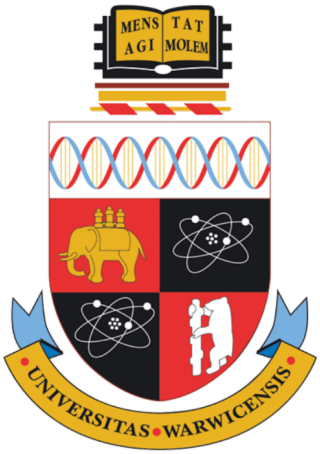


An Intrusion Detection Scheme for Identifying Known and Unknown Web Attacks (I-WEB)

By

Muhammad Hilmi Kamarudin

A dissertation submitted in fulfilment of the requirements for the degree of Doctor of Philosophy

At

WMG

University of Warwick

April 2018

Abstract

The number of utilised features could increase the system's computational effort when processing large network traffic. In reality, it is pointless to use all features considering that redundant or irrelevant features would deteriorate the detection performance. Meanwhile, statistical approaches are extensively practised in the Anomaly Based Detection System (ABDS) environment. These statistical techniques do not require any prior knowledge on attack traffic; this advantage has therefore attracted many researchers to employ this method. Nevertheless, the performance is still unsatisfactory since it produces high false detection rates. In recent years, the demand for data mining (DM) techniques in the field of anomaly detection has significantly increased. Even though this approach could distinguish normal and attack behaviour effectively, the performance (true positive, true negative, false positive and false negative) is still not achieving the expected improvement rate. Moreover, the need to re-initiate the whole learning procedure, despite the attack traffic having previously been detected, seems to contribute to the poor system performance.

This study aims to improve the detection of normal and abnormal traffic by determining the prominent features and recognising the outlier data points more precisely. To achieve this objective, the study proposes a novel Intrusion Detection Scheme for Identifying Known and Unknown Web Attacks (I-WEB) which combines various strategies and methods. The proposed I-WEB is divided into three phases namely pre-processing, anomaly detection and post-processing. In the pre-processing phase, the strengths of both filter and wrapper procedures are combined to select the optimal set of features. In the filter, Correlation-based Feature Selection (CFS) is proposed, whereas the Random Forest (RF) classifier is chosen to evaluate feature subsets in wrapper procedures. In the anomaly detection phase, the statistical analysis is used to formulate a normal profile as well as calculate the traffic normality score for every traffic. The threshold measurement is defined using Euclidean Distance (ED) alongside the Chebyshev Inequality Theorem (CIT) with the aim of improving the attack recognition rate by eliminating the set of outlier data points accurately. To improve the attack identification and reduce the misclassification rates that are first detected by statistical analysis, ensemble-learning particularly using a boosting classifier is proposed. This method uses using LogitBoost as the meta-classifier and RF as the base-classifier. Furthermore, verified attack traffic detected by ensemble learning is then extracted and computed as signatures before storing it in the signature library for future identification. This helps to reduce the detection time since similar traffic behaviour will not have to be re-executed in future.

The I-WEB performance is evaluated with different sets of performance metrics (True Positive, True Negative, False Positive, False Negative, False Alarm Rate, False Negative Rate, Attack Detection Rate, Normal Detection Rate, Accuracy Rate and Detection Time), including four publicly available benchmark intrusion detection datasets (DARPA 1999, NSL KDD, ISCX 2012 and UNSW-NB15). The experimental results demonstrate that I-WEB is comparable, accurate and more efficient in detecting both known and unknown attacks compared to the traditional approaches. In addition, the detection time is significantly reduced when the attack signature is employed as part of the detection strategy. Thus, I-WEB is a better solution for anomaly detection in detecting both known and unknown web attack traffic.

Table of Contents

List of Tables

List of Figures

Abbreviations

ABDS	Anomaly Based Detection System
AB-RF	AdaBoost and Random Forest
ACC	Accuracy
ACCS	Australian Centre for Cyber Security
A-DR	Attack Detection Rate
AFRL	Air Force Research Laboratory
AIS	Artificial Immune Systems
ANN	Artificial Neural Network
ANFIS	Adaptive Neural Fuzzy Inference System
AP	Active Point
AS	Active Score
BER	Backward Elimination Ranking
BS	British Standard
CAGE	Cellular Genetic Programing
CART	Classification and Regression Tree
CFS	Correlation Feature Selection
CIA	Confidentiality, Integrity and Availability
CIT	Chebyshev Inequality Theorem
DBMS	Database Management System
DoS	Denial of Service
DDoS	Distributed Denial of Service
DM	Data Mining
DNS	Domain Name Server
DS	Dempster-Shafer
DT	Decision Table
EAFB	Eyrie Air Force Base
ED	Euclidean Distance

ESVDF	Enhanced Support Vector Decision Function
FAR	False Alarm Rate
FBFR	Filter-based Feature Ranking
FBSE	Filter-based Subset Evaluation
FBI	Federal Bureau of Investigation
FL	Fuzzy Logic
FP	False Positive
FN	False Negative
FSR	Forward Selection Ranking
GA	Genetic Algorithm
GP	Genetic Programming
HFS	Hybrid Feature Selection
HIDS	Host-based Intrusion Detection System
HMM	Hidden Markov Model
HMP	Hybrid Multilayer Perceptron
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
ICMP	Internet Control Message Protocol
IDS	Intrusion Detection System
IGR	Information Gain Ratio
IoT	Internet of Things
IPM	Intrusion Prioritisation Model
IPSec	Internet Protocol Security
ISO	International Organization for Standardization
ISMS	Information Security Management System
I-WEB	Intrusion Detection Scheme for Identifying Known and Unknown Web Attacks
J48	Decision Tree
LGP	Linear Genetic Programming
LR	Logistic Regression

LNID	Lightweight Network Intrusion Detection
LB-RF	LogitBoost and Random Forest
MARS	Multivariate Adaptive Regression Splines
MBDS	Misuse Based Detection System
MBP	Multilayer Backpropagation Perceptron
MLP	Multilayer Perceptron
NAT	Network Address Translation
NB	Naïve Bayes
N-DR	Normal Detection Rate
NN	Neural Network
NHS	National Health Service
NIDS	Network-based Intrusion Detection System
NSA	National Security Agency
NSL KDD	Network Security Laboratory Knowledge Discovery and Data Mining
OWASP	Open Web Application Security Project
PART	Partial Decision Tree
PbPHAD	Protocol-based Packet Header Anomaly Detection
PHAD	Packet Header Anomaly Detection
PP	Passive Point
PS	Passive Score
PSO	Particle Swarm Optimization
RF	Random Forests
R2L	Remote to Local
RIDOR	Ripple Down Rule
SAD	Statistical Analysis Detection
SBAD	Statistical Based Anomaly Detection
SMB	Server Message Block
SQL	Sequel Query Language
SSH	Secure Shell
SSL	Secure Socket Layer

SVM	Support Vector Machines
TCP	Transmission Control Protocol
TELNET	Terminal Network
TN	True Negative
TP	True Positive
UDP	User Datagram Protocol
UNB	University of Brunswick
US-CERT	United State Computer Emergency Readiness Team
U2R	User to Root
WBSE	Wrapper-based Subset Evaluation
WEP	Wired Equivalent Privacy
WEKA	Waikato Environment for Knowledge Analysis
XSS	Cross-Site Scripting
5-NN	Five-Nearest Neighbours

Declarations

I hereby declare that this dissertation entitled ***An Intrusion Detection Scheme In Identifying Known and Unknown Web Attacks (I-Web)*** is an original work and has not been submitted for a degree or diploma or other qualification at any other University

Coventry, United Kingdom

Acknowledgement

First and foremost, all praise and gratitude shall be bestowed to Allah the Almighty and The Merciful for all the insight given by Him which has led to the completion of this research.

I would like to express my sincerest gratitude and appreciation to my supervisors Prof. Carsten Maple, Prof. Tim Watson and Dr Nader Sohrabi Safa for their valuable guidance, support and motivation throughout my PhD journey. This thesis will not even be possible without their continues encouragement, enthusiasm and positive critics.

To my source of inspiration, my wife Hasliza Sofian, my son Muhammad Ashraf and for the one who is coming soon, I am truly thankful for their encouragements, patients, supports, and understanding throughout this endeavour. Not to forget, the deepest appreciation goes to my father Mr Kamarudin, my mother Mrs Siti Johariah, father-in-law Mr Sofian, and mother-in-law Mrs Sharifah Robiah for their blessing and endless support throughout this study.

I would also like to thank my friends, colleagues at the Cyber Security Centre, WMG, University of Warwick, and to all individuals who are involved either directly or indirectly in making this project successful.

List of Publications

Conference:

1. Kamarudin, M.H., Maple, C., Watson, T., and Sofian, H., (2016). Packet Header Intrusion Detection with Binary Logistic Regression Approach in Detecting R2L and U2R Attacks. *Proceedings - 4th International Conference on Cyber Security, Cyber Warfare, and Digital Forensics, CyberSec 2015*, pp.101–106.

Journal:

1. Kamarudin, M.H., Maple, C. and Watson, T., (2016). Hybrid feature selection technique for intrusion detection system. *Int. J. High Performance Computing and Networking*.
2. Kamarudin, M.H., Maple, C., Watson, T., and Safa, N.S. (2017). A LogitBoost-based Algorithm for Detecting Known and Unknown Web Attacks. *IEEE Access*, *5*, 26190–26200. http://doi.org/10.1109/ACCESS.2017.2766844
3. Kamarudin, M.H., Maple, C., Watson, T., and Safa, N.S. (2017). A New Unified Intrusion Anomaly Detection in Identifying Unseen Web Attacks. *Security and Communication Networks*, *2017*, 1–18. http://doi.org/10.1155/2017/2539034

Chapter 1

Introduction

1.1 Background

The continuous growth of the Internet technologies and massive data exchange have created a new paradigm named big data. One of the challenges is when a huge amount of sensitive information stored in the servers and transmitted over the Internet becomes a primary target. Web-based applications and web servers have been a popular target in recent years considering that most communications involving client-server queries. A comprehensive analysis carried out by Symantec (2017) revealed that 76% of websites were scanned and found to have vulnerabilities, where 9% fall into the critical category.

Cyber-attack is usually performed by unethical users, either from organisations or individuals, against vulnerable systems such as computer systems, network infrastructures or business information, with the intention of modifying, destroying or stealing information. The types of problems caused by cyber-attack include web defacement, denial of service, password stealing and root access. For example, the attack on 21st October 2016 was specifically designed to target Dyn, a major Internet infrastructure company (Cyber Attacks, 2016). The attack is recognised as one of the largest attacks with millions of source IP addresses used to request DNS lookup. Dyn is responsible for providing DNS service translations, i.e. translating human-friendly site names into machine-readable Internet addresses. The attack nearly brought down the entire US Internet service. Vulnerable Internet of Things (IoT) devices such as webcams and digital videos can be used to distribute malicious software and spam. Mirai is an example of software that was designed to exploit the vulnerabilities in IoT devices by infecting them. The infected devices were turned into slave or zombie

devices and formed an army of bots that was used to perform large scale Distributed Denial of Services (DDOS) attacks from multiple different locations. The attacks caused outages and slowness for many of Dyn's customers including Twitter, Paypal, CNN, and some businesses hosted by Amazon.com Inc.

A more recent massive cyber-attack took place on 12th May 2017 and had a major impact on a significant element of the UK's National Health Service (NHS), other health industries and created chaos in hospitals across England (Jones, 2017). Thousands of computers at hospitals and GPs' surgeries became victims of global ransomware attacks, derivatives of the WannaCry attack, which are believed to have exploited a vulnerability first discovered by the National Security Agency (NSA). In particular, the attack exploited a vulnerability in the Windows Server Message Block (SMB) protocol and installed backdoor tools to deliver and run a WannaCry ransomware package. Although the Internet is widely recognised as a convenient means for providing real-time information services to the public, the potential threats to confidentiality, integrity and availability (CIA) issues need to be addressed more effectively and permanently (Thakare and Gore, 2014).

1.2 Motivation

In the early Internet era, most hackers needed a high knowledge level to assist them in developing their own methods to break into a system. Unfortunately, the existence of readily available intrusion and hacking tools has allowed almost anyone to initiate an attack. Today, in digital communications, cyber-attack can be compared to a missile used during the war. Figure 1.1 indicates cyber-attack events collected from the global major cyber events among the published open sources. The timeliness produced by Passeri (2017) illustrates an overview of the threat landscape. For instance, the total cyber-attack events have significantly increased by 182 cyber-attack events (with total of 1061 cyber-attack events) in the year 2016, compared with 2014 (with total of 879 cyber-attack events).

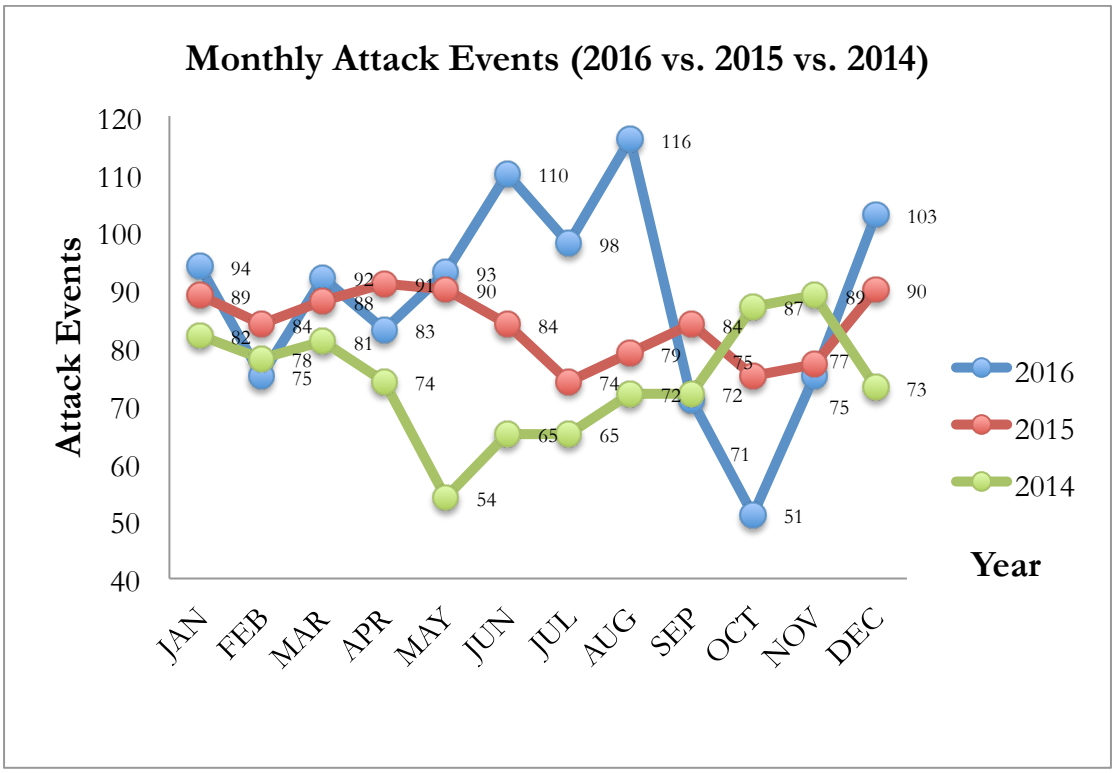


Figure 1.1: Monthly Attack Events Activity for Year 2014, 2015 and 2016

One of the biggest concerns of information security is to protect or defend the information infrastructure (Tirenin and Faatz, 1999). This further explains the need to identify the source of threats and then analyse them for future preventive action. The process complies with the International Organization for Standardization (ISO) 17799 and the identical British Standard (BS) 7799 that act as the codes of practice for information security management systems (ISMS) (Chan *et al.,* 2005). Recent developments of the Internet have given rise to the enhanced capability of the Internet of Things (IoT) to ease people's lives, particularly in solving issues related to communication, financial and time constraints.

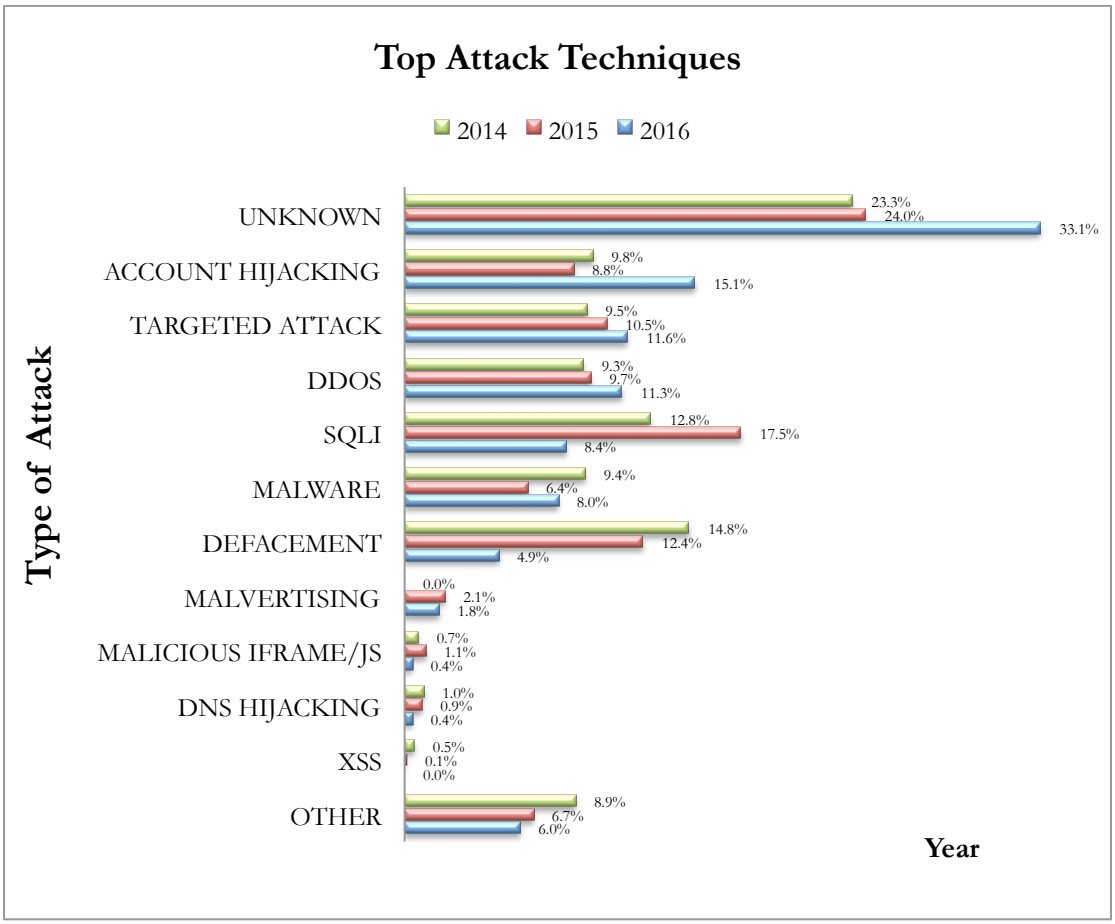


Figure 1.2: Top Attack Techniques in Year 2014, 2015 and 2016

Figure 1.2 shows the top attack techniques represented in total percentage for each year from 2014 until 2016. It demonstrates that "unknown" attacks had recorded an increase of almost 10% in 2016 compared to 2014 (Passeri, 2017). The highest proportion of "unknown" attacks each year has emphasised the serious need for defence mechanisms. The unknown attack is generally known as a zero-day attack in the network security field (Levy, 2004). The persistent growth of vulnerability and threats has also emphasised the serious need for defence mechanisms. The main technology of network security focuses on access control, firewall and information encryption. However, it is also important to acknowledge the common issues related to bugs and deficiencies. For instance, a firewall alone is unable to detect intrusions occurring from within the network (Wankhade and Chatur, 2014). This further explains why the intrusion detection system (IDS) has become a popular option. In addition, IDS is recognised as one of the components in the security arsenal as "defense in depth" (Northcutt *et al.,*

2008), that acts as a complement to the existing security appliances. Although the IDS does not guarantee the security aspect, it will be greatly enhanced if integrated with other security measures, such as vulnerability assessments, data encryption, user authentication, access control, and firewalls.

1.3 Problem Statement

1. The rapid growth of Internet communication has led to the creation of myriad data. Hence, longer processing time is needed due to the high dimensionality of data communication, which is believed to significantly affect the system performance by reducing the attack detection speed (Davis and Clark, 2011). The main factor that contributes to performance reduction is due to the system having to process redundant and irrelevant features; thus it is crucial to develop a method that could efficiently remove them.

2. Despite the fact that numerous statistical detections have been developed and studies made in the past, achieving exceptionally low false detection with high attack recognition capabilities still remains a major challenge (Acarali *et al.,* 2016). Most of the previous statistical detection managed to achieve an unsatisfactory attack detection rate (referred to as true positive). This action has led to the generation of a high false detection rate due to the traffic being more likely to be classified as anomalous.

3. In classifying data type, the chosen algorithm plays an important role because it is highly associated with the derived features. For example, the classification process is inefficient if the derived features are not able to contribute enough to help the algorithm in the decision-making process. This will lead the algorithm to misclassify normal data as attack data (false positive) and attack data as normal data (false negative). The inaccurate result has compromised the reliability of the system to flag a true attack.

4. In a conventional detection approach, researchers often disregard the time need for the re-execution process for every piece of traffic, which further delays the determination of the traffic pattern. In some cases, the re-initiation of the whole procedure can be a mess when processing huge amounts of traffics. As such, the detection system should be fast enough to alert the administrator to prevent information loss. In addition, due to the number of incidents rising significantly, the incidents should be sorted according to their severity level. For instance, a critical incident would requires a quick response from a security analyst compared to less critical incidents (Anuar *et al.,* 2013). Furthermore, such an approach could provide better insight for security analyst regarding incident responses.

1.4 Research Question

In this thesis, this research attempts to address the overarching question: *"How could system detection performance be improved in order to identify known and unknown web attacks?* Four sub-questions then follow hereafter:

1. *What approach can be used to select prominent features within the dataset?*

2. *How can the false detection rate produced using conventional statistical techniques be reduced?*

3. *What is the suitable combination of classifiers in boosting algorithms that could improve the attack detection performance?*

4. *How can the detection ability be improved in order to identify similar attacks in the future?*

1.5 Thesis Outline

This section presents the structure of the thesis as follows:

Chapter 2 contains related studies on the subject matter including intrusion detection system (IDSs), feature selection, statistical, and data mining-based anomaly detection.

Chapter 3 describes the research methodology adopted in this study. The chapter gives an overview on how the proposed scheme is designed and implemented. It also covers the experimental setup. The selection of the required methods for the research and the criteria used to evaluate the performance are also highlighted.

Chapter 4 describes the proposed scheme that is based on three major phases: pre-processing, anomaly detection and post-processing. The overview of steps taken in pre-processing, anomaly detection as well as post-processing is also discussed in this chapter.

Chapter 5 presents the implementation of three major phases in the proposed detection scheme using WEKA data mining tools, SQL script and MySQL database. The procedure for each step of implementation is explained in this chapter.

Chapter 6 discusses the performance evaluation and analysis of the proposed scheme. The effectiveness of Pre-Processing, Anomaly Detection and Post-Processing is assessed using four widely used IDS datasets with various performance metrics.

Chapter 7 summarises the entire thesis. The limitations and the proposals for the possible future enhancement to this research are briefly mentioned.

Chapter 2

Literature Review

2.1 Introduction

This chapter aims to review most of the published works that are relevant to the field of IDS as well as its implication. Section 2.2 begins with a discussion of attacks on web traffic. Section 2.3 highlights an overview of IDS. Section 2.4 presents the pre-processing phase, particularly using a feature selection procedure. Section 2.5 discusses the anomaly detection employed as statistical and data mining (DM) approaches in classifying attack traffic. Section 2.6 emphasises the post-processing phase and the existing techniques. Finally, Section 2.7 summaries the overall findings.

2.2 Web Attacks

Recent development of online web systems has attracted numerous users to further utilise the infrastructure. The web traffic is analysed through the responses of requests that come from the clients and servers via the HTTP and HTTPS protocol. The security aspect of the web application is crucial because it stores countless sensitive user information and provides the means to access beneficial venture assets such as e-banking, online purchase, online stock trading and services with the financial organisation. For instance, the service will be considered unreliable if it fails to protect the sensitive credit card information of the users. In those cases, the service provider will incur heavy losses concerning data, money, and business opportunity or availability. For example, the recent massive attack was targeted at the online banking system. Almost 40,000 Tesco bank customer accounts were undergoing suspicious transactions, with 9,000 of the customers losing money (Tesco Banks, 2016). This incident is the largest breach to have hit a UK bank. Pending investigation, banks have taken

immediate action by freezing all online transactions in order to prevent more customers from being the victim of further unlawful activity. It was reported that the said attack had cost Tesco a hefty £25 million. The attack also resulted in the biggest drop in the value of Tesco stock on record. Hence, it becomes imperative to secure the related information assets with robust and reliable security systems.

SQL-injection is a commonly used attack employed to exploit a vulnerability known as command injection. Typically, SQL-injection is a technique that is employed to inject fictitious data into an SQL statement or query for malicious purposes through web page input. The SQL-injection attack makes it possible to alter, delete, or insert information from the affected web service and compromise its security system (Jang and Choi, 2014). Critical information such as full name, date of birth, previous or current home address can be collected from the server as part of identity fraud (Abdallah *et al.,* 2016). Another popular target for hackers is the banking sector. It is understood that the credit card information stored in the banking server's database is the most valuable information. Hence, the breach of credit card information will definitely cause major disruption, and most importantly, the compensation for the loss resulting from this action is unimaginable. There are known solutions to alleviate the impact of these attacks, which include encrypting the database system, limiting user privileges, and implementing data validation.

Another type of web attack is known as cross-site scripting (XSS) which comes under cross-domain security issues. The improper system coding writing style provides a good opportunity for the attacker to exploit the known vulnerability. The attack does not target the victim directly, but exploits the vulnerability of the victim who visits the website. In other words, the vulnerable website acts as a medium for the attacker to deliver malicious code to the victim's web browser (Razak *et al.,* 2016). First, the attacker must find a way to inject the malicious codes onto the targeted vulnerable website to launch the XSS attack. This vulnerable website usually allows user input on its pages, which then enables the attacker to insert the malicious code that subsequently is executed by the victims. The executed malicious input allows the attacker to access the victim's cookies, webcam, geolocation, microphone, or at worst the specific user's file system that can grant access to control of the victim's computer.

The exposure of the vulnerable web applications and sensitive information on the Internet has further emphasised the need to investigate the network security element. This has resulted in an increasing number of aggressive attacks incidents that caused serious damage to the targeted web-based information system. According to the top 10 applications of security risk by Open Web Application Security Project OWASP (2017), the injection flaws attacks became the most critical security risk to web applications, which subsequently led to more studies being conducted in this field. The research by Laranjeiro *et al.* (2010) proposed a learning-based approach to secure web servers through the detection of SQL and Xpath injection attack. The detection is based on input query where the attacker usually adds extra conditions to the original SQL commands. The proposed methods required examination of the structure as well as the type of inputs and outputs of the operation that exist in the XSD file. After the necessary information is collected, the workload generator is used to inspect the set of data that access the SQL/Xpath presented in the source code. In detection mode, the SQL query is compared with the normal SQL query that does not store any attacks in the lookup map. The execution will stop processing the query to avoid probable hazardous requests if the SQL query is not found. This approach is capable of alerting developers and service administrators to stop the XPath/SQL injection before the system is harmed.

Another way of identifying attack is by analysing a user's access request. The research by Threepak and Watcharapupong (2014) assumed the attack patterns to be more complex than the common access request. This complexity is used as a benchmark in detecting attacks. The recorded request log is inspected using Shannon entropy analysis and utilised to calculate the complexity level. In defining the entropy level, a normal log requests in training set is used as the benchmark of the legitimate profile. Meanwhile, the boundaries (threshold) in detection are measured using the average and standard deviation of the period for each entropy. Log requests that surpass the predefined complexity threshold are flagged as a potential intrusion. However, the false detection rate needs further improvement, although the proposed attack detection approach is able to detect attacks at a satisfactory rate.

The research conducted by Zolotukhin *et al.* (2014) was focused on payload, where HTTP log requests were extracted. The normal HTTP log requests have been used as a training set that describes the model of normal users' behaviour. This approach is similar to work performed by Kim and Lee (2014) in which they made use of the query to detect SQL attacks. Initially, the query trees are converted to dimensional vectors for feature extraction and feature transformation. The work has been carried out using a DM technique by utilising the SVM algorithm for classification purposes. The result has demonstrated conspicuous performance improvement concerning computational time reduction and attacks detection accuracy rate.

Traditionally, IDS works with the principle of "deep packet inspection" whereby the packet contents are inspected in order to detect any malicious activities. The increased usage of network communication has led to more demands for secure communication using a cryptography approach. In an encrypted traffic environment, SSL, VPN or IPsec protocols are utilised to offer better privacy and confidentiality. Most established works in detecting web attacks are mainly focused on investigating the log/payload content. As the traffic is encrypted, the payload (log) is unavailable as the content is indecipherable.

2.3 Intrusion Detection System

An IDS is an application system or device that functions to identify either hostile activities or policy violation activities within a network. Anderson (1980) in a technical report described how audit trails containing patterns of legitimate information (user behaviour) could be utilised to distinguish and identify abnormal behaviour. The main interest in securing a network infrastructure is to design a network that is able to protect the confidentiality and integrity of data information while also ensuring the resource's availability. According to Thai and De Oliveira (2013), a defective network design with limited misconfiguration of the software can lead to a more serious vulnerability issue, which makes it easier for an attacker to attack an organisation.

2.3.1 Types of IDS

With regard to IDS deployment, the scope can be classified as host-based IDS (HIDS) and network-based IDS (NIDS) (Muda *et al.,* 2011a). The scope is based on the location of IDS which is deployed to inspect suspicious traffic. More specifically, NIDS captures the whole network segment and analyses it to detect for signs of hostile traffic. Meanwhile, HIDS focuses on a specific host and analyses information such as system calls, logs, and packets. In that manner, HIDS is regarded to be the better option in helping to identify internal attacks compared to NIDS (Iii, 2007).

2.3.2 IDS Detection Methods

The IDS detection methods are divided into two types: misuse-based detection system (MBDS), referred to as signature-based detection and anomaly-based detection system (ABDS) known as behaviour-based detection (Chen *et al.*, 2010).

2.3.2.1 Misuse Based Detection System (MBDS)

MBDS adopts predefined signatures that were previously stored in the library to detect known attacks. MBDS is more similar to a virus scanner in term of using signature as the detection approach. The example of tools that use MBDS are given in Snort (2002). The signatures are in the form an understandable and straightforward structure that helps to identify attack activities (Fugate, 2012). The signature is used to identify specific known threats considering that each signature represents a unique threat. This approach is believed to significantly reduce the false negative because it initially contains collections of attack signatures. The implementation requires the signature to be compact and straightforward to minimise the signature size to allow it to work under heavy networks. Besides, MBDS approaches can also be employed to examine incoming traffic (packets) to find relevant attributes such as IP addresses, protocols, bytes length, and ports, which can mostly be obtained from packet headers information and payloads (Fugate, 2012). However, MBDS has its limitations, considering that it depends solely on regular signature updates. Consequently, it is impossible to recognise unknown or new attacks that are passing through the system (Louvieris *et al.,* 2013). In view of the rapid growth of threats, it is difficult for misuse detection to keep up with

higher detection rates. Despite its limitation in maintenance and detecting an unseen attack, MBDS has been recognised as an efficient system for identifying the considerable number of attacks, and being easy to use and suitable for huge environmental diversity.

2.3.2.2 Anomaly Based Detection System (ABDS)

An anomaly is a state of action or behaviour that deviates from the legitimate state (Wang, 2004). The anomalies are initiated through the variation of unusual activities that are vital for data inspection, such as cyber-attacks, e-banking fraud, and social engineering. It is important to define the state of abnormality in determining anomaly. According to Ahmed *et al.* (2016), the category of anomaly can be classified into the following three types:

1) Collective anomaly: In this case, a collection of a similar event that is different from the entire dataset is classified as a collective anomaly. Sync flood is one of the Denial of Service (DoS) attack types that occur when there is a superfluous request made in an attempt to flood the targeted server. Hence, it would not be considered as an anomalous event if a single request were detected to be unresponsive. Meanwhile, an abnormal phenomenon is bound to occur when the collective of the unresponsive pattern is received from clients, which will cause the targeted server to be unresponsive to any other requests.

2) Contextual anomaly: Conditional or contextual anomaly is recognised when the data are performed abnormally in a specific context. For example, port scanning is usually performed when a security expert conducts penetration testing in order to discover the system's vulnerability to prevent it from being exploited by hackers. These scheduled activities normally generate more traffic flows inside the network during the testing. The traffic is considered to be an anomaly if the testing is permitted, despite the high network traffic caused by the scanning activities. On top of that, the increasing amount of traffic flow caused by the scanning activities during the non-scheduled program can be interpreted as a contextual anomaly.

3) Point anomaly: A point anomaly occurs when a particular single data instance from the dataset is different from the normal scheme. For instance, normal users will access their account daily on average, and there is the possibility that they might insert an incorrect password probably one or two times. However, point anomaly involves attempts to access the account, which may be more than average, considering that the attacker launches a brute force attack to gain unauthorised access to the account.

In ABDS, the point anomaly detection scheme has been widely applied. It is crucial to note that the statistical measurement (Chen *et al.,* 2010), distance-based measurement (Bayarjargal and Cho, 2014) and clustering (Louvieris *et al.,* 2013) are the main factors that help to identify or estimate the point anomaly (outliers). Chen *et al.* (2010), in their research constructed a normal profile by clustering the traffic attributes from the packet headers, in which any attribute's value that deviates from this profile is considered as an outlier. Concerning this, Bayarjargal and Cho (2014) employed Mahalanobis distance on selected packet attributes to compute the distance between normal and abnormal traffics. Meanwhile, Louvieris *et al.* (2013) performed outlier detection using the nearest neighbour algorithm by calculating the distance between connection vectors. Unlike MBDS, this detection-based method does not require specific knowledge about an attack to be known beforehand. According to Guo *et al.* (2016), anomaly-based detection usually elevates more false alarm, compared to the misuse-based detection. False alarm occurs when ABDS inaccurately defines the normal traffic as abnormal traffic. The main focus of ABDS is to design high detection and prediction with smaller amounts of false alarms to avoid IDS performance reduction.

Generally, anomalous behaviour is regularly defined as an intrusion in detecting anomaly traffic. Considering this assumption, the traffic behaviour can be divided into two classes: (1) intrusion caused by anomalous behaviour and (2) intrusion caused by non-anomalous behaviour. As stated by Ahmed *et al.* (2016) with regard to the three types of anomaly, it can be indicated that anomalous activities may not totally be demonstrated as true intrusion action. In many cases, the abnormal behaviour is flagged as intrusion; however, it is considered as legitimate activity after the investigation. Furthermore, it is difficult for anomaly detection to define abnormal behaviour when it is not really different from the normal pattern (Liao *et al.,* 2013). Hence, this will cause

the system to flag abnormal data instances as normal instances. Therefore, the aforesaid situation has generated another two types of classes known as: (1) legitimate action caused by normal behaviour, and (2) legitimate action caused by intrusion behaviour. The four major classes used in measuring the IDS performance and effectiveness are (1) True Positive, (2) False Positive, (3) True Negative, and (4) False Negative. In most established works, various techniques have been performed in anomaly detection such as statistical and DM approaches.

2.3.3 IDS Datasets

There are many synthetic datasets available in the field of IDS. This includes DARPA 1999 (Lippmann *et al.,* 2000), KDD 99 (1999), NSL KDD (Tavallaee *et al.,* 2009), CAIDA (2011), DEFCON (2000), The Internet Traffic Archive (2008), LBNL (2005), ISCX 2012 (Shiravi *et al.,* 2012) and UNSW-NB15 (Moustafa and Slay, 2016). Most of these datasets are unlabelled, and it requires comprehensive search activity to tag attack traffics.

The DARPA 1999 was constructed to simulate the traffic of a medium sized US Air Force Base. Despite heavy criticism by McHugh (2000) and Brown *et al.* (2009) on their limited ability and accuracy in demonstrating real-live traffic, the dataset remain to be extensively adopted in the field of IDS. A similar simulation environment was employed to generate a KDD 99 dataset where the difference is that the dataset was captured under the DARPA 1998 project. The NSL KDD dataset was recognised as an improved version of the KDD 99 dataset in the way that it has removed huge numbers of redundant and duplicated records in both training and testing sets.

In the CAIDA dataset, most of the attacks were generated according to very specific and particular events that make it suitable for researchers to develop a solution for detecting certain attacks. However, some of the backbone traces are anonymized by payload, with some other information such as protocol information and destination being completely unavailable.

The DEFCON dataset was known as a commonly employed dataset to evaluate IDS performance. It was generated during the competition named Capture The Flag (CTF)

and does not represent a real network environment. This is due to the dataset containing a substantial amount of attack traffic, compared to benign traffic, which has made this dataset suitable for alert correlation methods.

Other datasets such as The Internet Traffic Archive and LBNL, have suffered from heavy anonymization which resulted in a lack of packet information, such as individual IP address. On top of that, the datasets were developed in the '90s, which create further questions regarding their relevancy to represent the modern traffic environment.

The recently published datasets ISCX 2012 and UNSW-NB 15 were claimed to be more realistic due to their containing of recent sophisticated attacks. It is appropriate to adopt these datasets for a better projection of presenting the modern network traffic environment.

2.4 Pre-processing Phase

In IDS, the pre-processing phase is required with aim of easing data analysis and improving the processing time. Feature selection is widely employed to reduce high dimensionality data while removing insignificant information. This procedure is vital in order to improve the detection speed when processing the huge amount of traffic (Ji *et al.,* 2016).

2.4.1 Feature Selection

Feature selection is a foundation of machine learning that has been studied for many years (Liu and Motoda, 1998). It is commonly used as a pre-processing phase in IDS to discover the most prominent features of learning algorithms. In this case, the most useful data will be utilised to obtain better future projection. Hence, the redundant or irrelevant features will be removed to prevent a biased classifier. The algorithm that is adopted in selecting the best feature is one of the principal elements in determining IDS effectiveness. It is important to minimise the error of selecting feature that can reduce the detection of abnormal behaviour. This is because the effectiveness of the selected algorithm is highly dependent on the features selection. There are a few advantages of feature selection that particularly reduce the data dimension, enhance the projection

accuracy, and improve the processing time to be significantly faster and efficient.

The two general methods for feature selection are the filters and wrappers approaches (El-Khatib, 2010). The filters approach is divided into two categories: filter-based feature ranking (FBFR) and filter-based subset evaluation (FBSE). FBFR ranks the relevant features by assigning weights to features individually. The assigning is based on the score of every single feature to the target classes without paying attention to the interaction between features. Feature ranking is faster than FBSE that processes 2^n (n=number of features) because it only computes the features once. However, it is impossible to get rid of redundant features due to the fact that filter ranking processes each feature independently (Khammassi and Krichen, 2017). Hence, FBSE was introduced to overcome these problems (redundant feature). It examines the whole subset (not just selecting the relevant features) and explores the degree of relationship between features. Overall, FBSE is more desirable in selecting feature in IDS compared to FBFR (Nguyen *et al.,* 2010).

2.4.1.1 Filter and Wrapper Methods

FBSE is a heuristic-based method that employs probabilities and statistical measures to search and evaluate the usefulness of all the identified features. Alternatively, wrapper-based subset evaluation (WBSE) utilises a classifier to estimate the merit of each feature subset. Commonly, WBSE is known to have a better predictive accuracy than FBSE considering that the selection approach is optimised when evaluating each feature subset with a particular classification algorithm. Conversely, it is very expensive to be executed considering it evaluating each set of features (Bolón-Canedo *et al.,* 2015). Moreover, the wrapper can become uncontrollable when dealing with a large database that consists of many features (Hall, 1999). Wrappers are also associated with the classifier's algorithm, which makes it more difficult to shift from one classifier to another considering that the selection process needs complete re-initiation. Unlike wrappers, the selection criteria of filters use distance measures and correlation functions because they do not require re-execution for different learning classifiers (Cleetus, 2014). It has been observed that the execution is much faster than that of the wrapper approach. Filters are best suited to large database environments that contain many

features. On top of that, researchers have often used the filter as an alternative to the wrapper, since the latter is more expensive and time-consuming.

The feature selection process has attracted the interest of many researchers due to its potential in reducing high dimensional data. According to Mukkamala *et al.* (2004), the feature ranking algorithm was introduced merely to select the top six features based on rank. They adopt three ranking algorithms, namely support vector machines (SVM), multivariate adaptive regression splines (MARS), and linear genetic programming (LGP). The algorithm will select the best feature and then compare the performance of each algorithm. The detection will be programmed to detect Probe and DoS attacks. The LGP can achieve a higher accuracy rate in detecting both types of attacks compared to other algorithms. However, this approach is only effective to specific types of attacks. The speedy computation ability of the filter ranking has made it suitable to be applied for huge datasets. For instance, Wald *et al.* (2013) used filter ranking to reduce 480 features to a total of 40. They compared three different approaches of feature selection which are filter-rank, FBSE, and WBSE in order to find the best method to select the relevant features. Three different feature selections with six different classifiers, five-nearest neighbours (5-NN), logistic regression (LR), multi-layer perceptron (MLP), Naive Bayes (NB), RF with 100 trees (RF100), and SVM were utilised to achieve the best results. As recommended by these authors, the filter ranking process executed in high dimension data tends to perform better with SVM classifier techniques. Although, they claimed that the filter ranking method is more competent compared to the FBSE and the WBSE. However, there was no explanation of the methods implemented in choosing the top 40 features from the ranking table, which might affect the final optimal set of features, considering the irrelevant features.

Another filter ranking was implemented by Ambusaidi *et al.* (2014) in which a hybrid selection was performed by combining both mutual information (filter ranking) and wrapper. The wrapper approach utilised least square-SVM (LS-SVM) for features optimisation. Mutual information provides a good measurement to find the relevant features by quantifying the amount of information to the output class. However, the false positive and the detection rate can still be improved, despite the significantly reduced number of features. In relation to the calculation of features selection relevancy,

El-Khatib (2010) proposed the information gain ratio (IGR) to replace the traditional information gain (IG) calculation method, considering that IG is normally biased towards features that contain high distinct value. The selected features are ranked based on score derived from the IGR calculation. The K-means classifier is then used to determine the best-fit feature-set based on the performance results accuracy. The selection process will end when the current subset performance drops below the previous subset accuracy. Next, the selected features are tested with three types of artificial neural network (ANN) architecture listed as follows: (1) perceptron, (2) multilayer backpropagation perceptron (MBP), and (3) hybrid multilayer perceptron (HMP). HMP was found to have a lower false positive rate and required the longest time in the learning model; however, its classifier has outperformed both the perceptron and the MBP. Nevertheless, there were no significant detection rate differences between the proposed HMP and MBP.

The classifier was shown to be less accurate when used solely to evaluate performance accuracy compared to the ensemble technique (combining more than one classifier) (Mukkamala *et al.,* 2005). Another research by Zainal *et al.* (2008), proposed adaptive neural fuzzy inference system (ANFIS) and LGP algorithms in detecting four main types of attacks, namely Probe, DoS, R2L, and U2R. This ensemble technique was implemented with a reduced set of features (between six to eight) for each type of attack. This technique has managed to achieve more than 99% detection rate for R2L and U2R attack types, including an average of 99.15% accuracy for all attack types.

In selecting the best-fit classifier, Zaman and Karray (2009) have compared their approach with two different classifiers which are neural network (NN) and SVM. They proposed a novel method called the enhanced support vector decision function (ESVDF) to select features based on rank, while backward elimination ranking (BER) and forward selection ranking (FSR) are used to calculate the correlation between features. The comparison of both algorithms revealed that the NN performed better than the SVM. The proposed enhanced method further reduced the number of features and the time taken to build a model, by a negligible margin of 0.08% and 0.11% for NN and SVM, respectively. Nevertheless, the accuracy rate is still lower than the full features. The need to achieve low false detection and high attack recognition

capabilities is still a major challenge despite the introduction of numerous feature selection approaches.

Table 2.1: Comparison of Previous Work (Feature Selection)

Authors	**Features**	**Techniques**	**Pros**	**Cons**	**Dataset**
Mukkamala *et al.* (2004)	41	Ranking using SVM, MARS and LGP	Useful in detecting Probe and DoS attack types	Not suitable for detecting R2L and U2R attack types	KDD 99
Zainal *et al.* (2008)	41	Linear Genetic Programming ensemble with Adaptive Neural Fuzzy Inference System	Effective in detecting DoS and R2L using ensemble approaches	Not effective for detecting Probe and U2R attack types	KDD 99
Zaman and Karray, (2009)	41	Enhanced Support Vector Machines	Time effective in building model	Reduced features output was lower when compared with using full features	KDD 99
Ravale *et al.* (2015*)*	41	Hybrid Selection using K Means & RBF Support Vector Machines	Reduced features achieved better detection rate compared with using all features, with improvement around 9%	Single classifier using SVM has outperformed the proposed hybrid approaches	KDD 99
Kakavand *et al.* (2016)	256	Dimensionality Reduction using Text Mining Model	The features had reduced from 256 to 25 features. The model had achieved 97% detection rate with 1.2% false alarm rate	The experiment was conducted on payload traffic which required additional computational effort. In addition, only 25% of testing data used when compare to training data	ISCX 2012
Aljawarneh *et al.* (2017)	41	Hybrid selection using Information Gain	Reduced features from 41 to 8 features with 99.81% accuracy rate	The classification model is built by combining seven classifiers which are expensive to execute	NSL KDD

2.4.2 Summary

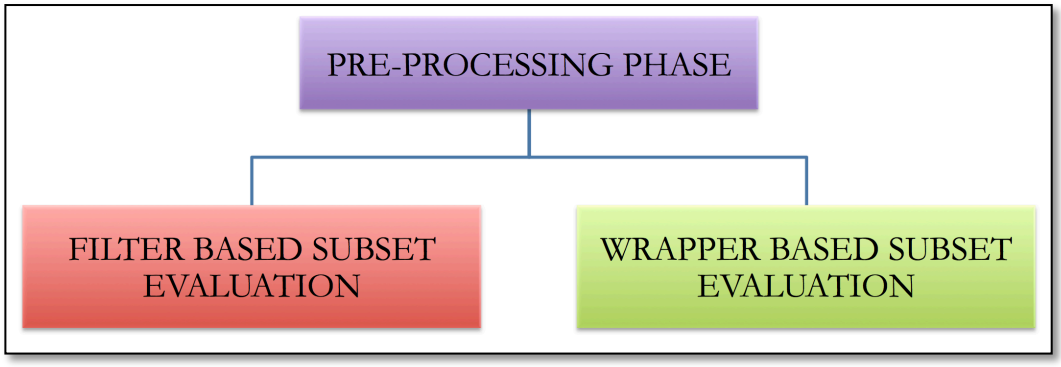


Figure 2.1: The components of the pre-processing phase

The previous work had highlighted the function of selecting prominent features is to minimise the data dimensionality in order to achieve a better discrimination boundary in classifying attack behaviour.

Figure 2.1 shows the two components proposed in the pre-processing phase. In line with this, two selection methods of FBSE and WBSE are further discussed in this study. According to Nguyen *et al.* (2010), FBSE can be easily deployed and is capable of removing redundant features effectively. Meanwhile, the better feature optimisation can be achieved using WBSE where the merit of the feature set is measured by the classification algorithm (Cleetus, 2014). However, WBSE is expensive to be executed compared to FBSE (Bolón-Canedo *et al.,* 2015). Hence, the combination of both FBSE and WBSE methods are further investigated with the aim of assist the selection of prominent features.

2.5 Anomaly Detection Approaches

There are two detection methods that are regularly employed in the anomaly detection field: statistical and DM approaches. The advantages and limitations of both approaches are further discussed in the next sub-sections.

2.5.1 Statistics based Anomaly Detection (SBAD)

The statistical method in IDS was first introduced by Denning (1987). The detection approach primarily relies on a collection of data history to create normal profile behaviour. In this approach, only benign traffic data collected over a period of time is utilised to detect intrusion (Denning, 1987). The collected benign traffic is utilised to generate a profile behaviour, in which any incoming traffic that deviates from the profile will be recognised as suspicious traffic. As a result, the intruder behaviour form can be detected as a possible attacker through this detection approach (Patcha and Park, 2007). The main advantage of this approach is its ability to employ statistical procedures that could extract the traffic features in representing the behaviour pattern of the data, which can either be normal or attack traffics. A considerable amount of established works have employed statistical measures, e.g. Mahoney and Chan (2001), Shamsuddin and Woodward (2007), Chen *et al.* (2010), and Xiong *et al.* (2013).

On top of that, some of the published works proposed a statistical model in more specific areas such as Packet Header Anomaly Detection (PHAD). In PHAD, packet characteristics and behaviours are used to recognise abnormal patterns. According to Mahoney and Chan (2001), PHAD uses statistical measurements from activities history to construct a normal profile. The traffic that deviates from the normal profile and behaves abnormally will be identified as an intruder through this detection method (Patcha and Park, 2007). PHAD uses all information inside the packet header instead of just IP addresses and port numbers (Mahoney and Chan, 2001). The 33 attributes in the packet header represent the information of 3 layers in the OSI 7 layers model, i.e. data link, network, and transport layers.

The information in the attributes was used to measure the probability of each packet, either towards benign or abnormal behaviour. An anomaly score will be given when any dissimilarity is detected from the match against normal data. Finally, the sum anomaly score of each packet is summed up and flagged as anomalous if the score surpasses the pre-set threshold.

In contrast to conventional PHAD systems, Shamsuddin and Woodward (2007), proposed Protocol-based Packet Header Anomaly Detection (PbPHAD) in two different environments, namely network-based and host-based. The proposed method adopts three main protocols which are TCP, UDP, and ICMP to construct a normal profile that contains normal behaviour. Similarly to the traditional PHAD system, this approach uses all 33 packet header attributes to produce an anomaly score. The score will individually rate the degree of incoming traffic. There is still room for further improvement, despite the fact that PbPHAD exceeds the results from the PHAD and DARPA best System (Lippmann *et al.,* 2000), with 57.83% detection rate.

To identify malicious packets present in within TELNET traffic, Chen *et al.* (2010) proposed the Lightweight Network Intrusion Detection (LNID) System. In LNID, benign behaviour extracted from training data is used to construct a normal profile. Additionally, the normal profile is used as the indicator to compute the anomaly score that was given during the matching process between testing and training data. The packets are flagged as malicious when the score surpasses the pre-set threshold. Insignificant features from training data are removed during the pre-processing phase to reduce the computational cost.

Although the scoring approach in LNID managed to achieve higher detection rate for U2R and R2L up to 86.4%, it still has the opportunity for further improvement. For instance, the traffic processed by LNID uses whole features derived from the packet header and payload data which may contain some redundant and irrelevant attributes that will increase the computation effort. Another area of concern is the need for re-execution for each set of traffic, regardless of whether the traffic has been determined earlier.

In addition, it is difficult for the detection approach to identify an attack that possesses similar behaviour to normal traffic. This is due to the anomaly approach being solely dependent on a normal profile as a baseline in determining the traffic status (normal or attack). It is important to note that the traffic could be anomalous if several outliers are present in the traffic, particularly taking into account that the predefined threshold is determined without performing further analysis. Besides, the detection methods only emphasise R2L and U2R without giving any concern to other risky attack types such as Probe and DoS. On top of that, the approach requires payload extraction that is limited to unencrypted traffic environment.

Profile generation has led Xiong *et al.* (2013) to propose catastrophe and equilibrium surface theory to extract common behaviour present within the network. The standard equilibrium surface is used to indicate the change of packet behaviour, which makes it suitable for inspecting incoming traffics. The real challenge is to obtain the best detection rate together with the lowest false alarm rate, despite the fact that the evaluation of true positive slightly increased over 86% for TELNET traffic.

Table 2.2 presents the established works that employed the statistical approach based on the DARPA 1999 dataset. The methods presented in the table solely depend on anomaly scores to differentiate between actual attack and normal data. Therefore, the results achieved are not particularly encouraging and still have room for further improvement. Alternatively, DM approaches are extensively employed and can be continuously improved for better detection capabilities.

Table 2.2: Comparison of Previous Work (Statistical Approaches)

Authors	**Techniques**	**Pros**	**Cons**	**Dataset**
Mahoney and Chan (2001)	Statistic based, Anomaly score, predefine threshold	Recognise 70 attacks out of 180 with 39% detection rate and 10% false alarm rate	The proposed method could not achieve higher detection rate	DARPA 1999
Shamsuddin and Woodward (2007)	Statistic based, stationary model, Anomaly score method, predefine threshold	Detection Rate 57.80% (Identified 48 out of 83 attack instances)	The detection rate recorded by proposed method has room for further improvement	DARPA 1999
Chen *et al.* (2010)	Statistic based, stationary model, Feature Extraction, Anomaly score method, predefine threshold	Achieved 72.70% of detection rate with 1.36% false alarm rate	Only focusing on detecting R2L and U2R attacks inside Telnet traffic	DARPA 1999
Xiong *et al.* (2013)	Statistic based, Catastrophe Theory, distance based, predefined threshold	Detection rate average at 86.3% with 3.2% false alarm rate	The predefined threshold has generated high false alarm rate	DARPA 1999

2.5.2 Data Mining Based Anomaly Detection (DMBAD)

DM is a technique for discovering a systematic data relationship as well as determining the fundamental data information (Louvieris *et al.,* 2013). DM can be divided into two broad categories, which are unsupervised and supervised approaches. Furthermore, clustering and classification are the respective examples of unsupervised and supervised algorithms. In clustering, the group of objects is categorised based on the characteristic data points. In this case, every single data point in a cluster is similar to those within its cluster but different from those in different clusters (Hair *et al.,* 2009). The purpose of grouping similar data into one or more clusters is simply to ease the abnormality identification. However, this approach will potentially increase the false alarm rate (Hubballi *et al.,* 2013). In view of the fact that the IDS performance is highly dependent on low false alarms, its capabilities can be downgraded if high false alarms continuously occur. Hence, classification is considered the better approach for classifying data (e.g. benign or anomalous), especially in reducing the false alarm rate.

Meanwhile, in supervised learning, the knowledge structure is created to recognise and classify newly found instances into predetermined classes. The collections of samples provided are inserted into machine learning to further classify them into classes. On another note, a classification model is developed as an output of the learning process based on the instances information provided in the learning stages. In short, the focus of supervised learning is to model the input/output relationships with the objective of recognising a mapping from input attributes to an output class. The output, such as the regularities among attributes of the same class or differences between them, can be demonstrated as a decision tree, a flowchart, and classification rules, which are to classify a novel unseen instance.

There are two stages required in classification: training and testing. In training, the labelled data provided (training dataset) are examined using a classification algorithm to construct a classification model. The model that is created during training is able to recognise instances. The model is then used to classify the testing dataset (unlabelled dataset) which contains the unseen instances. The outputs are calculated based on the classifier used in constructing the model. Overall, four classification metrics managed to

be produced: True Positive, True Negative, False Positive, and False Negative.

2.5.2.1 Classification Techniques

Classification is a supervised approach that is able to differentiate unusual data patterns, thus making it the most suitable option to identify unseen attack patterns (Farid *et al.,* 2014). A classifier will gather the knowledge by training the pre-classified sample representing the classes. Furthermore, it can act as a predictor for some unknown samples, or a descriptor for classified samples. On top of that, classification has been widely used considering its strong ability to identify attack and normal structure accurately, which helps to reduce false detection (Muda *et al.,* 2011a). Most of the established works employed the following as a single classifier in the field of intrusion detection: artificial immune systems (AIS), fuzzy logic (FL), one rule (OneR), hidden markov model (HMM), genetic algorithm (GA), neural network (NN), naïve bayes (NB), decision table (DT), decision tree (J48), random forest (RF), support vector machine (SVM) and multilayer perceptron (MLP).

AIS was inspired by natural immunity models. The implementation algorithm that was inspired by the immune system has been widely applied to various real world applications. In view of ABDS, the AIS performs the detection by generating the pattern of abnormality given from a given set of normal data (Wu and Banzhaf, 2010). The abnormal pattern is used as a benchmark for detecting the anomaly data. In the learning process, the efficiency of the algorithm is highly dependent on the traffic contents (Hosseinpour *et al.,* 2014). Hence, the dataset that contains continuous attribute values will ease the process of generating the abnormal pattern.

Apart from AIS, another artificial intelligence classification technique is FL. FL is able to differentiate an object that belongs to a different class at the same time. In the intrusion detection field, a model that was built from numeric data is bound to produce errors due to the behaviour that slightly deviates it from the model, thus causing false detection. The FL technique is believed to achieve low false detection rates. However, the major drawback of this classifier because it requires IF condition- THEN conclusion rules to encounter the problem as well as attention on fine tuning to success.

OneR classifier is known as a rule-based algorithm that generates rules with the basic purpose of selecting the features and appropriately ranking them. The algorithm establishes rules for every value in the feature by testing a single feature at a time. In this algorithm, a set of classification rules for the particular tested features is generated based on the value of a single feature, whereby the feature with the lowest error rate is chosen as "one rule". The error rate is generated by the proportion of instances that do not belong to the majority class of the corresponding feature. Overall, it is very helpful in selecting features but not for classifying data that contains many features because OneR only considers one feature (Muda *et al.,* 2011b).

The Hidden Markov Model (HMM) approach has been widely used in various fields of application such as speech synthesis, crypto analysis, speech recognition, and classification problem-solving. In the ABDS classification model, HMM has the capability to distinguish between normal and abnormal behaviour. Unfortunately, the model is more suitable for one-dimension sequence classification, such as wave spectrum or voice (Choraś, 2015). As in ABDS, the data are in multi-dimensional sequences (continuous and discrete) mixed together, which makes them unsuitable for detecting anomalous traffics.

Apart from HMM, GA has been widely applied as a selection feature in the field of intrusion detection (Kamarudin *et al.,* 2016). In the view of GA as a classification approach to achieve high accuracy, unfortunately, the drawbacks have outweighed the advantages of this algorithm. The main downside refers to the computational effort in processing the crossover, mutation, selection, iteration, and combination stages (Tsai *et al.,* 2009). The methods seem to be unsuitable to be used as a detection approach in dealing with high volumes of traffic.

After reviewing and considering the deficiency of AIS, FL, OneR, HMM, and GA (Raj Kumar and Selvakumar, 2011; Farid *et al.,* 2014; Kosamkar and Chaudhari, 2014; Aditi and Hitesh, 2013; Farnaaz and Jabbar, 2016) the research continues to investigate several other classification algorithms namely MLP, NB, J48, SVM, DT, and RF which are the notable common methods used in the field of intrusion detection that are capable of producing easily understandable and realistic detection results.

1. **Multilayer Perceptron (MLP)**

   Mapping a set of input data into outputs can be performed using simple feed-forwards NN or MLP. In MLP, multiple layers of neurons are placed in layers that always flow towards the output layer. Single perceptron refers to only one layer, while multilayer perceptron has multiple layers. The class of multilayers usually operates in a feed-forward way. Each neuron placed in a layer is connected directly to another neuron on a subsequent layer. The algorithm applies a sigmoid function as an activation function. These classification algorithms have been popular to be applied in back propagation techniques in training the network. Using this technique, the predefined error-function value is computed by comparing the output values with the correct answer. The error generated is then fed back through the network. The information gathered from the previous stage is used to calibrate the weight of each connection with the aim of reducing the error function. The process is repeated for an adequate number of rounds until it reaches the state where the calculation of error is small. In this state, it shows that the network has acquired a certain level of function. Normally, there will be an issue for the algorithm in classifying instances that were not present in the training set. In some cases, this is vital, considering that the training sample is limited in availability. As a result, the algorithm tends to over-fit and is unable to capture the true statistical process in producing the data. In the area of neural networks, there is an early stopping criterion for simple heuristics in ensuring that the algorithm can be well-generalised to instance that does not exist in the training data.

2. **Naive Bayes (NB)**

   In classification, the NB classifier can be described as having a simple probability classifier which has a strong independence assumption among the attributes, depending on the class variable. It is typically used to generate conditional probability when analysing relationships between independent and dependent features. The NB theorem delivers a way to measure posterior probability P(H/X) based on the equation P(H/X)=[P(X/H).P(H)/P(X)], where P (H) represents the class prior to the probability and P(X) corresponds to the predictor prior probability (Muda and Yassin, 2011). NB classifier has

been implemented in various fields with the aim of solving classification down side, such as low detection accuracy and false detection rates. Moreover, it is also effective in handling continuous data and missing values alongside its simple and easy implementation (Farid *et al.,* 2014).

3. **Decision Tree (J48)**

The J48 classifier is one of the present notable methods used in DM techniques. The algorithm was first introduced by Quinlan (1986), and can be viewed as a tree from a set of attributes to a particular class. It consists of three indispensable sections: the root node that illustrates the stipulation on a data point, the branch that corresponds to the probability feature values, and a leaf node that is labelled with the decision value for a classification category in order to classify its entity. The process of constructing a decision tree is similar to the procedure of divide and conquer. Hence, the IG criterion is used to select attributes that can provide maximum information in forming a decision to achieve the finest feature splitting (AL-Nabi and Ahmed, 2013). It then seeks to calculate the information content by attempting to produce the answer in a series of bits. Therefore, a one-bit answer is encoded in the bit of yes or no. The conventional approach to constructing a J48 algorithm is called the classification and regression tree (CART). The J48 algorithm with discrete class labelled is known as a classification tree, while the regression tree can be described when J48 algorithm contains a scale of continuous values.

4. **Support Vector Machines (SVM)**

Most of the early works adopted SVM, which was first proposed by Cortes and Vapnik (1995). In the intrusion detection field, SVM is frequently used as a supervised learning to design binary classification. Specifically, it works by obtaining an optimal separating hyper-plane by first mapping the input vector into the high dimensional feature space. The set of training vectors are isolated into normal (class +1) and abnormal (class -1) data points. The aim of using SVM is to determine a linear optimised hyper-plane where the decision boundary among classes is maximised. Finally, the specified parameter is used as a penalty factor to allow the user to make a trade-off between the misclassified instances and the width of a decision boundary (Tsai *et al.,* 2009).

5. **Decision Table (DT)**

   Among other classifiers, DT is one of the most straightforward and easily understood classifiers. The classifier can be viewed as a programming tool that helps to represent discrete functions. Additionally, the matrix table is used to represent conditions and actions. In the matrix table, the upper row of the matrix is used to correspond to the sets of conditions, while the action that needs to be taken when a condition is satisfied will be placed in the lower rows. Hence, the column is called a rule that represents a set of procedures "if conditions, then actions" (Aditi and Hitesh, 2013). In processing unlabelled data, the table classifier first looks for correct matches in the decision table using features in the scheme. The majority class of all matching instances will be returned if the instances managed to be found or else the decision table itself will be returned. The induction algorithm needs to decide the appropriate features to be inserted in the scheme or to the body before building the decision table.

6. **Random Forest (RF)**

   The RF (Breiman, 2001) algorithm is classified as an ensemble CART which is widely used in DM techniques for prediction, pattern recognition and probability estimation (Zhang *et al.,* 2008; Attal *et al.,* 2015; Khoshgoftaar *et al.,* 2007). RF is a combination of many tree predictors where each tree is constructed by a different bootstrap sample from the original dataset. RF is an example of a bagging technique in an ensemble where more than one decision trees is used to build a classification model. In RF, the trees are composed independently with random samples. The outputs are chosen based on votes from each tree, which indicate the tree's decision on the class object. The most votes for the object are from the best individual trees. The forest chooses the class with the most votes for the object. The main advantages and strength of RF are its robustness in handling high dimensional data (Hastie *et al.,* 2009) while solving over-fitting issues. Additionally, this method is effective to use with a small number of available learning samples. This is because, during tree creation (Htun and Khaing, 2013), the pruning stage is discarded and only the small set of data is used to perform the searching procedure.

The above algorithms, which include MLP, NB, J48, SVM, DT, and RF, have been the focus of the research in the field of intrusion detection due to their effectiveness in processing and producing excellent detection outputs. The processing time for each classifier is different and highly dependent on the simplicity of the algorithm used in data processing. This will further lead to a longer processing time (a large number of instances) to be executed in building a model to detect attacks that may lead to higher misclassification rates (Panda *et al.,* 2012). Conversely, some of the classifiers took a longer processing time to build a detection model, but most of the time, they will achieve better detection results due to the complex procedure of deep analysis of data instances. For instance, MLP is able to achieve better detection accuracy compared to SVM and J48 (Raj Kumar and Selvakumar, 2011), but at the same time, the algorithm has consumed more time compared to DT, J48, and RF (Tribak *et al.,* 2012). In addition, RF is the ensemble approach that consists of many decision trees with an added advantage that allows it to process both numerical and categorical data, thus enabling a finer prediction output to be produced compared to J48 alone. Hence, the RF classifier has turned out to be more feasible than an individual J48.

The selection of an effective and rigorous method is crucial in the DM approach, especially in classifying the level of accuracy, detection, and misclassification rate that are highly dependent on the classifiers. According to Ben Amor *et al.* (2004), the comparative analysis between J48 and NB has concluded that the J48 managed to produce finer results in terms of detection accuracy considering that the decision node was made by the best features selected during the tree creation. Meanwhile, the algorithm in NB has made tight independence assumptions during the probation (observation) among attributes, which then leads to lower detection accuracy. However, NB is much faster than J48 in terms of execution procedure in developing the detection model (Ben Amor *et al.,* 2004). In detecting a known attack, (Panda and Patra, 2007) managed to discover that NB had outperformed J48 in detecting known attack behaviour, while J48 was found to be successful in recognising new attack behaviour.

As reported by Hasan *et al.* (2014) in their research, both RF and SVM are widely used in intrusion detection areas with the aim of solving issues caused by complex and dynamic datasets. Their experiment was carried out to investigate the performance of both algorithms in terms of detection accuracy and time taken to build the detection model. According to the results, SVM was slightly better with 1.5% in terms of detection capabilities over RF. However, RF was shown to be four times faster than SVM in the aspect of time taken to build the detection model. In addition, in terms of misclassification, the performances for both classifiers were inadequate with more than 30% instances being misclassified. Hence, according to their research, the performance outcome from both classifiers is an on going process and needs to be extensively explored to achieve better results. The performance of SVM over several classifiers such as NB and MLP has been evaluated by Nyakundi (2015) in his thesis. The proposed SVM managed to achieve the highest detection rate over MLP and NB, with NB showing the lowest detection rate.

In another study, Jain *et al.* (2016) compared several classification algorithms which include DT and RF. Their experiments were tested on multi-class attack types with the objective of finding the best algorithm for a specific class of attack type. For example, DT was found to be the fastest algorithm for detecting U2R attack based on the lowest time taken to build the detection model in comparison with RF that took eight times longer. RF achieved the highest detection accuracy in detecting U2R and DoS attack types, despite its longer processing time. In another work conducted by Jalil *et al.* (2010), the researchers compared multiple classifiers which include J48, SVM, and NN to detect network intrusion. The results of the experiments revealed that J48 managed to achieve better accuracy rates compared to SVM and NN.

The experiments conducted by Aziz *et al.* (2016) investigated the capabilities of single classifiers to detect an intrusion with a limited attack sample in the training data by comparing the NB, RF, J48, and MLP classification algorithms. The outcome revealed that NB excellently recognised attacks with a limited amount of samples in training such as U2R and R2L attacks. Other types of attack, such as DoS and Probe, were successfully dominated by RF, while J48 showed slightly lower results compared to RF. A study performed by Abhaya *et al.* (2014) reviewed the performance of different

classifiers such as SVM, NB, and J48. The focus of the performance matrix was to measure detection accuracy, false detection, and execution time in building a detection model. Most of the studies found that SVM was able to achieve higher detection accuracy, even with a small set of data, compared to J48, which is more suitable to be used in a large dataset. Regarding the time taken to build the detection model, NB was discovered to be a more suitable option because its implementation is fast and straightforward compared to J48 and SVM which are more time -consuming.

Research conducted by Farnaaz and Jabbar (2016) claimed that RF is better than J48 algorithms, considering the fact that RF is more robust in handling dimensional data (Hastie *et al.,* 2009), while J48 highly depends on a single tree for prediction analysis (Thaseen and Kumar, 2013). The advantage of combining more than one classifier (ensemble) to achieve a better accuracy rate while preserving the low false alarm rate, has attracted many researchers to explore the field intensively. This technique is further discussed in the next section.

2.5.2.2 Ensemble-based Classifiers

The ensemble technique in classification has attracted an increased number of researchers to perform a combination of several classifiers to obtain better predictions on accuracy performance (Chebrolu *et al.,* 2005; Folino *et al.,* 2010; Nguyen *et al.,* 2011) and. Previous research conducted by Dietterich (2000) and Mukkamala *et al.* (2005) had shown that the use of ensembles techniques is preferred, compared to the single use of a classifier in measuring the overall classification accuracy. The ensemble methods are divided into three main approaches: bagging, stack generalisation, and boosting.

1. **Bagging**

   Bagging was first introduced by Breiman (1996), and is one of the first ensemble-based techniques that utilises natural and simple ways to achieve high accuracy. Bagging is often known as bootstrap aggregating, which can improve detection accuracy by fusing the outputs of learned classifiers into a single prediction with the use of a majority vote. An example of an approach originating from bagging is RF. The RF is built from a number of decision trees

(Breiman, 2001). The algorithm achieved high classification accuracy by fusing random decision trees based on a bagging technique.

2. **Stack Generalisation**

   Stack generalisation, or stacking, basically involves the combination of predictions from several learning algorithms. In some cases, the instances are difficult to be classified because they are too close to the decision boundary which makes it easier to be misclassified by the classifier. In some of the cases that are very straightforward, the instances are placed far behind the decision boundary, which allows it to be well classified. This generates the question of whether the classifier will perform consistently correct outputs or vice versa. Hence, the idea behind stacking generalisation proposed by Wolpert (1992), describes how the prediction output from base-level classifiers serves as the input to another second level meta-classifier to achieve high generalisation accuracy.

3. **Boosting**

   Boosting is mainly used to boost a weak classifier or weak learner in order to achieve a higher accuracy classifier (Kamarudin *et al.,* 2017). In other words, boosting can be considered as a meta-learning algorithm. In relation to this, the incorrectly classified instances from the previous model are used to build an ensemble. The weak classifier such as a decision stump, which is based on a decision tree with a root node and two leaf nodes, is regularly used in boosting techniques (Fakhraei *et al.,* 2014). Adaptive boosting (AdaBoost) is the most popular boosting algorithm which was first introduced by Freund and Schapire (1995). Hence, the ability of this algorithm to produce good accuracy has attracted researchers, such as Hu *et al.* (2008), Panda and Patra (2009) and Li and Li (2010), to apply it in the IDS field.

The combination of multiple sources of information for analysis has improved the ability of IDS to identify threats (Folino *et al.,* 2010). According to the distributed IDS, data information from each IDS is combined to give insight of the overall network traffics for the purpose of enhancing its ability to detect malicious attack more accurately. One of the advantages is that when an IDS is temporarily unavailable; the

other available IDS could still provide information for detection analysis. Folino *et al.* (2010) proposed the use of a genetic programming (GP) classifier considering that it can be used in large dataset. The GP classifier is used to build a decision tree classifier. The generated classifier from each IDS is combined into an ensemble using the AdaBoost technique, particularly the AdaBoost.M2 algorithm. The evaluation utilised KDD 99 dataset as the data source. The experimental results showed that the proposed method is in the same class as the other two top winners in KDD Cup 1999. However, this approach is only limited to recognise the DoS type of attack despite the fact that it is suitable to be deployed in a distributed environment, in which the finding found the inability to detect R2L and U2R attack types with only 3.6% and 5.18% of detection accuracy, respectively.

Similarly to Folino *et al.* (2010), Nguyen *et al.* (2011) in their approach, build a single dataset collected from multiple sources, such as system log, system audit, and network traffic. The experiments were performed on the benchmark KDD_99 dataset. The dataset was labelled by domain expert knowledge with certain classification techniques to create the training set which will be split into proportions of 75% training and 25% validation. In the earlier stage, the data were first pre-processed with a K-means clustering algorithm to produce secondary features. Subsequently, the approach chose to employ the J48 algorithm to create an ensemble of different individual classifiers. The ensemble was measured using the weighted mean, which is dependent on the classification abilities of the individual classifier derived from the validation stage. The results reported that the proposed approaches managed to achieve the highest detection rate compared to other ensemble techniques such as boosting and bagging. The approach took 257 times longer than the boosting during the training phase although the approach managed to outperform other ensemble techniques. In some cases that combination of multiple sources of information could achieve better outcome, but the process can be infelicitous due to encrypted traffic (unreadable payload), privacy restriction (system audit), or failure of some IDS (limited information).

In another ensemble-based approach, Gaikwad and Thool (2015) conducted a bagging scheme and compared it with the other different classifiers, namely NB, bagged NB, C4.5, bagged C4.5, partial decision tree (PART), and bagged PART. A total 41 input features were reduced to 15 using GA during the pre-processing phase. The experiments were conducted using a well-known NSL KDD dataset. The results obtained from the experiment revealed that the proposed ensemble bagged PART is slightly better than C4.5 with 99.72% and 99.69% detection accuracy, respectively. The proposed approach took 7.6 times longer than C4.5 in building the training model despite performing slightly better in terms of detection accuracy. Hence, it can be concluded that the approach is not suitable for online detection. It is also important to note that the results are only based on detecting known attacks.

In another work, Syarif *et al.* (2012) investigated three ensemble techniques (bagging, boosting, and stacking) in detecting known and unknown network intrusion. The main objective of their research was to enhance detection accuracy and minimise the false detection rate in the NSL KDD dataset. For bagging and boosting, the method was performed along with four classification algorithms, namely J48, JRip, IBK, and NB. Meanwhile, the stacking approach employed the four algorithms that were previously mentioned as meta-level classification. The experiment reported that the approach managed to successfully achieve more than 99% detection accuracy in detecting known attacks. In contrast, the approach was found to be unsuitable to be used in detecting novel attack because it only managed to achieve 60% detection accuracy.

In a study conducted by Hu *et al.* (2008), AdaBoost with decision stump was proposed as a weak classifier. The noise and outliers existing in the dataset were initially removed by training the full data. The sample data that contained high weight are considered to be noise and outliers. The false alarm rate (FAR) was still at 8.9%, although the detection rate was almost 92%. Similarly, in Folino *et al.* (2016), Cellular Genetic Programming (CAGE) was proposed to evolve the combination function presented in ensemble approaches. The approach was tested on the ISCX 2012 dataset and managed to achieve a 91.37% attack detection rate. The recorded high false alarm rate constitutes a limit to the system's capability despite the high detection rate.

In choosing the right weak classifier for AdaBoost, Panda and Patra (2009) compared four classifiers, namely non-nested generalised exemplars (NNge), extended repeated incremental pruning (JRip), ripple-down rule (RIDOR), and DT as a base classifier for AdaBoost. The proposed AdaBoost with NNge managed to achieve the highest detection rate in detecting U2R and R2L types of attack. Meanwhile, the combination of AdaBoost with decision tables was found to be efficient in detecting DoS attacks. In a similar concept to Hu *et al.* (2008), Li and Li (2010) proposed the NB algorithm as a weak classifier. The overall performance (84% detection rate with 4.2% false alarm rate) is still much lower compared to Hu *et al.* (2008) although the proposed algorithm had recognised all of the DoS attacks.

The introduction of the LogitBoost algorithm was by Friedman *et al.* (2000) as an alternative solution to address the drawback of AdaBoost in handling noise and outliers. The LogitBoost algorithm uses binomial log-likelihood that changes the loss function linearly. On the other hand, the AdaBoost employed an exponential loss function that changes exponentially with the classification error. This further indicates why LogitBoost turned out to be less sensitive to outliers and noise. Furthermore, no research to date has investigated the performance of the LogitBoost algorithm in the field of ABDS. A comparison of previous work is presented in Table 2.3.

Table 2.3: Comparison of Previous Work (Ensemble Classifications)

Authors	**Techniques**	**Pros**	**Cons**	**Dataset**
Hu *et al.* (2008)	AdaBoost with decision stump	Low computational complexity	High false alarm rate	KDD 99
Panda and Patra (2009)	AdaBoost with Non-Nested generalised exampler	Effective in detecting rare attack such as U2R and R2L	Was not tested on unseen attacks	KDD 99
Folino *et al.* (2010)	AdaBoost with genetic programming	Suitable for distributed intrusion detection environments	Unable to perform in single environment or with limited information resources	KDD 99
Nguyen *et al.* (2011)	K-Means Clustering	Suitable for distributed intrusion detection environments	Was not tested on unseen attacks	KDD 99
Syarif *et al.* (2012)	Bagging, stacking and boosting	Suitable for detecting known attack types	i) Unable to detect new attack ii) No significant improvement on boosting and bagging in detection accuracy.	NSL KDD
Gaikwad and Thool (2015)	Bagged Partial Decision Tree scheme	Ensemble approach achieved better detection accuracy compared to using single classifier	Not suitable for online detection because it is more time-consuming to construct training model	NSL KDD
Folino *et al.* (2016)	Cellular Genetic Programming (CAGE)	The complexity load was reduced through parallel approach where the algorithm is independent from each other	The detection rate exceeds 90%, false alarm rate was not recorded	ISCX 2012

2.5.3 Summary

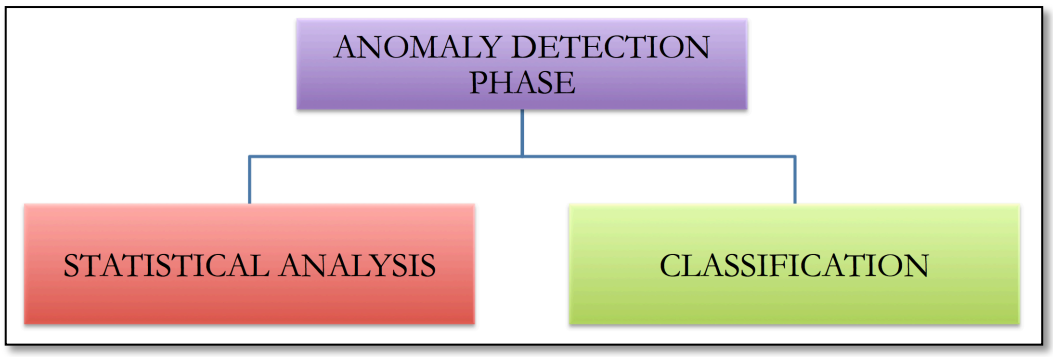


Figure 2.2: The components of the anomaly detection phase

Figure 2.2 shows the two components proposed in the anomaly detection phase. Following is the justification behind the proposed methods.

The statistical analysis adopted in previous work such as Mahoney and Chan (2001) and Chen *et al.* (2010) has revealed the capability of these methods to detect unknown attack. On the one hand, the detection approach does not require any attack information beforehand. On the other hand, the method would easily flag the normal data as an attack or vice versa (false positive and false negative) since it is highly dependent on outliers found in the traffic. Thus further analysis using traffic size is possible to overcome this drawback.

Based on the recent evidence shown in the literature, the classification approach is capable of achieving a high detection rate with a lower false detection rate compared to the clustering method (Jalil *et al.,* 2010; Aziz *et al.,* 2016). However, the discriminative model generated by classification is highly dependent on the features and the selection of the classification algorithm. Thus, it is important to select a suitable classifier to achieve a better detection capability. The usage of more than one classifier has been proven to produce better results (Folino *et al.,* 2010), for which the ensemble methods are further recommended to be investigated in this study.

2.6 Post-Processing Phase

In the IDS, the post-processing phase is frequently employed to manage and examine incidents (attacks) with the objective of storing and mapping them with appropriate action based upon their criticality. For example, true attacks detected are analysed and stored as historic information for future attack detection, where at the same time, the attacks are sorted according to their priorities.

2.6.1 Incident Prioritisation

An incident in the field of network security can be described as an event or intrusion detected by the system that may be a potential threat to or violation of the system (Anuar *et al.,* 2013). Hence, it is important to prioritise the most urgent incidents instead of the normal ones in order to enable a quick response from the security analyst. The first alert ranking computational model used in incident prioritisation is known as the M-Correlator which was introduced by Porras *et al.* (2002). There are two scores used by the model: relevance and priority scores. More specifically, the relevance score measures the validity of an incident, while the priority score calculates the severity of the incident by focusing on the assets worth. Similarly to Porras *et al.* (2002), Noel and Jajodia (2008) proposed an alert prioritisation model, whereby the metric measures the proximity of the targeted critical asset. Hence, the alert that is the closest to the critical assets would be assigned as top priority compared to those that are far away.

Another work by Zomlot *et al.* (2011) proposed a prioritisation model that is based on Dempster-Shafer (DS) theory. The function of DS is to measure the degree of belief for each alert that is generated by a correlation system to allow the incidents to be sorted. The experiment was performed using an open source IDS Snort rule. In another study, Anuar *et al.* (2013) employed a ratio scale approach to measure the weight of the incident critically. The ratio scale adopts a simple sequence of numbers which is 1, 2, 3, and 4, to illustrate the rank of the incident. In this case, the value illustrates that the value of 2 is twice the value of 1, while the value of 4 is two times better than the scale of 2 and four times better than the scale of 1. Meanwhile, Chakir *et al.* (2017) proposed an alert prioritisation model that is based on risk assessment to solve the issues related

to the high amount of alerts identified by IDSs. The Snort rule was employed and only highly critical alerts that represent real threats were presented to the security administrator. However, there are some limitations that need to be taken into account, although the aforementioned previous work had shown the ability to prioritise incidents. For instance, the datasets used were lacks of modern dynamic intrinsic network environments and the methods employed are solely based on network knowledge, which is restricted to ranking incident for a known attack signature (MBDS). As a result, incidents generated through ABDS remain unsorted.

To date, there are two unresolved problems that impede the effectiveness of the conventional anomaly detection system. First, the overall detection process requires a longer time to reinitiate the whole process in detecting similar observed behaviours. Second, the attack generated by the system must be analysed and processed by the security analyst for further incident response. In this case, the absence of proper attack severity (prioritisation) instruction will lead the security analyst to end up analysing the most recent attack first, or even worse if the detected attack turns out to be a false attack. As a result, it will lead to a more serious impact if the security analyst frequently overlooks the true detected attacks.

2.6.2 Summary

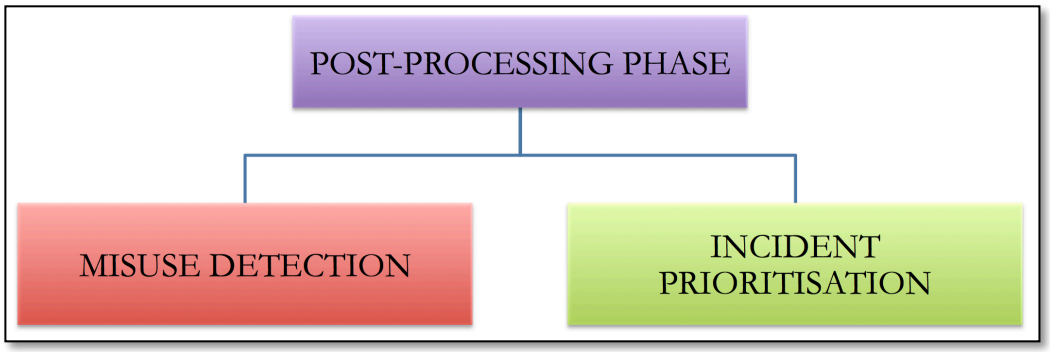


Figure 2.3: The components of the post-processing phase

Figure 2.3 shows the two components proposed in the post-processing phase. The following gives the justification behind the propose methods.

From the literature, the ability of the MBDS approach to quickly recognise a known attack that was previously detected has made it suitable to employ in the field of IDS (Meng and Kwok, 2014). The MBDS method is capable of achieving a high detection rate; however, it is limited to recognising known attacks (Louvieris *et al.,* 2013). Thus, in this study, the usage of MBDS as part of the detection strategy in the ABDS environment could reduce the re-initiation procedure for detecting similar attacks in the future.

In addition, prioritising attack traffic is important in order to obtain a quick response from the security analyst (Porras *et al.,* 2002; Noel and Jajodia, 2008). With a prioritising approach, the incident is measured based on its criticality. For instance, a critical incident requires a quicker response from a security analyst compared to less critical incidents. The advantages of prioritisation have been highlighted by Anuar *et al.* (2013) and Chakir *et al.* (2017); however, less focus has been given to adopting this technique in the ABDS environment. Thus, more attention should be given to the adoption of this approach in the ABDS environment.

2.7 Summary

This chapter has reviewed several aspects related to the topic of this study, which include the fundamentals of the IDS and a description of its deployment, detection category, and the techniques that are widely employed in this field. Apart from that, this chapter also underlined the current state-of-the-art on various methods such as feature selection, statistical analysis, and DM based detection that are adopted in the field of intrusion detection. A comparison table was generated to discuss the advantages, limitations, techniques, and the results of previous works.

Several limitations to the previous studies are acknowledged and further examined, while their advantages should be adopted as a guideline to design a solution that could enhance and improve the detection capability in general. In response, this study proposed a new detection scheme to address the aforementioned drawbacks.

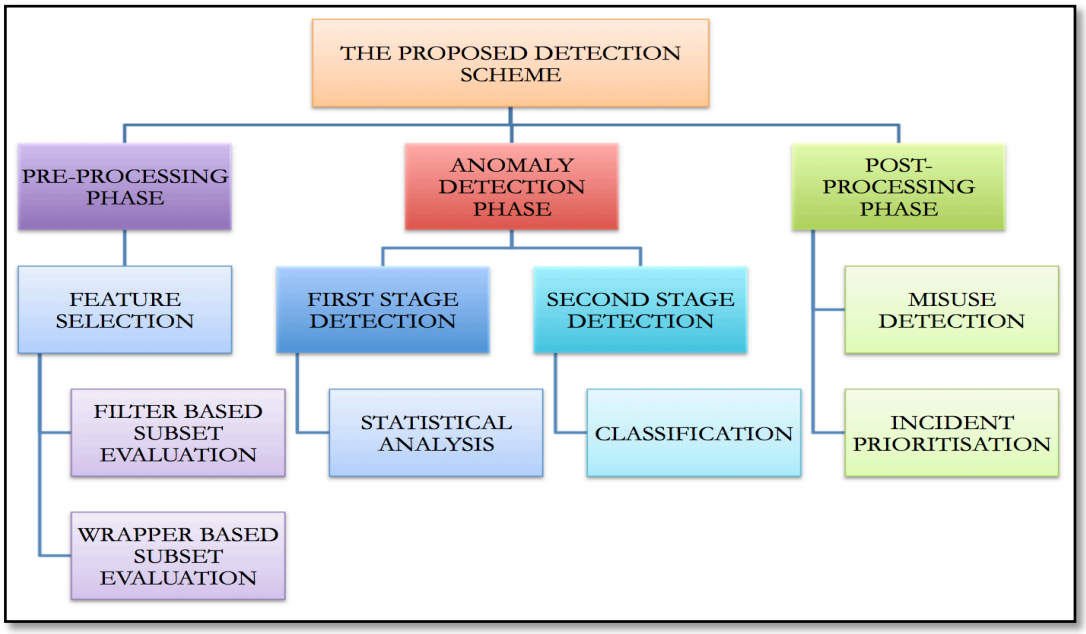


Figure 2.4: The conceptual framework

Figure 2.4 shows a conceptual framework of the proposed detection scheme that is divided into three phases, namely Pre-Processing (Feature Selection), Anomaly Detection (Statistical and Classification) and Post-Processing (Misuse and Incident Prioritisation). The proposed scheme is presented with the aim of addressing the overarching question: *"How could system detection performance be improved in order to identify known and unknown web attacks?"* The specific research questions mention in Section 1.4 are answered based on the following:

1) In the pre-processing phase, the FBSE method is adopted to overcome the issues related to feature ranking where the correlation between features is ignored (Talavera, 2005). The approach measures the merit of each feature to its class and eliminates any redundant features by exploring the strength of a relationship between the features (Wald *et al.,* 2013). In WBSE, the merits of each feature are optimised using the classification algorithm (Louvieris *et al.,* 2013). In this study, the Hybrid Feature Selection (HFS) is proposed with the purpose of improving the detection ability, while also reducing the data dimensionality. The HFS consists of a combination FBSE and WBSE methods by leveraging the strength of both methods to form a better-synergised approach.

2) In the anomaly detection phase, a 2-stage detection approach that consists of statistical analysis and ensemble classification methods is proposed. In the first stage detection, statistical analysis uses the normal profile utilised in Chen *et al.* (2010) as a benchmark for normal traffic behaviour. To identify the outliers in the traffic, Euclidean distance (ED) is employed, given its adequacy in computing basic distances. Meanwhile, the Chebyshev Inequality Theorem (CIT) is utilised to measure traffic regularity, whereas the mean and variance of traffic size from anomalous source traffic is extracted and compared with normal web traffic. The CIT has been chosen due to its excellent performance in defining the threshold in previous works, such as Thomas and Balakrishnan (2009) and Martignoni *et al.* (2010). Next, the second stage of the detection process is employed to complement the first stage detection using the ensemble classification technique. This approach shares similar objectives with the previous statistical approach, which is to achieve high detection accuracy with low false alarm rate. The ensemble classification consists of a LogitBoost classifier as meta-learning, and RF as a weak classifier. LogitBoost is one of the well-known ensembles boosting in the classification algorithm family. The advantage of LogitBoost over AdaBoost is its robustness in handling noisy and outlier data (Zhang and Fang, 2007).
3) The inability to re-initiate a detection procedure and attack processing has led to the creation of a signature with severity prioritisation model that is designed and proposed as part of the solutions in the post-processing phase. The true attacks detected by the system are extricated and transformed as attack signatures, which are then stored in the signature library to detect new entry web traffic (testing data). Hence, this further reduces the detection time because similar attack behaviour can be easily filtered out. Meanwhile, the intrusion prioritisation model (IPM) is introduced to ease the post attack analysis with the purpose of classifying each detected attack by severity level, namely highest, high, low, and lowest.

In the next chapter, the research design and experimental setup will be presented along with the justification for the methods selected.

Chapter 3

Methodology

3.1 Introduction

This chapter describes the methodology and methods used in this study. Section 3.2 discusses the research design proposed in this study. In Section 3.3, the experimental setup is outlined along with the experimental design and evaluation measurement. Finally, Section 3.4 summarises the overall development process.

3.2 Research Design

In this study, the quantitative approach is preferred as the main method due to certain characteristics, such as performance measures, dataset evaluations and the usability of the results. This research has employed a deductive cycle because it seems to be more appropriate to test the proposed solutions. Based on the proposed framework, the research design is categorised into 3-phases, namely Pre-Processing (Hybrid Feature Selection), Anomaly Detection (Statistical and Ensemble Classification), and Post-Processing (Signature and Severity Generation), which will further discussed in the next sub-sections.

3.2.1 Pre-Processing Phase

Feature selection is a part of the pre-processing phase in IDS that aims to reduce the data dimensionality by removing irrelevant and redundant features. According to El-Khatib (2010), there are two general methods for feature selection: filter and wrapper. Based on the literature, two types of filter methods are filter rank and FBSE. Filter rank is faster than FBSE, but its limitation in removing redundant features have made it unsuitable to employ in this study (Talavera, 2005). Mutual information and

information gain are examples of filter ranks. On the other hand, FBSE uses Correlation-based Feature Selection (CFS), which was found to be more effective in removing redundant features by measuring the correlation between the features and its class label (Bolón-Canedo *et al.,* 2015). Wrapper-based subset evaluation (WBSE) employed a specific algorithm to evaluate the merit of each feature which in turn could produce better selection (Louvieris *et al.,* 2013).

Due to the advantages of FBSE and WBSE, the Hybrid Feature Selection (HFS) is proposed in this study. The HFS consists of a combination of FBSE and WBSE methods by leveraging the strength of both methods to form a better-synergised approach. In view of both the filter and wrapper subset evaluation methods requiring a heuristic search to produce a feature subset, the four most employed search techniques in feature selection named: best-first, greedy stepwise, genetic search (GS) and particle swarm optimisation (PSO), are evaluated for choosing the best combination (Wald *et al.,* 2013) and (Khammassi and Krichen, 2017). Furthermore, five widely used classification algorithms named: MLP, SVM, NB, J48 and RF are examined with 10-fold cross validation as this could provide good generalisation performance, as suggested by De La Hoz *et al.* (2014).

3.2.2 Anomaly Detection Phase

Based on the literature, two widely used approaches in anomaly detection are based on statistical analysis and classification. In statistical analysis, a normal profile is created as a baseline that represents the normal traffic in order to examine the behaviour of incoming traffic by divulging irregular patterns. The usage of benign traffic as a profile is deemed more appropriate than the abnormal behaviour because the intruder tends to employ certain evasion techniques. The basic idea of generating a normal profile is proposed by Mahoney and Chan (2001) using a non-stationary model. The non-stationary model is developed based on the time of an event, which is highly dependent on its last occurrence. The traffics' discrepancy probability is calculated using statistical techniques. The techniques will then assigns an anomaly score function in order to determine the difference between anomalous and benign traffic. The advantage of this method is that it does not require any prior knowledge about the attack. On the other

hand, as it solely relies on a normal profile, this method suffers from outlier's drawbacks, which could easily flag normal traffic as attack traffic or vice versa (false positive and false negative).

The study performed by Chen *et al.* (2010) concluded that the non-stationary model is not suitable to detect similar attacks that occur in a different time scale. For instance, two *httptunnel* attacks share the same traffic content T and T', where T occurs 1 second after the previous attack, while T' takes place 30 mins after T. As such, the differences in T value between the two attacks (T=1 and T'= 1800) seemed to result in different anomaly scores. The different anomaly scores for both packets reflect the gap time that occurred between T and T'. As a result, the anomaly score for T' is 1800 times greater than T. As both attacks are sharing similar content, conveniently they should have similar anomaly scores. Therefore, if T' occurs before T and the threshold is set to a certain level of anomaly score, T might be ignored by the system after 'T' has been detected. Chen *et al.* (2010) introduce stationary models that ignore the time dependent scheme in order to address the problem of the non-stationary model.

Unlike statistical analysis, the classification technique uses both normal and attack traffic as a sample to develop a discriminative model to distinguish between legitimate and illegitimate traffic. This method is able to recognised unknown traffic effectively, but the decision is highly dependent on the selected algorithm. The two basic types of commonly used approaches are known as single classifier and ensemble classifier (Aburomman and Reaz, 2017). In a single classifier, the detection performance is measured using only one classifier or learner. In contrast to a single classifier, an ensemble classifier technique is a combination of multiple classifiers that is used to perform classification. The combination of more than one classifier is found to be superior than using a single classifier as it can capitalise on the strength of multiple classifiers (Wozniak *et al.,* 2014). Boosting algorithms are one of the ensemble classifications, which were first introduced by Freund and Schapire (1995). Generally, in boosting methods, the distribution of the training sets is adaptively changed depending on how complex it is to classify each instance. The main drawback of these techniques is that they have to reinitiate the training phase for several rounds. This will result in consuming more time and processing large datasets can be difficult.

Due to the advantages of statistical analysis and classification in recognising unknown attacks, both methods are adopted in this study as a hybrid approach. In this study, the hybrid approach consisting of statistical analysis as the first stage detection, followed by the classification technique as second stage detection, is proposed.

In first stage detection, this study adopted the previous work by Mahoney and Chan (2001) and Chen *et al.* (2010) in generating a normal profile. The purpose of choosing these approaches lies in the fact that they are able to demonstrate the degree of traffic characteristics. Nevertheless, the proposed approach is different in the following three ways from the approaches introduced by Mahoney and Chan (2001), and Chen *et al.* (2010).

1) First, superfluous and irrelevant features are eliminated using the proposed HFS method in the proposed scheme, while less attention is given to adopt this approach in the previous work (Mahoney and Chan, 2001), and (Chen *et al.,* 2010).
2) Second, the normal score is employed in conjunction with the traffic size in order to produce a better threshold mechanism. In this research, the normal score measurement is utilised instead of calculating the anomaly score. The main reason for calculating the normal score as an alternative to the anomaly score proposed by Mahoney and Chan (2001), Shamsuddin and Woodward (2007) and Chen *et al.* (2010) is because the latter is not sufficiently sensitive to consider new attribute values. Chen *et al.* (2010) emphasise that benign traffic is likely to have a higher distinct attribute value than malign traffic. Furthermore, in real environments, there is more benign traffic compared to malign traffic. Thus, analysing the degree of normal field values in the traffic is more appropriate and easier, rather than analysing attack traffic.
3) Third, in the previous works, a statistical approach has been solely employed for attack detection. This approach has greater potential to generate a high false alarm rate, considering that its high dependency on outliers exists within the traffic.

To overcome the drawbacks of outliers' pre-set threshold in the conventional statistical approach, two methods are proposed in this study. The first method employs Euclidean distance (ED) to measure the distance between normal and attack traffic due to its simplicity in calculating distance between two points (Mitchell and Chen, 2014). The second method adopted is the Chebyshev Inequality Theorem (CIT) for threshold measurement. The previous work by Thomas and Balakrishnan (2009) and Martignoni et al. (2010) has highlighted the advantage of using CIT considering that it does not rely on the knowledge of how the data are distributed.

In second stage detection, a classification technique that employs the ensemble classification approach is proposed. The boosting technique is proposed in this study over other ensemble approaches such as bagging and stacking due to its excellent performances demonstrated in the previous work by Hu *et al.* (2008), Li and Li (2010) and Syarif *et al.* (2012). An example of a boosting algorithm that is commonly used is known as the AdaBoost classification method. However, this method is not suitable when dealing with outlier and noisy data, despite its good performance in classifying instances. The AdaBoost uses a loss function that changes exponentially with the classification error which will make it sensitive to noise and outliers (Cao *et al.* 2012). Unlike AdaBoost, LogitBoost uses log-likelihood that changes the loss function linearly to be less sensitive to outliers and noise (Li and Bradic, 2018). In view of the AdaBoost limitation, LogitBoost is proposed as meta-classifier due to its strength in handling noisy and outlier data compared to the AdaBoost algorithm (Zhang and Fang, 2007). To date, no research has investigated the performance of the LogitBoost over AdaBoost algorithm in the field of ABDS environment.

3.2.3 Post-Processing Phase

In IDS, the post-processing phase is usually a phase when attacks that have been identified by the system are further processed. The re-initiation procedure of detecting similar attacks in the future seldom disregard which will result in consuming more time and resources. The advantage of the signature approach is to reduce the detection time for detecting similar attack in the future (Meng and Kwok, 2014). Thus in this study, the signature approach is proposed as part of th detection strategy whereby the true

attacks detected by the system are transformed into a set of signatures. Further analysis is important to make sure the detected attack is responded to as soon as it has been identified. Hence, the requirement to propose attack prioritisation is vital in the field of ABDS. Previous studies have focussed on prioritising known attacks where fewer studies have adopted this technique for prioritising unknown attacks (Noel and Jajodia, 2008; Zomlot *et al.,* 2011; Anuar *et al.,* 2013). In this study, the intrusion prioritisation model (IPM) is proposed to sort unknown attacks from the most critical attacks followed by the less critical attacks according to four-severity levels: highest, high, low and lowest.

3.2.4 Data Source Selection

The current need to employ more than one synthetic dataset is caused by several factors, including dataset age, data size, updated malicious activity and new attack portion residing in the test data. On top of that, all these elements will result in different complexity between the datasets. Based on the aforementioned factors, four synthetic publicly available datasets, namely DARPA 1999, NSL KDD, ISCX 2012 and UNSW-NB15, have been chosen to evaluate the proposed methods in this study. These datasets are labelled and have been used as a standard benchmark by many researchers in this field.

DARPA 1999 and NSL KDD were generated in early year 2000. Basically, they consist of massive traffic volume with different attack types. In most cases, these datasets are still relevant due to their complex characteristics of having varieties of attack types. In such an event, it makes the comparison tasks against other approaches easier. Moreover, the advancements in modern network technology has further emphasised the increased need for more updated datasets. Despite heavy criticism by McHugh (2000) and Brown *et al.* (2009) on their limited ability and accuracy in demonstrating real-live traffic, both datasets continue to be extensively adopted in this field.

In view of the research communities increasing demand for more and recent datasets, the performance of the proposed approaches is evaluated using the updated benchmark ISCX 2012 and UNSW-NB 15 datasets. The recently published datasets ISCX 2012 and

UNSW-NB 15 are incorporated with various types of recent sophisticated attacks which are claimed to be more realistic in the modern network traffic environment. The use of updated datasets for the proposed approaches will create a new platform for the future benchmark. The next sub-section describes in detailed the datasets used in this study.

3.2.4.1 DARPA 1999

MIT Lincoln Lab has made DARPA 1999 dataset publicly available. The 5-week dataset consists of three weeks of training and two weeks of testing data. The traffic is captured in tcpdump format and contains comprehensive TCP/IP information which is useful for traffic analysis. Weeks 1, 2 and 3 represent benign traffic and free from attack. Meanwhile, in weeks 4 and 5, the data contain attacks in the middle of benign traffics.

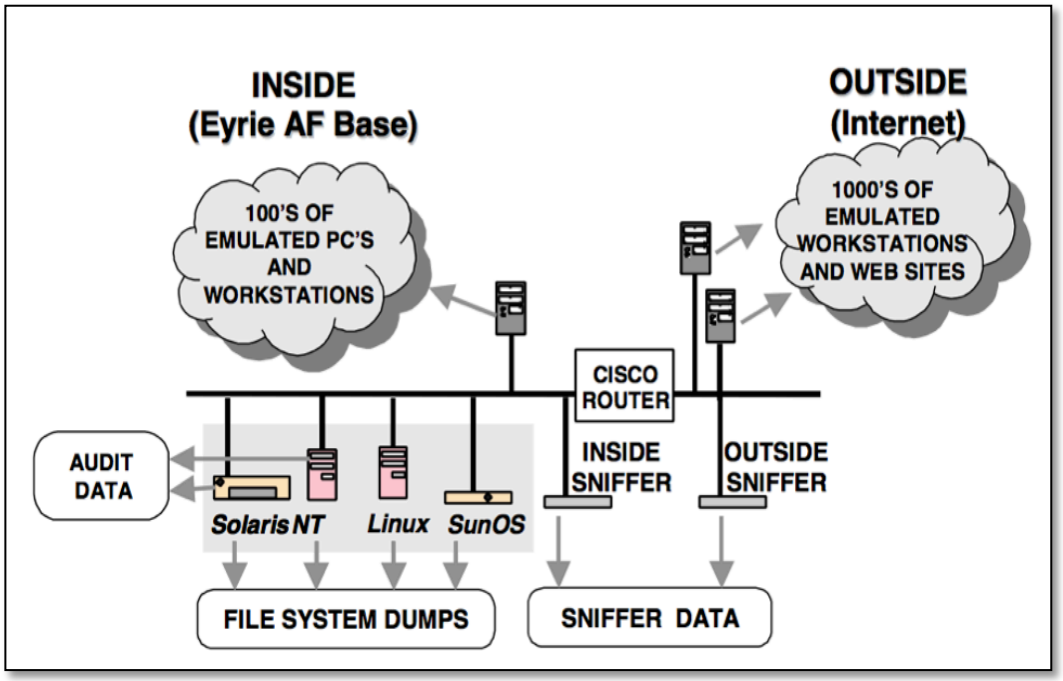


Figure 3.1: Block diagram of DARPA 1999 test bed (Lippmann *et al.,* 2000)

Figure 3.1 shows the generated simulation based on a scripting technique that is performed to produce live benign and attack traffics. The scenario is equivalent to the flowing traffic from the internal Eyrie Air Force Base (EAFB) to the Internet at large. The test bed generates rich background traffic to simulate as the traffic is initiated by

thousands of hosts from hundreds of users. All attacks are set to be automatically launched against the targeted machines (UNIX OS) and the external hosts router. The sensors known as 'sniffers' are placed within the internal and external network to capture all the traffic that is broadcasted through the network.

Table 3.1: Distribution of Web Traffic for DARPA 1999 Dataset

Dataset	Date	Normal Traffic	Attack Traffic
Training Week 4	03/29/1999	8,998	728
	03/30/1999	101	643
	03/31/1999	5,202	456
	04/01/1999	11,413	605
	04/02/1999	0	0
	04/03/1999	0	0
	04/04/1999	0	0
Testing Week 5	04/05/1999	6,632	723
	04/06/1999	6,873	993
	04/07/1999	5,800	1,807
	04/08/1999	77,039	640
	04/09/1999	0	8,073
	04/10/1999	174	62
Total	136,962		

In the DARPA 1999 dataset, both weeks 4 and 5 data have different attack distributions. In other words, some of the attacks in week 5 do not appear in week 4. The different attack distribution provides an opportunity for researchers to seek methods that can detect new or novel attacks. According to the total traffic generated, 28,146 http traffic are produced in week 4 and 108,816 http traffics are from week 5, as described in Table 3.1.

3.2.4.2 NSL KDD

The Network Security Laboratory Knowledge Discovery and Data Mining (NSL KDD) dataset was generated by Tavallaee *et al.* (2009) based on the KDD 99 dataset. The dataset is part of the DARPA 1998 Intrusion Detection System (IDS) Evaluation dataset project that was created by Lincoln Lab (Lippmann *et al.,* 2000). The lab simulates the traffic environment using artificial data in a closed network environment. Some of the networks are proprietary network traffic with manually injected attacks.

Table 3.2: Distribution of Web Traffic for NSL KDD Dataset

NSL KDD dataset	**Training Data**		**Testing Data**	
	Normal	Attack	Normal	Attack
	3,817	683	2,856	2,785

The simulation is a replication of the medium sized traffic found in US Air Force bases in collaboration with Air Force Research Laboratory (AFRL). Since KDD 99 suffered from some drawbacks, the dataset has been revised by Tavallaee *et al.* (2009) to remove the duplicated and redundant traffic within the dataset. NSL KDD has managed to undergo further improvement through the removal of 78% and 75% of duplicated traffic in the training and testing data respectively. Table 3.2 presents the reduced dataset generated, with a respective total of 4,500 and 5,641 instances in the training and testing datasets of http traffic.

3.2.4.3 ISCX 2012

The ISCX 2012 dataset was developed by Shiravi *et al.* (2012) from University of Brunswick (UNB) with the aim of addressing the issues in other existing datasets such as DARPA, CAIDA and DEFCON. The distribution model is based on the dataset effectiveness in relation to realism, evaluation, malicious activity and capabilities. A considerable number of multi-phase attacks events are induced to create the anomaly trace for the dataset which include HTTP DoS, Botnet, Distributed Denial of Service (DDoS) and Brute Force SSH. The simulation is performed by mimicking user behaviour activity. Meanwhile, profile-based user behaviour is created by executing a

user-profile that was synthetically generated at random synchronised times. The dataset is labelled to assist the researcher in the process of testing, comparison and evaluation.

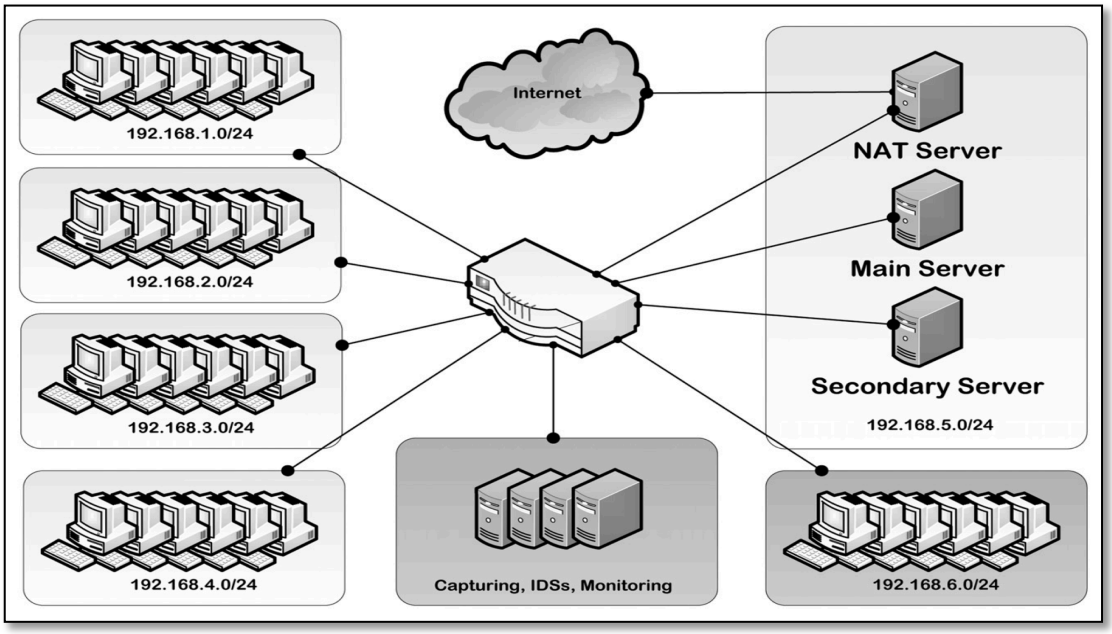


Figure 3.2: ISCX 2012 Testbed Network Architecture (Shiravi *et al.,* 2012)

Figure 3.2 shows the ISCX 2012 test bed network, which contains a total of 21 interconnected Windows workstations. Those workstations are equipped with Windows operating system as a platform to launch attacks against the test bed environment. More specifically, a total of 17 workstations are installed with Windows XP SP1, two with SP2, one with SP3 and a workstation with Windows 7. The network architecture divides the workstation into four distinct LANs with the purpose of representing a real connectivity network environment. The servers located at the fifth LAN provide Web, E-mail, Domain Name Server (DNS), and Network Address Translation (NAT) services. The NAT server (192.168.5.124) is placed at the entry point of the network to ensure that the firewall only permits authorised access. The primary main server (192.168.5.122) is accountable for email services, delivering websites and performs as internal name resolver. The secondary server (192.168.5.123) is made responsible for handling internal ASP.NET applications. It sits on Windows Server 2003 machines. It is important to note that both main and NAT servers run on Linux operating system and are configured with Ubuntu 10.04.

Table 3.3: Distribution of Web Traffic for ISCX 2012 Dataset

Date	**Training Data**		**Testing Data**	
	Normal	Attack	Normal	Attack
6/11/2010	0	0	0	0
6/12/2010	528	0	2,074	0
6/13/2010	0	84	0	108
6/14/2010	826	873	782	1,096
6/15/2010	1,468	2,757	1,973	27,125
6/16/2010	432	0	1,237	0
6/17/2010	1,032	0	562	0
Total	4,286	3,714	6,628	28,329

The whole datasets are captured for seven days, with 3-days of attack-free traffic and 4-days mixed benign and malign traffic. Table 3.3 illustrates the portion of training and testing dataset which respectively consist of 10,914 and 32,043 for both normal and attack web traffic in the ISCX 2012 dataset.

3.2.4.4 UNSW-NB 15

The UNSW-NB 15 was simulated by Moustafa and Slay (2016) using the IXIA PerfectStorm tool in the Cyber Range Lab at the Australian Centre for Cyber Security (ACCS). The dataset is developed based on the combination of synthetic attack activities along with real modern normal behaviours.

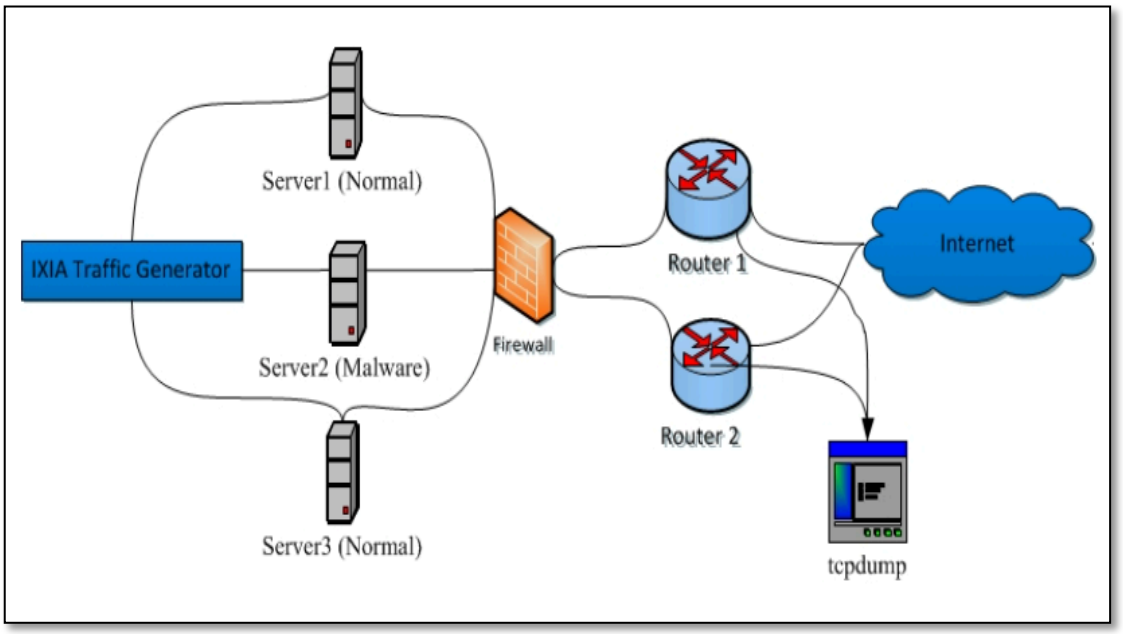


Figure 3.3: UNSW-NB15 Testbed Network Architecture (Moustafa and Slay, 2016)

Figure 3.3 illustrates the test bed configuration of the UNSW-15 dataset. The full dataset contains captured raw traffic of 100GB with the following nine synthetic types of attacks: Backdoors, DoS, Analysis, Fuzzers, Generic, Worms, Reconnaissance, Shellcode and Exploits.

Table 3.4: Distribution of Web Traffic for UNSW-NB 15 Dataset

UNSW-NB 15 Dataset	**Training Data**		**Testing Data**	
	Normal	Attack	Normal	Attack
	4,013	4,274	5,348	13,376

The features and the class label are generated using Argus and Bro-IDS tools, in conjunction with to the 12 algorithms. The total recorded traffic captured is 2,540,044. However, several parts of these data are divided into two sets, namely training and testing that consist of 175,341 and 82,332 instances respectively. As shown in Table 3.4, a total of 8,287 and 18,724 of http traffics is obtained from training and testing data respectively.

3.3 Experimental Setup

The details of the experimental setup, design and evaluation measurement of the proposed detection scheme are discussed in the following sub-sections.

3.3.1.1 MySQL

In this study, the amount of web traffics applied for evaluations are varied. The experiments are conducted using MySQL Database Management System (DBMS) for the purpose of managing a significant amount of web data traffic. Moreover, it also aims to assist data management and analysis tasks. In addition, the DBMS has simplified the data analysis process in terms of time taken and efforts spent compared to the conventional filing approach. The DBMS is also used to store true attack signatures, which can simplify future attack detection.

3.3.1.2 WEKA Data Mining Tools

The data are further processed and analysed using the statistical WEKA DM tool (Frank *et al.,* 2016). WEKA is described as a collection of machine learning algorithms and pre-processing tools that was developed at the University of Waikato in New Zealand. Moreover, WEKA is written in Java under GNU general public license. Moreover, processing data using WEKA DM tools is straightforward compared to other machine learning workbenches because it contains a variety of classifiers for feature analysis and classification. In addition, it is easier for beginners as the package comes with GUI without having to deal with the programming task. The WEKA is chosen as a tool in this study because it has been widely adopted by many researchers due to its flexibility of open source license and availability of many classifiers (Amancio *et al.,* 2014).

3.3.2 Experimental Design

Experiment 1: Hybrid Feature Selection

The HFS design involves the combination of the strength of both FBSE and WBSE. In HFS, four different search techniques are compared to find the finest search method that could produce the highest detection accuracy rate, namely best first, greedy, genetic search, and PSO. The preliminary experiments are conducted by eliminating irrelevant and redundant features in order to select the optimal features.

Initially, the purpose of employing FBSE is to reduce the computational effort of WBSE by filtering the insignificant and redundant features. In addition, the process is continued with a search for the optimal subset in order to improve the classification performance selected earlier by the FBSE. The final features subset generated from the hybrid process is tested using RF classifier and 10-fold cross-validation.

Experiment 2: Statistic based Anomaly Detection (First stage detection)

The experiments and analysis are carried out to find the finest threshold in distinguishing normal and abnormal web traffic. The thresholds are defined by the combination of ED and CIT. The usage of ED is to measure the distance of each testing and normal data as presented in the standard profile. On the other hand, the CIT is used to produce the finest threshold by calculating the mean distance and how far it deviates from the normal data. The goal of producing the finest threshold is to obtain better normal and attack detection rates. Although the statistics method is capable of demonstrating some level of detection ability, it is affected by numerous voluminous false alarms. The setback is due to the non-existence of an attack sample in the training stage. Thus, further improvement using ensemble classification algorithm is necessary.

Experiment 3: Single Classification Algorithms (Second stage detection)

The classification algorithm is introduced to improve the false alarm and detection accuracy that was achieved in the first stage detection. A preliminary experiment is conducted by evaluating six single classifiers to select the best one (e.g. MLP, J48, DT, RF, SVM, and NB). The single classifier that is able to achieve the highest performance will be further induced with LogitBoost to achieve better detection accuracy and maintain low false alarms.

Experiment 4: Ensemble-based Classification

In the boosting algorithm, LogitBoost is chosen as meta learning instead of AdaBoost due to its robustness. The best classifier identified in the previous experiment is chosen to be induced with the LogitBoost classifier. The comparitive performance between the chosen individual classifier and the previously adopted AdaBoost is presented. The training model is built using the set of training data and 10-fold cross-validation approach.

Experiment 5: Misuse-based Web Attacks Detection

The signature generation process is to detect any identified attack behaviour in the past. An evaluation is conducted to further assess the use of signature in detecting attack. The purpose of generating the signature is to reduce the whole re-initiation procedure. The true attack (true positive) identified during the anomaly detection phase will be transformed into signature for the purpose of future attack detection.

3.3.3 Evaluation Measurement

This sub-section analyses the performance metrics used in this study. The main performance metrics used in the IDS field are measured in terms of their detection, accuracy and false alarm rates. However, to compute the main performance metrics, other major indicators such as (True Positive, True Negative, False Positive and False Negative) are also needed. Thus, four indicators are considered as the main contributors because the detection capability is highly dependent on the value produced. For instance, misclassification rate (false positive and false negative) is caused by the system that mistakenly flagged normal data as attack and vice versa and would reduce the detection accuracy. Conversely, a low misclassification rate indicates higher data detection accuracy rate. The experiment output for the current research is analysed with additional performance metrics, which include false alarm rate, attack detection rate, normal detection rate, and accuracy. It is worth noting that the time taken by the classification model to identify attack records is measured as detection time.

The following describes the four standard performances used in the intrusion detection evaluation of the proposed detection scheme:

I. True Positive (TP) is described as the quantity of true attack data that has been flagged correctly.
II. True Negative (TN) refers to the quantity of true normal data that has been classified correctly.
III. False Positive (FP) is defined as the quantity of normal data that was falsely detected as attack data. Ideally, the detection system should achieve a lower false detection rate for better incident handling responses.
IV. False Negative (FN) is the quantity of attack data that was falsely detected as normal data. The attack can be damaging due to its failure to be detected by the system. Generally, FN is difficult to compute as no flaw can be found by the IDS when it happens. An ideal detection system should achieve a lower FN, particularly close to zero.

An ideal IDS requires high detection and good detection rates, including the need to maintain low false detection rates. The additional performance metrics used to evaluate the proposed approach performance are shown in the following equations:

$$\text{False Alarm Rate (FAR)} = \frac{\text{FP}}{\text{FP} + \text{TN}} \tag{3.1}$$

$$\text{Attack Detection Rate (A} - \text{DR)} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{3.2}$$

$$\text{Normal Detection Rate (N} - \text{DR)} = \frac{\text{TN}}{\text{TN} + \text{FP}} \tag{3.3}$$

$$\text{Accuracy (ACC)} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{3.4}$$

I. FAR is a total percentage of normal data that was falsely detected as actual attack data from the total of the normal data
II. A-DR is the total percentage of true attack detected over the total attack available. An ideal detection system is capable of achieving attack records close to 100%.
III. N-DR is the total percentage of actual normal data detected over the total normal data available.
IV. The ACC rate is a total percentage of true attack and normal data detected with the exclusion of false detection

3.4 Summary

In this chapter, the general methodology is presented, which includes research design and experimental setup for the proposed detection scheme. The rationale of designing the solution has been briefly explained along with the performance metrics that are used to evaluate the proposed scheme. In addition, a more specific detail for each contribution of the proposed detection scheme and its implementation are explained in detail in Chapters 4 and 5 respectively.

Chapter 4

An Intrusion Detection Scheme for Identifying Known and Unknown Web Attacks (I-WEB)

4.1 Introduction

As briefly discussed in Chapter 2, various approaches that utilise feature selection, statistical-based anomaly detection and ensemble classification have yet to achieve satisfactory performance and should be further improved. This chapter aims to highlight the proposed scheme based on the reviewed literature and studies in the intrusion detection field. The rest of the chapter is organised as follows: Section 4.2 describes the overview of the proposed scheme. Sections 4.3, 4.4 and 4.5 present the details of the proposed detection scheme with the aim of minimising data dimensionality, improving the known and unknown web attack traffic, and simplifying the re-initiation process for future detection. Section 4.6 summarises the overall proposed detection scheme.

4.2 The Proposed I-WEB

In this study, several methods are proposed for feature selection, statistical analysis, ensemble classification and signature detection specifically for better attack detection capability. The proposed detection is developed based on the advantages and limitations identified from previous works. The proposed scheme is divided into three phases, namely Pre-Processing (Hybrid Feature Selection), Anomaly Detection (Statistical and Ensemble Classification) and Post-Processing (Signature and Severity Generation). The details of each phase are presented in the next sections.

4.3 Pre-Processing Phase

This research proposed the novel hybrid approach by leveraging the strength of both filter and wrapper-based selections to form a better synergies approach (Kamarudin et al., 2017a). The hybrid selection aims to select the most prominent features that can reduce data dimensionality together with the purpose of maximising the accuracy of the classifier. In the pre-processing phase, FBSE is adopted due to its ability to measure the redundancy level among features (Bolón-Canedo *et al.,* 2015). Moreover, the approach measures the merits for each features towards its class as well as eliminates redundant features by exploring the degree of relationship between the features (Wald *et al.,* 2013). The main purpose of employing FBSE along with WBSE is to reduce the WBSE complexity by only processing the reduced set of features instead of all the original features.

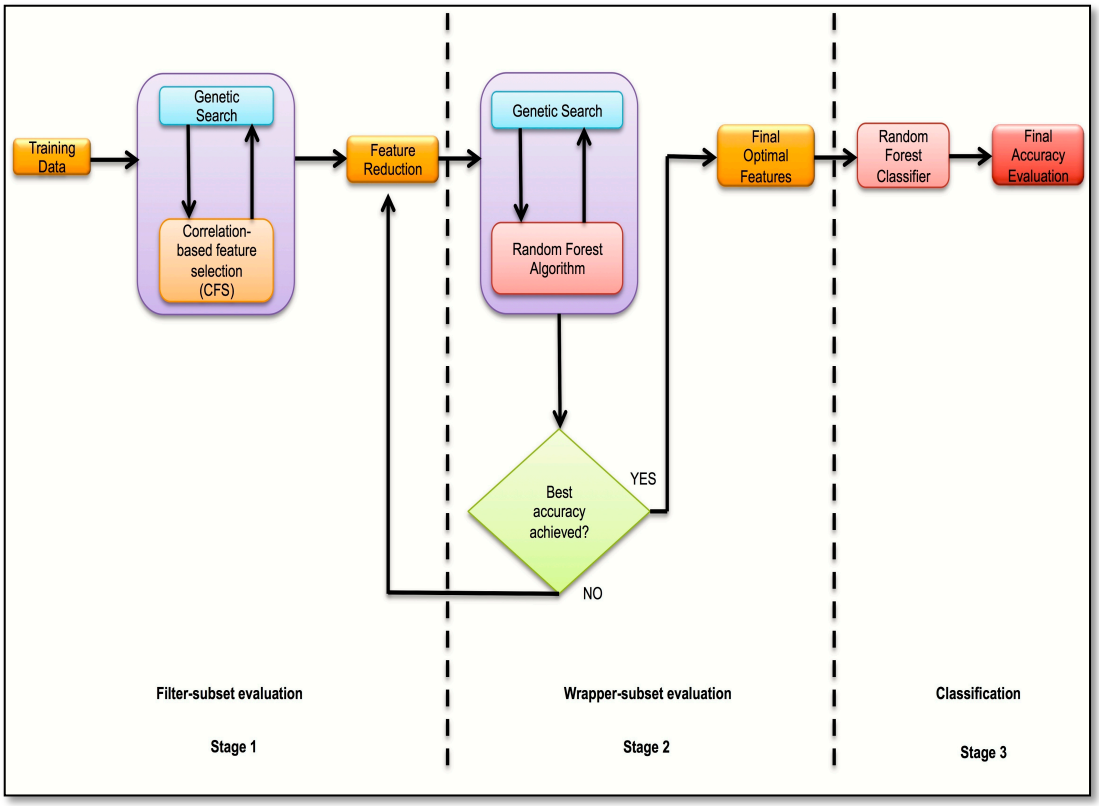


Figure 4.1: Hybrid Feature Selection (HFS) design (Kamarudin et al., 2017a)

Figure 4.1 shows the process flows in building HFS. The process is classified into three stages and these are explained in the next sub-sections.

4.3.1 Filter-subset Evaluation (Stage 1)

In this stage, the FBSE method is employed to process the original features *M* and produce a new set *L* of reduced features, where $L \subseteq M$. The CFS approach is adopted to measure the correlation between features and features to class due to its robustness in removing redundant and irrelevant features (Bolón-Canedo *et al.,* 2015). This approach is able to overcome the issue of redundant features because the relationship between features is measured using eq. (4.1). The CFS is described as an intelligible filter algorithm that evaluates subsets of features based on heuristic evaluation functions.

$$Ms = \frac{k\overline{rcf}}{\sqrt{k + k(k-1)\overline{\overline{rff}}}} \tag{4.1}$$

Eq. (4.1) shows how the merit function, *M*, is used to select a subset *S* that contains *k* number of features. Both redundant and irrelevant features are determined by the $\overline{\text{rcf}}$ which represents the mean of the relationship of each feature to its class while $\overline{\overline{\text{rff}}}$ is represented as the mean of the relationship among the features.

The deployment of a high complexity exhaustive search is not feasible because it only works in a small dataset (Guyon, 2003). Alternatively, heuristic search techniques are employed and Genetic Algorithm (GA) is selected as the search function. This is because the preliminary experiment reveals that GA is able to provide a global optimum solution and is more robust than the best-first, greedy and PSO search methods. This stage is crucial as the wrapper computational effort will be truncated because it only deals with the reduced set of features instead of the original set.

4.3.2 Wrapper-subset Evaluation (Stage 2)

In this stage, the reduced feature set *L* gathered from the FBSE is further processed with WBSE in order to produce the final optimal features *K,* where $K \subseteq L \subseteq M$. The

proposed hybridisation approach leverages the strengths of each to produce a much better result in terms of accuracy, false detection rate and fewer redundant and irrelevant features. This is because the filter approach alone is unable find the best available subset, since it is less dependent on the classifier (Peng *et al.,* 2010). On the other hand, the wrapper approach is proven to be more effective and able to produce better accuracy (Wahba *et al.,* 2015). Nevertheless, it is computationally expensive when dealing with large datasets. In view of the above limitations, the strength of both methods is leveraged to form a better-synergised approach. In WBSE, an RF classifier is used to evaluate the selected features along with the genetic search in order to determine the final *K* feature subset. The searches will continue to train a new model for each subset and will only stop once the final optimum subset is found.

4.3.3 Classification (Stage 3)

In the final stage, the final optimum subset *K,* produced by WBSE is tested using the RF classifier with 10-fold cross-validation. The overall classification accuracy derived from HFS is then compared with the performance of all original features.

In this study, the feature selection procedures are conducted using training data that consist of a mixture of normal and attack traffic. The significance features are measured using a correlation function in the filter process. The features that managed to achieve high merit scores and are highly correlated to the class will be selected. Conversely, those features that are highly correlated with other features are considered to be redundant. Meanwhile, a classifier is used to identify subsets of relevant features in the wrapper process. As such, irrelevant and redundant features will be removed in Stages 1 and 2. Further analysis on the features selected by the proposed method will be further discussed in Chapter 6: Results and Discussion.

4.4 Anomaly Detection Phase

In the anomaly detection phase, a 2-stage detection strategy comprising statistical and ensemble classification approaches is proposed and briefly explained in the next sub-section.

4.4.1 Statistical-based Anomaly Detection (First stage detection)

Work undertaken in the past had indicated how the abnormalities can be identified using header traffic (Mahoney and Chan, 2001; Chen *et al.,* 2010); however, it still does not take into account the influence of packet size as the additional steps to differentiate between benign and abnormal traffic. Therefore, in this research, the attack detection is computed by calculating the traffic normality along with the analysis of the feature size through statistical analysis.

4.4.1.1 Normal profile

Normal profile is described as a profile that represents a benchmark of normal characteristic behaviour (Shenzheng, 2009). The profile contains attributes scores that represent normal web traffic which is created using distinct values of attack-free traffics that consist of historic information and unique values for each host within the network. In addition, the profile is used as a benchmark of normal web traffic against incoming web traffic. Previous works for example Mahoney and Chan (2001), Shamsuddin and Woodward (2007) and Chen *et al.* (2010) measured packet abnormality by summing up the anomaly scores given to the traffic field. This process of identifying the outliers was unfeasible since it relied solely on the anomaly score without further analysing the observed traffic. To overcome the aforementioned issues, a normal score is proposed along with further analysis on the feature size.

Table 4.1: Normal Profile of DARPA 1999

k	Features/Attributes Label	Rk	Nk	Normal Score
1	ethersize	235	53533	**0.0836**
2	ethersourcehi	4	53533	**0.1463**
3	ethersourcelo	5	53533	**0.1429**
4	iplength	36736	53533	**0.0058**
5	ipfragid	236	53533	**0.0835**
6	ipsource	15	53533	**0.1259**
7	tcpsourceport	5134	53533	**0.0361**
8	tcpheaderlen	2	53533	**0.1569**
9	tcpflag	5	53533	**0.1429**
10	tcpwindowsize	382	53533	**0.0761**
Total Score		**1**		

During the first stage detection, the reduced features derived from HFS are processed, while the normal score for each traffic feature is measured based on eq. (4.2). In relation to this, the normal profile for DARPA 1999 is demonstrated in Table 4.1.

$$Normal\ Score = \sum_{k=1}^{n} (\log_{10} Rk - \log_{10} Nk) \qquad (4.2)$$

The attributes are indexed as k, where k = 1,2,3,4,…n. Rk is a distinct accumulation of normal packet characteristic while Nk refer to the total number of traffics related to each attribute. As Rk and Nk vary greatly, the score is computed in the form of a logarithm.

$$Traffic\ Normality\ Score = \sum_{k=1}^{n} Mi * 100 \quad (4.3)$$

The traffic normality score is utilised to determine the degree of normality for every traffic as computed in eq. (4.3). Let *Mi* be equivalent to the normal score of feature as *k*, where *k* = 1,2,3,4,…*n*. Each feature score is converted into a percentage to indicate the basic pattern of the traffic either normal or anomalous.

Table 4.2: Example of Computation Score for Traffic (*n*) in DARPA 1999

Http Traffic *n*				
k	**Attributes field name**	**Passive Score**	**Active Score**	**Traffic Normality Score**
1	ethersize	0.0836	0.0000	0.00
2	ethersrchi	0.1463	0.1463	14.63
3	ethersrclo	0.1429	0.1429	14.29
4	iplentgh	0.0058	0.0058	0.58
5	ipfragid	0.0835	0.0835	8.35
6	ipsource	0.1259	0.1259	12.59
7	tcpsrcport	0.0361	0.0000	0.00
8	tcpheaderlen	0.1569	0.0000	0.00
9	tcpflag	0.1429	0.1429	14.29
10	tcpwindowsize	0.0761	0.0761	7.61
Total Normal Field				7
Total Anomalous Field				3
Distance Value				0.1814
Traffic Normality Score				72.34%

Table 4.2 presents the feature with its score value, including the newly generated features such as anomalous, normal, distance value and traffic normality score created during the matching procedure. These newly generated features are also known as derivative features. According to Louvieris *et al.* (2013), the information composition may improve the performance of detection ability for both normal and abnormal behaviours. Next, the feature that contains normal scores other than a zero score is counted as a normal field while the remaining are labelled as anomalous field.

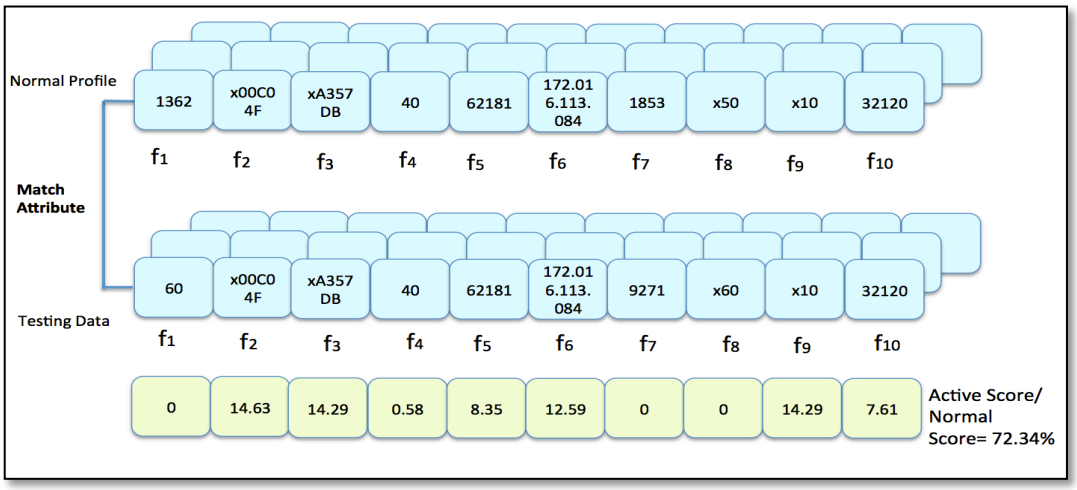


Figure 4.2: An example of attributes matching on testing data (DARPA 1999)

Figure 4.2 illustrates the score derived from matching procedure between normal profile and testing data. The basic Euclidean Distance (ED) is proposed to measure the distance between passive and active scores. The ED is employed as it is adequate to compute a basic distance between the two points (benign and outliers) (Mitchell and Chen, 2014). The distance is computed by transforming PS and AS into data point, whereby PS is converted into a Passive Point (PP) while AS is transformed into an Active Point (AP). In short, PS represents normal behaviour while AP is described as mixed traffic that contains normal and abnormal behaviour. On top of that, the degree of normality is defined by calculating the distance between active and passive data point. The distance between active and passive data points is computed as follows:

$$Euclidean\ Distance = \sqrt{(A1 - P1)^2 + (A2 - P2)^2 + \cdots (An - Pn)^2} \quad (4.4)$$

Thus, the distance between AP and PP can be simplified into:

$$Distance\ AP = \sqrt{\sum_{k=1}^{n} (Ak - Pk)^2} \quad (4.5)$$

where Ak is the active point, while Pk represents the passive point and n is the total number of features. The traffic will be notified as suspicious if the distance of the tested traffic differs from that of the benign traffic.

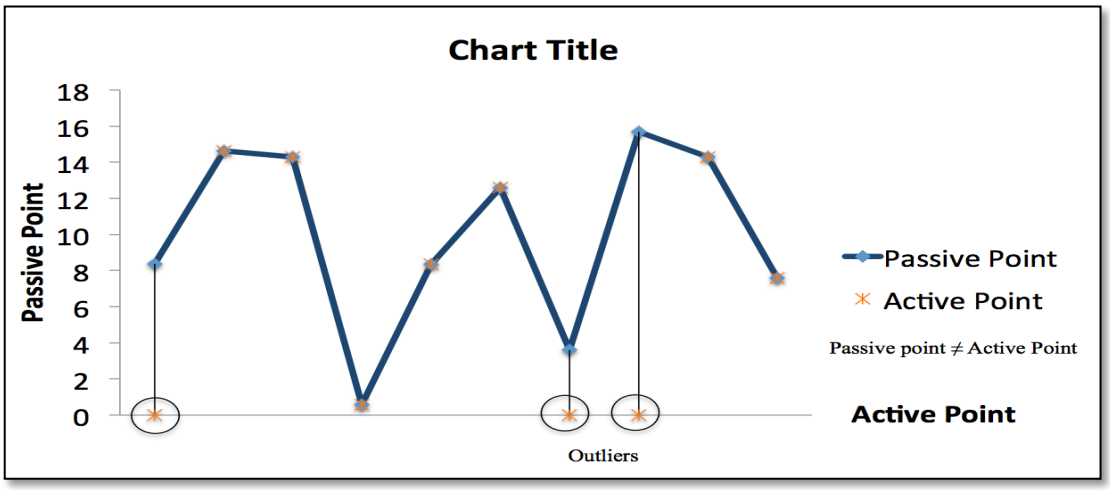


Figure 4.3: Example of Anomalous traffic behaviour (DARPA 1999)

Figure 4.3 presents an example of anomalous traffic behaviour in which active points are separated from passive points, which leads to the production of some outliers. If the threshold depends only on the outliers and scores, the tendency of normal traffic to be recognised as anomalous traffic is high. In reality, the number of outliers that exist within the traffic is unknown. Thus, it will be good to perform further traffic behaviour analysis for a better threshold mechanism instead of depending solely on the score. Further analysis should be performed to measure the feature size of each suspected anomalous traffic. To do this, the flagged anomalous source is further compared against its normal records. Previous works performed by Estévez-Tapiador *et al.* (2004), Kruegel *et al.* (2005), Yamada *et al.* (2007), Zhang and White (2007) and Louvieris *et al.* (2013) have proven that the feature size (in bytes) can be used to measure traffic regularity. Moreover, this fact has been validated by the nature of client-server input service request.

4.4.1.2 Influence of Feature Size

In normal client-server access, the increase in packet size will be filled with a small number of bytes when the requests are made from the same source address. In return, the server will respond with a large number of bytes. Thus, a large number of requests can be considered as or suspected to be abnormal. For that reason, the inconsistent input size is expected to cause anomalous action. This normally happens when malicious input is bound together within the legitimate traffic. For instance, one of the

top web attacks, XSS may target web pages in an attempt to add scripts to the website (OWASP, 2017). However, this activity requires more data that significantly exceed the length of the average parameter.

With regard to the SQL injection attack type, the attacker's input may include malicious code that can misdirect the program execution. The code is in special strings which make it possible to alter the SQL statement with the intention of compromising the intended database files. Consequently, the malicious packets may contain up to several thousand bytes. Thus, the feature size of anomalous source traffic is statistically measured to identify anomalous traffics.

To measure the difference between queried and normal traffic, the mean and variance are calculated for CIT measurement. The mean $(\bar{x})$ known as the average in simple arithmetic is calculated by summing up the total bytes of a particular source address and dividing by the total number of traffic for that specific feature. Let *n* be the total number of traffic for a particular source while sum xi is the total size of feature that is derived from the same source. Therefore, the equation to measure the mean $\bar{x}$ is as follows:

$$\overline{x} = \frac{\sum xi}{n} \tag{4.6}$$

In this case, the standard deviation (SD) is used to quantify the variation of a set of data values from the mean. The bigger the variation for each feature, the greater the deviation value from the mean. Thus, the formula to calculate SD is shown as follows:

$$SD = \sqrt{\frac{\sum(xi - \overline{x})^2}{n-1}} \tag{4.7}$$

In this study, CIT is applied to find the right boundary as well as to determine the finest threshold to achieve higher detection rate. The theorem defines the right upper bound (threshold) for a random distribution of a particular source IP address, which has deviated from its average in attack-free data.

The mean and variance of normal activity are used to determine the regularity in the testing data. Then, the probability of traffic to become irregular is measured based on the following equation:

$$P(|x-\mu|) \geq \tau \leq \frac{\sigma^2}{\tau^2} \tag{4.8}$$

The advantage of using CIT is that it does not rely on the knowledge of how the data are distributed, since in a real environment the traffic distribution varies. The upper bound is placed based on the possibility that the deviation between the value of the random variables x and μ is greater than the threshold τ for a random distribution with variance σ^2 and mean μ. The threshold τ is substituted with the difference between feature size *S* and the mean μ of the feature size distribution. The probability of upper bound is defined when a particular source IP address feature size tends to deviate more than the mean in comparison to the normal traffic. The probability value *P*(*S*) for feature size *S* is calculated as below:

$$P(|x-\mu|) \geq |S-\mu| \leq p(S) = \frac{\sigma^2}{(S-\mu)^2} \tag{4.9}$$

Several additional features, such as normality_score, predicted_field, normal_fields, anomaly_field, and distance_value are further utilised in the ensemble classification algorithm to improve the discriminative model of the proposed algorithms. In the next stage, the proposed ensemble classification techniques (second stage detection) are further discussed.

4.4.2 Ensemble Classification Algorithm (Second stage detection)

The high dependency of the first stage detection on normal traffic tends to limit the ability of the system to classify traffic behaviour at a satisfactory level. Alternatively, the second stage detection is proposed to complement the first stage detection using an ensemble classification technique. The method uses supervised approaches that contain samples of both normal and abnormal web traffic in generating the classification model. Particularly, LogitBoost is employed with a combination of Random Forest (RF) in this study.

In the LogitBoost algorithm, a training data set with N samples is divided into two classes (abnormal and normal). They are defined as y ∈ {-1,+1}, e.g. samples in class y=+1 are normal traffic while y=-1 are the samples of attack traffic. Let the set of training data be {*(x1, y1),….., (xi,yi),… (xn,yn)*}, where *xi* is the feature vector, and *yn* is the target class. Thus, the ensemble classification algorithm is implemented as follows:

1) Input data set N= {(x1, y1),….., (xi, yi),… (xn, yn)}, where xi∈ X and yi ∈ Y = {-1,+1}. Input number of iterations K.
2) Initialize the weights wi= 1/N, i= 1, 2, …, N; start committee function *F(x)*=0 and probabilities estimates P(*xi*)=1/2.
3) Repeat for *k*= 1,2,…, K:

a. Calculate the weights and working response

$$wi = p(xi)(1 - p(xi)) \tag{4.10}$$

$$zi = \frac{yi - p(xi)}{p(xi)(1 - p(xi))} \tag{4.11}$$

b. Fit the function $fk(x)$ by a weighted least squares regression of *zi* to *x*i using weights *w*i. In the case of this study, RF is employed as a weak classifier to fit the data using weights *w*i.

c. Update

$$F(x) \leftarrow F(x) + \frac{1}{2} fk(x) \tag{4.12}$$

and

$$p(x) \leftarrow \frac{e^{F(x)}}{e^{F(x)} + e^{-F(x)}} \tag{4.13}$$

5) Output the classifier:

$$sign\,[F(x)] = sign\left[\sum_{k=1}^{K} fk(x)\right] \tag{4.14}$$

At this point, sign [*F* (*x*)] is a function that has two possible output classes:

$$sign\,[F(x)] = \begin{cases} +1, & if\ F(x) < 0 \\ -1, & if\ F(x) \geq 0 \end{cases} \tag{4.15}$$

One of the key factors that influence the performance of the boosting algorithm is the construction of the weak classifier. The weak classifier $fk(x)$ chosen in eq. (4.12) should be resistant to data over-fitting and be able to manage data reweighting. Based on the preliminary experiment conducted in Appendix [A.2], RF algorithm is chosen as the weak classifier for LogitBoost classification. Therefore, in this research, the base algorithm for LogitBoost is RF.

4.5 Post-Processing Phase

The two major stages involved in the post-processing phase are attack signature generation and attack prioritisation assignment. Signature generation is mainly employed to reduce the processing time whereby previously detected unknown attacks are stored in the signature library for the future detection process. The prioritisation assignment is designed to assist the security analyst to further process the attack for better incident handling. The attacks are categorised into a four-severity level (highest, high, low and lowest). Both stages are crucial in reducing detection time and achieving better incident handling. Further details on both stages involved are discussed in the next sub-section.

4.5.1 Attack Signature Formation

The ability of MBDS in detecting a known attack in a short period of time as well as achieving a high detection rate has made many organisations adopt this method. Unlike MBDS, ABDS is a combination of many approaches that require more time to discover the abnormal behaviour pattern, thus further results in excessive computational utilisation. In addition, the detection procedure still needs to undergo the re-execution process that is deemed unnecessary, even though the same pattern or behaviour has been previously detected in ADBS.

The drawback of ABDS that is related to its adoption can be alleviated by leveraging on the strength of MBDS signature formation. Moreover, the new attack must be frequently updated to reduce the computational effort in detecting attack. Thus, the deployment of the proposed MBDS signature formation as part of detection strategy in ABDS deployment will be able to favourably improve the attack detection rate. Meanwhile, it also preserves the ability to detect unknown attack behaviours. In other words, the distinct behaviour of a true attack detected by the ensemble classification technique should be further transformed into attack signature for any future attack detection process.

<table>
<tr><td colspan="7">Binary Formation: 0111110011</td></tr>
<tr><td rowspan="6">Attack Signature</td><td rowspan="3">Attributes Contents</td><td>ethersize</td><td>ethersrchi</td><td>ethersrclo</td><td>iplength</td><td>ipfragid</td></tr>
<tr><td>60</td><td>X00C04F</td><td>XA357DB</td><td>40</td><td>62181</td></tr>
<tr><td>ipsource</td><td>tcpsourceport</td><td>tcpheaderlen</td><td>tcpflag</td><td>tcpwindowsize</td></tr>
<tr><td>172.016.113.084</td><td>9271</td><td>X60</td><td>X10</td><td>32120</td></tr>
<tr><td rowspan="2">Statistical Analysis</td><td>Total Normal Attribute</td><td>Total Anomalous Attribute</td><td colspan="3">Traffic Normality Score</td></tr>
<tr><td>7</td><td>3</td><td colspan="3">72.34%</td></tr>
</table>

Figure 4.4: An example of attack signature stored in the signature library

Figure 4.4 illustrates the example of attack signature that is used for future detection (incoming traffic 1 and 2) process. The signature matching procedure is computed using binary formation and traffic attributes content.

Incoming Traffic 1					
Binary Formation: 0111110011					
Attributes Contents	ethersize	ethersrchi	ethersrclo	iplength	ipfragid
	60	X00C04F	XA357DB	40	62181
	ipsource	tcpsourceport	tcpheaderlen	tcpflag	tcpwindowsize
	172.016.113.084	9271	X60	X10	32120

Figure 4.5: An example of new incoming traffic 1

Figure 4.5 presents incoming traffic 1 that contains both binary formation and attribute contents. The signatures in the signature library are used to match the incoming traffic 1. The traffic will be flagged as attack if the value for both binary formation and traffic attribute value are matched.

Incoming Traffic 2					
Binary Formation: 0111111010					
Attributes Contents	ethersize	ethersrchi	ethersrclo	iplength	ipfragid
	60	X00C04F	XA357DB	40	62181
	ipsource	tcpsourceport	tcpheaderlen	tcpflag	tcpwindowsize
	172.016.113.084	9271	X60	X10	8192

Figure 4.6: An example of new incoming traffic 2

Conversely, if mismatched, meaning the incoming traffic 2 does not tally with the signature, the traffic is considered to be unknown traffic. As a result, the traffic needs to undergo statistical analysis and ensemble classification to further identify the traffic attack type either as attack or normal. Figure 4.6 shows that the content attribute "tcpwindowsize" is not matched with the signature.

The algorithm for the signature matching procedure is as follows:

Start,

If the binary formation and the attribute value are matched or exist

Update the traffic by flagging the traffic as an attack traffic

Else,

Export the traffic into anomaly detection phase for further traffic analysis and classification operation

End

The attack behaviour will be converted to attack signature once the unknown attack traffic is detected from the ensemble classification output. The formulation of the signature for the mentioned behaviours is represented in the following algorithm:

Start,

Select distinct attack traffic and additional features from testing data

Insert the information into attack signature file

If the information exists, the insertion for this row is disregarded and continues with another row

Else

Insert all the information into the attack signature file

End

4.5.2 Attack Prioritisation

In this research, IPM is specifically introduced to aid security analyst to perform better incident handling process. The incidents are measured using this model based on its importance and urgency. The assumption is made to measure the risk level for each attack and map it into different types of group based on the severity level and quadrants. For instance, the critical incident will require a quicker response from the security analyst compared to the less critical incidents. In this research, the severity level generated by IPM employs a combination of the two decision factors of traffic normality score and attack frequencies. The attacks are ranked quantitatively according to the severity level generated.

4.5.2.1 Decision Factors for IPM

In this research, a combination of two decision factors is employed for event prioritisation. Although previous works utilised multiple indicators to index the intrusions, this research is limited to two indicators only. This is due to the limitation of indicators available, considering that the proposed method is solely employing ABDS using a traffic header. According to Anuar *et al.* (2013), the fewer the number of indicators employed, the easier they are to obtain, measure and process.

The four-quadrant level in Figure 4.9 is based on two decision factors namely: normality score and attack frequency. Eq. (4.16) is employed to measure the boundary of normality score, either low or high. Meanwhile eq. (4.17) is performed to determine the attack frequency and its boundary which can be either low or high.

$$Average\ of\ Normality\ Score = \left(\sum \frac{Normality\ Score}{Attack\ Traffic}\right) \tag{4.16}$$

whereby, Lower boundary < Average Normality Score; Higher boundary > Average Normality Score

$$Average\ Attack\ Frequency = \left(\sum \frac{Attack\ Traffic}{Attacks\ Category}\right) \tag{4.17}$$

whereby, Lower boundary < Average Attack Frequency; Higher boundary > Average Attack Frequency

4.5.2.2 Severity Formation

In this research, the time management concept is applied in severity formation. Kirillov *et al.* (2015) demonstrate four different severity levels that represent priorities of actions based on the time management concept. The concept measures the tasks based on two distinct elements: urgent and important. Based on this concept, two distinct types of normality score and attack frequency are chosen to generate the severity level.

The fundamental of this concept, known as the *Eisenhower Matrix* has been widely used by many researchers for easy prioritisation. Such examples include risk management (Haimes, 2001), time management (Gonzalez *et al.,* 2008), email prioritisation (Yoo *et al.,* 2011) and incident prioritisation (Anuar *et al.,* 2013).

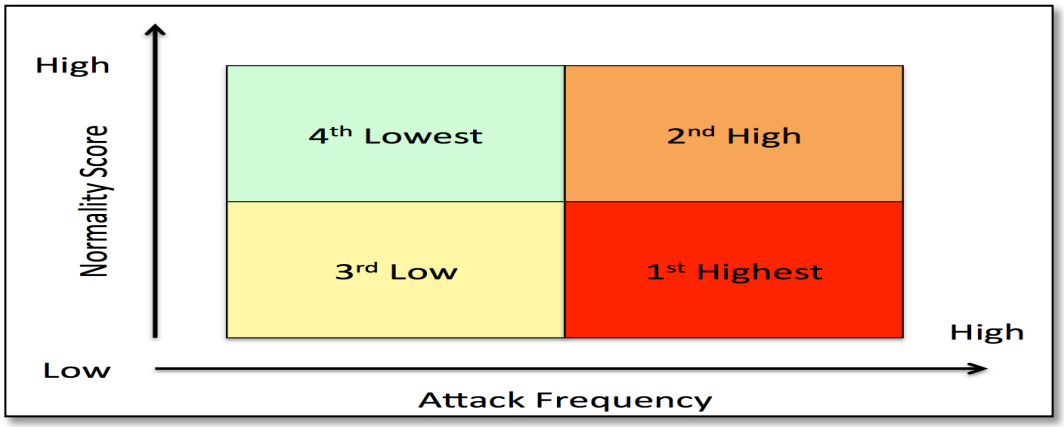


Figure 4.7: Severity Quadrants

Figure 4.7 illustrates four different quadrants that can be categorised into:

1) *1st Quadrant: Low Normality Score and High Attack Frequencies*. This quadrant alerts the highest priority incidents. In this category, the attack traffic represents a critical incident and requires immediate action by the security analyst. For example, the low normality score indicates that the traffic behaviour is more towards attack traffic behaviour.

2) *2nd Quadrant: High Normality Score and High Attack Frequencies.* This quadrant indicates a less urgent situation compared to the 1st quadrant. Although it is less urgent, an adequate action is still required by the security analyst as it is still considered to be a top primacy quadrant. For example, the high attack frequencies indicate more attempts have been launched by an intruder and are therefore still to be considered dangerous.
3) *3rd Quadrant: Low Normality Score and Low Attack Frequencies.* This quadrant is categorised as low level since fewer attack attempts are detected. In view of the traffic normality level also being low, some attention by the security analyst is still needed. This situation can be similar to the 1st quadrant in a way of recording a low normality score. However, for demonstration purposes, the 3rd quadrant is considered as lower priority than the 2nd quadrant due to its recorded lower attack frequency.
4) *4rd Quadrant: High Normality Score and Low Attack Frequencies.* This quadrant indicates the lowest priority and minimum attention required from a security analyst. For example, this category represents attack traffic that is more similar to the 3rd quadrant, and at the same time is less dangerous.

4.6 Summary

In this chapter, the proposed detection scheme has been briefly discussed with the aim to minimising data dimensionality, improving the known and unknown web attack traffic, and simplifying the re-initiation process for future detection. In addition, a more specific detailed implementation of the proposed detection scheme is now discussed in the next Chapter 5.

Chapter 5

The Implementation of I-WEB

5.1 Introduction

This chapter discusses the overall implementation of the proposed work using WEKA tools (Frank *et al.,* 2016) and SQL script. Details of each process will be explained in the following sections. Sections 5.1 highlights the overview of the proposed detection scheme. Section 5.2 presents the steps taken in the pre-processing phase. Section 5.3 performs the steps under the anomaly detection phase whereby the SQL scripts and ensemble learning procedure are performed. In Section 5.4, the attack signature and script for signature detection are formulated and presented, followed by a summary of the chapter in Section 5.5.

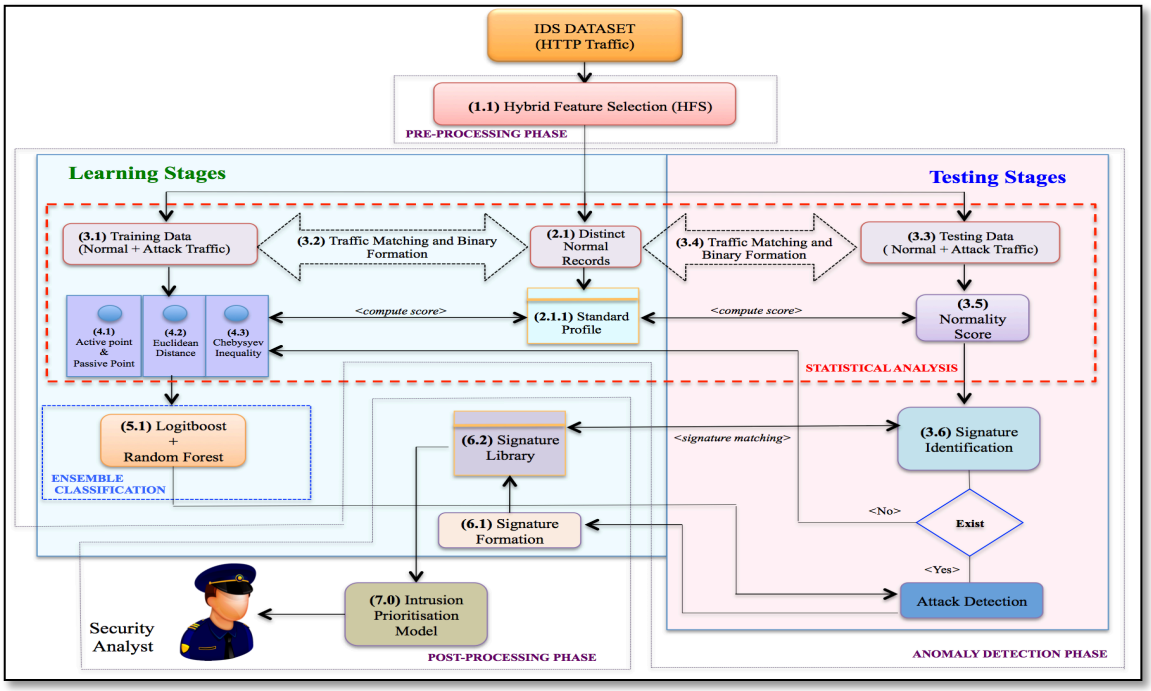


Figure 5.1: The Proposed Detection Model (I-WEB)

Figure 5.1 illustrates the whole procedure required in the proposed detection model. There are three major phases involved in this proposed scheme: pre-processing, anomaly detection and post-processing phases.

In the pre-processing phase (shown as Step 1.1), the intrusion dataset used in the experiment needs to go through the HFS process. The HFS procedure is comprised of two selection approaches: filter and wrapper. The purpose of employing this procedure is to reduce the data dimensionality by removing irrelevant and redundant features.

Meanwhile, the anomaly detection phase consists of two stages of detection methods. In the first-stage detection, the distinct normal (attack-free) records in (Step 2.1) are extracted to compute a normal profile that will be used as a benchmark for identifying novel (unseen) abnormal behaviours. Subsequently, each training and the testing data field value is compared with the feature value of distinct normal records as laid out in (Steps 3.2 and 3.4). If the feature value matches with the value in the normal records, the binary value of '1' is assigned, otherwise a value '0' will be awarded. The entire set of binary values for each traffic is computed to form binary formation. For example, "1111111000" represents 10 features of a single traffic. The processes are continued with the computation of the normal score, distance measurement and threshold mechanism. In (Step 3.5), the normal scores are computed by shifting the binary formation value '0' and '1' with the scores derived from the normal profile. At this point (Step 4.1), two scores i.e. PS and AS, are produced. The total scores for each feature are summed up to represent the degree of normality traffic. Both scores are transformed into data points that represent coordinates in order to measure the distance between AS and PS. In (Step 4.2), the ED is used to measure the distance between normal and outlier points. Moreover, CIT in (Step 4.3) is deployed to further define the traffic with the threshold measurement. The traffic is flagged as anomalous if it surpasses the threshold value while others will be considered to be normal traffic.

In second-stage detection, (Step 5.1) is performed to improve the detection ability derived from the earlier first-stage detection process. The training and the testing data that consist of normal and attack behaviours are used to train and test the ensemble LB-RF classifiers model. Finally, the classification output that is comprised of true attacks behaviour is exported to the attack detection file for signature formation (Step 6.1).

In the post-processing phase, the signature generated will be kept and stored in the signature library (Step 6.2). The usage of attack signature as part of the detection strategy has reduced the re-computation process for incoming traffic. For example, (Step 3.6) is performed to identify attack traffic that matches with the signature library. For every matched signature found, the attack detection file will be updated for further incident response by the security analyst. However, if the signature is not matched, the remaining (Steps 4.1 to 6.2) need to be performed. Finally, (Step 7.0) is employed to prioritise the attack based on the four-severity levels, which are highest, high, low and lowest. The next section discusses in detail the implementation steps involved in each phase.

5.2 Pre-Processing Phase

Initially, in the pre-processing phase, feature selection is employed to remove all irrelevant and redundant features with the aim of reducing the overall data dimensionality. There are three steps taken in implementing HFS using WEKA DM tools.

Step 1: The training data contains both attack and normal samples with ARFF format exported into the WEKA explorer.

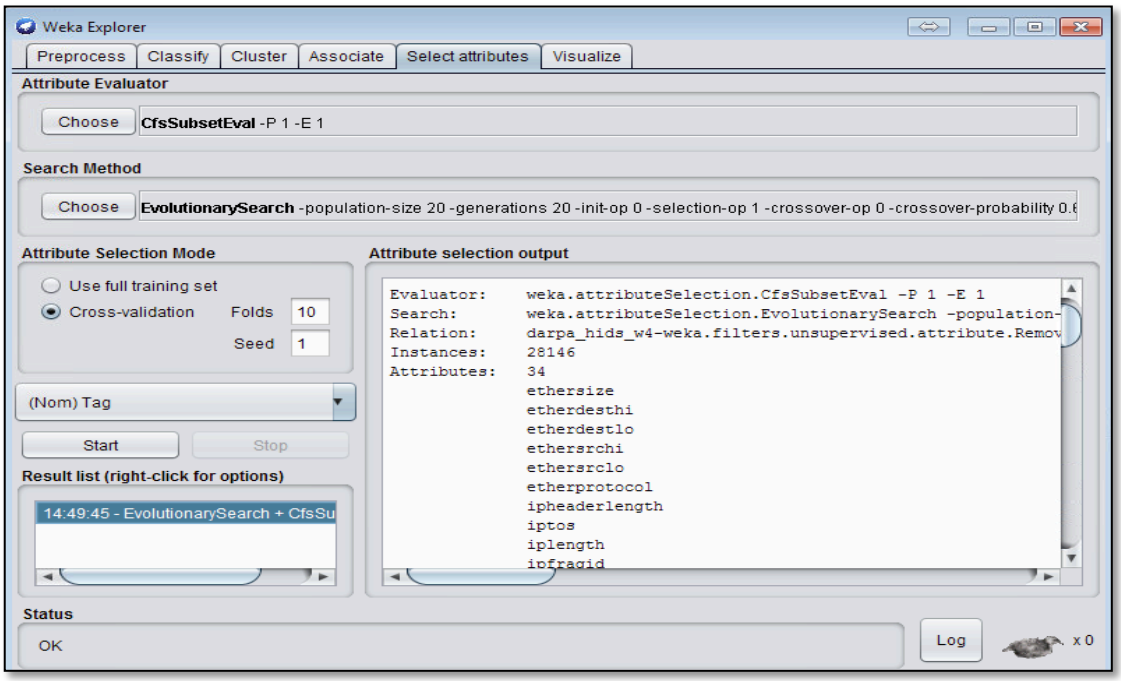


Figure 5.2: Process of selecting features using FBSE

Figure 5.2 illustrates the process taken in FBSE in the DARPA 1999 dataset. The 34 attributes represent 33 original features and an additional one feature for a class label. The experiment is conducted using the 10-fold cross-validation test method.

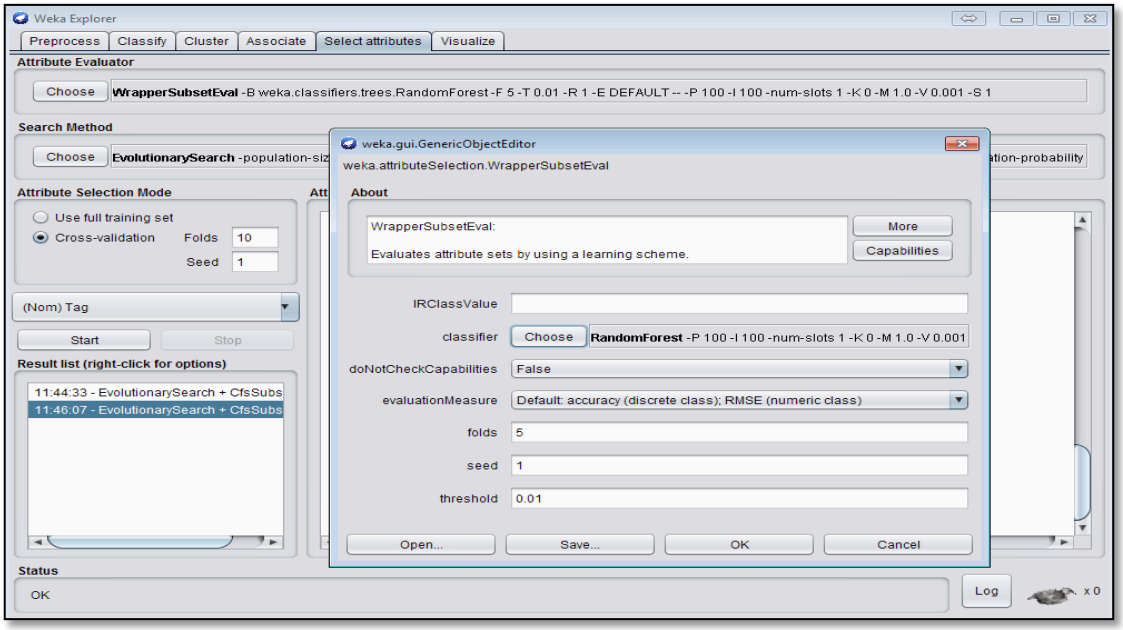


Figure 5.3: Process of selecting features using WBSE

Step 2: The selected features by FBSE are further analysed using WBSE with RF as a classifier. Figure 5.3 presents the process taken by WBSE in the DARPA 1999 dataset. The same testing method using 10-fold cross-validation is executed in WBSE.

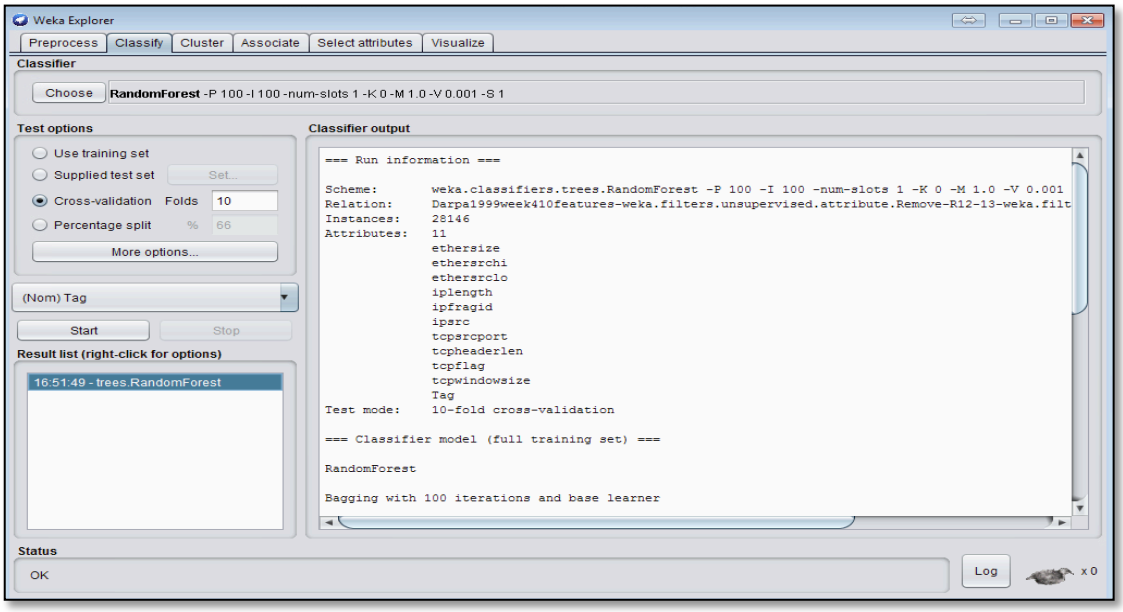


Figure 5.4: Classification method on final selected features

Step 3: The final set of selected features are evaluated with RF 10-fold cross-validation. The remaining features set has indicated its importance in determining attack and normal traffic in the dataset.

Figure 5.4 demonstrates the evaluation of the final set of features. In this phase, the irrelevant and redundant features that could increase the computational effort are efficiently removed. Those selected features will be cross-validated as a benchmark in the subsequent anomaly detection process. The feature selection procedures are conducted using the training data, which contains normal and attack traffics. The significant feature in the filter is measured using a correlation function. Meanwhile, in wrapper, a classification algorithm is used. The features with high merit and correlated to the class are selected. In the event that the selected feature is found to be redundant, it will then be removed as indicated in the Steps 2 and 3. Further analysis of the features selected by the proposed method is discussed in the next Chapter 6: Results and Discussion.

5.3 Anomaly Detection Phase

Step 1: A normal profile is generated from attack free data using SQL script and act as a benchmark of standard behaviour normal http traffic. Following is the example of an SQL query executed to generate a normal profile.

I. Calculate *Rk*

UPDATE `normal_profile` SET R= (SELECT COUNT (DISTINCT `ethersize`) FROM `week3_DARPA1999`) WHERE field =`ethersize`;

II. Calculate *Nk*

UPDATE `normal_profile` SET N= (SELECT COUNT (`ethersize`) FROM `week3_DARPA1999`) WHERE `field` =`ethersize`;

III. Calculate normal score for each feature

UPDATE `normal_profile` SET `traffic_score`= (SELECT log10(R)-log10(N));

Step 2: Traffic matching is the process of matching features content between attack free data in the normal profile with traffic in the testing data. The binary form is used to ease score allocation and to illustrate the sum of normal and anomalous fields. The SQL scripts related to binary form application are presented in the following manner:

I. Traffic matching procedure

The distinct feature value in the attack-free data is compared with the corresponding feature value in the testing data. The scores derived from the normal profile are assigned to the test dataset. All values within the test dataset are closely examined. If their unique values are matched with the profile, a binary value '1' will be awarded. However, if the test dataset values are absent in the normal profile, a value of '0' is assigned to the particular attributes as follows:

UPDATE `testdata_DARPA1999` SET `ethersize_label`=0;

UPDATE `testdata_DARPA1999` SET `ethersize_label`=1 WHERE `ethersize` IN (SELECT DISTINCT `ethersize` FROM `week3_DARPA1999`);

II. Binary formation creation procedure

A binary value of '0' or '1' for each feature is consolidated as a binary sequence (e.g. 1111100101 to represent a series of binary over 10 features of the traffic). The SQL scripts used for generating the binary sequence are as follows:

UPDATE `testdata_DARPA1999` SET `binary_form`= CONCAT (`ethersize_label`, `ethersrchi_label`, `ethersrclo_label`, `iplength_label`,…..); -- Binary sequence for testdata_DARPA1999

III. Process of calculating normal and anomalous columns

From the binary sequences, the amounts of normal and anomalous columns in the traffic are further computed. Both amounts of normal and anomalous columns are used as additional features in the ensemble learning. The queries to compute both columns are as follows:

- Normal Column

UPDATE `testdata_DARPA1999` SET `normal_column` = (SELECT (SUM(IF(`ethersize_label`=0,0,1)))+(SUM(IF(`ethersrchi_label`=0,0,1)))…..+ (SUM(IF(`tcpwindowsize_label`=0,0,1)));

- Anomalous Column

UPDATE `testdata_DARPA1999` SET `anomalous_column`= (10-`normal_column`);

IV. Scores Designation

The binary sequence generated from the matching process in the testing data, is used to ease score allocation. The passive score (PS) is generated to represent a fixed normal score for each column as in the normal profile. The query is as follows:

UPDATE `testdata_DARPA1999` SET `ethersize_passive_score` = (SELECT `feature_normal_score` FROM `normal_profile` WHERE `feature_field` LIKE 'ethersize');

The active score (AS) is generated based on binary designation series in the testing data. For instance, the binary value of '1' is awarded with normal scores while field value '0' will remain:

UPDATE `testdata_DARPA1999` SET `ethersize_active_score` = (SELECT `feature_normal_score` FROM `normal_profile` WHERE `feature_field` LIKE 'ethersize') WHERE `ethersize_label`>0;

UPDATE `testdata_DARPA1999` SET `ethersize_active_score` =0 WHERE `ethersize_label`=0;

Next, the active score in the testing data is computed. The sums of the active scores represent the degree of normality for each traffic in the testing data.

UPDATE `testdata_DARPA1999` SET `traffic_normal_score` = (SELECT SUM (`ethersize_active_score`+`ethersrchi_active_score`+`ethersrclo_active_score`+`iplength_active_score`+…..));

Step 3: To identify outliers in the testing data, the distance between active score and passive score is measured using ED. The SQL scripts to calculate the distance are as follows:

UPDATE `testdata_DARPA1999` SET `distance_value`= SQRT(POWER(`ethersize_passive_score`- `ethersize_active_score`,2)+POWER(`ethersrchi_passive_score`- `ethersrchi_active_score`,2)+….);

UPDATE `testdata_DARPA1999` SET `predicted`= (`Normal`) WHERE `distance_value`=0;

UPDATE `testdata_DARPA1999` SET `predicted`= (`Anomalous`) WHERE `distance_value`>0;

UPDATE `testdata_DARPA1999` SET 'source_anomalous` = (`Anomalous') WHERE `ipsrc` IN (SELECT `ipsrc` FROM `testdata_DARPA1999` WHERE `tag` IN ('Anomalous'));

In conjunction with ED, CIT is employed to find the right boundary and determine the finest threshold to achieve a higher detection rate. The SQL query to measure the upper bound and the threshold is as follows:

UPDATE `week3_DARPA1999` SET `mean_tcpwindowsize`= AVG(`tcpwindowsize`);

UPDATE `week3_DARPA1999` SET `variance_tcpwindowsize`= (POWER('mean_tcpwindowsize`-`tcpwindowsize`)) WHERE `source_anomalous` IN (`Anomalous');

UPDATE `week3_DARPA1999` SET `standard_deviation_tcpwindowsize`= SQRT (`variance_tcpwindowsize`);

UPDATE `week3_DARPA1999` SET `upper_bound`= (`mean_tcpwindowsize`+`standard_deviation`);

UPDATE `testdata_DARPA1999` SET `predicted`=('Normal') WHERE `tcpwindowsize`<(SELECT `upper_bound` FROM `week3_DARPA1999`) and `source` in ('source_anomalous');

Step 4: The ensemble classification is performed whereby LogitBoost + RF are employed to classify the traffic of either normal or attack. Figure 5.5 presents the 10 features that are selected by HFS. The five additional features generated during statistical analysis: *"predicted", "normality_score", "distance_value", "normal_columns"* and *"anomalous_columns"* are also induced to the WEKA DM tool.

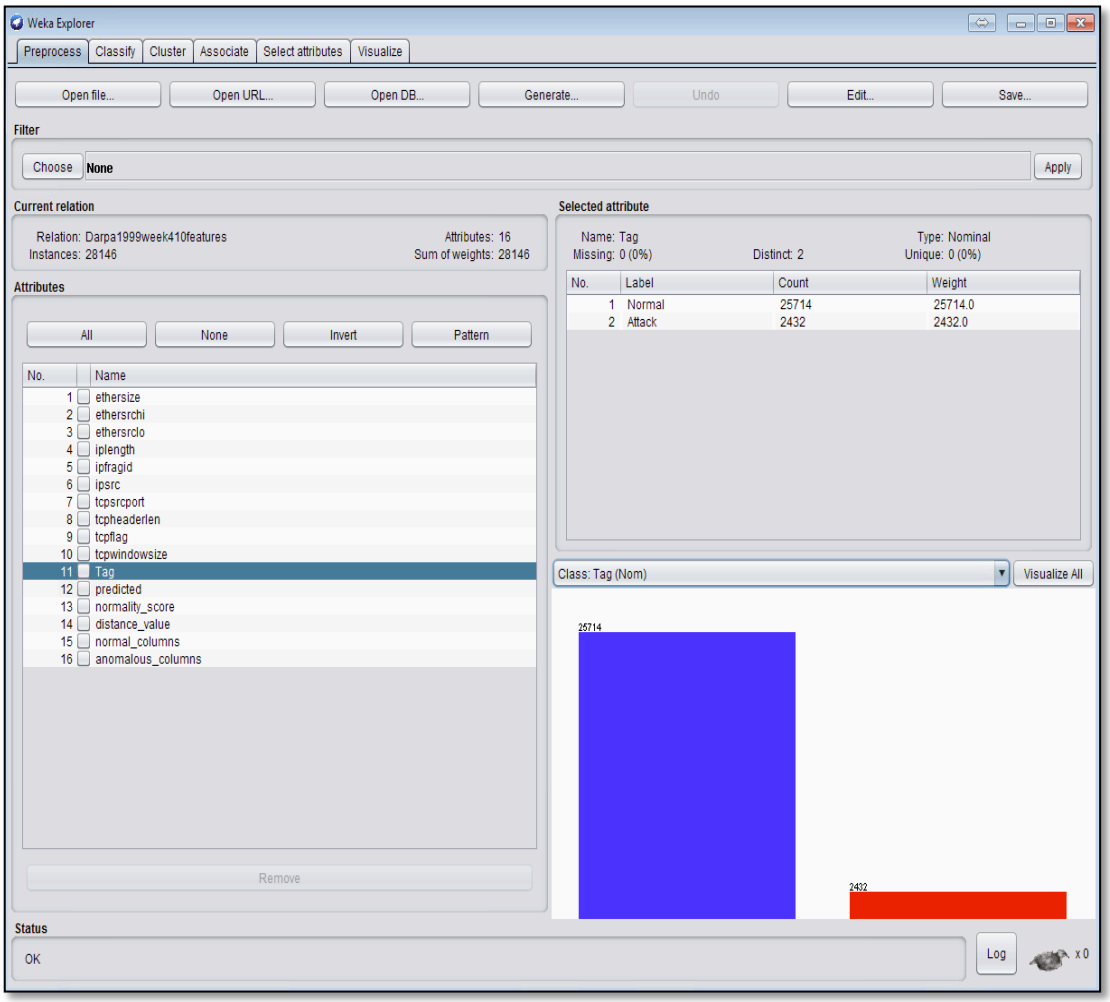


Figure 5.5: The week 4 data of DARPA 1999 dataset

Step 5: As depicted in Figure 5.6, the classification model is built and executed by training the week 4 data using 10-fold cross-validation. Thereafter, the week 5 dataset that consists of 108,816 instances is selected as the supplied test set in this experiment. The experiment is executed and the outputs are presented in the confusion matrix.

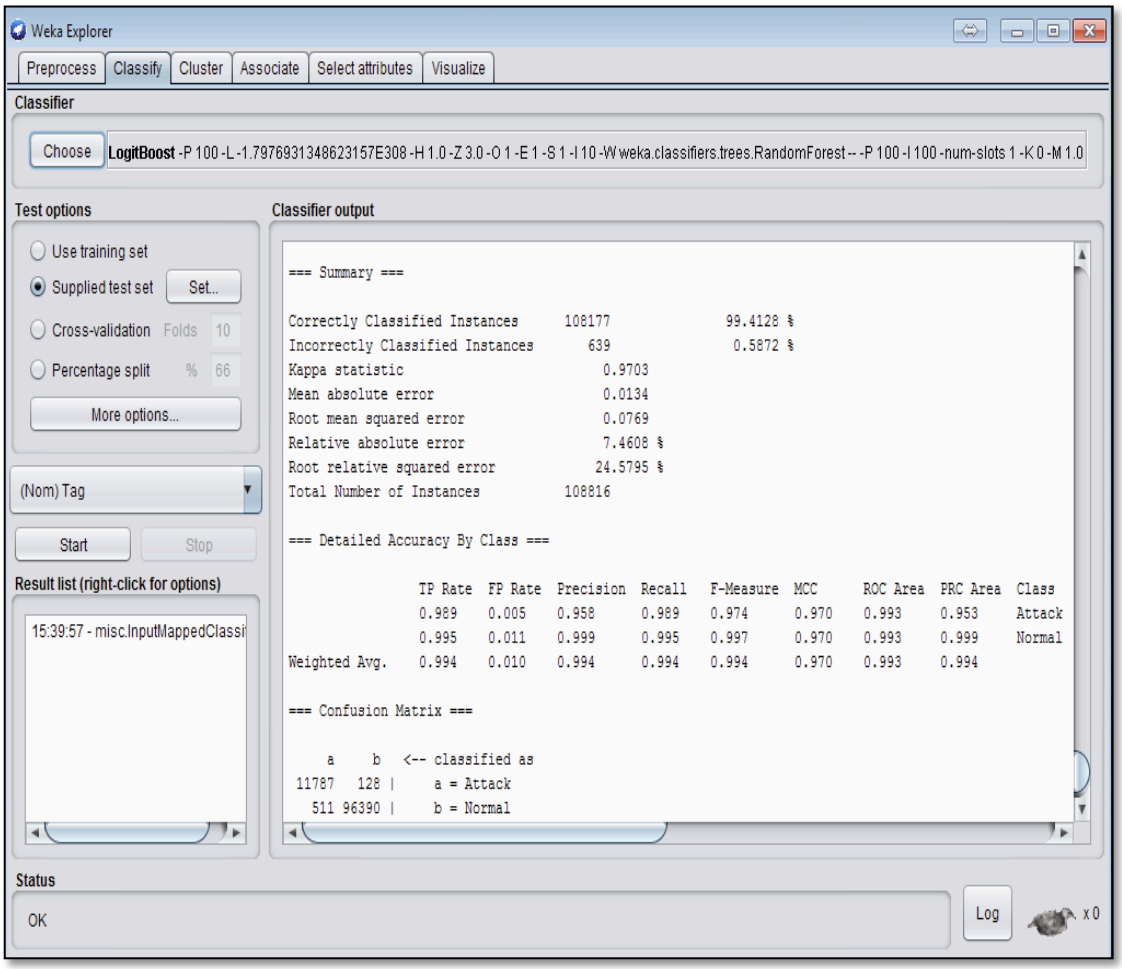


Figure 5.6: The evaluation process of DARPA 1999 dataset

5.4 Post-Processing Phase

There are three steps required in this phase as follows:

I. Extraction of detected attack traffic behaviour from the testing data
II. Importing of selected signature behaviour into the signature library

Steps I and II are performed using the following SQL query,

UPDATE `output_test_week5_with_signature` SET `signature`= CONCAT (`ethersize`,`ethersrchi`,`ethersrclo`,`iplength`,…..) WHERE `predicted` IN ('Attack'); -- signature formation for detected true attack by LB-RF

UPDATE `signature_library` SET `signature` = `signature.output_test_week5_with_signature`;

To eliminate redundant attack signature in the future, the signature library is frequently updated using the following query.

SELECT DISTINCT `signature` FROM `signature_library`;

Step III is performed to match the current signature with the incoming traffic.

III. The following query is used to match any incoming future traffic with signature in the signature library:

UPDATE `new_test_data` SET `signature`= CONCAT (`ethersize`,`ethersrchi`,`ethersrclo`,`iplength`,…..); -- signature formation for new data

UPDATE `new_test_data` SET `signature_label`= 1 WHERE `signature` in (SELECT DISTINCT `signature` FROM `signature_library`);

The incoming traffic that matched the signature in the signature library is further analysed by the security analyst, while the remaining unmatched traffic is further processed in the anomaly detection phase.

*SELECT * FROM `new_test_data` WHERE `signature` IN ('1'); -- 1='Attack'*

The following scripts are used to determine the boundary of normality score. The boundary is computed based on the eq. (4.16) under Section 4.5.2.1.

UPDATE `darpa_1999_severity` SET score_lohi='low' WHERE normality_score<('average_normality_score');

UPDATE `darpa_1999_severity` SET score_lohi='high' WHERE normality_score> ('average_normality_score');

Meanwhile, the following scripts are to determine the boundary of the attack frequency. The boundary is computed based on the eq. (4.17) under Section 4.5.2.1.

UPDATE `darpa_1999_severity` SET freq_lohi=low' WHERE attack_frequency< ('average_attack_frequency);

UPDATE `darpa_1999_severity` SET freq_lohi='high' WHERE attack_frequency> ('average_attack_frequency);

The severity is computed according to (highest, high, low, lowest) using the following SQL scripts:

UPDATE `darpa_1999_intrusion_prioritisation` SET severity_level='highest' WHERE score_lohi='low' AND freq_lohi='high';

UPDATE `darpa_1999_intrusion_prioritisation` SET severity_level='high' WHERE score_lohi='high' AND freq_lohi='high';

UPDATE `darpa_1999_intrusion_prioritisation` SET severity_level='low' WHERE score_lohi='low' AND freq_lohi='low';

UPDATE `darpa_1999_intrusion_prioritisation` SET severity_level='lowest' WHERE score_lohi='high' AND freq_lohi=low';

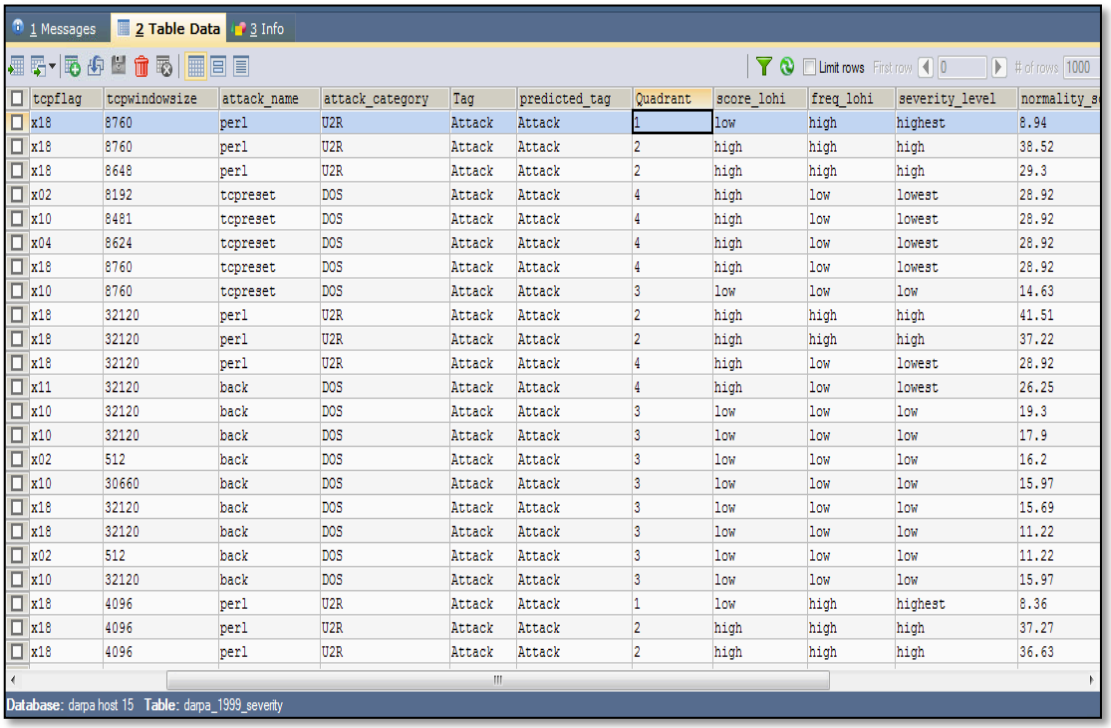
The quadrant level is determined by executing the following script.

UPDATE `darpa_1999_intrusion_prioritisation` SET Quadrant=1 WHERE severity_level IN ('highest');

UPDATE `darpa_1999_intrusion_prioritisation` SET Quadrant=2 WHERE severity_level IN ('high');

UPDATE `darpa_1999_intrusion_prioritisation` SET Quadrant=3 WHERE severity_level IN ('low');

UPDATE `darpa_1999_intrusion_prioritisation` SET Quadrant=4 WHERE severity_level IN ('lowest');



1 Messages | 2 Table Data | 3 Info

Limit rows | First row: 0 | # of rows: 1000

tcpflag	tcpwindowsize	attack_name	attack_category	Tag	predicted_tag	Quadrant	score_lohi	freq_lohi	severity_level	normality_s
x18	8760	perl	U2R	Attack	Attack	1	low	high	highest	8.94
x18	8760	perl	U2R	Attack	Attack	2	high	high	high	38.52
x18	8648	perl	U2R	Attack	Attack	2	high	high	high	29.3
x02	8192	tcpreset	DOS	Attack	Attack	4	high	low	lowest	28.92
x10	8481	tcpreset	DOS	Attack	Attack	4	high	low	lowest	28.92
x04	8624	tcpreset	DOS	Attack	Attack	4	high	low	lowest	28.92
x18	8760	tcpreset	DOS	Attack	Attack	4	high	low	lowest	28.92
x10	8760	tcpreset	DOS	Attack	Attack	3	low	low	low	14.63
x18	32120	perl	U2R	Attack	Attack	2	high	high	high	41.51
x18	32120	perl	U2R	Attack	Attack	2	high	high	high	37.22
x18	32120	perl	U2R	Attack	Attack	4	high	low	lowest	28.92
x11	32120	back	DOS	Attack	Attack	4	high	low	lowest	26.25
x10	32120	back	DOS	Attack	Attack	3	low	low	low	19.3
x10	32120	back	DOS	Attack	Attack	3	low	low	low	17.9
x02	512	back	DOS	Attack	Attack	3	low	low	low	16.2
x10	30660	back	DOS	Attack	Attack	3	low	low	low	15.97
x18	32120	back	DOS	Attack	Attack	3	low	low	low	15.69
x18	32120	back	DOS	Attack	Attack	3	low	low	low	11.22
x02	512	back	DOS	Attack	Attack	3	low	low	low	11.22
x10	32120	back	DOS	Attack	Attack	3	low	low	low	15.97
x18	4096	perl	U2R	Attack	Attack	1	low	high	highest	8.36
x18	4096	perl	U2R	Attack	Attack	2	high	high	high	37.27
x18	4096	perl	U2R	Attack	Attack	2	high	high	high	36.63

Database: darpa host 15 Table: darpa_1999_severity

Figure 5.7: The intrusion prioritisation process of the DARPA 1999 Dataset

Figure 5.7 presents the prioritisation table in MySQL for attack detected by the proposed system. The table are illustrated according to quadrant (1,2,3,4) with aim to assist the security analyst for further incidents response.

5.5 Summary

This chapter described the implementation procedure of the proposed detection scheme. Several formulas, algorithms and SQL scripts are employed in this chapter. The effectiveness of the proposed approach is evaluated with a series of experiments. The results derived from the experiments are now presented and discussed in Chapter 6.

Chapter 6

Results and Discussion

6.1 Introduction

This chapter presents the results and analysis discussion of the proposed detection scheme (I-WEB). Section 6.2 presents several preliminary experiments that were specifically conducted during the pre-processing and anomaly detection phases. Section 6.3 details the experimental results of the proposed detection scheme that consists of the pre-processing, anomaly detection and post-processing phases. Section 6.4 presents the discussion of attack analysis with the comparison of previous work using DARPA 1999, NSL KDD, ISCX 2012 and UNSW-NB15 datasets respectively. Finally, Section 6.5 summarises the whole chapter.

6.2 Preliminary Experiments

There are two preliminary experiments conducted in this study. The first experiment is to seek a suitable search algorithm for selecting the most prominent features during the pre-processing phase. The second preliminary experiment is performed in the anomaly detection phase, in which a suitable base classifier is selected for combination with the LogitBoost algorithm.

6.2.1 First Preliminary Experiment

In the early pre-processing phase, FBSE is employed using CFS. The approach is tested with four different search algorithms: best-first, greedy-stepwise, genetic-search and PSO. Subsequently, the performances of all search techniques are compared and the best performer is chosen to fuse with the wrapper in the subsequent stage. Thereafter, the WBSE is employed for further features optimisation procedure.

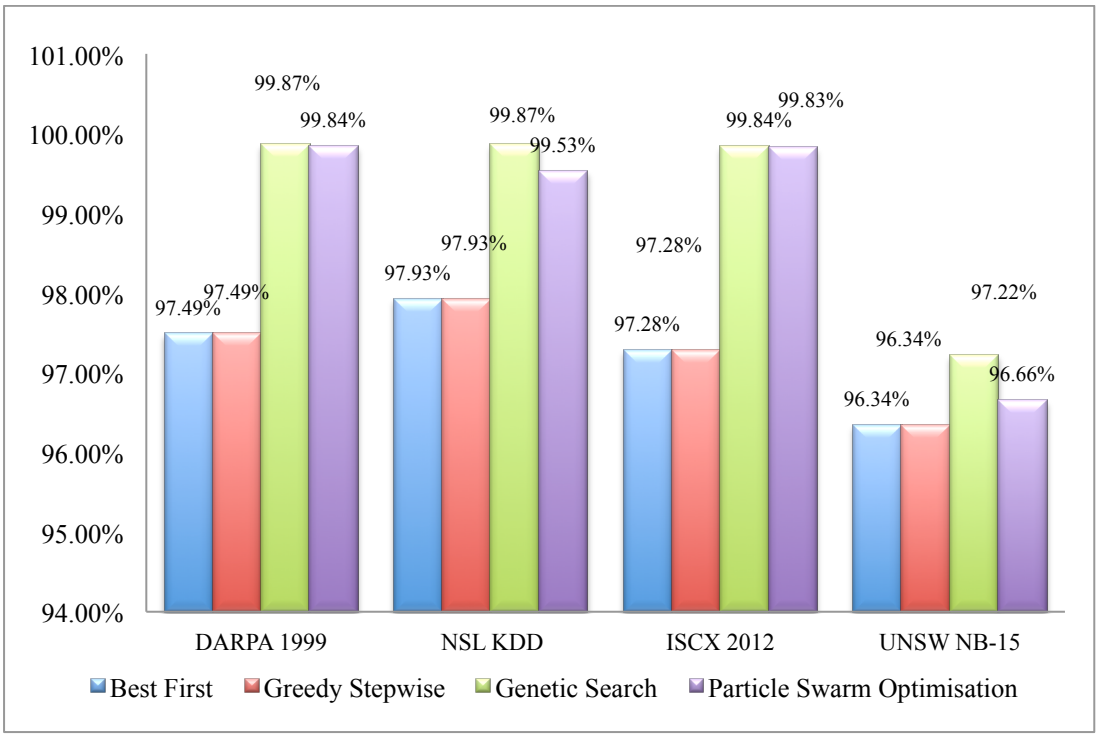


Figure 6.1: Comparison of Filter Approaches Performance over IDS Datasets

Figure 6.1 shows the accuracy rate performance of the DARPA 1999, NSL KDD, ISCX 2012 and UNSW-NB15 datasets. It can be seen that both best-first and greedy stepwise search algorithms have recorded similar performance for all datasets. This is because of both algorithms are sharing a similar searching algorithm technique where the only different is that best-first algorithm could go back to track the previous subset if the search output is found to be less promising. Among the four search algorithms employed, genetic search had indicated the highest accuracy over different datasets. With a view to good detection system performance, ideally the lowest false detection with highest detection accuracy is preferred. Considering those factors, the genetic search algorithm seems to be the most suitable to be adopted as the search algorithm in the pre-processing phase.

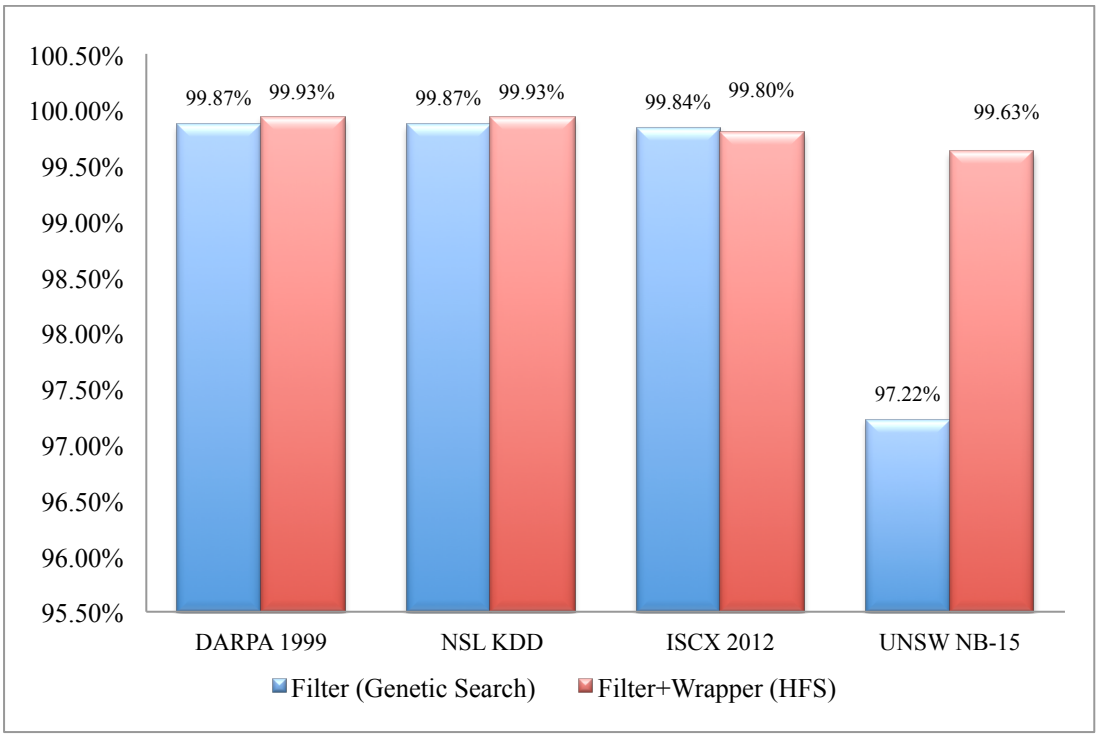


Figure 6.2: Comparison of Performance Accuracy between Filter and HFS

In the previous stage, the genetic-search algorithm has shown comparable performance compared to other search methods. Moving on, the selected features are forwarded to WBSE for a further feature optimisation process. In WBSE, the features merit is measured using the RF classifier. The combination of Filter and Wrapper is known as HFS. Figure 6.2 shows that the new hybrid HFS had recorded significant improvement performance over the Filter. The improvement is contributed by the convincing performance of filter and wrapper in eliminating redundant and irrelevant features efficiently. The results had justified the need for combination approaches between filter and wrapper methods for selecting prominent features.

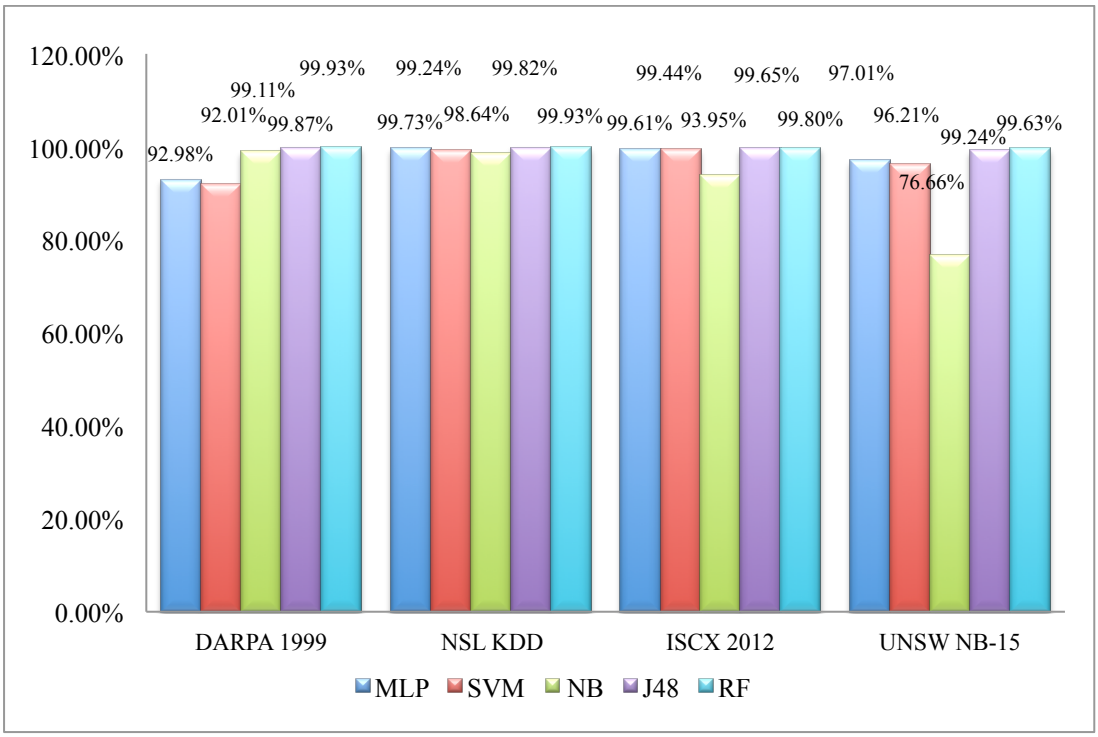


Figure 6.3: Comparison of Classification Algorithms Performance on HFS

In the classification stage, the features identified by HFS are further evaluated for classification accuracies with five widely known classifiers, which are MLP, SVM, NB, J48 and RF, using 10-fold cross-validation. Figure 6.3 shows the RF classifier had consistently achieved highest accuracy performance over other classifiers. Thus, it has been chosen as a benchmark to evaluate the HFS performance. The analysis of the selected features are further explained in the Section 6.3.1.

6.2.2 Second Preliminary Experiment

The second preliminary experiment is performed in the anomaly detection phase, where a suitable base classifier is selected to combine with the LogitBoost algorithm. As the LogitBoost requires a base classifier to fuse with, the classifier with the highest detection accuracy and lowest false detection rate within a reasonable processing time is preferred. Based on past initiatives, six well-known classification algorithms MLP, SVM, NB, DT, J48 and RF are compared, evaluated and briefly discussed.

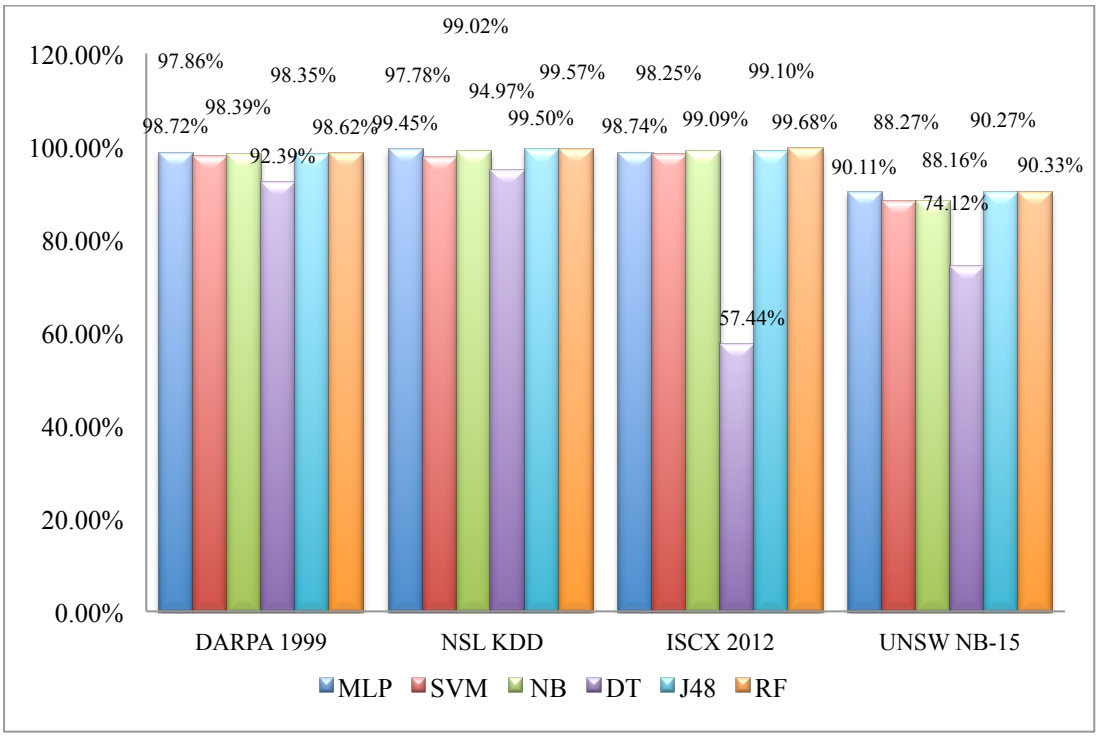


Figure 6.4: Comparison of Classification Algorithms Performance on IDS Datasets

Figure 6.4 shows the accuracy rate performance on DARPA 1999, NSL KDD, ISCX 2012 and UNSW-NB15 datasets. It can be seen that RF had recorded consistent performance over other classification algorithms on four different datasets. Although in DARPA 1999, the performance of MLP is better than RF by 0.1%, the processing time taken by MLP is higher when compared to RF [Appendix A.2]. Thus, considering aforementioned factor, RF has been chosen as the base classifier to ensemble with LogitBoost.

6.3 Performance Evaluation of Proposed Detection Scheme

6.3.1 Pre-Processing Phase

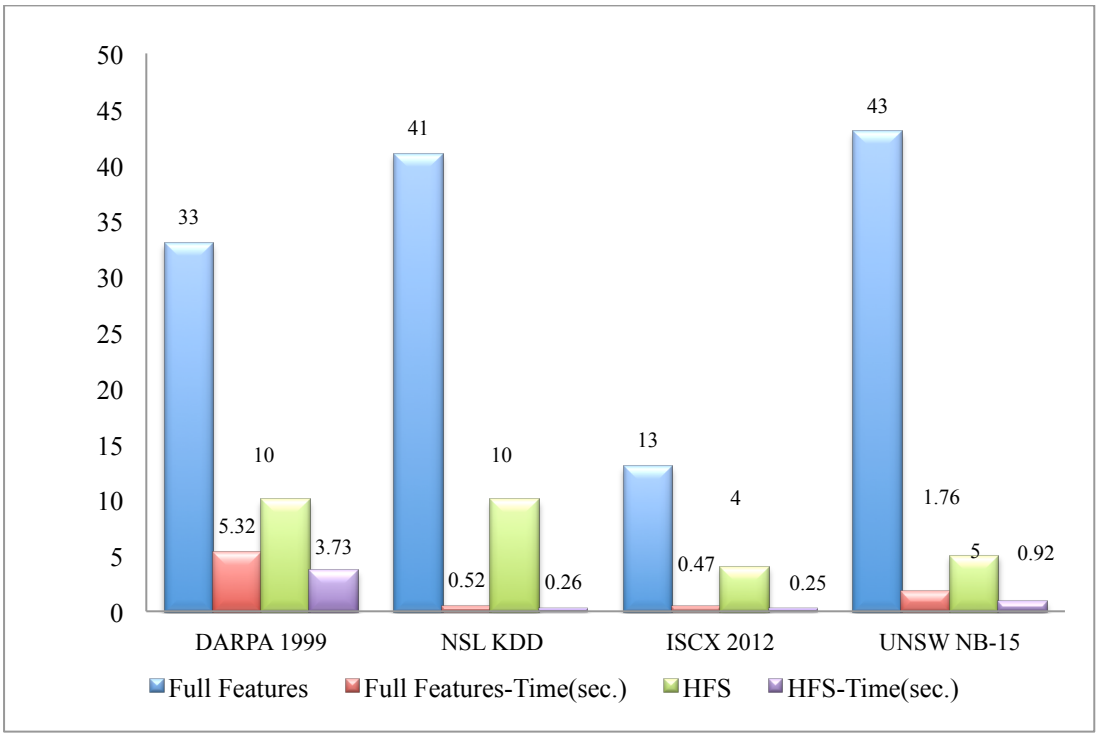


Figure 6.5: Feature Count and Time taken to Built Training Model over IDS Datasets

Figure 6.5 presents the performance result of the proposed HFS approach and original features over DARPA 1999, NSL KDD, ISCX 2012 and UNSW-NB15 datasets. The HFS ability to efficiently identify prominent features has improved the overall attack detection. Thus, it became the main contribution to the feature reduction process. The time taken by HFS is much lower than using full features. This is because the time taken in building the classification model is highly dependent on the feature counts. The fewer feature counts executed the quicker the classification model is built.

In the DARPA 1999 dataset, the proposed HFS shows significant improvement with 69.7% and 30% in feature reduction and time taken in building the classification model respectively. The 10 significant features selected by HFS are *ethersize, ethersrchi, ethersrclo, iplength, ipfragid, ipsrc, srcport, tcpheaderlen, tcpflag, and tcpwindowsize*. Further in-depth study has revealed that the most relevant features are needed to classify traffic behaviour

status (normal or attack). For instance, the U2R attack named '*perl*', usually has *tcpwindowsize* and *iplength* range values above 18,980 and between 40 and 219 respectively. Conversely, normal traffic behaviour contains *tcpwindowsize* and *iplength* range values between 4416 to 8760 and between 40 and 80 respectively. These behaviours have justified the finding that those features are significantly important to differentiate between normal and attack traffic behaviour.

Meanwhile in the NSL KDD dataset, the proposed HFS shows significant improvement with 75.6% and 50% in feature reduction and time taken in building the classification model respectively. Upon closer investigation, the 10 significant features selected by HFS are *src_bytes, count, srv-count, same_srv_rate, srv_diff_host_rate, dst_host_srv_count, dst_host_same_srv_rate, dst_host_diff_srv_rate, dst_host_srv_diff_host_rate* and *dst_host_srv_serror_rate.* Further in-depth study has revealed that the most relevant features are needed to classify traffic behaviour status (normal or attack). For instance, the DoS attacks named '*apache2'* and *'neptune'*, are triggered when there is a huge number of connections establish by the same host. Thus, the features such as *'count', 'srv_count'* are important in identifying those attacks. Similar to *'portsweep'* and *'ipsweep'* attacks, these attacks are performed with the aims of finding the system vulnerabilities. As the nature of the attack itself need to examine the innumerable hosts, *'srv_diff_host_rate'* and *'diff_host_serv_rate'* are important to measure the connection establish by the different hosts. Conversely, normal traffic behaviour usually contains the range values of '*scr_bytes'* between 140 and 340. These behaviours have justified the finding that those features are significantly important to differentiate between normal and attack traffic behaviour.

Furthermore in the ISCX 2012 dataset, the proposed HFS shows significant improvement with 69.2% and 47% in feature reduction and time taken in building the classification model respectively. Upon closer investigation, the four significant features selected by HFS are *totalSourceBytes, source, sourceTCPFlagsDescription and sourcePort.* Further in-depth study has revealed that the most relevant features are needed to classify traffic behaviour status (normal or attack). For instance, attack behaviour instances usually contain *sourcePort* range values between 29190 and 31537. On the other hand, normal traffic behaviour normally contains *totalSourceBytes* range values between 64 and 6,385.

These behaviours have justified the finding that those features are significantly important to differentiate between normal and attack traffic behaviour.

Finally in the UNSW-NB15 dataset, the proposed HFS shows significant improvement with 69.2% and 47% in feature reduction and time taken in building the classification model respectively. Upon closer investigation, the five significant features selected by HFS are *sbytes, tcprtt, synack, dmean and response_body_len.* Further in-depth study has revealed that the most relevant features are needed to classify traffic behaviour status (normal or attack). For instance, the attack named '*DoS'*, usually has low range values of *response_body_len* between 100 and 700 and range values of *synack* are above 0.05 respectively. Conversely, normal traffic behaviour contains *sbytes* range values between 4770 and 10168 and *syncack* range values less than 0.04. These behaviours have justified the finding that those features are significantly important to differentiate between normal and attack traffic behaviour.

In the next phase, the reduced set of features selected by the proposed HFS is further processed using statistical analysis and classification approaches.

6.3.2 Anomaly Detection Phase

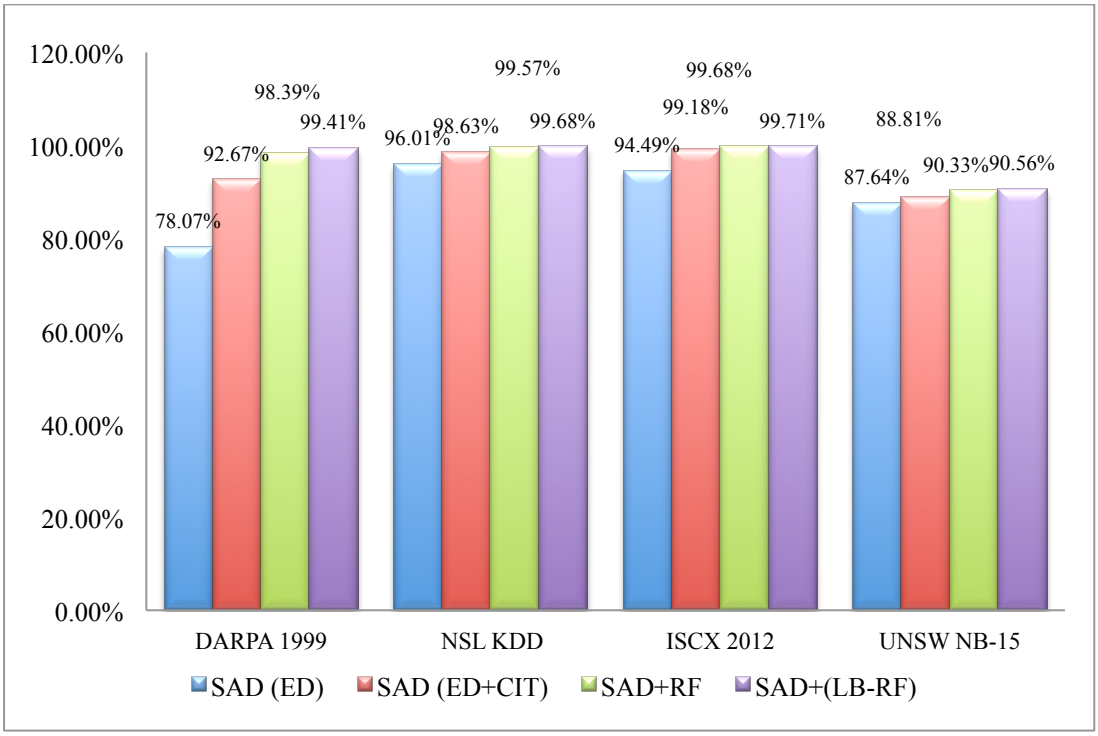


Figure 6.6: The Accuracy Rate Performance over IDS Datasets

Figure 6.6 shows the accuracy rate performance of the DARPA 1999, NSL KDD, ISCX 2012 and UNSW-NB15 datasets. In this study, the SAD (ED) and SAD (ED+CIT) are known as first stage detection while SAD+RF and SAD+(LB-RF) are known as second stage detection. The significant improvement performance of SAD+(LB-RF) over SAD and SAD+RF on four different datasets indicates that the proposed ensemble classification method is suitable for detecting known and unknown web attacks.

In DARPA 1999, it can be seen from Figure 6.6, SAD (ED+ CIT) had significantly improved the detection accuracy of SAD (ED) alone, at 92.67%. The significant improvement of 14.60% accuracy rate had justified the need for using the CIT method to improve the detection performance in first stage detection. To improve the performance recorded in first stage detection, the ensemble classification approach is proposed as second stage detection. The convincing performance recorded by SAD+(LB-RF) over SAD has demonstrated that the second stage detection is capable

of identifying an additional 6.74% of detection accuracy that was first missed out by SAD (ED+CIT) during its first stage of detection process.

Meanwhile in the NSL KDD dataset, it can be seen that the employment of SAD (ED+CIT) has significantly improved the detection accuracy of SAD (ED) alone with 99.68%. To improve the performance recorded in first stage detection, the ensemble classification approach is proposed as second stage detection. The convincing performance recorded by SAD+(LB-RF) over SAD has demonstrated that the second stage detection is capable of identifying an additional 1.05% of detection accuracy that was first missed out by SAD (ED+CIT) during its first stage of detection process.

Furthermore in the ISCX 2012 dataset, the significant improvement of the SAD (ED+CIT) approach over SAD (ED) with 4.69% accuracy rate had highlighted the contribution of the CIT method to improve the performance of first stage detection. To further improve performance in first stage detection, the ensemble classification approach is proposed as the second stage detection. Figure 6.6 shows that the improvement by SAD+(LB-RF) over SAD (ED+CIT) indicated the need for ensemble classification to further identify an additional 0.53% of detection accuracy that was missed out by SAD (ED+CIT) in the first stage detection.

Finally in the UNSW-NB15 dataset, the improvement of 1.17% accuracy rate of SAD (ED+CIT) over SAD (ED) indicated the need for CIT employment to improve the performance of first stage detection. The ensemble classification approach is proposed as second stage detection to improve the detection performance of first stage detection. The performance recorded by SAD+(LB-RF) over SAD has demonstrated that the second stage detection is capable of identifying an additional 1.75% of detection accuracy that was first missed out by SAD (ED+CIT) during its first stage of detection process.

6.3.3 Post-Processing Phase

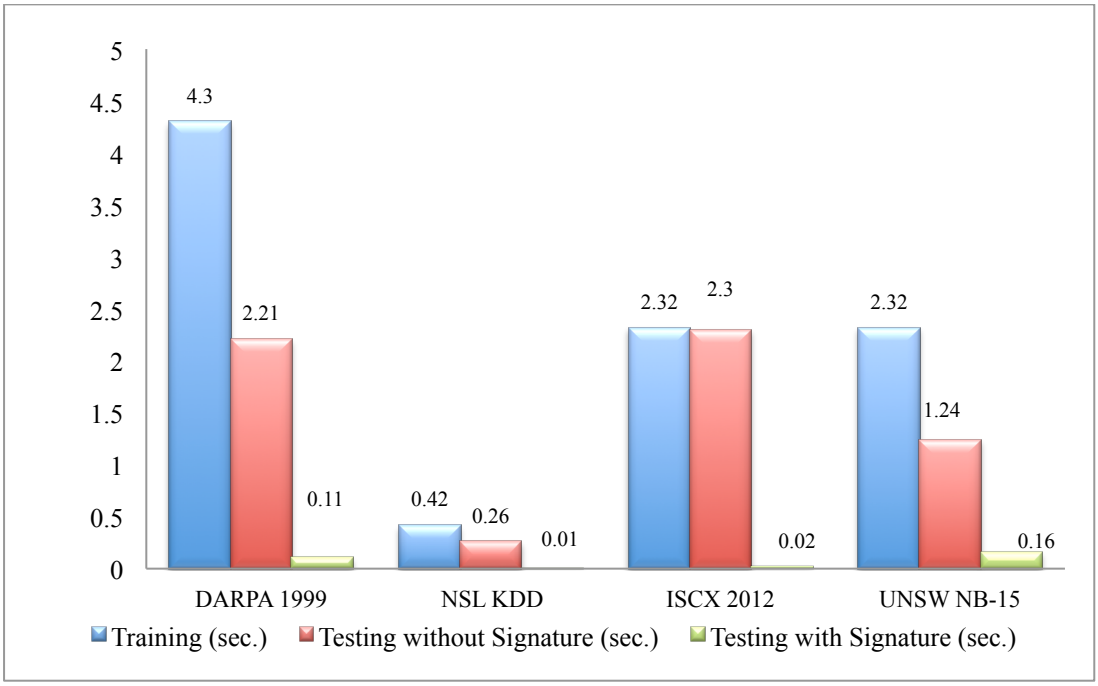


Figure 6.7: The Necessity of Signature Implementation over IDS Datasets

Figure 6.7 shows the effectiveness of the proposed detection scheme by employing the attack signature as part of the detection strategy. Initially, the attacks detected by SAD+(LB-RF) are transformed into signatures and stored in the signature library where only the significant header contents that are first selected by HFS are extracted. In view of the fact that the signatures with straightforward structure are employed, the detection time is significantly reduced.

In the DARPA 1999 dataset, the total numbers of attacks available in the testing data are 12,298 while the attack signatures generated by SAD+(LB-RF) are 11,787. Figure 6.7 shows the detection time recorded by the proposed detection scheme (108,816 instances without attack signature) is 2.21 seconds. However, with the implementation of signature as part of the detection strategy, the detection time has significantly reduced by 95% to 0.11 seconds. This is due to the reduction in attack instances when the system is processing only the remaining 511 attack instances that were not matched with the signatures along with 96,518 normal instances.

Meanwhile in the NSL KDD dataset, the total numbers of attacks available in the testing data are 2,785 while the attack signatures generated by SAD+(LB-RF) are 2,768. Figure 6.7 shows the detection time of the proposed detection scheme (5,641 instances without attack signature) is 0.26 seconds. However, with the implementation of signature as part of the detection strategy, the detection time has significantly reduced by 96% to 0.01 seconds. This is due to the reduction in attack instances when the system is processing only the remaining 17 attack instances that were not matched with the signatures, along with 2,856 normal instances.

Furthermore in the ISCX 2012 dataset, the total numbers of attacks available in the testing data are 28,329 while the attack signatures generated by SAD+(LB-RF) are 28,234. Figure 6.7 shows the detection time recorded by the proposed detection scheme (34,957 instances without attack signature) is 2.3 seconds. However with the implementation of signature as part of the detection strategy, the detection time has significantly reduced by 99.13% to 0.02 seconds. This is due to the reduction in attack instances when the system is processing only the remaining 95 attack instances that were not matched with the signatures, along with 6,628 normal instances.

Finally in the UNSW-NB15 dataset, the total numbers of attacks available in the testing data are 13,376 while the attack signatures generated by SAD+(LB-RF) are 12,046. Figure 6.7 shows the detection time of the proposed detection scheme (18,724 instances without attack signature) is 1.24 seconds. However, with the implementation of the signature as part of detection strategy, the detection time has significantly reduced by 87% to 0.16 seconds. This is due to the reduction in attack instances when the system is processing only the remaining 1330 attack instances that were not matched with the signatures, along with 5,348 normal instances.

In this phase, the reduction in detection time is only tested on the same traffic. However, in a real traffic environment, the incoming attack traffic usually contains similar behaviour to the previously detected attack traffic. Thus, by implementing an attack signature approach in the initial stage, similar attack behaviour could be recognised and filtered out. As a result, the remaining traffics that are not identified by those signatures will turn out to be fewer when compared to the original traffic volume.

6.4 Attack Analysis and Comparison of Previous Work

6.4.1 DARPA 1999 Dataset

Table 6.1: Performance of proposed I-WEB using the DARPA 1999 testing dataset

Attack Category	Attack Name	Attack Traffic in Training Dataset	Attack Traffic in Testing Dataset	Attack Traffic Detected by I-WEB	%age of Detected Attack traffic
DoS	back	25	1,300	**983**	**75.71**
	tcpreset	-	5	**5**	
Probe	ipsweep	106	598	**404**	**67.56**
U2R	perl	1,677	10,333	**10,333**	**100**
R2L	phf	624	-	**-**	**-**
DATA (New)	secret	-	62	**62**	**100**
Total	-	2,432	12,298	**11,787**	**95.84**

Table 6.1 lists six types of attack available in both weeks 4 and 5 from the DARPA 1999 dataset. The four types of attack existed in week 4 (training dataset) are *back, ipsweep, perl and phf*. Subsequently on week 5 (testing dataset), five types of attack: *back, ipsweep, perl* plus two new attacks named *secret and tcpreset* are identified. The proposed approach successfully recognised 95.84% of attack instances in the testing dataset. The attack types with the highest detection rate are U2R (100%) and DATA (100%), followed by DoS (75.71%) and Probe (67.56%).

Upon closer analysis, the poor performance of Probe is due to the nature of the attacks themselves that share similar characteristics with normal traffic behaviour. As the Probe attacks nature is to gather system information and to discover known vulnerabilities, the relevant kind of traffic seems to be legitimate and mostly classified as normal by the system. With regard to the DoS attack type, the low detection percentage of *back* attack is caused by the lack of samples available in the training dataset. The sample is 52 times smaller than the attack in the testing dataset. However, it is worth mentioning that the

proposed approach has successfully identified two new attacks, namely *tcpreset* and *secret* that are only present in the testing dataset.

Table 6.2: Performance comparisons using the DARPA 1999 dataset

Methods	False Alarm Rate (%)	Detection Rate (%)	Accuracy (%)
Improved IDS with Fuzzy Logic by Shanmugam and Idris (2009)	6.10	88.71	N/A
Lightweight IDS by Chen *et al.* (2010)	1.36	72.70	N/A
Ensemble Neural Classifier by Raj Kumar and Selvakumar (2011)	3.70	99.40	N/A
Sequential Differentiate Method by Raja *et al.* (2012)	3.38	100.00	N/A
Hybrid Data Mining by Agarwal and Mittal (2012)	2.75	97.25	N/A
Distribution IDS by Hakimi and Faez (2013)	N/A	96.00	N/A
Catastrophe Theory by Xiong *et al.* (2013)	3.38	87.39	N/A
I-WEB (2018)	**0.13**	**95.84**	**99.41**

Table 6.2 shows the proposed I-WEB performance in terms of FAR, DR and ACC compared with the previous methods tested on the DARPA 1999 dataset. The comparisons are for reference only due to many researchers having used different proportions of traffic types, sampling methods and pre-processing techniques. The

proposed detection approach has demonstrated a comparable performance in terms of detection rate. Although the study by (Raja *et al.,* 2012) achieved 100% detection rate, the false alarm produced is relatively high compared to other approaches. In addition, the significant reduction on the false alarm rate, which had validated the proposed detection approach, is suitable to employ in the field of IDS.

6.4.2 NSL KDD Dataset

Table 6.3: Performance of proposed I-WEB using the NSL KDD testing dataset

Attack Category	Attack Name	Attack Traffic in Training Dataset	Attack Traffic in Testing Dataset	Attack Traffic Detected by I-WEB	%age of Detected Attack Traffic
DoS	back	203	1112	**1112**	**99.75**
	apache2	434	302	**301**	
	neptune	44	1334	**1328**	
Probe	portsweep	1	16	**13**	**74.19**
	ipsweep	-	7	**3**	
	satan	-	7	**6**	
	nmap	-	1	**1**	
	saint	1	-	**-**	
R2L	phf	-	6	**4**	**66.67**
U2R	-	-	-	**-**	**-**
Total	-	682	2,785	**2,768**	**99.39**

Table 6.3 lists nine types of attack available in both training and testing of the NSL KDD dataset. The five types of attack that existed in the training dataset are *back, apache2, neptune, portsweep and saint*. Subsequently in the testing dataset, eight types of attack: *back, apache2, neptune, portsweep* plus four new attacks named *ipsweep, satan, nmap and phf* are identified. The proposed approach has successfully recognised 99.39% of attack instances in the testing dataset. The attack type with the highest detection rate is DoS (99.75%), followed by Probe (74.19%) and the lowest is R2L (66.67%).

Upon closer analysis, the poor performance of R2L is due to the feature value in *"src_bytes"* containing similar values to the features of normal traffic. Thus, the system is keen to recognise R2L attacks as normal traffic. This is justified by the nature of the attack itself being when the users tried to access the server remotely, as the relevant kind of traffic seems to be legitimate and mostly the traffic is classified as normal by the system.

With regard to the Probe attack type, the low detection percentage is caused by the lack of samples available in the training dataset. For instance, the sample of *portsweep* is 16 times smaller than the amount of attack available in the testing dataset. However, it is worth mentioning that the proposed approach successfully identified all four new attacks, namely *ipsweep, satan, nmap* and *phf* that are only present in the testing dataset.

Table 6.4: Performance comparisons obtained on KDD and NSL KDD datasets

Methods	**Feature Selection**	**Features**	**Normal Detection Rate**	**DoS**	**Probe**	**R2L**	**U2R**	**Detection Rate**	**False Alarm Rate**
ACC by Tsang and Kwong (2006)	Yes	N/A	98.8	97.3	87.5	12.6	30.7	N/A	N/A
GP-Transformation Function by Faraoun and Boukelif (2006)	No	41	99.93	98.81	**97.29**	45.2	80.22	N/A	N/A
Hierarchical SOM by Gunes Kayacik *et al.* (2007)	No	41	98.40	96.90	67.60	7.30	15.70	90.6	1.57
MOGFIDS by Tsang *et al.* (2007)	Yes	25	98.36	97.20	88.59	15.78	11.01	92.76	N/A
Multinomial Naïve Bayes by Panda *et al.* (2010)	No	41	N/A	N/A	N/A	N/A	N/A	96.5	3.00
GHSOM-MOF by De La Hoz *et al.* (2014)	Yes	29	N/A	N/A	N/A	N/A	N/A	99.12	2.24
N-KPCA-GA-SVM by Kuang *et al.* (2014)	Yes	N/A	N/A	N/A	N/A	N/A	N/A	95.26	1.03
OS-LEM by Singh *et al.* (2015)	Yes	21	99.07	99.14	90.35	78.10	56.75	97.67	1.74
TVCPSO-SVM by Bamakan *et al.* (2016)	Yes	17	99.13	98.84	89.29	67.84	40.38	97.03	0.87
Ramp-KSVCR by Bamakan *et al.* (2017)	No	41	99.14	99.49	93.58	**91.09**	**68.75**	98.48	0.86
I-WEB (2018)	Yes	**10**	**99.96**	**99.75**	74.19	66.67	N/A	**99.39**	**0.04**

Table 6.4 shows the comparison of proposed I-WEB performances over the previous methods tested on KDD and the NSL KDD datasets (Bamakan *et al.,* 2017). The best performances are highlighted in the bold face. However, it should be noted that the comparisons are for reference only due to many researchers having used different proportions of traffic types, sampling methods and pre-processing techniques.

In this study, the proposed detection approach has achieved a significant improvement in detecting th DoS type with 99.75%, except for Probe and R2L attack types with 74.19% and 66.67% respectively. Further investigation has revealed that the poor performances recorded are due to the fewer attack samples existing for both Probe and R2L attack types in the training set. However, it is worth mentioning that despite only using 10 features, the proposed detection method has indicated a significant improvement in half reduction in false alarm rate compared to the recent study by (Bamakan *et al.,* 2017).

6.4.3 ISCX 2012 Dataset

Table 6.5: Performance of proposed I-WEB using the ISCX 2012 testing dataset

Attack Traffic in Training Dataset	**Attack Traffic in Testing Dataset**	**Attack Traffic Detected by I-WEB**	**%age of Detected Attack traffic**
3,714	28,329	**28,234**	**99.66%**

As shown in Table 6.5, attack instances in the testing dataset are at least 7.6 times higher than attack instances in the training data. Although only a limited number of attacks are available in the training dataset, it is worth mentioning that the proposed approach is able to recognise attack instances in the testing dataset with a 99.66% detection rate.

Table 6.6: Performance comparisons obtained from the ISCX 2012 dataset

Methods	False Alarm Rate (%)	Detection Rate (%)	Accuracy (%)
Packet Header Anomaly Detection by Yassin *et al.* (2014)	N/A	99.04	N/A
SVM Anomaly Detection by Nyakundi (2015)	4.50	99.10	N/A
Computer Vision Techniques by Tan *et al.* (2015)	3.70	99.40	N/A
Payload based Anomaly Detection by Kakavand *et al.* (2016)	1.20	97.00	N/A
Evolved Specialized Ensembles by Folino *et al.* (2016)	N/A	91.37	N/A
Distributed SVM by Huang *et al.* (2017)	1.10	98.50	N/A
I-WEB (2018)	**0.08**	**99.66**	**99.71**

Table 6.6 shows the proposed I-WEB performance in terms of FAR, DR and ACC compared with the previous methods tested on the ISCX 2012 dataset. The comparisons are for reference only due to many researchers having used different proportions of traffic types, sampling methods and pre-processing techniques. In this study, the proposed detection approach has demonstrated a slight improvement in term of detection rate and false alarm rate compared to the recent study by Huang *et al.* (2017) with 1.16% and 1.02% respectively. The proposed detection method achieved a detection rate above 99% along with a false alarm rate less than 0.1%. This had indicates that the proposed detection approach is suitable to employ in the field of IDS.

6.4.4 UNSW-NB15 Dataset

Table 6.7: Performance of proposed I-WEB using the UNSW-NB15 testing dataset

Dataset	**Backdoor**	**Fuzzers**	**Reconnaissance**	**Exploits**	**DoS**	**Worms**	**Generic**	**Analysis**
Training (instances)	9	251	470	2804	493	34	213	-
Testing (instances)	83	836	1603	8677	1216	114	289	558
Detected by I-WEB	83	787	1592	7730	1065	114	265	410
Detection Rate %age	100.00	94.14	99.31	89.09	87.58	100.00	91.70	73.48

Table 6.7 lists the attacks available in both training and testing of the UNSW-NB15 dataset. As mentioned in Section 3.2.4, this dataset is comprised of variety imbalanced synthetic attack traffic, which resulted in this dataset being more challenging to evaluate. In the training dataset, there are seven types of attack presents: *backdoor, fuzzers, reconnaissance, exploits, DoS, worms and generic,* whilst eight types of attack: *backdoor, fuzzers, reconnaissance, exploits, analysis, DoS, worms and generic* are in the testing dataset. As can be seen, the main difference between the training data and the testing data is that the latter contains a new attack named *"analysis"*.

The proposed approach successfully obtained 90.06% detection rate among all attack traffic existing in the testing dataset. The attack types with the highest detection rate are *backdoor* and *worms* with (100%), followed by *reconnaissance* (99.31%), *fuzzers* (94.14%), *generic* (91.70%), *exploits* (89.09%), *DoS* (87.58%) and the lowest is *analysis* with (73.48%).

The result has indicated five out of eight types of attack have achieved more than 90% detection rate by the proposed approach. Upon closer analysis, the poor performance of *analysis* is due to the unavailability of samples residing in the training dataset, which makes it difficult for the system to classify it as an attack. However, it is worth mentioning that the proposed approach is still able to recognise *analysis* in 73.48% of the time.

Table 6.8: Performance comparisons obtained on the UNSW-NB15 dataset

Classifiers	Accuracy Rate %age	False Alarm Rate %age
DT	85.56	15.78
LR	83.15	18.48
NB	82.07	18.56
ANN	81.34	21.13
EM clustering	78.47	23.79
I-WEB (2018)	**90.56**	**8.19**

Table 6.8 shows the proposed I-WEB performance in terms of ACC and FAR compared with the previous methods tested on the UNSW-NB 15 dataset, as reported in Moustafa and Slay (2016). The comparisons are for reference only due to the different proportions of traffic types, sampling methods and pre-processing techniques used in the study. In this study, the proposed detection approach has demonstrated significant improvement in term of overall accuracy with fewer false alarm rates of 5.00% and 10.29% respectively over the best performance of Moustafa and Slay (2016). In addition, the results have indicated that the proposed detection approach is able to perform well in identifying sophisticated attacks within the modern network environment.

6.5 Summary

This chapter presented the results and discussion of the proposed detection scheme which is divided into 3-phases (pre-processing, anomaly detection and post-processing). The comparative performances of each method are analysed and compared with existing methods.

Table 6.9: Performances of four different datasets

Dataset/ Metrics	DARPA 1999	NSL KDD	ISCX 2012	UNSW-NB15
Normal Detection Rate (N-DR)	99.87%	99.96%	99.92%	91.81%
Attack Detection Rate (A-DR)	95.84%	99.39%	99.66%	90.06%
False Alarm Rate (FAR)	0.13%	0.04%	0.08%	8.19%
False Negative Rate (FNR)	4.16%	0.61%	0.34%	9.94%
Accuracy Rate (ACC)	99.41%	99.68%	99.71%	90.56%
Known Attack Detected	95.82%	99.64%	-	90.78%
Unknown Attack Detected	100%	78.57%	-	73.48%

Table 6.9 shows the summary of the proposed detection performances on four different datasets. The performance of the proposed method varying over different datasets is due to the nature of the dataset itself. Overall, the proposed approach had recorded above a 99% accuracy rate on DARPA 1999, NSL KDD and ISCX 2012 datasets. Despite the poor performance shown in the UNSW-NB15 dataset, the proposed method is still able to achieved 90.56% detection accuracy rate. In the next chapter, the overall conclusion, contribution of this research and potential for future work are discussed.

Chapter 7

Conclusion and Future Work

7.1 Introduction

This chapter presents the concluding chapter of the thesis. It has several sections, in particular, the conclusion of the thesis and the findings are presented according to the research questions, followed by contributions, limitations of the study and the future work recommendation.

7.2 Main Findings and the Summary of the Thesis

In this study, the proposed detection scheme consists of three major phases which are known as Pre-Processing, Anomaly Detection and Post-Processing. The detection scheme is specially designed to address the main research question on "*How should system detection performance be improved in order to identify known and unknown web attacks?"* Thus, four specific research questions are discussed in the following sub-sections.

7.2.1 Research Question One

The first question was *"What approach can be used to select prominent features within the dataset?"* The usage of all features that may contain redundant and irrelevant features can deteriorate the attack detection performance. In addition, to process all features, significant resources are needed which will make the system expensive. In this study, the proposed HFS had leverages both in strengths from filter and wrapper methods to efficiently obtain the most prominent features that could contribute to reducing the data dimensionality and improving the overall detection accuracy. This work has been published in (Kamarudin *et al.,* 2016).

7.2.2 Research Question Two

The second question was *"How can the false detection rate produced using conventional statistical techniques be reduced?"* The conventional statistical technique was solely dependent on outliers presented in the traffic for attack identification, which cause it to generate a high false detection rate. Thus, this study investigates the effect of feature size along with the implementation of ED and CIT to overcome the aforementioned drawback. The experimental results have demonstrated significant improvement in terms of overall false detection rate reduction. This work has been published in (Kamarudin *et al.,* 2017b).

7.2.3 Research Question Three

The third question was *"What is the suitable combination of classifiers in boosting algorithms that could improve the attack detection performance?"* In view of boosting technique required base algorithm, in this study, several experiments were conducted to choose a suitable base algorithm that could perform well with LogitBoost. LogitBoost was selected as an alternative solution to address the drawback of Adaboost in handling noise and outliers. The results indicated that the Random Forest (RF) algorithm is the most suitable base algorithm to combine with LogitBoost. The combination of LogitBoost and RF (LB+RF) demonstrated improvement on detection accuracy with lowest false detection rate when compared with other algorithms. This work has been published in (Kamarudin *et al.,* 2017a).

7.2.4 Research Question Four

The fourth research question was *"How can the detection ability be improved in order to identify similar attacks in the future?"* Protecting the business needs requires IDS to quickly identify and respond to any possibility of attacks. The total re-initiation procedure can be reduced by truncating some unnecessary processes such as detecting similar attacks in the future. In this study, it was found that the implementation of attack signature as part of the detection strategy has significantly reduced the system detection time. Moreover, the prioritisation model proposed in this study had eased the task of security analysts for further incident management.

It is undeniable that ABDS approaches have been extensively applied in the past. Nevertheless, methods using statistical and DM particularly classification, still remain the most popular choice and have become active research areas. This is due to their capability in determining patterns that could identify and distinguish known and unknown attack traffic more efficiently. The effectiveness and performances of the proposed detection scheme have been evaluated under four different datasets namely DARPA 1999, NSL KDD, ISCX 2012 and UNSW-NB15 for the purpose of allowing different integration testing environments that contain multiple types of attack traffics. The experimental results have demonstrated that the proposed detection scheme has successfully recognised some unknown attacks and achieved comparable performance with other established state-of-the-art IDS algorithms.

7.3 Contributions of the Study

This research has presented a significant contribution to network security in both theory and practice. From a theoretical perspective, the proposed detection scheme is smartly designed to recognise and identify known and unknown attack traffic more precisely. In terms of dimensionality reduction, this study has proposed hybrid approaches that leveraged the strengths of both filter and wrapper selection to effectively select the optimal features in detecting intrusion. In addition, the synergy between statistical analysis and classification approaches has significantly improved overall detection accuracy as well as reduced the false detection rate. Furthermore, the employment of a signature approach to recognise a similar attack has truncated the total re-initiation process as the similar attack can be filtered out. Moreover, the introduction of IPM in this study has made the incident response procedure remarkably straightforward. The thesis also makes a practical contribution through the findings in this study. This study offers substantial benefits to organisations, particularly in terms of identifying attack traffic and reducing the system's operational costs at the same time. The experimental results have demonstrated that the combination of detection methods has enabled I-WEB to identify a variety of unknown attacks. Due to these factors, most organisations today find an IDS to be an important component to complement other security measures that could enhance overall network security strategy.

7.4 Limitations of the Study

In Chapter 6, the proposed detection scheme has adequately achieved its aims and objectives. However, some limitations have been found in this research and they are listed as follows for future consideration:

1. *Datasets.* The datasets used for experiments and evaluations are limited to synthetic datasets only. This is because a real-live dataset is difficult to obtain due to confidentiality and privacy issues.
2. *HTTP Traffic.* The employment of HTTP traffic is due to the unavailability of HTTPS traffic in the synthetic dataset. To represent HTTPS protocol, a traffic header is employed without the payload information.
3. *Computational Cost.* The aim of the signature generation is to reduce the total re-initiation process of detecting similar attacks in the future. However, less focus was given to reducing the computational cost of the statistical and ensemble approaches.
4. *Attack Signature.* The attack signatures generated in this study are not tested in a real traffic environment. Thus, the time reduction may not reflect the real performance.
5. *Attack Prioritisation.* The prioritisation technique proposed in this study is limited to demonstration purposes and easy understanding. The study considers assumptions in order to obtain the severity level. Thus, the generated attack quadrant may not illustrate the real criticality level because it is not tested with real data.

7.5 Future Work

To deal with myriad traffic connection each day, the IDS would require adequate, proficient and updated normal traffic behaviour to maintain detection accuracy. One of the solutions is to frequently update the normal profile on a regular basis. Thus the adaptive approach is highly recommended in an effort to keep the normal profile relevant at all times.

It is essential to employ a hybrid detection system in the real environment whereby the MBDS and ABDS methods are combined. The strength of fast detection in MBDS and the convenience of ABDS in detecting unknown attacks could enhance the detection system ability. However, further investigation to reduce the impact of generating a false signature is required as it could reduce the system performance.

To cope with the current demand for a more secure communication environment, some cryptography techniques such as IPSec, VPN and SSL protocols are implemented within the networks. As such, in future, the proposed detection model can be used to experiment on such protocols to discover attacks over encrypted traffic.

The identified attacks will usually require further processing by a security analyst to discover its impact on the system. This procedure can be damaging if the attack's criticality is not appropriately managed. The prioritisation model requires further evaluation in order to determine its true effectiveness.

References

A. Aziz, A. S., Hanafi, S. E.-O., & Hassanien, A. E. (2016). Comparison of classification techniques applied for network intrusion detection and classification. *Journal of Applied Logic*, *1*, 1–10. http://doi.org/10.1016/j.jal.2016.11.018

Abdallah, A., Maarof, M. A., & Zainal, A. (2016). Fraud detection system: A survey. *Journal of Network and Computer Applications*, *68*, 90–113. http://doi.org/10.1016/j.jnca.2016.04.007

Abhaya, K., Jha, R., & Afroz, S. (2014). Data Mining Techniques for Intrusion Detection: A Review. *International Journal of Advanced Research in Computer and Communication Engineering*, *3*(6), 6938–6942.

Aburomman, A. A., & Reaz, M. B. I. (2017). A survey of intrusion detection systems based on ensemble and hybrid classifiers. *Computers & Security*, *65*, 135–152. http://doi.org/10.1016/j.cose.2016.11.004

Acarali, D., Rajarajan, M., Komninos, N., & Herwono, I. (2016). Survey of approaches and features for the identification of HTTP-based botnet traffic. *Journal of Network and Computer Applications*, *76*(October), 1–15. http://doi.org/10.1016/j.jnca.2016.10.007

Aditi, P., & Hitesh, G. (2013). A New Approach of Intrusion Detection System using Clustering , Classification and Decision Table. In *Proc. of Int. Conf. on Advances in Computer Science and Application*.

Agarwal, B., & Mittal, N. (2012). Hybrid Approach for Detection of Anomaly Network Traffic using Data Mining Techniques, *6*, 996–1003. http://doi.org/10.1016/j.protcy.2012.10.121

Ahmed, M., Naser Mahmood, A., & Hu, J. (2016). A survey of network anomaly detection techniques. *Journal of Network and Computer Applications*, *60*, 19–31. http://doi.org/10.1016/j.jnca.2015.11.016

AL-Nabi, D., & Ahmed, S. (2013). Survey on Classification Algorithms for Data

Mining:(Comparison and Evaluation). *Computer Engineering and Intelligent Systems*, *1719*(8), 18–25. Retrieved from http://www.iiste.org/Journals/index.php/CEIS/article/view/6575

Aljawarneh, S., Aldwairi, M., & Yassein, M. B. (2017). Anomaly-Based Intrusion Detection System Through Feature Selection Analysis And Building Hybrid Efficient Model. *Journal of Computational Science*, 1–9. http://doi.org/10.1016/j.jocs.2017.03.006

Amancio, D. R., Comin, C. H., Casanova, D., Travieso, G., Bruno, O. M., Rodrigues, F. A., & da Fontoura Costa, L. (2014). A Systematic Comparison of Supervised Classifiers. *PLoS ONE*, *9*(4), e94137. http://doi.org/10.1371/journal.pone.0094137

Ambusaidi, M. a., He, X., Tan, Z., Nanda, P., Lu, L. F., & Nagar, U. T. (2014). A Novel Feature Selection Approach for Intrusion Detection Data Classification. *2014 IEEE 13th International Conference on Trust, Security and Privacy in Computing and Communications*, 82–89. http://doi.org/10.1109/TrustCom.2014.15

Anderson, J. P. (1980). Computer Security Threat Monitoring and Surveillance. *Tech. Rep. February 26*.

Anuar, N. B., Papadaki, M., Furnell, S., & Clarke, N. (2013). Incident prioritisation using analytic hierarchy process (AHP): Risk Index Model (RIM). *Security and Communication Networks*, *6*(9), 1087–1116. http://doi.org/10.1002/sec.673

Attal, F., Boubezoul, A., Oukhellou, L., & Espie, S. (2015). Powered two-wheeler riding pattern recognition using a machine-learning framework. *IEEE Transactions on Intelligent Transportation Systems*, *16*(1), 475–487. http://doi.org/10.1109/TITS.2014.2346243

Bamakan, S. M. H., Wang, H., & Shi, Y. (2017). Ramp loss K-Support Vector Classification-Regression; a robust and sparse multi-class approach to the intrusion detection problem. *Knowledge-Based Systems*, *126*, 113–126. http://doi.org/10.1016/j.knosys.2017.03.012

Bamakan, S. M. H., Wang, H., Yingjie, T., & Shi, Y. (2016). An effective intrusion detection framework based on MCLP/SVM optimized by time-varying chaos particle swarm optimization. *Neurocomputing*, *199*, 90–102. http://doi.org/10.1016/j.neucom.2016.03.031

Bayarjargal, D., & Cho, G. (2014). Detecting an Anomalous Traffic Attack Area based on Entropy Distribution and Mahalanobis Distance. *International Journal of Security and Its Applications*, *8*(2), 87–94. http://doi.org/10.14257/ijsia.2014.8.2.09

Ben Amor, N. Ben, Benferhat, S., & Elouedi, Z. (2004). Naive Bayes vs decision trees in intrusion detection systems. *Proceedings of the 2004 ACM Symposium on Applied Computing*, 420–424. http://doi.org/10.1145/967900.967989

Bolón-Canedo, V., Sánchez-Maroño, N., & Alonso-Betanzos, A. (2015). *Feature Selection for High-Dimensional Data*. *Springer*. Cham: Springer International Publishing. http://doi.org/10.1007/978-3-319-21858-8

Breiman, L. (1996). Bagging predictors. *Machine Learning*, *24*(2), 123–140. http://doi.org/10.1007/BF00058655

Breiman, L. (2001). Random forests. *Machine Learning*, *45*(1), 5–32. http://doi.org/10.1023/A:1010933404324

Brown, C., Cowperthwaite, A., Hijazi, A., & Somayaji, A. (2009). Analysis of the 1999 DARPA/Lincoln Laboratory IDS Evaluation Data with NetADHICT. *IEEE Symposium on Computational Intelligence for Security and Defense Applications, CISDA 2009*, (Cisda). http://doi.org/10.1109/CISDA.2009.5356522

CAIDA. (2011). Retrieved March 9, 2018, from http://www.caida.org/research/traffic-analysis/

Cao, J., Kwong, S., & Wang, R. (2012). A noise-detection based AdaBoost algorithm for mislabeled data. *Pattern Recognition*, *45*(12), 4451–4465. http://doi.org/10.1016/j.patcog.2012.05.002

Chakir, E. M., Moughit, M., & Khamlichi, Y. I. (2017). An Effecient Method for

Evaluating Alerts of Intrusion Detection Systems National School of Applied Sciences USMBA. http://doi.org/10.1109/WITS.2017.7934678

Chan, K. K., Tong, K. S., & Huang, Y. H. (2005). Implementation of BS7799 standard on the PACS in hospital. *IEEE International Engineering Management Conference*, *II*, 661–662. http://doi.org/10.1109/IEMC.2005.1559231

Chebrolu, S., Abraham, A., & Thomas, J. P. (2005). Feature deduction and ensemble design of intrusion detection systems. *Computers & Security*, *24*(4), 295–307. http://doi.org/10.1016/j.cose.2004.09.008

Chen, C.-M., Chen, Y.-L., & Lin, H.-C. (2010). An efficient network intrusion detection. *Computer Communications*, *33*(4), 477–484. http://doi.org/10.1016/j.comcom.2009.10.010

Choraś, R. S. (2015). Image Processing & Communications Challenges 6 (Vol. 313). Springer International Publishing. http://doi.org/10.1007/978-3-319-10662-5

Cleetus, N. (2014). Genetic algorithm with Different Feature Selection Method for Intrusion Detection. *First International Conference on Computational Systems and Communications (ICCSC)*, (December), 220–225.

Cortes, C., & Vapnik, V. (1995). Support-Vector Networks. *Machine Learning*, *20*(3), 273–297. http://doi.org/10.1023/A:1022627411411

Cyber Attacks. (2016). Cyber Attacks. Retrieved October 21, 2016, from http://www.reuters.com/article/us-usa-cyber-idUSKCN12L1ME

Davis, J. J., & Clark, A. J. (2011). Data preprocessing for anomaly based network intrusion detection: A review. *Computers & Security*, *30*(6–7), 353–375. http://doi.org/10.1016/j.cose.2011.05.008

De La Hoz, E., De La Hoz, E., Ortiz, A., Ortega, J., & Martinez-Alvarez, A. (2014). Feature selection by multi-objective optimisation: Application to network anomaly detection by hierarchical self-organising maps. *Knowledge-Based Systems*, *71*, 322–338. http://doi.org/10.1016/j.knosys.2014.08.013

DEFCON: Defcon capture the flag (CTF) contest. (2000). Retrieved March 9, 2018, from https://www.defcon.org/html/defcon-8/defcon-8-post.html

Denning, D. E. (1987). Intrusion-Detection Model. *IEEE Transactions on Software Engineering*, (2), 222–232.

Dietterich, T. G. (2000). Ensemble Methods in Machine Learning. *Lecture Notes in Computer Science*, *1857*, 1–15. http://doi.org/10.1007/3-540-45014-9_1

El-Khatib, K. (2010). Impact of feature reduction on the efficiency of wireless intrusion detection systems. *IEEE Transactions on Parallel and Distributed Systems*, *21*(8), 1143–1149. http://doi.org/10.1109/TPDS.2009.142

Estévez-Tapiador, J. M., García-Teodoro, P., & Díaz-Verdejo, J. E. (2004). Measuring normality in HTTP traffic for anomaly-based intrusion detection. *Computer Networks*, *45*(2), 175–193. http://doi.org/10.1016/j.comnet.2003.12.016

Fakhraei, S., Soltanian-Zadeh, H., & Fotouhi, F. (2014). Bias and stability of single variable classifiers for feature ranking and selection. *Expert Systems with Applications*, *41*(15), 6945–6958. http://doi.org/10.1016/j.eswa.2014.05.007

Faraoun, K. M., & Boukelif, A. (2006). Multi-Category Pattern Classification Applied. *International Journal of Computational Intelligence and Applications*, *6*(1), 77–99.

Farid, D. M., Zhang, L., Rahman, C. M., Hossain, M. A., & Strachan, R. (2014). Hybrid decision tree and naïve Bayes classifiers for multi-class classification tasks. *Expert Systems with Applications*, *41*(4 PART 2), 1937–1946. http://doi.org/10.1016/j.eswa.2013.08.089

Farnaaz, N., & Jabbar, M. A. (2016). Random Forest Modeling for Network Intrusion Detection System. *Procedia Computer Science*, *89*, 213–217. http://doi.org/10.1016/j.procs.2016.06.047

Folino, G., Pisani, F. S., & Sabatino, P. (2016). A distributed intrusion detection framework based on evolved specialized ensembles of classifiers. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture*

Notes in Bioinformatics), *9597*, 315–331. http://doi.org/10.1007/978-3-319-31204-0

Folino, G., Pizzuti, C., & Spezzano, G. (2010). An ensemble-based evolutionary framework for coping with distributed intrusion detection. *Genetic Programming and Evolvable Machines*, *11*(2), 131–146. http://doi.org/10.1007/s10710-010-9101-6

Frank, E., Hall, M. A., & Witten, I. H. (2016). *The WEKA Workbench. Morgan Kaufmann, Fourth Edition* (Fourth Edi). Morgan Kaufmann. Retrieved from https://www.cs.waikato.ac.nz/ml/weka/Witten_et_al_2016_appendix.pdf

Freund, Y., & Schapire, R. (1995). A desicion-theoretic generalization of on-line learning and an application to boosting. *Computational Learning Theory*, *55*, 119–139. http://doi.org/10.1006/jcss.1997.1504

Friedman, J., Hastie, T., & Tibshirani, R. (2000). Additive Logistic Regression. *The Annals of Statistics*. http://doi.org/10.1214/aos/1016218223

Fugate, S. J. (2012). Methods for Speculatively Bootstrapping Better Intrusion Detection System Performance. *UMI Dissertations Publishing*.

Gaikwad, D. P., & Thool, R. C. (2015). Intrusion Detection System Using Bagging with Partial Decision TreeBase Classifier. *Procedia Computer Science*, *49*, 92–98. http://doi.org/10.1016/j.procs.2015.04.231

Gonzalez, V. M., Galicia, L., & Favela, J. (2008). Understanding and supporting personal activity management by IT service workers. *Proceedings of the 2nd ACM Symposium on Computer Human Interaction for Management of Information Technology - CHiMiT '08*, 1. http://doi.org/10.1145/1477973.1477976

Gunes Kayacik, H., Nur Zincir-Heywood, A., & Heywood, M. I. (2007). A hierarchical SOM-based intrusion detection system. *Engineering Applications of Artificial Intelligence*, *20*(4), 439–451. http://doi.org/10.1016/j.engappai.2006.09.005

Guo, C., Ping, Y., Liu, N., & Luo, S. S. (2016). A two-level hybrid approach for intrusion detection. *Neurocomputing*, *214*, 391–400. http://doi.org/10.1016/j.neucom.2016.06.021

Guyon, I. (2003). An Introduction to Variable and Feature Selection. *Journal of Machine Learning Research 3*, *3*, 1157–1182. http://doi.org/10.1023/A:1012487302797

Haimes, Y. Y. (2001). Risk Analysis , Systems Analysis , and Covey ' s Seven Habits. *Risk Analysis*, *21*(2), 217–224.

Hair, J. F., Black, C. W., Babin, B. J., & Anderson, R. E. (2009). Multivariate Data Analysis: A Global Perspective. *Prentice Hall*, 816. http://doi.org/10.1038/259433b0

Hakimi, Z., & Faez, K. (2013). An Efficient Architecture for Distributed Intrusion Detection System. *10th International ISC Conference on Information Security and Cryptology (ISCISC), 2013*. http://doi.org/10.1109/ISCISC.2013.6767356

Hall, M. A. (1999). *Correlation-based Feature Subset Selection for Machine Learning. Methodology*. Hamilton, New Zeland. Retrieved from http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.37.4643&rep=rep1&type=pdf

Hasan, A. M., Nasser, M., Pal, B., & Ahmad, S. (2014). Support Vector Machine and Random Forest Modeling for Intrusion Detection System (IDS). *Journal of Intelligent Learning Systems and Applications*, *6*(February), 45–52.

Hosseinpour, F., Vahdani Amoli, P., Farahnakian, F., Plosila, J., & Hamalainen, T. (2014). Artificial Immune System Based Intrusion Detection: Innate Immunity using an Unsupervised Learning Approach. *International Journal of Digital Content Technology and Its Applications*, *8*(5), 1–12. http://doi.org/10.1007/978-81-322-0970-6

Htun, P. T., & Khaing, K. T. (2013). Detection Model for Daniel-of-Service Attacks using Random Forest and k-Nearest Neighbors. *International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)*, *2*(5).

Hu, W., Hu, W., & Maybank, S. (2008). AdaBoost-based algorithm for network intrusion detection. *IEEE Transactions on Systems, Man, and Cybernetics*, *38*(2), 577–83. http://doi.org/10.1109/TSMCB.2007.914695

Huang, H., Khalid, R. S., & Yu, H. (2017). *Data Science and Big Data: An Environment of Computational Intelligence* (Vol. 24). http://doi.org/10.1007/978-3-319-53474-9

Hubballi, N., Biswas, S., & Nandi, S. (2013). Towards reducing false alarms in network intrusion detection systems with data summarization technique. *Security and Communication Networks*, *6*(3), 275–285. http://doi.org/10.1002/sec.562

Iii, K. L. I. (2007). *Anomaly Detection for HTTP Intrusion Detection: Algorithm Comparisons and the Effect of Generalization on Accuracy*. Retrieved from http://digitalrepository.unm.edu/cs_etds

Jain, A., Bhupendra, & J.L., R. (2016). CLASSIFIER SELECTION MODELS FOR INTRUSION DETECTION SYSTEM (IDS). *Informatics Engineering, an International Journal (IEIJ), 4*(1), 1–11. http://doi.org/10.5121/ieij.2016.4101

Jalil, K. a, Kamarudin, M. H., & Masrek, M. N. (2010). Comparison of Machine Learning algorithms performance in detecting network intrusion. *Networking and Information Technology ICNIT 2010 International Conference on*, 221–226. Retrieved from http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5508526

Jang, Y. S., & Choi, J. Y. (2014). Detecting SQL injection attacks using query result size. *Computers and Security*, *44*, 104–118. http://doi.org/10.1016/j.cose.2014.04.007

Ji, S. Y., Jeong, B. K., Choi, S., & Jeong, D. H. (2016). A multi-level intrusion detection method for abnormal network behaviors. *Journal of Network and Computer Applications*, *62*, 9–17. http://doi.org/10.1016/j.jnca.2015.12.004

Jones, S. (2017). NHS seeks to recover from global cyber-attack as security concerns resurface. Retrieved June 2, 2017, from https://www.theguardian.com/society/2017/may/12/hospitals-across-england-hit-by-large-scale-cyber-attack

Kakavand, M., Mustapha, N., Mustapha, A., & Abdullah, M. T. (2016). Effective dimensionality reduction of payload-based anomaly detection in TMAD model for HTTP payload. *KSII Transactions on Internet and Information Systems*, *10*(8), 3884–3910. http://doi.org/10.3837/tiis.2016.08.025

Kamarudin, M.H., Maple, C. and Watson, T. (2016). Hybrid feature selection technique for intrusion detection system. *Int. J. High Performance Computing and Networking*.

Kamarudin, M. H., Maple, C., Watson, T., & Safa, N. S. (2017a). A LogitBoost-Based Algorithm for Detecting Known and Unknown Web Attacks. *IEEE Access*, *5*, 26190–26200. http://doi.org/10.1109/ACCESS.2017.2766844

Kamarudin, M. H., Maple, C., Watson, T., & Safa, N. S. (2017b). A New Unified Intrusion Anomaly Detection in Identifying Unseen Web Attacks. *Security and Communication Networks*, *2017*, 1–18. http://doi.org/10.1155/2017/2539034

KDD. (1999). Retrieved March 9, 2018, from http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html

Khammassi, C., & Krichen, S. (2017). A GA-LR wrapper approach for feature selection in network intrusion detection. *Computers & Security*, *70*, 255–277. http://doi.org/10.1016/j.cose.2017.06.005

Khoshgoftaar, T. M., Seiffert, C., Van Hulse, J., Napolitano, A., & Folleco, A. (2007). Estimating Class Probabilities in Random Forest. *Proceedings - 6th International Conference on Machine Learning and Applications, ICMLA 2007*, 348–353. http://doi.org/10.1109/ICMLA.2007.64

Kim, M. Y., & Lee, D. H. (2014). Data-mining based SQL injection attack detection using internal query trees. *Expert Systems with Applications*, *41*(11), 5416–5430. http://doi.org/10.1016/j.eswa.2014.02.041

Kirillov, A. V., Tanatova, D. K., Vinichenko, M. V., & Makushkin, S. A. (2015). Theory and practice of time-management in education. *Asian Social Science*, *11*(19), 193–204. http://doi.org/10.5539/ass.v11n19p193

Kosamkar, V., & Chaudhari, S. S. (2014). Improved Intrusion Detection System using C4 . 5 Decision Tree and Support Vector Machine. *International Journal of Computer Science and Information Technologies*, *5*(2), 1463–1467.

Kruegel, C., Vigna, G., & Robertson, W. (2005). A multi-model approach to the

detection of web-based attacks. *Computer Networks*, *48*(5), 717–738. http://doi.org/10.1016/j.comnet.2005.01.009

Kuang, F., Xu, W., & Zhang, S. (2014). A novel hybrid KPCA and SVM with GA model for intrusion detection. *Applied Soft Computing Journal*, *18*, 178–184. http://doi.org/10.1016/j.asoc.2014.01.028

Laranjeiro, N., Vieira, M., & Madeira, H. (2010). A learning-based approach to secure web services from SQL/XPath Injection attacks. *Proceedings - 16th IEEE Pacific Rim International Symposium on Dependable Computing, PRDC 2010*, 191–198. http://doi.org/10.1109/PRDC.2010.24

LBNL. (2005). Retrieved March 9, 2018, from http://www.icir.org/enterprise-tracing/

Levy, E. (2004). Approaching zero. *IEEE Security and Privacy*, *2*(4), 65–66. http://doi.org/10.1109/MSP.2004.33

Li, A. H., & Bradic, J. (2018). Boosting in the Presence of Outliers : Adaptive Classification With Nonconvex Loss Functions. *Journal of the American Statistical Association*, *0*(0), 1–15. http://doi.org/10.1080/01621459.2016.1273116

Li, W., & Li, Q. (2010). Using Naive Bayes with AdaBoost to Enhance Network Anomaly Intrusion Detection. *Intelligent Networks and Intelligent Systems (ICINIS), 2010 3rd International Conference on*. http://doi.org/10.1109/ICINIS.2010.133

Liao, H. J., R. Lin, & Lin, C. H. (2013). Intrusion detection system: A comprehensive review. *Journal of Network and Computer Applications*.

Lippmann, R., Haines, J. W., Fried, D. J., Korba, J., & Das, K. (2000). The 1999 DARPA off-line intrusion detection evaluation. *Computer Networks*, *34*, 579–595.

Liu, H., & Motoda, H. (1998). *Feature selection for knowledge discovery and data mining*. Kluwer Academic Print Publisher. Retrieved from http://books.google.com/books?id=NBAsVJpCVjIC

Louvieris, P., Clewley, N., & Liu, X. (2013). Effects-based feature identification for network intrusion detection. *Neurocomputing*, *121*, 265–273.

http://doi.org/10.1016/j.neucom.2013.04.038

Mahoney, M. V., & Chan, P. K. (2001). PHAD: Packet Header Anomaly Detection for Identifying Hostile Network Traffic. *Florida Technology, Tech. Rep. CS-2001*.

Martignoni, L., Paleari, R., & Bruschi, D. (2010). Conqueror: Tamper-Proof Code Execution on Legacy Systems. In *Detection of Intrusions and Malware, and Vulnerability Assessment* (pp. 21–40). Springer Berlin Heidelberg. http://doi.org/10.1007/978-3-642-14215-4_2

McHugh, J. (2000). Testing Intrusion detection systems: a critique of the 1998 and 1999 DARPA intrusion detection system evaluations as performed by Lincoln Laboratory. *ACM Transactions on Information and System Security*, *3*(4), 262–294. http://doi.org/10.1145/382912.382923

Meng, Y., & Kwok, L. F. (2014). Adaptive blacklist-based packet filter with a statistic-based approach in network intrusion detection. *Journal of Network and Computer Applications*, *39*(1), 83–92. http://doi.org/10.1016/j.jnca.2013.05.009

Mitchell, R., & Chen, I.-R. (2014). Behavior Rule Specification-based Intrusion Detection for Safety Critical Medical Cyber Physical Systems. *IEEE Transactions on Dependable and Secure Computing*, *12*(1), 1–1. http://doi.org/10.1109/TDSC.2014.2312327

Moustafa, N., & Slay, J. (2016). The evaluation of Network Anomaly Detection Systems: Statistical analysis of the UNSW-NB15 data set and the comparison with the KDD99 data set. *Information Security Journal: A Global Perspective*, *25*(1–3), 18–31. http://doi.org/10.1080/19393555.2015.1125974

Muda, Z., & W. Yassin. (2011). A K-Means and Naive Bayes learning approach for better intrusion detection. *IEEE Information Technology*.

Muda, Z., Yassin, W., Sulaiman, M. N., & Udzir, N. I. (2011a). Intrusion Detection based on K-Means Clustering and Naïve Bayes Classification. *7th International Conference on IT in Asia (CITA) Intrusion*, 1–6. http://doi.org/10.1109/CITA.2011.5999520

Muda, Z., Yassin, W., Sulaiman, M. N., & Udzir, N. I. (2011b). Intrusion detection based on K-means clustering and OneR classification. *Proceedings of the 2011 7th International Conference on Information Assurance and Security, IAS 2011*, 192–197. http://doi.org/10.1109/ISIAS.2011.6122818

Mukkamala, S., Sung, A. H., & Abraham, A. (2005). Intrusion detection using an ensemble of intelligent paradigms. *Journal of Network and Computer Applications*, *28*(2), 167–182. http://doi.org/10.1016/j.jnca.2004.01.003

Mukkamala, S., Sung, a H., & Abraham, a. (2004). Modeling intrusion detection systems using linear genetic programming approach. *Innovations in Applied Artificial …*, *1*, 633–642. Retrieved from http://link.springer.com/chapter/10.1007/978-3-540-24677-0_65

Nguyen, H. H., Harbi, N., & Darmont, J. (2011). An efficient local region and clustering-based ensemble system for intrusion detection. *Proceedings of the 15th Symposium on International Database Engineering & Applications - IDEAS '11*, 185. http://doi.org/10.1145/2076623.2076647

Nguyen, H. T., Petrović, S., & Franke, K. (2010). A comparison of feature-selection methods for intrusion detection. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *6258 LNCS*, 242–255. http://doi.org/10.1007/978-3-642-14706-7_19

Noel, S., & Jajodia, S. (2008). Optimal IDS sensor placement and alert prioritization using attack graphs. *Journal of Network and Systems Management*, *16*(3), 259–275. http://doi.org/10.1007/s10922-008-9109-x

Nyakundi, E. M. (2015). *Using Support Vector Machines in Anomaly Intrusion Detection by*. The University of Guelph.

OWASP. Top 10 2017. (n.d.). OWASP. Retrieved October 2, 2017, from https://www.owasp.org

Panda, M., Abraham, A., & Patra, M. R. (2010). Discriminative multinomial naive bayes for network intrusion detection. *2010 6th International Conference on Information*

Assurance and Security, IAS 2010, 5–10. http://doi.org/10.1109/ISIAS.2010.5604193

Panda, M., Abraham, A., & Patra, M. R. (2012). A hybrid intelligent approach for network intrusion detection. *Procedia Engineering*, *30*(2011), 1–9. http://doi.org/10.1016/j.proeng.2012.01.827

Panda, M., & Patra, M. R. (2007). Network Intrusion Detection Using Naïve Bayes. *International Journal of Computer Science and Network Security*, *7*(12), 258–263. http://doi.org/10.1.1.128.936

Panda, M., & Patra, M. R. (2009). Ensembling rule based classifiers for detecting network intrusions. *ARTCom 2009 - International Conference on Advances in Recent Technologies in Communication and Computing*, 19–22. http://doi.org/10.1109/ARTCom.2009.121

Passeri, P. (2017). Hackmageddon. Retrieved September 11, 2017, from http://www.hackmageddon.com/2017/01/19/2016-cyber-attacks-statistics/

Patcha, A., & Park, J. M. (2007). An overview of anomaly detection techniques-existing solutions and latest technologies.pdf. *Computer Networks*.

Peng, Y., Wu, Z., & Jiang, J. (2010). A novel feature selection approach for biomedical data classification. *Journal of Biomedical Informatics*, *43*(1), 15–23. http://doi.org/10.1016/j.jbi.2009.07.008

Porras, P. A., Fong, M. W., & Valdes, A. (2002). A mission-impact-based approach to INFOSEC alarm correlation. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *2516*, 95–114. http://doi.org/10.1007/3-540-36084-0_6

Quinlan, J. R. (1986). Induction of Decision Trees. *Machine Learning*, *1*(1), 81–106. http://doi.org/10.1023/A:1022643204877

Raja, N. K., Arulanandam, K., & Rajeswari, B. R. (2012). Two-level packet inspection using sequential differentiate method. *International Journal of Computer Science and*

Network Security, *12*(4), 156–164. http://doi.org/10.1109/ICACC.2012.10

Raj Kumar, P. A., & Selvakumar, S. (2011). Distributed denial of service attack detection using an ensemble of neural classifier. *Computer Communications*, *34*(11), 1328–1341. http://doi.org/10.1016/j.comcom.2011.01.012

Ravale, U., Marathe, N., & Padiya, P. (2015). Feature Selection Based Hybrid Anomaly Intrusion Detection System Using K Means and RBF Kernel Function. *Procedia Computer Science*, *45*, 428–435. http://doi.org/10.1016/j.procs.2015.03.174

Razak, M. F. A., Anuar, N. B., Salleh, R., & Firdaus, A. (2016). The rise of "malware": Bibliometric analysis of malware study. *Journal of Network and Computer Applications*, *75*, 58–76. http://doi.org/10.1016/j.jnca.2016.08.022

S.Northcutt, Butler, J. M., & G.A. Board. (2008). Can you build a Defense in Depth architecture without an architect?, (May 13th).

Shamsuddin, S. B., & Woodward, M. E. (2007). Modeling Protocol Based Packet Header Anomaly Detector for Network and Host Intrusion Detection Systems. *Proceedings of the 6th International Conference on Cryptology and Network Security*, 209–227. http://doi.org/10.1007/978-3-540-76969-9

Shanmugam, B., & Idris, N. B. (2009). Improved intrusion detection system using fuzzy logic for detecting anamoly and misuse type of attacks. *SoCPaR 2009 - Soft Computing and Pattern Recognition*, 212–217. http://doi.org/10.1109/SoCPaR.2009.51

Shenzheng, Z. (2009). A dynamic normal profiling for anomaly detection. *Proceedings - 5th International Conference on Wireless Communications, Networking and Mobile Computing, WiCOM 2009*, 5–8. http://doi.org/10.1109/WICOM.2009.5301988

Shiravi, A., Shiravi, H., Tavallaee, M., & Ghorbani, A. a. (2012). Toward developing a systematic approach to generate benchmark datasets for intrusion detection. *Computers and Security*, *31*(3), 357–374. http://doi.org/10.1016/j.cose.2011.12.012

Singh, R., Kumar, H., & Singla, R. K. (2015). An intrusion detection system using

network traffic profiling and online sequential extreme learning machine. *Expert Systems with Applications*, *42*(22), 8609–8624. http://doi.org/10.1016/j.eswa.2015.07.015

Snort: (2002). Open source Network Intrusion Detection System, http://www.snort.org.

Syarif, I., Zaluska, E., Prugel-Bennett, A., & Wills, G. (2012). Application of bagging, boosting and stacking to intrusion detection. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *7376 LNAI*, 593–602. http://doi.org/10.1007/978-3-642-31537-4_46

Symantec Corporation. (2017). Internet Security Threat Report, *22*(April).

T. Hastie, R. Tibshirani, J. F. (2009). The Elements of Statistical Learning: Data Mining, Inference, and Prediction. *The Mathematical Intelligencer*, *27*(2), 83–85. http://doi.org/10.1007/b94608

Talavera, L. (2005). An evaluation of filter and wrapper methods for feature selection in categorical clustering. *Advances in Intelligent Data Analysis VI*, 742. http://doi.org/10.1007/11552253_40

Tan, Z., Jamdagni, A., He, X., & Member, S. (2015). Detection of Denial-of-Service Attacks Based on Computer Vision Techniques. *IEEE Transactions on Computers*, *64*(9), 2519–2533.

Tavallaee, M., Bagheri, E., Lu, W., & Ghorbani, A. A. (2009). A detailed analysis of the KDD CUP 99 data set. *IEEE Symposium on Computational Intelligence for Security and Defense Applications, CISDA 2009*, (Cisda), 1–6. http://doi.org/10.1109/CISDA.2009.5356528

Tesco Banks. (2016). Tesco Bank. Retrieved November 16, 2016, from http://www.bbc.co.uk/news/business-37915755

Thai, P., & De Oliveira, J. C. (2013). Decoupling policy from routing with software

defined interdomain management: Interdomain routing for SDN-based networks. *Proceedings - International Conference on Computer Communications and Networks, ICCCN*. http://doi.org/10.1109/ICCCN.2013.6614121

Thakare, S. V., & Gore, D. V. (2014). Comparative Study of CIA. *2014 Fourth International Conference on Communication Systems and Network Technologies*, 713–718. http://doi.org/10.1109/CSNT.2014.150

Thaseen, S., & Kumar, C. A. (2013). An analysis of supervised tree based classifiers for intrusion detection system. *Proceedings of the 2013 International Conference on Pattern Recognition, Informatics and Mobile Engineering, PRIME 2013*, 294–299. http://doi.org/10.1109/ICPRIME.2013.6496489

The Internet Traffic Archive. (2008).

Thomas, C., & Balakrishnan, N. (2009). Improvement in Intrusion Detection With Advances in Sensor Fusion. *IEEE Transactions on Information Forensics and Security*, *4*(3), 542–551. http://doi.org/10.1109/TIFS.2009.2026954

Threepak, T., & Watcharapupong, A. (2014). Web attack detection using entropy-based analysis. *International Conference on Information Networking*, (3), 244–247. http://doi.org/10.1109/ICOIN.2014.6799699

Tirenin, W., & Faatz, D. (1999). A concept for strategic cyber defense. *IEEE Military Communication Conference Proceeding*.

Tribak, H., Delgado-Marquez, B. L., Rojas, P., Valenzuela, O., Pomares, H., & Rojas, I. (2012). Statistical analysis of different artificial intelligent techniques applied to Intrusion Detection System. *2012 International Conference on Multimedia Computing and Systems*, 434–440. http://doi.org/10.1109/ICMCS.2012.6320275

Tsai, C. F., Hsu, Y. F., Lin, C. Y., & Lin, W. Y. (2009). Intrusion detection by machine learning: A review. *Expert Systems with Applications*, *36*(10), 11994–12000. http://doi.org/10.1016/j.eswa.2009.05.029

Tsang, C.-H., & Kwong, S. (2006). Ant Colony Clustering and Feature Extraction for

Anomaly Intrusion Detection. In *Studies in Computational Intelligence* (Vol. 34, pp. 101–123). http://doi.org/10.1007/978-3-540-34956-3_5

Tsang, C. H., Kwong, S., & Wang, H. (2007). Genetic-fuzzy rule mining approach and evaluation of feature selection techniques for anomaly intrusion detection. *Pattern Recognition*, *40*(9), 2373–2391. http://doi.org/10.1016/j.patcog.2006.12.009

Wahba, Y., Elsalamouny, E., & Eltaweel, G. (2015). Improving the Performance of Multi-class Intrusion Detection Systems using Feature Reduction Improving the Performance of Multi-class Intrusion Detection Systems using Feature Reduction, (JUNE 2015).

Wald, R., Khoshgoftaar, T., & Napolitano, A. (2013). Filter- and wrapper-based feature selection for predicting user interaction with Twitter bots. *Proceedings of the 2013 IEEE 14th International Conference on Information Reuse and Integration, IEEE IRI 2013*, 416–423. http://doi.org/10.1109/IRI.2013.6642501

Wang, Y. (2004). A hybrid intrusion detection system. *Iowa State University*. Retrieved from http://lib.dr.iastate.edu/rtd/1129

Wankhade, A. D., & Chatur, P. N. (2014). Comparison of Firewall and Intrusion Detection System, *5*(1), 674–678. Retrieved from www.ijcsit.com/docs/Volume 5/vol5issue01/ijcsit20140501145.pdf

Wolpert, D. H. (1992). Stacked generalization. *Neural Networks*, *5*(2), 241–259. http://doi.org/10.1016/S0893-6080(05)80023-1

Wozniak, M., Grana, M., & Corchado, E. (2014). A survey of multiple classifier systems as hybrid systems. *Information Fusion*, *16*(1), 3–17. http://doi.org/10.1016/j.inffus.2013.04.006

Wu, S. X., & Banzhaf, W. (2010). The use of computational intelligence in intrusion detection systems: A review. *Applied Soft Computing Journal*, *10*, 1–35. http://doi.org/10.1016/j.asoc.2009.06.019

Xiong, W., Xiong, N., Yang, L. T., Park, J. H., & Q.Wang. (2013). An Anomaly-based

Detection in Ubiquitous Network Using the Equilibrium State of the Catastrophe Theory. *Journal of Supercomputing*.

Yamada, A., Miyake, Y., Takemori, K., Studer, A., & Perrig, A. (2007). Intrusion detection for encrypted web accesses. *Proceedings - 21st International Conference on Advanced Information Networking and Applications Workshops/Symposia, AINAW'07*, *2*, 569–576. http://doi.org/10.1109/AINAW.2007.212

Yassin, W., Udzir, N., Abdullah, A., Abdullah, M., Muda, Z., & Zulzalil, H. (2014). Packet Header Anomaly Detection Using Statistical Analysis. *International Joint Conference SOCO'14-CISIS'14-ICEUTE'14 SE - 47*, *299*, 473–482. http://doi.org/10.1007/978-3-319-07995-0_47

Yoo, S., Yang, Y., & Carbonell, J. (2011). Modeling personalized email prioritization. In *Proceedings of the 20th ACM international conference on Information and knowledge management - CIKM '11* (p. 729). New York, New York, USA: ACM Press. http://doi.org/10.1145/2063576.2063683

Zainal, A., Maarof, M. a., & Shamsuddin, S. M. (2008). Data Reduction and Ensemble Classifiers in Intrusion Detection. *2008 Second Asia International Conference on Modelling and Simulation (AMS)*, 591–596. http://doi.org/10.1109/AMS.2008.146

Zaman, S., & Karray, F. (2009). Features Selection for Intrusion Detection Systems Based on Support Vector Machines. *2009 6th IEEE Consumer Communications and Networking Conference*, 1–8. http://doi.org/10.1109/CCNC.2009.4784780

Zhang, G., & Fang, B. (2007). LogitBoost classifier for discriminating thermophilic and mesophilic proteins. *Journal of Biotechnology*, *127*(3), 417–424. http://doi.org/10.1016/j.jbiotec.2006.07.020

Zhang, J., Zulkernine, M., & Haque, A. (2008). Random-Forests-Based Network Intrusion. *MAN and Cybernetics*, *38*(5), 649–659.

Zhang, L., & White, G. B. (2007). Anomaly detection for application level network attacks using payload keywords. *Proceedings of the 2007 IEEE Symposium on Computational Intelligence in Security and Defense Applications, CISDA 2007*, (Cisda),

178–185. http://doi.org/10.1109/CISDA.2007.368151

Zolotukhin, M., Hamalainen, T., Kokkonen, T., & Siltanen, J. (2014). Analysis of HTTP requests for anomaly detection of web attacks. *Proceedings - 2014 World Ubiquitous Science Congress: 2014 IEEE 12th International Conference on Dependable, Autonomic and Secure Computing, DASC 2014*, 406–411. http://doi.org/10.1109/DASC.2014.79

Zomlot, L., Sundaramurthy, S. C., Luo, K., Ou, X., & Rajagopalan, S. R. (2011). Prioritizing intrusion analysis using Dempster-Shafer theory. *Proceedings of the 4th ACM Workshop on Security and Artificial Intelligence - AISec '11*, 59. http://doi.org/10.1145/2046684.2046694

Appendix A

A.1 First Preliminary Experiment

A.1.1.1 DARPA 1999 Dataset

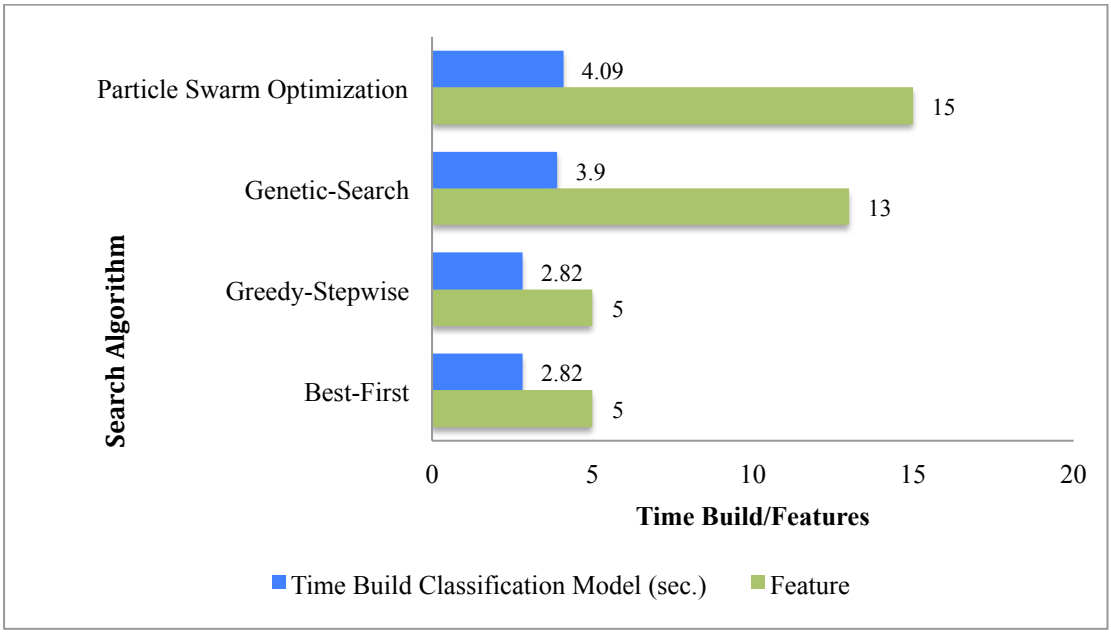


Figure A.1: Building Time and Feature Selected by Search Algorithm with DARPA 1999

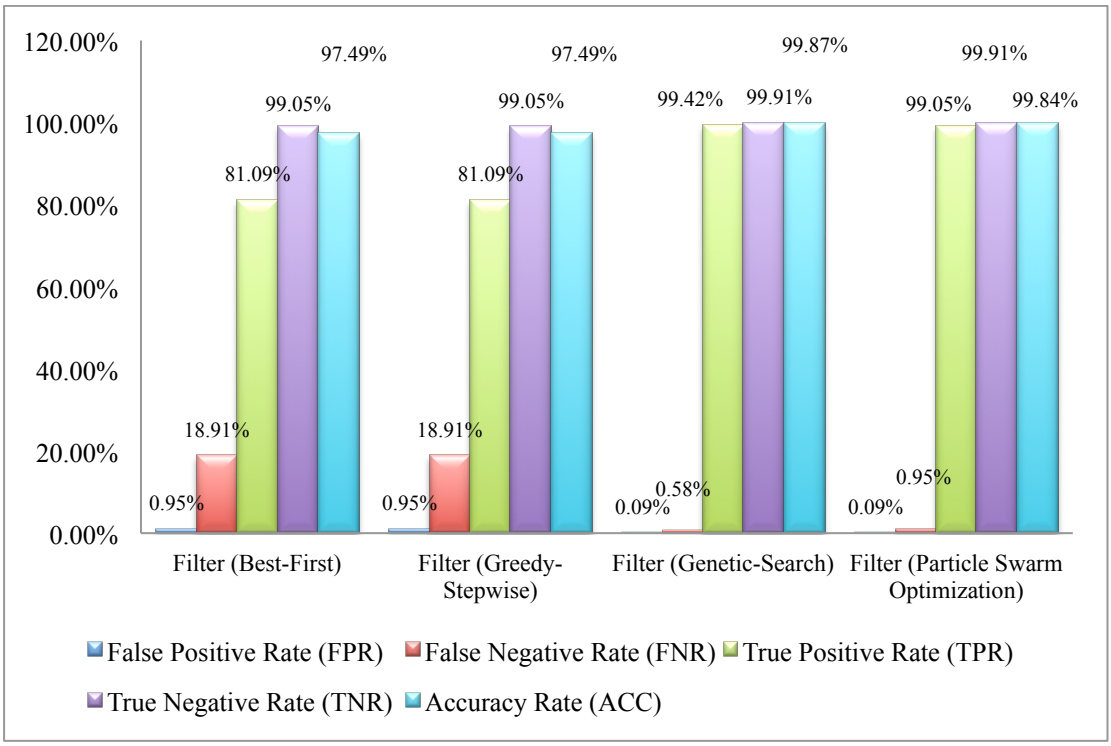


Figure A.2: Comparison of Filter Approaches on DARPA 1999 Dataset

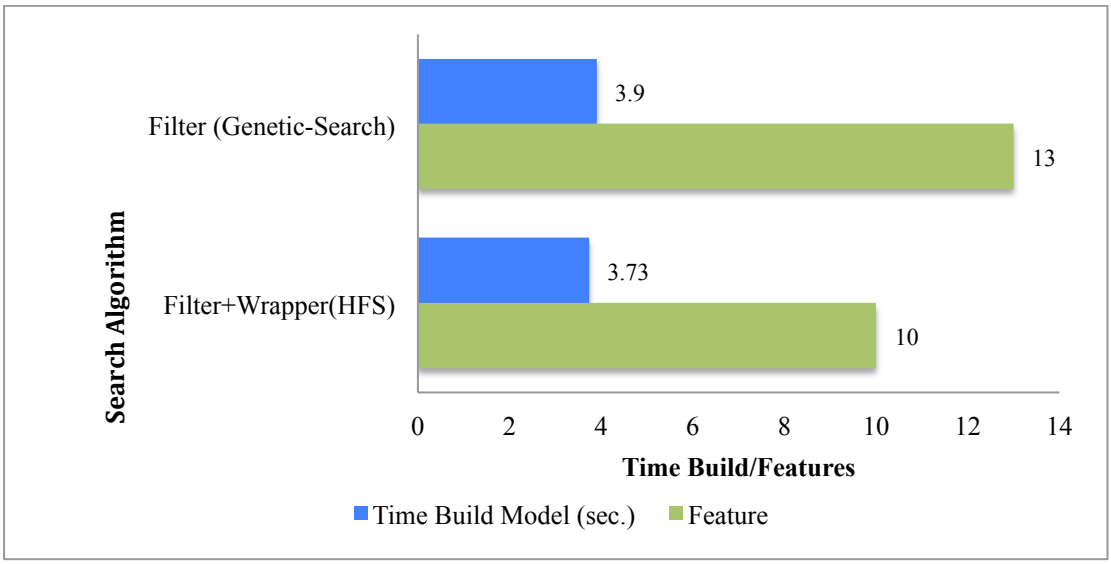


Figure A.3: Building Time of Feature Selection Methods on DARPA 1999 Dataset

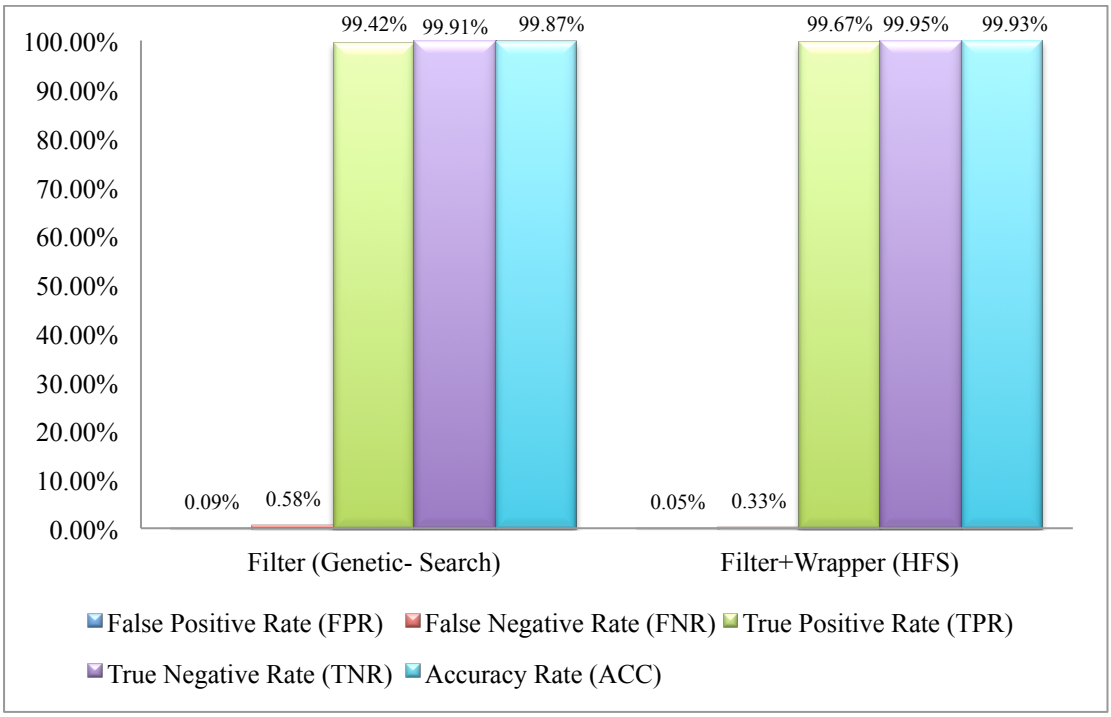


Figure A.4 Comparison of Feature Selection Methods on DARPA 1999 Dataset

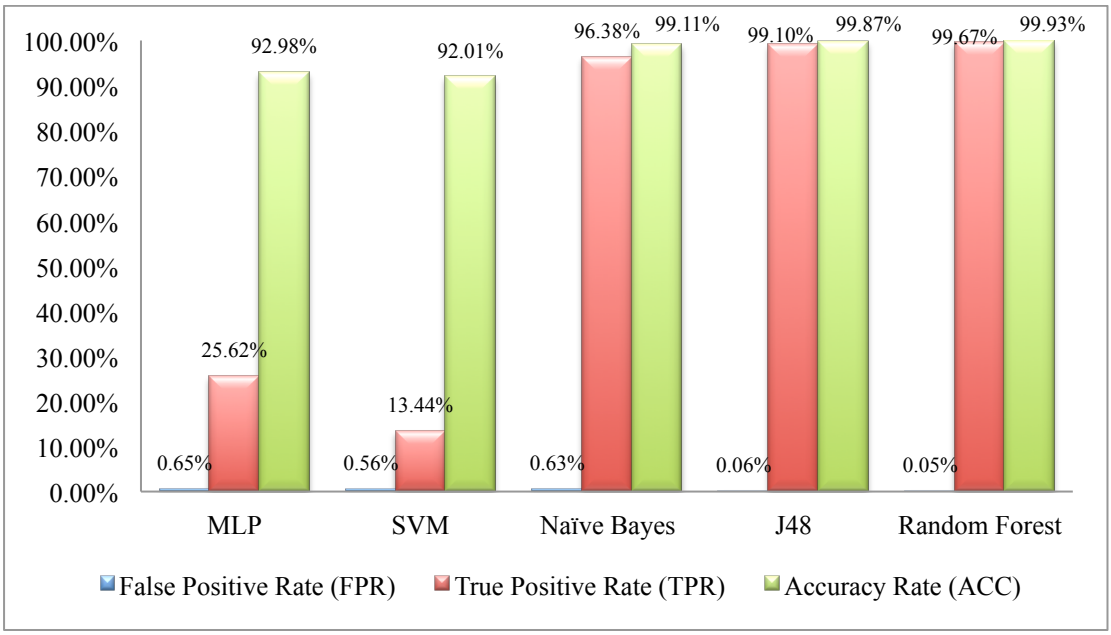


Figure A.5: Comparison of Classification Algorithms on DARPA 1999 Dataset

A.1.1.2 NSL KDD Dataset

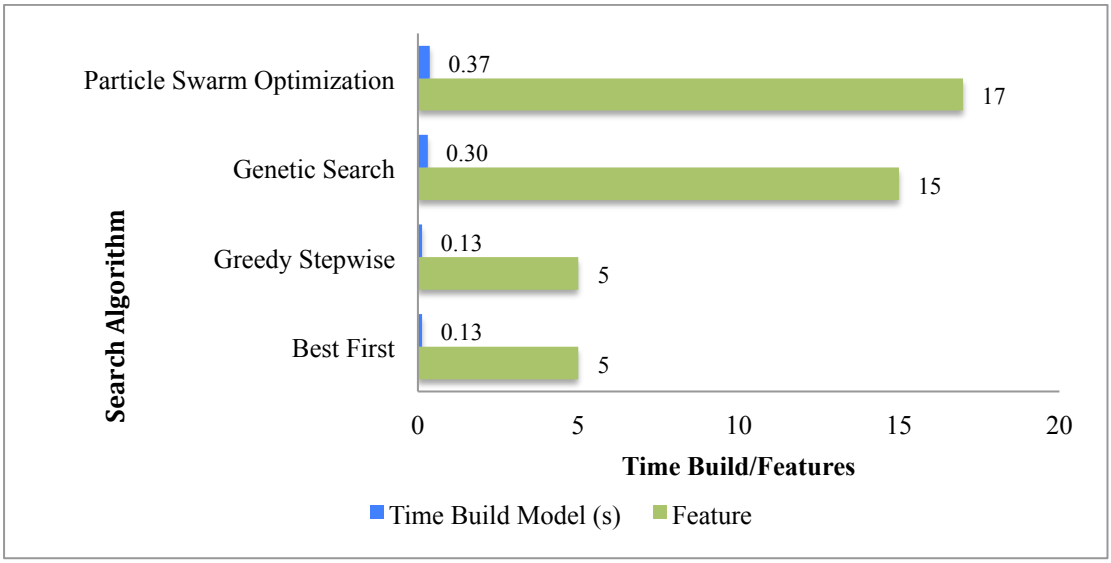


Figure A.6: Building Time and Feature Selected by Search Algorithm with NSL KDD Dataset

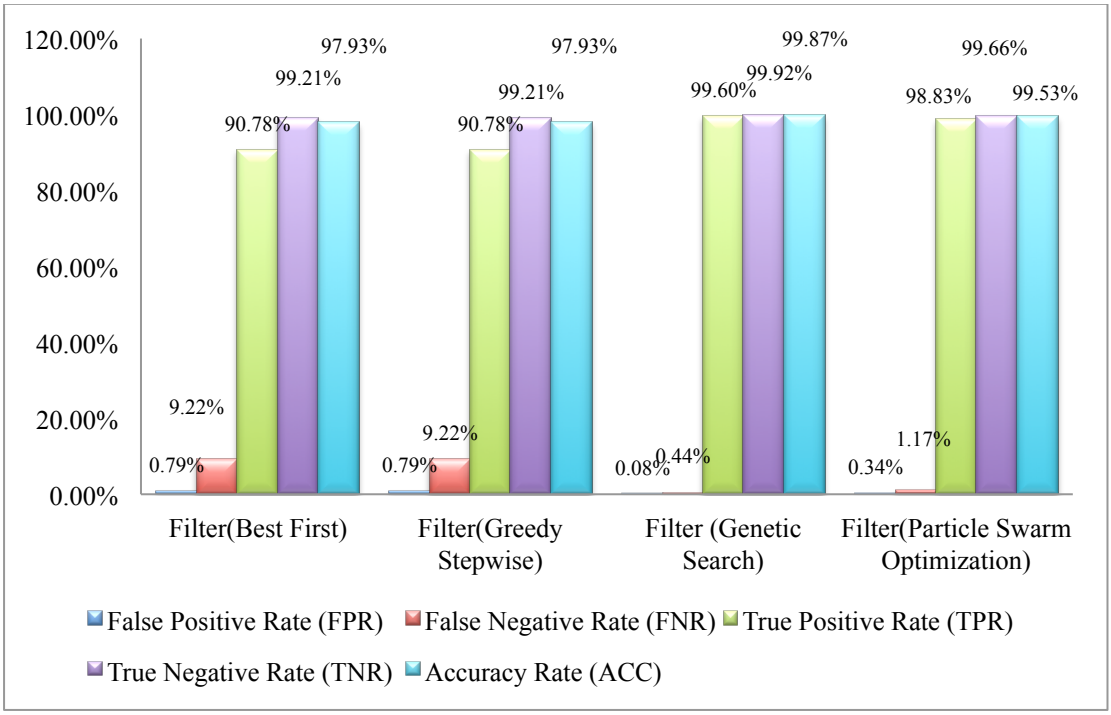


Figure A.7: Comparison of Filter Approaches on NSL KDD Dataset

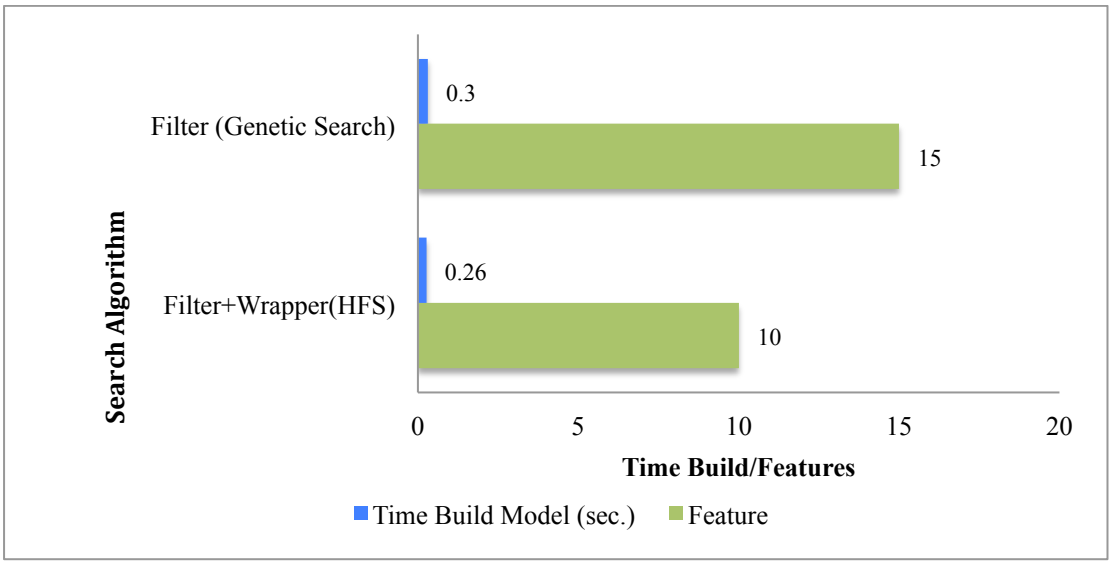


Figure A.8: Building Time of Feature Selection Methods on NSL KDD Dataset

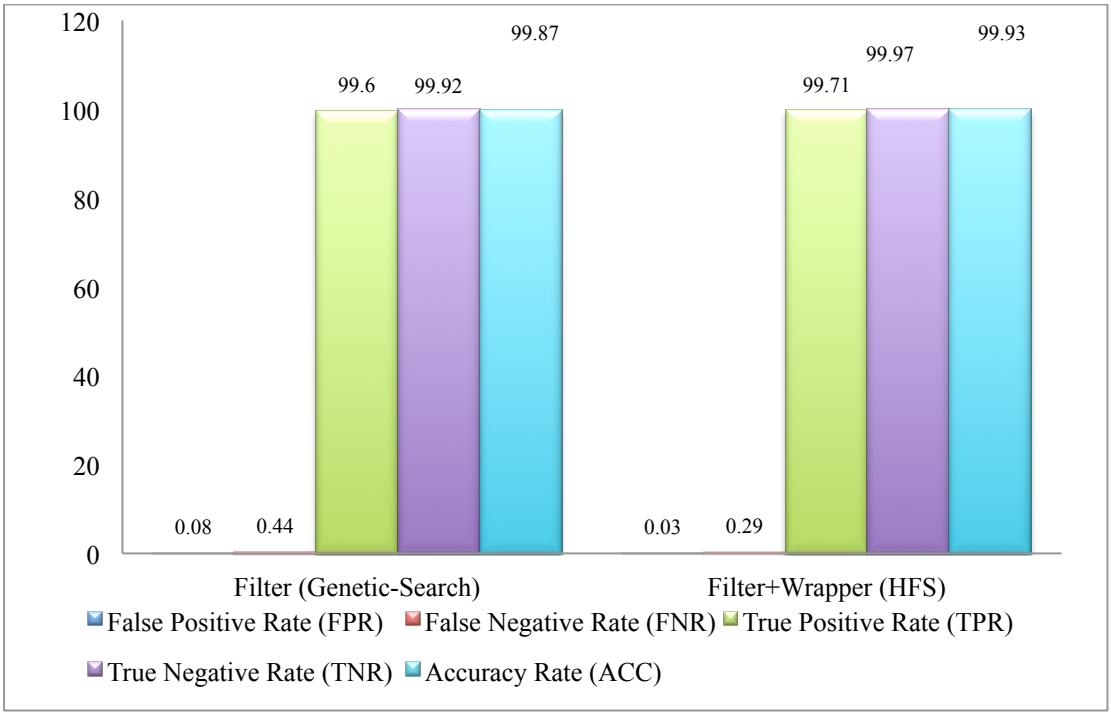


Figure A.9 Comparison of Feature Selection Methods on NSL KDD Dataset

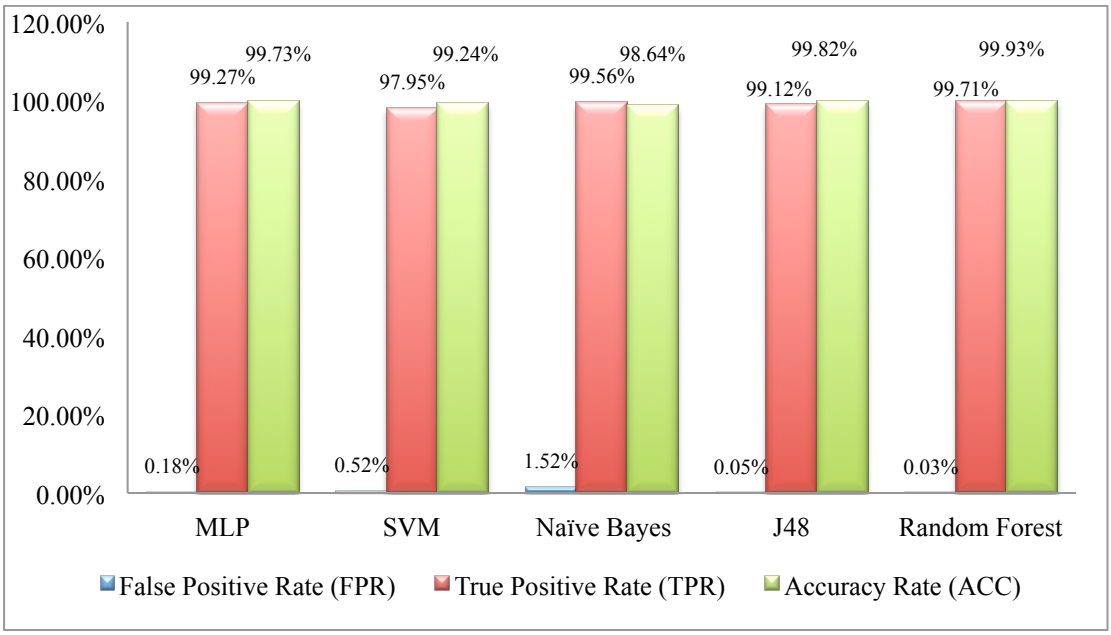


Figure A.10: Comparison of Classification Algorithms on NSL KDD Dataset

A.1.1.3 ISCX 2012 Dataset

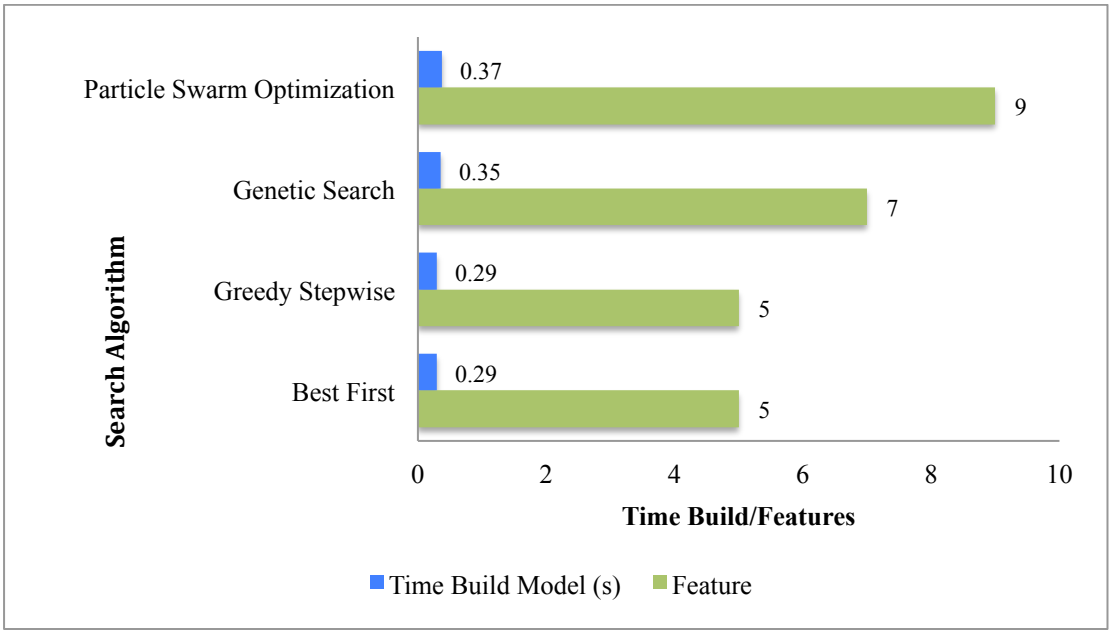


Figure A.11: Building Time and Feature Selected by Search Algorithm with ISCX 2012 Dataset

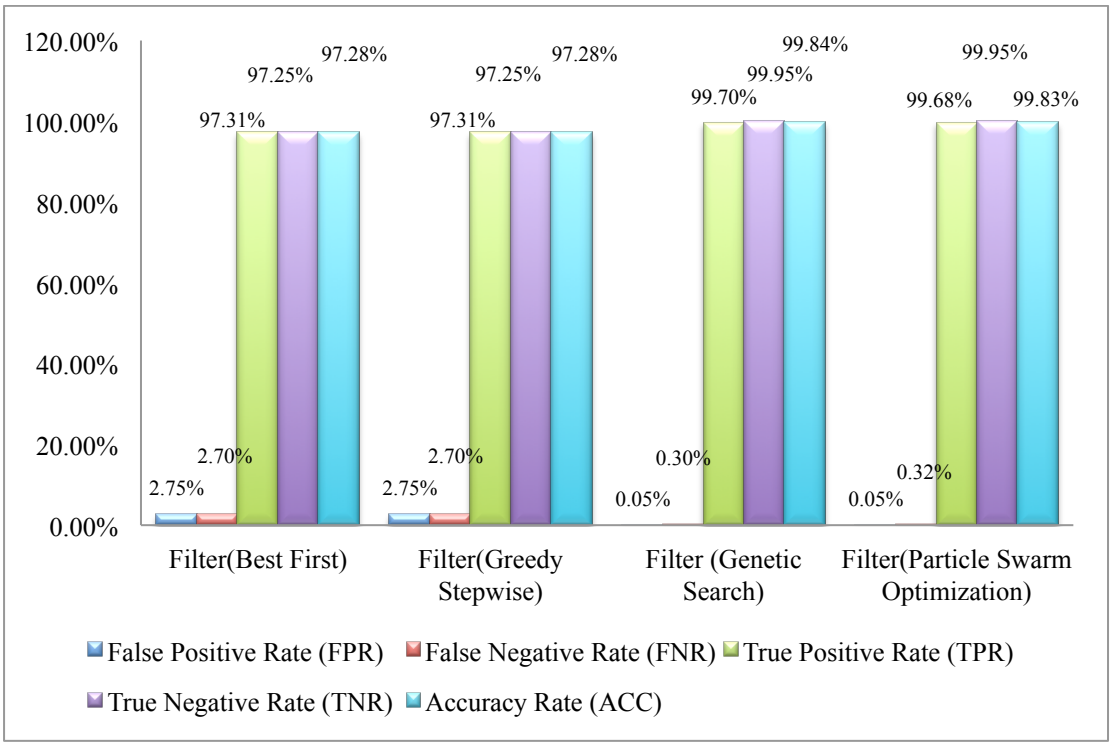


Figure A.12: Comparison of Filter Approaches on ISCX 2012 Dataset

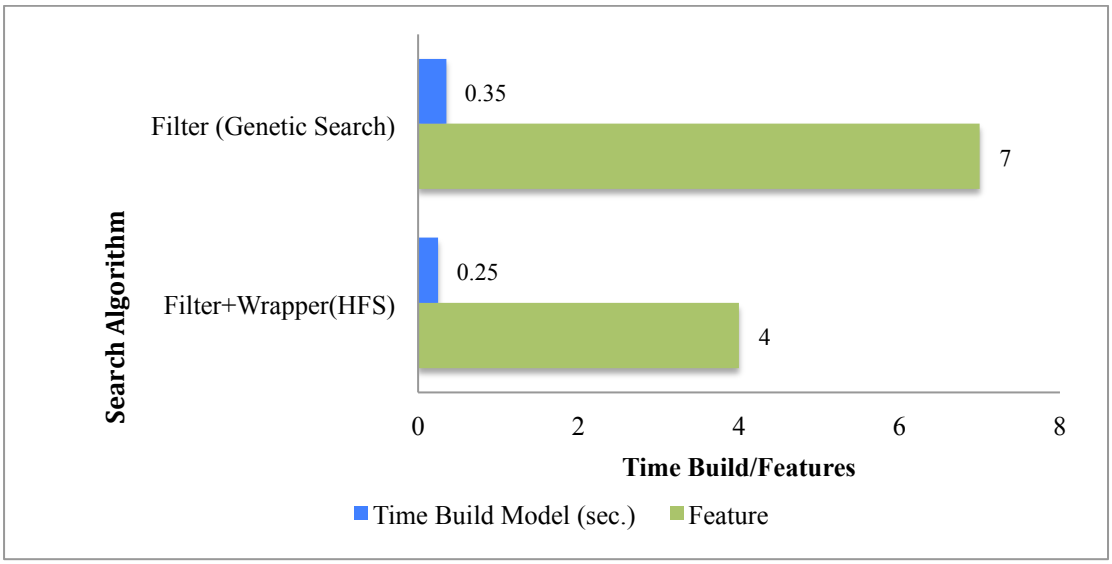


Figure A.13: Building Time of Feature Selection Methods on ISCX 2012 Dataset

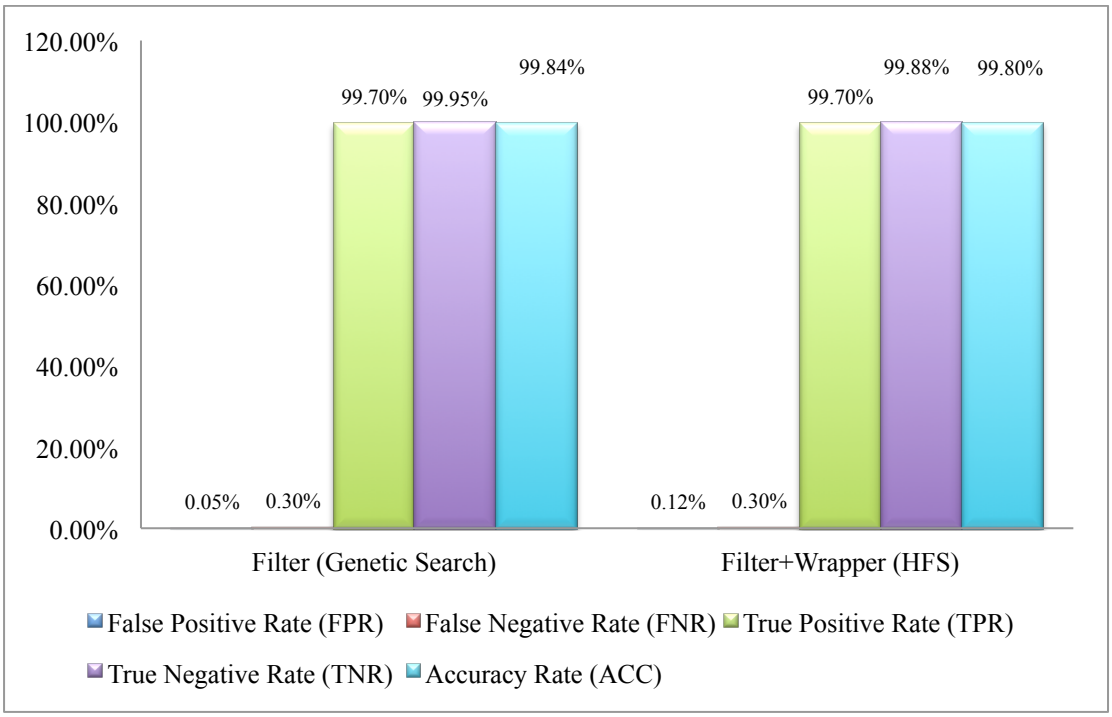


Figure A.14: Comparison of Feature Selection Methods on ISCX 2012 Dataset

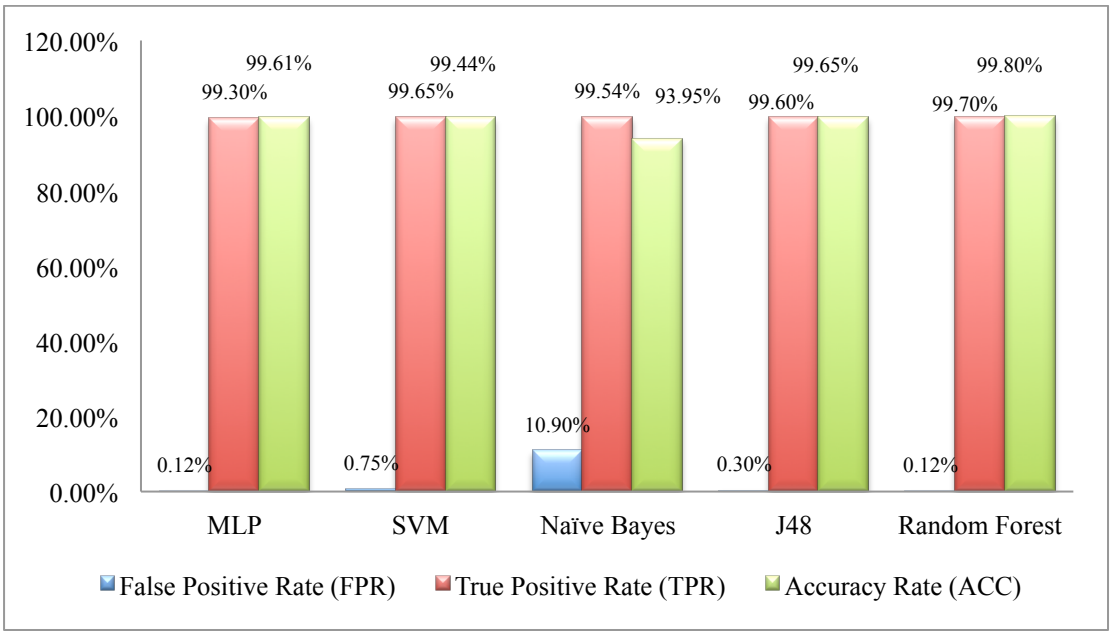


Figure A.15: Comparison of Classification Algorithms on ISCX 2012 Dataset

A.1.1.4 UNSW-NB15 Dataset

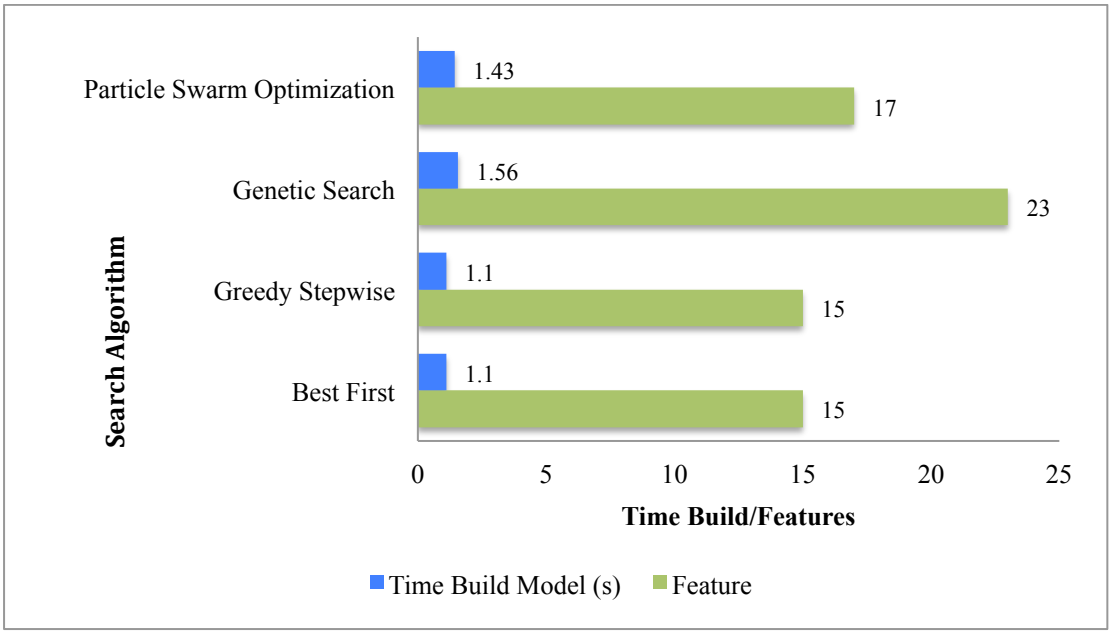


Figure A.16: Building Time and Feature Selected by Search Algorithm with UNSW-NB15 Dataset

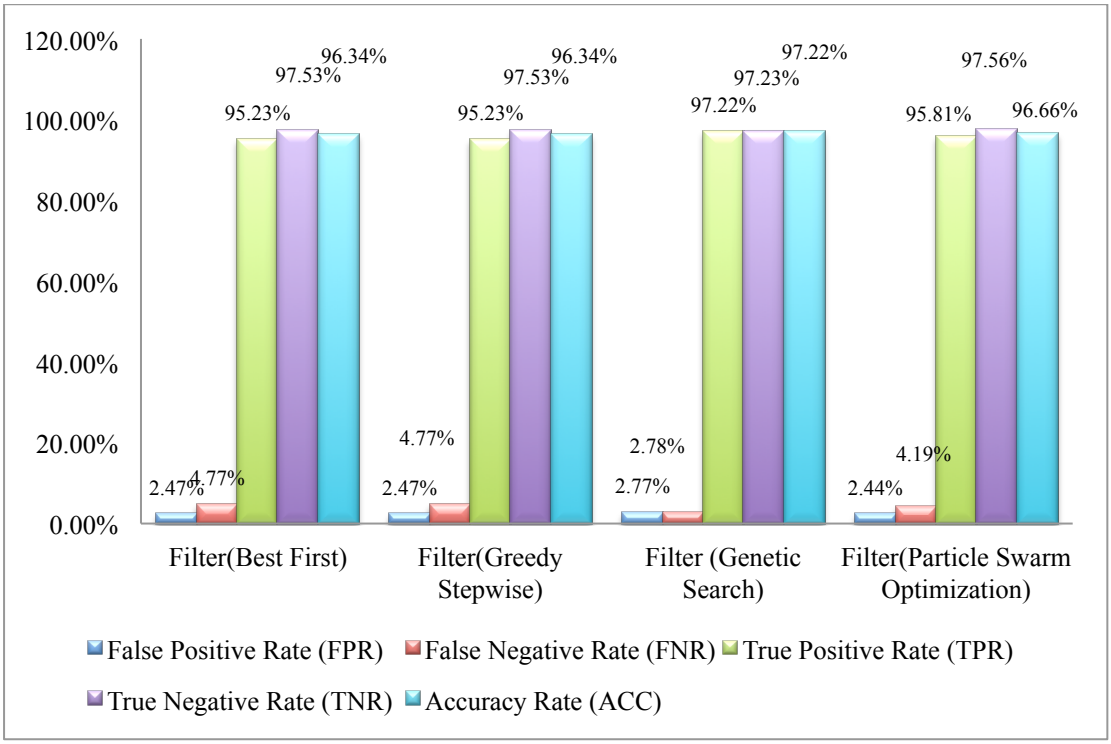


Figure A.17: Comparison of Filter Approaches on UNSW-NB15 Dataset

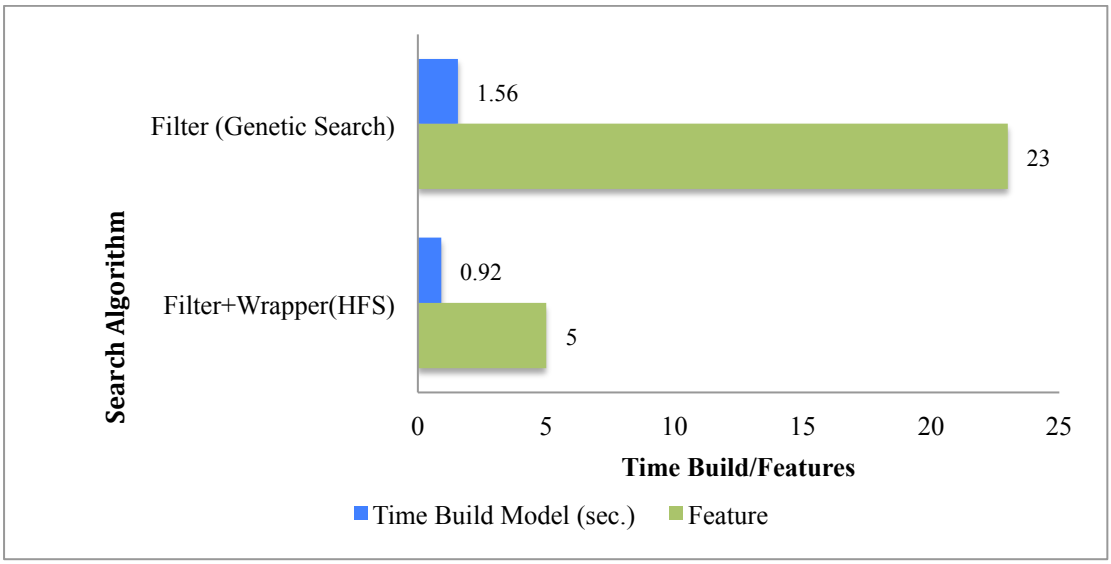


Figure A.18: Building Time of Feature Selection Methods on UNSW-NB15 Dataset

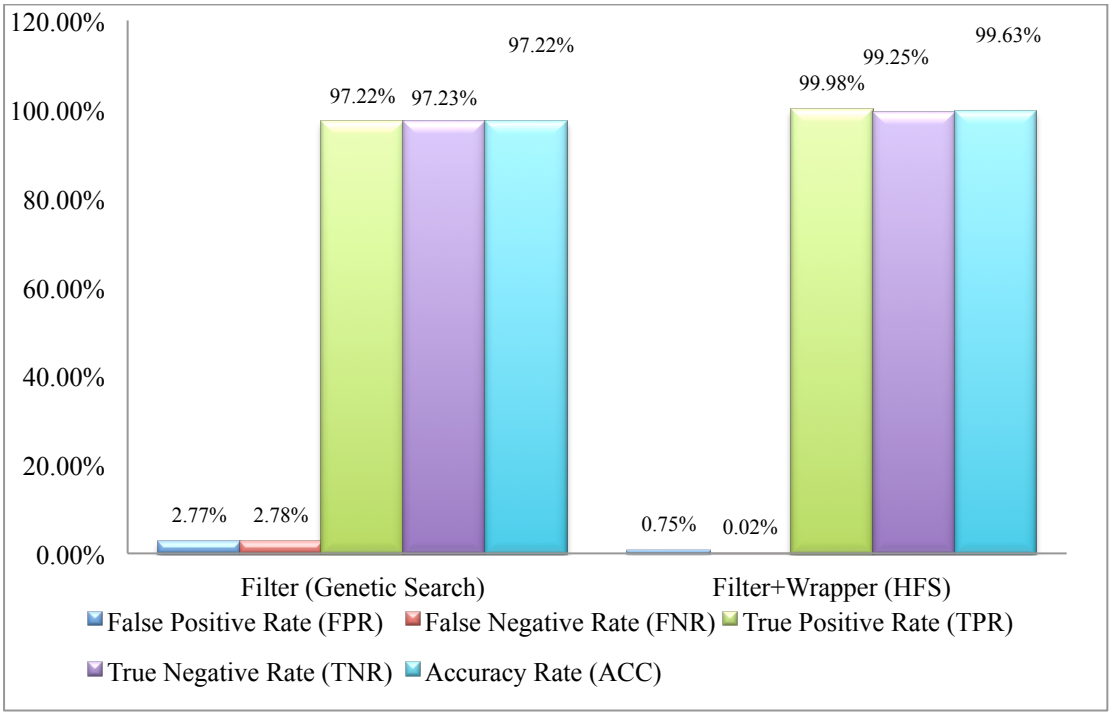


Figure A.19: Comparison of Feature Selection Methods on UNSW-NB15 Dataset

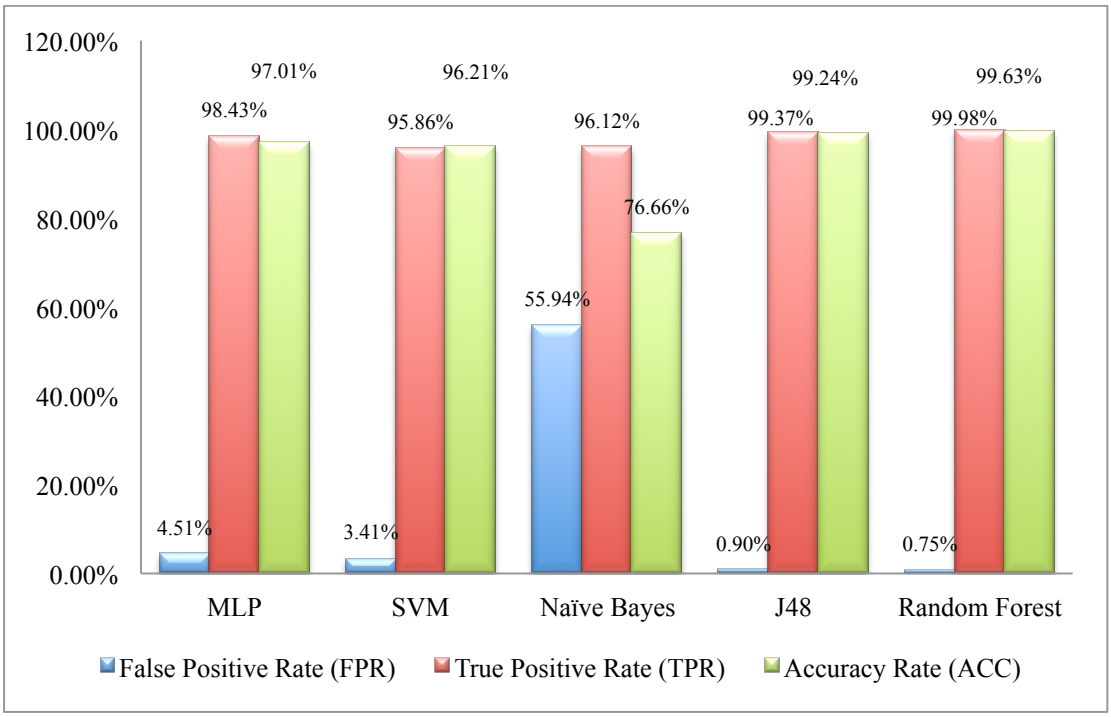


Figure A.20: Comparison of Classification Algorithms on UNSW-NB15 Dataset

A.2 Second Preliminary Experiment

A.2.1.1 DARPA 1999 Dataset

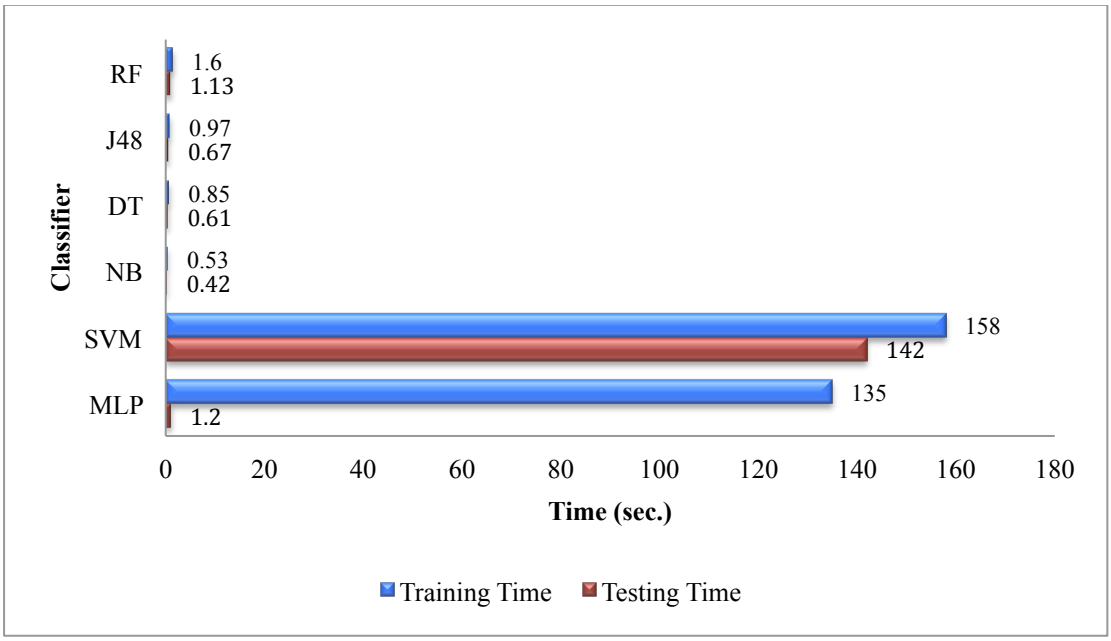


Figure A.21: Building Model and Detection Time Taken by Single Classifier Using DARPA 1999 Dataset

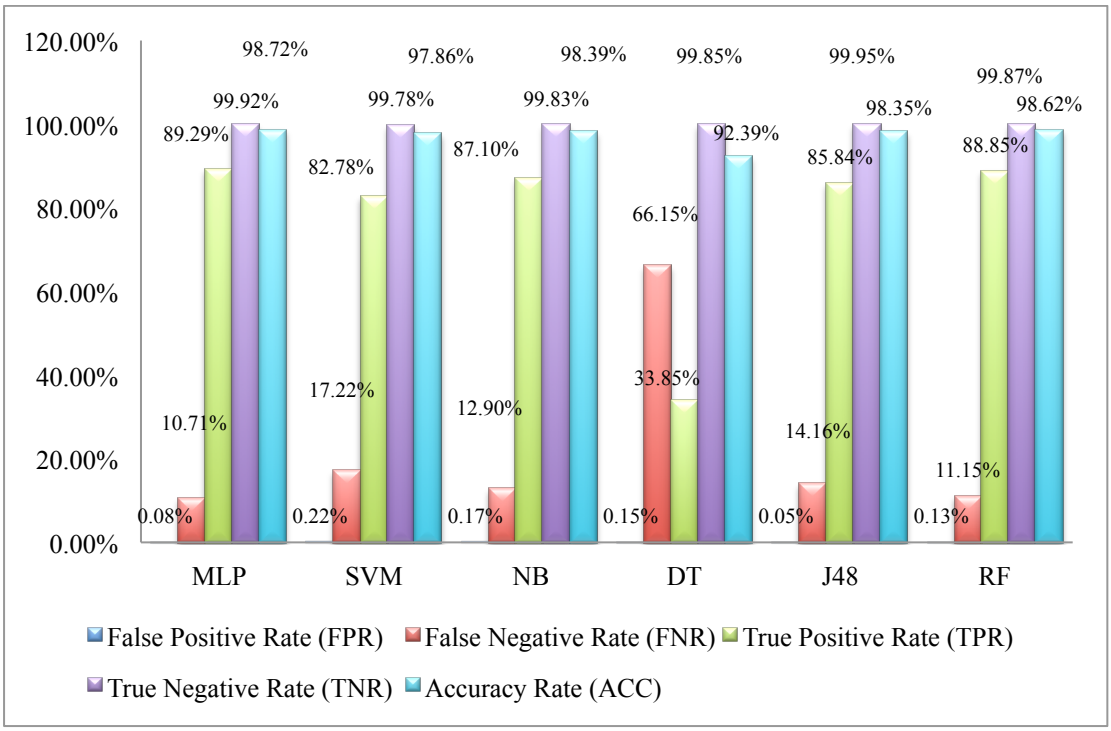


Figure A.22: Comparison Performances of Single Classifier Using DARPA 1999 Dataset

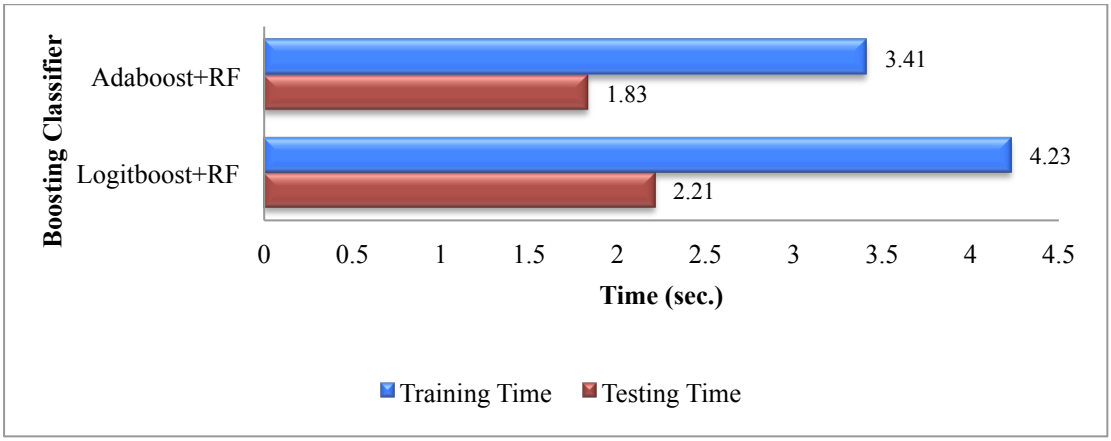


Figure A.23: Building Model and Detection Time Taken by Boosting Classifiers Using DARPA 1999 Dataset

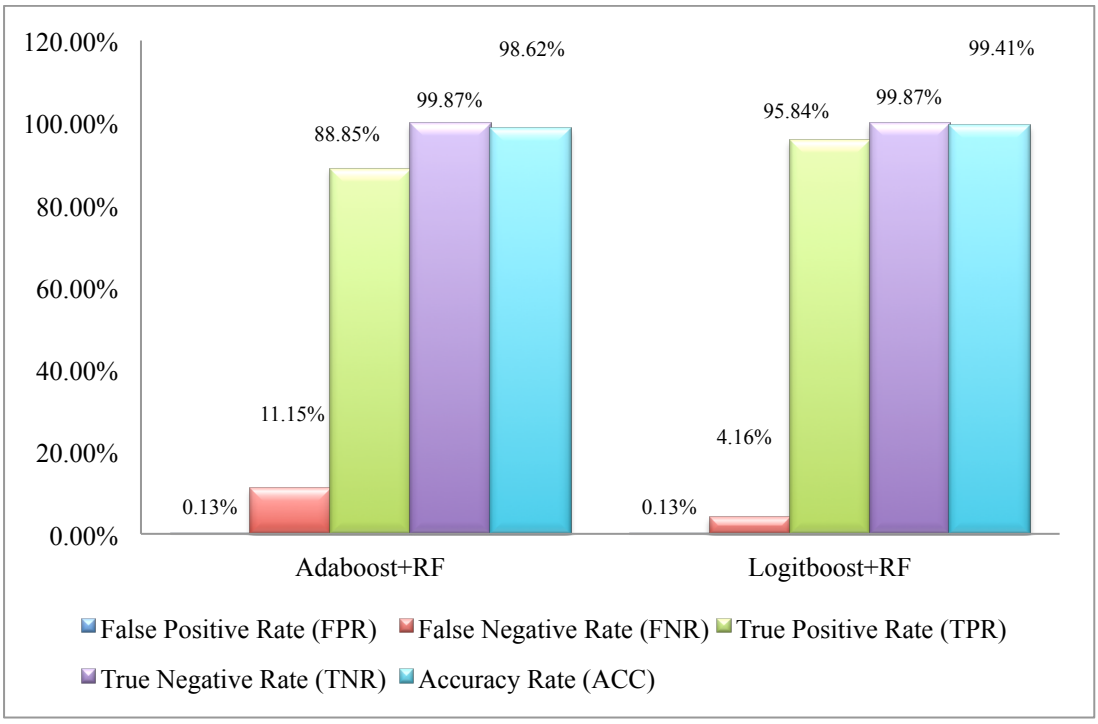


Figure A.24: Comparison Performances of Boosting Classifiers Using DARPA 1999 Dataset

A.2.1.2 NSL KDD Dataset

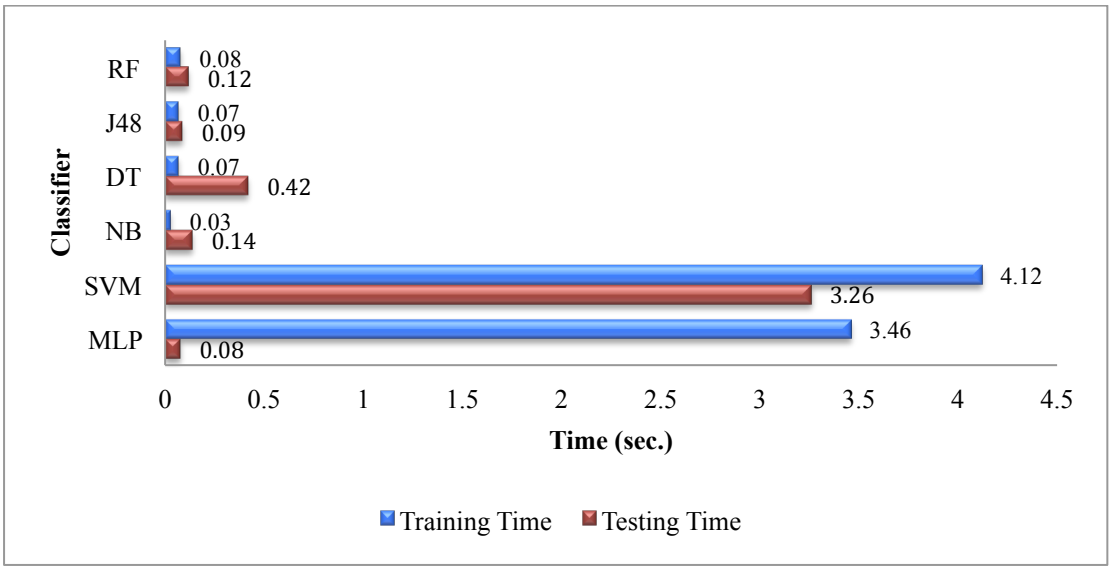


Figure A.25: Building Model and Detection Time Taken by Single Classifier Using NSL KDD Dataset

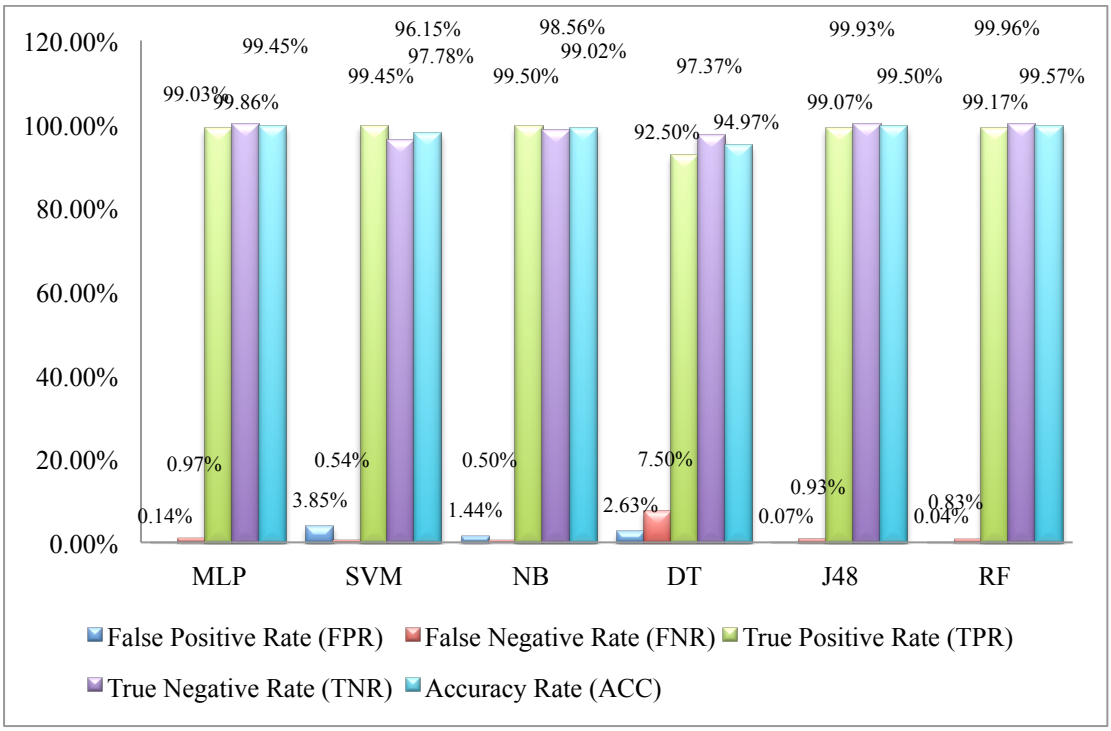


Figure A.26: Comparison Performances of Single Classifier Using NSL KDD Dataset

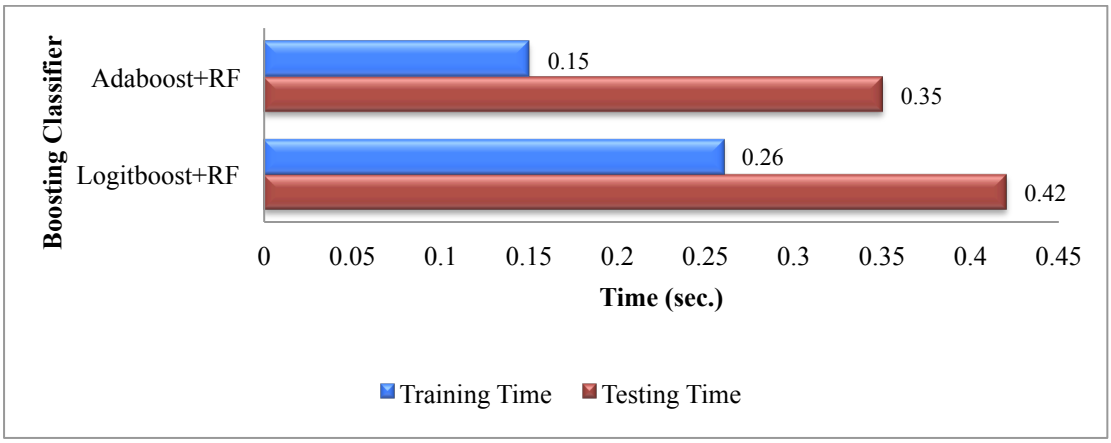


Figure A.27: Building Model and Detection Time Taken by Boosting Classifiers Using NSL KDD Dataset

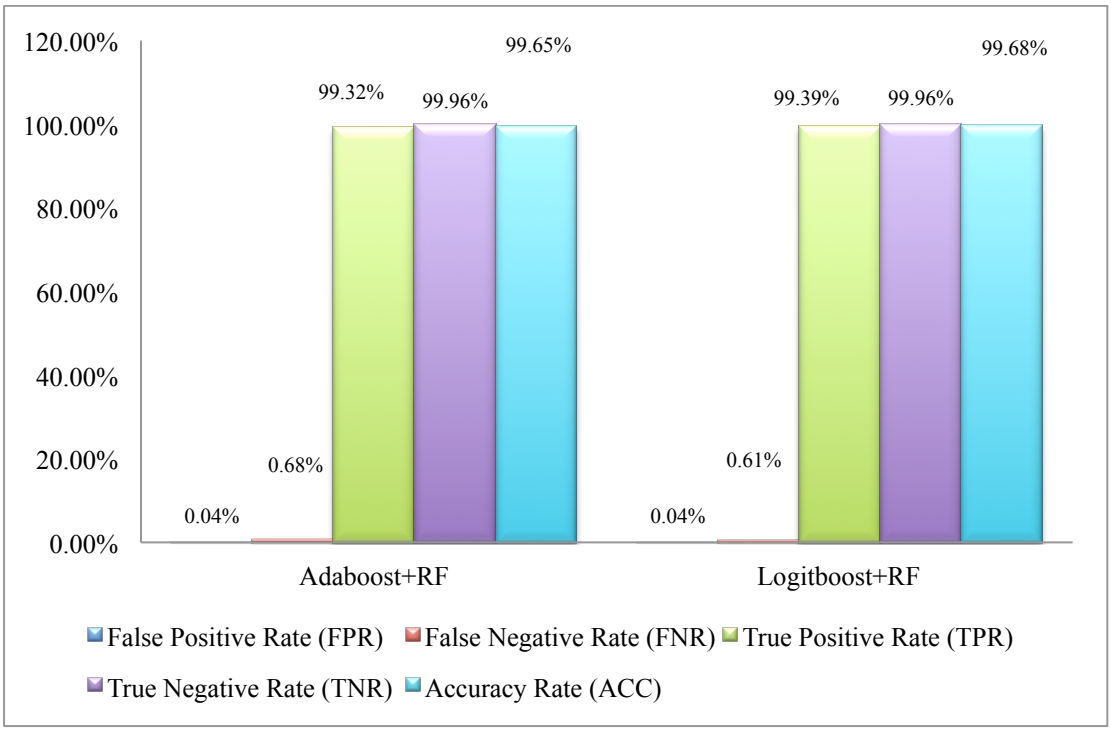


Figure A.28: Comparison Performances of Boosting Classifiers Using NSL KDD Dataset

A.2.1.3 ISCX 2012 Dataset

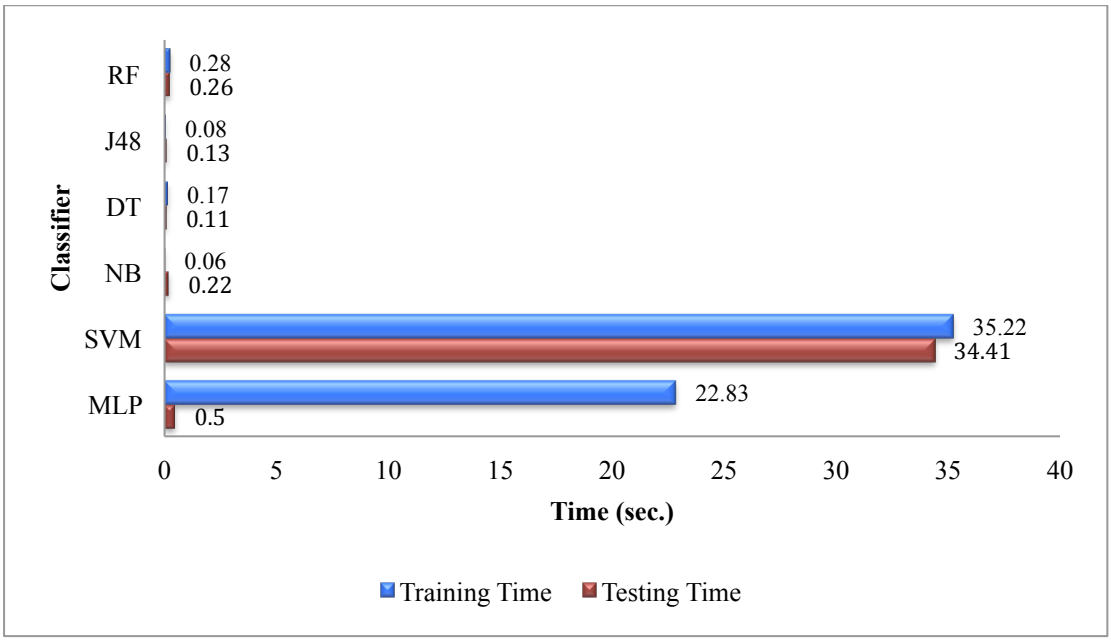


Figure A.29: Building Model and Detection Time Taken by Single Classifier Using ISCX 2012 Dataset

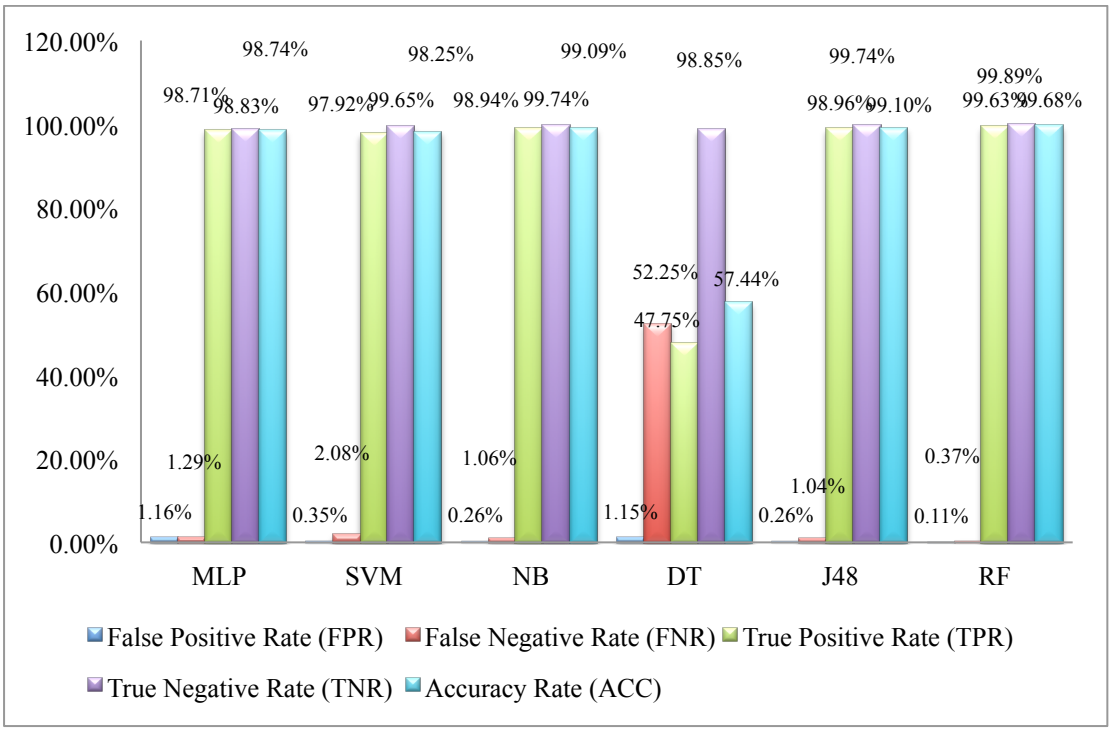


Figure A.30: Comparison Performances of Single Classifier Using ISCX 2012 Dataset

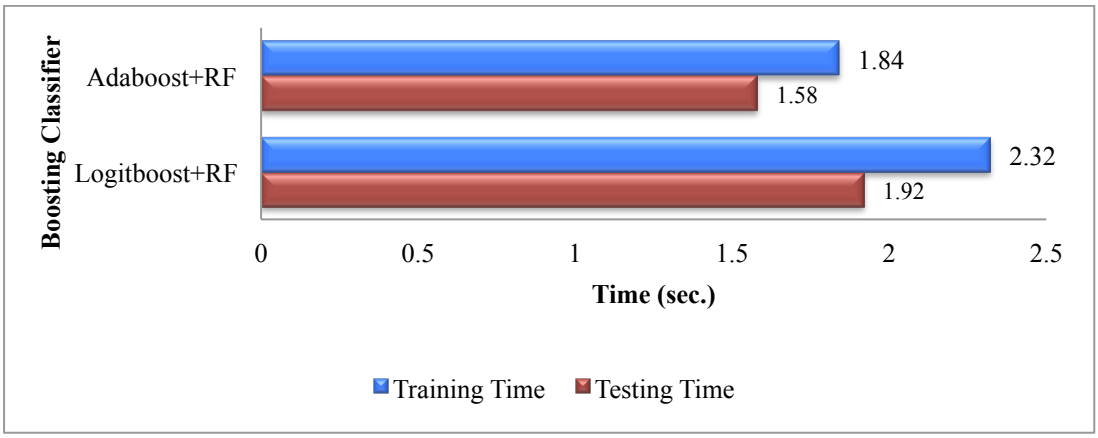


Figure A.31: Building Model and Detection Time Taken by Boosting Classifiers Using ISCX 2012 Dataset

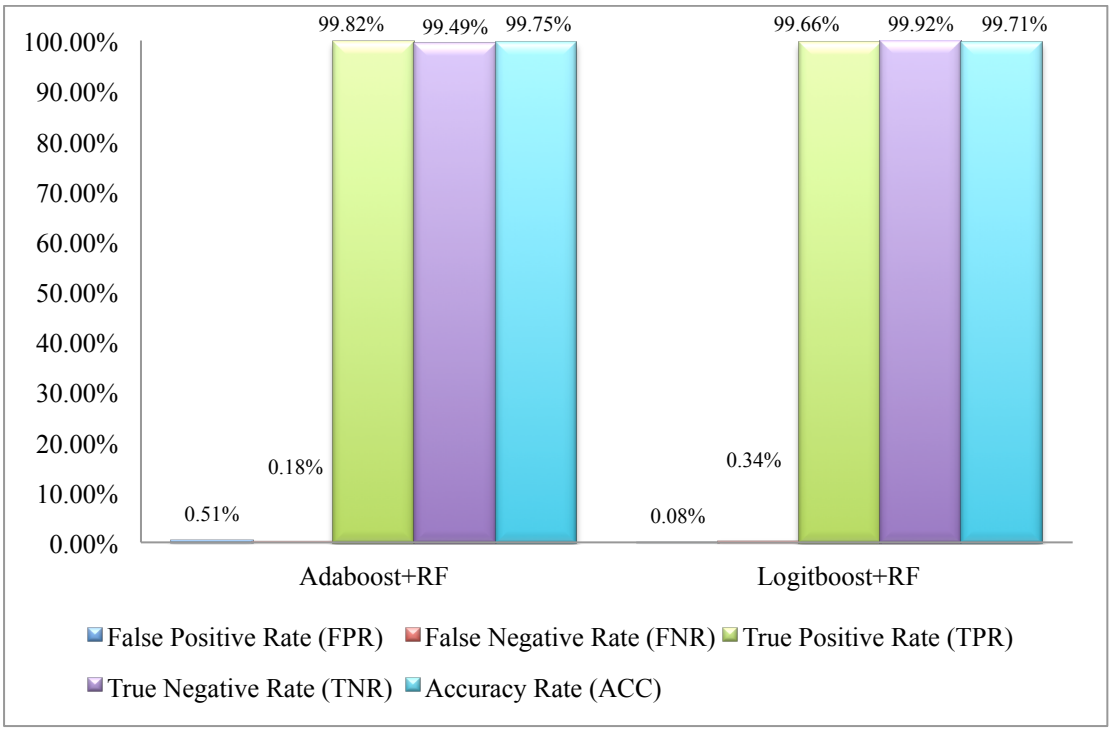


Figure A.32: Comparison Performances of Boosting Classifiers Using ISCX 2012 Dataset

A.2.1.4 UNSW-NB15 Dataset

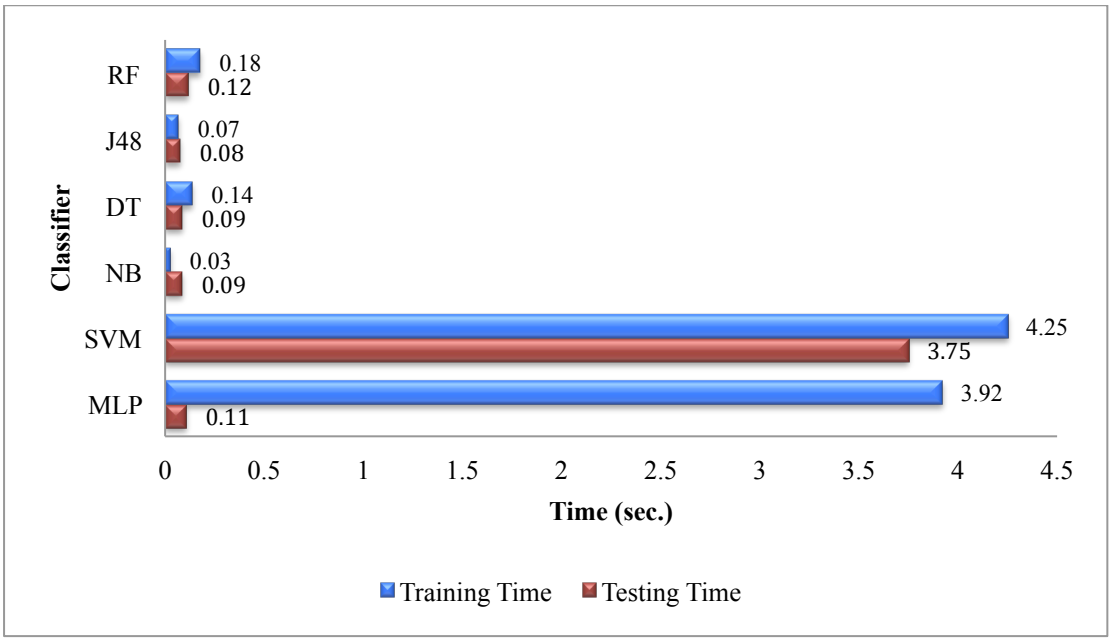


Figure A.33: Building Model and Detection Time Taken by Single Classifier Using UNSW-NB15 Dataset

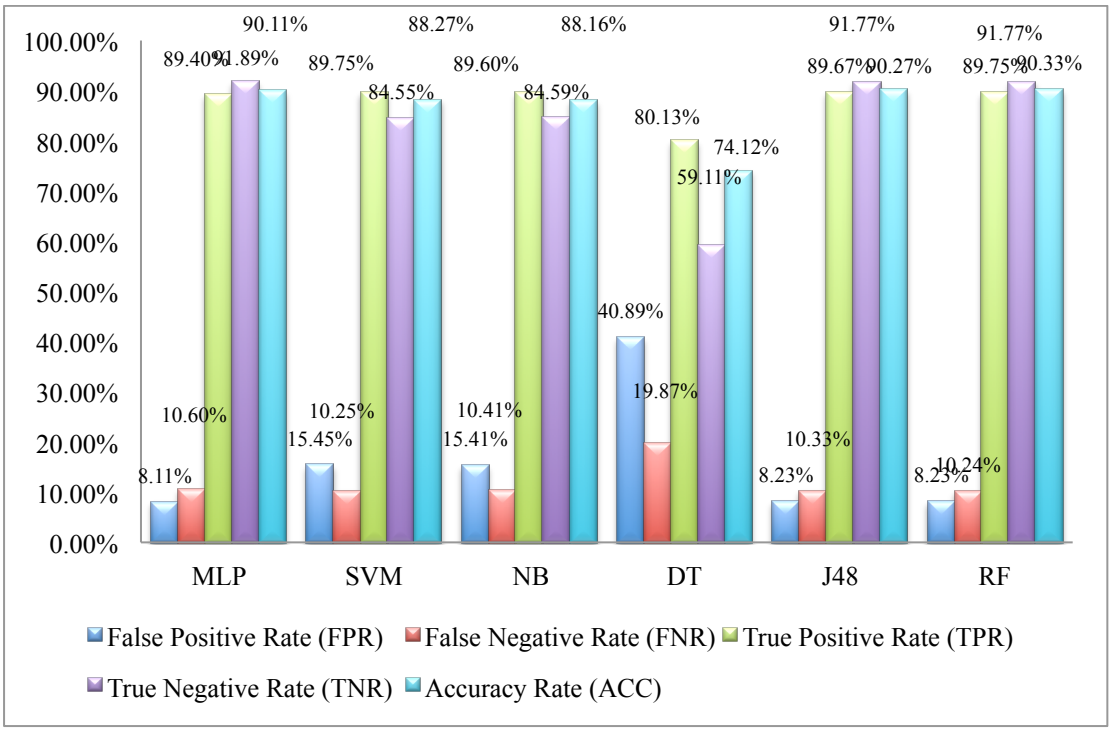


Figure A.34: Comparison Performances of Single Classifier Using UNSW-NB15 Dataset

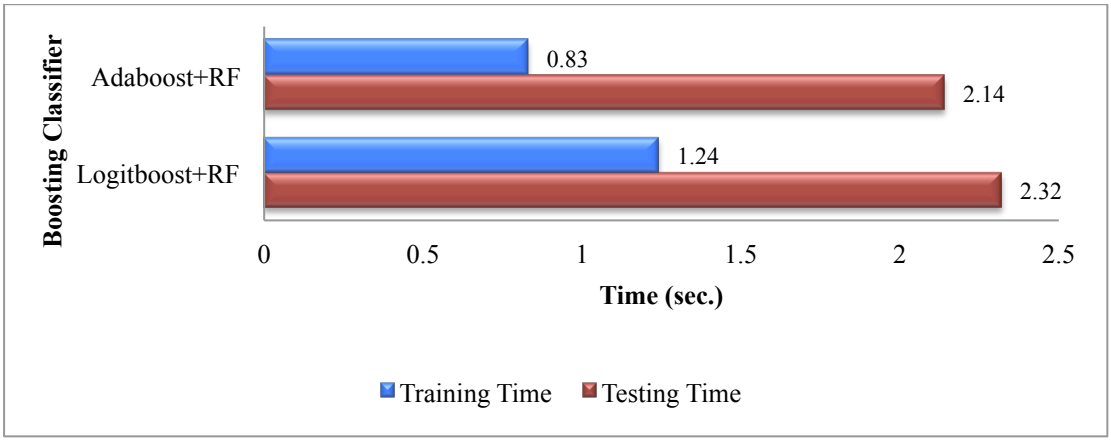


Figure A.35: Building Model and Detection Time Taken by Boosting Classifiers Using UNSW-NB15 Dataset

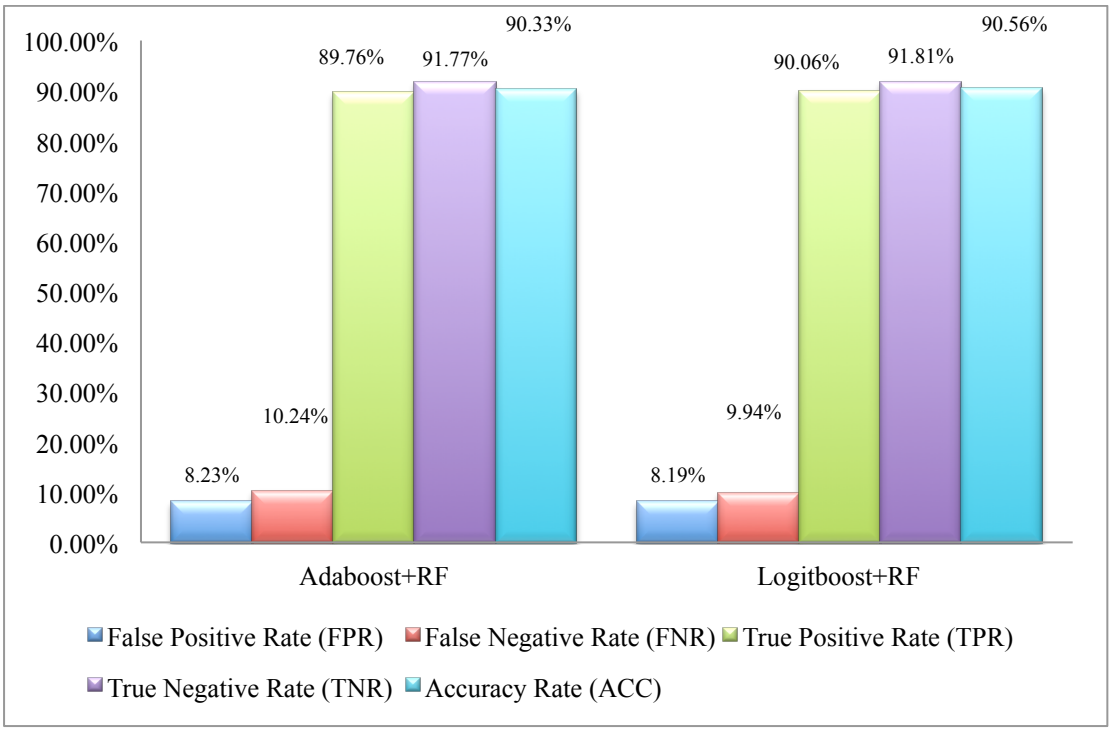


Figure A.36: Comparison Performances of Boosting Classifiers Using UNSW-NB15 Dataset

Appendix B

B.1 Performance Evaluation on DARPA 1999

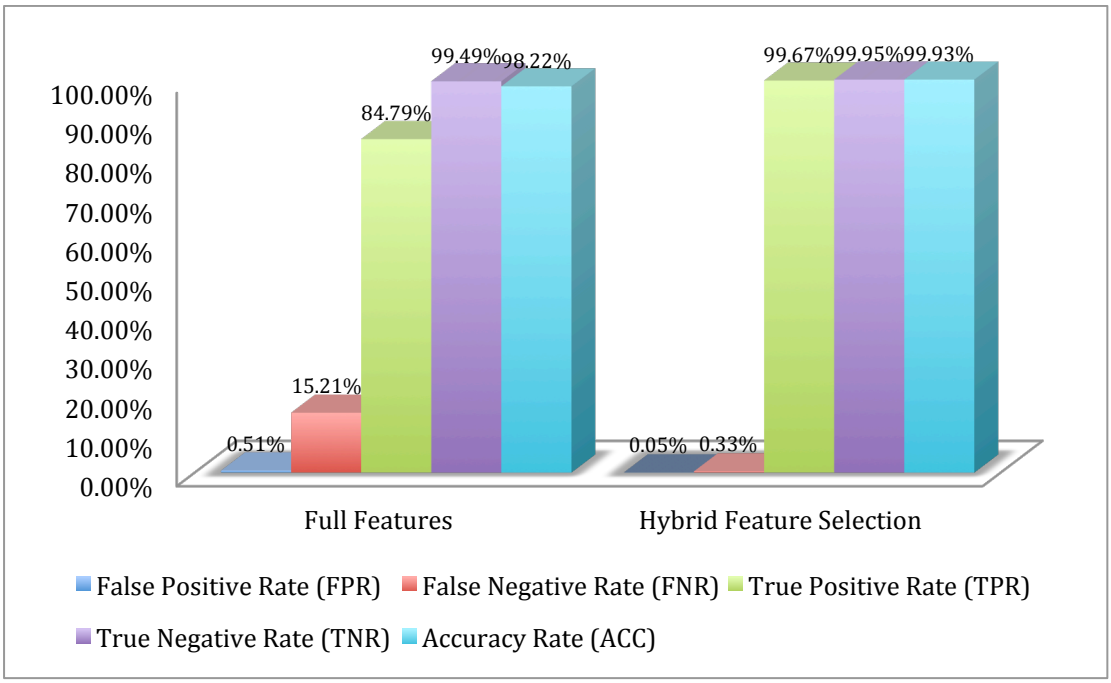


Figure B.1: Performance of HFS Using DARPA 1999 Dataset

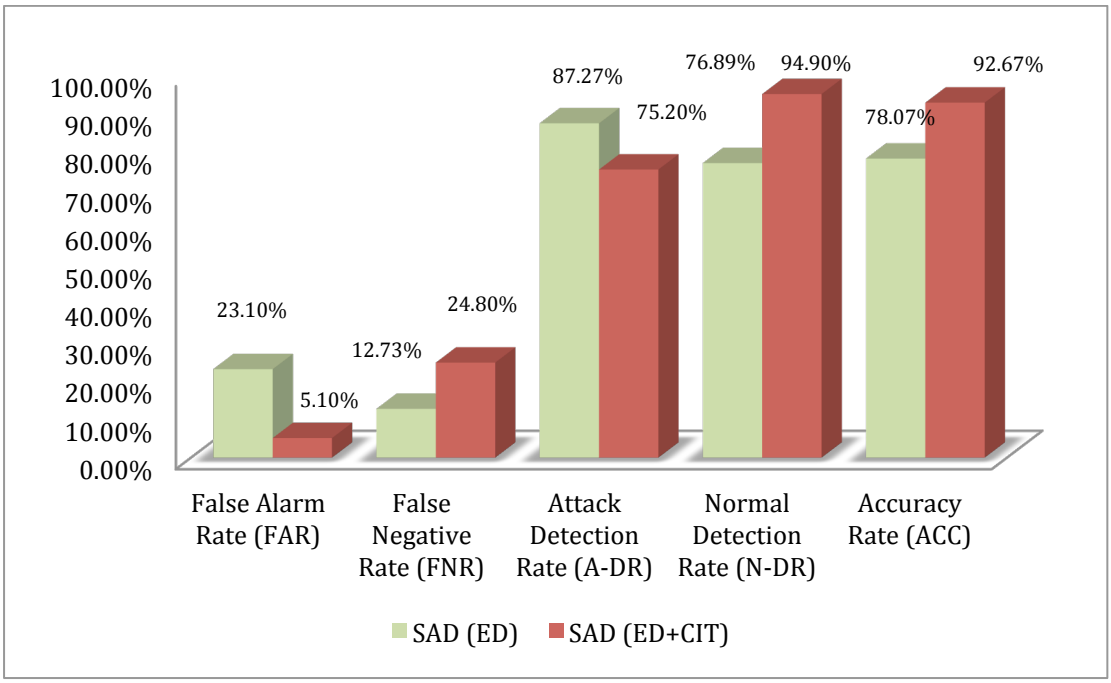


Figure B.2: Performance of Statistical Analysis Detection on DARPA 1999 Dataset

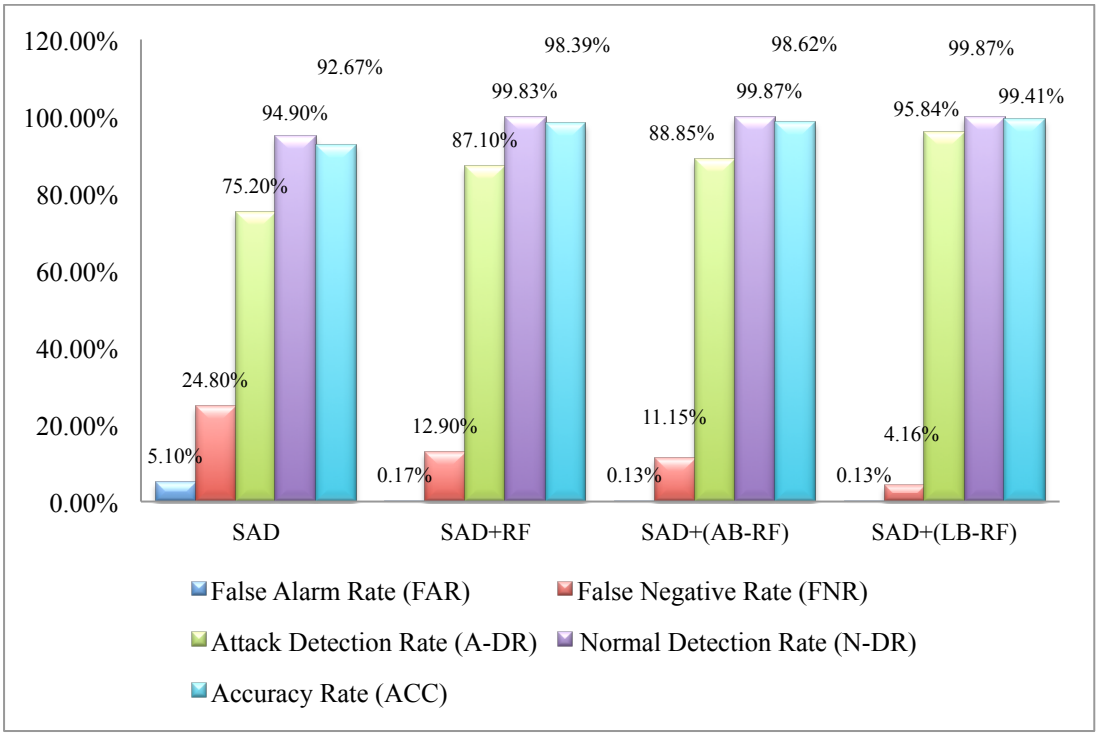


Figure B.3: Performance of proposed approaches on DARPA 1999 Dataset

B.2 Performance Evaluation on NSL KDD

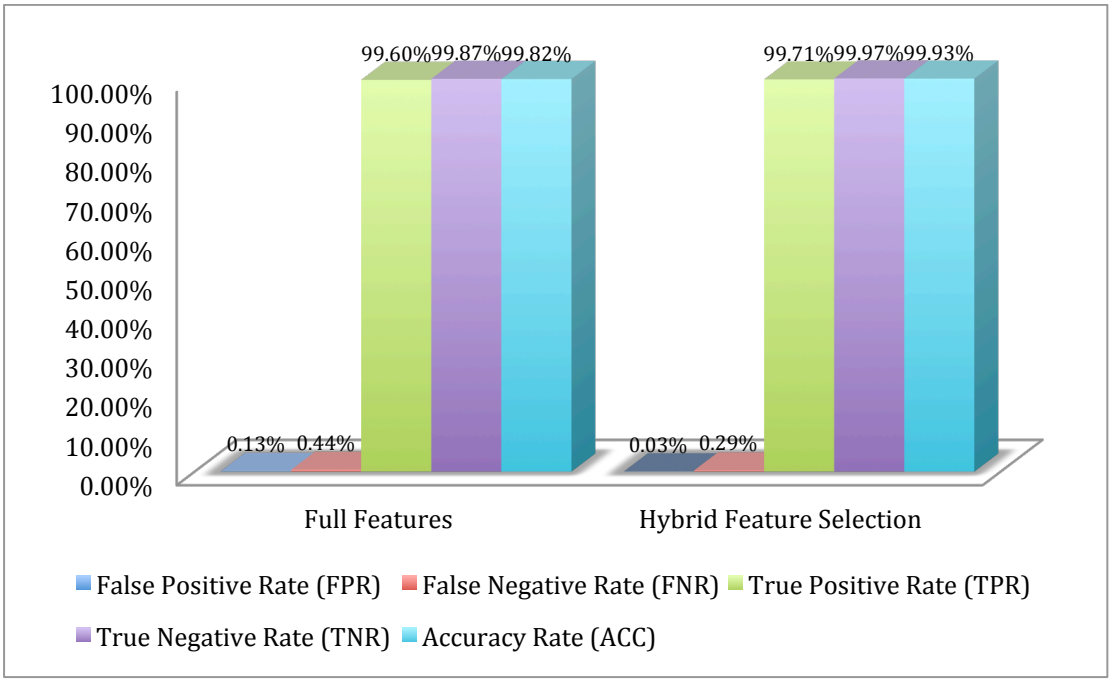


Figure B.4: Performance of HFS Using NSL KDD Dataset

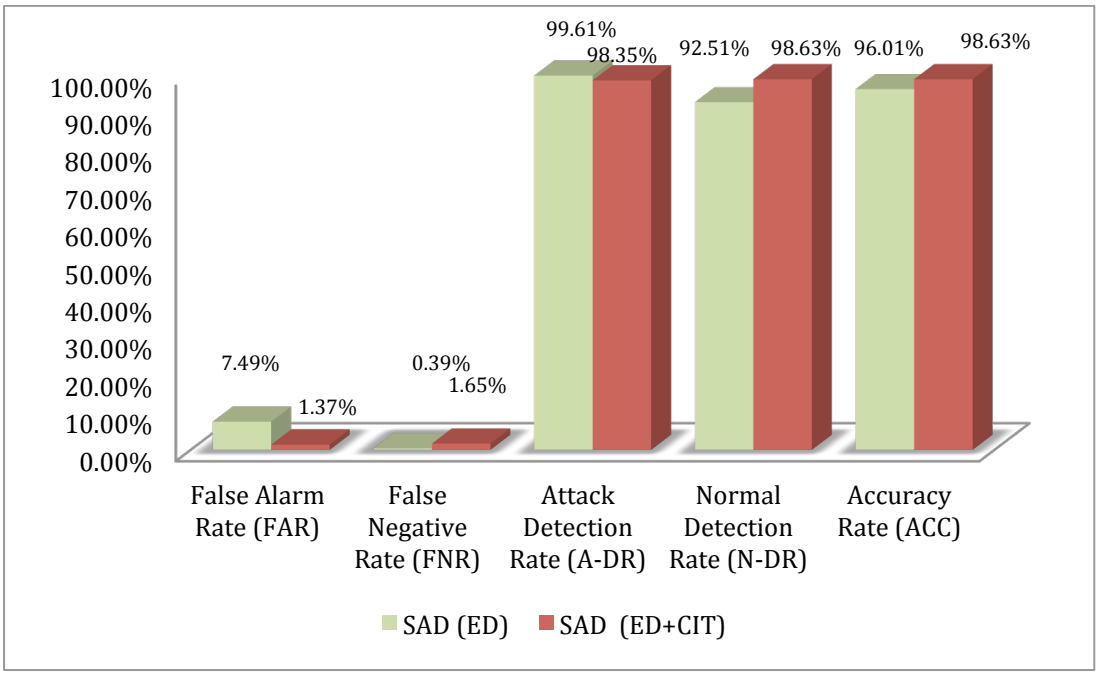


Figure B.5: Performance of Statistical Analysis Detection on NSL KDD Dataset

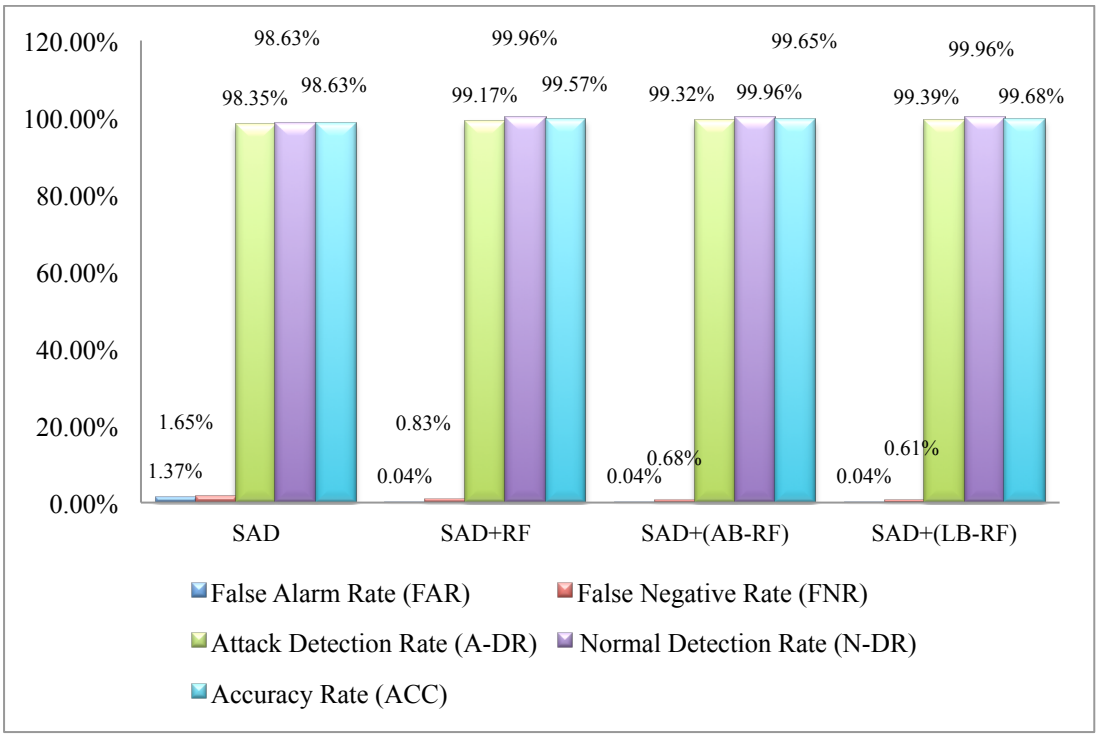


Figure B.6: Performance of Anomaly Detection Approaches on NSL KDD Dataset

B.3 Performance Evaluation on ISCX 2012

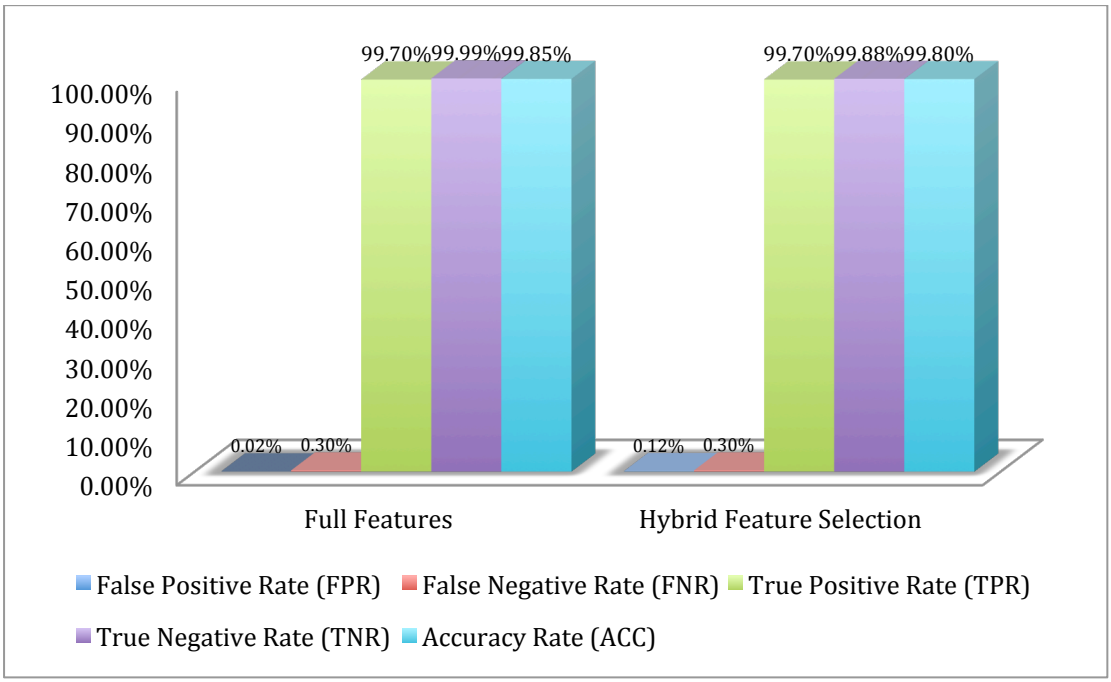


Figure B.7: Performance of HFS Using ISCX 2012

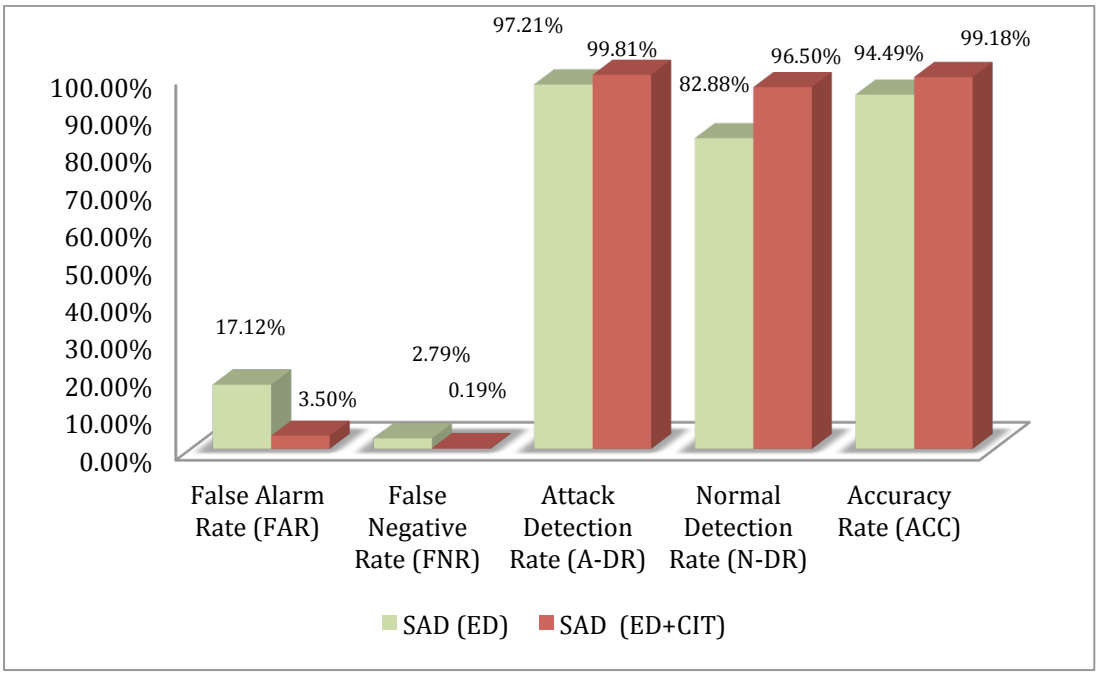


Figure B.8: Performance of Statistical Analysis Detection on ISCX Dataset

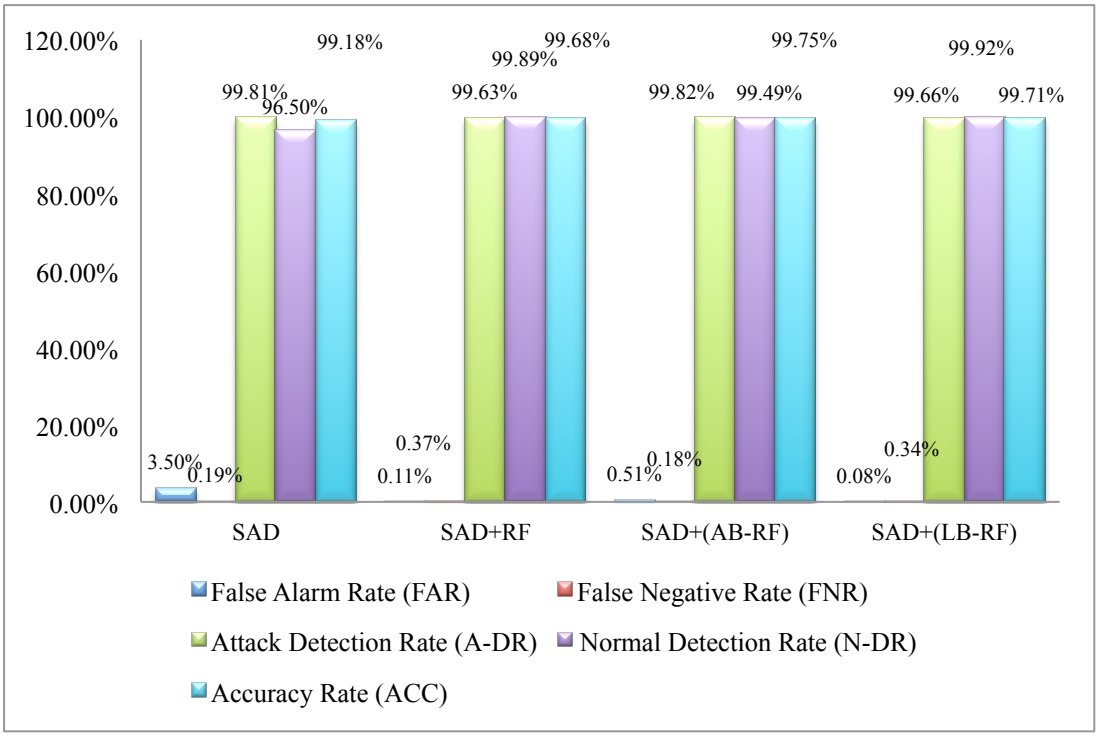


Figure B.9: Performance of Anomaly Detection Approaches on ISCX 2012 Dataset

B.4 Performance Evaluation on UNSW-NB15

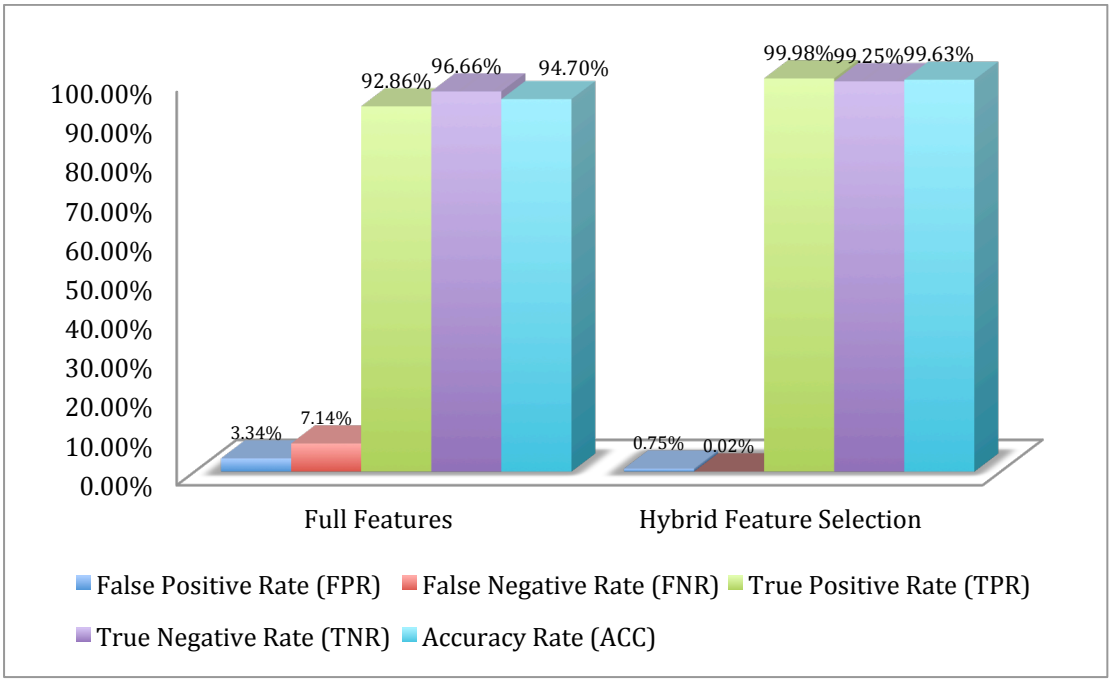


Figure B.10: Performance of HFS Using UNSW-NB15 Dataset

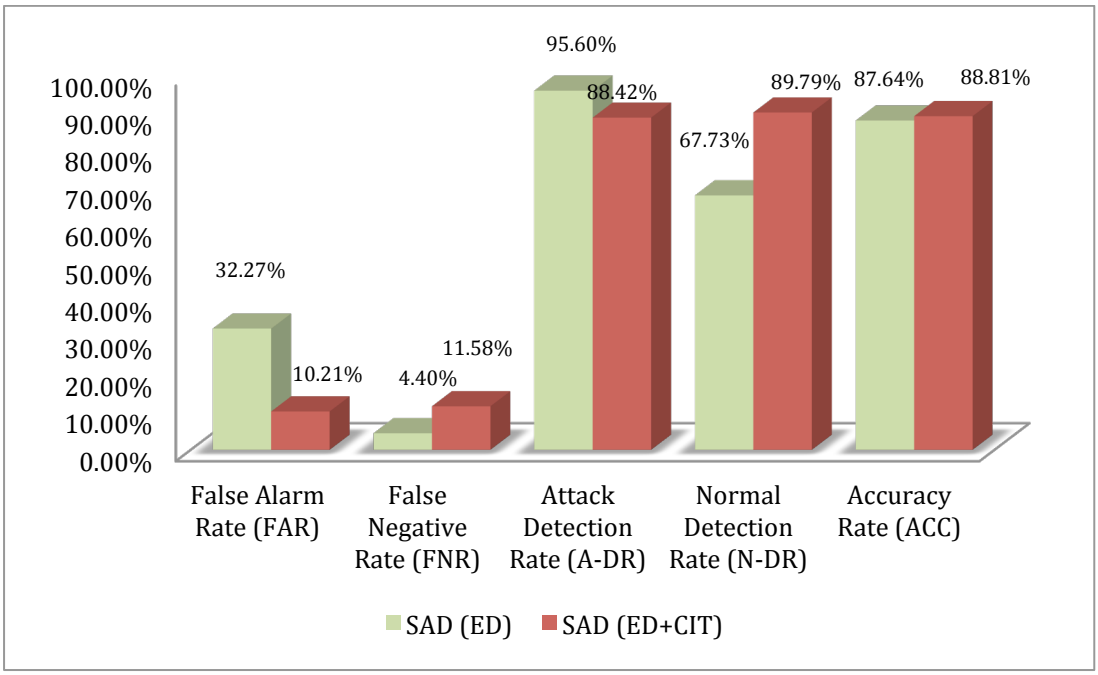


Figure B.11: Performance of Statistical Analysis Detection on UNSW-NB15 Dataset

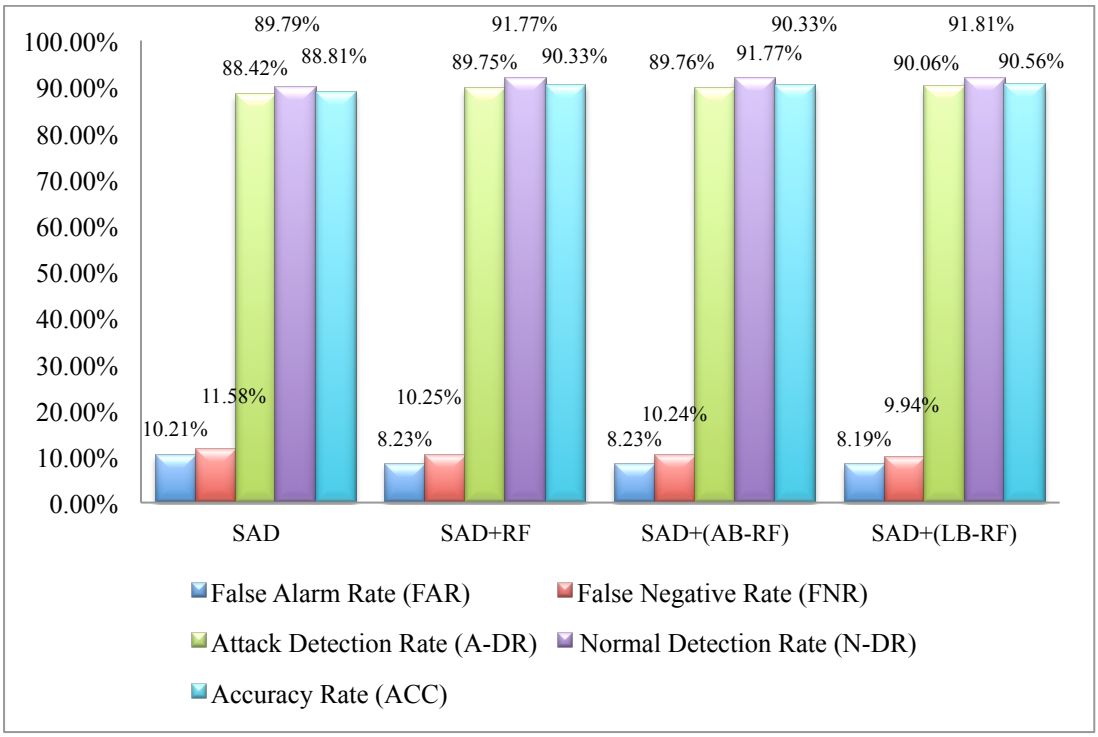


Figure B.12: Performance of Anomaly Detection Approaches on UNSW-NB15 Dataset