ACKNOWLEDGMENT

All praise for Almighty Allah and may He shower His blessings on Prophet Muhammad (peace be upon him), who is always a source of knowledge and guidance for humanity

I would like to offer my truthful acknowledgement to Cdr Dr Tariq Mairaj Rasool Khan PN, for giving me the opportunity to work under his supervision and guidance. His professional abilities, academic proficiencies and technical guidance helped me a lot in completion of my Thesis.

I would also like to offer gratitude to my GEC members Cdr Dr Faisal Amir PN, Dr. Sameer Qazi and Dr. Muhammad Bilal Kadri for their time to time valuable suggestions in the project.

I would like to give special thanks to my family for supporting and encouraging me during thesis work. Without their support, I wouldn't have been able complete the Thesis.

Finally, I would like to offer thanks to all my colleagues at NDT Centre for their support, especially Mr. Taha Ali for the encouragement and moral support he has provided during the research phase.

ABSTRACT

Zero day attack exploits unknown vulnerabilities present in computer applications causing a lot of trouble for the network administrators. Network Anomaly Detection Systems (NADS) is considered as one of the possible remedies for detecting the zero day attacks. These systems classify traffic on the basis of behavioural heuristics rather than matching traffic against known attack signature database. Various theories and methodologies have been proposed behind the implementation of these systems; however the key to all these methodologies is to classify traffic on the basis of statistical discriminators. With a vast diversity in network traffic and enormous flexibility at the application layer, a large number of discriminators can be used for the classification process. As the discriminators grow in number, the dimensions as well as complexity of the system also rises.

For a practically realizable system, a reduction in the dimensions is highly desirable. This work presents the application of dimensionality reduction techniques for minimizing the complexity of the Statistical Network Anomaly Detectors. Independent Feature Analysis, Probability Distribution function approaches and Fisher score computation techniques has been applied for the development of automated dimensionality reduction mechanism within the scope of this work. The results represent a significant reduction in complexity of the system with little compromise over accuracy. The developed software has also been benchmarked over one of the ARM embedded targets.

TABLE OF CONETENTS

Contents

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

NADS	Network Anomaly Detection System
NIDS	Network Intrusion Detection System
SABRE	Semi Automatic Business Research Environment
OSI	Open System Interconnection
ARPA	Advanced Research Projects Agency
NCP	Network Communication Protocol
DOS	Denial Of Service
CERT	Computer Emergency Response Team
IoT	Internet of Things
IFA	Independent Feature Analysis
PDF	Probability Density Function
PCA	Principal Component Analysis
ML	Machine Learning
AI	Artificial Intelligence
GPGPU	General Purpose Graphical Processing Unit
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
IP	Internet Protocol
kNN	Kth Nearest Neighbours
WORM	Write Once Read Many
ICCC	International Computer Communication Conference

ARFF	Attribute Relation File Format
PCAP	Packet Capture

CHAPTER 1

INTRODUCTION

1.1. Motivation

The concept of computer networks was first introduced in the late 1960s with the realization of first computer network based over military radar system Semi-Automatic Ground Environment. With the success of computer networks in the military systems, this concept was later adopted into commercial aerospace systems. The first commercial project involving the computer networks was Semi-Automatic Business Research Environment (SABRE), primarily designed as an airline reservation system. Later the advancements in the networks led to the invention of 10Mbps Ethernet, 100Mbps Fast Ethernet and later Gigabit Ethernet. The technology not only brought a vital change in the speed but also the signal types with a transition from electrical to optical technologies. This advancement has not been limited to the hardware aspects of the underlying infrastructure, but also the software aspects. In fact the software aspects advanced with more agility, leading to the development of a variety of protocols. Today hundreds of protocols can be run at the application layer of the Open Systems Interconnection (OSI) model using the same TCP/UDP transport layer stack.

The security wasn't much critical until the development of public network i.e. the Internet. With private networks, the security of computers was a trivial task that can be managed by monitoring the external storage devices connected to the computer. The sharing of resources concept of the introduction opened up vulnerabilities that can be caused to the device. One can create specialized programs, i.e. Trojan horses, WORMs etc, and by connecting to the public network can easily spread these malicious programs to others.

1.2. History of Internet

Modern day Internet founds its bases in the Advanced Research Projects Agency Network (ARPANet) that has been developed in late 1960s by the Department of Defence (DOD). This early network was designed solely for facilitating the researchers and scientists for efficient sharing and collaboration over different research projects. Initially, the ARPANet takes the form of a private network with only its access for researchers and scientists. The first public demonstration of this network was held at International Computer Communication Conference (ICCC) in October 1972 [1]. Network Communication Protocol (NCP) was used as communication protocol; however transition was made from this protocol in 1980s to earlier form of TCP/IP stack. By 1985, Internet was readily available as a technology supporting a number of applications including electronic mail.

The post-1985 involved a lot of commercialization and publicizing of the internet. In this era, a lot of changes were brought to the Internet stack, both in terms of hardware and software. The development of World Wide Web (WWW) has been one of the significant developments carried out in this era.

1.3. History of Network Security

The history of information security dates back prior to the development of computer networks. Securing information with mono-substitution ciphers dates back to ancient times. Enigma machine has been one of the first forms of the encryption engines that converted plain texts to encrypted texts. This primarily has been developed by the Germans after the First World War; however the ciphered texts were successfully broken by Polish researchers. During the Second World War, a large number of Enigma ciphers were intercepted by the British.

With the development of ARPANet, the hacking and other cyber crime activities started to emerge. Ian Murphy became the first convicted hacker in the computer history. The Computer Fraud and Abuse Act of 1986 was the first forms of laws giving the definitions of cyber crimes [2]. Morris Worms was one of the first computer worms that were widely distributed through Internet. It was launched from MIT in 1988 and proved to be the first of the Denial of Service (DOS) Attacks.

In response to the Morris Worm Attack, Computer Emergency Response Team (CERT) was created to notify and alert the computer users about the relevant computer networking issues. With the awareness of the possible damage that can be caused through this medium, several mitigation techniques were developed for securing this communication medium. This involved a large number of techniques employed at various layers of the networking stacks, including the development of cryptographic functions for networking stack (like in the form of IPSec), deployment of Firewall for blocking traffic at the application layer as well as development of different Intrusion Detection Software.

1.4. Modern Challenges

With the adoption of computer networks to low powered embedded devices, securing the miniature devices has become one of the vital issues. Internet of Things (IoT) aims to connect all the embedded computing devices to the Internet, ranging from the home appliances, automotive cars to industrial control applications. Prior to connecting these devices to the Internet, security needs to be ensured both toward the known attacks as well as the zero day attacks. The challenge here lies in securing these devices in a computational efficient way; so that the power budget of these embedded devices remain under control.

Network Anomaly Detection Systems (NADS) are used for the detection of zero day attacks in computer networks. Several NADS systems do exist, however almost all of the system depends upon the

underlying benign model that is created. These models rely upon the statistical discriminators and features. The entire complexity of the system depends upon the number of statistical discriminators that is used. However, the list of discriminators and statistical features may vary from one network environment to the other. In this thesis, different statistical techniques are adopted for reducing the complexity of the NADS system in a dynamic manner.

1.5. Objectives

The main objective of this thesis is to develop a method for the reducing the computational complexity of the NADS system with an acceptable compromise over the NADS accuracy. This objective was achieved through the following subsequent tasks.

- Analysis of different Anomaly Detection Techniques for the computer network applications.
- Collection and development of Data set in an educational campus environment.
- Development of Verification Criteria for the detection of Network Anomalies.
- Selection of Statistical Discriminators for the Anomaly Detection Process.
- Carrying out Independent Feature Analysis (IFA) for the Discriminator set over captured traffic.
- Carrying out Histogram based Probability Density Function (PDF) estimates for the discriminators over capture traffic.
- Application of scoring techniques for the reducing the number of discriminators.

1.6. Thesis Outline

CHAPTER 1: Introduces the motivation and main objectives of the Thesis as well as an outline of the contents of the following chapters.

CHAPTER 2: Provides a literature review of the Network Anomaly Detection Systems, with a brief discussion of the underlying physics behind these systems and its advantages.

CHAPTER 3: Provides the mechanism for the construction of data set from network packet captures and the criteria developed for simulation.

CHAPTER 4: Introduces the concept of Independent Feature Analysis (IFA) that is used for identifying the independent features and also the histogram based Probability Density Function (PDF) evaluation of the traffic instances for detailed analysis of traffic.

CHAPTER 5: Discusses the scoring techniques and the classification algorithms used for decreasing the complexity of the discriminators.

CHAPTER 6: Presents an analysis of the results obtained as a result of application of above techniques.

CHAPTER 7: Presents the Benchmarking results carried out over ARM target.

CHAPTER 8: Contains the conclusion of study and possible future work.

CHAPTER 2

NETWORK ANOMALY DETECTION SYSTEMS

2.1. Problem Background

The advancement in computer networks has laid the foundation for evolution of newer computer network security standards. With the introduction of hand held computing devices, tablets and smart-phones, more sophisticated malwares has been reported. As per the study report in 2013 [3], the threats trend has been changing with the shift towards newer platforms. In the early decades, social networks and cloud services has been a prime focus of the attackers. One of the biggest concerns raised regarding the security of these cloud services was due to password stolen incident to gain Dropbox access. This raised the concern over the effectiveness of the standard credential based authentication system. A similar kind of vulnerability was found in the iOS app of Dropbox that used to store login credentials in unencrypted file formats. This could cause the sensitive data to be theft with any physical access to the device.

The use of desktops and laptops has been diminishing in this modern era with a shift towards the hand held devices. According to a Gartner's survey [8], Android clearly dominates the as the operating system of these smarter devices with a market share of 53.2%. This share clearly attracts the malware authors to explore the vulnerability in these systems. In Australia and US, the Android threat exposure rate is reported to exceed that from the conventional computers. Currently, Android has been the victim of the largest number of not known or zero day attacks.

Also with the gaining popularity of the Internet of Things (IoT) framework and Wireless Sensor Networks (WSN), these low powered computing devices would be in a constant threat of being attacked. Since

most of these devices are battery powered devices, therefore conservation of power on these devices is also an issue.

Considering the current scenario, there are two challenges for efficiently securing the embedded devices.

i. Detection of Unknown or Zero Day Attacks

ii. Conservation of Power

Currently, two popular security systems are used for detecting the malicious traffic in the network.

i. Network Intrusion Detection Systems (Signature Based)

ii. Network Anomaly Detection Systems

Network Intrusion Detection Systems (NIDS) is the traditional signature based technique that has been quite popular for the detection of known attacks. These systems are based on maintaining a signature database of all the known threats. Each incoming traffic packet is matched against the database for the detection whether it is malicious or not; however these systems don't have the capacity for the detection of zero day attacks. Also these systems fail to find out the malicious traces in an encrypted traffic stream as they strictly follow searching criterion in the packet content.

Network Anomaly Detection Systems (NADS) are based on the traffic statistical patterns rather than the content searching mechanism. These systems are based on the rules that are defined for the network traffic, thus enabling the detection of zero day attacks in the network. This thesis focuses on the methods and techniques that can be employed for reducing the complexity so that it is computationally feasible on the low powered embedded devices.

2.2. Network Anomaly Detection

Network Anomaly Detection Systems (NADS) use one of the peculiar approaches for the identification of anomalous and malicious computer networks traffic patterns. These systems have been based on formulating the traffic pattern behaviour rather than the underlying contents of the network packets. Therefore, these systems depend more on the heuristics rather than the defined rules. There are enormous number of techniques that has been used for the development of NADS; however in this work focus has been laid over the Machine Learning (ML) based approaches.

2.2.1. Machine Learning

The term "Machine Learning" refers to the ability of the systems to learn from the input data, rather than following the explicitly programmed instructions [4]. This learning mechanism varies in between systems, there may be supervised, unsupervised and reinforcement based learning approaches.

Some of the techniques that can be used in the context of Network Anomaly Detection Systems (NADS) is discussed below.

i. Principal Component Analysis based Method

ii. Supervised Machine Learning based approaches

iii. Unsupervised Machined Learning based approaches

2.2.1.1. Principal Component Analysis based Method

Principal Component Analysis (PCA) based approaches has been quite popular in identifying anomalies present in systems. This particular technique is adopted to tackle the problem associated with high dimensional data.

Mathematically, **PCA is a technique in which n correlated random variables are transformed into d <= n uncorrelated variables**. The output represented with d variables can now be used representing the original data set containing n variables. In actual, these uncorrelated variables are the linear combination of the original variables. The first principal component of the transformation is projection in the direction in which the variance of the projection is maximized [5]. The second principal component represents the original variable with second largest variance. Similarly, the list goes on to the last principal component d.

The initial Principal Components contains the maximum discriminatory strength, therefore the rest can be disregarded with minimal loss as depicted in the figure below.

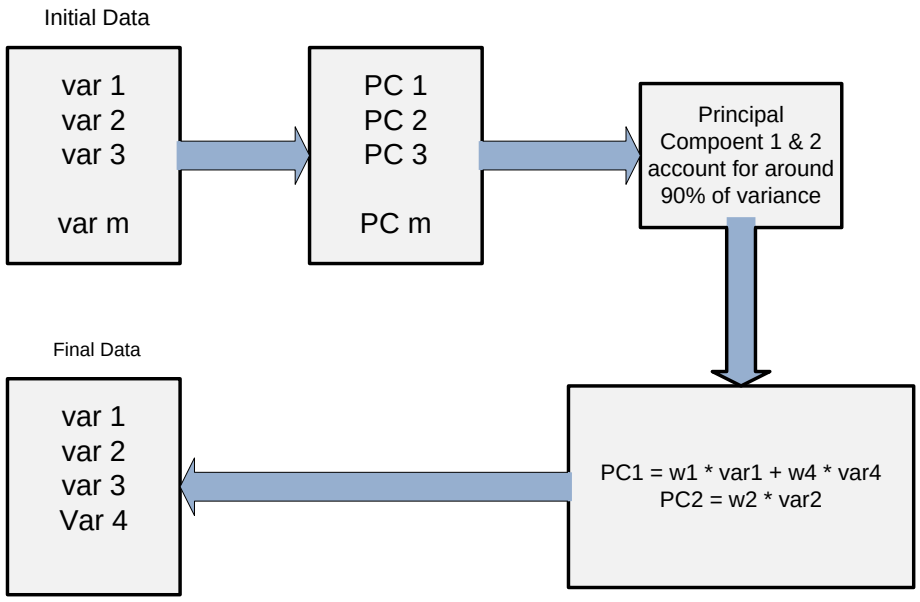

Figure 1: Principal Component Analysis

This particular property permits the utilization of this technique for the anomaly detection process. One such anomaly detection scheme has been presented in [6], in which PCA was used for reducing the dimensionality of audit data in anomaly detection process. In this work, Mahalanobis distance is computed based on the sum of squares of the principal component scores. The results from this work

demonstrate a better anomaly detection rate as compared to the Least Outlier Factor (LOF) approach and k^{th} Nearest Neighbour (kNN) approach. [7] and [8] presents other work employing the PCA approach.

2.2.1.2. Supervised Machine Learning Based Approach

This approach aims at treating the anomaly detection problem being equivalent to supervised learning from class labelled data sets [9]. Supervised ML requires the availability of a labelled data set; this data set is used for training the classification algorithm used in the process. The entire supervised ML anomaly detection process is represented in the figure below.

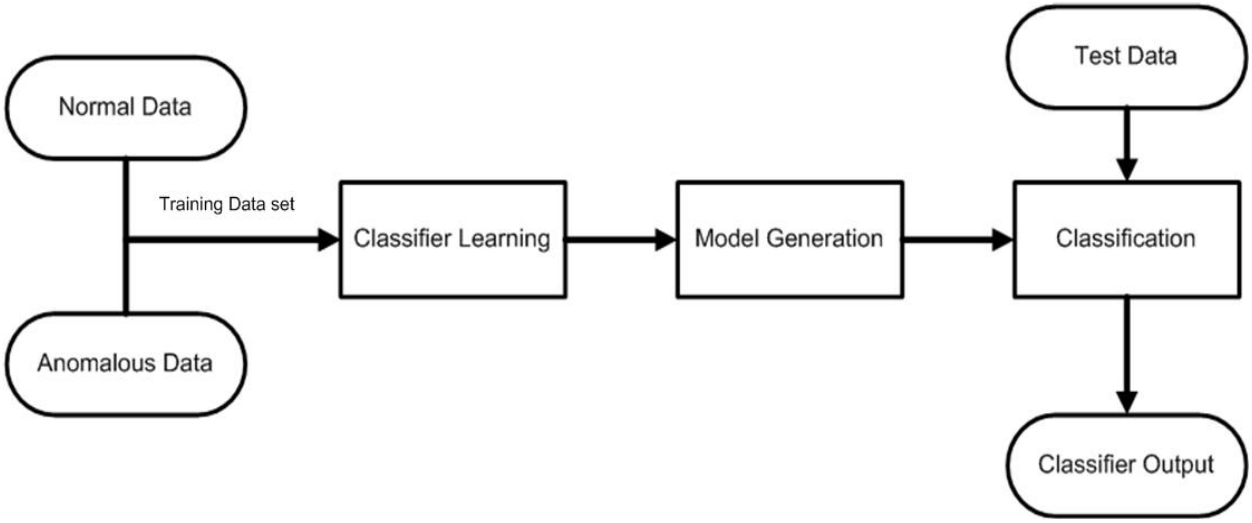

Figure 2: Supervised Machine Learning Process

One of the biggest challenges towards supervised ML is the deficiency of anomalous traffic in training data set. One of the possible mitigation techniques is to treat other traffic anomalous that deviates from the normal profile of training data set. This technique has a drawback that it leads to treating even the normal traffic not contained in the training set as anomalous.

Some of the supervised ML classification algorithms have been listed below.

i. C4.5 Decision Tree

ii. Naïve Bayesian Classifier

iii. Multi Layer Perceptron

iv. Regularized Discriminator Analysis

v. Linear Programming Machine

vi. Support Vector Machine

2.2.1.3. Unsupervised Machine Learning Based Approach

The availability of labelled traffic is not guaranteed in all cases; unsupervised classification algorithms cope up with these kinds of data sets in anomaly detection systems. These algorithms attempt to detect anomaly based on the hidden information in the traffic. Almost all of the unsupervised ML algorithms are based on the following two assumptions [9].

i. Number of Normal Instances is greater than the number of Anomalous Instances.

ii. Anomalous Instances differ distinctly than the normal instances.

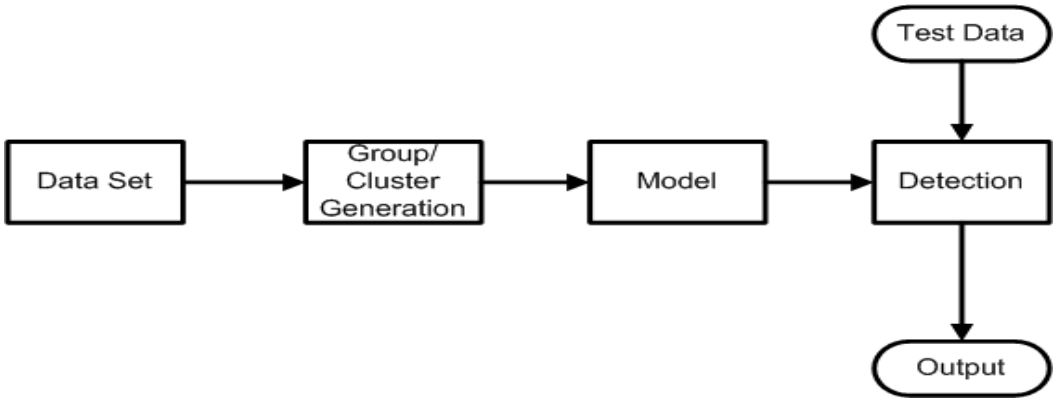

Figure 3: Unsupervised Machine Learning Flow

Since no training data set is required, there isn't any output class information available prior to classification. Hence, this method is sometimes also referred to as class discovery method [10]. Another

difference between supervised and unsupervised ML approaches is that this technique is based on Some of the established unsupervised ML algorithms include

i. γ – algorithm

ii. k – means Clustering

iii. Single Linkage Clustering

CHAPTER 3

TRAINING DATA SET

3.1. Introduction

This chapter describes about the characteristics of the selected NADS system for experimentation. As Supervised Machine Learning (ML) based technique has been adopted in this work, this chapter discussed the selected discriminators used for the classification process. Also this chapter highlights the criterion that has been developed regarding the result analysis.

3.1.1. Statistical Discriminators

In the context of supervised Machine Learning, a statistical discriminator is an individual measurable heuristic property of a phenomenon being observed in the experiment. Each ML process has its own set of discriminators and the classification algorithm associated with the process is trained with the specified set, therefore the success of any ML experiment is heavily dependent upon these discriminators.

Numerous researches have shown a large number of discriminators for the anomaly detection process in computer networks applications. We have used the 248 statistical discriminator set from [11] as baseline for this work. A complete list of statistical discriminator is attached in Appendix; however a brief list is presented below.

Table 1: NADS Statistical Discriminators

Server Port Number
Minimum Packet Inter Arrival Time

Median Inter Arrival Time
Third Quartile Inter Arrival Time
Variance Packet Inter Arrival Time
First Quartile Ethernet frame bytes
Mean Ethernet frame bytes
Maximum Ethernet frame bytes
Minimum IP Packet bytes
Client Port Number
First Quartile Packet Inter Arrival Time
Mean Inter Arrival Time
Maximum Packet Inter Arrival Time
Minimum Ethernet frame bytes
Median Ethernet frame bytes
Third Quartile Ethernet frame bytes
Variance Ethernet frame bytes
First Quartile IP Packet bytes
Mean IP Packet bytes

Many traffic characteristic from the specified list can be derived directly by counting the number of packets and packet header sizes. Also a significant number of features are calculated from the inter-arrival time and frequency analysis of the received packets.

3.2. Data Sets

3.2.1. Benchmarking Criterion

The classification process is highly influenced by the data sets used in the process. In order to closely approximate the classification behaviour with real traffic, a criterion has been developed regarding the selection of data sets for laboratory experiments [12]. The key characteristics of this benchmarking criterion include

i. The data sets used in experiments should be similar to those used in real world settings, so that the outcomes of the classifier approximate with the real world settings.
ii. The data set should contain appropriate protocol suite with respect to the experiment.
iii. The training data set should be large enough, so that the significant number of variables and observations are available for developing model.
iv. Similarly, cross validation data sets should contain sufficient traffic instances for accurately determining the classifier output.
v. The experiment needs to be performed repetitively in order to reduce the randomness effects.

3.2.2. Dataset Construction

In this work, we intend to model and benchmark the NADS system closer to real time environment. The accuracy of the benchmarking results is heavily dependent upon the data sets used in this process. In order to satisfy the developed criteria, a data set has been prepared. This data set contains network traffic traces captured over a fixed time over real networks. Since the entire network protocol is quite vast and it is impossible to construct a generalized data set addressing all the network protocols therefore this work only targets the anomaly detection for TCP session.

There were a couple of options available to us for the construction of evaluation data set.

i. Use publicly available datasets captured from dedicated servers

ii. Capture training data sets and create own data set

The final evaluation data set produced in this work has been obtained from the publicly available sources. The reason for the selection of reliance over publicly available data sets includes the initial high speed network setup requirement. Also tools need to be developed for automated labelling of traffic captured. However since each data set contains a certain degree of biasness towards its traffics, therefore we followed a hybrid approach by aggregating a couple of data sets. The two sources used in this work are

i. BRASIL data set from the Computer Laboratory University of Cambridge [13]

ii. DARPA data set from the MIT Lincoln laboratory

However, the resultant data set contains a major reflection from the BRASIL dataset.

3.2.3. Data Set Pre-processing

The two data sets have been in entirely different formats. Prior to utilization of these data sets in the classification process, the data sets need to be pre-processed in order to make them compatible with each other. Attribute Relation File Format (ARFF) has been selected as the target format. ARFF file is an ASCII character text file that describes instances along with attributes. An ARFF file generally consists of two sections, Header section followed by the data section. The Header section describes the following pieces of information.

i. Name of the Relation

ii. List of Attributes

iii. Data type for Attributes

Typically, the @RELATION and @ATTRIBUTE are the meta-data used to indicate the start of the relation or attribute field. The Data section begins with a @DATA keyword followed by a list of instances; each instance representing a particular network flow. ARFF file has an added advantage of being used with different classification algorithms in the WEKA tool suite. ARFF files can be generated from the captured traces by using Fullstats [14]; one of the open source software released under GNU GPL license. Merging of two data sets is spread uniformly in a unified data set in such a way that both the training and cross-validation data sets contain reflections from both the data sets. A trace from the resultant data set is shown in the figure below.

```
@attribute 257 numeric
@attribute 258 numeric
@attribute 259 numeric
@attribute 260 numeric
@attribute 261 numeric
@attribute 262 numeric
@attribute 263 numeric
@attribute 264 numeric
@attribute 265 numeric
@attribute 266 {ANOMALY,NORMAL}
@data
00080,02872,1.9073486328125e-
06,0.000354170799255371,0.000751495361328125,0.00620917479197184,
0.00267237424850464,0.0362350940704346,0.000153495517460642,54,54
,54,173.307692307692,179,590,43517.8974358974,40,40,40,159.307692
307692,165,576,43517.8974358974,14,14,14,14,14,14,0,7,6,6,6,4,0,0
,0,0,0,0,0,342,1197,1,3,342,1197,0,1,0,1,0,0,0,0,0,0,1,1,1,1,1,2,
?,?,?,?,?,?,Y,N,0,0,0,0,0,0,1380,536,342,536,342,125,341,398,6553
5,61440,65410,61440,0,0,65499,61440,342,1197,1,3,342,1197,0,0,0,0
,0,0,0.000000,0.000008,0.036718,0.037372,4590,16065,3,3,0.000253,
0.032930,0.000749,0.036235,0.000495,0.034397,0.000248,0.001683,0.
000483,0.036235,2,1,0.000253,0.036235,0.000483,0.036235,0.000368,
0.036231,0.000000,0.000000,0,0,0,1,1,1,0,0,0,1,0.000000,0.030688,
0.000000,0.030688,0.000000,0.030685,0.000000,0.000000,54,54,54,10
4,58,396,16588,40,40,40,90,44,382,16588,14,14,14,14,14,14,0,54,55
,118.5,254.166666666667,487.25,590,69966.5666666667,40,41,104.5,2
40.166666666667,473.25,576,69966.5666666667,14,14,14,14,14,14,0,0
.000387907028198242,0.000840663909912109,0.00185954570770264,0.01
24183495839437,0.0253528356552124,0.0367181301116943,0.0003032341
29607214,1.9073486328125e-06,5.96046447753906e-
06,0.00359416007995605,0.014332389831543,0.030687808990478 5,0.037
```

Figure 4: Data Set Sample Trace

The two data sets provide multitude of output classes, including MAIL, WEB, GAME, CHAT, ADMIN, P2P, ATTACK, etc. In order to simplify the problem, these multitudes of classes are converted into binary classes, that are

i. Normal Class

ii. Anomalous Class

These data sets provide traces of the data captured in an educational environment. Since majority of the P2P traffic correspond to file sharing purposes, therefore in this process we treat them as an anomaly for an educational environment. So ATTACK and P2P traffic has been categorized into Anomalous Data class where as the remaining traffic is converged into the Normal Data Class. We developed an automated utility aiding the Perl language's powerful text-processing features for the converting the classes to the anomaly process desired ones in the ARFF file. This utility transforms the classes both in the headers and the data portion of the ARFF file.

CHAPTER 4

STATISTICAL ANALYSIS

4.1. Introduction

This chapter describes about the initial statistical processing carried out for over captured data traffic. The processing includes the independent feature analysis (IFA) and probability distribution analysis (PDF) of the statistical discriminators.

4.2. Independent Feature Analysis

In this process, each discriminator is analyzed separately for both the normal and anomalous traffic. The values associated with each feature for both normal and anomalous traffic is analyzed. Prior to carrying out analysis, all the features associated with in the anomaly detection process are assumed to be independent of each other. By independence, it is meant that the value of one feature does not have an effect or influence over the value of the other [15]. According to the stochastic definition, two features are considered independent provided their joint probability is equal to the product of probabilities.

$$\boldsymbol{P(A \cap B) = P(A).P(B)} \qquad \text{------- eq (i)}$$

This assumption has been made in order to minimize the complexity of the relationship existing between such a large set of discriminators.

Due to the existence of large number of discriminators, i.e. 248, it is practically infeasible carrying manual analysis over each discriminator. Therefore automated utilities are developed within the scope of this work. NADS_IFA_extractor is the one used for carrying out Independent Feature Analysis and makes use of the combination of Perl and MATLAB. Figure presents pseudo algorithm for the NADS_IFA_extractor utility.

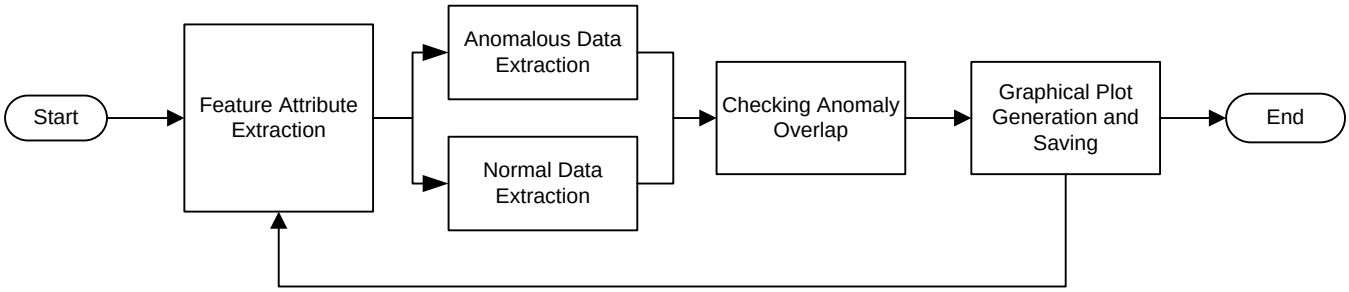

Figure 5: Independent Feature Analysis Flow Diagram

Following the above pseudo algorithm, we developed a set of 248 plots, each for a single discriminator. A couple of plots representing minimum overlap values and maximum overlap values are presented below.

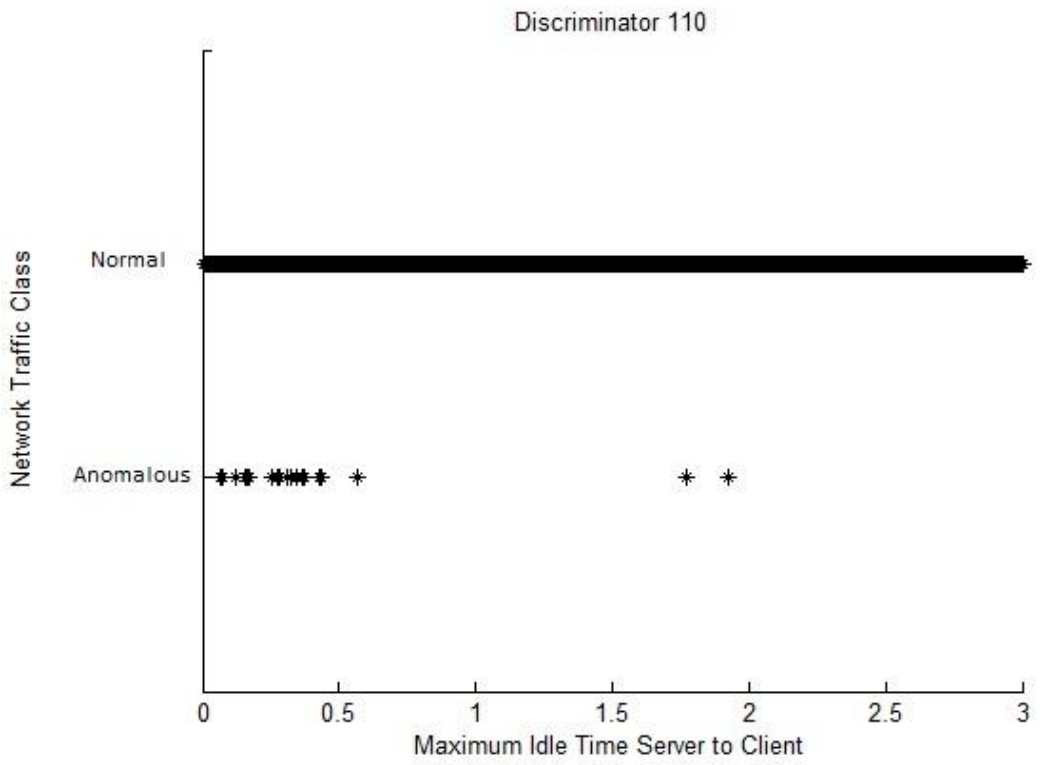

Figure 6: IFA plot for Discriminator 110

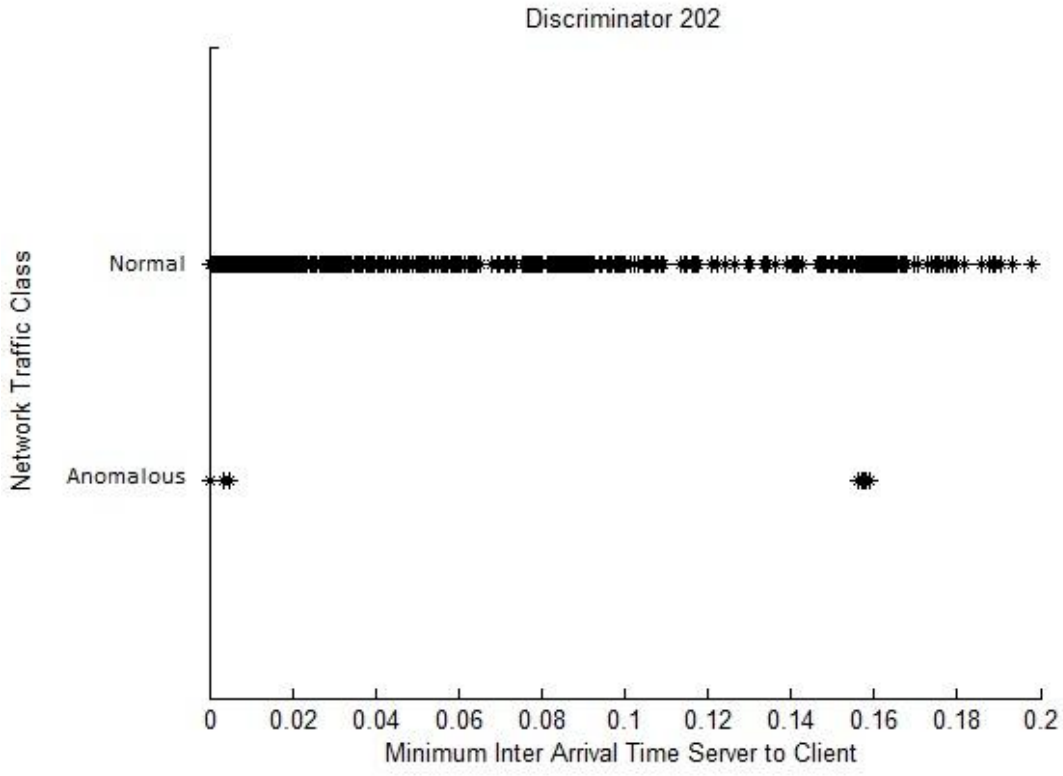

Figure 7: IFA plot for Discriminator 202

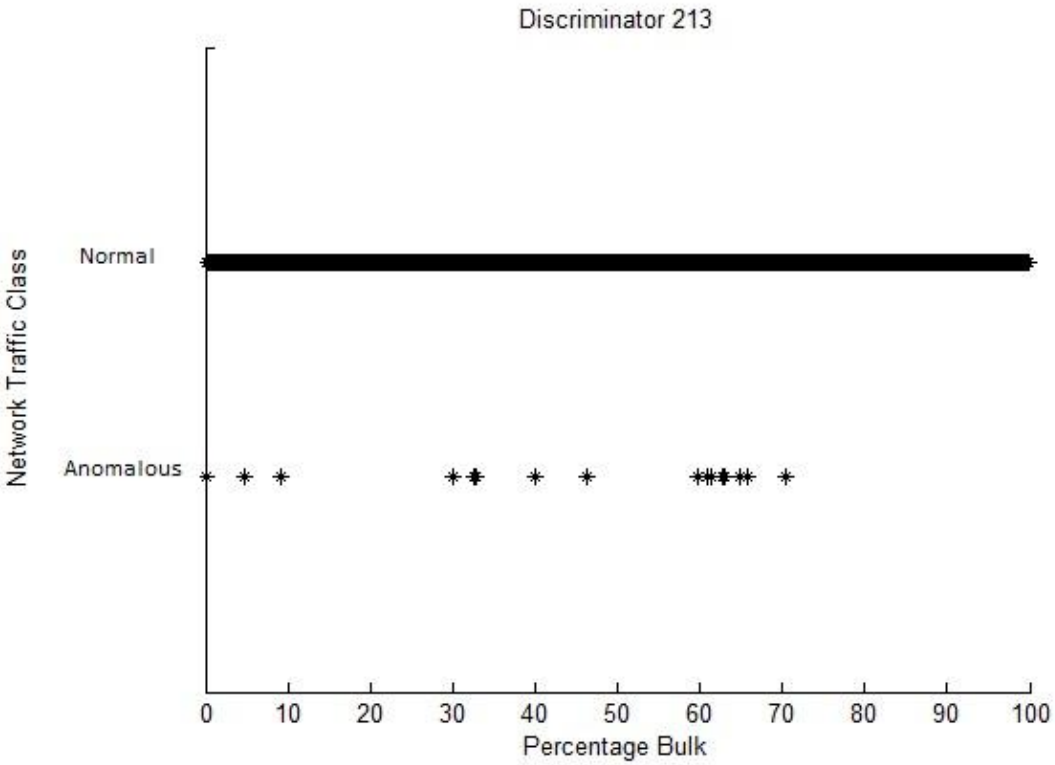

Figure 8: IFA plot for Discriminator 213

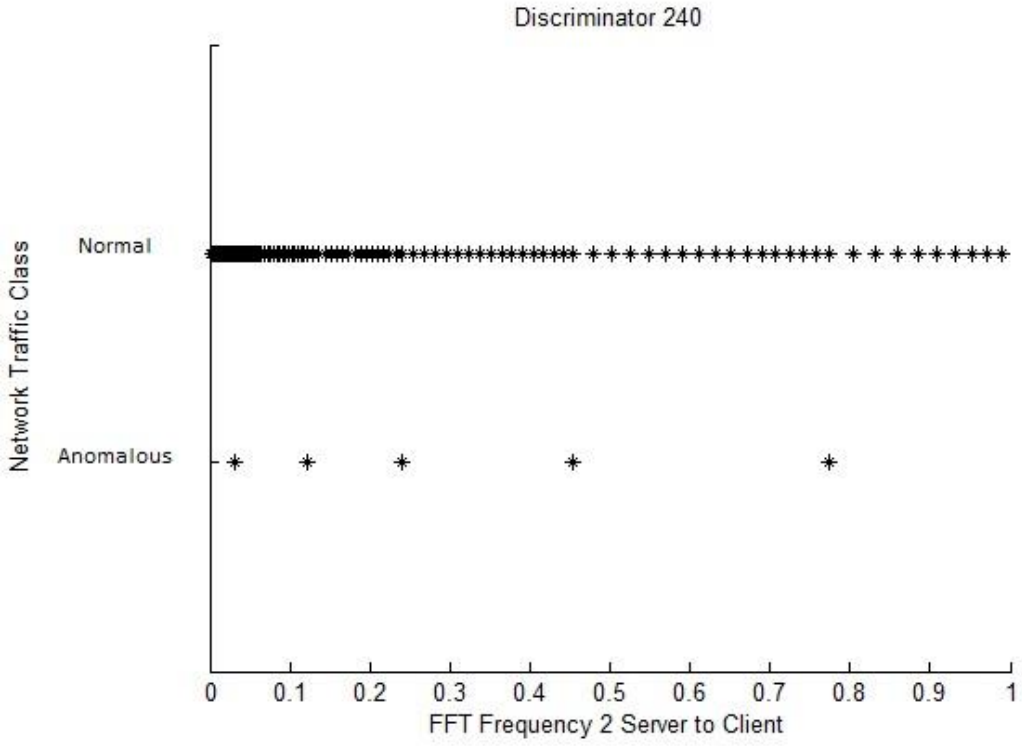

Figure 9: IFA plot for Discriminator 240

Observing the above four plots, it is clearly found that none of the statistical discriminator is without any overlap. Therefore, Independent Feature Analysis (IFA) reveals that none of the feature can be used to independently for the classification process. However the overlapping area on the x axis varies in between the discriminators. Of the four plots, Figure 10 represents the plot with minimum x-axis overlap span, whereas the Figure 11 presents the plot with maximum overlap span.

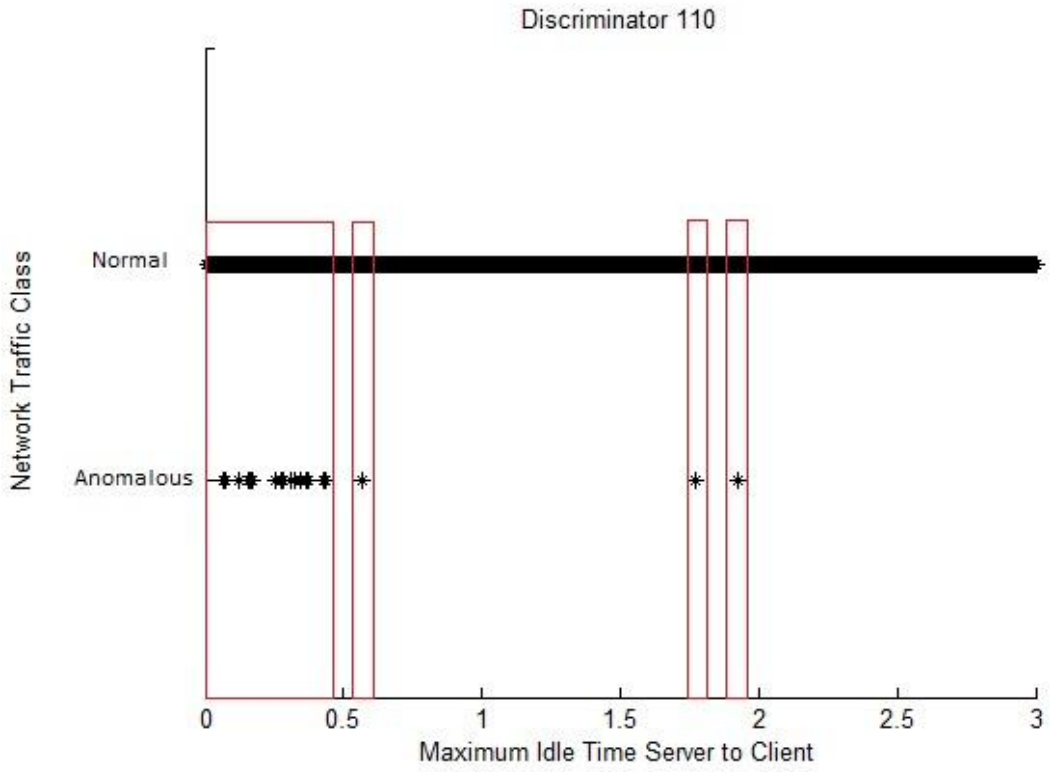

Figure 10: Overlap in IFA plot for Discriminator 110

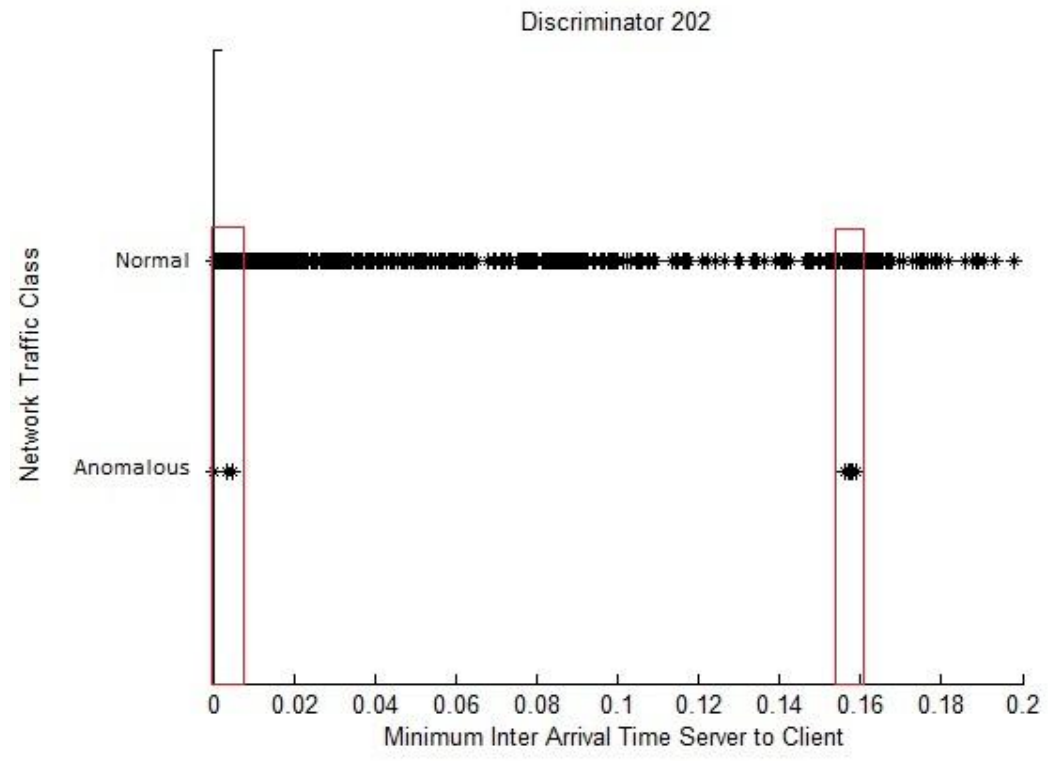

Figure 11: Overlap in IFA plot for Discriminator 202

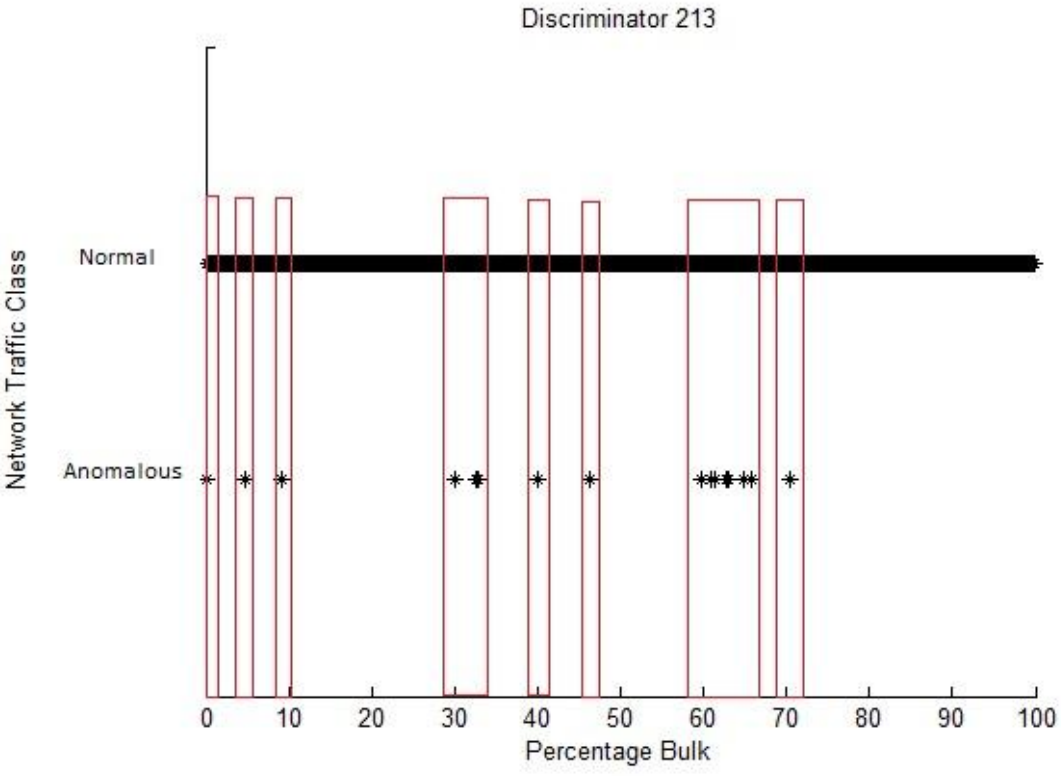

Figure 12: Overlap in IFA plot for Discriminator 213

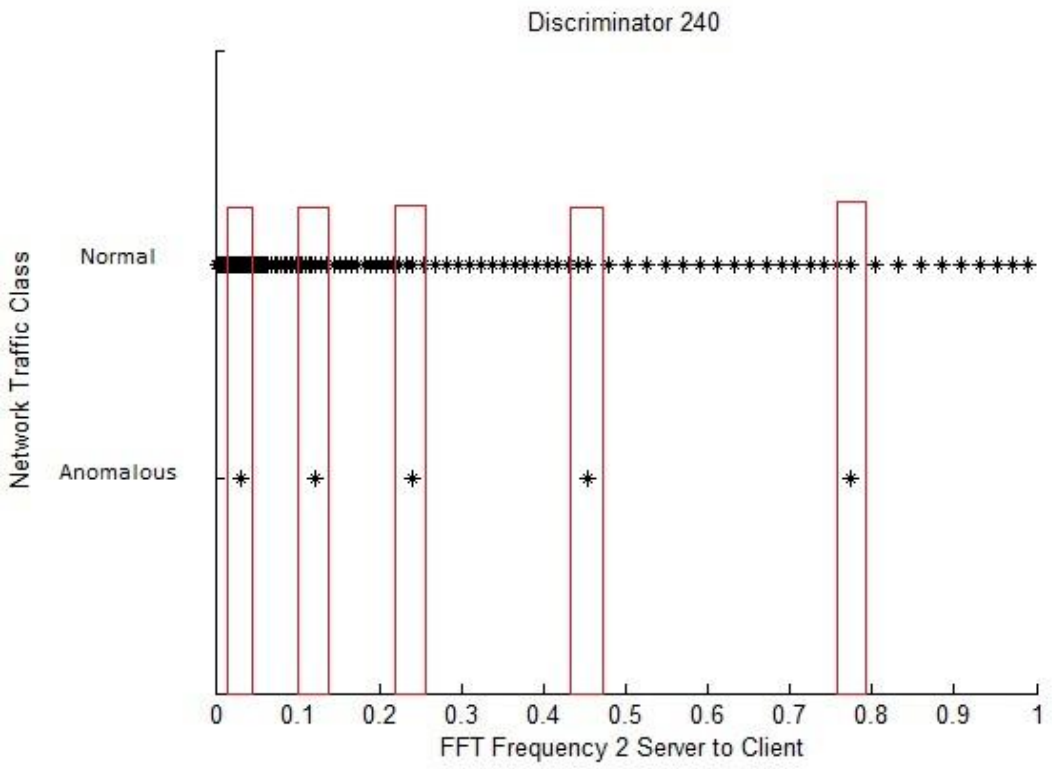

Figure 13: Overlap in IFA plot for Discriminator 240

4.3. Probability Distribution Function

None of the unique discriminators were found from the independent feature analysis (IFA). Now in order to measure the overlap that is found between the two output classes for the entire discriminator set, Probability Density Function (PDF) is one of the useful techniques. This discrete function actually measures the likelihood that the random variable (discriminator in this case) can take at a particular value. In order to perform repetitive operation over all the discriminators, another utility named "NADS_PDF_extractor" was developed. The overall process flow of this particular software is illustrated in the figure below.

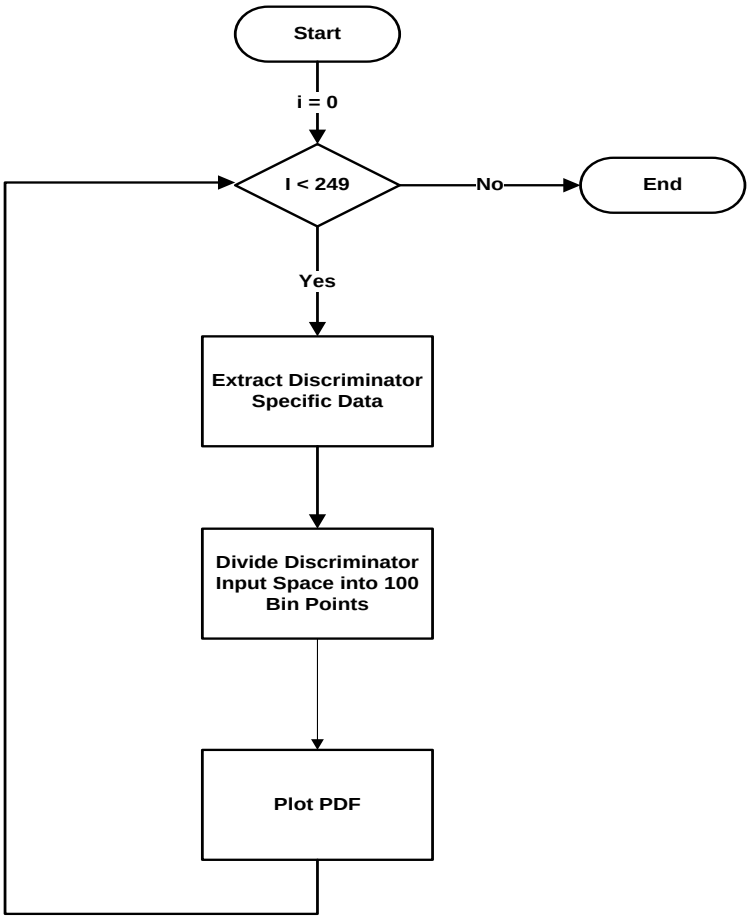

Figure 14: Probability Distribution Function Process Flow

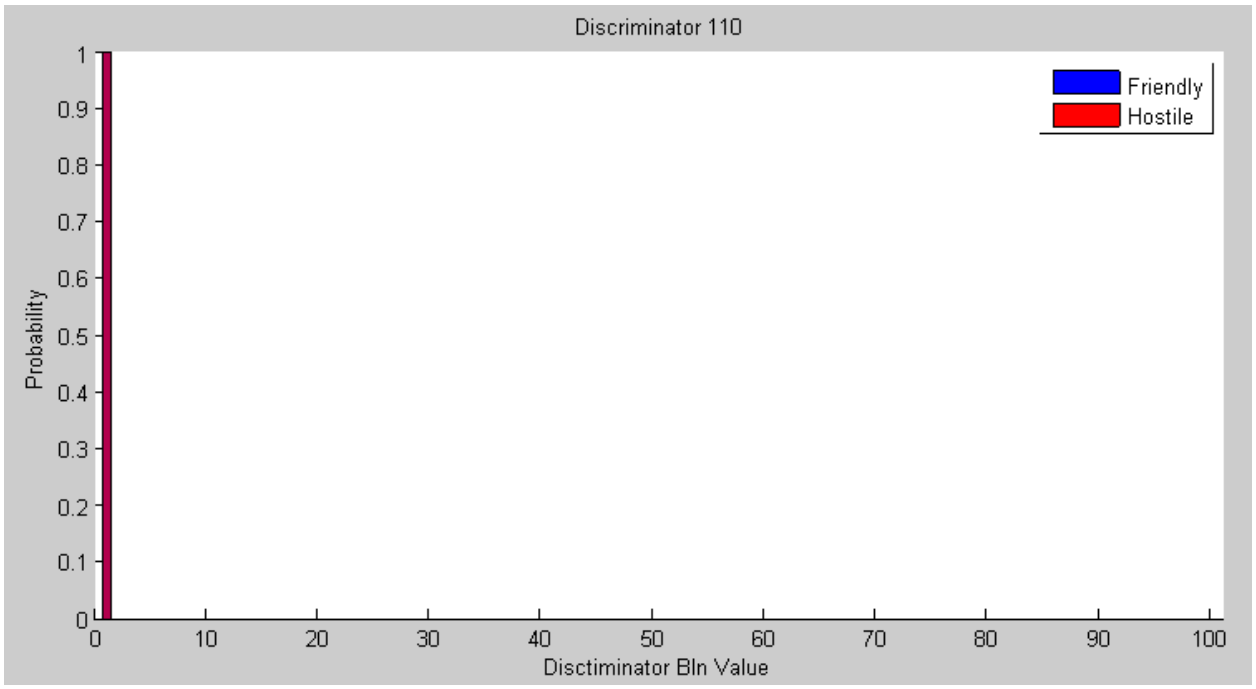

Figure 15: PDF plot for Discriminator 110

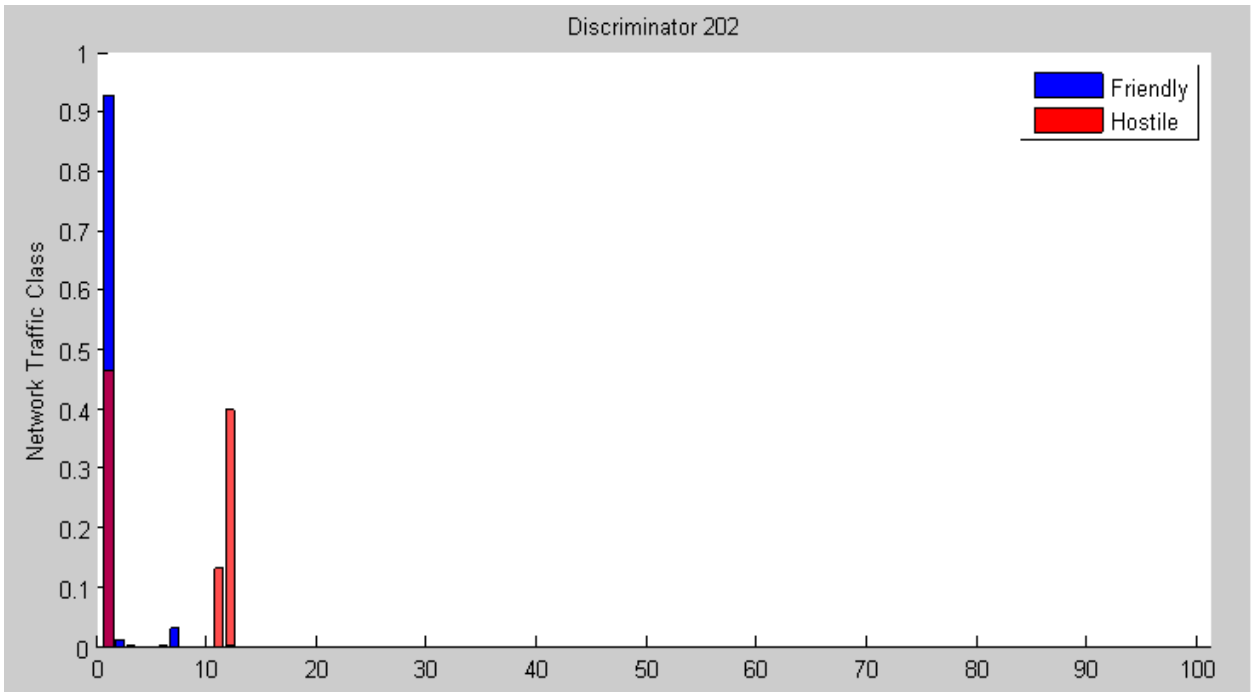

Figure 16: PDF plot for Discriminator 210

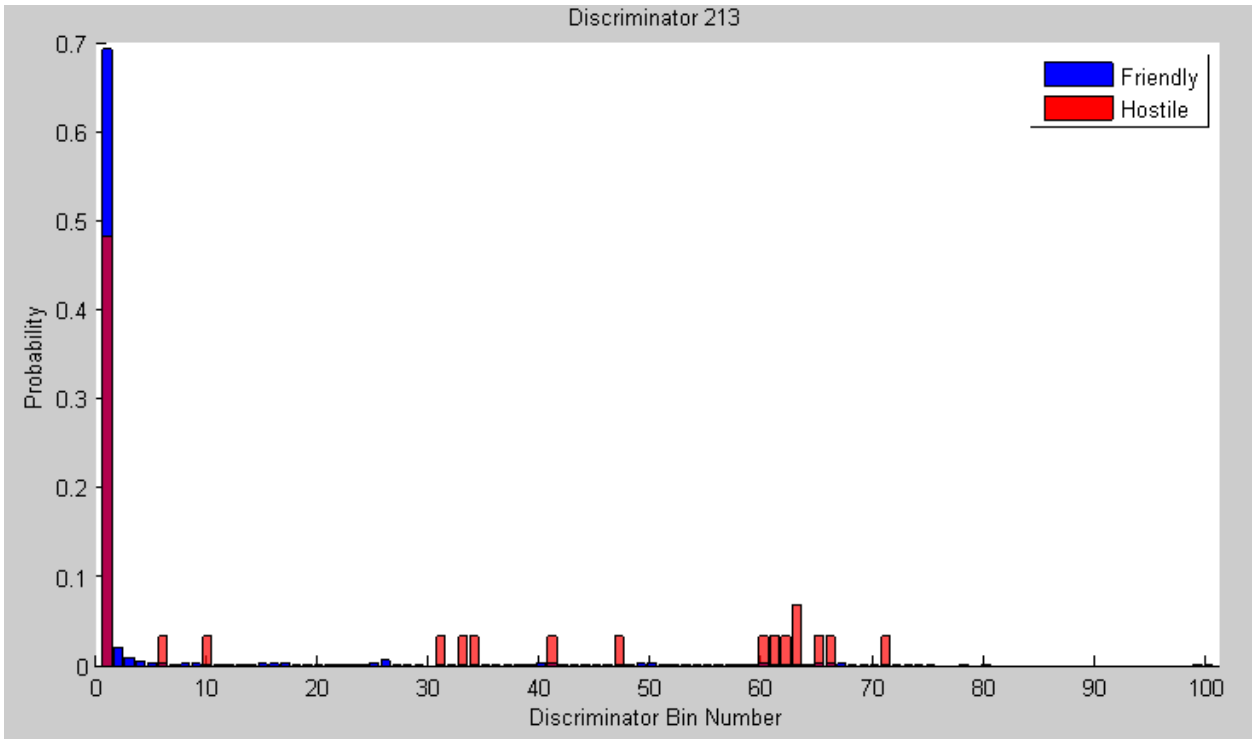

Figure 17: PDF plot for Discriminator 213

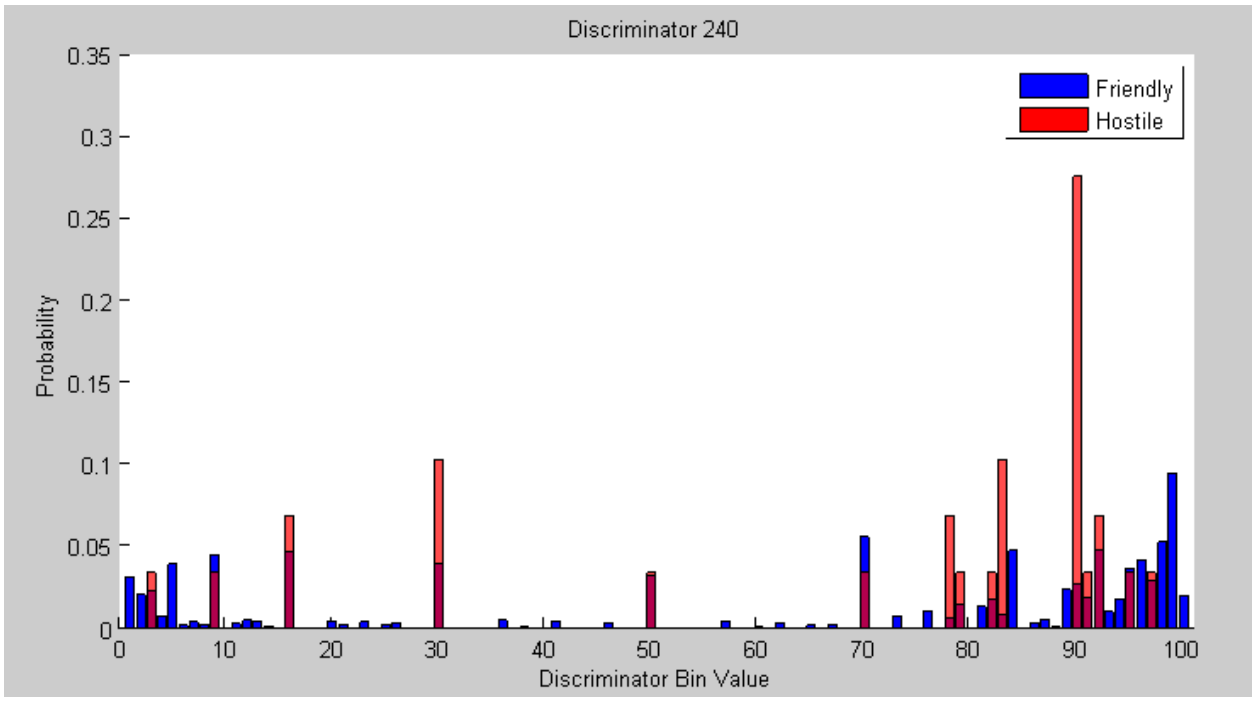

Figure 18: PDF plot for Discriminator 240

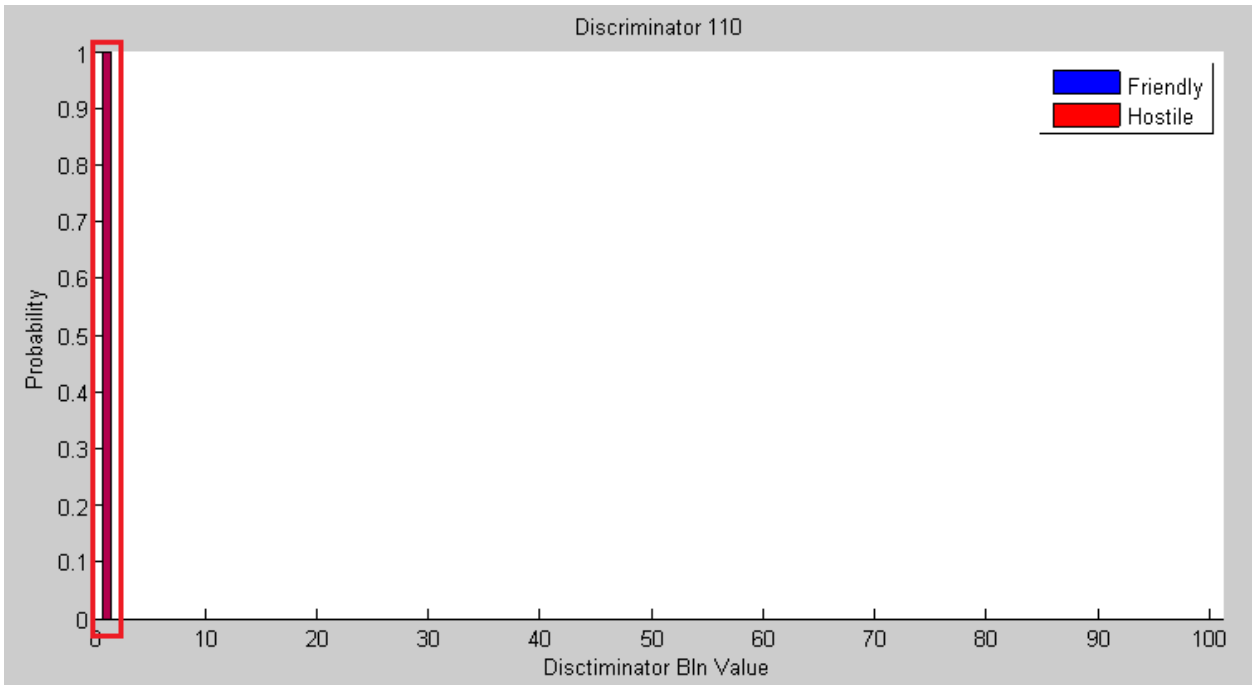

Figure 19: Overlap in PDF plot for Discriminator 110

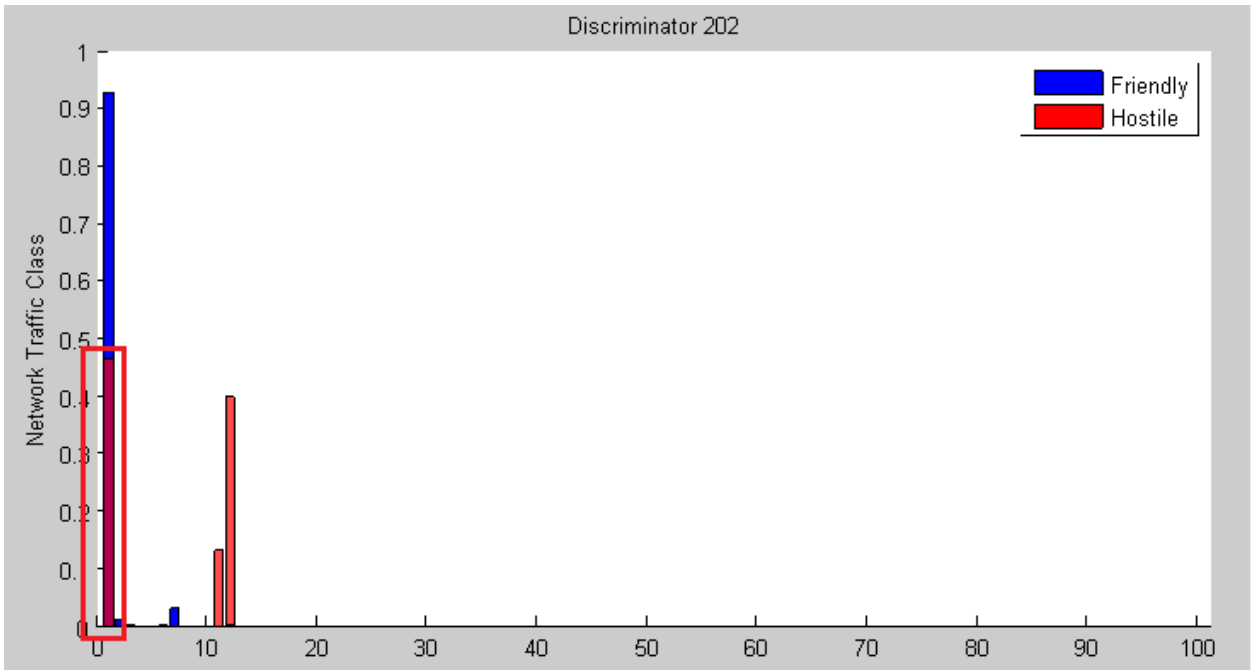

Figure 20: Overlap in PDF plot for Discriminator 202

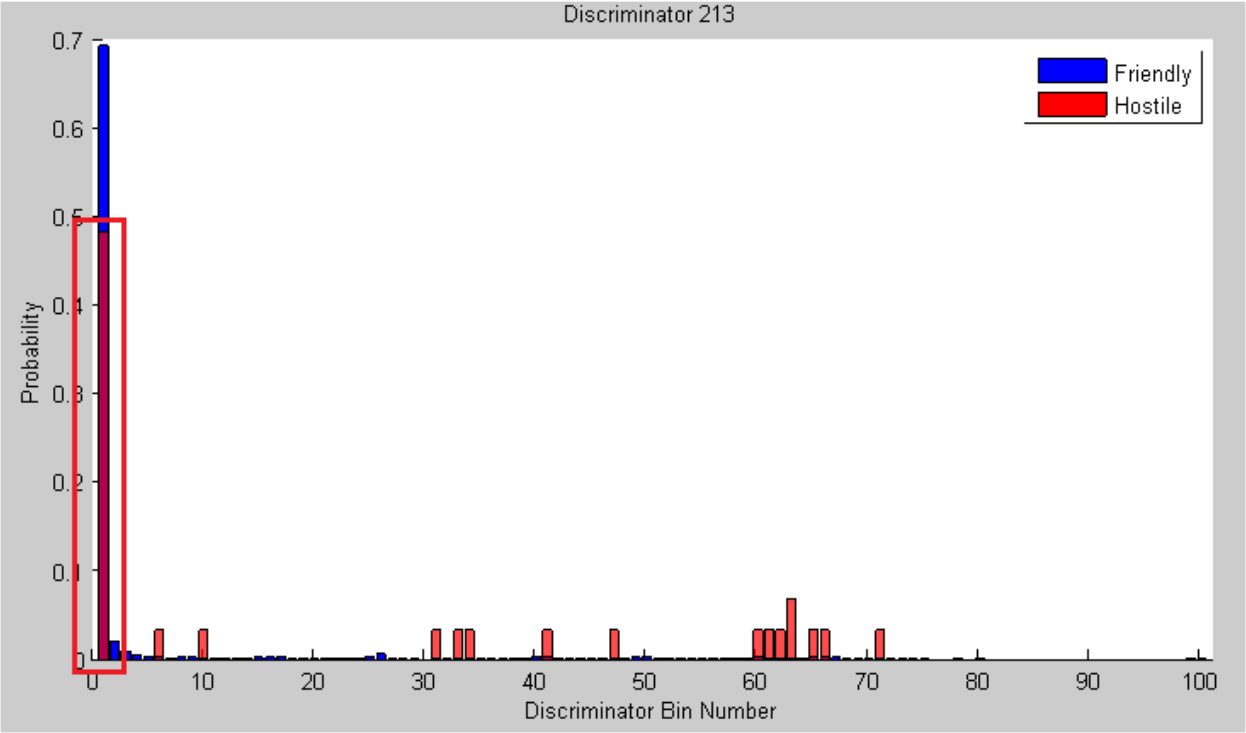

Figure 21: Overlap in PDF plot for Discriminator 213

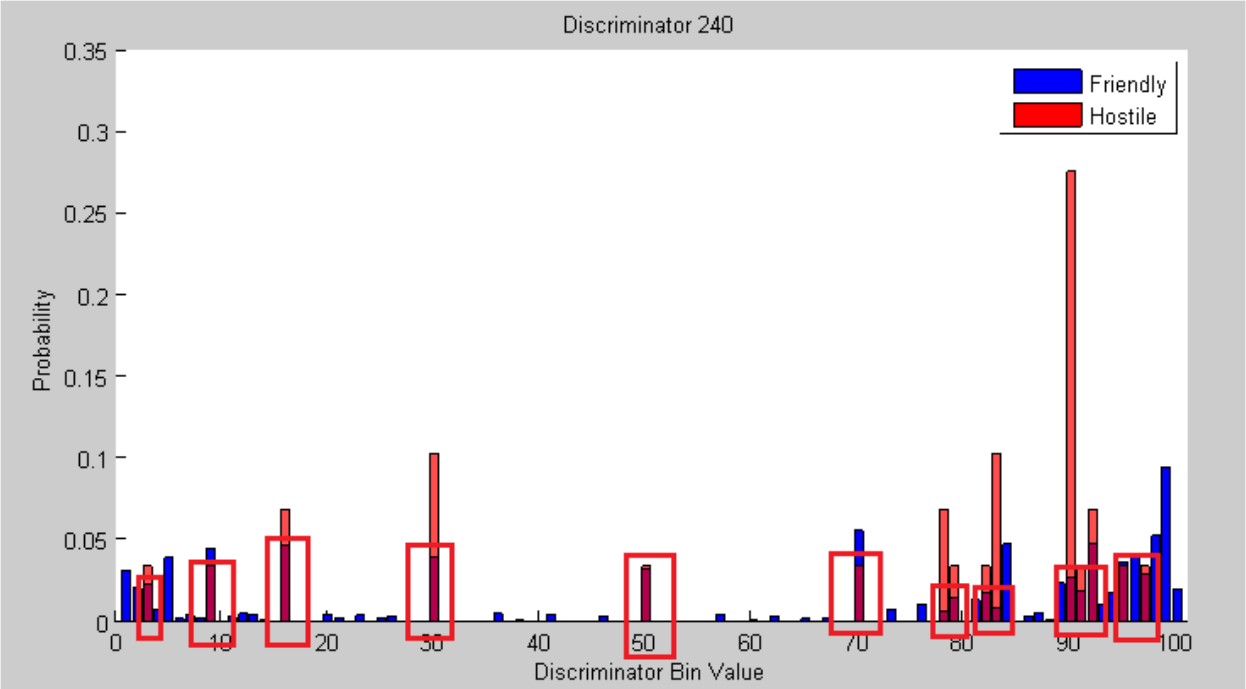

Figure 22: Overlap in PDF plot for Discriminator 240

CHAPTER 5

DIMENSIONALITY REDUCTION

5.1. Introduction

This chapter describes about the scoring techniques that has been utilized in this work for the dimensionality reduction of the classification process.

5.2. Feature Selection

With respect to the computer classification processes, high dimensional data is considered as a curse [16]. The high dimensional data sets have an adverse effect over both the time-space domains as it increases both the processing time and the memory requirements. Typically in a high dimensional data, it is quite common to have occurrence of a large number of irrelevant discriminators along with the presence of redundant discriminators. The only effect that these discriminators have is over fitting the classifiers. Feature selection methods aims to address this issue by making selection of subset features from the input high dimensional feature set. The following are the listed advantages of the feature selection methods.

i. Improving the performance of the ML algorithms

ii. Assistance in Data Visualization

iii. Reduction in Data Storage requirements

iv. Solving the problem with simpler methods

Generally, stochastic feature selection problem can be solved with three approaches [17].

i. Using Filter based Methods

ii. Using Wrapper based Methods

iii. Using Embedded Methods

5.2.1. Filter Based Methods

In Filter based methods, each feature is ranked with a scoring metric at the pre-processing stage. The features with higher score values are than selected for the classification process. These methods are adopted for supervised ML problems as filter based models depend upon the characteristics and attributes of the training data set.

Filter based Feature selection class comprises of a number of scoring techniques; some of them are listed below.

i. Fisher Score

ii. ReliefF

iii. Laplacian Score

iv. Hilbert Schmidt Independence Criterion (HSIC)

v. Trace Ratio Criterion

5.2.1.1. Fisher Score

Fisher Score tends to produce a scoring metric for each of the individual features or discriminators on the basis of the overlap. The features with distance between data points in different classes large and the distance between data points in same class small are assigned a higher Fisher Scoring metric.

Suppose the high dimensional data, $x \in R^d$, where **d** is the number of discriminators or features in input data. The output produced by Fisher Scoring technique is such that , $z \in R^m$, where $\mathbf{m} < \mathbf{d}$. Taking a

closer look at the problem, there is a total of d input features from which m features are to be extracted. This problem appears as a challenging combinatorial optimization problem with $\binom{\boldsymbol{d}}{\mathbf{m}}$ possible candidates for output variable Z [18]. The difficulty associated with the problem is overcome with the help of heuristic strategy, iterating through each feature independently (i.e. $\mathbf{x}^{\mathbf{j}} \in \boldsymbol{R}^{\boldsymbol{I}}$). In this case, we have $\binom{\boldsymbol{d}}{\mathbf{1}}$ candidates at a particular instant.

So this particular score can be computed with the help of the following equation

$$F(x^j) = \sum_{k=1}^{c} \frac{n_k(□_k{}^j - □^j)^2}{(\sigma^j)^2}$$

Where

j is the feature index number

c is the number of classes

μ is the mean

σ is the standard deviation

The standard deviation for the entire feature can be represented as

$$(\sigma^j)^2 = \sum_{k=0}^{c} n_k(\sigma_k{}^j)^2$$

For a binary class problem, the fisher scoring metric is listed as follows after performing some mathematical deductions.

$$F(x^j) = \frac{(□_2{}^j - □_1{}^j)^2}{(\sigma_2{}^j)^2 + (\sigma_2{}^j)^2}$$

5.2.1.1.1. Algorithm: Fisher Score Computation for Feature Selection

Input:

The selected feature number j,

Input Matrix X € R^d

Output:

The selected feature subset Z € R^m

Algorithm:

1. Extract Feature Information from ARFF formatted file
2. Calculate the score for each feature $F(x^j)$ as defined in above equation
3. Rank the features according to the score in descending order
4. Select the leading m features to form output data set.

5.2.1.2. Laplacian Score

Laplacian Score is another scoring algorithm that has its foundation over Locality Preserving Projection (LPP) which means that two data points belong to the same class if they are close to each other. This particular property makes this scoring technique more feasible for solving unsupervised ML problems.

Laplacian Score group data point into classes on the basis of an affinity matrix, K.

5.2.1.3. ReliefF

ReliefF is another feature selection algorithm that is used for binary classification [19]. This particular technique takes linear time in selecting number of features and training instances. Also this technique enables detection of conditional attributes in regression and classification [20]. However, this particular technique doesn't discriminate well redundant features.

5.2.1.4. Hilbert Schmidt Independence Criterion (HSIC)

HSIC is one of the techniques used for finding input feature subset primarily responsible for predicting the output [21] [22].

5.2.1.5. Trace Ratio Criterion

General scoring techniques require computation of score for each feature. However, the resulting sorted features don't necessarily guarantee the optimum subset-level score. This technique aims to efficiently find out the global optimal features such that the subset-level score is maximized [23].

5.2.2. Wrapper Based Methods

In Wrapper based methods, scores are assigned to features using the learning algorithms. These methods tend to produce better results as compared to filter based methods but are more computationally expensive.

5.2.3. Embedded Methods

In embedded based methods, the feature selection method doesn't exist as a separate entity and is more tightly coupled with the learning process.

Since our network anomaly detection problem is a supervised ML problem, this is intended to be solved using the Filter based methods. The subsequent section discusses Filter based methods in details.

5.3. Classification Algorithm

Conventionally, a classifier is defined as a function y = f(x), that maps input feature vectors x € X to output class y € {1, 2, 3, ……. C}, where C represents the total number of possible output labels for an experiment. This definition has a strong assumption that the output labels are mutually exclusive, that is occurrence of one label is independent of the occurrence of the other.

Like any of the supervised machine learning problem, the classifier plays a central role in the current anomaly detection problem in network traffic.

Broadly, the classifiers can be categorized as follows.

i. Probabilistic classifiers

ii. Tree classifiers

Off these different types of classifiers, this work makes use of one of the probabilistic classifiers, i.e. the Naïve Bayesian Classifier.

The selection of Naïve Bayesian classifier for benchmarking the reduced feature set has been done after a comprehensive analysis over five different classifiers [24].

i. Naïve Bayesian Classifier

Naïve Bayesian Classifier is based over one of the probabilistic classifiers with strong independence condition.

ii. C4.5 Decision Tree Classifier

C4.5 Decision Tree Classifier generates a decision tree from the set of training data on the concept of entropy

iii. Decision Table Classifier

This classifier simplifies the training data set and creates a set of logic decision over the attributes to determine the output class

iv. ZeroR Classifier

This is the simplest of classifiers which predicts the output class to belong to majority class

v. OneR Classifier

This class also belongs to rule based classifiers. This classifier predicts the output to the class with the best match based on the training data.

The benchmarking results obtained from the above analysis are listed in the table below.

Table 2: Benchmark Results for different Classifiers

Classifier	Accuracy (%)	Model Building Time (sec)
Naïve Bayesian Classifier	81.38	3.69
C4.5 Decision Tree	99.73	64.23
Decision Table	99.55	89.76
ZeroR	84.8	0.1
OneR	99.50	2.44

From the above analysis, it is evident that Naïve Bayesian, ZeroR and OneR Classifiers are more suitable for the adoption over embedded targets as the models generated are less time and memory complex. ZeroR has a disadvantage that it maps all the test instances to the majority class, this makes it infeasible for the network anomaly detection application. OneR classifier doesn't scale well with increasing features which makes it less suitable for the NADS problem with large number of discriminator. Therefore, the Naïve Bayesian classifier proved to be the most appropriate classifiers.

5.3.1. Naïve Bayesian Classifier

The Naïve Bayesian Classifier holds a strong assumption that all the features are conditionally independent for a class label [25].

$$\boldsymbol{P(X|Y=c) = \prod_{i=1}^{D} P(x_i|\, Y = c)}$$

The above equation gives "maximum likelihood" probability estimates, that is probability parameter values that maximizes the probability of the training examples. Although data in real environment doesn't always satisfy the above independence criterion, still the above Naïve Bayesian model works surprisingly well [26] [27].

5.3.2. Naïve Bayesian Classifier Implementation

As with other supervised ML classifiers, the Naïve Bayesian classifiers involved 2 stages.

i. Learning Phase
ii. Classification Phase

The entire system level diagram describing the entire Naïve Bayesian classification process is depicted in figure below.

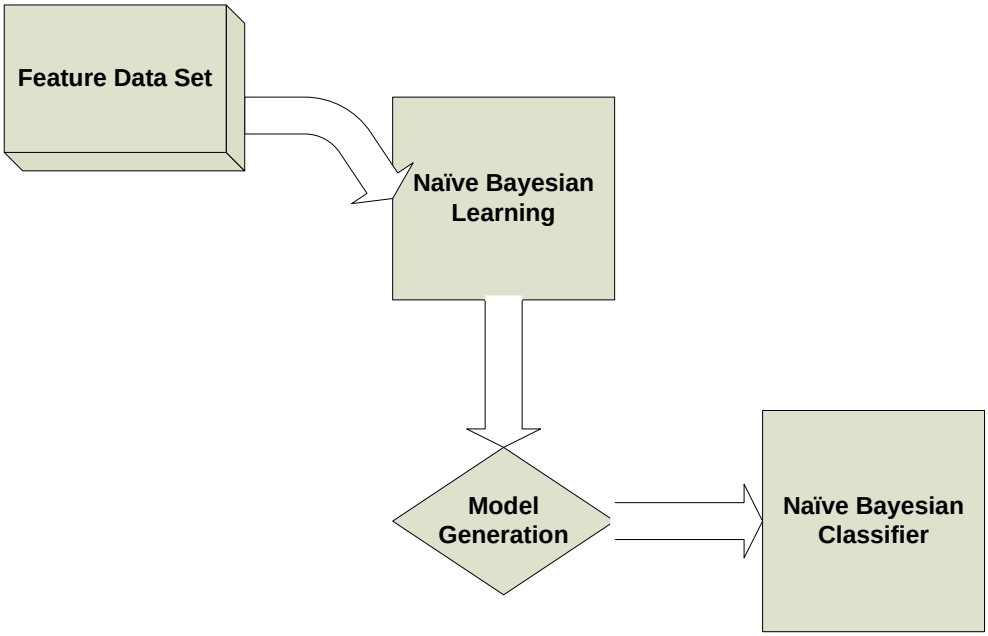

Figure 23: Naive Bayesian Classification Process

5.3.2.1. Naïve Bayesian Learning

Naïve Bayesian Learning algorithm formulates the probabilistic model from the input data set. This model is generated on the basis of Bayesian conditional probability theorem. This output model comprises of the following computed metrics.

i. Output Class Probabilities

ii. Conditional Feature Class Probabilities

In the learning phase, the probabilities for each of the output class are computed from the input data set. Once the class probabilities have been determined, the conditional feature probabilities with respect to class probabilities are computed. A pseudo algorithm describing the entire learning process is depicted in the figure below.

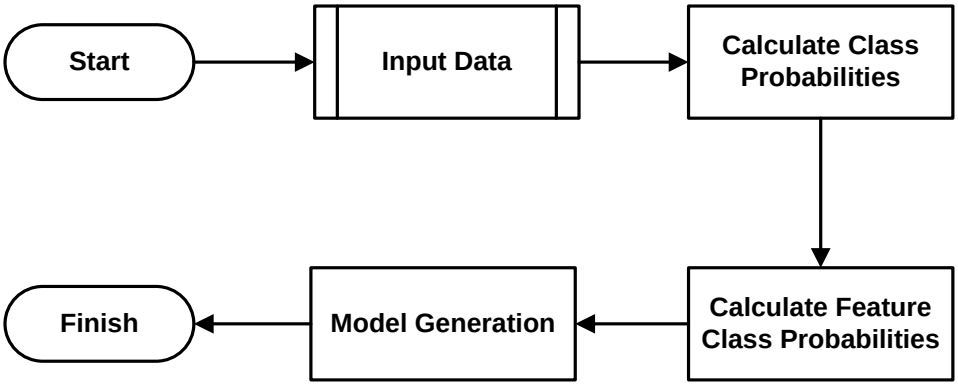

Figure 24: Naive Bayesian Learning Process

In this work the learning and classification process are separated by different process boundaries, therefore some interface need to be developed in between the classification and learning processes. This interface provided a mechanism of communicating the computed model metrics to the classification process. A file based interface has been developed that contains the output model information from the learning process.

5.3.2.2. Naïve Bayesian Classification

The first and foremost task of the classifier process is to load the model parameters into memory. Once the model is loaded, prediction score is computed on the basis of statistical feature values for each of the test instance. The test instance is assigned an output class on the basis of the prediction score and model data. A pseudo algorithm for the classification process is listed below.

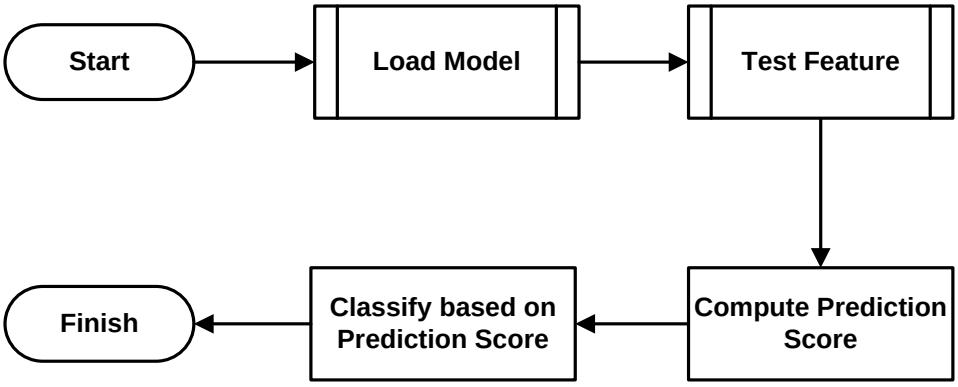

Figure 25: Naive Bayesian Classification Process

In the ML classifiers, there's a huge possibility to be caught by a large number of false positives. In order to mitigate such situation, it is always good to add some kind of threshold mechanism.

5.4. Classifier Benchmarking Parameters

The classifier bench marking parameters on the basis of which the classifier implementation will be evaluated is listed as under

5.4.1. Accuracy

The classifier efficiency is evaluated once the classifier is tested with online data. The following parameters have been used to evaluate the performance of different classification algorithms for an IDS application.

5.4.1.1. True Positive

True positive defines the flows correctly classified as an application class.

5.4.1.2. True Negative

True negative defines the flows correctly classified as not an application class.

5.4.1.3. False Positive

False Positive defines the flows incorrectly classified as an application class.

5.4.1.4. False Negative

False Negative defines the flows incorrectly classified as not an application class.

The above four factors together determine the accuracy of the classification algorithm.

5.4.2. Model Building Time

This corresponds to the time required by the classification algorithm to generate the baseline model from the training data set. This time has a direct relationship with the memory wise model complexity of the system. Generally, more complex operations require more processing time.

CHAPTER 6

SIMULATION RESULTS

6.1. Introduction

This chapter presents the results demonstrating the effectiveness of the dimensionality reduction techniques for decreasing the complexity of the anomaly detection process. The results produced have been tested with the Naïve Bayesian classifier. As per the testing criterion, around 50% of the samples have been used for verification purposes. The results have been tested over an Intel Core i5 processor with 4GB physical memory.

6.2. Fisher Score Computation Results

Fisher Score Metrics is computed for each of the discriminators or features independently as discussed in Chapter 5. The tables below lists the results for the top 25 fisher score computation results.

Table 3: Top 25 Fisher Score Metrics

Discriminator #	Discriminator Details	Fisher Score
88	Maximum Window Advertisement Seen from Server to Client	0.941469630862258
202	Minimum of Packet Inter Arrival Time from Server to Client	0.800478843772476
94	Average Window Advertisement Seen from Server to Client	0.769325467055373
165	Maximum of Total Bytes in IP Packet from Client to Server	0.513978715027258
158	Maximum of Bytes in Ethernet Packet from Client to Server	0.513978715027258
81	Maximum Segment Size Observed During the life-time of the connection from Server to Client	0.506895508884401
85	Average Segment Size Observed During the life-time of	0.440242145644163

	the connection from Client to Server	
95	Initial Window Size from Client to Server	0.335991410074874
90	Minimum Window Advertisement from Server to Client	0.308331028715019
163	Mean of total Bytes in IP packet from Client to Server	0.303763008239256
156	Mean of total Bytes in Ethernet packet from Client to Server	0.303763008239168
166	Variance of total Bytes in IP Packet from Client to Server	0.272947651339348
159	Variance of total Bytes in Ethernet Packet from Client to Server	0.272947651339348
83	Minimum Segment Size from Client to Server	0.244203069689799
84	Minimum Segment Size from Server to Client	0.234028559747396
96	Initial Window Size from Server to Client	0.208712952994593
70	Window Advertisement Scaling Factor from Server to Client	0.194122257034235
111	The average throughput calculated from client to Server	0.175545609655877
234	Fast Fourier Transform of Packet Inter Arrival Time from Client to Server	0.161589874293215
176	Median of Bytes in Ethernet Packet from Server to Client	0.161293840649441
183	Median of total Bytes in IP packets from Server to Client	0.161293678336868
126	Number of full size Round Trip Time Samples (RTT) from Server to Client	0.136083003982870
21	Third Quartile of total bytes in IP packet	0.135583331901671
14	Third Quartile of total bytes in Ethernet Packet	0.135583331901671
98	Initial Window Packets Seen from Server to Client	0.111474859735607

Profiling results for the Naïve Bayesian classifier show speedups with the reduction of discriminators as shown in Table 4.

Table 4: Profiling Results

Number of Discriminators	Model Building Time (sec)	Validation Time (sec)

248	8.74	10.71
124	4.94	5.43
100	3.97	4.42
62	1.8	2.56
50	1.71	2.06
31	0.94	1.33
20	0.71	0.96
10	0.5	0.68
5	0.2	0.38

As Naïve Bayesian Classifier is based over strong independence assumption between the statistical features, the validation time increases linearly as the number of features are increased as shown in Figure 9. The linear behaviour indicates that each feature contained in the feature set, requires equal amount of time in the Naïve Bayesian classification process.

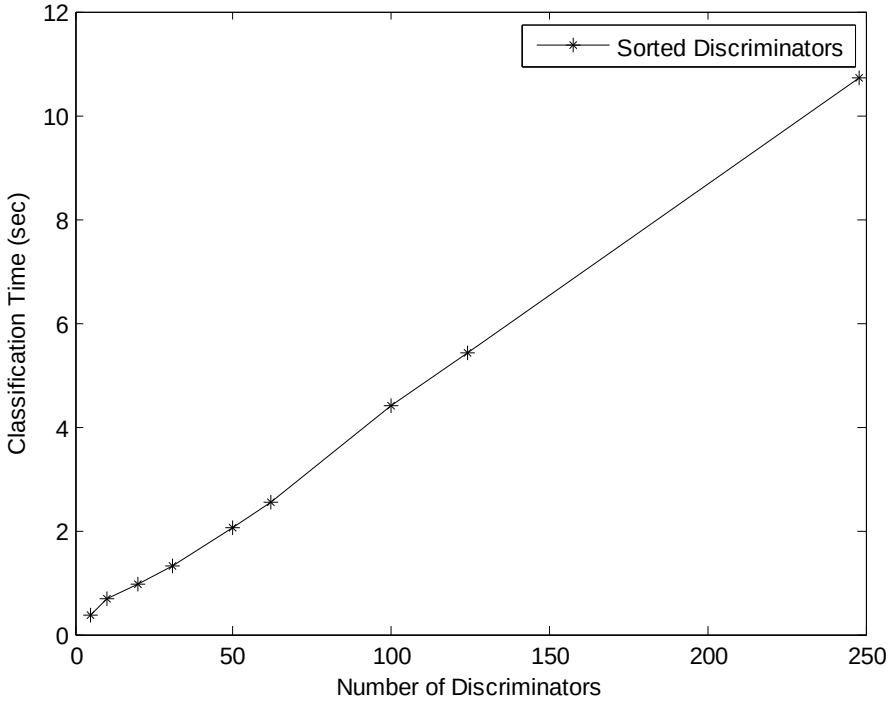

Figure 26: Classification Time vs Number of Discriminators

The accuracy results obtained from the experiment is presented in table 2.

Table 5: Classifier Accuracy

Number of Discriminators	Accuracy (Percentage)
248	99.9004
Top 124	99.8989
Top 100	99.8928
Top 62	99.8851
Top 50	99.8698
Top 31	99.6753
Top 20	99.6906
Top 10	99.4164
Top 5	95.4905

Bottom 5	53.8485

An analysis has been carried out over the trends with varying number of discriminators. A logarithmic relation has been found between the classifier accuracy and the number of discriminators as shown in Figure 10.

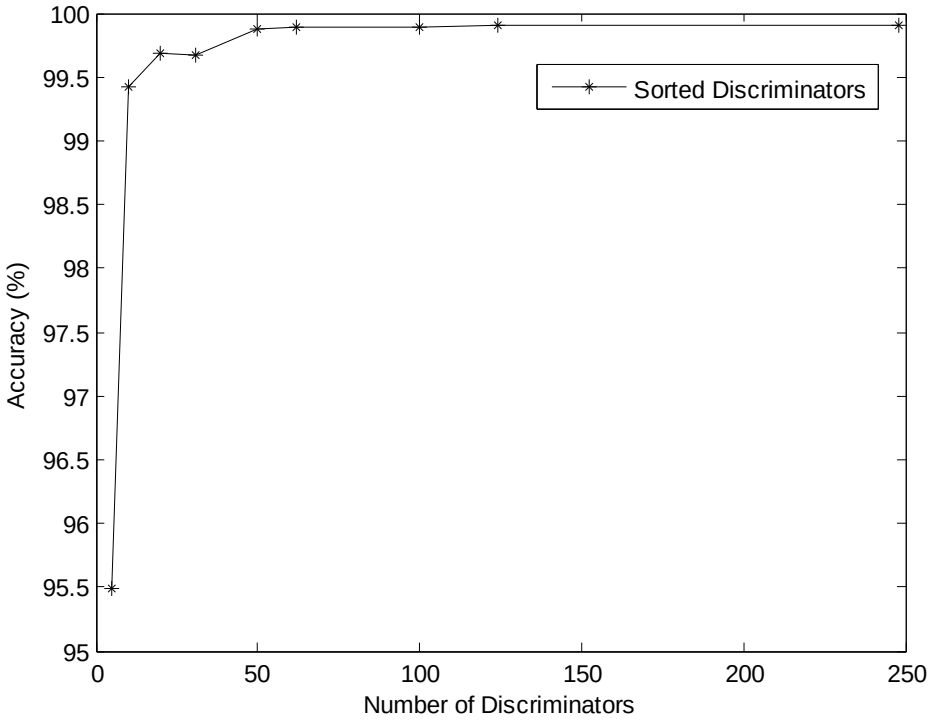

Figure 27: Classifier Accuracy vs. Number of Discriminators

The results clearly depict that the 5 discriminators with highest Fisher score constitute around 95.5% of classification information where-as the remaining 243 discriminators only add up to 5.4% of information to the classification process. Also accuracy of top 5 fisher score discriminators is compared with bottom 5 fisher score discriminators. It is evident that a higher Fisher score discriminator leads to better accuracy. The advantage of reducing the number of

discriminators resulted in significant decrease in the model building and classification time of the network traffic.

CHAPTER 7

BENCHMARKING OVER EMBEDDED TARGET

7.1. Introduction

Based on the simulation results, it was decided to benchmark the results over an embedded target. There are a number of readily available devices available ranging from micro-controllers to micro-processors. However, the natural choice has been the selection of micro-processor based target due to minimal efforts involved in porting the existing algorithm.

There are a number of micro-processor based embedded targets including;

1. Raspberry PI
2. Beagle Bone
3. Jetson Tegra K1

Of the many choices, the selection has been in favour of Raspberry PI, as it was readily available in the market with minimum cost.

7.2. Embedded Target Specifications

Figure 26 presents an image of the Raspberry PI target [28], with the specifications listed as under.

- 700MHz ARM processor
- 512 MB RAM

- Two USB ports
- 100 Mb Ethernet

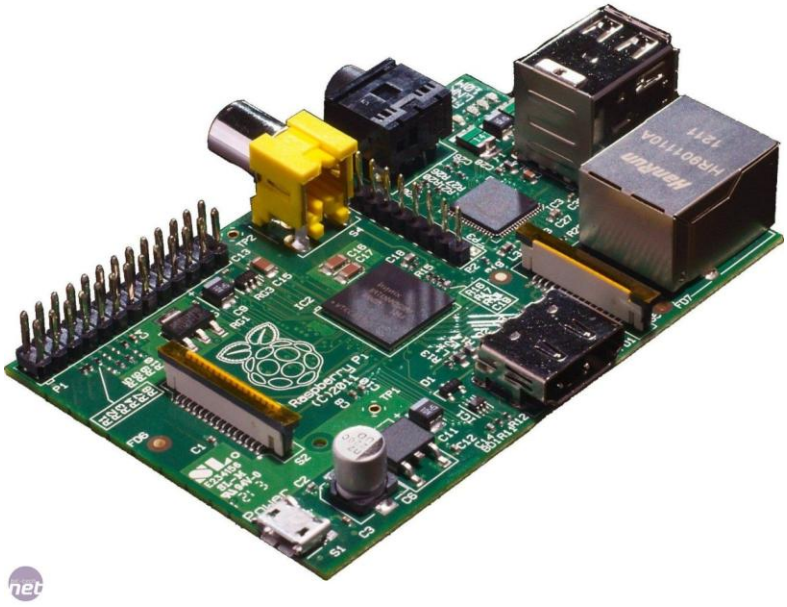

Figure 28: Raspberry PI rev B

7.3. Embedded Target Benchmark Results

The entire algorithm has been written in C++ so as to make the implementation portable to different architectures. This ability enabled us to benchmark the results without making changes to the classifier source code.

The following timing results have been observed by carrying out experiment over Raspberry PI target. The classifier implementation wasn't benchmarked for large number of discriminators because of the limited available memory.

Table 6: Benchmarking Results over Embedded Target

Number of Discriminators	Model Building Time (sec)	Validation Time (sec)
5	0.8	25
10	1.5	51.4
20	3.5	106.5

CHAPTER 8

CONCLUSION & RECOMMENDATIONS

8.1. Conclusion

Network Anomaly Detection is a field that is yielding wider importance in the scientific and engineering community. However, intelligently choosing the statistical discriminators in this process is vital to gain maximum results. Choosing a large number of statistical discriminators causes a performance penalty with an increase in complexity of the entire process.

Based on the scoring metrics, we have successfully reduced the complexity of the entire classification process in the NADS system. The results clearly indicate that removing less significant features from the feature vector set assists in accelerating the entire classification process with a little impact over the accuracy of the classifier. The memory footprint of the entire process also reduces with a reduction in the dimension of the process.

8.2. Future Work

There are a number of directions available for extending the current work in the future. Currently, independence in between different statistical discriminators has been the prime assumption. One future improvement includes addressing dependence in between different features.

Apart from this, acceleration of the entire classification algorithm is intended by using massive parallelism of the General Purpose Graphical Processing Units (GPGPUs).

BIBLIOGRAPHY

[1] Barry M. Leiner et al., "A Brief History of the Internet," *Computer Communication Review*, October 2009.

[2] J. Dressler, "United States v. Morris," in *Cases and Materials on Criminal Law*., 2007.

[3] SOPHOS, "Security Threat Report 2013," 2013.

[4] Christopher M. Bishop, *Pattern Recognition and Machine Learning*.: Springer, 2006.

[5] Animesh Patcha and Jung-Min Park, "An overview of anomaly detection techniques: Existing solutions and latest technological trends," *The International Journal of Computer and Telecommunications Networking*, vol. 51, no. 12, pp. 3448-3470, August 2007.

[6] M-L Shyu, S-C Chen, K Sarinnapakorn, and L Chang, "A novel anomaly detection scheme based on principal component classifier," in *IEEE Foundations and New Directions of Data Mining Workshop*, 2003, pp. 171-179.

[7] Yacine Bouzida, Frédéric Cuppens, Nora Cuppens-Boulahia, and Sylvain Gombault, "Efficient Intrusion Detection Using PrincipalComponent Analysis," in *3ème Conférence sur la Sécurité et Architectures Réseaux*, La Londe, 2004.

[8] Wei Wang, Xiaohong Guan, and Xiangliang Zhang, "A Novel Intrusion Detection Method Based on Principle Component Analysis in Computer Security," *Advances in Neural Networks - ISNN 2004*, vol. 3174, pp. 657-662, 2004.

[9] Xiaoshu Hang and Honghua Dai, "Applying both Positive and Negative Selection to Supervised Learning for Anomaly Detection," in *Proceedings of Genetic and Evolutionary Computation*, 2005.

[10] R. Gentleman and V. J. Carey, "Unsupervised Machine Learning," in *Bioconductor Case Studies*.: Springer, 2008.

[11] Andrew Moore, Denis Zeuv, and Michael Crogan, "Discriminators for use in flow-based classification," Queen Mary University of London, London, ISSN 1470-5559, 2005.

[12] Tony Pisegna, "A Benchmarking Technique to Identify Best Practices in SAS.," in *NESUG*, 2010.

[13] Wei Li, Marco Canini, Andrew W. Moore, and Raffaele Bolla, "Efficient Application Identification and the Temporal and Spatial Stability of Classification Schema," *Computer Networks*, vol. 53, no. 6, pp.

790-809, April 2009.

[14] Andrew W. Moore and Denis Zuev, "Internet traffic classification using bayesian analysis techniques," in *SIGMETRICS '05 Proceedings of the 2005 ACM SIGMETRICS international conference on Measurement and modeling of computer systems*, 2005, pp. 50-60.

[15] Stuart Russell and Peter Norvig, *Artificial Intelligence: A Modern Approach*, 3rd ed.: Prentice Hall, 2009.

[16] Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*, 2nd ed.: Wiley, 2001.

[17] Noelia Sánchez-Maroño, Amparo Alonso-Betanzos, and María Tombilla-Sanromán, "Filter Methods for Feature Selection – A Comparative Study," *Intelligent Data Engineering and Automated Learning - IDEAL 2007*, vol. 4881, pp. 178-187, 2007.

[18] Quanquan Gu, Zhenhui Li, and Jiawei Han, "Generalized Fisher Score for Feature Selection," in *Conference on Uncertainity in Artificial Intelligence*, Barcelona, 2011, pp. 266-273.

[19] Kenji Kira and Larry A. Rendell, "The Feature Selection Problem: Traditional Methods and a New Algorithm," *AAAI Proceedings*, 1992.

[20] Marko Robnik-Sikonja and Igor Kononenko, "Theoretical and Empirarical Analysis of ReliefF and RReliefF," *Machine Learning*, vol. 53, pp. 23-69, October 2003.

[21] Arthur Gretton, Olivier Bousquet, Alex Smola, and Bernhard Scholkopf, "Measuring Statistical Dependence with," *Lecture Notes in Computer Science*, vol. 3734, pp. 63-77, 2005.

[22] Makoto Yamada, W. Jitkrittum, L. Sigal, E. P. Xing, and M. Sugiyama, "High-Dimensional Feature Selection by Feature-Wise Non-Linear Lasso," *Neural Computation*, vol. 26, pp. 185-207, 2014.

[23] Feiping Nie, Shiming Xiang, Yangqing Jia, Changshui Zhang, and Shuicheng Yan, "Trace Ratio Criterion for Feature Selection," in *Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence*, 2008, pp. 671-676.

[24] Muhammad Junaid Muzammil, Sameer Qazi, and Taha Ali, "Comparative analysis of classification algorithms performance for statistical based intrusion detection system," in *3rd International Conference on Computer,Control & Communication (IC4)*, Karachi, 2013, pp. 1 - 6.

[25] Kevin P. Murphy, "Naive Bayesian Classifiers," October 2006.

[26] K. Ming Leung, "Naive Bayesian Classifier," November 2007.

[27] Ion Androutsopoulos, John Koutsias, Konstantinos V. Chandrinos, George Paliouras, and

Constantine D. Spyropoulos, "An evaluation of Naive Bayesian anti-spam filtering," *Proceedings of the workshop on Machine Learning in the New Information Age*, pp. 9-17, 2000.

[28] Bit-Tech. [Online]. http://images.bit-tech.net/content_images/2013/03/raspberry-pi-case-competition-update/pi1l.jpg

[29] Eleazar Eskin, Andrew Arnold, Michael Prerau, and Leonid Portnoy, "A Geometric Framework for Unsupervised Anomaly Detection: Detecting Intrusions in Unlabeled Data," in *Applications of Data Mining in Computer Security*, 2002.