3} & r_{p-4} & ... & r_0 & r_1 \\ r_{p-1} & r_{p-2} & r_{p-3} & ... & r_1 & r_0 \end{bmatrix} \begin{bmatrix} \phi_1 \\ \phi_1 \\ . \\ . \\ . \\ \phi_{p-1} \\ \phi_p \end{bmatrix} \tag{6.6}$$

Since we already know autocorrelation coefficient of the process is,

$$r_x(m) = \frac{R_x(m)}{R_x(0)} = \phi_{1,1}^m \tag{6.7}$$

The above equation is also,

$$\begin{bmatrix} r_1 \\ r_2 \\ . \\ . \\ . \\ r_{p-1} \\ r_p \end{bmatrix} = \begin{bmatrix} 1 & r_1 & r_2 & ... & r_{p-2} & r_{p-1} \\ r_1 & 1 & r_1 & ... & r_{p-3} & r_{p-2} \\ & . & & & & \\ & . & & & & \\ & . & & & & \\ r_{p-2} & r_{p-3} & r_{p-4} & ... & 1 & r_1 \\ r_{p-1} & r_{p-2} & r_{p-3} & ... & r_1 & 1 \end{bmatrix} \begin{bmatrix} \phi_1 \\ \phi_1 \\ . \\ . \\ . \\ \phi_{p-1} \\ \phi_p \end{bmatrix} \tag{6.8}$$

or succinctly

$$R\phi = r \tag{6.9}$$

Once θ was found σ^2 can be found from the equations below.

$$\begin{aligned} &E(X_t X_t) = \sum_{i=1}^{p} \phi_i X_{t-i} X_t + \epsilon X_t \\ &or \\ &R_x(0) = \textstyle\sum_{i=1}^{p} \phi_i R_i + \sigma_N^2, \\ &\sigma_N^2 = R_x(0) - \textstyle\sum_{i=1}^{p} \phi_i R_i \end{aligned} \tag{6.10}$$

6.1.2. Generalized Likelihood Ratio Test

Firstly we define,

$$R(t) = \{r_1(t), r_2(t), ..., r_{N_R}(t)\} \tag{6.11}$$

Here we can specify any $r_i(t)$ as $\tilde{r}_i(t)$ where $\tilde{r}_i(t) = r_i(t) - \mu$ and the μ is the mean of segment of $R(t)$. As an AR order p process with residual error ε_i, $\tilde{r}_i(t)$ can be estimated ;

$$\varepsilon_i(t) = \sum_{k=0}^{p} a_k \tilde{r}(t-k) \tag{6.12}$$

AR coefficients are $a_R = \{a_1, a_2....a_p\}$ and $\varepsilon_i(t)$ is assumed to be white noise. The joint probability density function of is given as [16],

$$f(\varepsilon_1, \varepsilon_2, ..., \varepsilon_i) = (2\pi\sigma)^{\frac{-N}{2}} exp\{-\frac{1}{2\sigma^2}\sum_{i=1}^{N}\varepsilon_t^2\} \tag{6.13}$$

Changing the ε_i in (6.13) and by the use of the $x_{1-p}, x_{2-p}, ..., x_0$ the likelihood function L of the $\{\varepsilon_i(t)\}$ is

$$L = (2\pi\sigma)^{\frac{-N}{2}} exp\{-\frac{1}{2\sigma^2}\sum_{t=1}^{N}(\sum_{i=0}^{p} a_i X_{t-i})^2\} \tag{6.14}$$

Which can restated as,

$$L = (2\pi\sigma)^{\frac{-N}{2}} exp\{-\frac{1}{2\sigma^2} Na'Ca\} \tag{6.15}$$

The column vector $a' = [1, a_1, ..., a_p]$ of a and $C = C_{ij}$ is the $(p+1)x(p+1)$ matrix of covariance given by,

$$c_{ij} = \frac{1}{N}\sum_{t=1}^{N} X_{t-i}X_{t-j} \quad i, j = 0, 1, ..., p \tag{6.16}$$

In order to acquire the maximum-likelihood estimates of σ^2 and a, the L must be maximized with respect to σ^2 and a. This leads to the estimates σ^2 and a where $\tilde{\sigma}$ (covariance estimate) where Peter *et al.* show,

$$\tilde{\sigma}^2 = a'Ca \tag{6.17}$$

If we go back to the notation, error is $N(0, \sigma_R)$ distribution. Joint likelihood of the residual error terms was acquired as;

$$f(\varepsilon_1, ..., \varepsilon_i / a_1, ..., a_p) = \left(\frac{1}{\sqrt{2\pi\sigma_R}} \right)^{N'_R} exp\left\{ - \frac{N'_R \tilde{\sigma}^2_R}{2\sigma^2_R} \right\} \tag{6.18}$$

Where σ^2_R is the error variance in section $R(t)$ and $N'_R = N_R - p$ and $\tilde{\sigma}^2_R$ is the covariance estimate of σ^2_R [17]. The residuals $R(t)$ and $S(t)$, their joint likelihood L ;

$$l = \left(\frac{1}{\sqrt{2\pi\sigma_R}} \right)^{N'_R} \left(\frac{1}{\sqrt{2\pi\sigma_S}} \right)^{N'_S} exp\left\{ - \frac{N'_R \tilde{\sigma}^2_R}{2\sigma^2_R} \right\} exp\left\{ - \frac{N'_S \tilde{\sigma}^2_S}{2\sigma^2_S} \right\} \tag{6.19}$$

Variance of the residual σ^2_S and $N'_S = N_S - p$ in the segment $S(t)$ two hypotheses are H_0 intimating that no change, H_1 implying a change. Under the hypothesis H_0 we have; $a_R = a_S$ and $\sigma^2_R = \sigma^2_S$ where σ^2_P is pooled variance.

$$l_P = \left(\frac{1}{\sqrt{2\pi\sigma_P^2}}\right)^{N'_R+N'_R} exp\left(\frac{-(N'_R+N'_S)\tilde{\sigma}_p^2}{2\sigma_p^2}\right) \tag{6.20}$$

We have; $a_R \neq a_S$ and $\sigma_R^2 \neq \sigma_S^2$ under the hypothesis H_1 and $a_R = a_S$ and $\sigma_R^2 = \sigma_S^2 = \sigma_P^2$ under the hypothesis H_0, thus using conditions we obtained the likelihood ratio as;

$$l = \sigma_p^{-(N'_R+N'_S)} \sigma_R^{N'_R} \sigma_S^{N'_S} exp\left(\frac{-\tilde{\sigma}_p^2(N'_R+N'_S)}{2\sigma_P^2} + \frac{1}{2}\left[\frac{N'_R\tilde{\sigma}_R^2}{\sigma_R^2} + \frac{N'_S\tilde{\sigma}_S^2}{\sigma_S^2}\right]\right) \tag{6.21}$$

Additionally on using the maximum likelihood estimates for the variance terms, the log likelihood ratio is obtained;

$$-lnl = N'_R(ln\tilde{\sigma}_P - ln\tilde{\sigma}_R) + N'_S(ln\tilde{\sigma}_P - ln\tilde{\sigma}_S) \tag{6.22}$$

When optimal threshold chosen "h'' and is compared with log likelihood ratio $-lnl$, whether exceeds the threshold "h'',that is defined;

$$\begin{aligned} -lnl > h &=> H_1\, change, \\ -lnl \leq h &=> H_0\, no\, change \end{aligned} \tag{6.23}$$

Lastly, on using the maximum likelihood estimates for the error variance terms in equations (6.20) and (6.21) we get;

$$\eta = \frac{\hat{\sigma}_R^{-N_R}\hat{\sigma}_S^{-N_S}}{\hat{\sigma}_R^{-N_R}\hat{\sigma}_S^{-N_S} + \hat{\sigma}_P^{-N_P}} \tag{6.24}$$

6.1.3. Design of the Operator A matrix

There was a different operator A matrix definition in [4] which is based on abnormality vector ψ and we call this matrix $A2$ type Operator Matrix in Results. A_{ip} defined as;

$$\begin{aligned} A_{ip} &= | < \psi_1(t), \psi_2(t) > | \\ A_{ip} &= \tfrac{1}{T} \textstyle\sum_{t=1}^{T} |\psi_1(t), \psi_2(t)| \end{aligned} \tag{6.25}$$

In our approach, we also tried correlation coefficients. Correlation coefficients indicate linear relationship between random variables. The value of ρ is always between *+1* and *-1*.

$$\rho_{\psi_1\psi_2} = \frac{Cov(\psi_1, \psi_2)}{\sigma_{\psi_1,\psi_2}} \tag{6.26}$$

Since we have two MIB variables, A matrix is created *2 x 2*. In order to normalize the matrix we need to set normalization constant K. Temporary A matrix is defined

as,

$$A_t = \begin{bmatrix} \rho_{\psi_1\psi_1} & \rho_{\psi_1\psi_2} \\ \rho_{\psi_2\psi_1} & \rho_{\psi_2\psi_2} \end{bmatrix} or \begin{bmatrix} 1 & \rho_{\psi_1\psi_2} \\ \rho_{\psi_2\psi_1} & 1 \end{bmatrix} \tag{6.27}$$

For normalization,

$$\begin{bmatrix} \rho_{\psi_1\psi_1} & \rho_{\psi_1\psi_2} \\ \rho_{\psi_2\psi_1} & \rho_{\psi_2\psi_2} \end{bmatrix} . \begin{bmatrix} K \\ K \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \tag{6.28}$$

K constant can be obtained,

$$k = A_t^{-1} * j \tag{6.29}$$

and finally A matrix can be found,

$$A = A_t * k \tag{6.30}$$

6.1.4. Akaike Test and Bayesian Information Criterion

Since we want to model our IoT data as AR process, we would like to know best resembling model. Akaike Information Criterion , which achieves this goal by providing an asymptotically unbiased estimate of the "distance" (actually, Kullback-

Leibler information) between the various fitted AR models and the truth, proposed by Akaike 1973 . The AIC defined in general as;

$$AIC = -2\tilde{L} + 2k \tag{6.31}$$

where log likelihood is maximized (Gaussian) log likelihood and k is our number of observations. For an AR(p) model fitted by the Yule Walker method one can be used;

$$AIC = n(log\tilde{\sigma}^2 + 1) + 2(p + 1) \tag{6.32}$$

We choose p to minimize our AIC, in this case AIC leads us to find best approximating model.

Bayesian Information Criterion is closely related to the Akaike Test. It is defined as [18],

$$BIC = ln(n)k - 2ln\tilde{L} \tag{6.33}$$

- $\tilde{L}$ = maximized value of likelihood function,
- x the observed data
- n the number of observed data
- k the number of parameters estimated by the model.

Under the assumption that ϵ_i (error terms) are independent and identically distributed BIC defined as,

$$BIC = n.ln\tilde{\sigma_\epsilon}^2 + k.ln(n) \tag{6.34}$$

As it is showed in the table 6.1 our data fitted AR(5) due to AIC and AR(3) for BIC since they reached the minimum depending on sample size n also means whole data.

Table 6.1. AIC and BIC Results

AR(p)	**AIC**	**BIC**
1	3389.809	2408.607
2	3273.679	2297.375
3	3231.067	2259.662
4	3226.251	2259.745
5	3224.339	2262.731
6	3227.639	2270.930
7	3231.263	2279.453
8	3233.944	2287.032

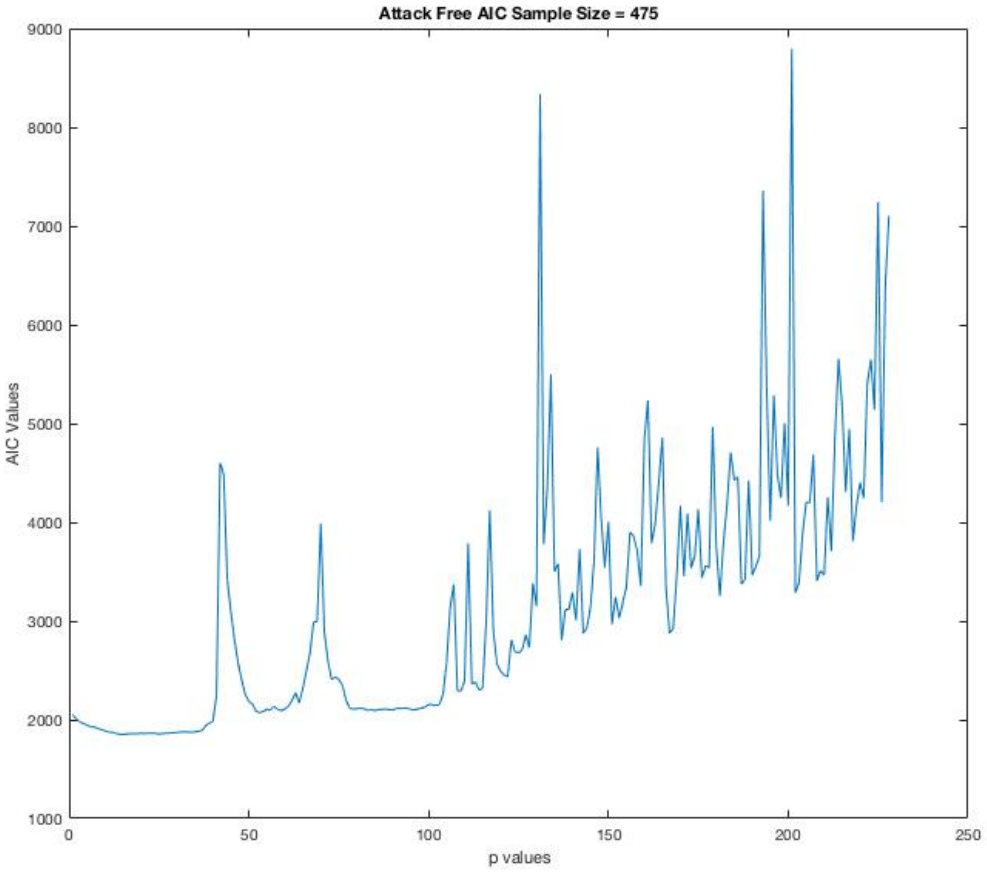

Figure 6.2. AIC Values with Respect to p

As illustrated above it is observed that when p values are taken up to N/2 , our estimation started to overfit.

6.1.5. Entropy

6.1.5.1. Shannon Entropy. The entropy or the Shannon Entropy is defined by,

$$H_S(x) = -\sum_{j=1}^{n} p_j log p_j \tag{6.35}$$

log is natural logarithm and we regard $0log0 = 0.log\frac{0}{0} = 0$. [19]

In this approach, entropy source ip has been measured in 5 sec and 10 sec intervals.

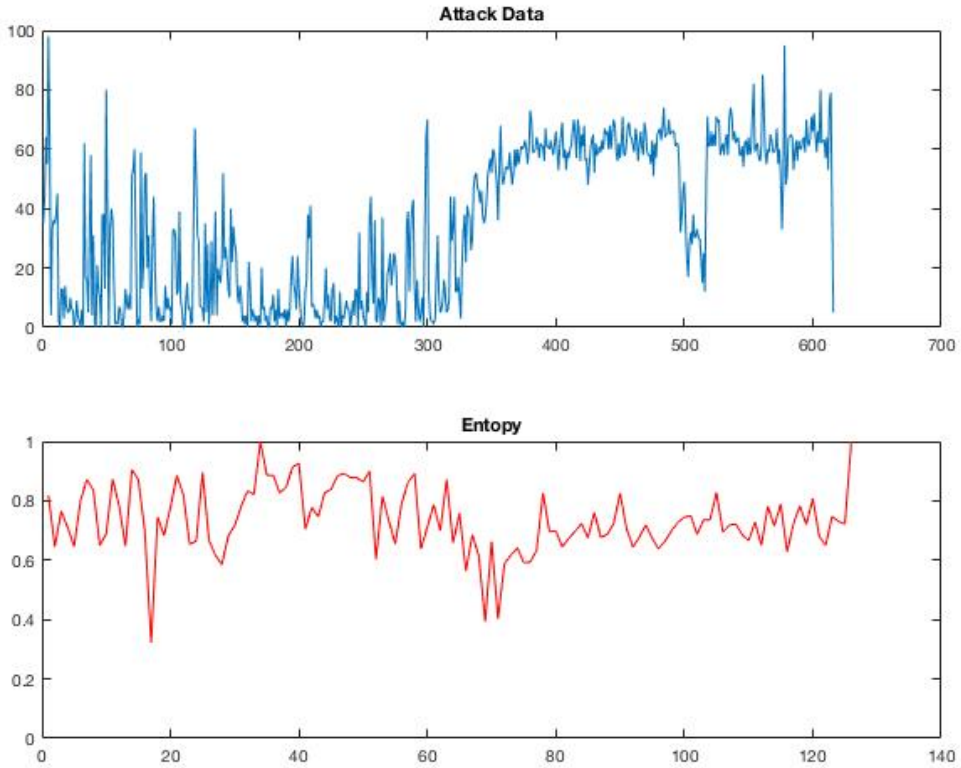

Figure 6.3. Entropy of Source ip's in 5 s successive windows

In order to increase entropy result, its sampling period (Fs) has been decreased.

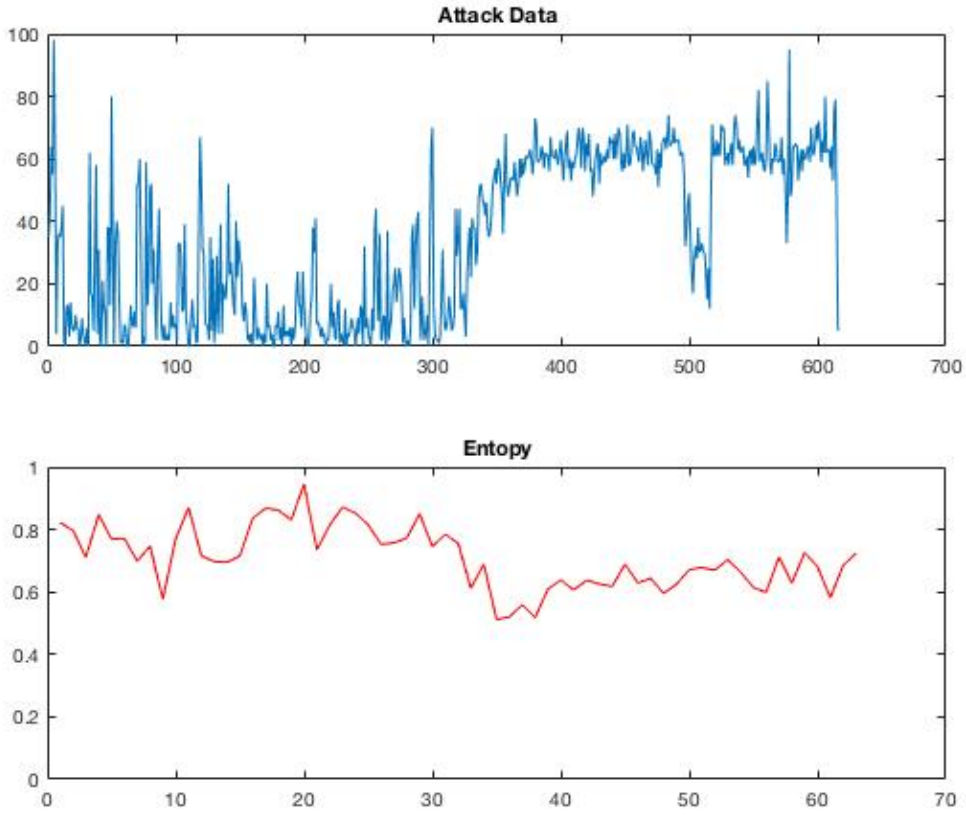

Figure 6.4. Entropy of Source ip's in 10 s successive windows

As it seen above, when attack is started after 300 sec, entropy values drops drastically. Also 10 sec windows give more accurate result since sampling period is dropped. In other words, the noise in the data is decreased.

6.1.5.2. Tsallis Entropy. A generalization of $H(x)$, was proposed by Tsallis [20], and named the Tsallis entropy or nonextensive entropy, and in a discrete system is defined as,

$$H_q = \frac{1}{q-1}\left(1 - \sum_{i=1}^{N} p_i^q\right) \tag{6.36}$$

in continuous case,

$$H_q = \frac{1 - \int f(x)^q dx}{q-1} \tag{6.37}$$

Tsallis Entropy has a q parameter that is also called and entropic parameter, quite different from the Shannon Entropy.

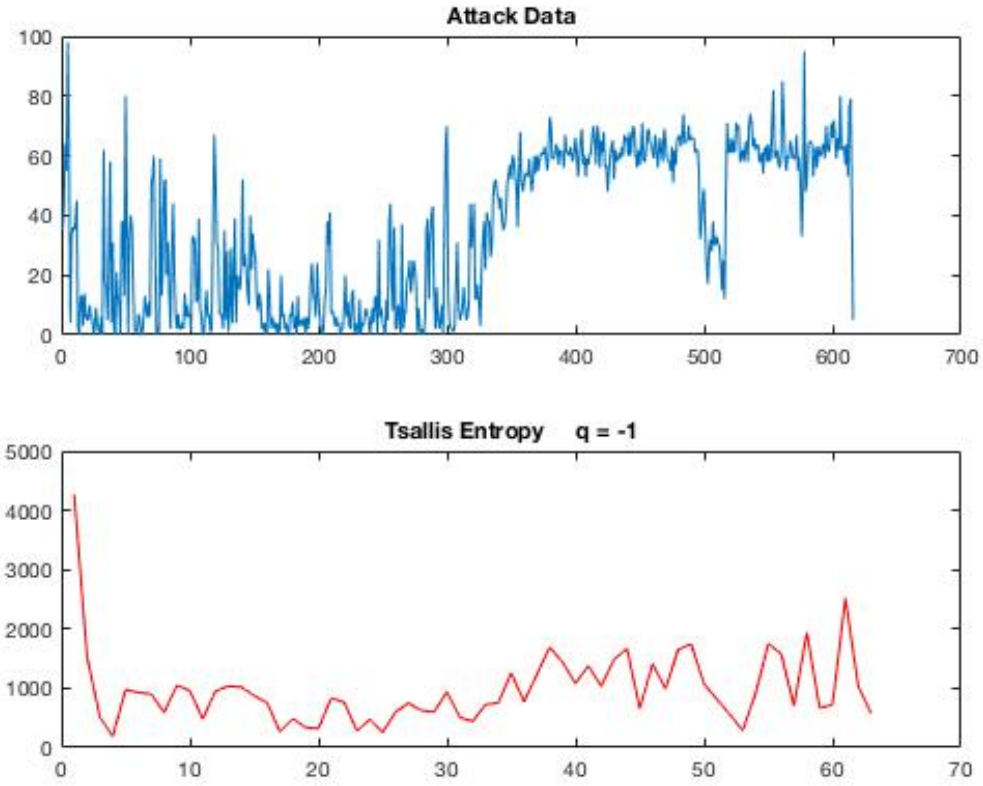

Figure 6.5. Tsallis Entropy q = -1

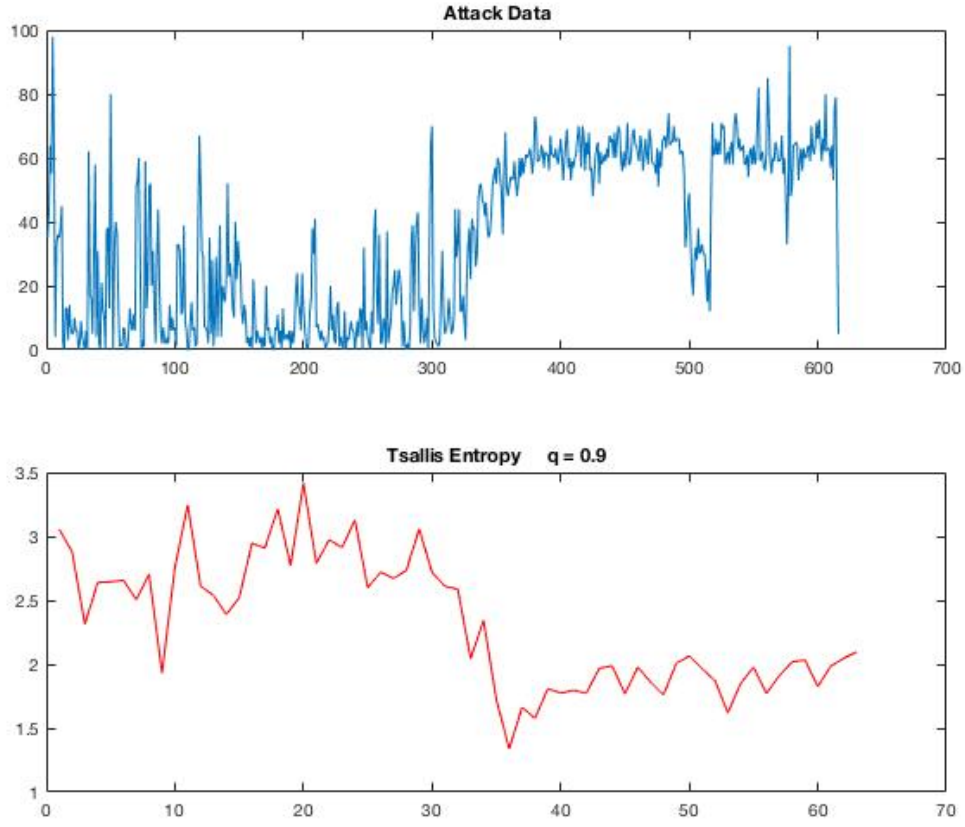

Figure 6.6. Tsallis Entropy q = 0.9

Tsallis [20] emphasized that when $q \to 1$, that is, $\lim_{q \to 1} H_q = H_S$. We can make adjustment whether high probabilities contribute more than lower one or the opposite case by the parameter q. As it is seen in the Figure 6.6 when $q = 0.9$, it gives quite similar results when it is compared to Figure 6.4

6.2. CUSUM

First we assume the data in a continuous case and we process data with fixed size N that already taken. At the end of each sample, a decision rule is applied over two hypothesis about the parameter θ:

$$\begin{aligned} H_0 &: \theta = \theta_0 \\ H_1 &: \theta = \theta_1 \end{aligned} \tag{6.38}$$

The following equations defined likelihood ratio for our observations from y_j to y_k.

$$\begin{aligned} S_j^k &= \textstyle\sum_{i=j}^{k} s_i \\ s_i &= ln\frac{p_{\theta_1(y_i)}}{p_{\theta_0(y_i)}} \end{aligned} \tag{6.39}$$

For a constant sample size N the decision rule d is given by,

$$d = \begin{cases} 0 & \text{if } S_1^N < h;\ H_0\ is\ chosen \\ 1 & \text{if } S_1^N \geq h;\ H_1\ is\ chosen \end{cases} \tag{6.40}$$

Let we consider modeling data using Gaussian with mean value μ and constant variance σ^2. In this sense the changing parameter θ is μ and probability density is

$$p_\theta(y) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(y-\mu)^2}{2\sigma^2}} \tag{6.41}$$

and since we move forward with known parameters, the sufficient statistic s_i is

$$s_i = \frac{\mu_1 - \mu_0}{\sigma^2}\left(y_i - \frac{\mu_0 + \mu_1}{2}\right) \tag{6.42}$$

which it can be written as,

$$\begin{aligned} s_i &= \tfrac{b}{\sigma}\left(y_i - \tfrac{\mu_0+\mu_1}{2}\right) \\ &= \tfrac{b}{\sigma}\left(y_i - \mu_0 - \tfrac{v}{2}\right) \end{aligned} \tag{6.43}$$

where

$$\begin{aligned} v &= \mu_1 - \mu_0 \\ b &= \tfrac{\mu_1-\mu_0}{\sigma} \end{aligned} \tag{6.44}$$

"This change detection algorithm is one of the oldest and most well-known algorithms for continuous inspection, and is called *Shewhart* control chart". [1]

6.2.1. Offline Statistical Derivation

Looking at an offline point of view, it is appropriate to introduce the following hypotheses about observations. $y_1, y_2, \dot{.}.., y_k$

$$\begin{aligned} H_0 :&\ \ \theta = \theta_0 \ \ for\ 1 \leq i \leq k \\ H_j :&\ \ \theta = \theta_0 \ \ for\ 1 \leq i \leq j-1 \\ &\ \ \theta = \theta_1 \ \ for\ j \leq i \leq k \end{aligned} \tag{6.45}$$

The likelihood ratio between hypotheses H_0 and H_j is

$$\Lambda_1^k(j) = \frac{\prod_{i=1}^{j-1} p_{\theta_0}(y_i) \cdot \prod_{i=j}^{k} p_{\theta_0}(y_i)}{\prod_{i=1}^{k} p_{\theta_0}(y_i)} \tag{6.46}$$

Thus the log-likelihood ratio is

$$S_j^k = \sum_{i=j}^{k} ln \frac{p_{\theta_1}(y_i)}{p_{\theta_0}(y_i)} \tag{6.47}$$

"When the change time j is unknown, the standard statistical approach consist of estimating it by using the maximum likelihood principle, which leads to us following decision function" [1] :

$$g_k = \max_{1 \leq j \leq k} S_j^k \tag{6.48}$$

As it is discussed in the formulation (6.45) it may be interpreted

$$S_1^j = \frac{\mu_1 - \mu_0}{\sigma^2} \sum_{i=1}^{j} \left(y_i - \frac{\mu_0 + \mu_1}{2} \right) \tag{6.49}$$

$$g_k = \left[g_{k-1} + \frac{\mu_1 - \mu_0}{\sigma^2} \left(y_k - \frac{\mu_1 + \mu_0}{2} \right) \right]^+ \tag{6.50}$$

and finally

$$g_k = \max_{1 \le j \le k} S_j^k \tag{6.51}$$

As stated in second formulation adaptive threshold applied and ROC curve plotted in the figure 8.12

6.2.2. Two-sided CUSUM Algorithm

As it is discussed in section 6.2 for detecting increase in the mean and the second one is detecting decrease in the mean as is shown below

$$\begin{aligned} g_k^+ &= \left[g_{k-1}^+ + y_k - \mu_0 - \tfrac{v}{2} \right] \\ g_k^- &= \left[g_{k-1}^- - y_k + \mu_0 - \tfrac{v}{2} \right] \end{aligned} \tag{6.52}$$

Formula that is given above known as *cumulative sum control chart* and widely used in continuous inspection for quality control [1].

6.2.3. Geometrical Interpretation in the Gaussian Case

To begin with, if the decision function (6.64) rewritten as, we obtain

$$g_k = \max_{1 \le j \le k} \sum_{i=j}^{k} (y_i - \mu_0 - \frac{v}{2}) \tag{6.53}$$

Let the corresponding decision rule, the alarm is set at time k at which there exists a time instant j_0 such that

$$\sum_{i=j}^{k} (y_i - \mu_0 - \frac{v}{2}) \ge \tilde{h} \tag{6.54}$$

The straight line with slope $\omega tan(\alpha)$, where ω is the horizontal distance between successive points in terms of a unit distance on the vertical scale, and α is the angle between this line and the horizontal one. It is obvious that [1]

$$tan(\alpha) = \frac{v}{2\omega} \tag{6.55}$$

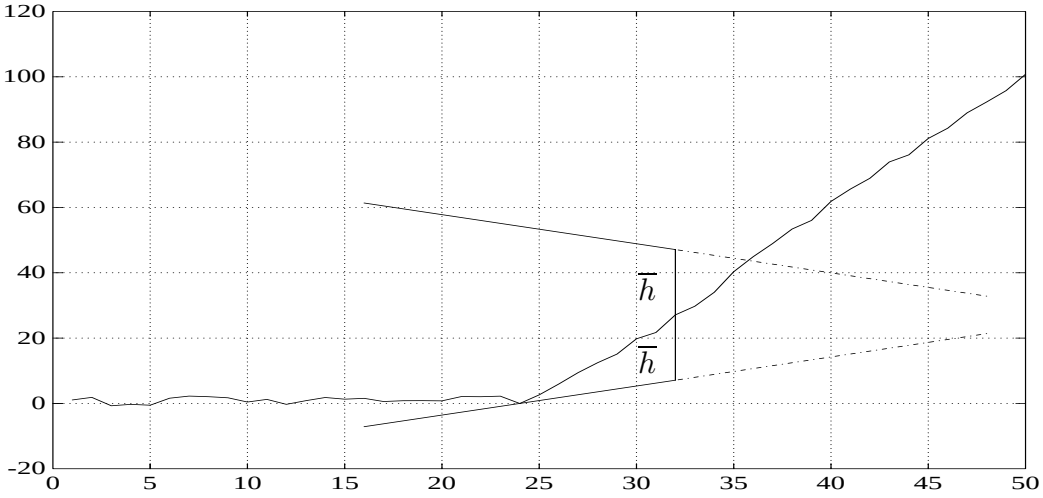

Figure 6.7. CUSUM V-mask, in the case $\mu_0 = 0$, $\sigma = 1$ [1]

As it is illustrated in the figure 6.7, half a V-mask is defined. Let $d = \tilde{h}/tan(\alpha)$ be the distance between the current sample point y_k and the vertex of the V-mask plotted forward. Then equation (6.57) can be rewritten in terms of these parameters [1]:

$$\sum_{i=j}^{k}(y_i - \mu_0 - \omega tan(\alpha)) \geq d\ tan(\alpha) \tag{6.56}$$

6.3. Literature Review For Simulation And Test Environment

6.3.1. Cooja Simulator and 6lowpan Architecture

The Internet of Things (IOT) is a network of embedded physical objects or objects with electronics, software, sensors, and network connectivity that enable them to aggregate and change data. The Internet of Things allows objects to be sensed and controlled remotely across existing network infrastructure. The latest developments in IOT, RFID, smart sensors, communications technology and Internet protocols have

been activated.

"Smart sensing systems that interconnect everyday home systems and their environmental monitoring systems can be designed and developed for IoT framework. For effective transmission and high throughput of data across the WSN- IoT framework, the measurements of smart sensor data types, number of sensing channels, and the sensor data sample intervals are to be optimally configured. Most applicable network protocols followed in WSN is the IEEE 802.15.4 (ZigBee) or 6LowPAN for effective and reliable communications. ZigBee was designed for use in local networks such as home automation environments. However, ZigBee/6LowPAN does not directly communicate with servers on the Internet without proper integrating mechanisms. Remote management and controlling of ZigBee/6LowPAN based devices over the IoT can be mechanized by following certain architectural design strategies of the IoT gateway." [21]

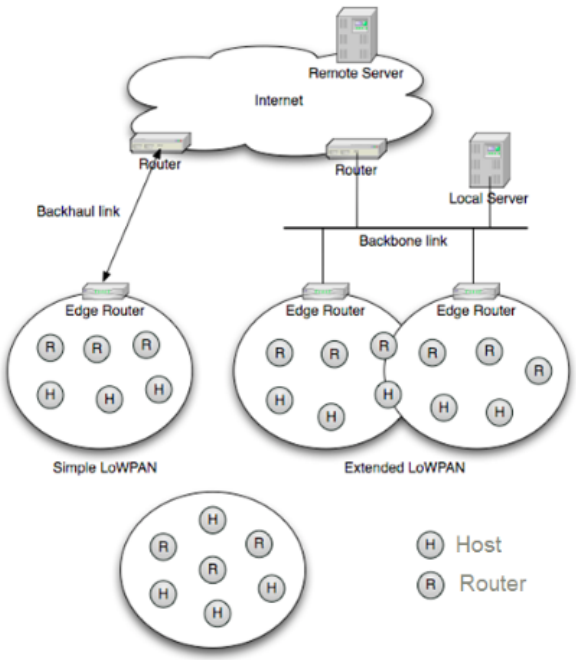

Figure 6.8. 6lowpan Architecture

Cooja Simulator is a network simulator specifically designed for Wireless Sensor Networks. A simulated Contiki Mote in COOJA [22] is an actual compiled and ex-

ecuting Contiki system. The system is controlled and analyzed by COOJA. This is accomplished by building the Contiki as a shared library for the local platform and installing Java into the library using Java Local Interfaces (JNI). Several different Contiki libraries can be compiled and loaded by representing different sensor nodes (heterogeneous networks) in different COOJA simulations. COOJA controls and analyzes a Contiki system with several functions. For example, the simulator informs the Contiki system to troubleshoot an event or brings the entire Contiki system memory for analysis. This approach gives full control of the simulated systems of the simulator. [23]

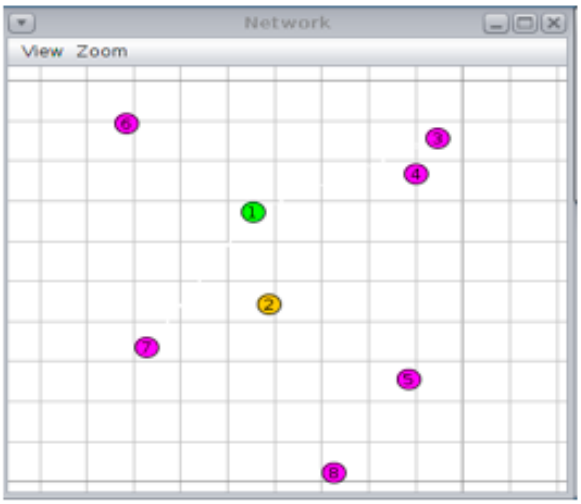

Figure 6.9. An Example of IoT Network Simulation(Yellow Node is the Malicous Node, Green Node Border Router(Gateway))

6.3.2. TinyOS

"TinyOS is an embedded, component-based operating system and platform for low-power wireless devices, such as those used in wireless sensor networks (WSNs), smartdust, ubiquitous computing, personal area networks, building automation, and smart meters. It is written in the programming language nesC, as a set of cooperating tasks and processes. It began as a collaboration between the University of California, Berkeley, Intel Research, and Crossbow Technology, was released as free and open-source software under a BSD license, and has since grown into an international

consortium, the TinyOS Alliance." [24]

6.3.3. mbedOS

"Developed by ARM in collaboration with its technological partners, mbed OS is developed for 32-bit ARM Cortex-M microcontrollers" [25]. It is an open source operating system and written using C and C++ language.

6.3.4. RTOS

"RTOS is an operating system that supports real-time applications by providing logically correct result within the deadline required." [26] Several variations of RTOS: FreeRTOS, FreeRTOS+Nabto and ChibiOS/RT and depending on the distribution, it will be complemented by a simulator. All versions of RTOS are written in C or C++.

6.3.5. OpenWSN

"The OpenWSN project is an open source implementation of a fully standards based protocol stack for capillary networks rooted in the new IEEE802.15.4e Time Synchronized Channel Hopping standard. IEEE802.15.4e, coupled with Internet of Things standards, such as 6LoWPAN, RPL and CoAP, enables ultra low power and highly reliable mesh networks, which are fully integrated into the Internet. The resulting protocol stack will be cornerstone to the upcoming machine to machine revolution." [27]

6.3.6. Workflow

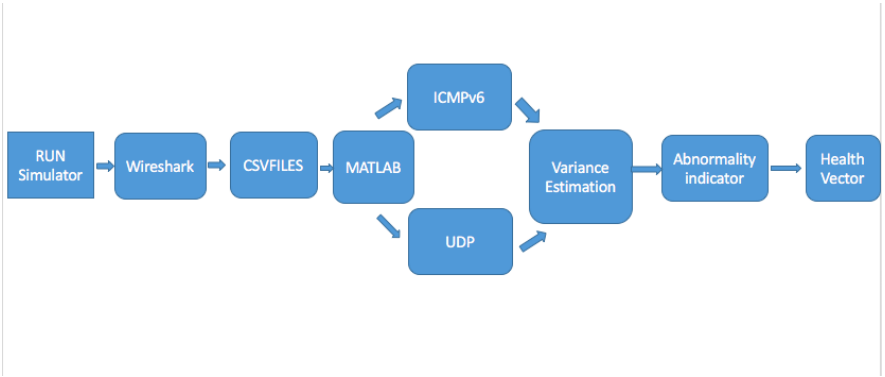

Figure 6.10. Flow Diagram

As you can see above, our workflow was described. Firstly we build IoT network in simulation. Then we attack gateway with 3 nodes and obtain pcap files. We stop simulation. Wireshark can be used to filter ICMPv6 and UDP packets, however, we prefer to use MATLAB. Variances are estimated and abnormality vector is build. Lastly as it is shown in the equation 6.3, health vector is obtained.

7. ONLINE IMPLEMENTATION WORKFLOW

Since these algorithms will work on IoT Gateways, it is vital that to show their memory performance . In our experiment, Raspberry Pi 2 Model B was used as an IoT Gateway.

Raspberry Pi 2 enhances Quad Core Processor and 1GB RAM which can also be used as home PC. The Pi 2 has 4 processors in one chip (the B+ has only one), an ARMv7 core vs an ARMv6, and 1 Gig of RAM vs 512 MB for the model B and B+ respectively. Its technical specifications has been added in the 9. Results Section.

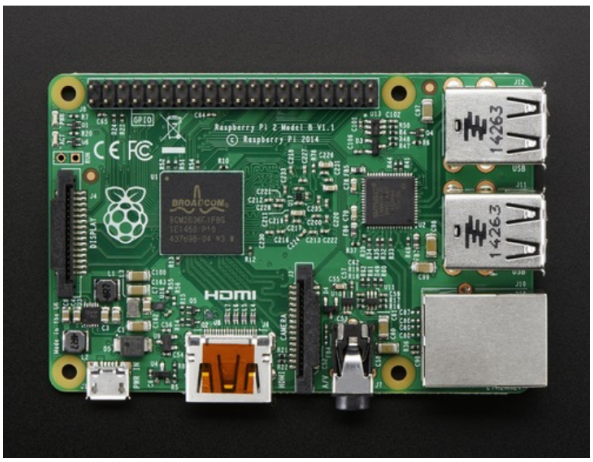

Figure 7.1. Raspberry Pi 2

For operating system, Raspbian OS which is a Linux based operating system is used. Main reason using Linux based system is, its being open source and having utilities.

An Apache web server was created for demo purposes and hosted a website. In addition, the connection point is installed in a wireless modem so that it can be reached beyond the connection point. Cusum Algorithm, Shannon algorithm and buffering mechanism was developed with Python 2.7.13.

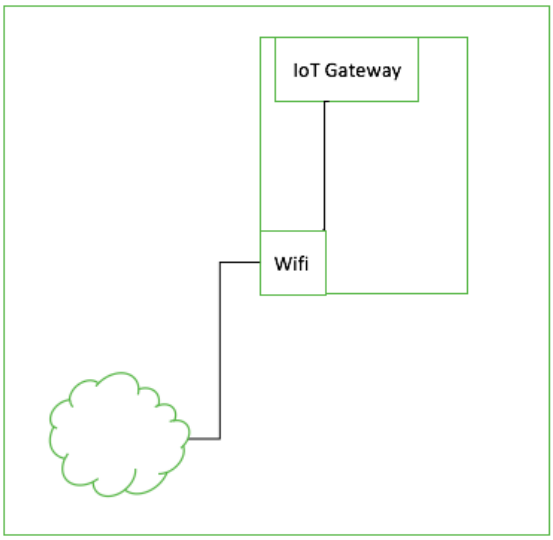

Figure 7.2. Main Architecture

The main architecture was illustrated above for simplicity. When it comes to build detection algorithms such Cumulative Sum and Shannon Algorithms, it is needed to count packets and observe Source IP's between an interval.

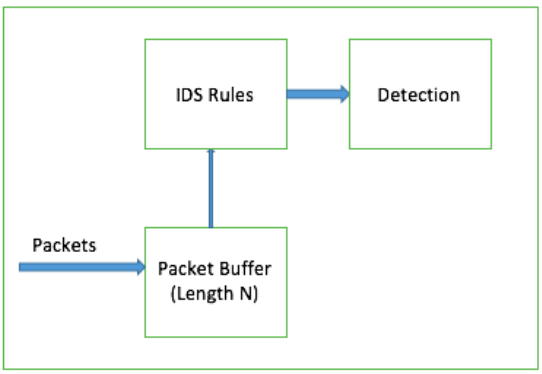

Figure 7.3. IDS Architecture

It is observed that, both algorithms seem to use % 3 percent of CPU which remain quite low when it is compared to other processes.

8. RESULTS

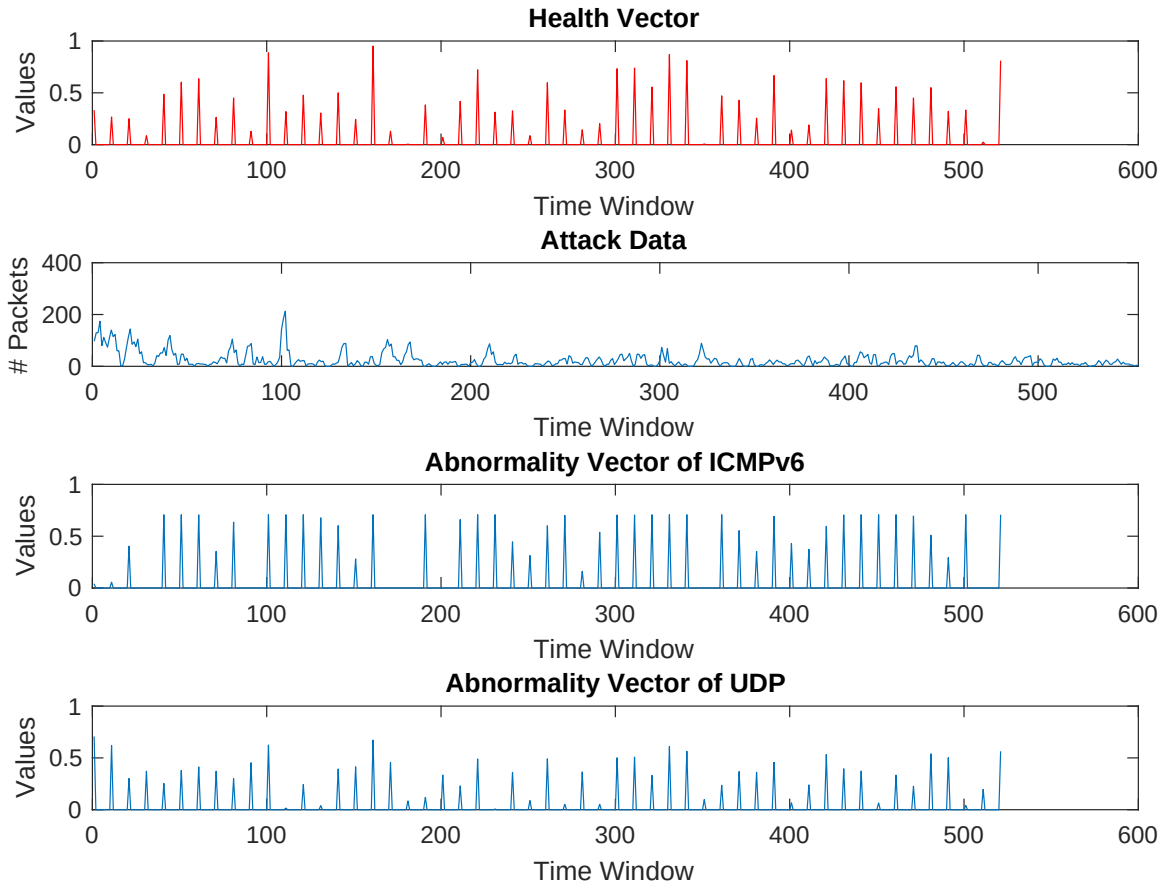

Figure 8.1. Attack Free Data

As you can see above attack free IoT network was built and samples were taken 1000 ms windows it is also called packet reception rate as described in [28]. Red figure is the health vector which is mentioned in section 6.1 AR(1) estimate. Other figures abnormality vectors of ICMPv6 and UDP packets.

It is built 15 nodes of mesh network and four Hello Flood Attack has been made against gateway as illustrated In the Figure 8.2. We applied threshold in order to build ROC Curve. In the figures 8.3 and 8.4 we foreground the health vector due to threshold. True Positive Rate and False Positive Rate has been calculated and ROC Curve was built in figure 8.5. In addition, in the figure 8.7 and figure 8.6 we increased order of

AR Model and we apply AR(4) model detection performance has been improved.

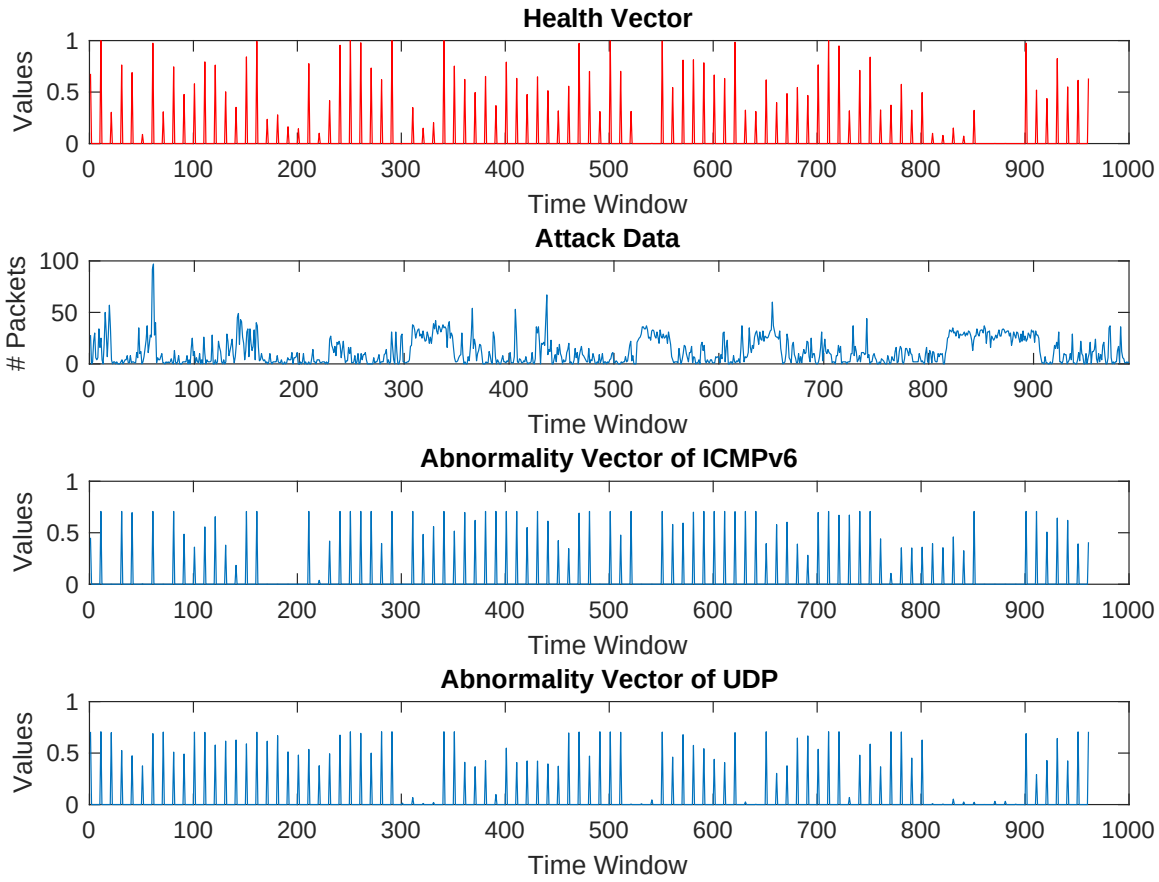

Figure 8.2. Data with four Hello Flood Attacks

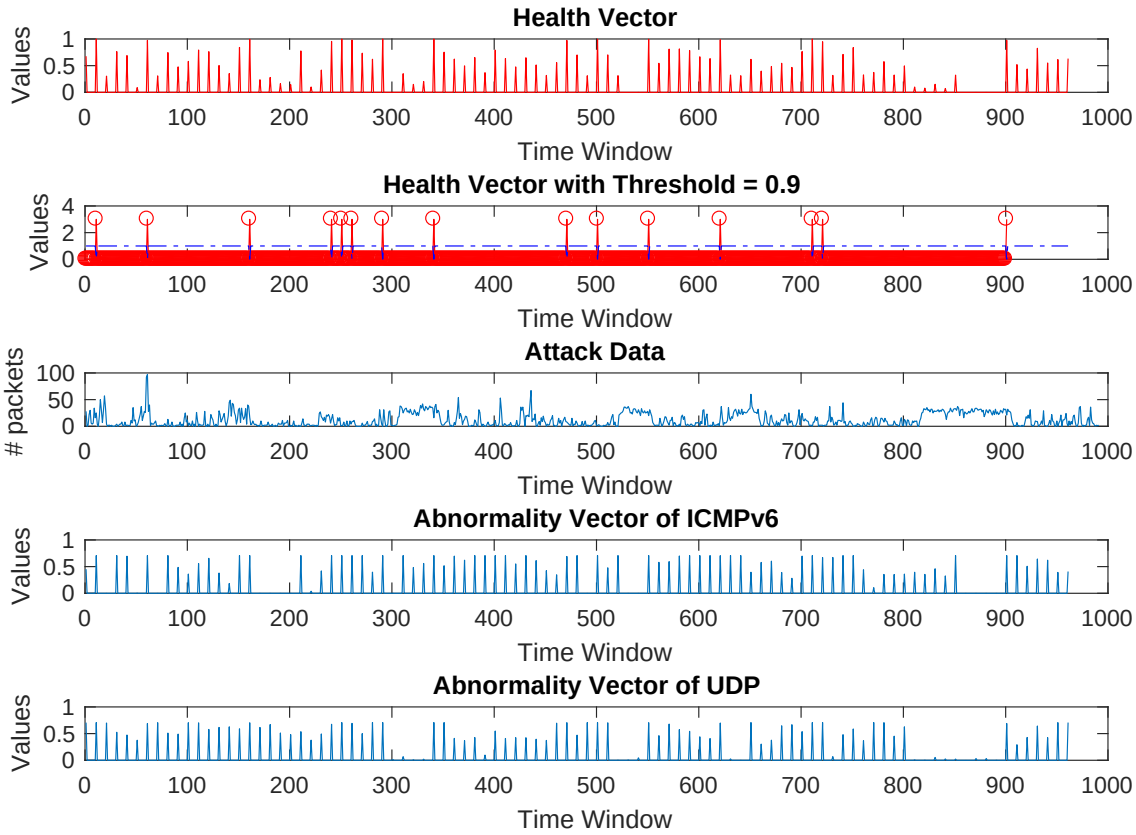

Figure 8.3. Data with four Hello Flood Attacks and Threshold = 0.9

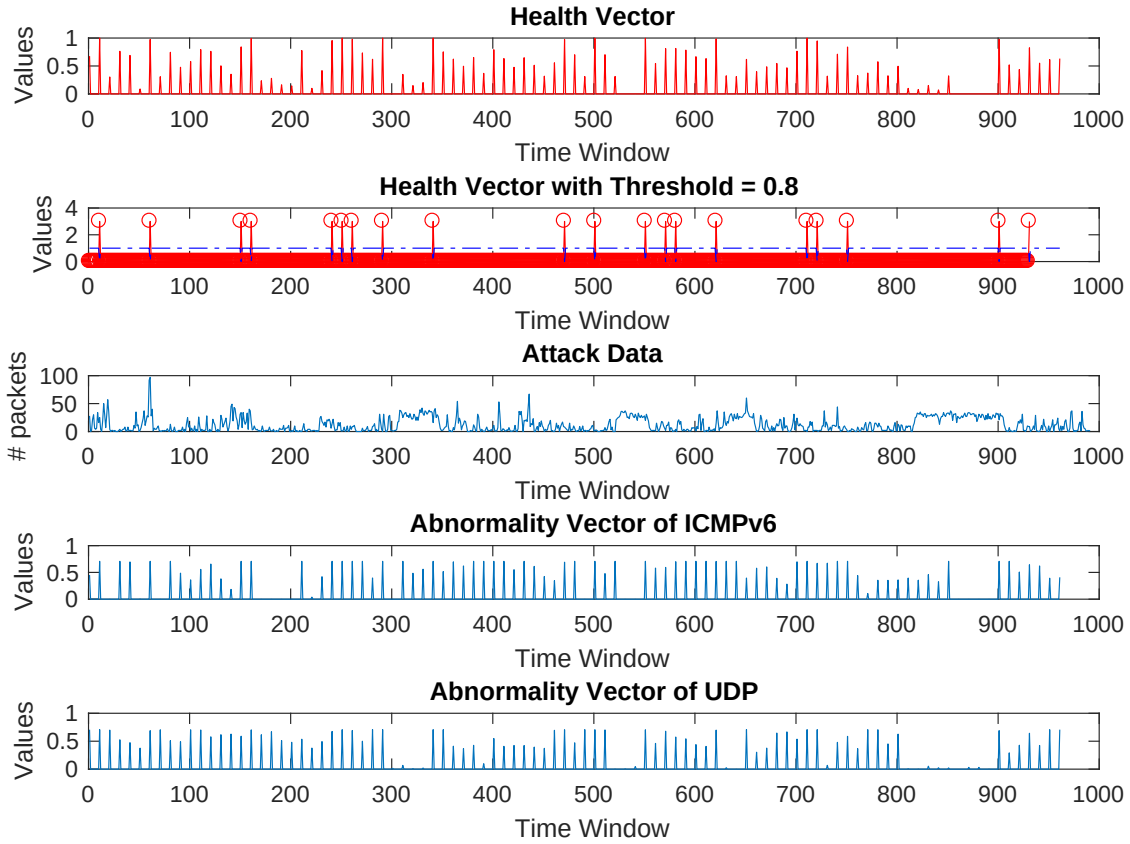

Figure 8.4. Data with four Hello Flood Attacks and Threshold = 0.8

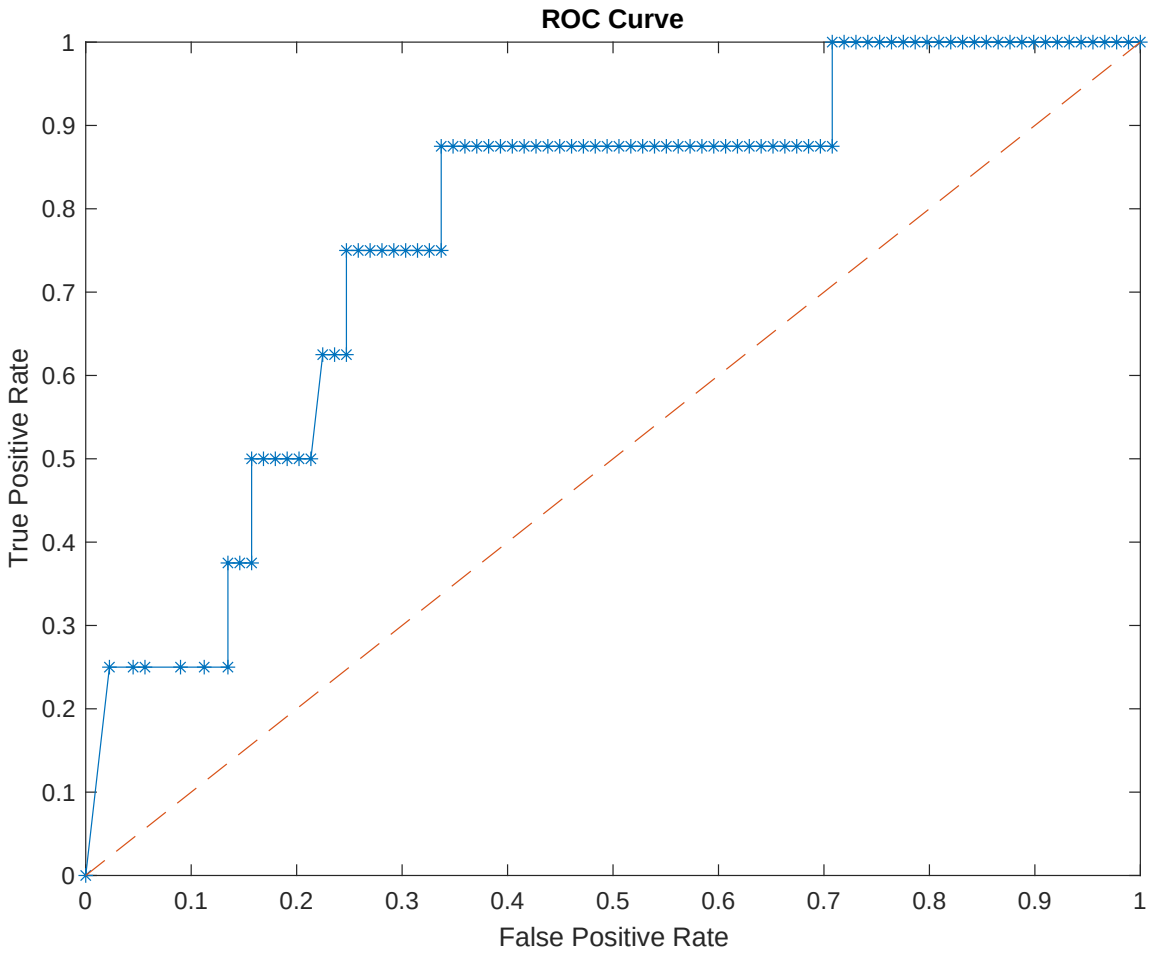

Figure 8.5. ROC Curve of UDP Data

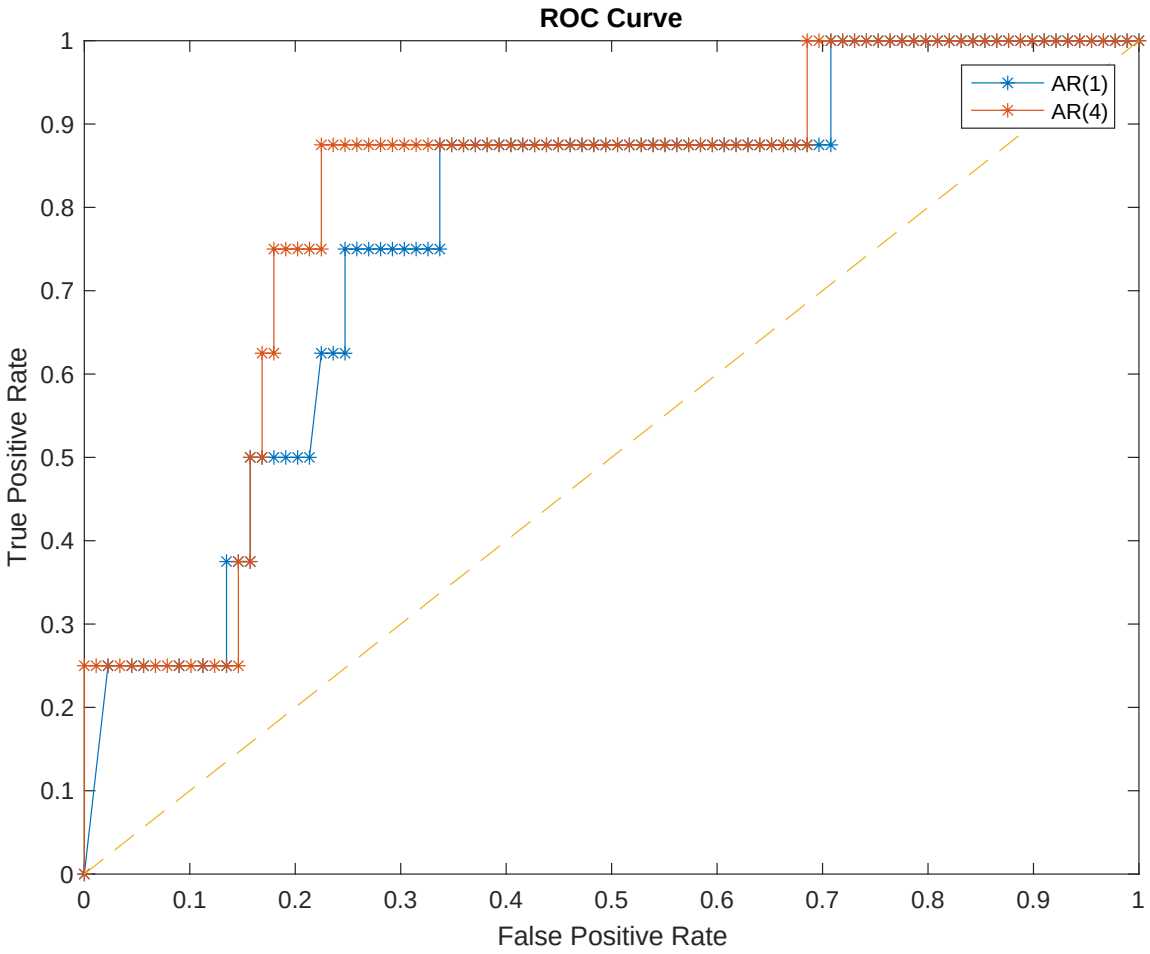

Figure 8.6. ROC Curve of UDP Data AR(1) Model and AR(4) Model Comparison

It is observed that the beginning of attacks and during the attacks Autoregressive models' algorithms are very successful. However, when it is implemented using *statsmodel*, for online inspection, it is observed that %90 CPU usage and quite high compared to other algorithms.

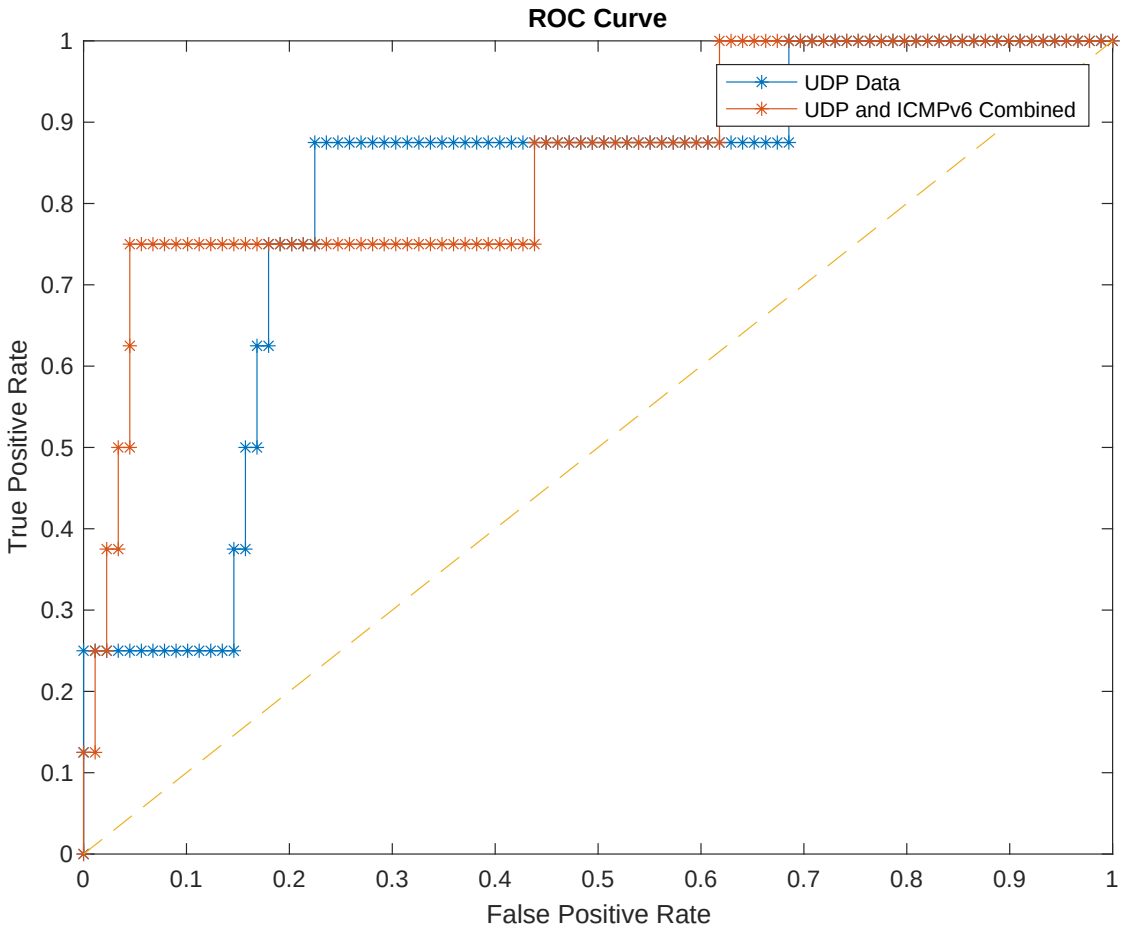

Figure 8.7. ROC Curve of UDP and ICMPv6 Combined

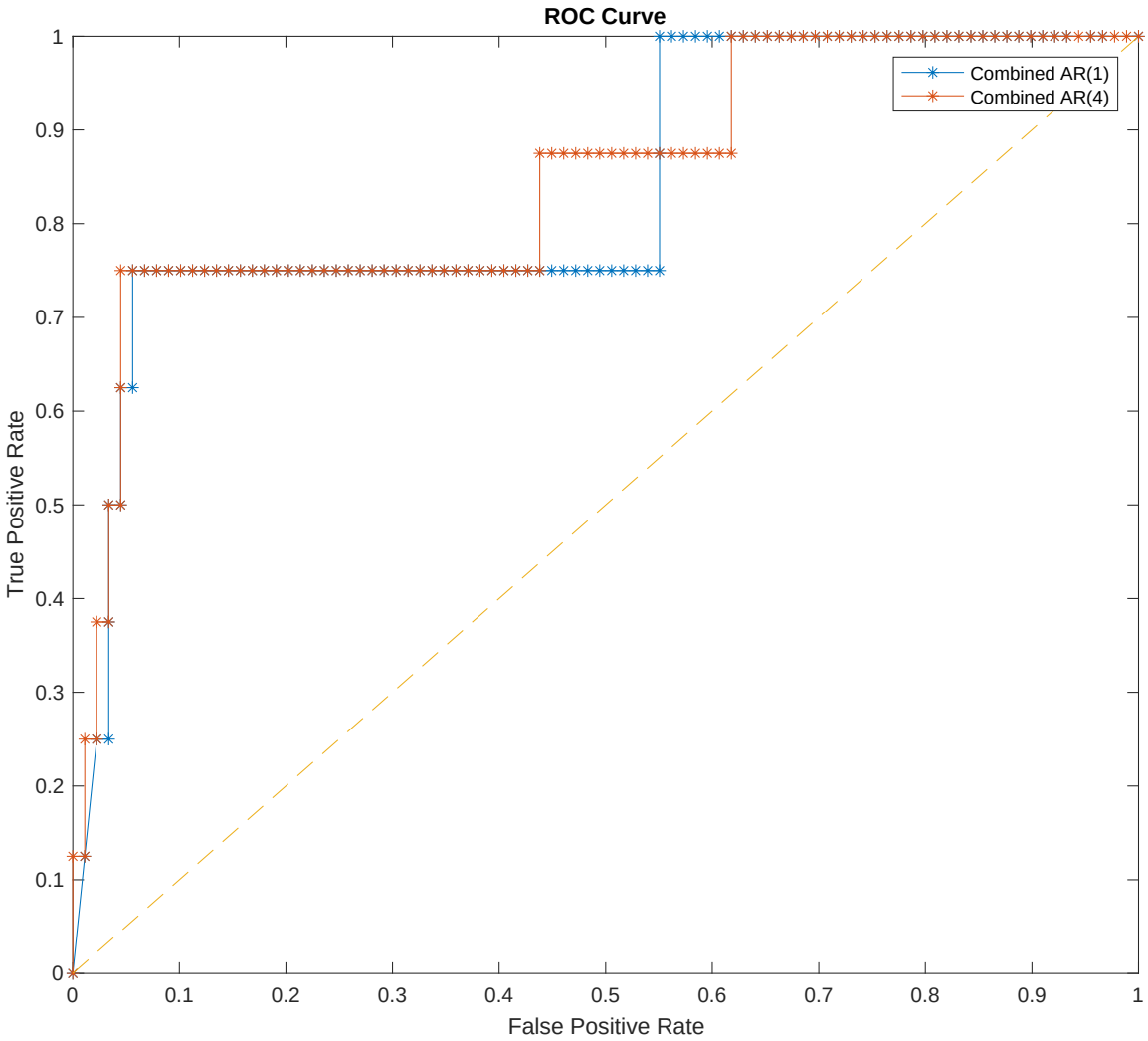

Figure 8.8. AR(4) Model vs AR(1) Model Performance Comparison

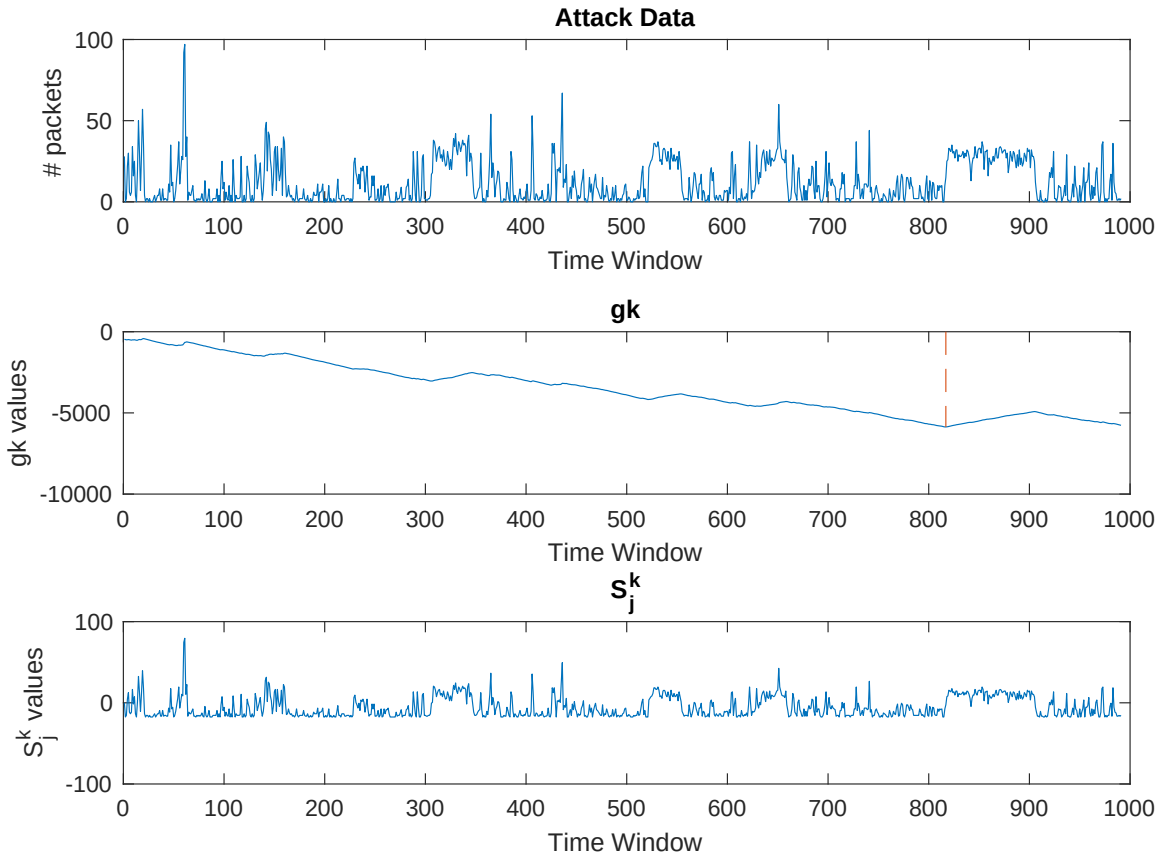

Figure 8.9. Attack Data and gk values

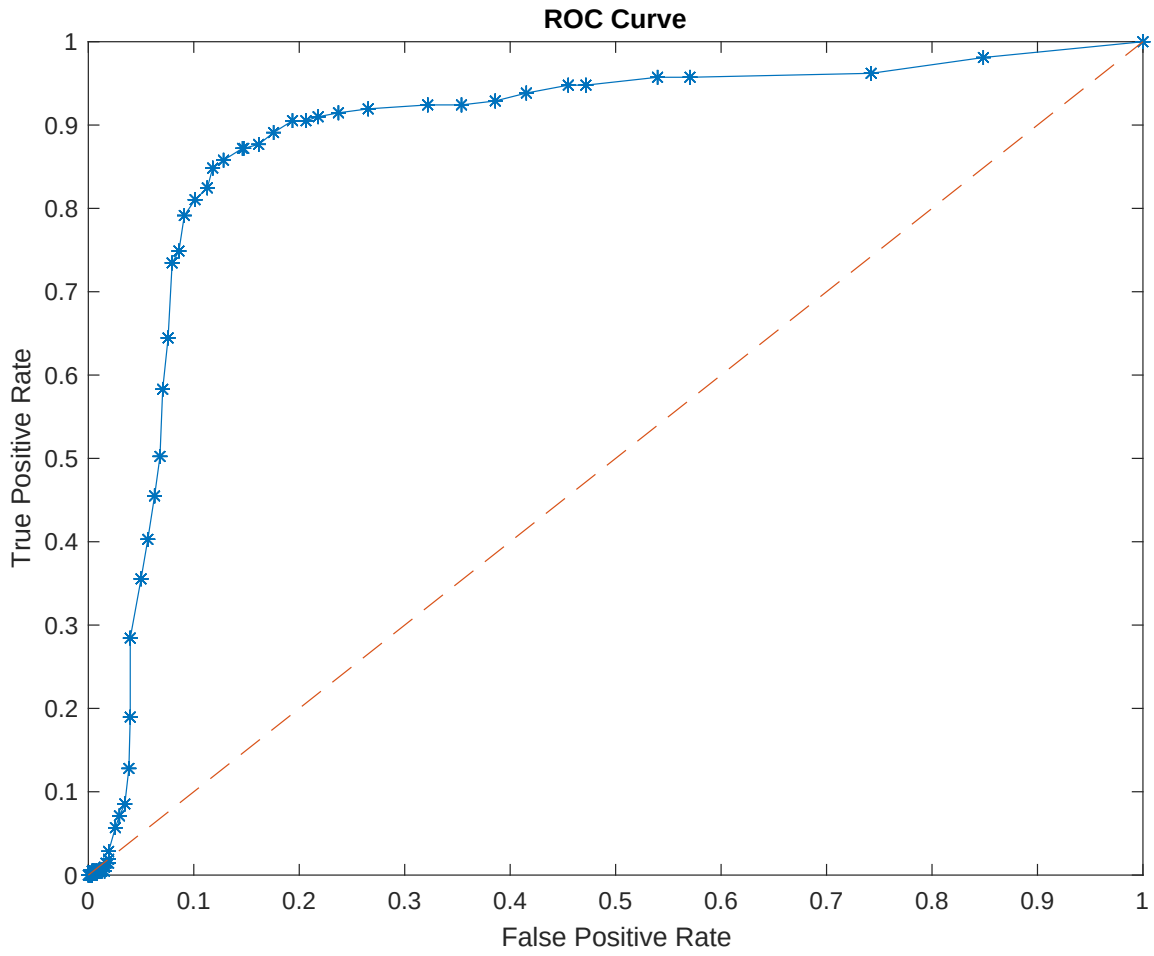

Figure 8.10. ROC Curve CUSUM Algorithm

As discussed in the Section 6.2.1 since S_j^k values added over gk values, threshold is applied $\min(k > j : S_j^k \geq h)$. As shown in the Figure 8.9 when attack occurred S_j^k values cross zero gkvalues start to increase .

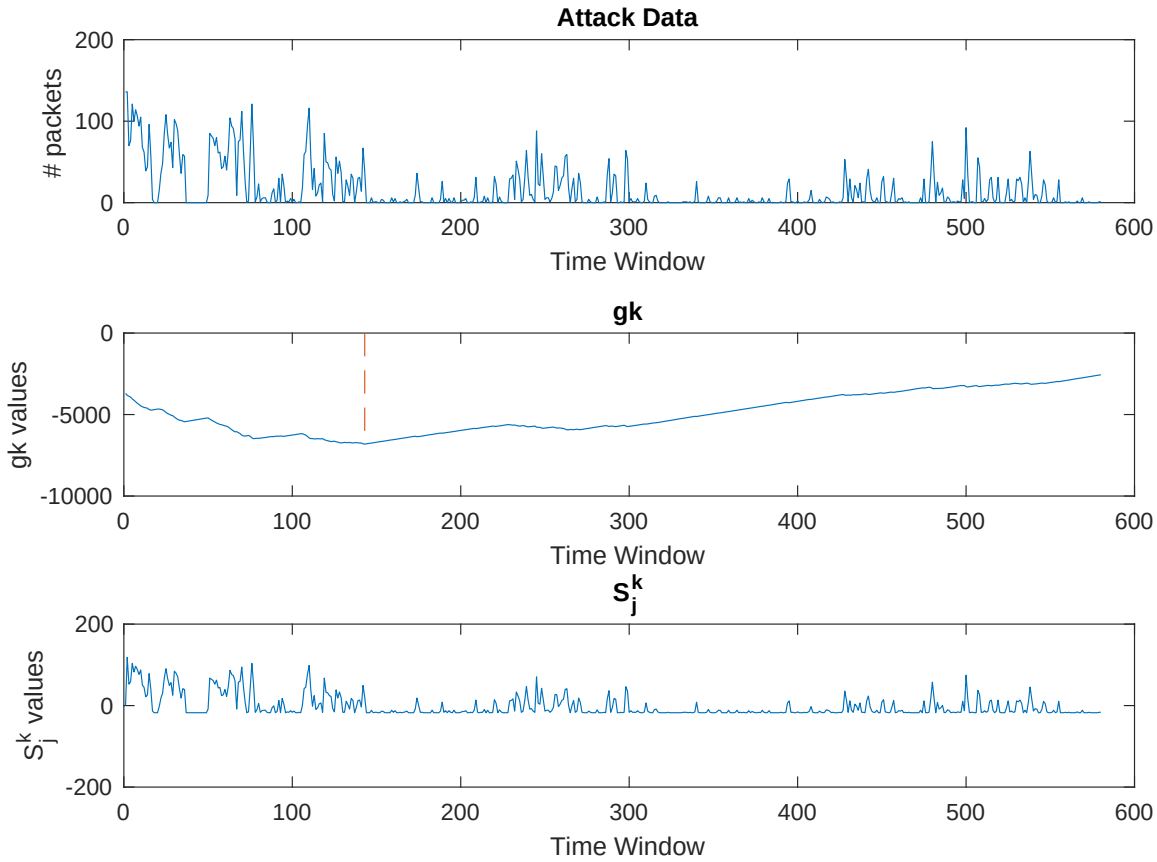

Figure 8.11. Blackhole Attack Data gk values

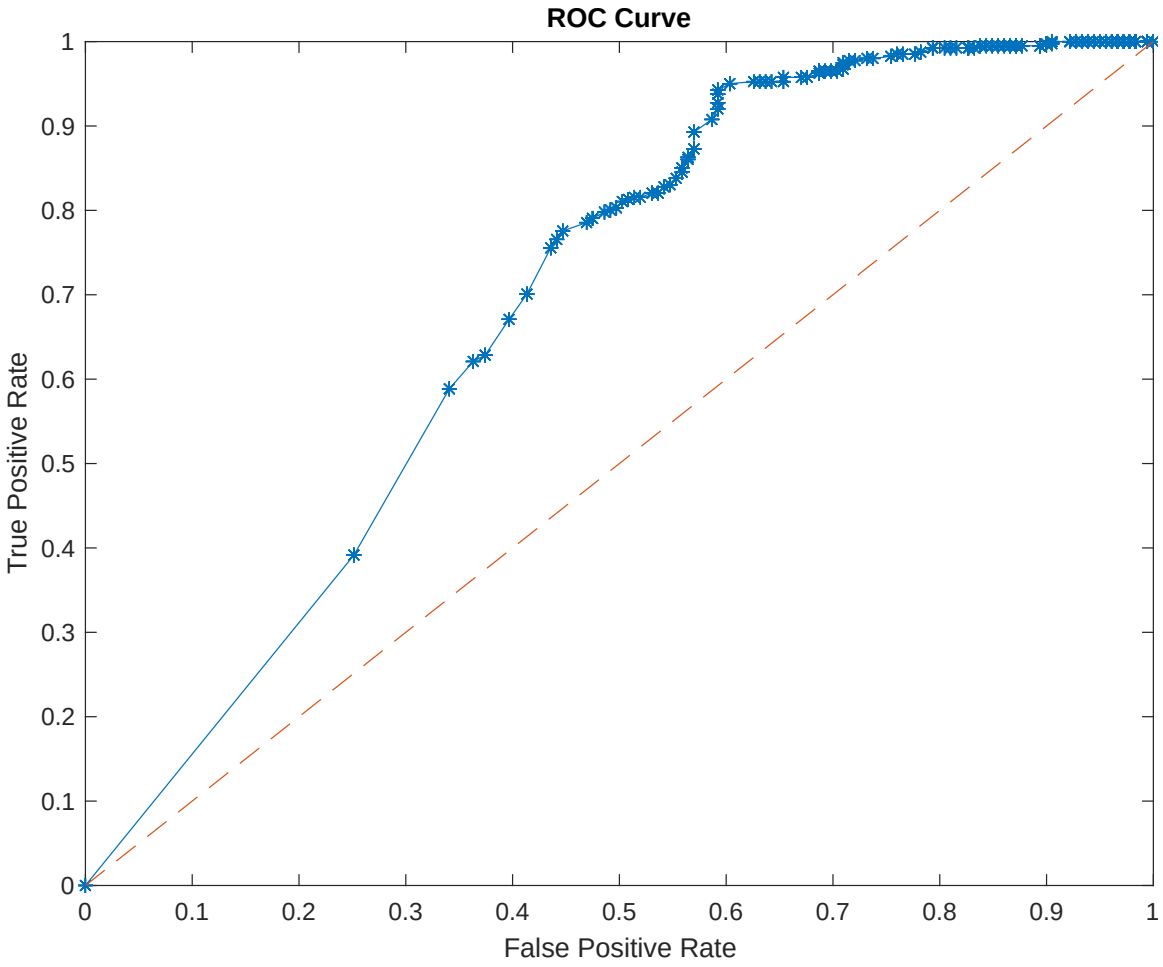

Figure 8.12. ROC Curve Blackhole CUSUM Algorithm

8.1. Raspberry Pi 2 Model B Specifications

Raspberry Pi 2 Model B technical specifications given as,

- Broadcom BCM2837 Arm7 Quad Core Processor powered Single Board Computer running at 900MHz
- 1GB RAM
- 40pin extended GPIO
- 4 x USB 2 ports
- 4 pole Stereo output and Composite video port

- Full size HDMI
- CSI camera port for connecting the Raspberry Pi camera
- Micro SD port for loading your operating system and storing data
- Micro USB power source

For features,

- Broadcom BCM2837 Arm7 Quad Core Processor powered Single Board Computer running at 900MHz
- 1GB RAM
- 40pin extended GPIO6
- 4 x USB 2 ports
- 4 pole Stereo output and Composite video port
- Full size HDMI
- CSI camera port for connecting the Raspberry Pi camera
- Micro SD port for loading your operating system and storing data
- Micro USB power source

REFERENCES

1. Basseville, M., I. V. Nikiforov *et al.*, *Detection of abrupt changes: theory and application*, Vol. 104, Prentice Hall Englewood Cliffs, 1993.

2. MARKETSANDMARKETS, *Mobile Edge Computing Market worth 838.6 Million USD by 2022*, 2017, `http://www.marketsandmarkets.com/PressReleases/mobile-edge-computing.asp`, accessed at February 2018.

3. MARKETSANDMARKETS, *Edge Analytics Market worth 7.96 Billion USD by 2021*, 2017, `http://www.marketsandmarkets.com/PressReleases/edge-analytics.asp`, accessed at February 2018.

4. Thottan, M. and C. Ji, "Anomaly detection in IP networks", *IEEE Transactions on signal processing*, Vol. 51, No. 8, pp. 2191–2204, 2003.

5. Wang, H., D. Zhang and K. G. Shin, "Detecting SYN flooding attacks", Vol. 3, pp. 1530–1539, 2002.

6. Brodsky, E. and B. S. Darkhovsky, *Nonparametric methods in change point problems*, Vol. 243, Springer Science & Business Media, 2013.

7. Kumar, S. and O. Gomez, "Denial of Service due to direct and Indirect ARP storm attacks in LAN environment", *Journal of Information Security*, Vol. 1, No. 02, p. 88, 2010.

8. Zargar, S. T., J. Joshi and D. Tipper, "A survey of defense mechanisms against distributed denial of service (DDoS) flooding attacks", *IEEE communications surveys & tutorials*, Vol. 15, No. 4, pp. 2046–2069, 2013.

9. Rghioui, A., A. Khannous and M. Bouhorma, "Denial-of-Service attacks on 6LoWPAN-RPL networks: Threats and an intrusion detection system proposi-

tion", *Journal of Advanced Computer Science & Technology*, Vol. 3, No. 2, p. 143, 2014.

10. Divya, R., T. Sundararajan and K. Deepak, "Effect of wormhole attack in hierarchical body area network and need for strict security measures", pp. 1–7, 2015.

11. Papadimitriou, A., F. Le Fessant, A. C. Viana and C. Sengul, "Cryptographic protocols to fight sinkhole attacks on tree-based routing in wireless sensor networks", pp. 43–48, 2009.

12. Alajmi, N. M. and K. Elleithy, "A new approach for detecting and monitoring of selective forwarding attack in wireless sensor networks", *Systems, Applications and Technology Conference (LISAT), 2016 IEEE Long Island*, pp. 1–6, IEEE, 2016.

13. Onat, I. and A. Miri, "An intrusion detection system for wireless sensor networks", *Wireless And Mobile Computing, Networking And Communications, 2005.(WiMob'2005), IEEE International Conference on*, Vol. 3, pp. 253–259, IEEE, 2005.

14. Onat, I. and A. Miri, "A real-time node-based traffic anomaly detection algorithm for wireless sensor networks", *Systems Communications, 2005. Proceedings*, pp. 422–427, IEEE, 2005.

15. Pacheco, J. and S. Hariri, "IoT security framework for smart cyber infrastructures", *Foundations and Applications of Self* Systems, IEEE International Workshops on*, pp. 242–247, IEEE, 2016.

16. De Souza, P., "Statistical tests and distance measures for LPC coefficients", *IEEE Transactions on Acoustics, Speech, and Signal Processing*, Vol. 25, No. 6, pp. 554–559, 1977.

17. Thottan, M. and C. Ji, "Adaptive thresholding for proactive network problem detection", *Systems Management, 1998. Proceedings of the IEEE Third International*

Workshop on, pp. 108–116, IEEE, 1998.

18. Wit, E., E. v. d. Heuvel and J.-W. Romeijn, "'All models are wrong...': an introduction to model uncertainty", *Statistica Neerlandica*, Vol. 66, No. 3, pp. 217–236, 2012.

19. ichir Kakihara, Y., *Abstract methods in information theory*, Vol. 4, World Scientific, 1999.

20. Tsallis, C., "Possible generalization of Boltzmann-Gibbs statistics", *Journal of statistical physics*, Vol. 52, No. 1-2, pp. 479–487, 1988.

21. Suryadevara, N. K. and S. C. Mukhopadhyay, *Smart homes: design, implementation and issues*, Vol. 14, Springer, 2015.

22. Dunkels, A., B. Gronvall and T. Voigt, "Contiki-a lightweight and flexible operating system for tiny networked sensors", pp. 455–462, 2004.

23. Cooja, *An Introduction to Cooja*, 2016, `https://github.com/contiki-os/contiki/wiki/An-Introduction-to-Cooja`, accessed at February 2018.

24. TinyOS, *TinyOS*, 2016, `https://github.com/contiki-os/contiki/wiki/An-Introduction-to-Cooja`, accessed at February 2018.

25. mbedOS, *mbedOS*, 2018, `https://mbed.org/technology/os/`, accessed at February 2018.

26. Trivedi, P. A., *Real Time Operating System (RTOS) With Its Effective Scheduling Techniques*, 2018, `https://www.ijedr.org/papers/IJEDR1302020.pdf`, accessed at March 2018.

27. Watteyne, T., X. Vilajosana, B. Kerkez, F. Chraim, K. Weekly, Q. Wang, S. Glaser and K. Pister, "OpenWSN: a standards-based low-power wireless development environment", *Transactions on Emerging Telecommunications Technologies*, Vol. 23,

No. 5, pp. 480–493, 2012.

28. Zhen, F., F. JingQi and S. Wei, "The gateway anomaly detection and diagnosis in WSNs", *Control and Decision Conference (CCDC), 2016 Chinese*, pp. 2401–2406, IEEE, 2016.