A Study of Malware Behaviour of Webpages

by

Alhanoof Faiz Alwaghid

A thesis submitted to the Faculty of Design and Creative Technologies

Auckland University of Technology

in partial fulfilment of the requirements for the degree of

Master of Information Security and Digital Forensics

School of Engineering, Computer and Mathematical Sciences

Auckland, New Zealand

2018

Declaration of Originality

I hereby declare that this submission is my own work and that, to the best of my knowledge and belief, it contains no material previously published or written by another person (except where explicitly defined in the acknowledgements), nor material which to a substantial extent has been submitted for the award of any other degree or diploma of a university or other institution of higher learning.

Signed: Alhanoof Alwaghid On: 14/12/2018

Abstract

Malware is the most common security threat experienced by the user when browsing webpages. The features of webpages such as the internet protocol, port, universal resource locator (URL), combo, Google index, email, web traffic, HTTPS token, and page rank are vital to study the behaviour of malware in a webpage. To analyse these behaviour, phishing and botnet data were obtained from University of California Irvine machine learning repository. To validate the findings, honeypot infrastructure was applied by using the Modern Honeypot Network (MHN) set-up in a Linode server. In this thesis, an experimental analysis was performed to identify the features in webpages that are most vulnerable to malware attack and its results were reported. To improve the feature selection accuracy, a machine learning technique called bagging was employed. As the data suffer from high variance in terms of the type of data in each row, bagging is chosen because it can classify binary class, date class, missing values, nominal class, numeric class, unary class and empty class. As a base classifier of bagging, random tree was applied because it can handle similar types of data as bagging, but better than other classifiers because it is faster and more accurate. Random tree had 88.22% test accuracy with the lowest run time (0.2 sec) and a receiver operating characteristic curve of 0.946.

The findings of the research showed that all features in botnet dataset were equally important to identify the malicious behaviour as all scored more than 97%, with the exception of TCP and UDP. Scan is an important feature as it obtained the highest score 99.79% accuracy in terms of "correlation coefficient" in test performance. This study identified that having a valid IP address does not guarantee that a website is reliable as it was vulnerable during malware attack and the number of phishy behaviour was 3,793 out of 11,055. Therefore, port feature is mostly secured during malware attack as the number of legitimate behaviours was 9,553 out of 11,055. The results showed that links pointing to a website may not always safe be, as they were 'suspicious' in 55% accuracy and 'phishy' in around 5%. This research revealed that the Alexa web ranking (which is very popular for Web of Trust certification) is not as safe as the Google index ranking since the page rank had phishy behaviour that ranked 8,201 out of 11,055 web hits while for Google-indexed pages, phishy behaviour was only 1,539 for the same number of web hits. During the research experiment, it was discovered that the accuracy of phishing and botnet datasets is more than 89% average in both cross validation and test analysis. The study concludes by offering recommendations and future research directions that may assist in future malware identification.

Dedication

This work is dedicated to My mother Alanoud, My beloved husband Hosam, lovely Children Salem, Alanoud and Almaha. For their unfailing love, support, and prayers throughout the course of this thesis

May God bless you all

Acknowledgements

First and foremost, I would like to thank God (Allah) for giving me the inspiration, patience, opportunity, and strength on this journey.

During the time of writing this Master thesis, I received support and help from many people. In particular, I wish to express my sincere appreciation and gratitude to my supervisor, Associate Professor. Nurul I Sarkar, for his support, guidance, valuable feedback, encouragement, and expertise that I needed during my thesis. I have appreciated his patience, and comments throughout the development of this study.

I am thankful to my lovely family for making success possible and rewarding. I will never ever forget to thank my great father Faiz, who passed away, but his words of wisdom are still on my mind. My special thanks go to my dear husband Hosam Aldhaher who encouraged me to reach my dreams and motivated me when I needed it the most. He supported and assisted me in each step to complete the thesis

I especially like to profoundly acknowledge my mother Alanoud for her constant love and endless support, for her prayers. To my sisters and brothers for their amazing and continuous encouragement who never gave up supporting and caring all the time. I am indeed blessed to have them in my life. To my beautiful little son Salem and daughters Alanoud and Almaha who have been my motivation and inspiration.

Finally, I am thankful to Aljouf University for their scholarship, and to my country for funding my education, and the Saudi Cultural Mission in New Zealand for their continuous support. Thanks for Elite Editing for the feedback and proofreading of my thesis.

Table of Contents

List of Abbreviations and Acronyms

Adware	Advertising-supported software
ANN	Artificial neural networks
AR	Association rules
ASU	Arizona State University
BART	Bayesian additive regression trees
Botnet	Bot network
C&C	Command and control
CHI	Chi-square
CART	Classification and regression trees
CBR-PDS	Case-based reasoning–phishing detection system
DDoS	Distributed denial of service
DGA	Domain-generation algorithm
DMF	Direct matched filter
DNS	Domain name system
DT	Decision trees
Email	Electronic mail
GUI	Graphical user interface
HTTP	Hypertext transfer protocol
HTTPS	Hyper Text Transfer Protocol Secure
IDS	Intrusion detection system
IG	Information gain
IoT	Internet of Things
IP	Internet protocol
ICMP	Internet control message protocol
ICVA	Intelligent cross-view analyser
KNN	K-nearest neighbour
LOIC	Low-orbit ion cannon
LR	Logistic regression
LSTM	Long/short-term memory (neural network)
MHN	Modern Honeypot Network
MSSQL	Microsoft structured query language
NAT	Network address translation

PC	Personal computer
PCA	Principal component analysis
PIN	Personal identification number
RF	Random forest
ROC	Receiver operating characteristic
SEO	Search engine optimisation
SIP	Session initiation protocol
SQLI	Structured query language injection
SSF	Spectral signature filter
SSH	Secure shell
STVN	Session traversal utilities for NAT
SVM	Support vector machines
TCP	Transmission control protocol
UCI	University of California Irvine
UDP	User datagram protocol
URL	Universal resource locator
VMI	Virtual machine introspection
VMware	Virtual machine ware
WWW	World wide web
XSS	Cross-site scripting

List of Figures

List of Tables

Chapter 1
Introduction

Malware is known as malicious software that represents a crucial threat to the security level of systems. At present, malware codes are hidden behind a huge amount of data, so existing defensive mechanisms often are not able to defend against malware attack. Malware attacks could cause damage to many internet-connected devices via viruses, worms and Trojans, among many others [1]. Since internet data are substantial, the pattern of malware attack may differ, but is identifiable by its nature. Malware in webpages is one of the biggest threats for both home users and organisations. Malware continues to be a cyber-threat and was observed in 2016 more than 357 million of malware variants [2]. AVTEST reported that 95 million websites were infected by malware in 2017 [3]. Cyber technicians are working to identify the types of malware attacks that can be prevented. It is hoped that several types of attack such as the adware, malware, rootkit and many others may be prevented earlier if their pattern or behaviour are known.

Malicious software is defined as malware [4], [5]. The behaviour of malware can be identified from a webpage and browsing history or data. Data from a malware can hint at the malware's properties but not the relationships among features of the data; and mostly these data do not identify 'suspicious' behaviour. Nonetheless, attackers try any possible approach to break into a victim's system. However, tactics are preferred by adversaries that allow them to attack a huge number of users in several minutes [6], [7]. Nowadays, the browsing time or number of browsing websites in a specific period (sec) is an important factor in a website's properties, along with identifying the behaviour of a malware. For instance, once recent very common attack is distributed denial of service (DDoS). DDoS tools such as low-orbit ion canon (LOIC) can hit a web system 4,800 times by the same internet protocol (IP) [8], [9], [10]. Therefore, IP address are important feature to identify malicious (DDoS) behaviour.

Most hackers today can effectively escape detection by security protocols [11] such as firewall and intrusion detection system (IDS), invaders have used techniques to spread their exploited code that include utilising online advertisements of website pages [12], [13], structured query language injection (SQLI), cross-site scripting (XSS) and another web scanner [14]. Hence, in many cases, identification of a hacker is not possible [15]. However, despite the potential security threat from attacks, it is possible to protect a

website/server from damage by recognising the behaviour of malware attacks [16]. The McAfee threat report identified malware as the most common form of cyber-attack. Therefore, the main concern is the behaviour of malware with the aim of suggesting a security protocol to prevent future damage in web space [16]. Malware behaviour on a website has been exclusively studied because malware is preventable if its nature is identified [17]. The nature of malware can be identified with feature selection techniques. When data are multivariate and required more pre-processing, classification with ensemble methods (a machine learning technique) may perform better to select suitable features. In most cases, malicious data are not in the correct format for suitable features to be selected from the data. However, machine learning offers a promising solution to identify different types of malicious behaviour [18]. The current study identifies the behaviour of malware by classification accuracy in terms of the number of occurrences.

The empirical investigation reported in this thesis provides clear guidelines for selecting features with appropriate classification techniques, which will help to identify the behaviour of malware. Thus, future computing may be better able to fight malware. The important features of malware may be having an IP address, port, universal resource locator (URL), pop up window or email, which are identified in this study. The primary aim of this research is to identify and analyse malicious webpage behaviour.

To achieve the research goal, research objectives and address the research question in line with the problem statement, this study considers the property of a webpage as having an IP address, port, requested URL, email browsing and web traffic, based on experimental datasets. The first dataset is donated by Mohammad et al. [19], [20], [21], from the University of California Irvine (UCI) machine learning repository; the second dataset is donated by Meidan et al. [22] from the UCI Machine Learning Repository. Honeypot data (see sec 2.7 for more explanation) are collected by deploying Modern Honeypot Network (MHN) software [23].

1.1 Objectives

The aim of this research is to study the malware behaviour of webpages to identify their techniques and make recommendation for future work. The outcomes of the research depend on the analysis of three datasets (phishing, botnet and honeypot) to achieve the aims of the study. We identify malware behaviour through feature selection, determine influential features that have been targeted by attackers, generate similarities between the properties of malicious webpages to identify the common target of exploitation, and predict malware vulnerability of specific features.

1.2 Problem Statement and Research Questions

Malware is one of the latest security threats to stand-alone computers or even a secured network. Existing malware detection systems are not capable of fully protecting a computer from malware attack [24]. A significant amount of research has been conducted on malware attack, but little attention has been paid to the behaviour of malware. In this scenario, a study on malware behaviour is required to understand malware attack. As most malware is spread through websites, this study aims to analyses the effects of malware features on websites, based on available data and from custom honeypot infrastructure applied via MHN technology on a Linode server. Most datasets on malware are not reliable as they contain insufficient data description and features are not clearly understandable. The research challenge is great when we have huge number of malware data without the meaning or the relationships among features of the data. This research collected data from different sources including honeypot infrastructure where the nature of the data is malware and there is information about IP, source port, protocol and requested URLs. However, the datasets do not identify malware behaviour among the features. Data that do not provide sufficient information are not suitable for identifying malware behaviour. Thus, identifying behaviour from these features that is closely related to that of malware may prevent future malware attack. It is better to use several datasets to identify malware features and to validate the findings. In this study, one dataset is based on honeypot infrastructure and the other two are real web-based server data obtained from the UCI Machine Learning Repository—the first being from detection of Internet of Things (IoT) botnet attacks (N BaIoT), and the second from phishing websites. The research findings are validated using different data sources.

The following main research question and sub-questions are addressed. The proposed main research question for the study is:

What research can be done to identify and analyse malicious webpage behaviour?

To address this research question, the following sub-questions are answered.

- How can malware behaviour be identified through the feature selection method used by attackers, based on their behaviour?
- What types of exploitation have been used by attackers?
- What are the similarities between malicious webpages properties and the known common targets of exploitation?
- How can we predict malware attacks using information from the datasets?

1.3 Contribution and Structure of This Thesis

The main contribution of the study is the identification of malware behaviour. Malware behaviour is identified using three datasets. This study also identifies similar features from different datasets that are targeted during malware attack. Finally, the findings of this study suggest the kinds of features that are identical or point to exploitation by hackers in malware attacks to reduce such attacks. In summary the research contributions are:

- Identification of the most targeted features via malware attack in three datasets: phishing, botnet and honeypot.
- Comparison of maliciousness in two available datasets and application of similar techniques to identify maliciousness in custom-built honeypot infrastructure. The identification is achieved via the accuracy of the number of occurrences for selected features.
- Identification of the most vulnerable features that are common to the three datasets.
- Identification of 'legitimate', 'phishy' and 'suspicious' behaviour in these features.
- Description of the difference between Google index and page rank in identifying malware behaviour, which is a significant achievement of this research along with identification of malware behaviour on webpages.
- Identification of the behaviour of malware as legitimate, suspicious and phishy.

The overall structure of this thesis is shown in Figure 1.1. To achieve the research goals and objectives, and address the research questions, a basic understanding of malware behaviours is required.

Chapter 1 began by presenting the motivation and background of the study, which also explained the current problem of malware attack. To protect malware attack it is important to understand the behaviour of malware, which is the main concern in this study as depicted in the research objectives. The research question provides a general idea of the research that is required for this study. Finally, this chapter ends by describing the research contribution/significance and the organisation of the thesis.

Chapter 2 reports the literature review, which discusses work related to this study in line with the problem statement and research objectives. In particular, it provides an introduction of identifying malware behaviour in webpages and classification and machine learning techniques. Feature selection in malware webpages, which is an

important issue, are described in detail. Therefore, malware behaviour of webpages is highlighted in this chapter. To provide a review of the literature in the area of malicious attacks, honeypot technology in relation to the aims, types, level of interaction, purpose and architecture are also surveyed in this chapter.

The original contributions of this thesis are presented in Chapters 3, 4, and 5, which primarily explain the key factors of feature selection being performed by attackers, based on their behaviours.

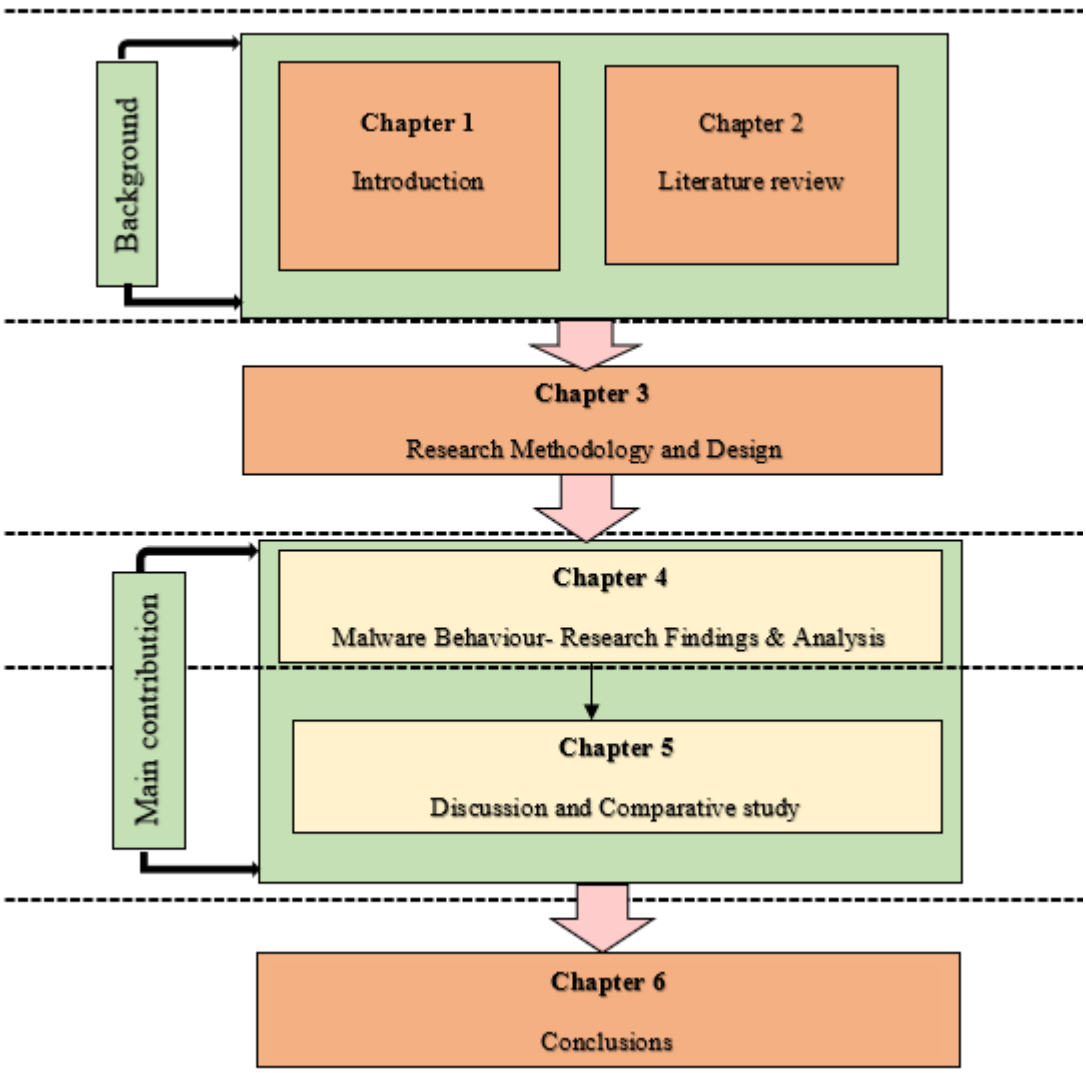


Figure 1.1: The structure of the thesis.

Chapter 3 reports the research methodology and design used in the thesis. The ensemble method used in data analysis is defined, along with the bagging algorithm. In this chapter, honeypot software utilised in the study is described in detail.

In Chapter 4, the major findings from the three datasets are investigated in detail. The primary objective of this empirical study was to obtain insights by comparing three datasets to identify malware behaviour in webpages.

Chapter 5 presents the main contribution by discussing the results from the three datasets. The thesis is summarised and concluded in Chapter 6, future research directions for future developments are also outlined.

Chapter 2
Literature Review

2.1 Introduction

Chapter 1 provided an outline of the problem statement, objectives, and contribution of this thesis. A primary objective of this thesis is to identify and study the malware behaviour of webpages. To achieve this aim, a general understanding of malware behaviour is required. This chapter aims to present an introduction to various key concepts of malware that are necessary to study malicious attack by webpages. Section 2.2 identifies malware behaviour in webpages. Classification and machine learning techniques used to identify malware behaviour are highlighted in Section 2.3. Section 2.4 presents feature selection in malware websites. Section 2.5 outlines the malware behaviour of webpages, specifically discussing the current and common sources of malware including IP, port, requested URLs, email and web traffic, which are important issues to be considered in a study focusing on the sources considered in this thesis. In Section 2.6, the malware behaviour of honeypot is defined. Section 2.7 describes various key parameters relating to active honeypot; the architecture of honeypot is also discussed in this section. Section 2.8 discusses types of honeypot: low, medium and high interaction. As this thesis focuses on the malware of webpages, different types of malware are the focus of this thesis (Section 2.9). Finally, the chapter is summarised in Section 2.10.

2.2 Identifying Malware Behaviour in Webpages

Malware is a threat that can copy a legitimate site such as PayPal or eBay to harm the user. It may access a victim's computer by offering money or incentives; sending warning or enticing emails; or posting URL links in social media. Thakur and Verma [25] tried to detect phishing websites based on a URL classification by the number of hits, and suggested that future work should combine this with a malware detector to detect malware. The current study aimed to detect malware by identifying behaviour of webpages focusing on three main categories: 'phishing', 'legitimate' and 'suspicious'.

Altaher [26] used several classification techniques, including Naïve Bayes, neural networks, support vector machines (SVM), decision trees (DT) and k-nearest neighbour (KNN) to identify the behaviour of websites. The author proposed a hybrid methodology that combines the KNN algorithm with SVM to classify websites as phishing, legitimate

or suspicious. First, KNN was applied to classify noisy data and then SVM was applied to improve the classification. KNN and SVM performed better than other classifiers with 87.45% and 83.76% accuracy, respectively. The hybrid methodology gave the highest accuracy of 90.04%. The important findings from the research were that phishing (website behaviour) always obtained more than 90% accuracy, which suggests that to identify malware behaviour in websites, phishing behaviour needs to be considered. Although Altaher's research quantified performance of several classifiers it did not consider the performance of DT or other popular classifiers such as ensemble. Therefore, the current study employed the DT classification and ensemble method, in line with [26]. By employing the ensemble bagging method, the current research also achieved more than 95% accuracy in classification of phishing behaviour in websites, which is better than [26].

Bahnsen et al. [27] proposed two methods to identify phishing URLs from websites. One method was feature engineering with a lexical and statistical URL analysis and random forest (RF) classifier. The second method, long/short-term memory (LSTM) neural network, was claimed by the authors to be novel as it had a model training accuracy score of 0.98 whereas RF had a model accuracy score of 0.93. Although LSTM was 5% more accurate than RF it may not be an acceptable option by the researcher because the run time was almost 4 hours, whereas RF required only 3 minutes. In the current study, near or similar accuracy was achieved by employing the ensemble method with random tree, and the run time was less than 3 minutes. Thus, the method employed in this study to identify phishing URL is superior in terms of time and accuracy.

2.3 Classification and Machine Learning to Identify Malware Behaviour

Machine learning methods are a suitable analysis technique for classifying websites as legitimate, phishy or suspicious because they utilise a binary classification [28]. The main point of these methodologies is to classify the behaviour (feature) instead of the user as many clients are unable to identify malware attacks [29]. Machine learning is depended on datasets that contain either previous experiences or a collection of examples. One example is a set of attributes or features [30]. Phishing attacks can be controlled using machine learning methods such as SVM, RF and logistic regression (LR). To detect phishing sites, Liu and Wenyin [31] suggested a method that finds sets of related webpages for a site. They discovered relationships for a given site by focusing on features such as similarities in text and webpage layout, ranking relationships and links. Their investigations resulted in 91.44% accuracy and around 3.40% false alarm rate. Several

machine learning methods were considered by Abu-Nimeh et al. [32], including Bayesian additive regression trees (BART), RF, LR, SVM, artificial neural networks (ANN) and classification and regression trees (CART) to predict phishing attacks in emails. They tested 2,889 samples in both phishy and legitimate emails, which helped them extract 43 features. In their research, LR performed better than the others; however, Basnet and Doleck [33] found RF to perform the best and SVM, the worst, when comparing seven methods of machine learning. Huang et al. [34] investigated use of an SVM technique to recognise phishing webpages.

A new methodology presented by Marchal et al. [35] considers relationships between the low-level domain and the upper-level domain; that is, the registered domain and path or query in URL, respectively. The authors characterised a new concept named intra-URL relatedness and used queries in Google and Yahoo to establish a relationship between the words. The features were extracted from the words composing the URL to help the authors assess the previous concept that is called intra-URL relatedness. They demonstrated a classification rate of 94.91% accuracy and false positive rate of only 1.44%. RF methods can manage many variables and can help to assess missing values and can build a random forest that causes difficulties in understanding the latter model and related results [28]. These days, attackers employ complicated URLs to trick end users.

Choi et al. [36] proposed a machine learning approach (SVM and KNN) to identify malicious URLs. They studied several features of URLs such as webpage content, domain name system (DNS) information and network traffic. The classification model was used to detect 40,000 benign URLs and identify 32,000 types of malicious URL attack, with over 98% accuracy and more than 93% accuracy, respectively.

To recognise attacks by phishing webpages, the authors in [37], [38], [39] and [40] utilised URL-based features. The study of Abutair and Belghith [41] focused on extracting URL features and the similarity of key words in the domain name and URL; for instance, the subdomain and query parts. The authors claimed that none of the classification methods for phishing attacks can identify and prevent phishing webpage attack because of the short life cycle of phishing webpages. They presented a case-based reasoning–phishing detection system (CBR-PDS) approach with the CBR technique as the main part and proposed that this system can detect any new phishing website, unlike other classification systems, which must be prepared in advance to detect attacks; their technique is adaptive and dynamic. When testing their system utilising 572 phishing and

legitimate URLs, the classification result of 95.62% accuracy was achieved with limited dataset and features.

Machine learning is an advanced technology to study malware and is important for protection from malware. Al-Garadi et al. [42] discussed several machine learning algorithms including DT, SVM, Bayesian algorithms, KNN, RF, association rules (AR), ensemble, learning, k-means clustering and principal component analysis (PCA), along with their advantages, disadvantages and applications in security. It is often noted that SVM classifiers may outperform DT. However, the DT with ensemble method may enhance the performance of DT, which may supersede SVM.

Hoang and Nguyen [43] examined the effectiveness of supervised learning techniques to select suitable features in botnet data from the Alexa top-level domain using some common supervised machine learning algorithms including KNN, DT, RF and Naïve Bayes. The authors focused on DNS queries and obtained over 90% accuracy in general. However, they did not focus on individual features such as the effect of transmission control protocol (TCP), user datagram protocol (UDP), junk, benign and so on. Thus, a study of botnet data is required that might reveal the individual effects of those features. The current research applied DT algorithms more precisely and used an ensemble DT algorithm with random tree as a base classifier. The study also identified related features that are the cause of malware attack, according to classification accuracy.

Jain and Gupta [44] suggested a machine learning technique that depends on an anti-phishing method. The system, called PHISH-SAFE, is based on URL features. They conducted a study to identify websites that were either phishing or non-phishing, and 14 URL features were used to test the system's performance. Two types of classifier (SVM and Naïve Bayes) were used to prepare the system, with around 33,000 URLs (phishing and legitimate). The accuracy of recognising phishing sites was over 90%.

Mirza et al. [45] focused on two main issues in the domain of malware detection; (a) accurately identifying a malware and (b) enhancing the efficiency of the detection mechanism. The authors employed CloudIntell machine learning techniques to enhance the malware detection rate and cloud-based architecture to support and host the methodology implementation. They used DT and SVM and then applied boosting (ensemble) to DT to improve the performance of weak classifiers. Rieck et al. [1] proposed an automatic malware behaviour analysis framework based on a clustering and classification method in the sandbox environment, although their research ignored scenarios in real webpages. The authors criticised the scale super-linear method and claimed that it cannot be directly applied to malware analysis. However, they did not

demonstrate that their clustering and classification analysis was better than the scale super-linear method. It would be better if their analysis had identified the source of malware and malware behaviour instead of simply classifying malware.

Mohaisen et al. [46] discussed a malware analysis tool that is automatically based on malware behaviour. The tool, called AMAL, has two subsystems—AutoMal and MaLabel—both of which are based on medium-scale and large-scale datasets. The author criticised the hierarchical clustering method and claimed that it provides limited insight into illustrated malware samples. However, the collection step using AutoMal and clustering step with classification using MaLabel were the only steps followed; the authors did not identify malware features and behaviour.

Table 2.1: Key researchers and their contributions in machine learning algorithms.

Publication	Contribution	Year	Description/key concept
Altaher [26]	Naïve Bayes, neural network, SVM, DT and KNN	2017	Identifying behaviour of websites as phishing, legitimate or suspicious
Bahnsen et al. [27]	LSTM and RF	2017	Detecting URLs from websites
Abu-Nimeh et al. [32]	BART, RF, LR, SVM, ANN and CART	2007	Predicting email attacks
Basnet and Doleck [33]	RF and SVM	2015	Comparing performance of RF and SVM with seven methods of machine learning
Marchal et al. [35]	Intra-URL relatedness	2014	Using queries in Google and Yahoo to establish a relationship between words
Abutair and Belghith [41]	CBR-PDS	2017	Extracting URL features and identifying similarity of key words in the domain name and URL
Al-Garadi et al. [42]	DT, SVM, Bayesian algorithms, KNN, RF, AR, ensemble, learning, k-means clustering and PCA	2018	Discussing several machine learning algorithms
Hoang and Nguyen [43]	KNN, DT, RF and Naïve Bayes	2018	Examining effectiveness of supervised learning techniques to select suitable features in botnet data
Jain and Gupta [44]	PHISH-SAFE	2018	Identify websites as phishing or non-phishing
Mirza et al. [45]	DT, SVM and ensemble	2018	Enhancing malware detection rate and developing cloud-based architecture
Kumara and Jaidhar [47]	VMI	2018	Characterising unknown benign and malware data to conduct forensic analysis inside memory

Detection of hidden and malicious processes executed on a virtual machine (VM) using virtual machine introspection (VMI) in a virtualised environment presents several problems, including that the information is not meaningful unless its behaviour is identifiable. Kumara and Jaidhar [47] utilised VMI technology to characterise unknown benign and malware data to conduct a forensic analysis of inside memory; and intelligent cross-view analyser (ICVA) to identify hidden, dead and dubious processes data. They employed 10-fold cross-validation to detect unknown malware but did not present their test results. Table 2.1 lists the key researchers and their main contribution in machine learning algorithms.

2.4 Feature Selection in Malware Websites

Feature selection is important when data are high dimensional and computational power needs to be minimised. To achieve better accuracy and faster run times, random feature selection is better and can sometimes be done based on feature relevance in terms of accuracy [48]. The feature selection process is crucial to delete irrelevant features or noisy data based on certain criteria to enhance the performance of machine learning techniques [49]. Zhao et al. [49] developed a feature selection algorithm repository called ASU, which is a combination of common feature selection algorithms classified based on three different models: filter, wrapper and embedded.

Some organisations and end users depend on antivirus tools and security techniques to secure their devices. However, the techniques utilised by such programs are inadequate for identifying and preventing malware performance. Mirza et al. [45] used several machine learning methods on features taken from a large dataset of benign and malicious files using a feature extraction device. The features were extracted from data by applying SVM, DT and boosting on DT to achieve the highest possible detection rate.

Basnet et al. [50] evaluated two feature selection methods to identify phishing attacks: correlation-based and wrapper-based feature selection; and three machine learning classifiers, Naïve Bayes, LR and RF, were compared. The authors demonstrated that the feature selection method that affected classification results in their study was wrapper-based feature selection, which was slower than correlation-based. However, they collected their dataset without analysing the features and compared the feature selection methods based only on error rates—false positive and false negative. Based on this, the current study has chosen correlation-based accuracy when employing bagging and random tree.

Rajab [51] considered two feature selection approaches in his study: information gain (IG) and chi-square (CHI). The author's goal was to develop a metric for feature selection by finding the score of classification accuracy from preprocessed phishing data using the two methods of feature selection. The study concluded that a new feature selection method that assesses by picking the relevant features influences the phishing data detection rate.

Basnet et al. [52] classified phishing URLs by utilising a heuristic-based method whose classifier is based on data offered only in URLs, without examining webpage contents. The authors studied phishing and benign URLs, and features were extracted by running several scripts. To classify phishing URLs, features were selected based on four categories: lexical, keyword, search engine and reputation. The authors aimed to identify URLs as either phishing or non-phishing. In their study, several machine learning techniques were compared to determine which was a better classifier for phishing URLs. However, the authors did not examine the suspicious feature of URLs, as was done in the current research.

This research has selected the most common features—IP, port, request URLs, Hypertext transfer protocol secure (HTTPS) token, Google index, submitting to email, web traffic, page rank, abnormal URLs, pop up window and links pointing to page—to achieve better accuracy with shorter run times to identify malware behaviour in websites.

2.5 Malware Behaviour of Webpages

Every year, the number of malwares increases substantially compared with the previous year [53] and it is almost impossible to study or examine the features and nature of individual malwares produced globally [54]. The study by AVTEST [54] illustrated the trends in malware attack per-year from 2008 to 2017, the number of attacks increased from 100 million to 600 million, respectively.

Malicious behaviour can be observed in websites as well as in IoT devices connected to internet. Malware attack occurs because of weakness in a security system. Numerous malicious attacks occur through DDoS, structured query language (SQL), XSS, HTTPS token, web traffic, pop up window, email or third party software [42].

Network activities are continually monitored using tools and methods to identify and recognise phishing webpages and targeted sites. Monitoring network traffic logs involves analysing events or generating information related to requested URLs. Based on analysis of requested webpages, some webpages could be considered phishing sites. To detect phishing websites, set-up implementation and functionality of a browser or network

router can assist in this operation [55]. Malware is a kind of software that is designed to damage a computer system without the consent of the owner. Computer viruses, worms, Trojan horses, Rootkit, adware, spyware and any other unwanted programs that have malicious behaviour are considered malware [56]. At present, malware is considered the greatest threat to web security as it most often target's a victim's computer, although it is not easy to install malware without the victim's consent [24]. However, malware can be installed without the victim's consent by a single click on ordinary images, web links and so on.

Internet communication technology is mainly based on TCP and IP, which were designed to provide secure communication. However, TCP and IP became vulnerable because of a design fault (IP spoofing, DDoS on IP and so on) [57]. Network resources are not safe either as they are exposed to unauthorised access, such as an email that can be accessed by an unauthorised person and hacked using social engineering or malware [58]. CertNZ [58] mentioned that websites are one of the resources that may suffer from unauthorised access. An unauthorised person may gain access to usernames, passwords or login details by using different types of malware or dictionary-based software such as brute force. Thus, this study considered email/junk, username and password as an important feature to detect malware behaviour.

Pandey and Saini [57] conducted a study on TCP, IP and UDP to understand attack mechanisms. They used several tools to identify the vulnerability of a network based on these three features and suggested that it is necessary to learn how to protect network security rather than simply identifying vulnerability. To meet this goa, the current study proposed machine learning techniques to identify related features such as TCP, IP, junk (email), port and their malicious behaviour, to identify future malware trends.

Attackers targets the web site by using botnet to manipulate the original IP for a long time. Botnet can be distinguished from typical malware because it take control on the infected machines by 'bot-herder' [59]. Botnet allows an attacker to initiate remote control over a victim's computer to install malware. Li et al. [59] described botnet attacks based on DNS and reported several studies of botnet techniques. However, the research did not focus on the features of a botnet, such as which features are more related to security vulnerability. To fill this gap, the current research studied botnet data and identified relevant features that represent malware behaviour (Section 4.3.2).

DDoS, floods and DNS security pharming are the main way to manipulate the IP to plant malware [60]. Seo and Lee [61] calculated the frequency of network-based packet attributes and analysed anomalies in attributes to detect IP-spoofed DDoS attacks via the

direct matched filter (DMF) algorithm. The DMF uses a spectral signature filter (SSF) instead of a thresholding process; the SSF includes a numerical description about the relevance. Their proposed method for effective detection of malware infection systems is based on accuracy of triggering IP-spoofed DDoS attacks on an edge network. Detection accuracy and performance of the collected real-time traffic on a core network meant that DDoS attacks on the internal network were detected in real time, and whether IP addresses were spoofed was confirmed. The current research employed machine learning techniques (bagging with random tree as a base classifier) to study the most suitable base classifier (Table 4.1) for identifying the relevance of malware features in similar types of botnet data. A correlation analysis based on the machine learning algorithm shows how much the features are related (Section 4.3). This study did not employ SSF techniques, but defined malware terms as '1' for legitimate, '0' for Suspicious, and '–1' for phishy.

Black et al. [62] described a Dyre, SOCKS and web injection protocol that enables malware to work through the IP. Dyre malware contains a domain-generation algorithm (DGA) that uses the date as a key to generate the C&C server's IP address and port pairs [63]. Dyre malware uses session traversal utilities for network address translation (NAT) (STUN) to find the IP address of the infected computer via a NAT internet connection [64], [65]. Dyre performs a man-in-the-middle attack by redirecting website requests through an attacker-controlled proxy server IP. A SOCK proxy provides TCP access to the attacker with the ability to login to an internet banking website; banking security software assumes it is legitimate as the banking session originated from the IP address of the victim's computer [66], [67]. Webinjects are added when a website response is received from that IP. Storing the webinjects on an attacker-controlled server has the advantage of hindering efforts to access the webinjects for analysis [68].

HTTP protocols run the world wide web (WWW), which is one of the most widely used network protocols [69]. A huge number of studies have been conducted on HTTP protocols. For example, Hoang and Nguyen [43] used SVM learning algorithms to develop a model that is capable to classify both botnet and legitimate traffic; Kheir et al. [70] focused on HTTP-based botnets' C&C patterns to classify network traffic into clusters; Tariq and Baig [71] studied machine learning-based botnet detection. A significant amount of research has been conducted on HTTP-based botnet detection. However, only a small proportion has focused solely on detecting the effect of the HTTP feature of websites to identify malware behaviour.

Similar to IP, HTTP has become a point of interest for C&C communication of botnets. C&C communication codes are easily hidden because of the massive amount of web

traffic. However, detecting anomalies in HTTP is not an easy task. To detect an anomaly, Sakib and Huang [69] proposed HTTP-based C&C traffic which uses statistical features based on client-generated HTTP request packets and DNS server-generated response packets. The authors employed unsupervised learning based on feature selection (frequency, mean, one-class SVM) on botnet data and achieved over 90% detection rate. However, the authors ignored the performance of supervised machine learning techniques.

HTTP are the primary target of bot masters within C&C infrastructure because they blend web traffic with benign. Several studies have aimed to characterise or detect HTTP-based bots, and many have used network communication features as identifiers of botnet behaviour. Acaralia et al. [72] undertook a detailed survey of HTTP bots to determine how a HTTP bot is different from normal traffic and tried to identify the relationships among the features. However, they only reported previous studies while the current research studied behaviour of HTTPS in terms of the relevance to malware behaviour.

Cyber threats such as phishing can affect a huge number of users within several minutes and may target financial data or online information. Phishing attacks are one of the cyber threats for which there is no specific solution to assist with stopping the attacks. Kaytan and Hanbay [73] proposed a model to identify phishing websites using machine learning. There are several types of website, based on website features. The authors developed several new rules to identify efficient features. Their model consists of 30 inputs and 1 output, and a cross-validation algorithm (10-fold), which resulted in 95.05% accuracy.

Phishing attacks are considered a security threat from attackers attempting to steal personal data, for instance, PINs and credit card details [44]. Based on Nivedha et al. [74], they are used to obtain sensitive data such as usernames and passwords. Such attacks continue to be a serious issue for web users including in the field of electronic commerce. There are several machine learning techniques and classifications for URL features that can help protect users against phishing webpages [75]. Phishing URLs are one of the main features targeted by malware. Malware in URLs pose a security threat that is of current concern [76]. In many studies, ‘drive by download’ has been identified as malware that is making computer security vulnerable. Tanaka et al. [77] investigated the behaviour of malware in download sites and detected ‘43,000’ malicious URLs over an 18-month period. The author developed a monitoring system that is able to find malicious webpages based on URL features and to observe whether the malware of URLs has changed or remained the same. The study detected three kinds of URL malware: unchanged, every

time changed and changed occasionally. However, the author did not focus on several types of URLs, including request URL and abnormal URL, which were examined in the current research.

Phishing attacks are causing security issues around the world. Phisher can launch attacks from anywhere and phishing can be done by an individual with a low level of technical skills [78]. The process of protecting a company's users has become more complicated with the increasing number of emerging websites that should be considered malware webpages before users access them. Many entertaining and tricky techniques are used by attackers to attempt to make a site seem legitimate [79]. Redirecting the victim to a phishing URL is the main aim of a phisher and impersonating a legitimate URL is their preferred strategy to achieve their goal.

Hybrid approaches such as the KNN and SVM algorithms are probably a better option for classifying websites as phishing, legitimate or suspicious [26]. Bearing this in mind, this study has applied an ensemble method, which is a hybrid DT approach to identify phishing, legitimate or suspicious webpages.

Machine learning techniques such as RF are better classifiers of the behaviour of malware sites based on URLs. Bahnsen et al. [27] compared the combination of lexical and statistical analysis of URLs as input for a RF classifier and claimed that their study was novel in employing a LSTM network in a recurrent neural network. Although their method does not require the manual extraction of features, it is not efficient as shown by the training results used for comparisons. The authors considered only a 3-fold cross-validation but ignored the most acceptable performance among the 'test' results. For instance, the current study proposed use of a DT algorithm to identify similar behaviour based on 10-fold cross-validation and test performance.

Attackers employ any possible approach to break into a victim's system. However, some tactics preferred by adversaries guide them to attack a huge number of users in the course of several minutes. There are current and common sources of malware that might result in security threats, including IP address, port, malicious URL, email and web traffic.

The numbers in an IP address are allocated to a PC to arrange interfaces that are used to connect devices such as computers, servers, and printers [80]. Devices can be identified by their IP address and these data are transferred when a network uses an IP address. There are a variety of security threats involving IP addresses; some attackers use threat models, such as IP piracy [81], to compromise or steal IP addresses and sell them illegally. However, automated tools such as InferIP [82] can be used to identify and detect

malicious IP addresses and provide an early warning that captures attacker behaviour to decide whether it is malicious or not.

The port is responsible for sending and delivering data and messages within a network by establishing a channel between two devices [83]. Many backdoor attacks focus on scanning methods such as port scanning to compromise a system; the attacker aims to collect information about a targeted system and its operating system [84], [85]. Hackers utilise scanning tools such as Nmap, Amap and Unicornscan to check whether a port is open or closed and to obtain more information [86].

A URL is considered a web address as it contains a link to guide the user to the requested website [87], [88]. The communication between the client and the server is based on the HTTP method (GET/POST method); to perform an action the client requests the resource from the web server and then the server responds [89], [90]. Malicious URLs is malicious software that uses an encryption method for some sections of a URL to bypass detection by signatures [91] and attempt to inject malicious content such as via SQLI and XSS within the link [90].

Email, or electronic mail, is used to communicate by sending documents file, folders, links and images via a particular port, which has two parts: a body and a header [92]. The security threat relating to sending email between users has different aspects, one of which is attacks, where the attacker spreads malicious content by sending email to a targeted system [93].

A website owner usually has the goal of searching for approaches to attract visitors, which may provide more benefits for their website such as the potential financial gain. Advertising officers in companies aim to exhibit clean codes that means it does not contain any malicious code in their advertisement banners to achieve their goal of convincing other website masters to support their services without the weaknesses or implementation errors that can lead to an unexpected event, and to avoid ruining their reputation. Therefore, website owners put their trust in the company officer to utilise a safe code in their banner, as they will be inserting this code into JavaScript code for the webpages [94]. Normally, JavaScript code creates a different banner for each visitor, but this approach may be used to present a vulnerability; for instance, this code could be targeted by malicious code added by an attacker, which may cause substantial harm for large numbers of users without their knowing [95].

2.6 Malware Behaviour of Honeypot

Use of honeypots allows a malware attack scenario to be achieved even without access to vulnerable software [96]. To identify the spread of malware in a honeypot, Kaur and Kaur [97] described the detection of malware programs linked to webpages. Honeypots are security devices that detect malicious webpages on a network. Cabaj and Gawkowski [96] deployed honeypots at the Institute of Computer Science to test their practicality and observed that the number of attacks was correlated with the complexity of the web application on the honeypot.

2.7 Honeypot

To deflect malicious intent, security experts need to infer or observe an attacker's approaches, strategies and techniques. According to Spitzner [98], "a honeypot is a security resource whose value lies in being probed, attacked, or compromised" (p.58). Döring and Erbs [99] provided a substitutional definition as "a computer connected to a network. It can be used to examine vulnerabilities of the operating system or network" (p.1). The purpose of honeypot technology is to gather information in visible form and testimony about attacks and attackers by monitoring the machine being assailable, instead of that, it can mimic the operating system services to discover attacks. Therefore, any internet traffic a honeypot receives via the network or external environment might be an attempt to intrude or to break into the system, because the honeypot works as supplemental software created by the owner of the network that should not experience any external traffic unless someone is trying to attack it. Figure 2.2 shows an overview of honeypot technology.

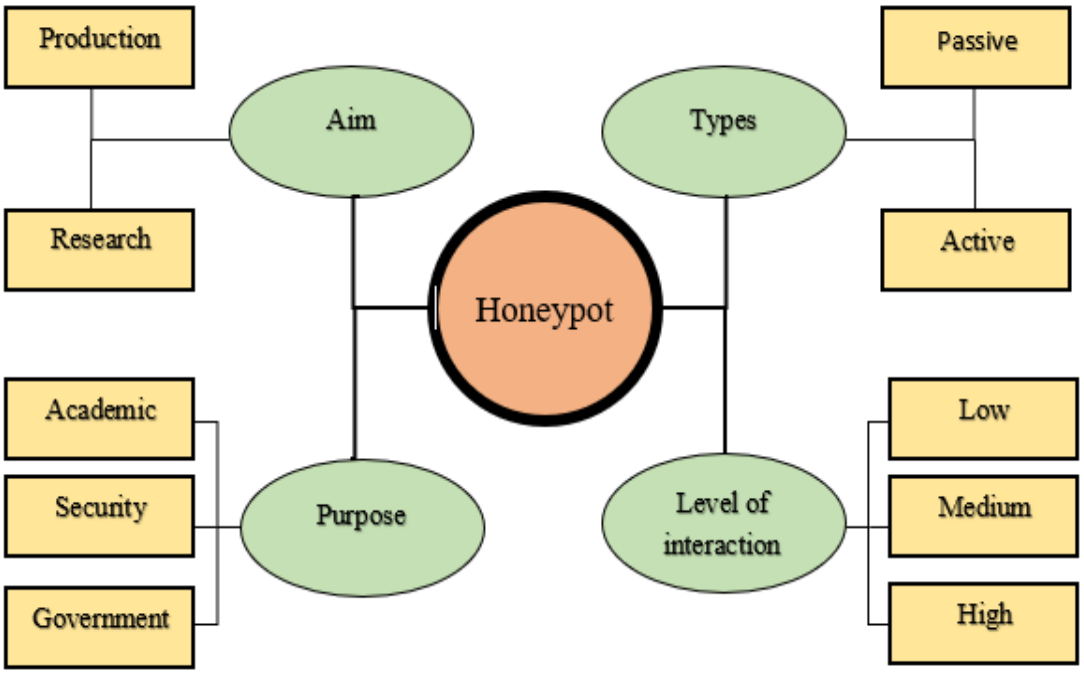


Figure 2.1: Overview of honeypot technology.

Typically, honeypots are of two main types: passive and active [100]. These are designed differently but share goals such as developing the security level of an operating system and network to collect data about an adversary's behaviour and attitude.

Passive honeypot: This type aims to set up a vulnerable system or simulated system; honeypot software inspects a system by monitoring a hacker's movements to observe any attacks such as on the firewall.

Active honeypot: This type involves searching for attackers instead of waiting for them passively.

There are two main aims of using honeypots for searching or production [101]:

Research honeypot: This is a traditional honeypot used to recognise strategies used by attackers and techniques in the intruder community. Normally, it is used to detect tools and thus learn how attackers work, to gain information about their methods. It is focused on discovering the aims of the attacker and in this way, finding and discovering new malware and attacks being used.

Production honeypot: This is a honeypot used within an internal network, such as an organisation's environment, to protect the system and reduce attack traffic, rather than simply watching how attackers work and gathering information. It monitors intrusion activities using real-time alerting, which helps to create a counter measure against predicted dangers.

To be specific, the information gathered by a honeypot or honeynet (more than one honeypot) is described and can be utilised for different goals by different organisations, such as:

Academic purposes: An example of this is a campus network. Network administrators collect data to help researchers in their papers into viruses, trojan horses or attacker tactics. The campus network presents a variety of external and internal services that might attract multiple attackers [102].

Security purposes: Some institutions and companies use honeypots to identify malicious intent, which helps them to understand and eliminate risks from malicious activities and create a secure environment by generating a database of new viruses.

Government purposes: Some government ministries use honeypots to help them frustrate malicious attackers and determine their location.

According to Peter and Schiller [103], using honeypots has positive aspects that can be briefly summarised such as there are fewer false positives as no valid traffic is captured by honeypots, honeypots can detect unknown attack signatures, only illegal movements of malware are discovered, so there is no requirement to store huge logs, and honeypots use an encrypted environment.

According to Akkaya and Thalgott [104], the advantages of honeypots include that using a honeypot can motivate researchers to find new security solutions by detecting new malicious attacks, any computer or system can be used as a honeypot, so there is no need for an additional budget to build one. However, there are some disadvantages: A honeypot can be used by unethical users to subvert or compromise other systems. In some cases, honeypots can be recognised by attackers; expert hackers can use fingerprinting to identify them.

Honeypots can work in different structures, either in a real computer or by mimicking a variety of operating systems. These structures can create some risks for a network, but may also bring benefits; nonetheless, honeypots should be executed carefully. According to Mokube and Adams[105] , a honeypot should be used in a legitimate and authorised way, and some countries and cities have their own rules about performing and using honeypots to avoid risks to their networks. There are some general and legal issues that should be taken into consideration when using a honeypot, which include privacy and liability.

A honeypot system can be compromised by attackers to attack other systems on the same network. This is called an uplink liability. There are some legal issues relating to such liability that should be considered; for example, if a compromise occurs, what

preventative measures should be taken to prevent uplink liability. Most the honeypot problems can be avoided by having the correct implementation and architecture (or building) within the network. For instance, experts recommend isolating a honeypot system from the production network to reduce risks such as uplink liability [105]. Recently, researchers have been focusing on active honeypots rather than passive ones, which involve passively waiting for attackers rather than going and searching for them. Honeypot technology has one type that is called a client honeypot.

2.7.1 Active Honeypots (Client Honeypots)

In recent times, the main goal of intruders has been causing harm to client applications (web application) such as web mail and web browsers; in the past, they aimed to subvert servers. Accordingly, techniques are required to deal with this issue by determining attacker tactics and seizing malicious code to increase security. An active honeypot is one way to improve security. It is activated differently from a passive honeypot, which works by waiting passively for attack attempts. Joho and Riedl [106] described **"**an advanced honeypot system. In contrary to traditional honeypots that undergo passively all attack attempts, active honeypot systems actively react to them" (p.55).

2.7.2 Architecture of Honeypots

There is a general architecture for any active honeypot [104]; it involves the following three basic parts (see Figure 2.2), which should be considered with caution:

1) **Honeypot server:** This is a server that sets up a virtual honeypot(s); it is not an actual honeypot.
2) **Management server:** This is the main part of the system structure, as it has the responsibility for collecting the data from a honeypot. The collected data are then processed and can be used to determine the quality of the honeypot:
    a) Data collection: The management server must gather as much data as possible about an attacker's activities.
    b) Data processing: This activity provides some information such as the experience level of the intruder, how many minutes or seconds the attacker spends on the fake target, and the activity of the attacker and his tools.
    c) Reaction evaluation: The data collection and processing steps help to assess the quality of the system, identify the goal of the attacker and observe the attacker's techniques. This might make it possible for researchers to determine how successfully the system can impede the attacker. Finally, logging this

information will guide experts in how to upgrade and develop the active honeypot.

3) **Database:** This holds all types of information, such as the data that have been gathering from a honeypot or firewall. It works as memory for the honeypot and thus it is important to save all the collected data in a database. Some examples of the information that can be gathered in a database are the attack date and time, the IP address of the source, and any modification that occurs in the file system.

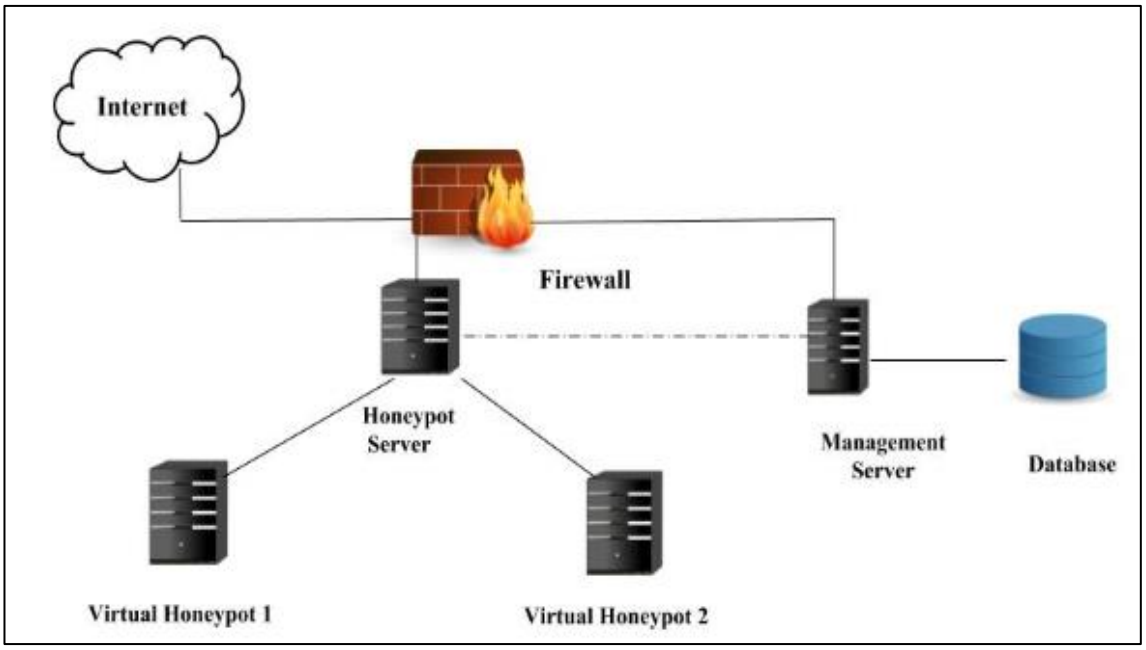


Figure 2.2: Architecture of an active honeypot.

2.7.3 Types of Honeypot

This section presents a brief overview of honeypot types and the differences between them, with examples of how to use each. For security researchers, the main purpose of using honeypots is to extract information about malicious intent from attackers to understand their activities by generating a weak configuration for the system. In addition to classification as either production or research honeypots, they can be classified based on the level of interaction between attackers and the system.

Each type will be presented in the next sections to achieve a variety of aims based on the following aspects: installation and configuration which will determine the time of the installation. Then, collection of information which the acquisition process for data is based on the level of interaction. Finally, level of risk which is based on the complexity of the honeypot. Honeypots can be low interaction, medium interaction or high interaction.

Low-interaction Honeypots

This type mimics a client or a real system. It imitates only services that are difficult for an attacker to utilise, to avoid having complete access to the honeypot [107]. It provides limited communication between the attacker and the honeypot. As a result, it collects limited information. A low interaction has a simple structure that makes it easy to install because it is not a real operating system. Therefore, it is a kind of program with services to store the logs. The level of risk in a low-interaction honeypot is very low. Most of low-interaction honeypots can only log the following: time and date of the event, the IP address of the source and destination, and the port of the connection for the source and destination. There are many low-interaction honeypot software programs, but common examples are:

1) **HoneyD:** is an open source program created by Niels Provos (a security investigator). It can be performed on Unix or Linux operating systems and its main purpose is to discover unauthorised activity within the network. Generally, any attempts coming through an unused IP address are considered malicious. HoneyD is able to mimic a variety of operating systems at the same time on a network [107]. It has two outstanding disadvantages. The main one is that advanced intruders may realise that they are not in a real operating system because it uses a non-existent address or unassigned IP address. Also, HoneyD can detect only known attack patterns, because it uses a signature database.
2) **Specter:** simulates the IDS, but the design of this software was created to be a honeypot to gather information and testimony against intruders. This software has some merits [103], It provides fake information for attackers to access and download, this information gathers hidden evidence because it makes marks on the computer of an intruder. Therefore, it tries to gather data about each attacker.

Medium-interaction Honeypots

These are also called mixed-interactive honeypots [108]. Honeypots considered to be medium interactive are a little more complex than low-interaction honeypots, but less developed than high-interaction honeypots. An example of a medium-interaction honeypot software is:

- **Honeytrap:** It is a program that generates network services that are not predefined. It considered as a medium-interaction honeypot, while Seifert et al. [109] reported that Honeytrap is a low-interaction honeypot. Known and unknown attacks over a TCP network can be monitored by Honeytrap.

High-interaction Honeypots

It can be applied in a physical computer system and used to identify malware; for example, worms and viruses. It may be built as a controlled environment that can detect attack events [110]. Ma et al. [111] utilised a high-interaction honeypot to analyse SQLI attacks. They found that the honeypot system assessed in analysing the SQLI with providing attack graphs. An example of a high-interaction honeypot software is:

- **Capture-HPC:** This security system, also known as Honeyclient, can help the security experts to capture the malware traffic and can find the malicious activities on compromised system [112].

2.8 Types of Malware

There are different types, patterns and techniques of attacks that allow hackers to gain control over a system or machine and generate increased damage [113]. This section aims to provide an overview and definition of attack types including viruses, worms, spyware, backdoor, Rootkit, Trojans and adware. Application and operating systems have the potential to be affected by malware code, links and script. Malware was described by Skoudis and Zeltser [114] as "a set of instructions that run on your computer and make your system do something that an attacker wants it to do" (p.2). Once a user's machine is compromised, an attacker can download any kind of malware from the following, as explained and classified by Sikorski and Honig [115].

A virus is the most common and traditional model of malware and has various patterns. Viruses can damage a system at different levels, from changing desktop backgrounds to modifying hard drives. They are programs that can attach themselves to files or applications and then transfer themselves to other computers, causing subversion of the entire network [116], [117]. However, a virus demands the user's intervention to run in the system, which means the user must initiate activation of the infected program.

Worms are identical to viruses as a malicious code but do not need human interaction. A worm can install itself automatically, which makes it more dangerous and more difficult to identify [118]. Therefore, worms are still used by attackers because worms help them to affect other computers on the same network [119], [120].

Spyware includes software or programs formed to gather sensitive data from a user's computer, such as passwords, or to change the computer's settings and website logins. It can also check the user's browsing activities, such as search keywords and history [121], [122]. Spyware can be secretly installed on a user's system to monitor it.

Backdoor malware normally works based on previous method that was done by users, such as compromising a system using viruses or worms. It is installed to admit easy access in the future or to allow adversaries entry [123].

Rootkit is software set up to provide remote access to adversaries, allowing them to alter files or applications. It works as a group of tools installed to obtain access for the attacker as an administrator [124]. In this way they can manage and control the system by avoiding being discovered or visible. Rootkit works at three levels: the application, library and kernel levels [125].

Trojan works as a 'trojan horse', which has a unique way to trick the user into downloading or running software that includes malicious code. However, it appears as legitimate to make the user activate it [126]. When the victim downloads the software, the Trojan will be installed alongside it.

Adware is short for advertising-supported software, which is any program that automatically displays pop up advertisements on the computer's screen. Usually, it is provided to the user free or at a small fee to attract them [127]. Madware is a kind of adware malware that is found on Android phones and is used to steal private information by spreading ads via the app store such as through Google play [128], [129].

2.9 Summary

This literature review has shown that there are many issues regarding the malware behaviour of webpages. The chapter can be summarised into four main sections: identification of malware behaviour in webpages; classification of several machine learning techniques; feature selection in malware websites; and malware behaviour of honeypots. The first section discussed the detection of phishing websites based on URL, and several methods used by researchers were outlined. The second section explored techniques to classify websites into categories as legitimate, phishy or suspicious. The next section identified feature selection in malware websites. Finally, the chapter focused on malware behaviour in honeypot data and provided key information about active honeypots, types of honeypots and types of malware. The research methodology and design for this study are presented in the next chapter.

Chapter 3
Research Methodology and Design

3.1 Introduction

Chapter 2 reviewed the substantial literature on identifying malware of webpages, classification and machine learning techniques to identify malware behaviour, feature selection in malware websites was outlined. Therefore, the honeypot technology was described in detail. Malicious activities are a serious threat and demand examination [130]. This chapter identifies the research methodology and design to be used in this research. The ensemble method used for data analysis is a machine learning technique involving multiple classifier systems; that is, ensemble learning uses several models and the final model is the most accurate and better predicted [131]. The ensemble method with random tree as its base classifier produces one of the best models when data are multivariate or has repeated instances [132]. A bagging (bootstrap aggregation) algorithm was chosen for this study, it is a machine learning algorithm designed to improve the efficiency of classification process. Bagging provides better accuracy because the number of splits in the decision tree is 100 which improves the classification rate and to train multiple predictors [133]. Bagging can classify binary class, date class, missing values, nominal class, numeric class, unary class and empty class. Various techniques can be utilised prior to choosing the type of base classifier. In this study, cross-validation and percentage split also known as test analysis in WEKA were employed.

The remainder of this chapter is organised as follows: Section 3.2 presents the research methodology adopted, which is classified into five parts: identification of the problem; data collection; data description; data pre-processing; and data analysis. Section 3.3 presents the research design, which consists of four subsections: the honeypot deployment method; features in botnet data; features in data; feature selection; malware behaviour identification; and prediction of malware attack. The chapter is summarised in Section 3.4.

3.2 Research Methodology Adopted

The research methodology was selected based on the objectives presented in Section 1.1. The research was conducted using an experimental approach. When data classification/feature selection is involved, an experimental phase is required to achieve

highest accuracy because different classifiers may produce different accuracy. Experimental analysis enables analysis of data, as required by several classifiers—in this study, bagging and different base classifiers: decision stump, hoeffding tree, random tree, j48, RF and REPTtree. Random tree as a base classifier for bagging was chosen for the remainder of the analysis as it achieved better accuracy in experiments. This research used a hierarchical process model with five phases, each with a specific task. This methodology helped the researcher to follow the steps as required to complete the study. The first stage of the methodology involved understanding the problem to achieve the study objectives. Data collection required data description. The data pre-processing stage prepared the data to train machine learning algorithms. Data analysis was the final stage in identifying features. Figure 3.1 summarises the research methodology adopted.

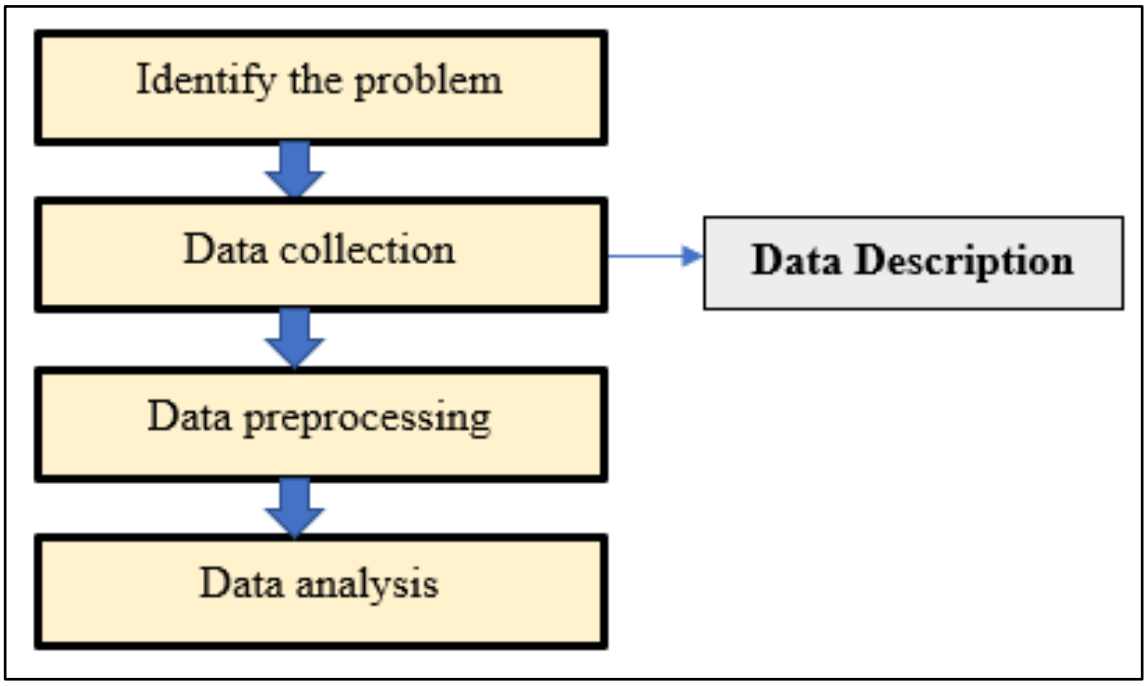


Figure 3.1: Phases of research methodology adopted.

3.2.1 Identify the Problem

The first phase that took place was identifying malware behaviour, which is still a research challenge. This problem was outlined in Section 1.1.

3.2.2 Data Collection

The existence of a massive amount of internet data with little information regarding the expected features of malware means that the identification of malware behaviour is not easy. Phishing, botnet and honeypot data were chosen for this study as they provide

a description of malware properties such as IP address, URL, email and abnormal URL. The second phase executed was the data collection step. Section 3.2.3 explains the attributes and number of instances of the three datasets collected for the study, which were as follows.

Phishing data: Phishing data were downloaded from the UCI Machine Learning Repository (https://archive.ics.uci.edu/ml/machine-learning-databases/00327/) In this dataset, the features that proved to be effective for predicting malware websites were studied.

Botnet data: Botnet data were taken from the UCI Machine Learning Repository (https://archive.ics.uci.edu/ml/machine-learning-databases/00442/) to capture network traffic patterns.

Honeypot data: Honeypot data were collected for several months to enable an in-depth analysis through application of MHN software; this phase involved a testing scenario to make it as real as possible. The detection of traffic, malware activities and the collection of data was possible after installing the operating system (Ubuntu 16.04) and software (MHN). The data were collected for the study after running the MHN software.

3.2.3 Data Description

The scenarios in the experiments were based on three types of data:

Phishing data: The number of attributes was 31 and there were 11,055 instances without any missing values where the data type was integer (see Figure 3.2). For data analysis, the Weka tool was utilised [134], it was used to generate an Excel file for the phishing dataset to enable examination and analysis using charts. Excel worksheets were utilised to graph results. Phishing data contains three types of behaviours: Phishy, suspicious and legitimate. Phishy behaviour is an attack designed to steal users' confidential information, it may cause substantial financial harm, phishing websites are those which designed to hijack websites and obtain users' sensitive information [135]. It could be used with a high-quality visual deception by the attackers [136]. Suspicious behaviour is an activity that may consider as phishy and could has malicious codes and links [137]. It also known as a suspicious URL, is one that is not obviously either malicious URL or non-malicious URL [138]. A legitimate webpage is a page with clean source code that means it does not contain any malicious code in its source code [135].

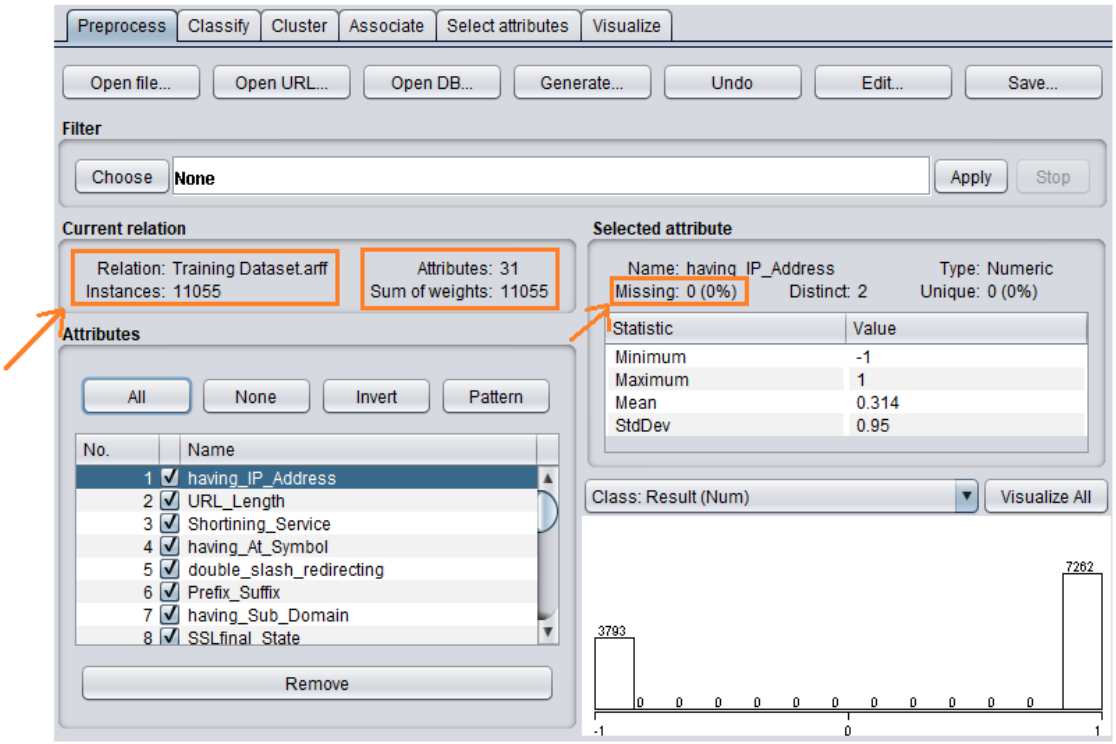


Figure 3.2: Information of phishing dataset.

Botnet data*:* The dataset had 115 attributes for each type of attack: 40,395 and 13,111 instances of benign for Danmini and Ecobee, respectively. Danmini is an anti-hacking tool that is similar to a hardware device. As this device is not resistant to being hacked, it was necessary to investigate types of malware attack it experiences to identify malware behaviour of webpages. If this device is safe, then the computer is safe too. Ecobee is a device model similar to Danmini; the device was used with the botnet dataset as well. Both Danmini and Ecobee are successors of IoT technology and devices [22]. The data type was numeric for Danmini and Ecobee (The arrows in Figure 3.3 and Figure 3.4 indicate the information of benign feature in Danmini and Ecobee, respectively). In this research, gafgyt attacks 'A type of botnet attacks that was found in Danmini and Ecobee' was studied. It is also known as BASHLITE, an IoT botnet attack that mostly compromises Linux servers using brute force attack [139].

This study focused only on features based on relevance or accuracy (see Appendix A.1 and A.2 for more figures of the rest features. Appendix A.1 shows the botnet dataset of Danmini and Appendix A2 has the botnet dataset of Ecobee. Both appendices have the same types of attacks: Combo, Junk, Scan, TCP, and UDP; all features have the same number of attributes with different number of instances based on the number of data that were collected. All data are numeric data). Weka (Waikato environment for knowledge analysis) software is a data mining tool written in java and used for data analysis to find

the accuracy of each feature and summarize them with graphical user interface. Weka uses different algorithms to classify data, it is open source for data preprocessing, classification and feature selection [140]. The features that were analysed in this study by weka: benign that is consider as a non-malicious traffic data [141], it is a normal traffic patterns. Combo refers to sending spam data to an email and opening connection time [22], junk is sending spam data, scan is to scan the network for any vulnerable data [142], TCP and UDP—using the bagging method and random tree as classifier; the classifier evaluation options were cross-validation and percentage split that known as test analysis.

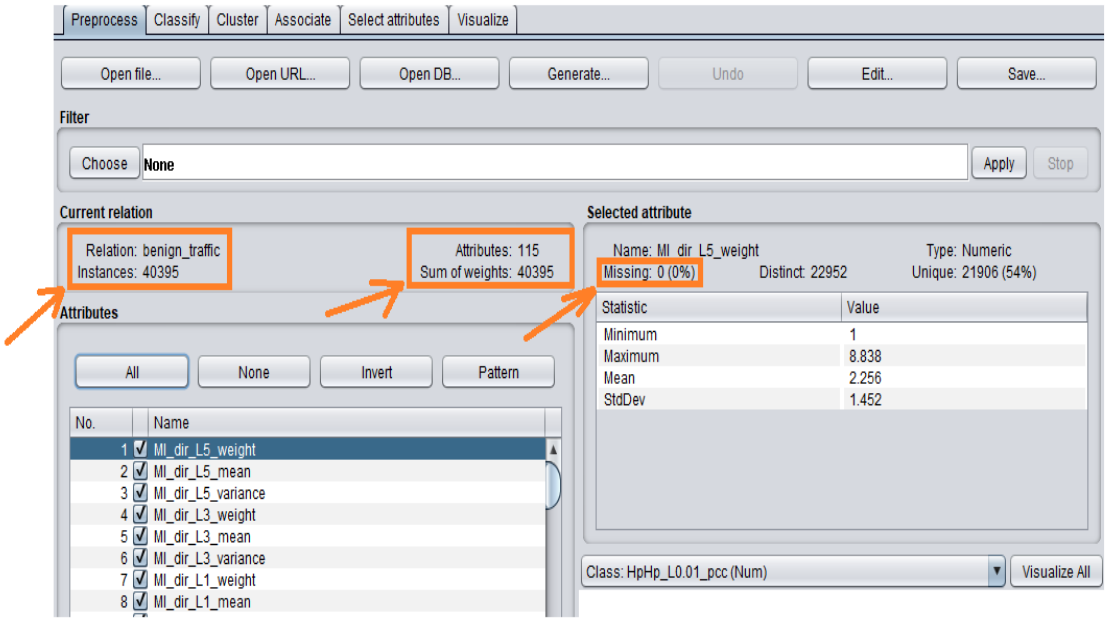


Figure 3.3: Information of benign feature detected in Danmini.

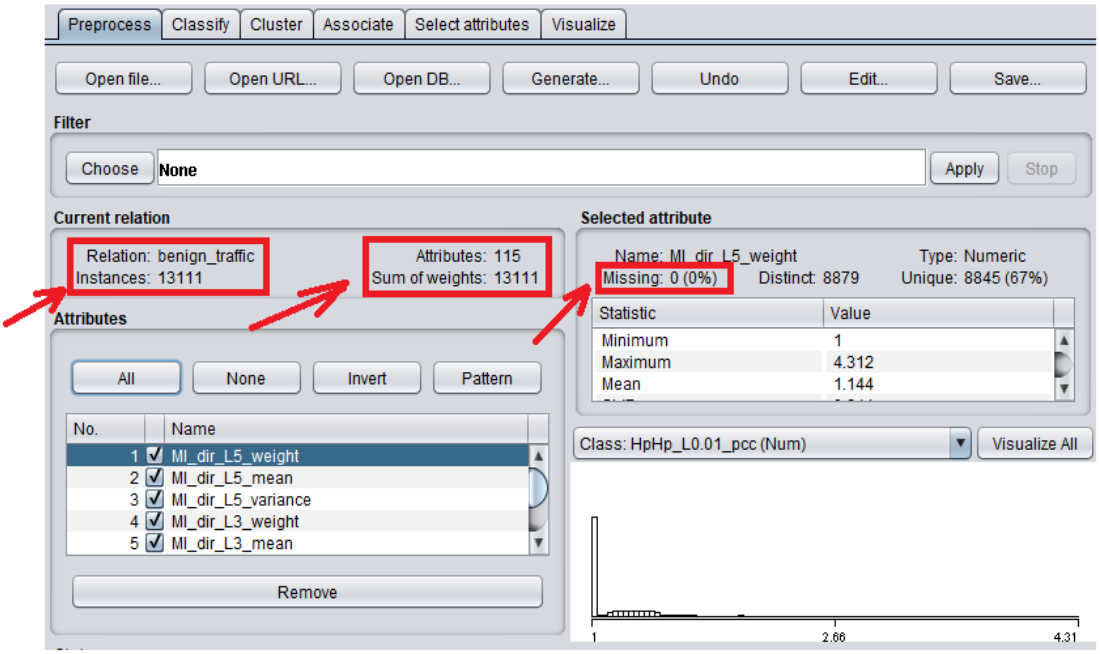


Figure 3.4: Information of benign feature detected in Ecobee.

Honeypot data: The dataset includes several types of attacks, such as IP address, port, protocols, usernames, passwords and requested URLs. There were around 80,462 attacks for the three types of honeypot sensors used in this research: Snort, Kippo and Glastopf. The reason for using Snort as one of the sensors was to collect more malicious information, it is the most common IDS to detect attacks. Snort is open source IDS that is used to discover and scan if someone is trying to get into your network; then it can log the alerts to a database [143]. Kippo was selected as one of the sensors in honeypot to identify different and unique data, such as the most used usernames and passwords, this was important to evaluate the skill level of attackers by logging their brute force attacks. Glastopf is one of the web application honeypot sensors that was deployed via the MHN server. It can mimic web vulnerabilities to collect data about attacks that are targeting the web server such as SQL injection which considered as 'unknown' pattern rather than 'SQL' and this is one of its limitation. It focuses on reguest-urls which the attacker's requirement are, and it can track the attack pattern. Glastopf is open source and it is free of cost, it captures data such as source IP address and time of event [144].

Figure 3.5 presents the payloads report for Snort where the attack's features found. Appendix A.3 provides for additional figures of Kippo and Glastopf including an example of payload report of Glastopf events (data) that explains the time of the attack, the pattern, the source IP and the request- URL. Also, it illustrates the top 10 passwords, top 10 usernames and the top 10 attackers that were registered by Kippo). Analysis of the data to test the proposed software was based on two types of machines: a physical machine

(laptop) and a VM. The Windows 10 (64-bit) operating system was installed on the physical machine and the Linux operating system (Ubuntu 16.04) was loaded in the Oracle VM VirtualBox. The aim of using a VM was to have a more secure environment in the case if the VM is infected. The Ubuntu 14.04 operating system was used to install MHN using a cloud web server hosted by Linode from the web address https://www.Linode.com/docs/platform/billing-and-support/Linode-beginners-guide/.

MHN was downloaded from https://github.com/threatstream/mhn.git and was controlled remotely using a secure shell (SSH) service. Analysis of malicious activities was performed using RStudio to determine the number of occurrences of each feature. Excel program was used as well to save data, Table 3.1 illustrates the system set-up.

Table 3.1: Pre-existing condition for this experimental study.

No	Component	Type	Description
1	Windows	Windows 10 operating system	Installed on physical machine (HP laptop)
2	Ubuntu	Linux operating system (Ubuntu 16.04)	Installed on VM VirtualBox
3	Ubuntu	Linux server (Ubuntu 14.04)	Cloud web server node hosted by Linode and located in Tokyo
4	MHN	Honeypot	MHN installed on the web server

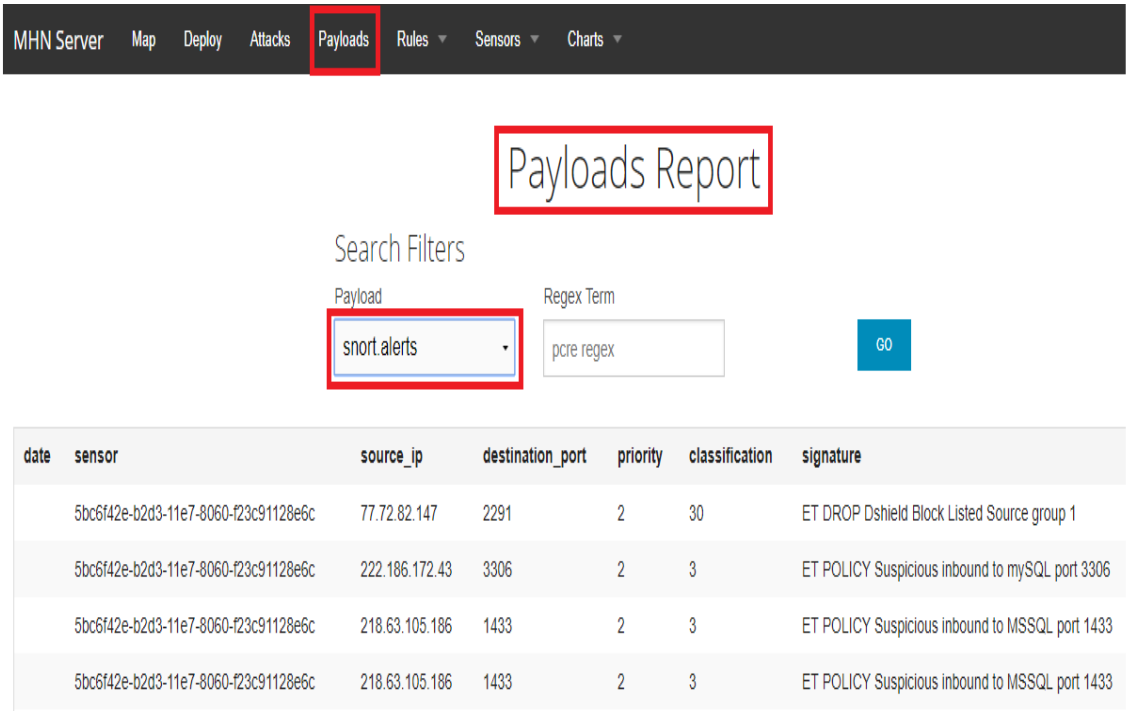


Figure 3.5: Payloads report of Snort.alerts.

3.2.4 Data Preprocessing

The data features chosen were those that were most relevant based on the literature review; the data of the phishing dataset was changed to 0, 1 and –1 based on Mohammad et al. [19]. The authors defined 1 as legitimate, 0 as suspicious and –1 as phishy. Data preprocessing was important for choosing suitable features and differentiating malicious behaviour. For honeypot and botnet data, preprocessing was not required as there were no missing values or outliers.

3.2.5 Data Analysis

For data analysis, a very popular machine learning ensemble classifier was employed. Bagging as an ensemble was chosen as it mostly performs better than a single classifier, bagging ensemble classifier can be utilized to expand the accuracy of classification (where two or more classifiers are combined into an ensemble). Random tree was the base classifier in bagging based on the high accuracy obtained in comparison with other base classifiers with bagging (Table 3.2). In the initial experimental analyses that was done by the author using WEKA tool, several base classifiers were employed with the bagging ensemble method (one of the meta-algorithms in Weka tools): decision stump, hoeffding tree, J48, RF, random tree and REPTree. The empirical analysis showed that of all the base classifiers, random tree performed better; thus, random tree was chosen as a base classifier for bagging. It was noted that random tree was the best with accuracy of 88.22%,

which has more relevant ROC of 0.938 in terms of time in only 0.2 sec. The result of ROC near 1 is better (Table 3.2). Bagging is an ensemble technique and its performance depends on the base classifier it uses. 10-fold cross-validation was the test option chosen, it means that the dataset is divided into 10 parts where one for testing and 9 times for training which then produce the classifier for the data. Most of the other test options are used if there are lots of datasets as it is evaluated just one time.

Table 3.2: A comparison of classification algorithms.

Classifier type	Time	Accuracy	True positive	False positive	ROC area
Decision stump	0.05	73.0167	0.730	0.404	0.757
Hoeffding tree	0.83	85.9060	0.859	0.190	0.910
J48	1.08	88.6296	0.886	0.145	0.949
RF	11.17	88.6024	0.886	0.147	0.949
Random tree	0.20	88.2225	0.882	0.146	0.938
REPTree	0.92	87.8697	0.879	0.153	0.946

Note: ROC, receiver operating characteristic

3.3 Research Design

To address the research questions, experimental research was conducted to gather evidence, using three datasets: phishing website data; botnet attacks; and honeypot attacks. The study collected the first two datasets from the UCI Machine Learning Repository, and the third dataset from payloads of honeypot sensors deployed within MHN, which were Snort, Kippo and Glastopf.

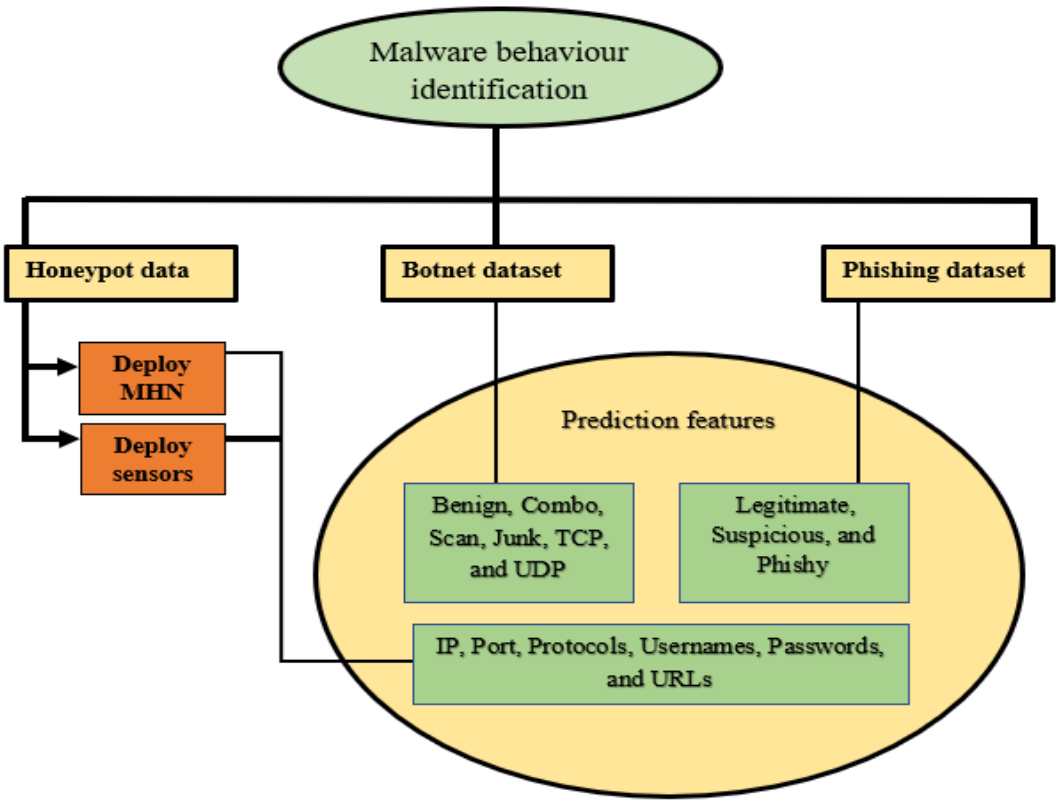


Figure 3.6: Research design phases.

This section provides a brief introduction to the research design, which was divided into four main phases. Figure 3.6 shows the phases included to achieve the research goals. Each of these processes is described to explain the research design.

3.3.1 Honeypot Deployment Method

This study employed a virtual honeypot that can emulate services or a network and was considered a low-interactive honeypot. In the preparation stage, the operating system and software characteristics were identified and set up for collecting the data from the honeypot. This included the web server setting, virtual machine ware (VMware), and the internet connection arrangements to prepare for the next phase, including any required software/packages to support the main goal of operating readily during the experiment. Some of the experimental settings, including the operating system and the main software used (MHN), were open source so were downloaded from the internet. The cloud web server (Linux server) hosting from Linode was closed source which required payment for the hosting, the hardware components (lab top and its belongings) were sourced locally.

Figure 3.7 shows the honeypot data collection process. The diagram illustrates that Linode web server was accessed remotely using Windows 10, located in New Zealand

(Auckland). The VMware software (Virtual Box) was installed in Windows 10; it was used to setup the MHN software and the sensors (Snort, Kippo and Glastopf) remotely in Linode server. Linode web server had Ubuntu 14.04 to be a platform for MHN and the geolocation was in Japan (Tokyo). MHN was setup in Ubuntu 14.04 using VMware. The data was collected using MHN's sensors and it was stored in Linode web server.

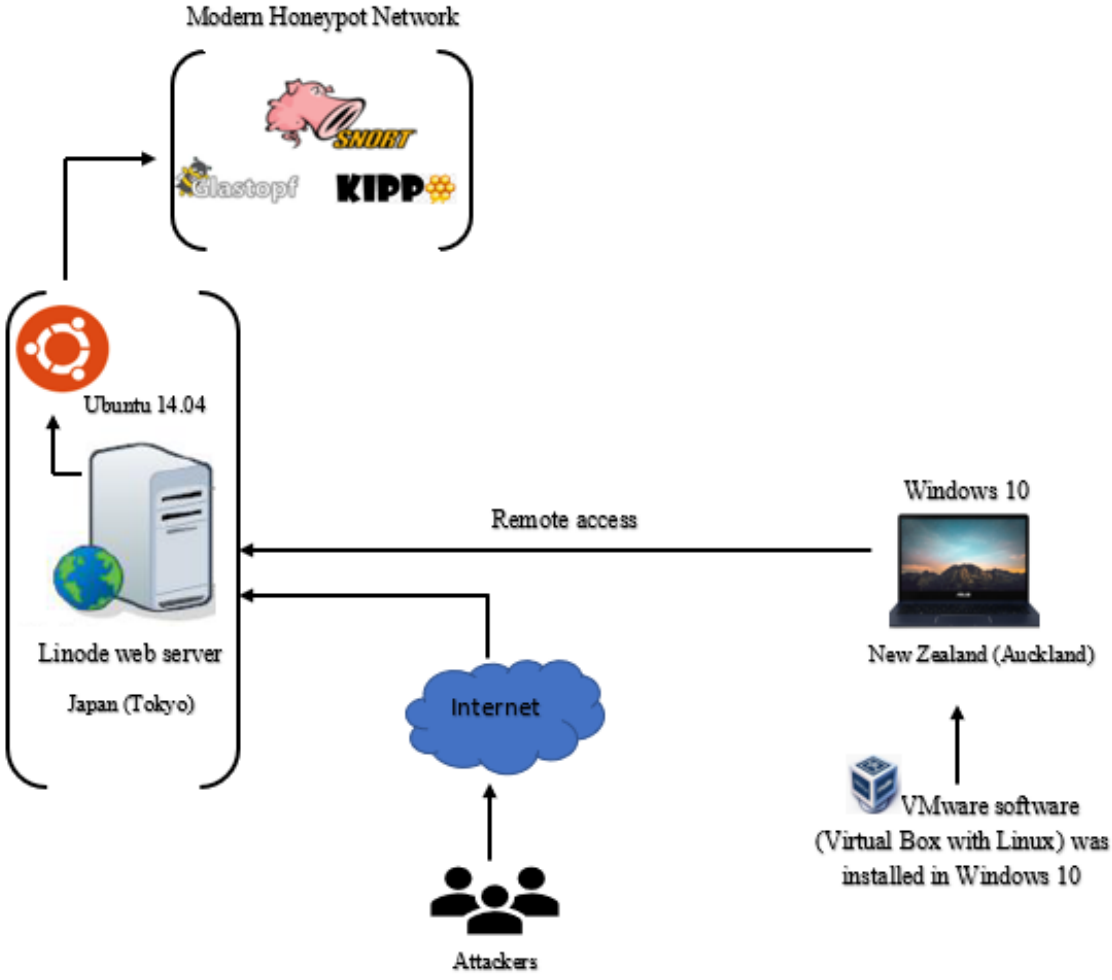


Figure 3.7: Overview of the honeypot deployment network.

To activate the honeypot (deployment steps), the following commands were required.

Deploy MHN management server based on Linux command prompt

MHN server designed to catch malicious attacks. Several Linux commands needed to deploy the MHN server on Linux, as shown in

Table 3.3 (more details are in Appendix A.4; it shows how the MHN was deployed using specific commands).

Table 3.3: Commands for deploying MHN.

Command	Description
Sudo apt-get install git	To assist in downloading the required program from the internet
Cd /opt	To move to option directory in the system
Sudo git clone https://github.com/threatstream/mhn.git	To generate a local copy of the required code to install program
Cd./mhn/	To change the directory to MHN folder

After installation of the MHN, the 'ifconfig' command was employed to find the public IP address generated by Linode for the web server. Once the IP address was copied and pasted into the search engine (Google chrome), the MHN webpage appeared. An email address and password were required to be entered and these were used during the configuration step to view the MHN GUI (graphical user interface).

Deploy sensors

One of the main icons of the MHN GUI, called 'deploy', was utilised to deploy the honeypots (sensors such as Snort, Kippo and Glastopf) by using special command offered in this icon for each sensor, as shown in Table 3.4. The malicious behaviour and traffic studied were taken from the honeypot's payloads.

Table 3.4: Information of the sensors (Honeypots) used in MHN.

Sensors	Description	Deployment command
Snort	An IDS to detect attacks	Wget "http://139.162.127.134/api/script/?text=true&script_id=3" -O deploy.sh && sudo bash deploy.sh http://139.162.127.134 nT6KfpXr
Kippo	A medium-interaction SSH honeypot to detect brute force attacks	Wget "http://139.162.127.134/api/script/?text=true&script_id=7" -O deploy.sh && sudo bash deploy.sh http://139.162.127.134 nT6KfpXr
Glastopf	A web server honeypot	wget "http://139.162.127.134/api/script/?text=true&script_id=8" -O deploy.sh && sudo bash deploy.sh http://139.162.127.134 nT6KfpXr

3.3.2 Feature Selection

The common properties or common nature of features in the three datasets were chosen to validate the results. Detailed explanations of other classification and relevance criteria with empirical results are provided in Section 4.3.

Features in Phishing Data

The ensemble method with random tree as a base classifier was utilised to classify features as legitimate, phishy or suspicious. The features were classified according the number of occurrences and their relevance. For example, the feature 'having IP address' was selected by bagging (random tree) based on its relevance and maximum number of occurrences of 8,000; while 3,793 occurrences were considered phishy. The other relevant features based on the accuracy and relevance are IP, port, request URL, Google index, submitting to email, web traffic, page rank, HTTPS (hypertext transfer protocol secure) token, abnormal URL, pop up window and links pointing to page.

Features in Botnet data

The features chosen for the botnet data were benign, combo, scan, junk, TCP and UDP. The relevant features based on accuracy and relevance.

Features in Honeypot data

The feature selection consisted of IP address, port, usernames, passwords, requested URLs, TCP, UDP and internet control message protocol (ICMP).

3.3.3 Malware Behaviour Identification

The analysis proposed in Section 4.3.3 shows that the most malware attacks occurred between October 2017 and February 2018; the attack period identified was similar to that of the McAfee threat map by month. The study shows the virus threat map created by McAfee Antivirus software in 2017 [16]. They found that attackers contacted an IP address or domain that was used to host a malicious document. These attacks were based on IP. Another report from the McAfee lab shows that malware attacks via spam were high during the last quartile (Q4: October, November and December) of 2017 and the first quartile of 2018 (Q1: January, February and March). The source of McAfee data are McAfee spam traps, crawlers and customer submissions [16]. The study confirms that the attacks were mostly from October 2017 to February 2018, as identified in the current study (Section 4.3.3).

The above discussion and the findings in the next chapter show that studying the behaviour of malware is crucial. The features selected and the time frame to collect honeypot data for this study are in line with the McAfee virus threat map. Thus, predictions about malware are required to ensure future safe webpages.

3.3.4 Predicting Malware Attacks

There are several ways to predict the behaviour of malware, including examination of previous malware data (generated from honeypot infrastructure). Assessing the legitimacy of IP, port, request URL, Google index, email, web traffic, pop up window, links, page rank, HTTPS token, abnormal URL, combo, junk, scan, TCP, UDP, ICMP, password and username can provide a better idea of how malware behaves and what areas are targeted in a malware attack. Further details and experimental results are provided in the next chapter.

3.4 Summary

This chapter presented the research methodology and design. The methodology includes identifying the problem, data collection, data description, data preprocessing and data analysis. The research design includes several phases: honeypot deployment, feature selection, malware behaviour identification and prediction of malware attack. The methodology involves three datasets to investigate the problem identified in this research.

The first dataset contains phishing data, which were identified and described in detail, and illustrated with figures. The second dataset contains botnet data, which were discussed based on two different devices (Danmini and Ecobee). The phishing and botnet data were downloaded from the UCI Machine Learning Repository and analysed using the chosen ensemble method with bagging (bootstrap aggregation) algorithm. In regard to the third dataset, set-up settings for MHN were highlighted. The main research findings and analysis are presented in Chapter 4.

Chapter 4

Research Findings and Analysis

4.1 Introduction

In Chapter 3, the research methodology and design were discussed in detail. A primary objective of this thesis is to identify key features identifying malware behaviour. With the growing popularity of attacking techniques, it is important to study malicious behaviour on webpages for their different features. The research question considered in this chapter is: *What research can be done to identify and analyse malicious webpage behaviour?* In particular, how can we recognize malware behaviour through the feature selection method being performed by attackers, based on their behaviours and activities? These questions are evaluated by comparing three datasets. *How can we predict malware attacks using information from the datasets?* This question is evaluated by analysing the datasets of Phising and Botnet using WEKA software.

Classification was employed to assign each data set to one of the predefined classes; classification helps to find related features/attributes in data. Attribute or feature selection is necessary to recognize the malware behaviour of webpages. Attributes were selected here to identify malware features and develop an understanding of malware behaviour. A study using three datasets was undertaken to achieve the main objective of this research. Two datasets were selected from the UCI Machine Learning Repository and labelled as phishing websites and detection of IoT botnet attacks (N BaIoT). The third was a honeypot dataset. Experimental details and an introduction to the empirical results are provided in Sections 4.2 and 4.3, respectively. The phishing dataset is presented in Section 4.3.1. Section 4.3.2 contains details of the botnet dataset, and the honeypot dataset is outlined in Section 4.3.3. Finally, the chapter contents are summarised in Section 4.4.

4.2 Experimental Details

In this empirical study, phishing, botnet and honeypot datasets were used for performance evaluation to predict attacks from phishing and botnet data. This study employed a bagging ensemble classifier (where two or more classifiers are combined into an ensemble). As random tree performed better, random tree was used as a base classifier in the ensemble. Table 4.1 illustrates the results for bagging with different base classifiers. In the initial experimental analyses, several base classifiers were employed with the

bagging ensemble method (one of the meta-algorithms in Weka tools): decision stump, hoeffding tree, J48, RF, random tree and REPTree. The empirical analysis showed that of all the base classifiers, random tree performed better; thus, random tree was chosen as a base classifier for bagging. It was noted that random tree was the best with accuracy of 88.22%, which has more relevant ROC of 0.938 in terms of time in only 0.2 sec (The result of ROC near 1 is better). Bagging is an ensemble technique and its performance depends on the base classifier it uses.

Table 4.1: Bagging algorithm (Random tree).

Classifier type	Time	Accuracy	True positive	False positive	ROC area
Decision stump	0.05	73.0167	0.730	0.404	0.757
Hoeffding tree	0.83	85.9060	0.859	0.190	0.910
J48	1.08	88.6296	0.886	0.145	0.949
RF	11.17	88.6024	0.886	0.147	0.949
Random tree	0.20	88.2225	0.882	0.146	0.938
REPTree	0.92	87.8697	0.879	0.153	0.946

Table 4.2 summarises the three datasets used in the study. Both 10-fold cross-validation and test (66% training data; the rest 34% of the data were test data) analysis show that the percentage of attacks in phishing data and botnet data were more than 89% average, and the total datasets contained less than 680,786 attacks. The honeypot infrastructure registered the number of malware hits; honeypot infrastructure was used as a test bed server; the number of attacks was 35% when the total data were not more than 80,462.

Table 4.2: Datasets used in this study.

Dataset source 1			
Phishing websites (UCI)	Percentage of attacks (10-fold cross-validation)	Percentage of attacks (test)	
Training dataset (Bagging)	97%	96%	
Dataset source 2			
Detection of IoT botnet attacks N BaIoT (UCI)	Percentage of attacks (10-fold cross-validation)	Percentage of attacks (test)	
Gafgyt attacks Danmini (bagging)	86%	88%	
Gafgyt attacks Ecobee (bagging)	89%	83%	
Dataset source 3			
Honeypot		Percentage of attacks	
Snort (IDS) Kippo (used to find the brute force attacks) Glastopf (web application honeypot sensor)		25% 74% 6%	Average 35%

4.3 Experimental Results

Three experimental results of datasets were described in this section. They are stated as following: Section 4.3.1 presented malware behaviour in phishing dataset; Section 4.3.2 illustrated malware behaviour in botnet dataset; and malware behaviour in honeypot datasets is shown in Section 4.3.3.

4.3.1 Malware Behaviour in the Phishing Dataset

Table 4.3 lists the 11 features that were examined to study the malware behaviour of phishing websites; these features assist in discovering phishing websites. The total number of attributes (features) relating to the phishing dataset was 31; in the current study only 11 features were selected based on [19], [20], [21]. Bagging (random tree) was used to compare the relevance of malware behaviour in terms of accuracy, between the three datasets. These features were chosen to distinguish websites as ‘phishy’, ‘suspicious or ‘legitimate’ based on Mohammad et al. [20]. If a result was returned as 1, 0 or –1, the website was labelled as legitimate, suspicious or phishy, respectively.

Table 4.3: Features examined in the phishing dataset.

Description	Feature
Having IP address	1
Port	2
Request URL	3
Google index	4
Submitting to email	5
Web traffic	6
Page rank	7
HTTPS token	8
Abnormal URL	9
Pop up window	10
Links pointing to page	11

Having IP address

Figure 4.1 depicts the nature of IP addresses based on the URL: if the IP address exists in the URL instead of the domain name, this typically means there has been an attempt to hijack or steal personal information; otherwise the webpage would be considered legitimate. In the phishing dataset, the results show that the number of URLs that did not have an IP address or it was masked, was 7,262 among the total behaviours considered legitimate as they returned 1 (see second paragraph above, this section). Only 3,793 URLs had an IP address and were classified as phishy, returning –1 as a result.

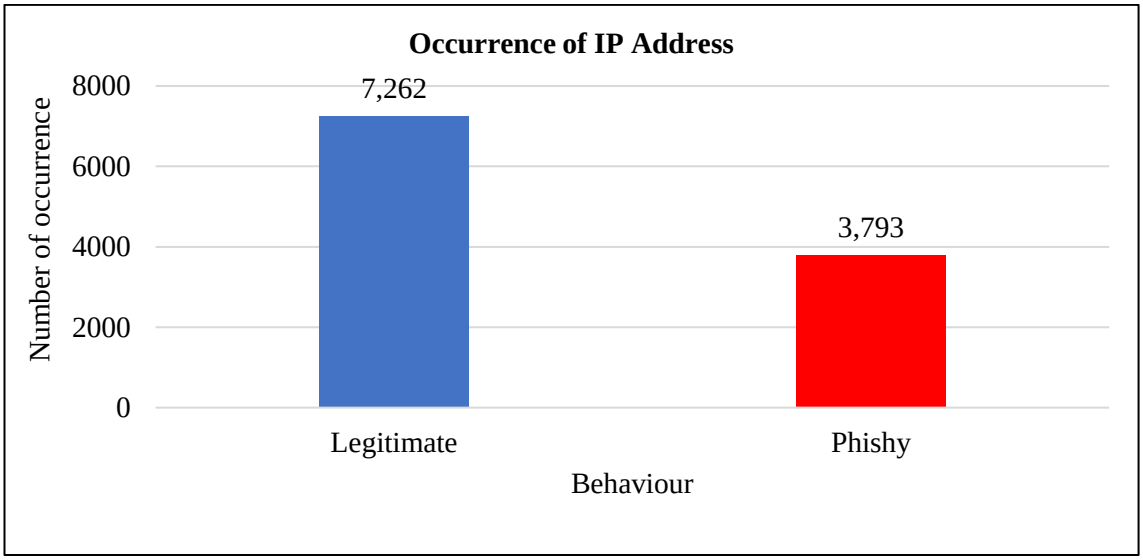


Figure 4.1: Number of occurrences of different behaviours for IP address feature in phishing dataset.

Port

Similar to the previous analysis, the number of malicious attacks through legitimate port browsing was 9,553 but there was still 1,502 phishy behaviour (see Figure 4.2). In this case, if a port is compromised, all hosted IPs are affected, if the IP address is affected then only specific webpages associated with that IP are affected, while the port remains safe. Malicious attacks on port are less common compared to IP address manipulation.

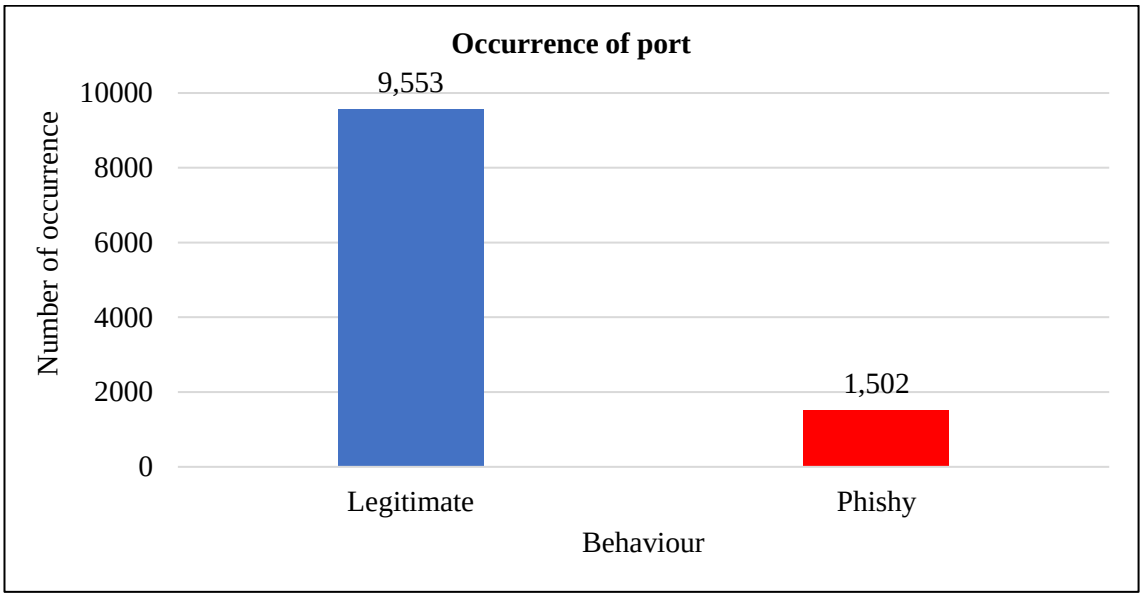


Figure 4.2: Number of occurrences of different behaviours for port feature in phishing dataset.

Request URL

Based on the previous two analyses, it may be concluded that malware occurrence through ports is relatively infrequent (1,502 times) while 'request URL' has a strong influence on malware behaviour, representing more than 40% of 11,055 web hits. In this study's experiments, the results classified 6,560 URLs as legitimate and 4,495 as phishy as shown in Figure 4.3.

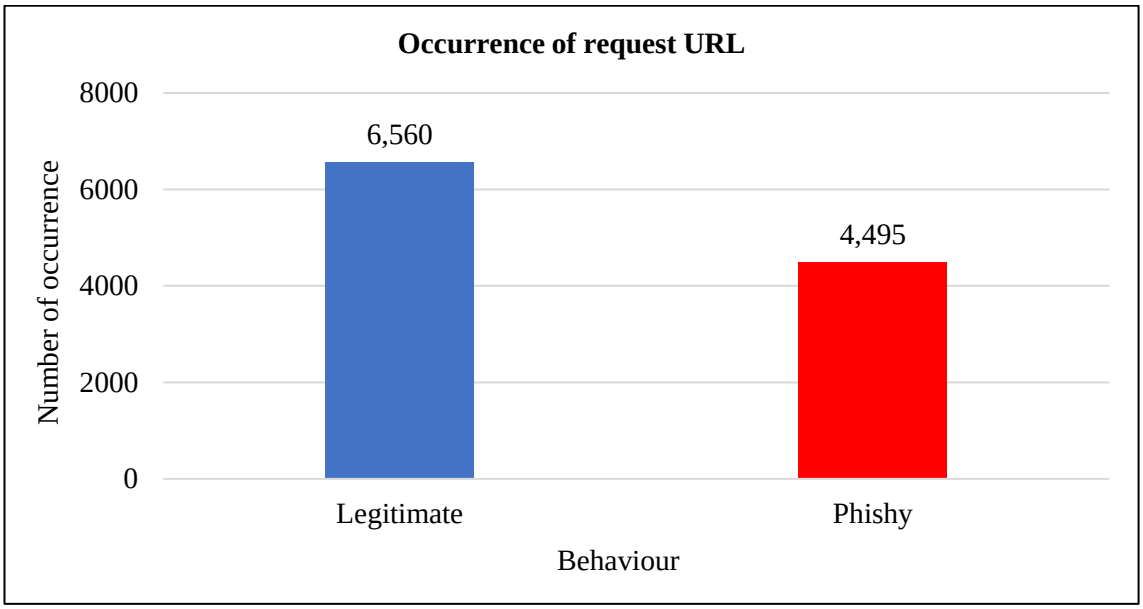


Figure 4.3: Number of occurrences of different behaviours for request URL feature in phishing dataset.

Google index

Based on the previous three analyses, it is clear that the number of occurrences of malware attack through Google index and through ports is nearly similar (37 more occurrences for Google index which is the difference in between them). In this study's experiments (see Figure 4.4), 9,516 of the URLs were shown to be legitimate, while 1,539 of the results were phishy.

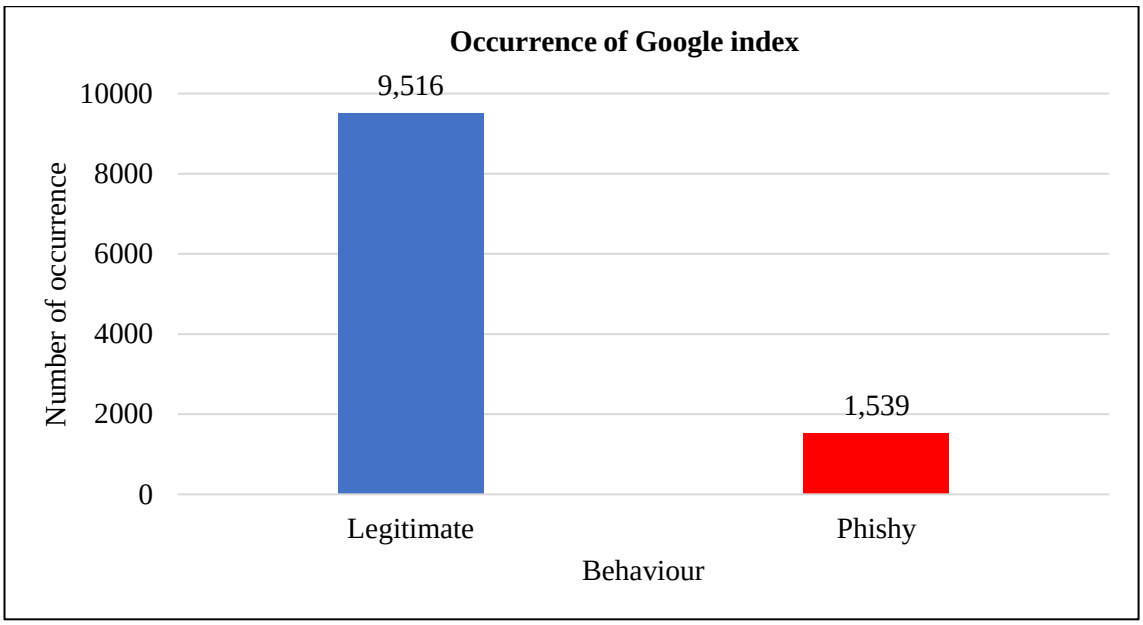


Figure 4.4: Number of occurrences of different behaviours for Google index feature in phishing dataset.

Submitting to email

Malicious behaviour using the feature 'submitting to email' led to more legitimate results (9,041) than phishy (2,014) (total number of hits, 11,055; see Figure 4.5). Compared with other features such as having IP address and request URL, the number of phishy sites was lower, but it was higher than malicious attacks through ports.

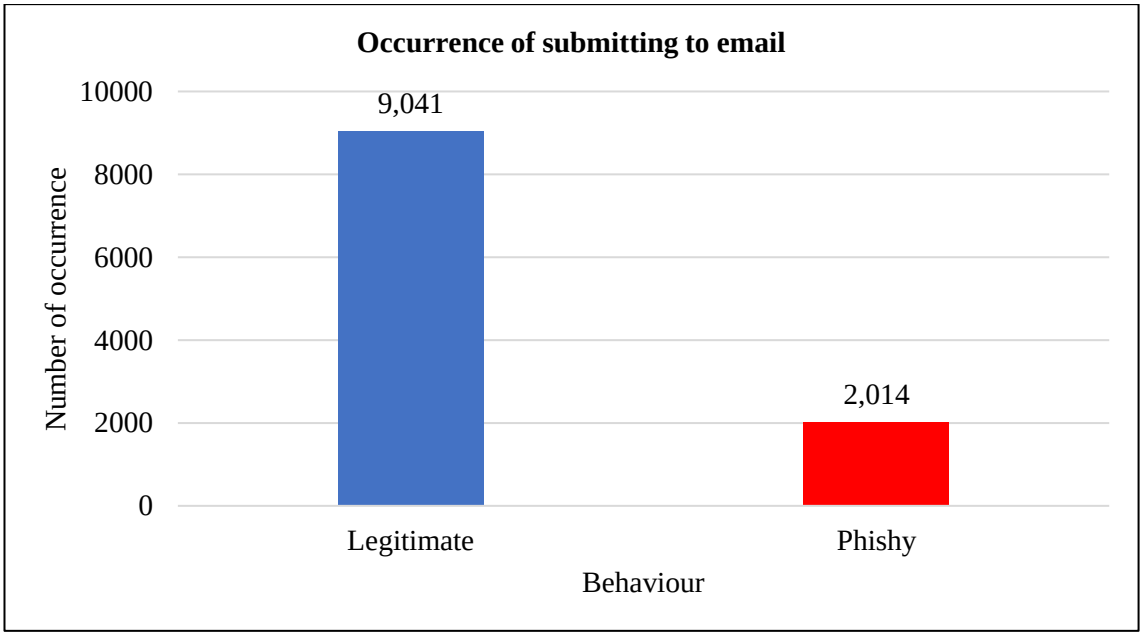


Figure 4.5: Number of occurrences of different behaviours for submitting to email feature in phishing dataset.

Web traffic

Based on the analytical results for web traffic, the nature of this feature is based on the number of visitors to webpage. In the phishing dataset, the number of webpages with malicious traffic was less than the number of legitimate webpages. The interesting finding in this feature was that suspicious never indicates whether it is legitimate or phishy. However, from this feature the number of legitimate webpages for browsing was only 50% of the total number of hits (11,055), which provides a clue that malicious behaviour may be closely related to web traffic, as presented in Figure 4.6.

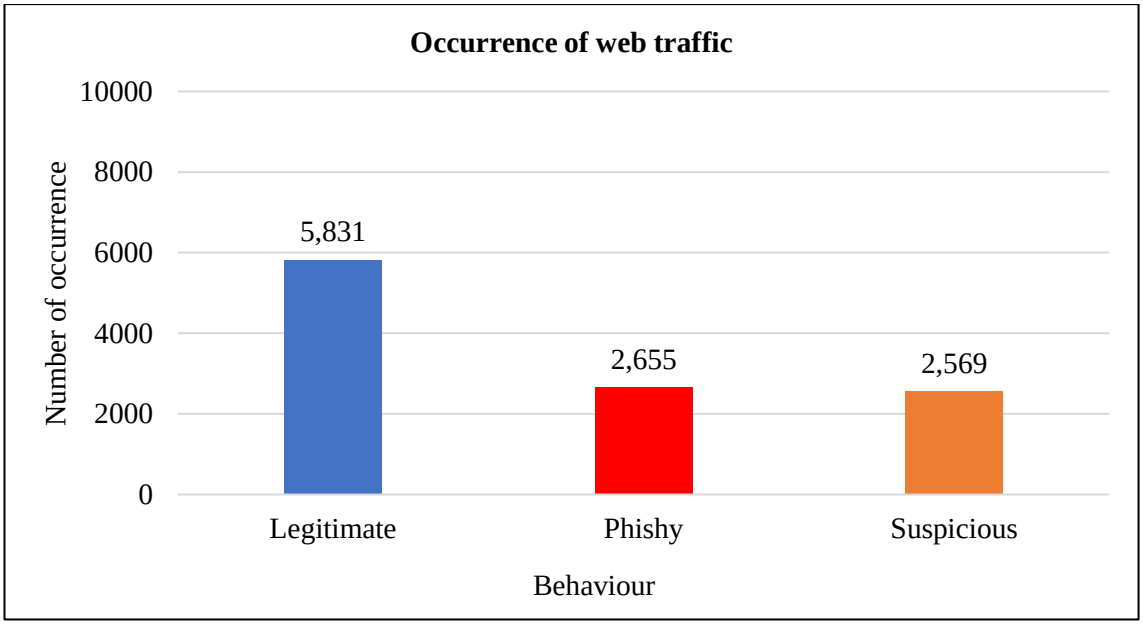


Figure 4.6: Number of occurrences of different behaviours for web traffic feature in phishing dataset.

Page rank

Compared with all other features in the phishing dataset, page rank provided the ability to discover the highest rate of phishy webpages, with 8,201 hits (see Figure 4.7), while Google index detected 1,539. However, the legitimate webpages were low with 2,854 hits, which was less than any other feature. This clearly shows that the higher ranked webpage may not be always safe as we think while Google indexed page are safer than ordinary page ranked in web browsing.

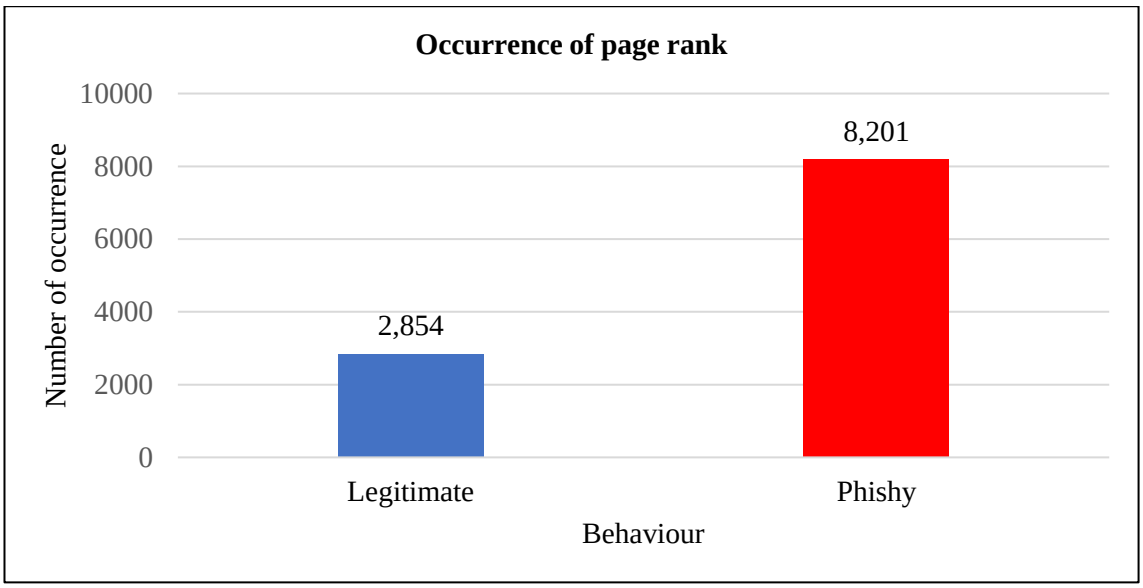


Figure 4.7: Number of occurrences of different behaviours for page rank feature in phishing dataset.

HTTPS token

Similar to previous results, HTTPS token resulted in very similar rates (almost 9,200) as legitimate webpages of features such as port, Google index and submitting to email. Turning to phishy results, these numbered 1,795, as illustrated in Figure 4.8.

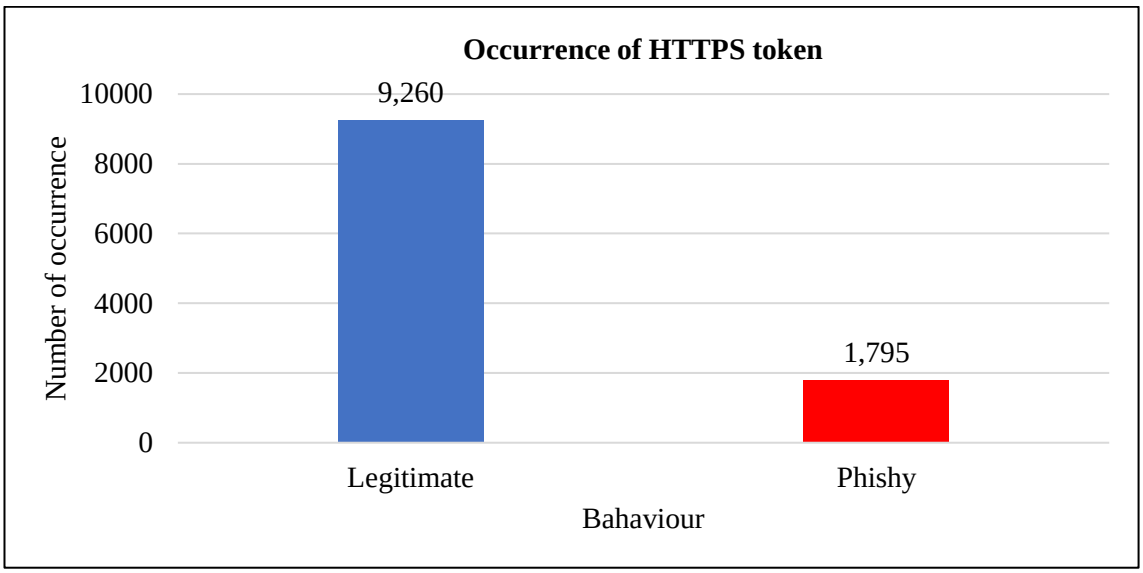


Figure 4.8: Number of occurrences of different behaviours for HTTPS token feature in phishing dataset.

Abnormal URL

The nature of this feature is based on the identity of URL. If a URL included the host name, it was considered legitimate; otherwise, it was considered phishy. As shown in

Figure 4.9, the number of abnormal URL that were legitimate was 9,426, which is one of the features that has high number of occurrence compared to some previous features; thus this is a strange result that requires further study. Only 1,629 hits were phishy, which also needs further investigation; however, his study was limited to identifying the malware behaviour of webpages.

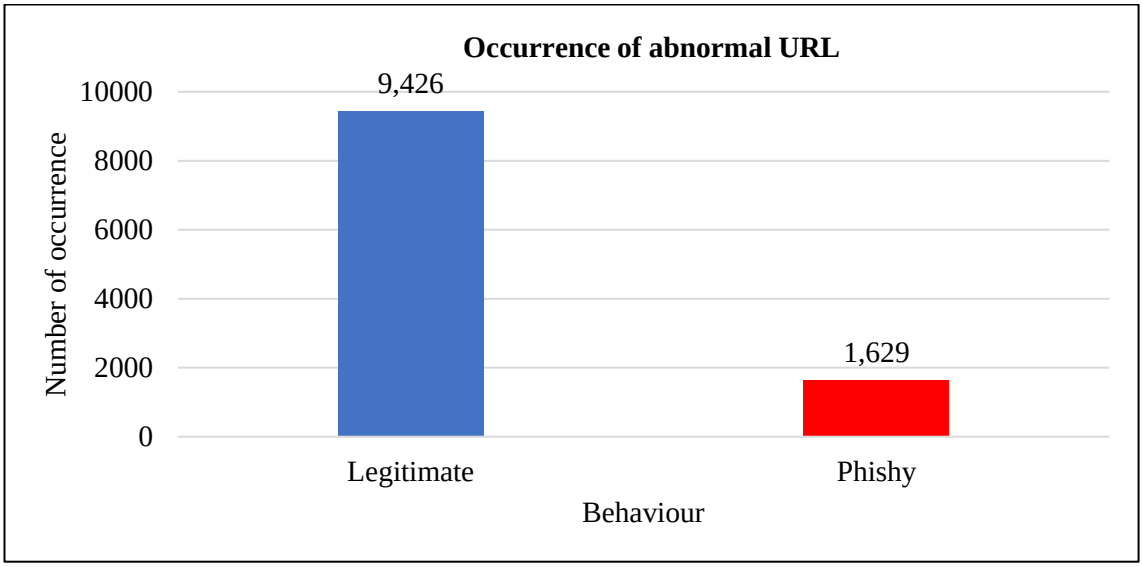


Figure 4.9: Number of occurrences of different behaviours for abnormal URL feature in phishing dataset.

Pop up window

The function of pop-up windows in webpages is to ask users for some credentials. In the current data, 8,918 webpages were found that did not use pop-up windows, which classified them as legitimate; whereas 2,137 were phishy (see Figure 4.10). Some pop-ups are based on adware, which is a next-generation malware, it means that information regarding the suspiciousness of this feature was not present in this dataset.

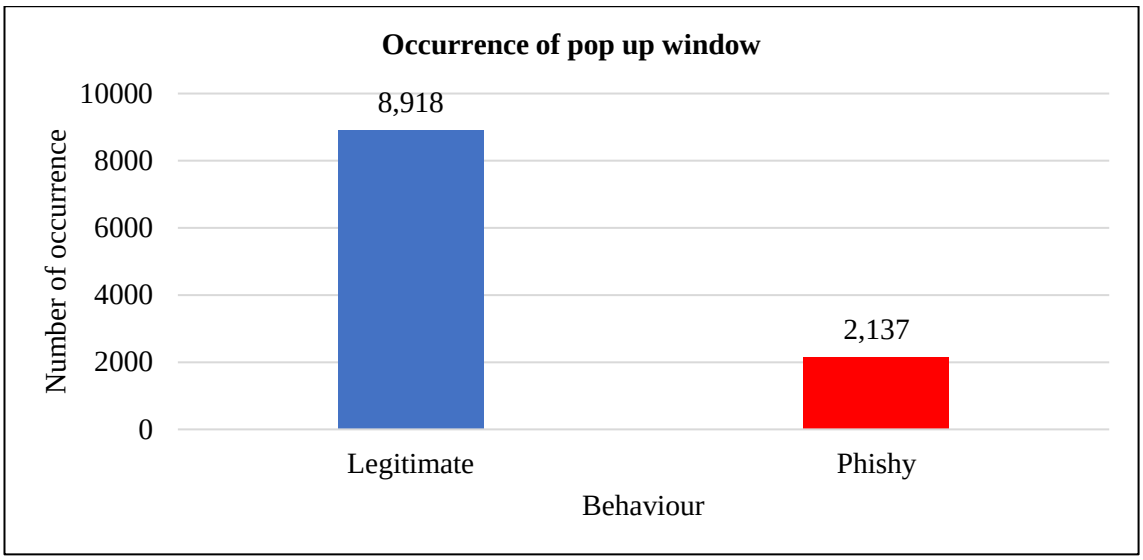


Figure 4.10: Number of occurrences of different behaviours for pop up window feature in phishing dataset.

Links pointing to page

This feature refers to links pointing to a specific URL (i.e., page or subpage). Figure 4.11 shows that 4,351 webpages were classified as legitimate and only 548 as phishy, which was the lowest rate among all features. However, suspicious webpages recorded the highest rate for this feature (6,156) compared to the web traffic feature, which had only 2,569.

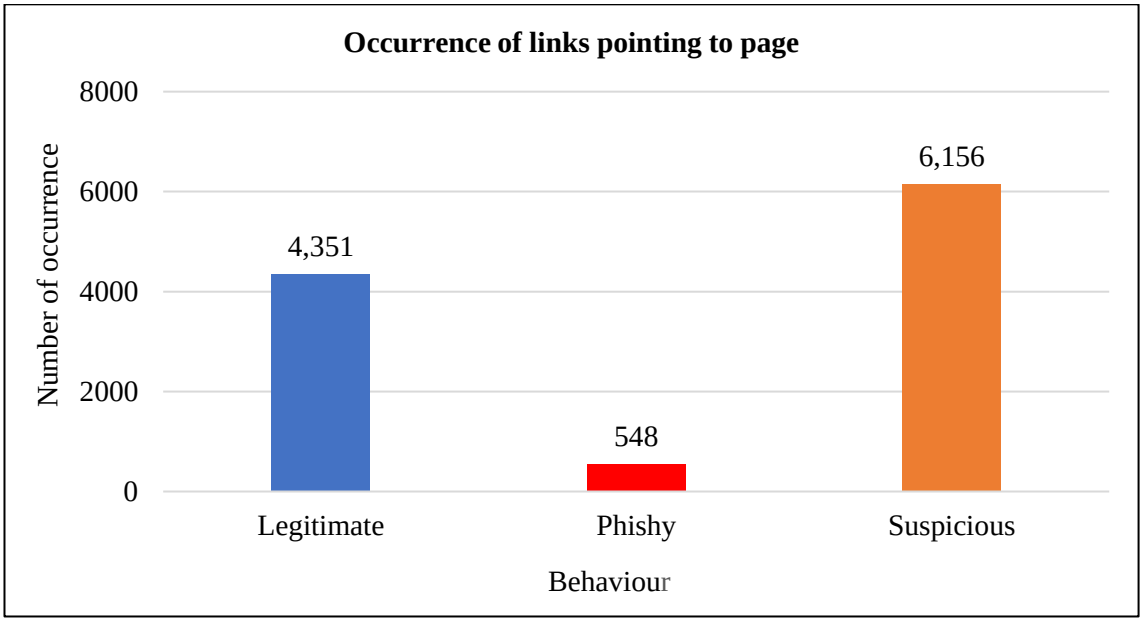


Figure 4.11: Number of occurrences of different behaviours for links pointing to page feature in phishing dataset.

4.3.2 Malware Behaviour in the Botnet Dataset

The second experiment involved findings for malware behaviour in regard to botnet attacks (Gafgyt attacks) from datasets from Danmini and Ecobee (see Section 3.2.3 for more information about Danmini, Ecobee and Gafgyt). Table 4.4 lists six features that were examined with regard to malware behaviour in botnets, based on Meidan et al. [22]. The total number of attributes in each feature was 115. The Weka tool was utilised to determine the accuracy of each feature using bagging with random tree, where random tree worked as a base classifier for bagging.

Table 4.4: Features examined in the botnet dataset.

Description	Feature
Benign traffic	1
Combo	2
Junk	3
Scan	4
TCP	5
UDP	6

Gafgyt attacks in Danmini (10-fold cross-validation analysis)

Figure 4.12 shows that the scan feature had the highest number of occurrences (99.23%) compared with other features, it is slightly more than combo and junk, which had 98.75% and 98.14%, respectively. Malware occurrence through benign was less than scan feature with a difference of around 2%. With regard to the TCP feature, it was less than benign with almost 77%. UDP was much lower than all the other features, with only around 45%.

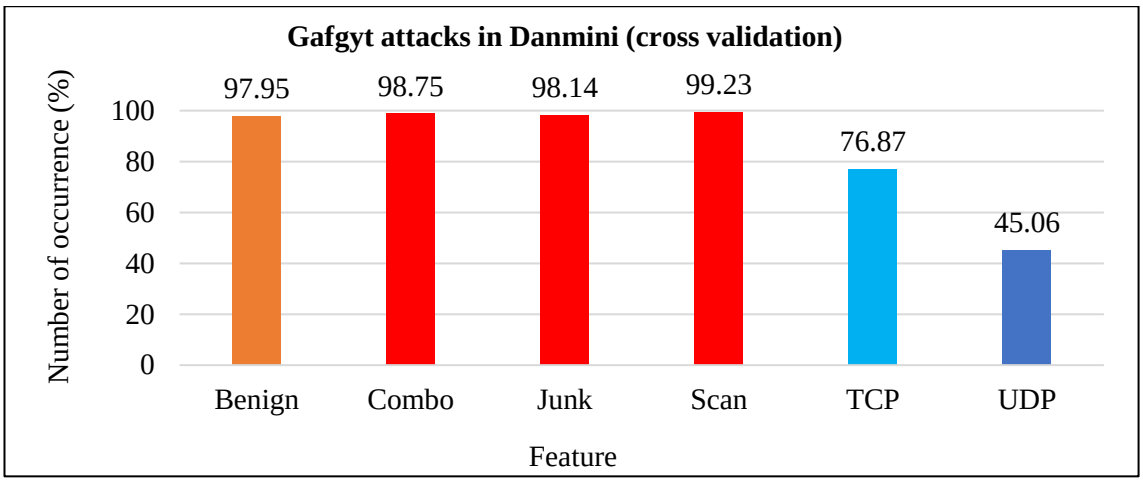


Figure 4.12: Cross validation analysis of gafgyt attacks in Danmini.

Gafgyt attacks in Ecobee (10-fold cross-validation analysis)

With Ecobee, the feature results for attacks were similar to those in Danmini, with only minor differences. The highest three rates of occurrence were in scan, junk and combo, with 99.73%, 99.53% and 99.51%, respectively (see Figure 4.13). While benign in Ecobee is more frequent than in Danmini, the difference was only 1%. The TCP rate in Ecobee was higher than that in Danmini, at around 89%. UDP remained the lowest rate, as seen in Danmini.

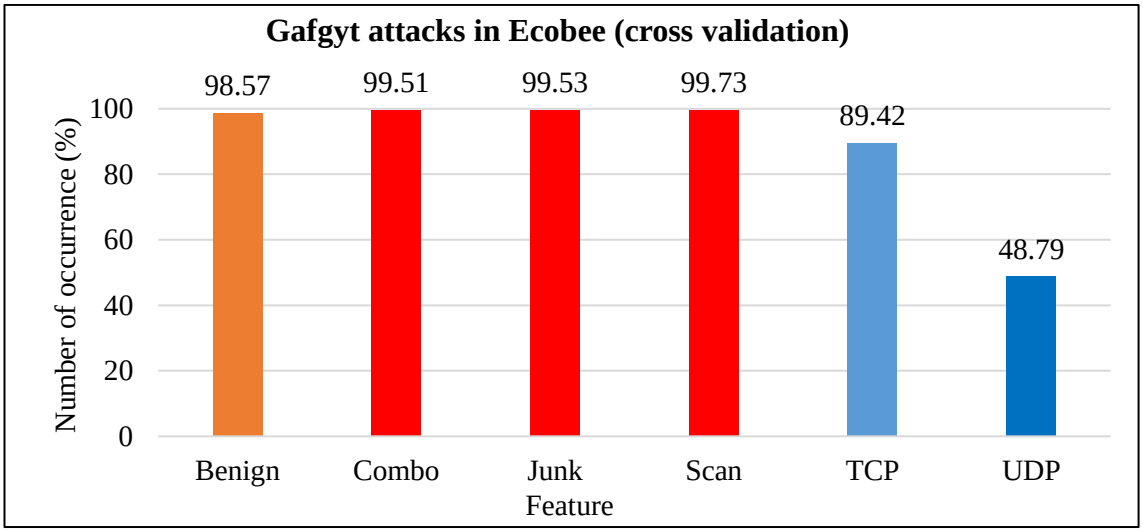


Figure 4.13: Cross validation analysis of gafgyt attacks in Ecobee.

Gafgyt attacks in Danmini (test analysis)

Figure 4.14 illustrates that the scan feature had the highest rate among all features, which was more than combo and benign, that had 99.40%, and 97.08%, respectively. It is clear that malware occurrence through combo was slightly similar to scan feature, with

a slight difference of only 0.01%. With regard to the TCP feature, it was less frequent than junk with around 80%. UDP was much less frequent than all previous features, with only around 58%.

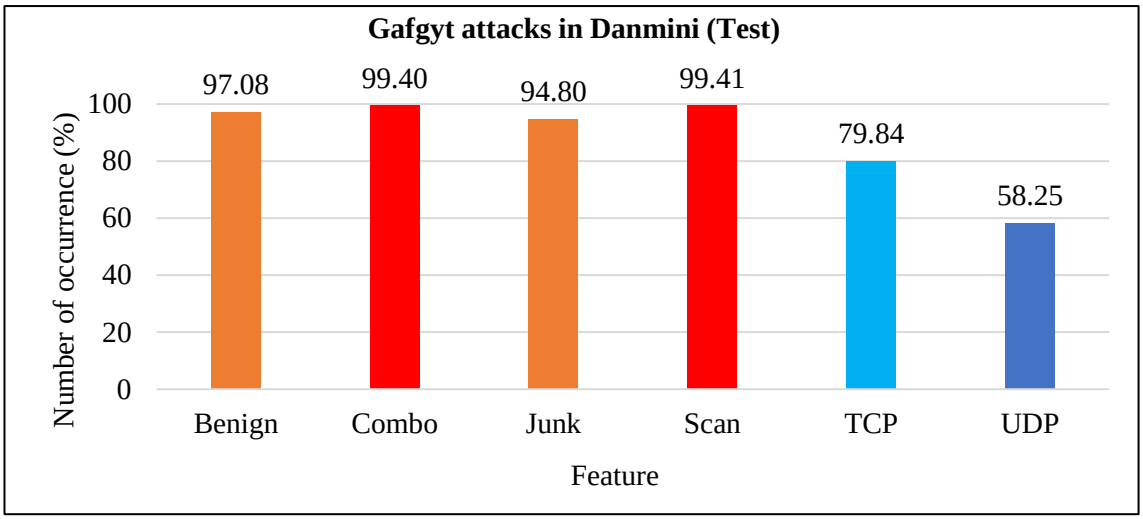


Figure 4.14: Test analysis of gafgyt attacks in Danmini.

Gafgyt attacks in Ecobee (test analysis)

With Ecobee the feature results for attacks were similar to those in Danmini, with only minor differences. The highest rates were for scan and combo: 99.79% and 99.17%, respectively (see Figure 4.15); benign in Ecobee was more frequent than that in Danmini but with a difference of only 1%. Malware occurrence through junk is slightly similar to that through the benign feature, with a slight difference of only around 0.13%. The TCP rate in Ecobee was less than that in Danmini, at around 66%. UDP was again the lowest rate, as in Danmini.

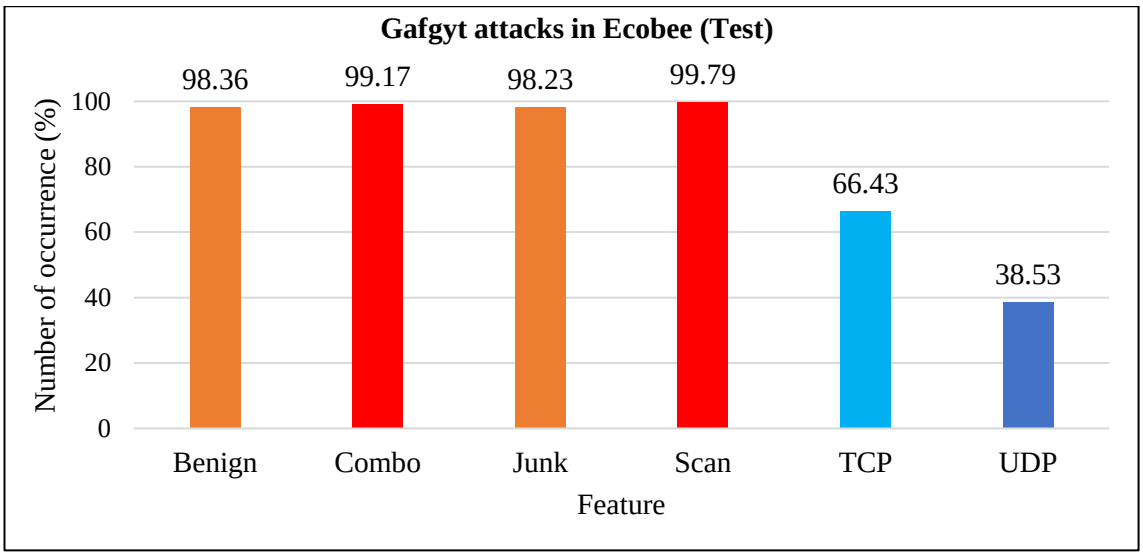


Figure 4.15: Test analysis of gafgyt attacks in Ecobee.

Based on the previous analysis, cross-validation and test provided similar results with test being lower for all features with the exception of scan, which had a slightly higher value than cross-validation (see Figure 4.16). Since the difference was less than 1% the scan feature remained an important feature to identify malware behaviour.

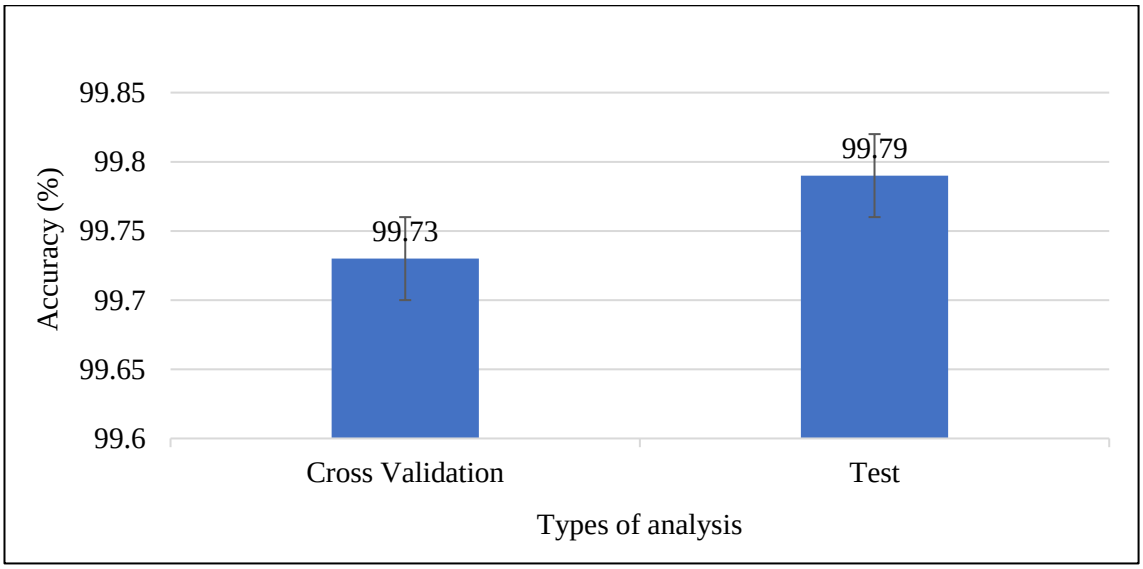


Figure 4.16: The accuracy of scan feature in two types of analysis 'cross-validation and test'.

Comparison of features between Danmini and Ecobee with test analysis

1. Benign

Figure 4.17 depicts the number of benign occurrences in Danmini and Ecobee; 97.08% were registered from Danmini but Ecobee experienced more, with 98.36%.

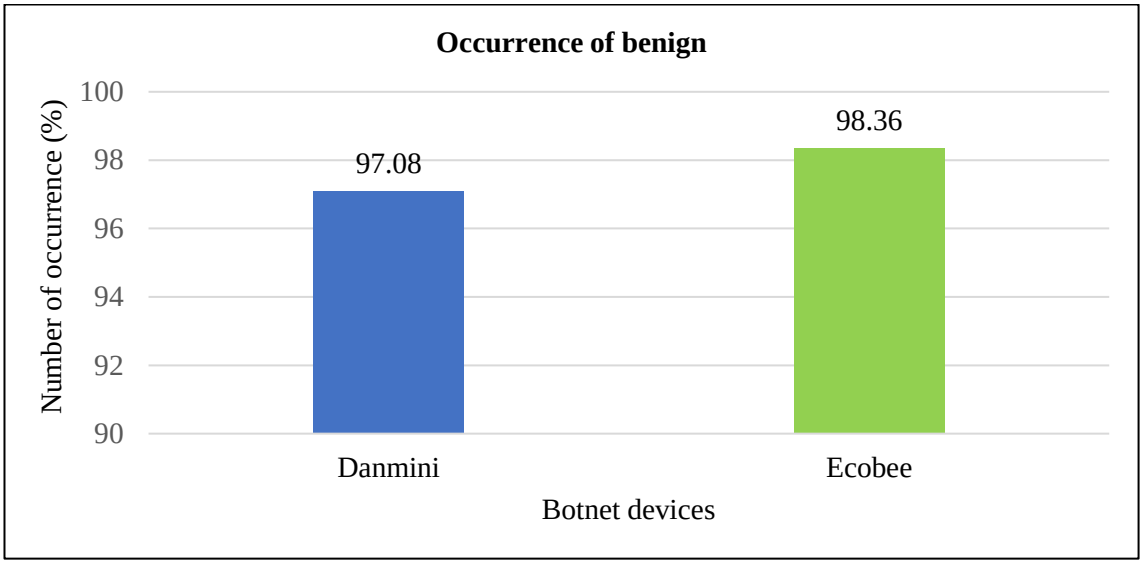


Figure 4.17: Number of occurrences of benign feature in Danmini and Ecobee.

2. Combo

Figure 4.18 shows that combo had a higher number of occurrences than the benign feature. The rate of attacks in Ecobee was slightly (by 0.23%) lower than that in Danmini.

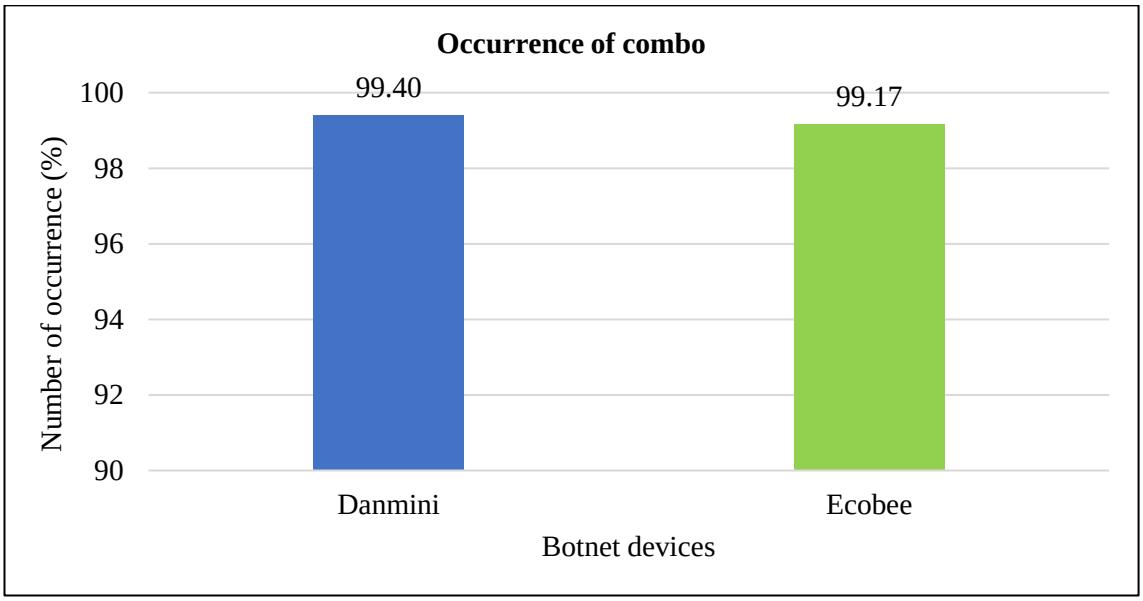


Figure 4.18: Number of occurrences of combo feature in Danmini and Ecobee.

3. Junk

Similar to the previous analysis of malicious attack in benign, the number of occurrences in Danmini was less than that Ecobee, with a difference of around 4% (Figure 4.19). In short, malicious attacks on Danmini (junk) were less common than on Danmini (benign) as shown in Figure 4.17.

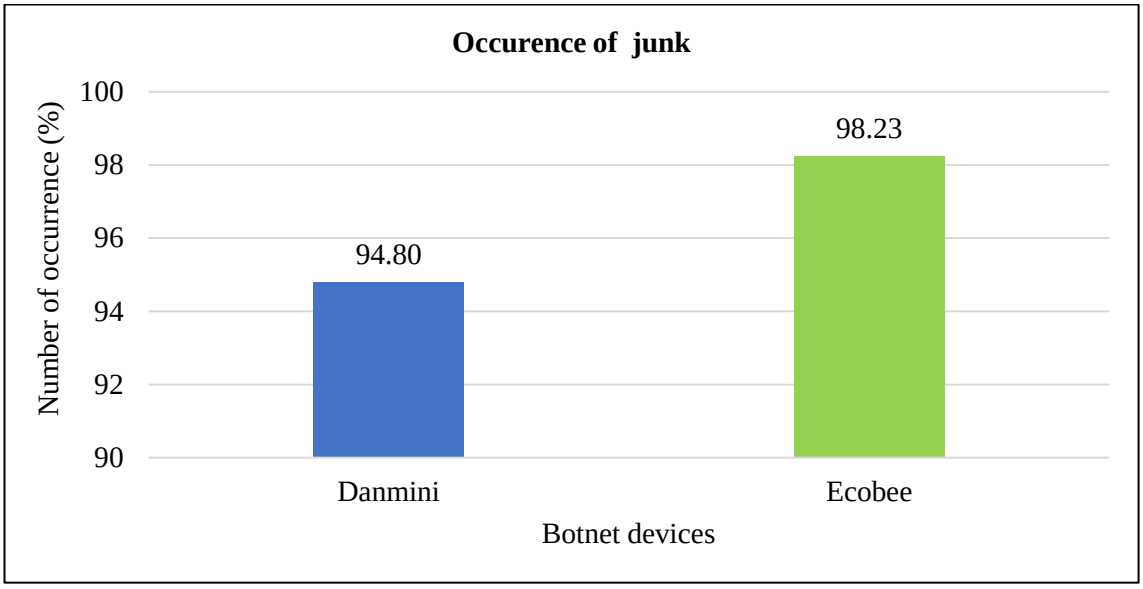


Figure 4.19: Number of occurrences of junk feature in Danmini and Ecobee.

4. Scan

Based on the previous analyses for benign and junk, the malware occurrence through scan showed a parallel pattern, with only a slight difference in percentage. In the experimental results shown in Figure 4.20, Danmini and Ecobee showed similar percentages around 99%, which was very similar to the combo occurrence (Figure 4.18).

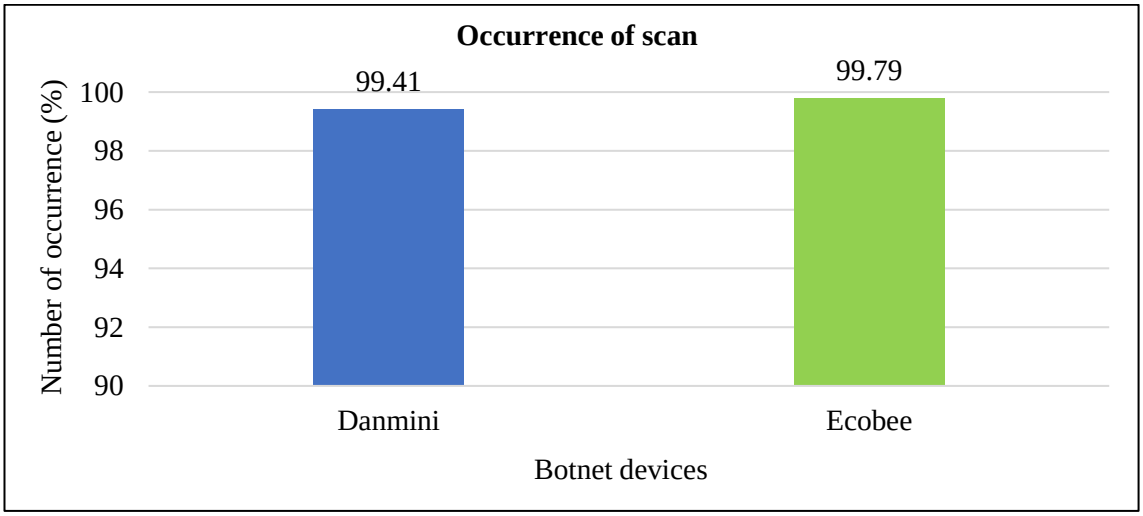


Figure 4.20: Number of occurrences of scan feature in Danmini and Ecobee.

5. TCP

Figure 4.21 shows that the TCP rate in Ecobee was less than that in Danmini with a difference of around 13%. Compared with the previous fives features, TCP had few occurrences.

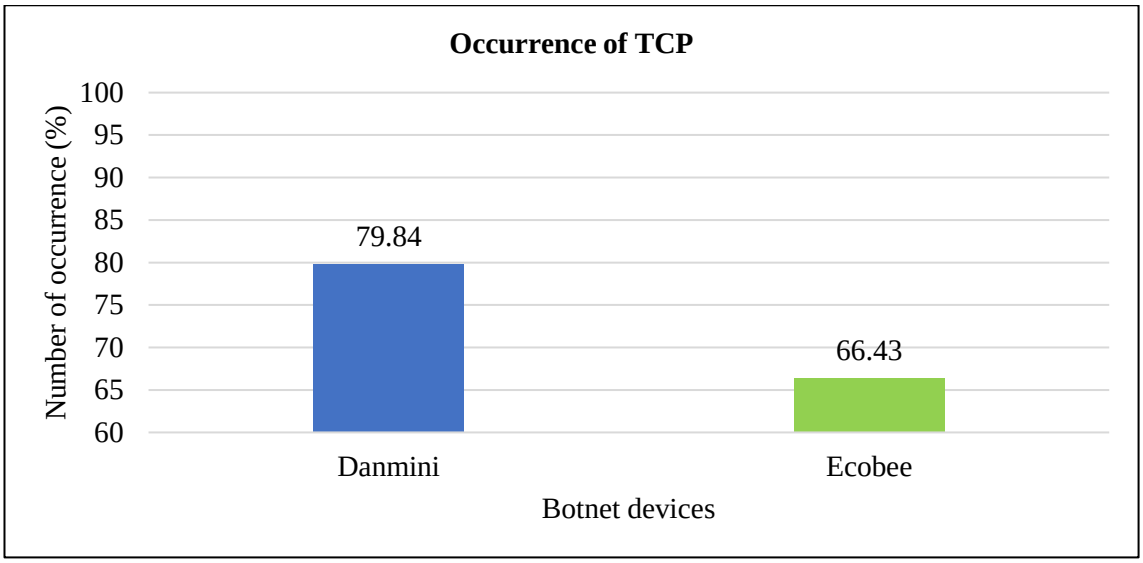


Figure 4.21: Number of occurrences of TCP feature in Danmini and Ecobee.

6. UDP

UDP had the lowest rate of occurrence among features in the botnet dataset, at 58.25% and 38.53% of hits in Danmini and Ecobee, respectively (

Figure 4.22).

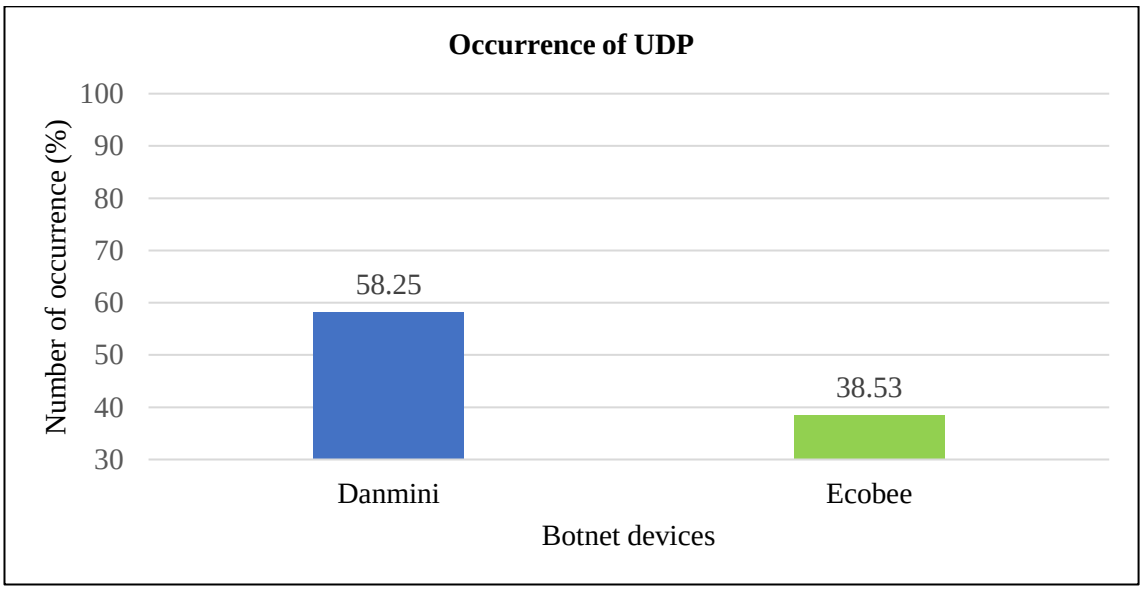


Figure 4.22: Number of occurrences of UDP feature in Danmini and Ecobee.

4.3.3 Malware Behaviour in the Honeypot Dataset

The third experiment examined malware behaviour in the honeypot dataset. This section presents findings from three honeypot sensors: Snort, Kippo and Glastopf. The section is divided into three primary sections: Snort.alerts, Kippo and Glastopf.events. The web server geolocation was in Japan (Tokyo), and operated for a period of several months from 17 October 2017 to 29 February 2018 from New Zealand (the study location). Throughout this period the server derived around 80,462 hits of malware activities, the features as shown in Table 4.5. The features IP, port, and URL shared with those in phishing dataset (Table 4.2).

Table 4.5: Features detected in honeypot data.

Description	Feature
IP address	1
Port	2
ICMP	3
TCP	4
UDP	5
Password	6
Username	7
Request URLs	8

The following three sections present results from the analysis of data received from each honeypot sensor used in this study. The datasets are identified in detail, either in graphs or tables, to illustrate malicious activities and assist in addressing the study questions.

a) Snort.Alert

This section examines the most attacked ports and protocols. The samples from Snort were taken from 17 October 2017 to 25 February 2018 (Table 4.6) as McAfee identified that attacks were more frequent during October– February.

Table 4.6: The days on which samples were taken during the main period.

Days	Months	Years
17, 18, 19	October	2017
1, 2, 3, 4, 5	November	2017
11	December	2017
15	January	2018
23, 24	February	2018

Port

The pie charts in Figure 4.23 shows the numbers of attacks received by ports between October 2017 to February 2018. At the beginning of the period, port 5060 which is for SIP traffic was the most significant port receiving attacks; it was the dominant port targeted by attackers in each of October 2017 and February 2018. It is obvious that some attackers aimed to build a communication session between the attacker device and the vulnerable server using the session initiation protocol (SIP) protocol to exchange data. In contrast, from November 2017 to January 2018, port 1433 was the largest segment and the lowest segment was made by other ports and services.

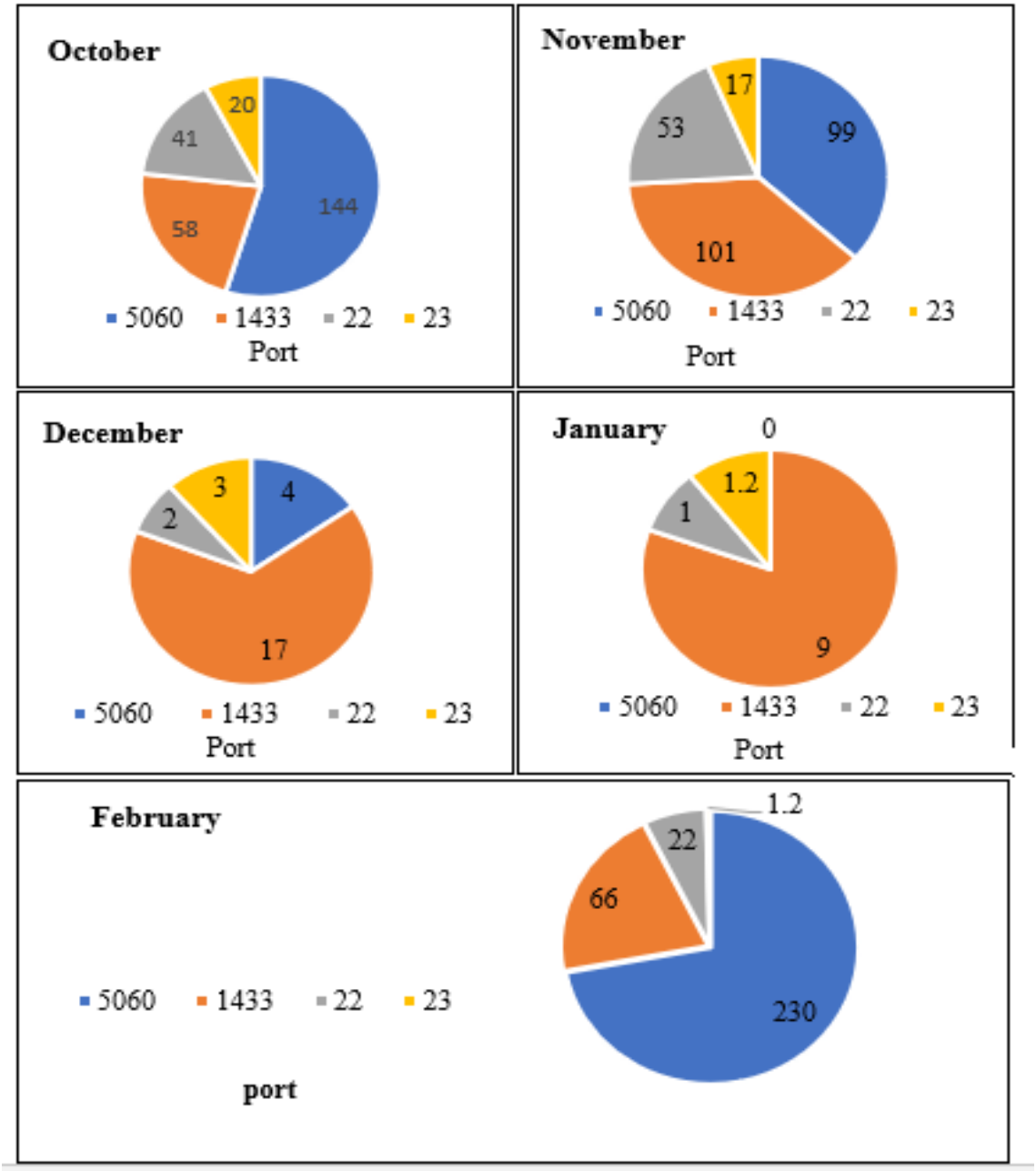


Figure 4.23: Number of attacks of the most attacked ports.

Port 5060 accounted for 144 attacks in October 2017, but the number was lower (99 hits) in November 2017. The number of attacks decreased further to 94 hits in December 2017, but in February 2018, it increased significantly to (230 attacks). For port 1433, attacks began with 58 hits in October 2017, which was followed by a clear increasing trend, to 101 hits in November 2017. The number of attacks on this port then decreased to 17 and 9 attacks in December 2017 and January 2018, respectively. At the end of the sampling period, February 2018, the number of attacks had again increased, to 66 hits for the month. When attackers used port 1433, it meant that they wanted to obtain privileges to access the SQL server by sending a request to TCP port 1433.

Ports 23 and 22 should be highlighted as two of the ports exploited by hackers. Port 23 was used by attackers for remote access for the purposes of secret espionage or to damage the system. It was targeted 20 times in October 2017, but target rates declined over the subsequent three months (2 hits in total) and then increased again to 2 hits in February 2018.

Another result of note was in relation to port 22, which is used for remote login; also, some Trojans use this port if there is any vulnerability. This means that hackers might exploit this vulnerability by using a special command through the SSH command line interface to gain unauthorised login to the system. In October 2017, port 22 received 41 attacks; this figure rose to 53 attacks in November 2017. It is apparent from the pie charts that attacks reduced significantly to 2 hits and 1 hit, in December 2017 and January 2018, respectively. At the end of the period, in February 2018, the number of attacks rose to 22 hits for the month.

In summary, the results from Figure 4.23 show that all the ports discussed in this section experienced a decrease in number of attacks in December 2017 and January 2018 by 50, 84, 18, 51 hits, in ports 5060, 1433, 23, 22, respectively. The number of attacks then rose again in February 2018.

Protocol

The bar graph inFigure 4.24 illustrates the attack rates for three protocols (TCP, UDP and ICMP) between October 2017 and February 2018. The attack rates on the TCP showed a steady but significant rise over the period from October 2017 to November 2017, while the number of attacks on UDP experienced a downward trend from October 2017 to December 2017. There was no evidence for ICMP attack throughout the period, except in November 2017. The TCP experienced a reducing in the number of attacks by 256 and 270 hits in December 2017 and January 2018, respectively. The UDP also experienced a reduction in the number of attacks by 110 and 109 hits in December 2017 and January 2018, respectively; it then experienced an increased trend in February 2018.

In October 2017, the number of attacks on the TCP and UDP were 257 and 172, respectively. The TCP attack rate increased to 303 hits during November 2017, but the UDP rate decreased to 114 hits in that month. Both December 2017 and January 2018 experienced a sharp decrease down to 33 hits for the TCP and 5 hits for the UDP. At the end of the period, the TCP and UDP rates showed a gradual increase and reached 185 hits and 234 hits, respectively. Figure 4.24 provides limited information about ICMP, this ICMP protocol registered only 12 hits, and that was in November 2017.

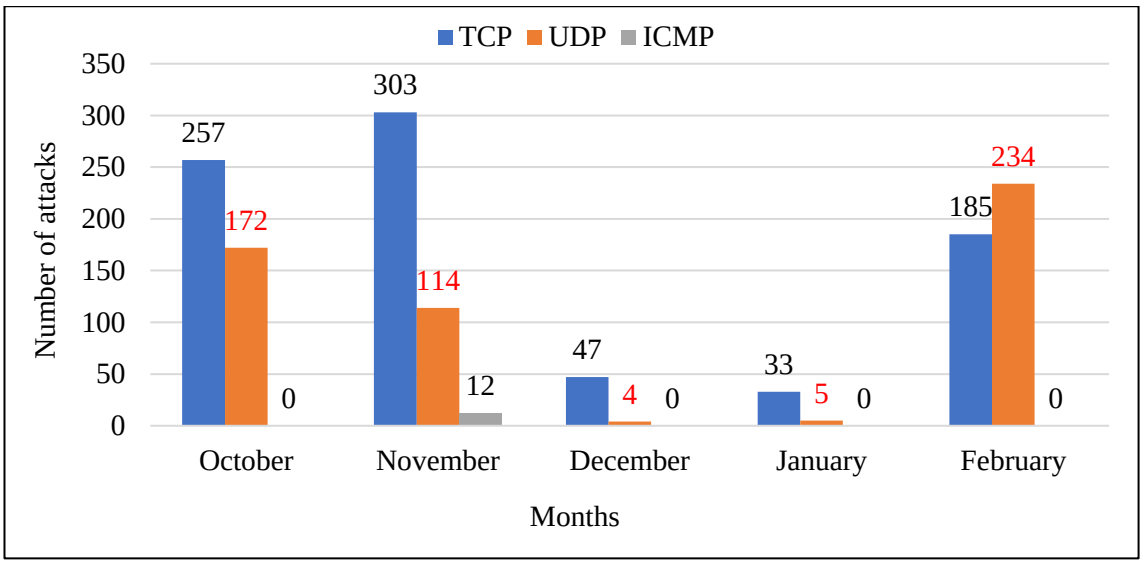


Figure 4.24: Number of attacks by month in three types of protocols.

b) Kippo

This section examines the top passwords, usernames, and it was used to study the behaviour of the top attackers.

Top passwords

Figure 4.25 shows the rate of use of the most common passwords employed by attackers in unauthorised access attempts. Overall, hackers aimed to obtain privileges to login to a victim's machine by using the brute force method. This technique works by using a random group of passwords. Usually, this approach can achieve access if system administrators use default or weak passwords. The most used (660 attempts) password attempt was '123456', while the least used was 'qwerty', with 164 attempts.

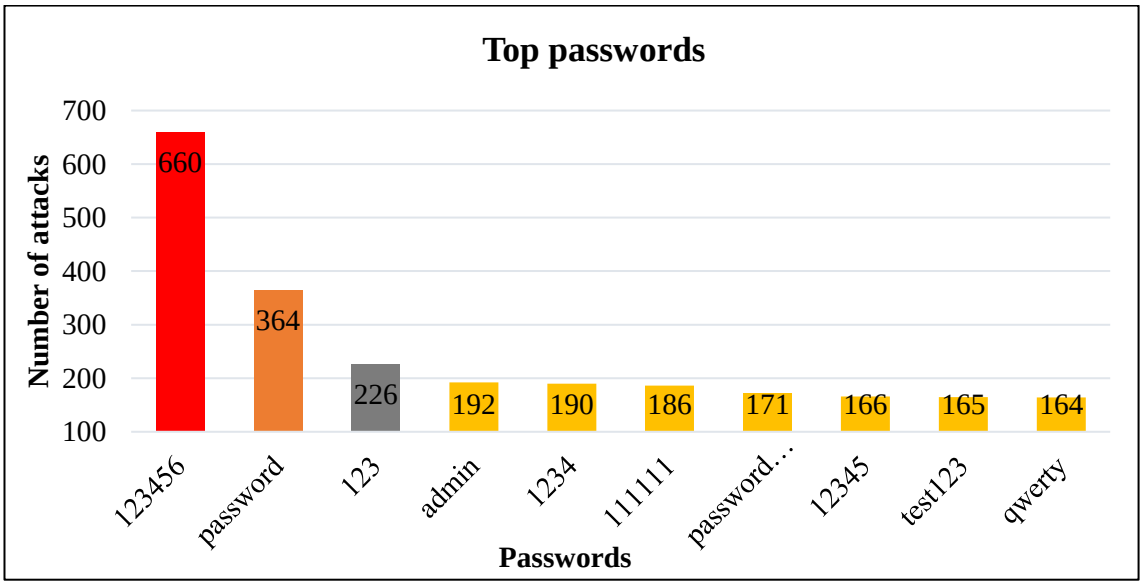


Figure 4.25: Number of attacks in top 10 passwords used by attackers**.**

Top usernames

Figure 4.26 provides a summary of the top 10 usernames employed by adversaries attempting to gain access to a vulnerable server. The most substantial rate (3000 times) of username attempts was for ‘root’, while ‘test’ had the lowest rate (102 attempts).

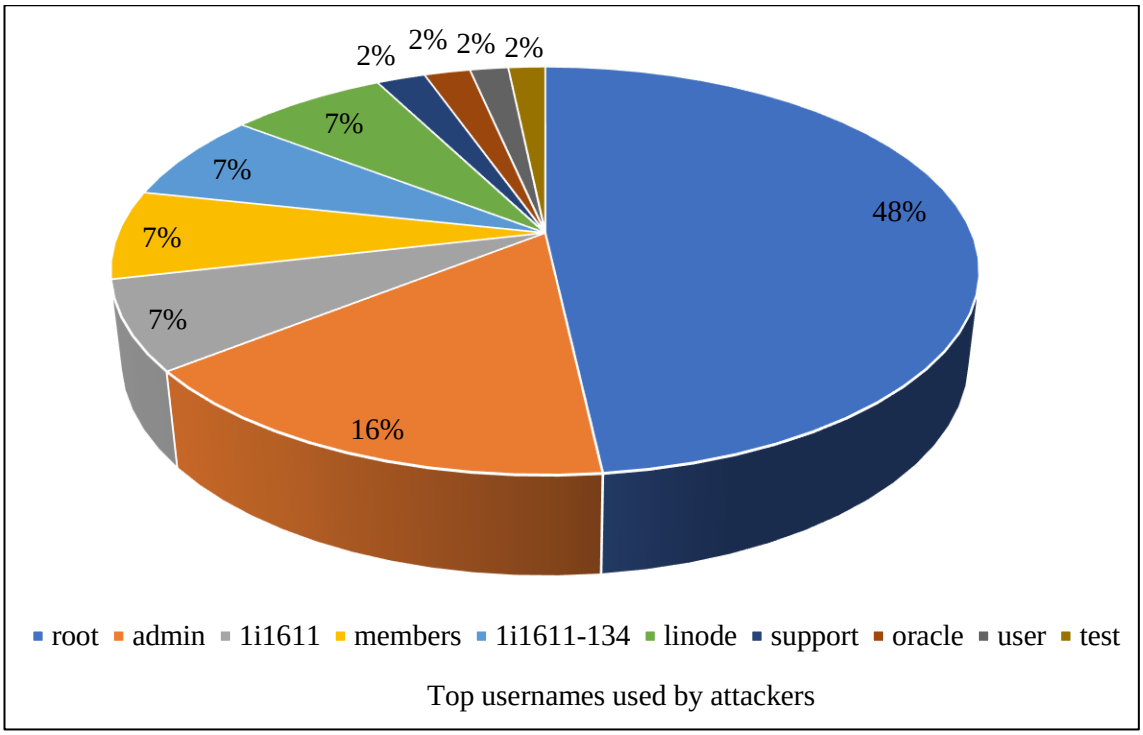


Figure 4.26: Number of attacks in top 10 usernames used by attackers.

Top usernames/passwords

This section shows that Kippo did a good job of revealing brute force attacks by attackers and reporting hacking attempts. Figure 4.27 shows that the most common combination of usernames/passwords used by attackers was ‘admin: admin’, which was employed 99 times. The combination ‘admin:1111’, ‘root:1234’ and ‘admin:1234’, were used in only 63 attempts by attackers.

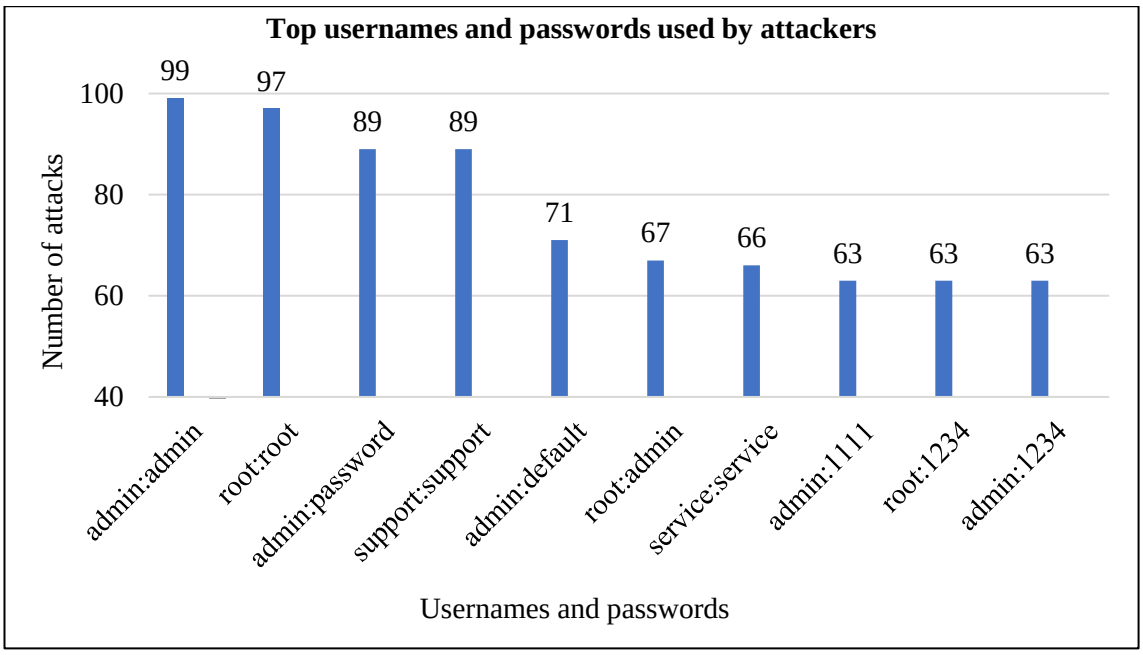


Figure 4.27: Number of attacks in top 10 usernames/passwords.

Top attackers

Table 4.7 shows the top 10 attacker IP addresses detected by the Kippo honeypot, and the frequencies of those attacks. The geographical location of the attackers is out of the scope of this study. The data/information presented in Figure 4.7 was collected and compared with the other datasets included in this study.

Table 4.7: The IPs of attackers with the number of attacks.

IP address	Number of attacks
177.39.121.252	8,512
184.106.219.63	7,532
186.251.208.49	3,423
112.78.4.85	3,311
185.25.122.3	1,580
193.70.40.191	1,408
51.254.123.147	1,047
176.53.0.87	931
185.165.29.198	873
183.192.189.133	774

c) Glastopf.events

The data classification presented in this section is based on IP address including ports and incoming URLs. The attacks were registered over a period of 10 days from 7 to 16

November 2017. This section covers only the top attackers, who targeted Port 80: protocol http with their activities.

- **IP: 45.77.149.77**

This attacker used 181 dynamic/private ports chosen from the port number of 32,918 to 60,968. The attacker used different request URLs and the attacks occurred on Sunday 12 November 2017 at 22:06. Examples of URLs that were requested and used by the hacker are:

/server-status

/favicon.ico

/path.php~

/index2.php.~1~

/Nmap/folder/check1510477545

/HNAP1

/robots.txt

- **IP: 94.177.237.15**

This attacker used 13 dynamic/private ports selected from 52,868 to 54,159. The attacker utilised a variety of request URLs. The attacks were carried out on Wednesday 15 November 2017 at 14:46. A few of the URLs that were requested and used by the hacker are:

/SQLite/main.php

/SQLiteManager/main.php

/agSearch/SQlite/main.php

/SQLiteManager-1.2.4/main.php

- **IP: 121.130.202.67**

This assaulter used eight dynamic/private ports selected from 47,788 to 48,427. The assaulter utilised difference of request URLs. The attacks occurred on Tuesday 14 November 2017 at 23:38. Several URLs were requested and utilised by the hacker, including:

/shell?echo+jaws+123456;cat+/proc/cpuinfo

/cgi-bin/user/Config.cgi?.cab&action=get&category=Account.*

/apply.cgi

/board.cgi?cmd=cat%20/etc/passwd

/upgrade_handle.php?cmd=writeuploaddir&uploaddir=%27;echo+nuuo+123456;%2

/system.ini?loginuse&loginpas

- **IP: 211.110.139.158**

This attacker used eight dynamic/private ports selected from 43,928 to 45,434. The assaulter utilised eight request URLs, on Sunday 12 November 2017 at 07:57. Examples of incoming URLs utilised by the hacker are:

/test.php

/phpinfo.php

/u.php?act=phpinfo

/info.php

/dashboard/phpinfo.php

/config.php

- **IP: 77.81.229.93**

This aggressor used six dynamic/private ports chosen from 37,964 to 39,408. The attacker utilised different request URLs, on Monday 13 November 2017 at 12:27, including:

/MyAdmin/scripts/setup.php

/muieblackcat

/myadmin/scripts/setup.php

/pma/scripts/setup.php

4.4 Summary

This chapter presented information on the findings from research focused on three types of datasets namely phishing, botnet and honeypot. The findings were obtained by utilising classification methods presented in this chapter. Based on the empirical study, bagging with random tree is a better classifier as it produces higher accuracy with less error. This study identified different features related to malware attack, including IP, port, protocol, URLs and web traffic, which were analysed based on their occurrence. Phishing data malware accuracy was 96% (11,055 occurrences); for botnet, accuracy was 85% (680,786 occurrences); while for honeypot it was only 35%. Although malware attack in the honeypot data appears less common, in fact it was higher based on the number of occurrences (80,462). To acquire honeypot data, MHN architecture and three sensors were applied. Phishing identified malware behaviour as legitimate, suspicious or phishy. Overall, the number of attacks on phishing data were higher. The URL was the most vulnerable feature during malware attack. A discussion of the findings and comparisons drawing on the analysis and research findings are presented in Chapter 5.

Chapter 5
Discussion and Comparative Study

5.1 Introduction

In Chapter 4, analysis was conducted on three datasets: phishing, botnet and honeypot. A variety of features were analysed, including IP, port, TCP, UDP, junk, combo, benign, passwords, usernames, Google index, email, web traffic, pop up window, page rank, HTTPS token, abnormal URL and request URL. This chapter discusses the analysis and research findings presented in Chapter 4 and is divided into four main sections: The malware behaviour in the phishing dataset is discussed in 5.1.1; the malware behaviour in the botnet dataset and honeypot data are presented in Sections 5.1.2 and 5.1.3, respectively. The last section (Section 5.1.4) compares features among the three datasets. The chapter ends with a summary in Section 5.2.

5.1.1 Malware Behaviour in the Phishing Dataset

This study focused on legitimate, suspicious and phishy characteristics as the behaviour of malware in website which are categorized into these three. Therefore, another interesting finding in Chapter 4 was the identification of the features in the data sets that are influential in malware behaviour. **Error! Reference source not found.** summarises important information about Google index (higher rank in google search). Having real-life experience of using McAfee web adviser [145], it is already in Google index and consequently it is legitimate and safe to browse, as Google is the overwhelming leader in the world [146].

Many website rankings or page ranks, for example, Alexa, are very high because of their content and browsing frequency [147], [148]. Every website has a ranking and it is based on the search term and keywords that is mostly used by SEO (search engine optimisation). Also, the content that attracts more users will make the site with top page rank. Websites that have prohibited videos organised or hidden very well and more visitors are given very high page ranks. In fact, such sites have more malware content than others. In the phishing dataset the number of occurrences of phishy behaviour that detected for page rank feature was 8,201, which was around 74% accuracy (see Appendix B.1 for calculation). Thus, the findings from this research include that more phishy

behaviour may be identified in websites that with high page ranks. In summary, Google indexed websites are safer to browse, as only 1,539 (13%) were associated with phishy behaviour, out of 11,055 which is the total number of phishing data.

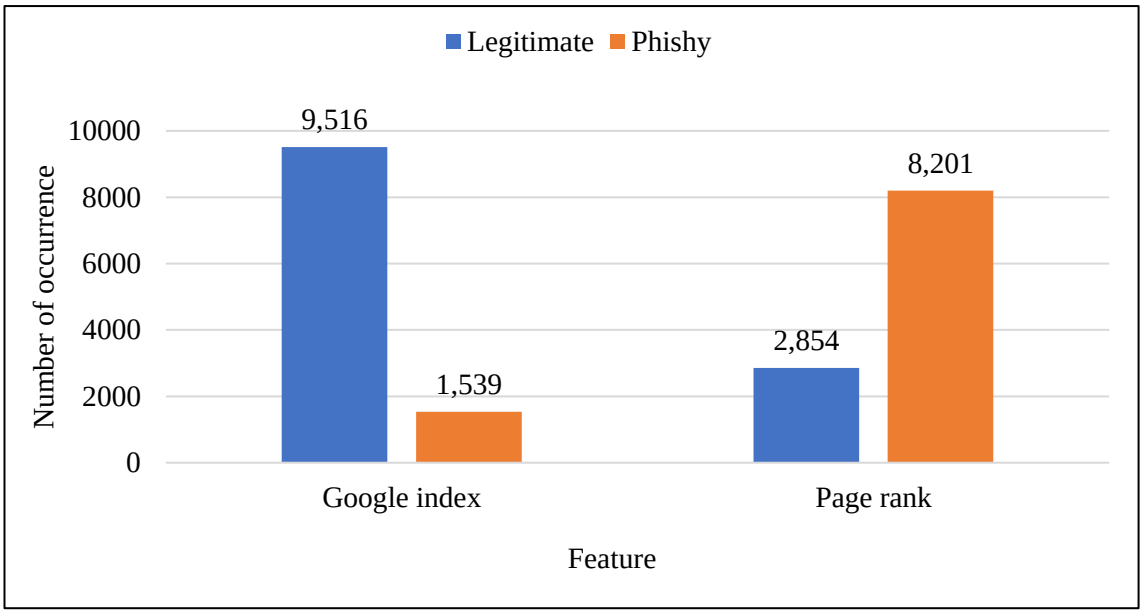


Figure 5.1: Number of occurrences in Google index and page rank features.

Figure 5.2 shows that all features recorded high rates in regard to legitimate webpages, but phishy webpages had higher page ranks than legitimate ones. Figure 5.2 shows the selected features from the phishing dataset. The results in Figure 5.2 suggest that having an IP address, an average amount of web traffic and a high page rank (randomly selected features) are not reliable key features to consider a website legitimate, which is in line with the findings in **Error! Reference source not found.**. When the Google index is high, that means sites are reliable in average as only 1,539 times phishy (13%) are detected out of 11,055 while average of phishy behaviour is nearly 43% (see Appendix B.2 for calculation) when combining the features having IP address, request URL, web traffic and page rank. Not all features in the phishing dataset provided information about suspiciousness. However, suspiciousness was noted in the following features: web traffic (23%) and links pointing to page (55%). The number of suspicious behaviours in web traffic was 2,569 and 6,156 in links pointing to page.

The findings of this study are valid as a prohibited website has a valid IP, high page ranking, many request URL, and links pointing to that site without very high Google indexed. However, Google is still on the top rank because of search engine optimisation (SEO) tools and techniques [149]. The main objective of SEO is to attract people to

specific and required site with good and attracted contents; so more visitors to the site lead to more google notices, then lead to a higher rank in google index [146]. In addition, a prohibited site mostly contains various links to point to similar type of pages that may be phishy.

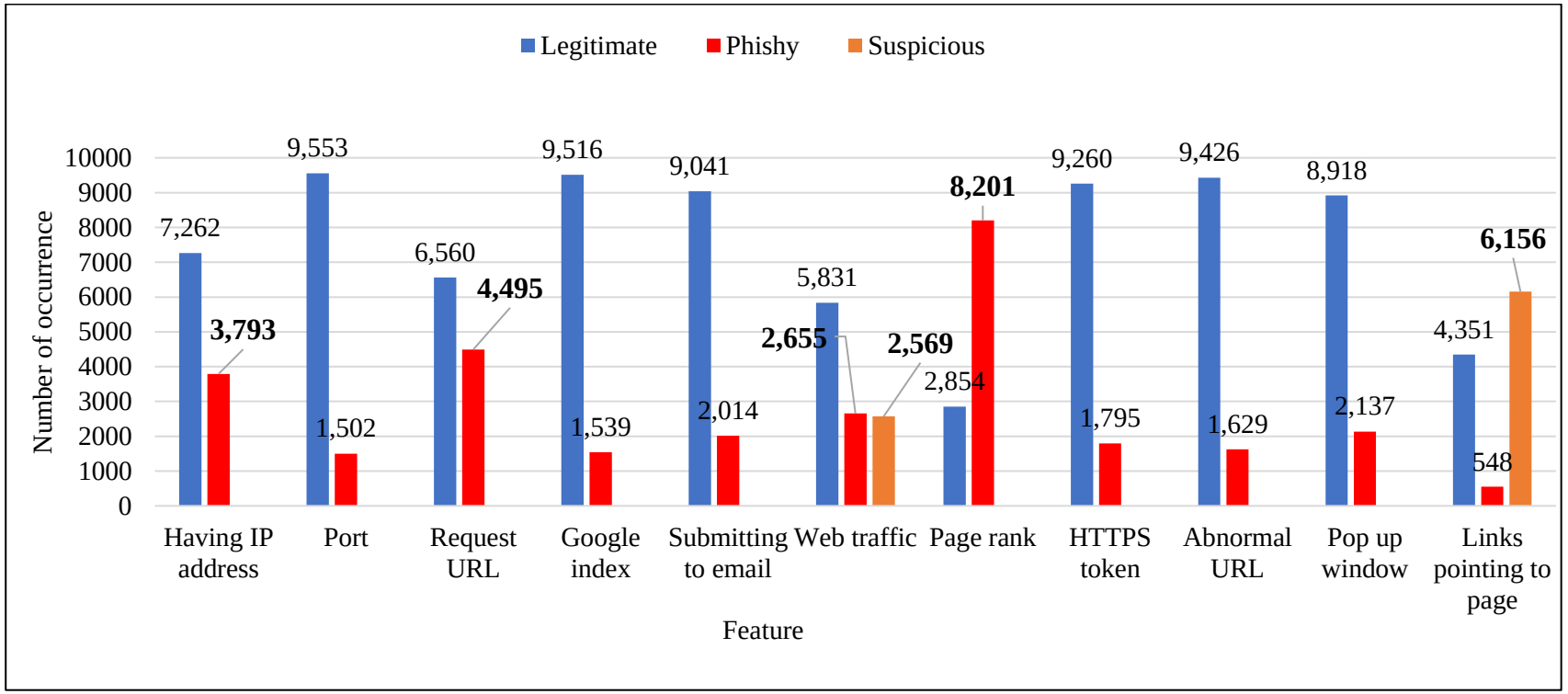


Figure 5.2: Number of occurrences of all features with comparison of legitimate, phishy and suspicious.

Figure 5.3 clearly shows that considering a website is reliable based on page ranking is not advisable because highly ranked websites had the highest occurrence of phishy behaviour in the dataset (8,201 hits). Another interesting point to note is that having an IP address does not guarantee that a website is reliable because the frequency of phishy behaviour was almost 34% accuracy (3,793 /11,055). Therefore, Links pointing to a website may not be valid, as they were phishy in around 5% accuracy (548 /11055) and were not free from being suspicious in 55% accuracy (6156 /11055), as shown in Figure 5.2.

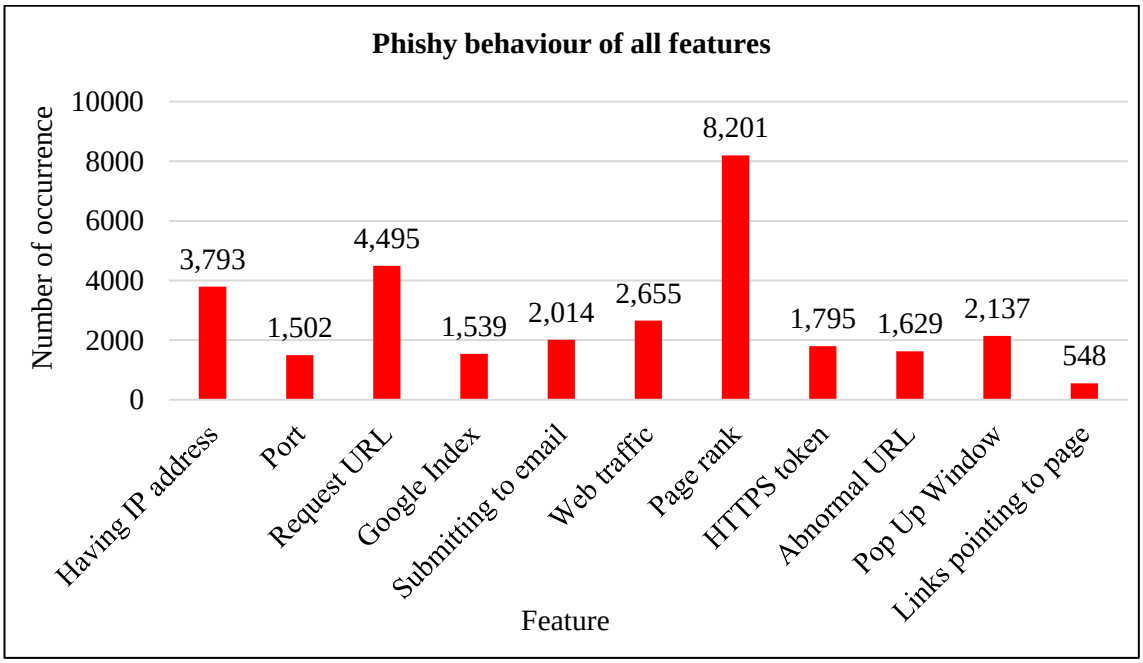


Figure 5.3: Number of occurrences for phishy behaviour of all features.

5.1.2 Malware Behaviour in the Botnet Dataset

Figure 5.4 summarises the malicious behaviour that detected by Danmini and Ecobee based on cross-validation analysis. It was evident that there was very little difference in terms of maliciousness in both devices. All features were equally important for identifying malicious behaviour as all scored more than 97%, with the exception of TCP and UDP. The TCP is used more often than the UDP. When a communication protocol is custom designed it is secure because it has no common port; rather, it has its own TCP or FTP, with or without encryption technology. A TCP is not a favourable target for hackers because it has an in-built security protocol. Several types of scans are available that claim to keep computers safe from viruses. However, many of these scanners are adware; for

example, some virus removal tools, driver finders and driver boosters are known to be adware. Thus, the scan feature had 99.23% accuracy in Danmini. The occurrence of combo was 98.75% accuracy, the second most common feature in Danmini. Similar trends were noted for Ecobee, with little difference.

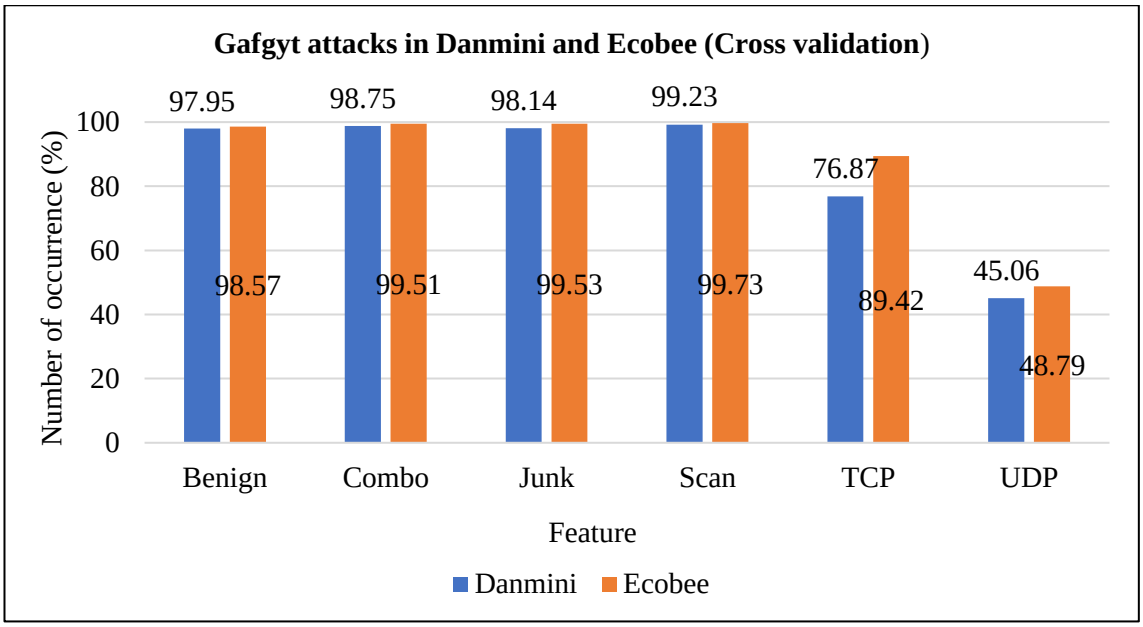


Figure 5.4: Number of occurrences in Danmini and Ecobee.

Figure 5.5 provides similar results to Figure 5.4, it summarises the malicious behaviour that detected by Danmini and Ecobee based on test analysis. An interesting point here is that Danmini occurrence was higher in some features (such as combo, TCP, and UDP) than Ecobee in Figure 5.4. Attack occurrence in Ecobee was not always lower in the test result. Combo in Ecobee scored slightly lower (99.17%) while combo in Danmini was 99.40%. A similar pattern was identified for the features TCP and UDP, with a noticeable difference of 13% and 20%, respectively.

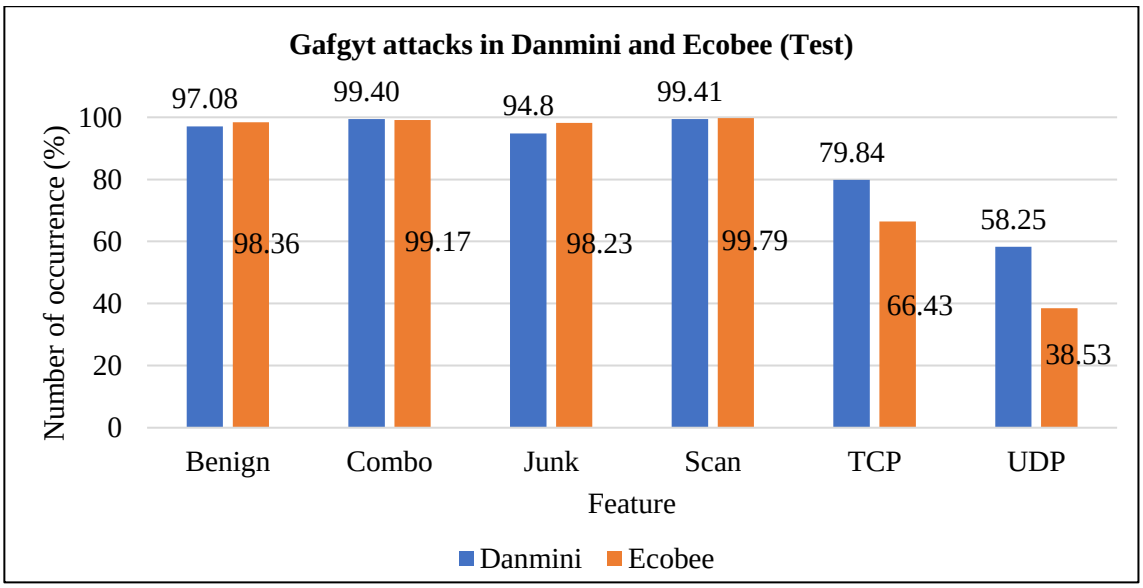


Figure 5.5: Number of occurrences in Danmini and Ecobee with test analysis.

Figure 5.6 shows that there is little difference in the test and cross-validation results between the features benign, combo and scan for Danmini and Ecobee. This clearly verifies that these features are equally important to identify malware behaviour. However, the features junk, TCP and UDP showed huge differences in test result for Danmini and Ecobee. This finding shows that TCP and UDP attack is less than all other features. Thus, future malware prevention platforms may require less focus on the features junk, TCP and UDP. Junk is no longer a great threat as most users are aware of it and are careful when opening junk email. Experimental results from Kheir et al. [150] relating to some botnet domain blacklists showed that the system called ‘Mentor’ is capable of accurately identifying legitimate domain names with low error rates.

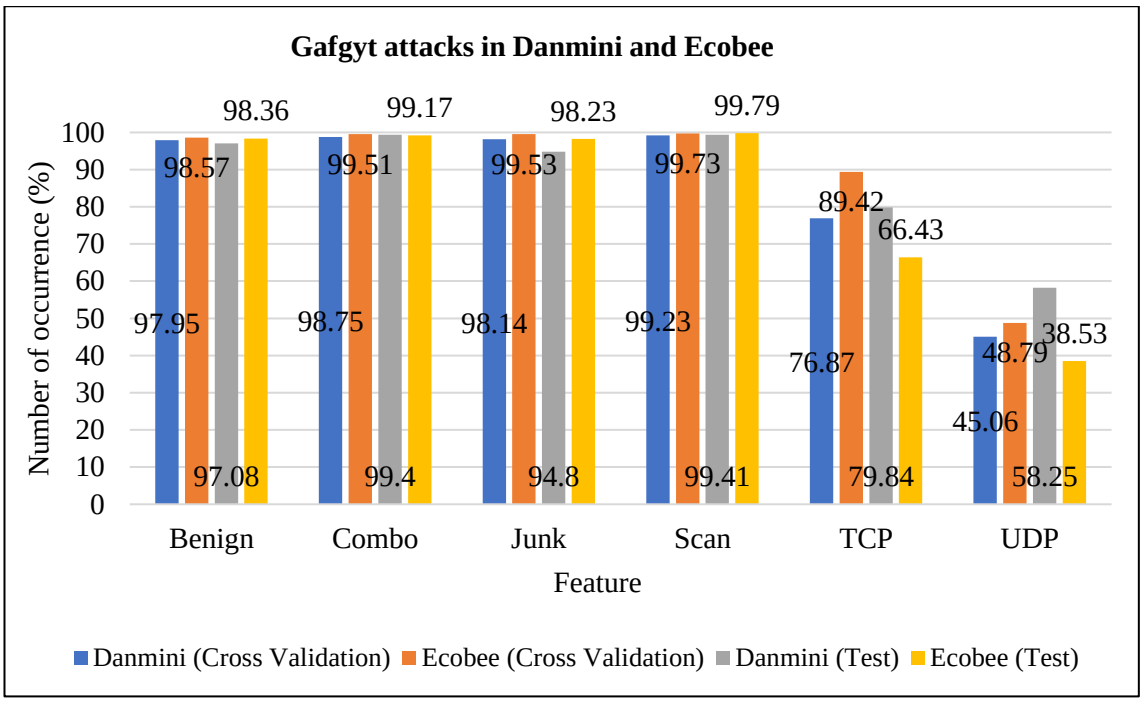


Figure 5.6: Number of occurrences in Danmini and Ecobee for cross validation and test analysis.

5.1.3 Malware Behaviour in the Honeypot Dataset

All the datasets describe that malware behaviour is closely influenced by type of IP, URL and other features. However, phishing and botnet datasets did not include the effect of username and password in a website. Hence, the honeypot infrastructure was applied to identify the effect of username and password in relation to malware attack. The honeypot dataset revealed the username (admin) and password (admin 123) that are the most prone to malware attack. The honeypot dataset also revealed malware attacks on IP port, URL and protocols, in line with the other two datasets.

In the Glastopf results presented in Section 4.3.3, the attacker's purpose may have been to execute an SQLI against the web server; the attacker used personal homepage (PHP) language to generate the script, as it was mainly focused on server-side scripting. If a web designer does not securely code the interaction between the website and the SQL database, attackers can take advantage of this mistake to sneak unexpected SQL queries onto the database server.

5.1.4 Comparison of Features among the Three Datasets

IP address feature in phishing and honeypot datasets

Figure 5.7 shows that the number of phishy occurrences in the phishing dataset was too low in comparing to the honeypot infrastructure data. The honeypot infrastructure is

a closed platform in Linode server, which has been configured with MHN software to create the honeypot platform, the IP results in honeypot data was based on Kippo analysis.

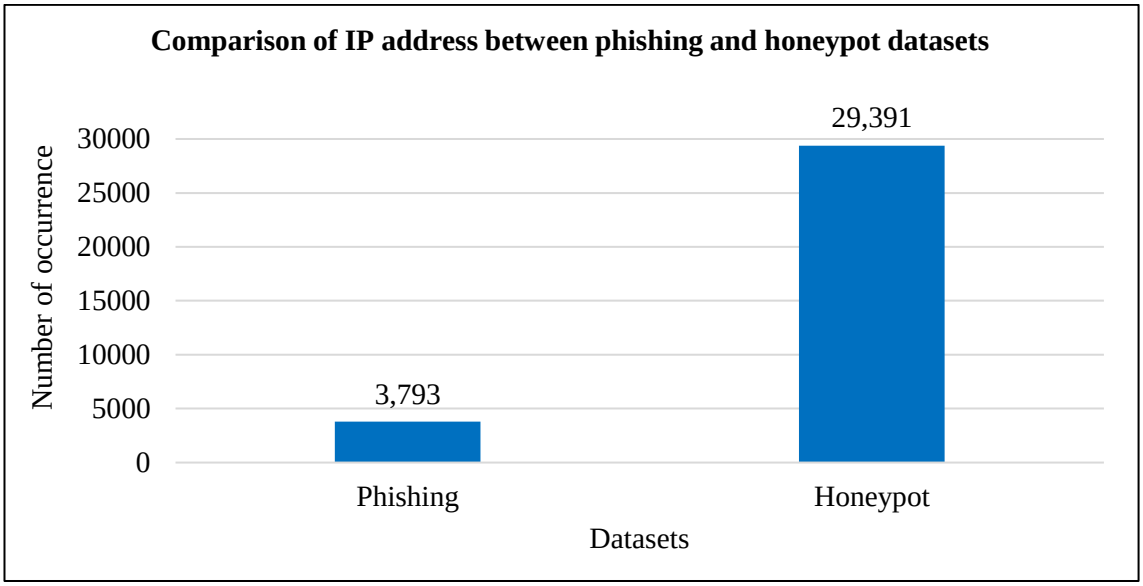


Figure 5.7: Number of occurrences of IP address feature in phishing and honeypot datasets.

Port feature in phishing and honeypot datasets

Figure 5.8 presents the data for phishy behaviour in the phishing and honeypot datasets. Occurrence of phishy behaviour in the phishing dataset was more 1.5% higher (1,502) than in the honeypot dataset (962), the port results in honeypot data was based on snort analysis

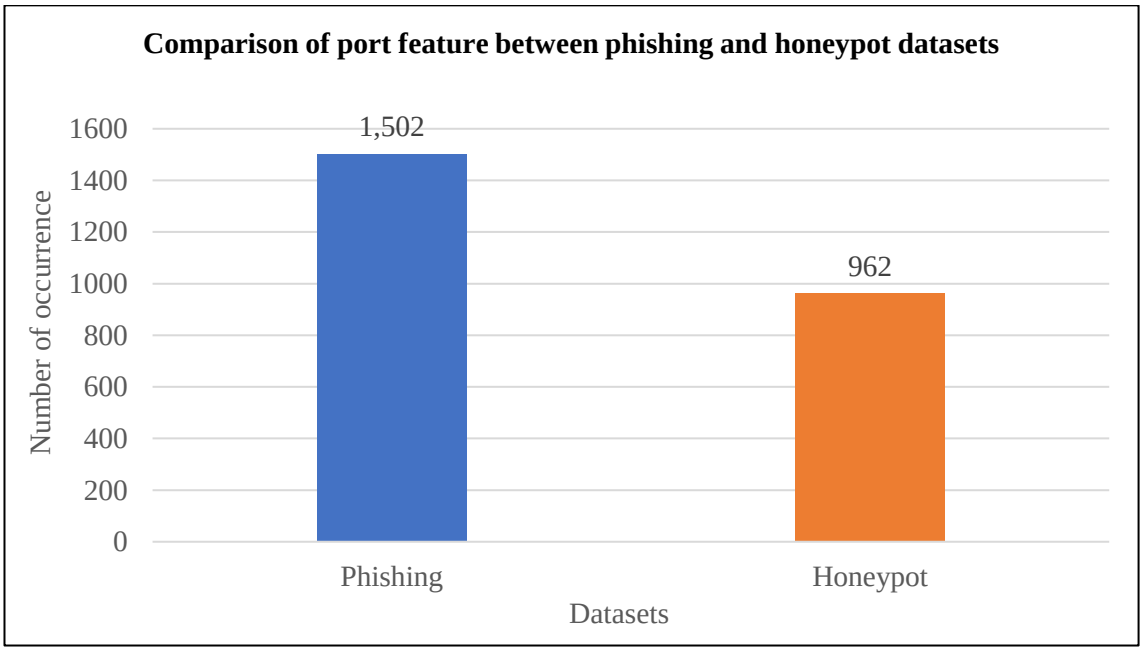


Figure 5.8: Number of occurrences of port feature in phishing and honeypot datasets.

URL feature in phishing and honeypot datasets

Figure 5.9 illustrates a similar pattern to Figure 5.7. It is noticeable that an IP address would normally be with an URL address. However, a single IP may have a different URL based on domain and subdomain characteristics. Thus, attacks on URL in the phishing dataset were more comment than in the honeypot dataset.

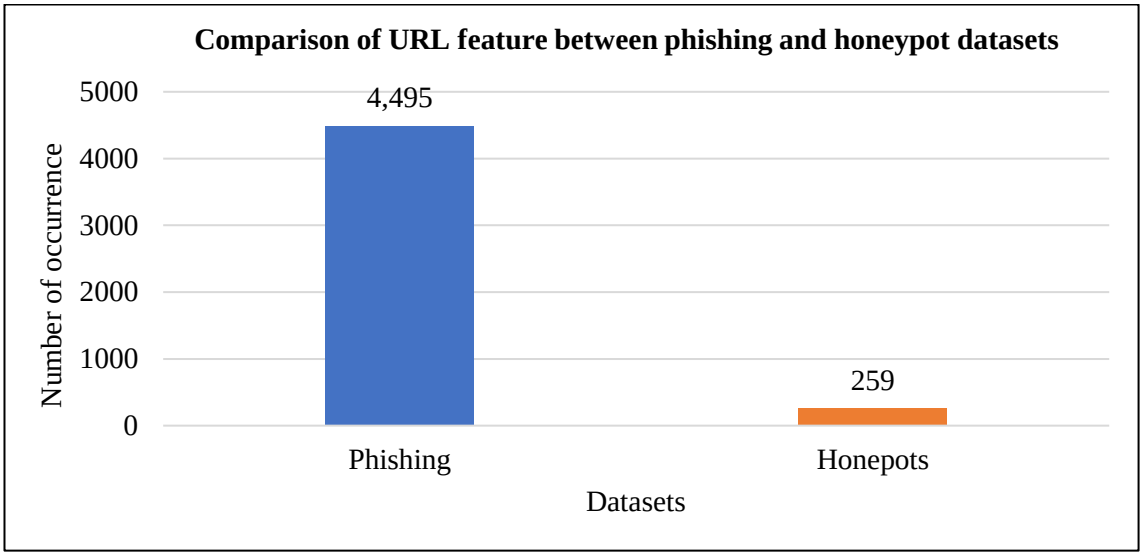


Figure 5.9: Number of occurrences of URL feature in phishing and honeypot datasets.

Email feature in phishing and botnet datasets

Figure 5.10 shows the results of email feature in phishing and botnet datasets. In botnet dataset there was a slight similarity between Danmini and Ecobee using the test analysis (94.8%, and 98.23%, respectively). When comparing the average '96.52%' (see Appendix B.3 for calculation) occurrence in the botnet dataset with the phishing dataset, it appears that the accuracy of phishing data was lower (20.14%) than the average of botnet data, which requires further future study. The total number of occurrences in the phishing dataset was 11,055 and in botnet dataset, 680,786 as each feature in Danmini and Ecobee had a different number of instances (see Appendix A.1 and A.2 for more information about the rest features. Appendix A.1 shows the botnet dataset of Danmini and Appendix A2 has the botnet dataset of Ecobee. Both appendices have the same types of attacks: Combo, Junk, Scan, TCP, and UDP; all features have the same number of attributes with different number of instances based on the number of data that were collected. All data are numeric data). When comparing the occurrence in email with the total number of occurrences for each dataset, the difference is only 9.82% (see Appendix

B.3 for calculation). Thus, the feature ‘email’ is very important to identify the behaviour of malware.

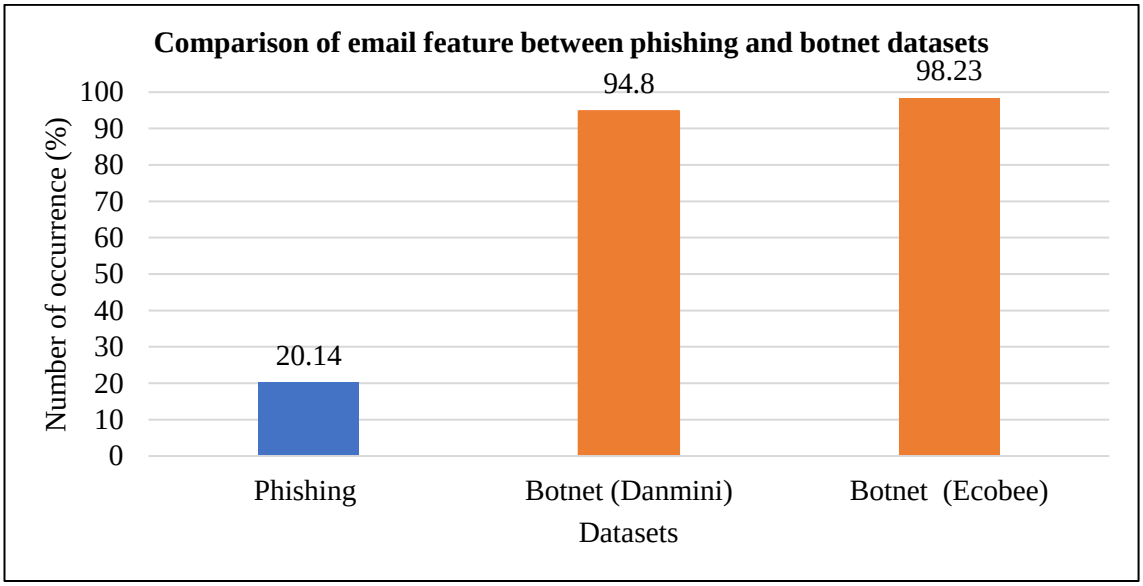


Figure 5.10: Number of occurrences of email feature in phishing and botnet datasets.

TCP feature in honeypot and botnet datasets

Figure 5.11 illustrates a similar pattern and relevance as Figure 5.10; the difference in occurrence between the honeypot and botnet (test analysis) datasets was 9.98% (see Appendix B.4 for calculation)). This clarifies that the feature TCP is related to malware behaviour but does not have similar relevance to combo, benign, scan.

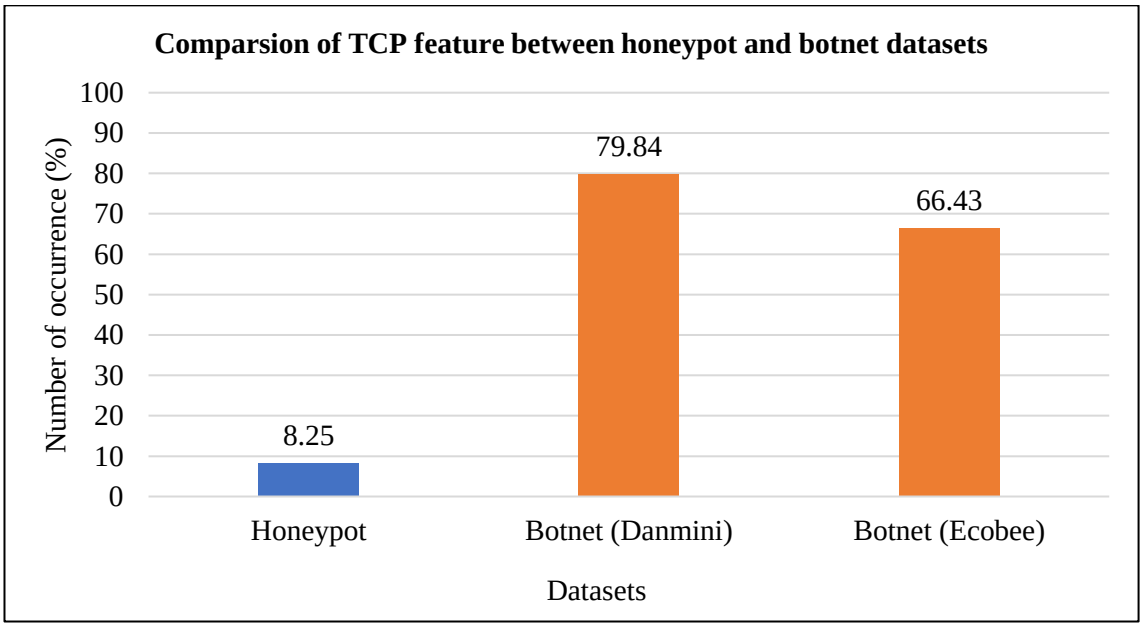


Figure 5.11: Number of occurrences of TCP feature in honeypot and botnet datasets.

UDP feature in honeypot and botnet datasets

UDP has a similar pattern of relevance to TCP. However, the average occurrence of UDP was less than the average occurrence of TCP in both the honeypot and botnet (test analysis) datasets. The average occurrence of TCP was 51.50% (see Figure 5.11) and the average UDP occurrence was 34.02% (Figure 5.12; see Appendix B.5 for calculation).

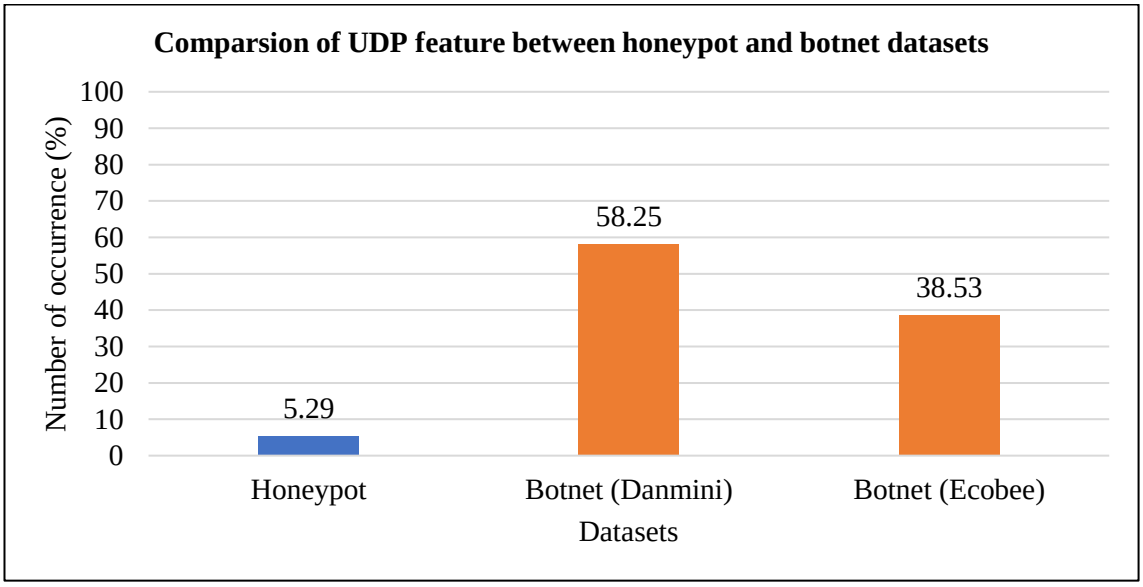


Figure 5.12: Number of occurrences of UDP feature in honeypot and botnet datasets.

5.2 Summary

This chapter discussed the findings relating to the three datasets by conducting a comparison of similar features. The discussion was based on the experimental results presented in Chapter 4 and highlighted some important results. This included malware behaviour in the phishing dataset and important information about Google index and page rank demonstrating that Google-indexed websites are safe to browse. The malware behaviour in the botnet dataset indicated that the scan feature obtained the highest percentage of accuracy, at around 99% in both Danmini and Ecobee. Interestingly, there was little difference between test and cross-validation results for the features benign, combo and scan for Danmini and Ecobee. The discussion included identification of the features shared between the three datasets: IP, port, URL, email, TCP and UDP. The conclusion is presented in Chapter 6 that includes future research direction to identify important issues for future study in the field of malware behaviour analysis.

Chapter 6
Conclusion and Future Work

This study focused on the malware behaviour of webpages. The main motivation for this thesis was the growing demand for information on feature selection in malware data to identify malware behaviour. The purpose of this study was to learn about and analyse the features that affect webpages. The conclusions mainly stem from Chapter 4, where the research findings and analysis are presented, and Chapter 5, where the results are discussed. The problem addressed in this research was outlined in Chapter 1. The current chapter presents recommendations that could be undertaken for future research. The analysis in this study was based on three datasets: phishing, botnet, honeypot. The first two were raw data gathered by UCI Machine Learning Repository and were analysed using Weka program. To achieve the aims of this study a number of objectives were outlined:

- Identify malware behaviour through feature selection.
- Determine influential features that have been targeted by attackers.
- Generate similarities between the properties of malicious webpages to identify the common target of exploitation.
- Predict malware vulnerability of specific features.

This research was designed to study malware behaviour on webpages. The ensemble method was used for data analysis in combination with the random tree as a classifier model. Several benefits were obtained by employing this combination. In this research, bagging (bootstrap aggregation) algorithm was selected for use. The study estimated and compared selected features—IP, port, URL, email, TCP, UDP—from the three datasets used in this thesis and these comparisons provided interesting, and useful results to identify malware behaviour.

Chapter 1 introduced the main objectives in line with problem statement and research. The flow of the thesis with the key contributions of the study was discussed. In Chapter 2, the literature relating to the research area was reviewed to provide a clear understanding to the research study. Types of malware and categories of honeypot were also outlined in detail. In Chapter 3, the research methodology and design were explained and evaluated. Several steps were adopted for the research methodology:

Step 1: Data collection, phishing and botnet datasets were taken from the UCI Machine Learning Repository and honeypot data were collected after deploying the MHN honeypot.

Step 2: Data description, the features for each dataset were listed in detail.

Step 3: Data preprocessing, Weka tool was used to generate an Excel file for the phishing dataset to enable examination and analysis using charts. The ensemble method with random tree was used for the botnet dataset to examine the raw data. MHN generated the honeypot data, which were saved in an Excel file to draw graphs.

Step 4: Data analysis, phishing and botnet datasets were analysed using the Weka software. In regarding to the experimental results from the honeypot data, RStudio was chosen as the best tool to calculate the frequencies of features.

The points to be noted from the research are:

i. Webpages ranked in Google index are less prone to malicious behaviour; malicious attack through ports is very infrequent. These findings clarify that if a port is secured and the site is Google indexed, malicious attack from websites can be minimised. Thus, this study contributes to identification of malware behaviour in web page; this information can be used to make web browsing safe and reliable.
ii. In the botnet dataset the feature 'Scan' had more than 99% of accuracy correlated with test performance (see Section 4.3.2).
iii. Ultimately, the main consideration of this research was malicious behaviour used by attackers against webpages. Information about the malware features studied here can be used by antivirus providers and vendors to focus on the main risky features and build appropriate algorithms to avoid them and secure end users over the internet.

6.1 Limitations of the Research

This thesis project identified and analysed malicious webpage behaviours; thus, the research was able to analyse only some randomly chosen features: IP, port, URL, email (junk), scan, TCP, UDP, combo, benign, usernames, passwords, web traffic, pop up window, Google index, page rank and HTTPS token. Second, the features analysed in this research were limited to phishing, botnet (Danmini and Ecobee) and honeypot data. Results for the feature occurrence in other platforms and digital devices may differ as they may have different setups. Similarly, the honeypot used in this experiment was limited to some of sensors (honeypots), three of which were used in this study: Snort, Kippo and Glastopf. Finally, the tool used to analyse the phishing and botnet datasets was

Weka and because of the limited time for the research, only two of test options were utilised: cross-validation and test analyses. Regarding the algorithms available in the ensemble method, only bagging was selected. In the same way, within the bagging algorithm there were different models for classifiers, but random tree was the best and most suitable for this study's experimental data.

6.2 Future Research Directions

The research was focused on analysis of malware behaviour of webpages. This thesis provides contributions in answering the research question "What research can be done to identify and analyse malicious webpage behaviour?" This section outlines the areas in which this work can be enhanced which would be a valuable contribution to this research in the future.

Analysis characteristics of botnet dataset

This thesis has focused on a malicious website that may affect different IoT devices. Gafgyt is one of the most common IoT botnet attack that affected Danmini and Ecobee devices was analysed in this research. However, there is another common IoT botnet attack called Mirai which was also within the botnet dataset. An in-depth study on Mirai attacks would be a useful contribution in order to understand and identify more features. Therefore, more analysis of IoT device such as Ennio which has also gafgyt attacks would improve the ability to identify even more webpages attacks.

Identifying behaviour of phishing and botnet datasets

In this thesis two classifier evaluation options 'cross validation' and 'percentage split' (also known as test analysis) are being used to analyse phishing and botnet datasets to find the percentage of accuracy. Both provided high percentage of accuracy of the results. However, only bagging (bootstrap aggregation) algorithm was chosen while there are variety of classifier techniques have not been performed in this thesis because of the limitation of time. As an extension to the work presented in this thesis, it would be particularly interesting and valuable to use other machine learning algorithms such as Neural network, Support vector machine and so on.

To collect data using different level interaction of honeypot

In this study, a low-interaction honeypot was selected and three types of sensors (honeypots) were applied in MHN to study the malware behaviour. Further, the security

risk and maintenance were thus less than they would be for high-interaction honeypots. Future research in this field may also include opportunities to carry out further honeypot experimental research. Other types of honeypot in the MHN software could be considered and included to extract more relevant data such as Dionaea and Cowrie, that were not considered in this study. Using high-interaction honeypots would provide more exposure to security risks and thus a deeper understanding of malware behaviours.

To analyse more features of malicious website

In this thesis, the researcher shed light on the important features of malicious website such as IP, port, URL, web traffic, junk, scan, combo, TCP, UDP, HTTPS token popup window, Google index and page rank. However, many other features are not conducted in this study, analysis of other types of malware behaviour is a logical extension to the study presented in this thesis. There are features still require examination to achieve a full understanding of malware, for example, the prefix/suffix was not studied in this research because of the limited time available for the study, URL length, double slash redirecting, having subdomain, favicon, links in tags, age of domain and DNS record, to make improvements to this field with regard to security aspects.

References

[1] K. Rieck, P. Trinius, C. Willems, and T. Holz, "Automatic analysis of malware behavior using machine learning," *Journal of Computer Security*, vol. 19, no. 4, pp. 639-668, 2011.

[2] Symantic, *Internet security threat report*, vol. 22, 2017.

[3] AVTEST, *Security report 2017/18*, The Independent IT-Security Institute, 2018.

[4] M. Wagner, F. Fischer, R. Luh, A. Haberson, A. Rind, D. A. Keim, W. Aigner, R. Borgo, F. Ganovelli, and I. Viola, "A survey of visualization systems for malware analysis," in EG Conference on Visualization (EuroVis)-STARs, 2015, pp. 105-125.

[5] S. Ranveer, and S. Hiray, "Comparative analysis of feature extraction methods of malware detection," *International Journal of Computer Applications*, vol. 120, no. 5, pp. 1-7, 2015.

[6] S. Yousaf, U. Iqbal, S. Farooqi, R. Ahmad, Z. Shafiq, and F. Zaffar, "Malware Slums: Measurement and Analysis of Malware on Traffic Exchanges," in Dependable Systems and Networks (DSN), 2016 46th Annual IEEE/IFIP International Conference on, 2016, pp. 572-582.

[7] N. Vratonjic, *Security, Privacy and Economics of Online Advertising*, EPFL, 2013.

[8] S. Bravo, and D. Mauricio, "DDoS attack detection mechanism in the application layer using user features," in Information and Computer Technologies (ICICT), 2018 International Conference on, 2018, pp. 97-100.

[9] V. Kumar, and K. Kumar, "Classification of DDoS attack tools and its handling techniques and strategy at application layer," in Advances in Computing, Communication, & Automation (ICACCA)(Fall), International Conference on, 2016, pp. 1-6.

[10] B. Nagpal, P. Sharma, N. Chauhan, and A. Panesar, "DDoS tools: Classification, analysis and comparison," in Computing for Sustainable Global Development (INDIACom), 2015 2nd International Conference on, 2015, pp. 342-346.

[11] G. Ranjith, J. Vijayachandra, B. Prathusha, and P. Sagarika, "Design and implementation of a defense system from TCP injection attacks," *Indian Journal of Science and Technology*, vol. 9, no. 40, 2016.

[12] D. Canali, D. Balzarotti, and A. Francillon, "The role of web hosting providers in detecting compromised websites," in Proceedings of the 22nd international conference on World Wide Web, 2013, pp. 177-188.

[13] D. R. Patil, and J. Patil, "Survey on malicious web pages detection techniques," *International Journal of U-and E-service, Science and Technology*, vol. 8, no. 5, pp. 195-206, 2015.

[14] M. Ceccato, P. Tonella, C. Basile, B. Coppens, B. De Sutter, P. Falcarin, and M. Torchiano, "How professional hackers understand protected code while performing attack tasks," in Program Comprehension (ICPC), 2017 IEEE/ACM 25th International Conference on, 2017, pp. 154-164.

[15] L. Batten, and G. Li, *Applications and techniques in information security : 6th International Conference, ATIS 2016, Cairns, QLD, Australia, October 26-28, 2016, Proceedings*: Singapore : Springer, 2016., 2016.

[16] McAfee, *McAfee labs threats report*, 2018.

[17] W. Fleshman, E. Raff, R. Zak, M. McLean, and C. Nicholas, "Static Malware Detection & Subterfuge: Quantifying the Robustness of Machine Learning and Current Anti-Virus," *arXiv:1806.04773*, 2018.

[18] R. J. Mangialardo, and J. C. Duarte, "Integrating static and dynamic malware analysis using machine learning," *IEEE Latin America Transactions,* vol. 13, no. 9, pp. 3080-3087, 2015.

[19] R. M. Mohammad, F. Thabtah, and L. McCluskey, "Intelligent rule-based phishing websites classification," *IET Information Security,* vol. 8, no. 3, pp. 153-160, 2014.

[20] R. M. Mohammad, F. Thabtah, and L. McCluskey, "Predicting phishing websites based on self-structuring neural network," *Neural Computing and Applications,* vol. 25, no. 2, pp. 443-458, 2014.

[21] R. M. Mohammad, F. Thabtah, and L. McCluskey, "An assessment of features related to phishing websites using an automated technique," in Internet Technology And Secured Transactions, 2012 International Conference for, 2012, pp. 492-497.

[22] Y. Meidan, M. Bohadana, Y. Mathov, Y. Mirsky, A. Shabtai, D. Breitenbacher, and Y. Elovici, "N-BaIoT—Network-Based Detection of IoT Botnet Attacks Using Deep Autoencoders," *IEEE Pervasive Computing,* vol. 17, no. 3, pp. 12-22, 2018.

[23] M. Shah, Jigneshkumar,, "Modern Honey Network " *International Journal of Research in Advent Technology*, pp. 156-162, 2016.

[24] L. Invernizzi, S. Miskovic, R. Torres, C. Kruegel, S. Saha, G. Vigna, S.-J. Lee, and M. Mellia, "Nazca: Detecting Malware Distribution in Large-Scale Networks," in NDSS, 2014, pp. 23-26.

[25] T. Thakur, and R. Verma, "Catching classical and hijack-based phishing attacks," in International Conference on Information Systems Security, 2014, pp. 318-337.

[26] A. Altaher, "Phishing websites classification using hybrid SVM and KNN approach," *International Journal of Advanced Computer Science and Applications,* vol. 8, no. 6, 2017.

[27] A. C. Bahnsen, E. C. Bohorquez, S. Villegas, J. Vargas, and F. A. González, "Classifying phishing URLs using recurrent neural networks," in Electronic Crime Research (eCrime), 2017 APWG Symposium on, 2017, pp. 1-8.

[28] K. Dunham, *Mobile malware attacks and defense*: Syngress, 2008.

[29] M. Khonji, Y. Iraqi, and A. Jones, "Phishing detection: a literature survey," *IEEE Communications Surveys & Tutorials,* vol. 15, no. 4, pp. 2091-2121, 2013.

[30] C. Murphy, and G. E. Kaiser, "Improving the dependability of machine learning applications," 2008.

[31] G. Liu, B. Qiu, and L. Wenyin, "Automatic detection of phishing target from phishing webpage," in Pattern Recognition (ICPR), 2010 20th International Conference on, 2010, pp. 4153-4156.

[32] S. Abu-Nimeh, D. Nappa, X. Wang, and S. Nair, "A comparison of machine learning techniques for phishing detection," in Proceedings of the anti-phishing working groups 2nd annual eCrime researchers summit, 2007, pp. 60-69.

[33] R. B. Basnet, and T. Doleck, "Towards developing a tool to detect phishing URLs: a machine learning approach," in 2015 IEEE International Conference on Computational Intelligence & Communication Technology, 2015, pp. 220-223.

[34] H. Huang, L. Qian, and Y. Wang, "A SVM-based technique to detect phishing URLs," *Information Technology Journal,* vol. 11, no. 7, pp. 921-925, 2012.

[35] S. Marchal, J. François, R. State, and T. Engel, "Phishstorm: Detecting phishing with streaming analytics," *IEEE Transactions on Network and Service Management,* vol. 11, no. 4, pp. 458-471, 2014.

[36] H. Choi, B. B. Zhu, and H. Lee, "Detecting Malicious Web Links and Identifying Their Attack Types," *WebApps,* vol. 11, pp. 11-11, 2011.

[37] M. N. Feroz, and S. Mengel, "Phishing URL detection using URL ranking," in Big Data (BigData Congress), 2015 IEEE International Congress on, 2015, pp. 635-638.

[38] K. Pradeepthi, and A. Kannan, "Performance study of classification techniques for phishing url detection," in Advanced Computing (ICoAC), 2014 Sixth International Conference on, 2014, pp. 135-139.

[39] L. A. T. Nguyen, B. L. To, H. K. Nguyen, and M. H. Nguyen, "A novel approach for phishing detection using URL-based heuristic," in Computing, Management and Telecommunications (ComManTel), 2014 International Conference on, 2014, pp. 298-303.

[40] C. L. Tan, and K. L. Chiew, "Phishing website detection using URL-assisted brand name weighting system," in Intelligent Signal Processing and Communication Systems (ISPACS), 2014 International Symposium on, 2014, pp. 054-059.

[41] H. Y. Abutair, and A. Belghith, "Using Case-Based Reasoning for Phishing Detection," *Procedia Computer Science,* vol. 109, pp. 281-288, 2017.

[42] M. A. Al-Garadi, A. Mohamed, A. Al-Ali, X. Du, and M. Guizani, "A Survey of Machine and Deep Learning Methods for Internet of Things (IoT) Security," *arXiv:1807.11023,* 2018.

[43] X. D. Hoang, and Q. C. Nguyen, "Botnet Detection Based On Machine Learning Techniques Using DNS Query Data," *Future Internet,* vol. 10, no. 5, pp. 43, 2018.

[44] A. K. Jain, and B. Gupta, "PHISH-SAFE: URL Features-Based Phishing Detection System Using Machine Learning," in Cyber Security: Proceedings of CSI 2015, 2018, pp. 467-474.

[45] Q. K. A. Mirza, I. Awan, and M. Younas, "CloudIntell: An intelligent malware detection system," *Future Generation Computer Systems,* vol. 86, pp. 1042-1053, 2018.

[46] A. Mohaisen, O. Alrawi, and M. Mohaisen, "Amal: High-fidelity, behavior-based automated malware analysis and classification," *computers & security,* vol. 52, pp. 251-266, 2015.

[47] A. Kumara, and C. Jaidhar, "Automated multi-level malware detection system based on reconstructed semantic view of executables using machine learning techniques at VMM," *Future Generation Computer Systems,* vol. 79, pp. 431-446, 2018.

[48] Z. Katzir, and Y. Elovici, "Quantifying the resilience of machine learning classifiers used for cyber security," *Expert Systems with Applications,* vol. 92, pp. 419-429, 2018.

[49] Z. Zhao, F. Morstatter, S. Sharma, S. Alelyani, A. Anand, and H. Liu, "Advancing feature selection research," *ASU feature selection repository,* pp. 1-28, 2010.

[50] R. B. Basnet, A. H. Sung, and Q. Liu, "Feature selection for improved phishing detection," in International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems, 2012, pp. 252-261.

[51] K. D. Rajab, "New hybrid features selection method: a case study on websites phishing," *Security and Communication Networks,* vol. 2017, 2017.

[52] R. B. Basnet, A. H. Sung, and Q. Liu, "Learning to detect phishing URLs," *International Journal of Research in Engineering and Technology,* vol. 3, no. 6, pp. 11-24, 2014.

[53] V. Harrison, and J. Pagliery, "Nearly 1 million new malware threats released every day [Online]. CNN Money," 2015.

[54] AV-TEST, *AV-TEST The Independent IT-Security Institute,* 2017.

[55] U. Buergi, and F. Angehrn, "Systems and methods for identifying phishing websites," Google Patents, 2017.

[56] J. Niemelä, *Malware detection*, US 8,844,038 B2,to F-Secure Oyj, 2014.

[57] A. Pandey, and J. R. Saini, "Attacks & defense mechanisms for TCP/IP based protocols," *International Journal of Engineering Innovations and Research,* vol. 3, no. 1, pp. 17, 2014.

[58] CertNZ. "Unauthorised access," [Online]. Available: https://www.cert.govt.nz/businesses-and-individuals/explore/unauthorised-access/?topic=unauthorised-access.

[59] X. Li, J. Wang, and X. Zhang, "Botnet Detection Technology Based on DNS," *Future Internet,* vol. 9, no. 4, pp. 55, 2017.

[60] A. Ross, "Network Attack and Defense," University Of Haifa, 2015.

[61] J. W. Seo, and S. J. Lee, "A study on efficient detection of network-based IP spoofing DDoS and malware-infected Systems," *SpringerPlus,* vol. 5, no. 1, pp. 1878, 2016.

[62] P. Black, I. Gondal, and R. Layton, "A survey of similarities in banking malware behaviours," *Computers & Security,* vol. 77, pp. 756-772, 2018.

[63] C. Alex, and V. Angel. "Threat Spotlight: Dyre/Dyreza: An Analysis to Discover the DGA," Accessed: March 2015; [Online]. Available: https://blogs.cisco.com/security/talos/threat-spotlight-dyre.

[64] B. S. Gross, and P. Khandhar. "Dyre Banking Trojan," Accessed on: 15 October 2017; [Online]. Available: https://www.secureworks.com/research/dyre-banking-trojan.

[65] T. Micro. "A closer look at dyre malware," Accessed on: 2014; [Online]. Available: http://blog.trendmicro.com/trendlabs-security-intelligence/a-closer-look-at-dyre-malware-part-1/

[66] J. Kroustek, "Analysis of banking trojan vawtrak," *Technical report, AVG,* 2015.

[67] J. Wyke, "Vawtrak–International Crimeware-as-a-Service," Sophos, 2014.

[68] K. John, M. Lance, and Limor Kessem, *The dyre wolf: Attacks on corporate banking accounts* IBM, 2015.

[69] M. N. Sakib, and C.-T. Huang, "Using anomaly detection based techniques to detect HTTP-based botnet C&C traffic," in Communications (ICC), 2016 IEEE International Conference on, 2016, pp. 1-6.

[70] N. Kheir, G. Blanc, H. Debar, J. Garcia-Alfaro, and D. Yang, "Automated classification of C&C connections through malware URL clustering," in IFIP International Information Security Conference, 2015, pp. 252-266.

[71] F. Tariq, and S. Baig, "Machine Learning Based Botnet Detection in Software Defined Networks," *International Journal of Security and its Applications,* vol. 11, no. 11, pp. 1-11, 2017.

[72] D. Acarali, M. Rajarajan, N. Komninos, and I. Herwono, "Survey of approaches and features for the identification of HTTP-based botnet traffic," *Journal of Network and Computer Applications,* vol. 76, pp. 1-15, 2016.

[73] M. Kaytan, and D. Hanbay, "Effective classification of phishing web pages based on new rules by using extreme learning machines," *Anatolian Journal of Computer Sciences,* vol. 2, no. 1, pp. 15-36, 2017.

[74] S. Nivedha, S. Gokulan, C. Karthik, R. Gopinath, and R. Gowshik, "Improving Phishing URL Detection Using Fuzzy Association Mining," *The International Journal of Engineering and Science (IJES),* vol. 6, 2017.

[75] C. Whittaker, B. Ryner, and M. Nazif, "Large-Scale Automatic Classification of Phishing Pages," in NDSS, 2010, pp. 2010.

[76] Y. Takata, S. Goto, and T. Mori, "Analysis of redirection caused by web-based malware," *Proceedings of the Asia-Pacific advanced network,* vol. 32, pp. 53-62, 2011.

[77] Y. Tanaka, M. Akiyama, and A. Goto, "Analysis of malware download sites by focusing on time series variation of malware," *Journal of computational science,* vol. 22, pp. 301-313, 2017.

[78] J. Vargas, A. C. Bahnsen, S. Villegas, and D. Ingevaldson, "Knowing your enemies: Leveraging data analysis to expose phishing patterns against a major US financial institution," in Electronic Crime Research (eCrime), 2016 APWG Symposium on, 2016, pp. 1-10.

[79] J. Ma, L. K. Saul, S. Savage, and G. M. Voelker, "Learning to detect malicious urls," *ACM Transactions on Intelligent Systems and Technology (TIST),* vol. 2, no. 3, pp. 30, 2011.

[80] ICANN, "Beginner's guide to Internet protocol (IP) addresses," 2011.

[81] S. Bhunia, S. Ray, and S. Sur-Kolay, *Fundamentals of IP and SoC Security: Design, Verification, and Debug*: Springer, 2017.

[82] J. Gharibshah, T. C. Li, A. Castro, K. Pelechrinis, E. E. Papalexakis, and M. Faloutsos, "Mining actionable information from security forums: the case of malicious IP addresses," *arXiv:1804.04800,* 2018.

[83] J. Postel, *User datagram protocol,* 2070-1721, 1980.

[84] B. A. Navamani, C. Yue, and X. Zhou, “An Analysis of Open Ports and Port Pairs in EC2 Instances,” in Cloud Computing (CLOUD), 2017 IEEE 10th International Conference on, 2017, pp. 790-793.

[85] Y. B. Luo, B. S. Wang, and G. L. Cai, “Analysis of port hopping for proactive cyber defense,” *International Journal of Security and its Applications,* vol. 9, no. 2, pp. 123-134, 2015.

[86] N. Hoque, M. H. Bhuyan, R. C. Baishya, D. K. Bhattacharyya, and J. K. Kalita, “Network attacks: Taxonomy, tools and systems,” *Journal of Network and Computer Applications,* vol. 40, pp. 307-324, 2014.

[87] D. Stuttard, and M. Pinto, *The web application hacker's handbook: Finding and exploiting security flaws*: John Wiley & Sons, 2011.

[88] L. D. Jackson, “Introduction to the Internet and Web Page Design,” Southern Utah University, 2009.

[89] P. Lamprakis, “Human or malware? Detection of malicious Web requests,” ETH Zürich, TIK Institut für Technische Informatik und Kommunikationsnetze, 2016.

[90] N. Särökaari, “How to identify malicious HTTP Requests,” *SANS Institute,* pp. 25, 2012.

[91] L. Machlica, K. Bartos, and M. Sofka, “Learning detectors of malicious web requests for intrusion detection in network traffic,” *arXiv:1702.02530,* 2017.

[92] S. B. Rathod, and T. M. Pattewar, “A comparative performance evaluation of content based spam and malicious URL detection in E-mail,” in Computer Graphics, Vision and Information Security (CGVIS), 2015 IEEE International Conference on, 2015, pp. 49-54.

[93] V. K. Devendran, H. Shahriar, and V. Clincy, “A comparative study of email forensic tools,” *Journal of Information Security,* vol. 6, no. 2, pp. 111, 2015.

[94] M. Haverbeke, *Eloquent JavaScript, 3rd edition*, 2014.

[95] D. Flanagan, and G. M. Novak, "Java-Script: The Definitive Guide," AIP, 1998.

[96] K. Cabaj, and P. Gawkowski, “HoneyPot systems in practice,” *Przegląd Elektrotechniczny,* vol. 91, no. 2, pp. 63-67, 2015.

[97] S. Kaur, and H. Kaur, “Client Honeypot Based Malware Program Detection Embedded Into Web Pages,” *International Journal of Engineering Research and Applications,* vol. 3, no. 6, pp. 849-854, 2013.

[98] L. Spitzner, "Honeypots: Tracking hackers Addison Wesley Professional," September, 2002.

[99] C. Döring, and H. Erbs, “Improving network security with honeypots,” *Honeypot project: Master’s thesis. Darmstadt: University of Applied Sciences Darmstadt*, 2005.

[100] Y.-D. Lin, C.-Y. Lee, Y.-S. Wu, P.-H. Ho, F.-Y. Wang, and Y.-L. Tsai, “Active versus passive malware collection,” *Computer,* vol. 47, no. 4, pp. 59-65, 2014.

[101] B. F. L. M. Sousa, R. P. da Cunha Neto, and F. d. F. de Lima Sousa, “Honeypots and mobile technology: discovering the attacker,” in Proceedings of the ASEE 2014 Zone I Conference, University of Bridgeport, Bridgeport, CT, April, 2014, pp. 3-5.

[102] K. Rakshitha, S, , A. Prajna, M, , S. Roopashree, and N. Poojitha, “Campus security using honeypot,” in International Conference on Advances in Computer and Electrical Engineering (ICACEE'2012), Manila (Philippines) 2012, pp. 75-78.

[103] E. Peter, and T. Schiller, “A practical guide to honeypots,” *Washington Univerity*, 2011.

[104] D. Akkaya, and F. Thalgott, "Honeypots in network security," 2010.

[105] I. Mokube, and M. Adams, “Honeypots: concepts, approaches, and challenges,” in Proceedings of the 45th annual southeast regional conference, 2007, pp. 321-326.

[106] D. Joho, and R. Riedl, “Active Honeypots,” Citeseer, 2004.

[107] R. Baumann, “Honeyd–A low involvement Honeypot in Action,” *Originall published as part of the GCIA practical*, pp. 14, 2003.

[108] S. Maitri, and V. Pranav, “Honeypot: Concepts, Types and Working ” *International Journal of Engineering Development and Research,* vol. 3, no. 4, pp. 596-598, 2015.

[109] C. Seifert, I. Welch, and P. Komisarczuk, “Taxonomy of honeypots,” 2006.

[110] V. Nicomette, M. Kaâniche, E. Alata, and M. Herrb, “Set-up and deployment of a high-interaction honeypot: experiment and lessons learned,” *Journal in computer virology,* vol. 7, no. 2, pp. 143-157, 2011.

[111] J. Ma, K. Chai, Y. Xiao, T. Lan, and W. Huang, “High-Interaction Honeypot System for SQL Injection Analysis,” in Information Technology, Computer Engineering and Management Sciences (ICM), 2011 International Conference on, 2011, pp. 274-277.

[112] T. H. project. "Capture-HPC Client Honeypot / Honeyclient," Accessd on: September 2008; [Online]. Available: https://projects.honeynet.org/capture-hpc.

[113] A. Bendovschi, “Cyber-attacks–trends, patterns and security countermeasures,” *Procedia Economics and Finance*, vol. 28, pp. 24-31, 2015.

[114] E. Skoudis, and L. Zeltser, *Malware: Fighting malicious code*: Prentice Hall Professional, 2004.

[115] M. Sikorski, and A. Honig, *Practical malware analysis: the hands-on guide to dissecting malicious software*: William Pollock, 2012.

[116] M. E. Newman, S. Forrest, and J. Balthrop, “Email networks and the spread of computer viruses,” *Physical Review E,* vol. 66, no. 3, pp. 035101, 2002.

[117] A. Hardikar, and J. Bambenek, “Malware 101-Viruses,” *SANS Institute*, 2008.

[118] S. Divya, and G. Padmavathi, “Computer Network Worms Propagation and its Defence Mechanisms: A Survey,” in Proc. of Int. Conf. on Advances in Communication, Network, and Computing, CNC, 2014, pp. 643-652.

[119] B. Rajesh, Y. J. Reddy, and B. D. K. Reddy, “A survey paper on malicous computer worms,” *International Journal of Advanced Research in Computer Science and Technology,* vol. 3, no. 2, pp. 161-167, 2015.

[120] Y. Tang, J. Luo, B. Xiao, and G. Wei, “Concept, characteristics and defending mechanism of worms,” *IEICE Transactions on Information and Systems,* vol. 92, no. 5, pp. 799-809, 2009.

[121] T. F. Stafford, and A. Urbaczewski, “Spyware: The ghost in the machine,” *The Communications of the Association for Information Systems,* vol. 14, no. 1, pp. 49, 2004.

[122] M. Boldt, B. Carlsson, and A. Jacobsson, “Exploring spyware effects,” in Nordsec 2004, 2004.

[123] H. R. Zeidanloo, F. Tabatabaei, P. V. Amoli, and A. Tajpour, “All About Malwares (Malicious Codes),” in Security and Management, 2010, pp. 342-348.

[124] D. Lobo, P. Watters, X.-W. Wu, and L. Sun, “Windows rootkits: Attacks and countermeasures,” in Cybercrime and Trustworthy Computing Workshop (CTC), 2010 Second, 2010, pp. 69-78.

[125] A. J. M. de Almeida, “Rootkits-Detection and prevention,” 2008.

[126] J. Crapanzano, “Deconstructing subseven, the trojan horse of choice,” *SANS Institute,* 2003.

[127] N. Provos, D. McNamee, P. Mavrommatis, K. Wang, and N. Modadugu, “The Ghost in the Browser: Analysis of Web-based Malware,” *HotBots,* vol. 7, pp. 4-4, 2007.

[128] I. Ideses, and A. Neuberger, “Adware detection and privacy control in mobile devices,” in Electrical & Electronics Engineers in Israel (IEEEI), 2014 IEEE 28th Convention of, 2014, pp. 1-5.

[129] B. Uscilowski, “Mobile adware and malware analysis,” *Symantec Corp,* vol. 1, 2013.

[130] Cisco, *Annual Cybersecurity Report*, 2018.

[131] M. R. Hasan, H. Gholamhosseini, and N. I. Sarkar, “A new ensemble classifier for multivariate medical data,” in 2017 27th International Telecommunication Networks and Applications Conference (ITNAC), 2017, pp. 1-6.

[132] M. R. Hasan, F. Siraj, and M. S. Sainin, “Improving ensemble decision tree performance using Adaboost and Bagging,” in AIP Conference Proceedings, 2015, pp. 030008.

[133] O. T. Yıldız, O. İrsoy, and E. Alpaydın, "Bagging soft decision trees," *in Machine Learning for Health Informatics,* pp. 25-36: Springer, 2016.

[134] I. H. Witten, E. Frank, M. A. Hall, and C. J. Pal, *Data Mining: Practical machine learning tools and techniques*: Morgan Kaufmann, 2016.

[135] G. Ramesh, J. Gupta, and P. Gamya, “Identification of phishing webpages and its target domains by analyzing the feign relationship,” *Journal of Information Security and Applications,* vol. 35, pp. 75-84, 2017.

[136] X. Dong, J. A. Clark, and J. L. Jacob, “Defending the weakest link: phishing websites detection by analysing user behaviours,” *Telecommunication Systems,* vol. 45, no. 2-3, pp. 215-226, 2010.

[137] C. Perez, M. Lemercier, B. Birregah, and A. Corpel, “SPOT 1.0: Scoring Suspicious Profiles on Twitter,” in International Conference on Advances in Social Networks Analysis and Mining Advances in Social Networks Analysis and Mining (ASONAM), 2011, pp. 377-381.

[138] M. Wu, and M. Yang, “Privacy Preservation for Detecting Malicious Web Sites from Suspicious URLs,” in Business Computing and Global Informatization (BCGIN), 2011 International Conference on, 2011, pp. 400-403.

[139] K. Angrishi, “Turning internet of things (iot) into internet of vulnerabilities (iov): Iot botnets,” *arXiv:1702.03681*, 2017.

[140] S. Singhal, and M. Jena, “A study on WEKA tool for data preprocessing, classification and clustering,” *International Journal of Innovative technology and exploring engineering (IJItee),* vol. 2, no. 6, pp. 250-253, 2013.

[141] H.-J. Boehm, “How to Miscompile Programs with" Benign" Data Races,” in HotPar, 2011, pp. 3-3.

[142] Y. Mirsky, T. Doitshman, Y. Elovici, and A. Shabtai, “Kitsune: an ensemble of autoencoders for online network intrusion detection,” *arXiv preprint arXiv:1802.09089*, 2018.

[143] R. U. Rehman, *Intrusion detection systems with Snort: advanced IDS techniques using Snort, Apache, MySQL, PHP, and ACID*: Prentice Hall Professional, 2003.

[144] B. Mphago, O. Bagwasi, B. Phofuetsile, and H. Hlomani, “Deception in dynamic web application honeypots: Case of glastopf,” in Proceedings of the International Conference on Security and Management (SAM), 2015, pp. 104.

[145] Intel, *McAfee AntiVirus for Education*, 2017.

[146] T. Kelsey, and B. Lyon, *Introduction to search engine optimization : a guide for absolute beginners*: Apress, 2017.

[147] P. Seshagiri, A. Vazhayil, and P. Sriram, “AMA: Static code analysis of web page for the detection of malicious scripts,” *Procedia Computer Science*, vol. 93, pp. 768-773, 2016.

[148] V. L. Pochat, T. Van Goethem, and W. Joosen, “Rigging Research Results by Manipulating Top Websites Rankings,” *arXiv:1806.01156v2*, 2018.

[149] M. P. Evans, “Analysing Google rankings through search engine optimization data,” *Internet research,* vol. 17, no. 1, pp. 21-37, 2007.

[150] N. Kheir, F. Tran, P. Caron, and N. Deschamps, “Mentor: positive DNS reputation to skim-off benign domains in botnet C&C blacklists,” in IFIP International Information Security Conference, 2014, pp. 1-14.

Appendices

Appendix A

Additional Screenshots for Chapter 3

A.1 Botnet dataset (Danmini)

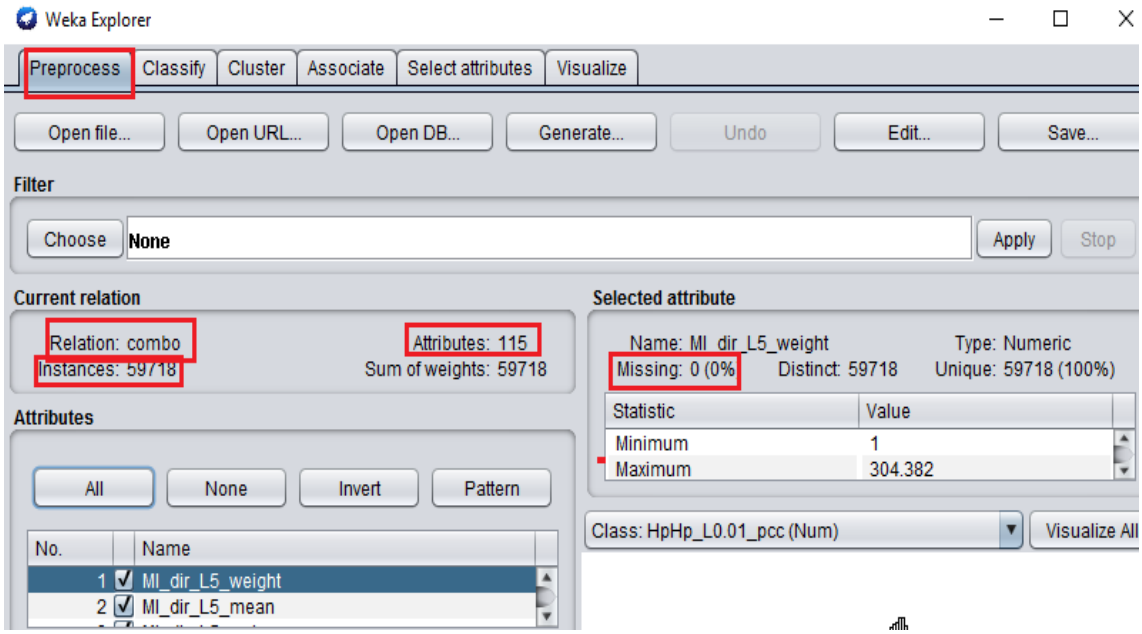


Figure A.1: Combo (Danmini)

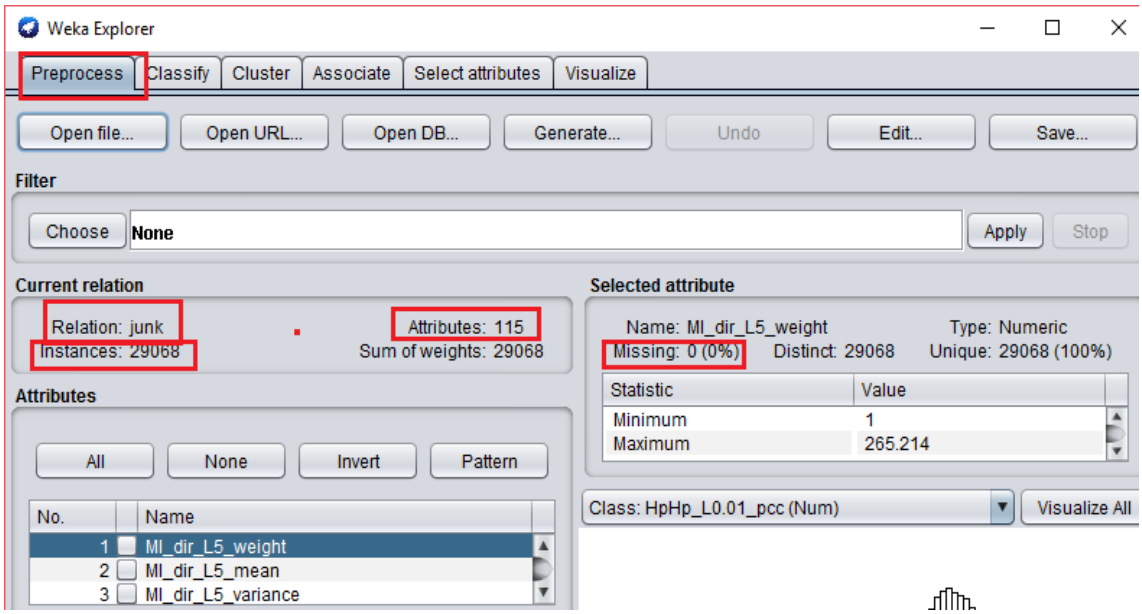


Figure A.2: Junk (Danmini)

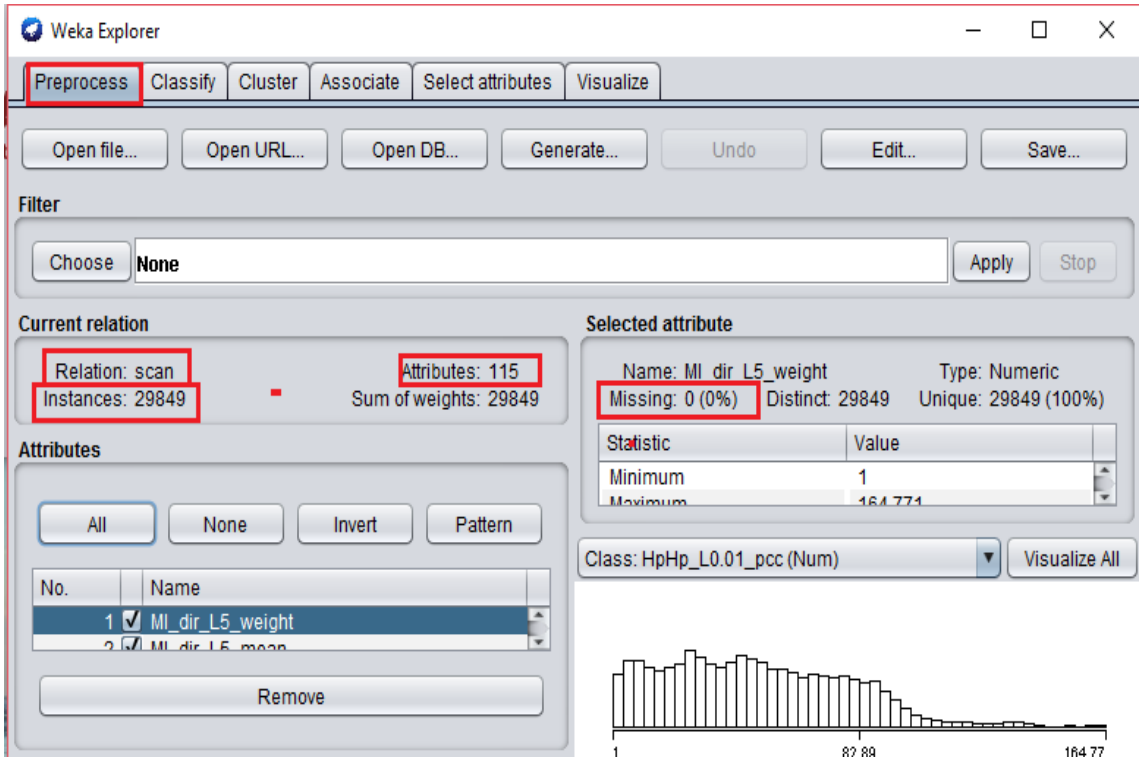


Figure A.3: Scan (Danmini)

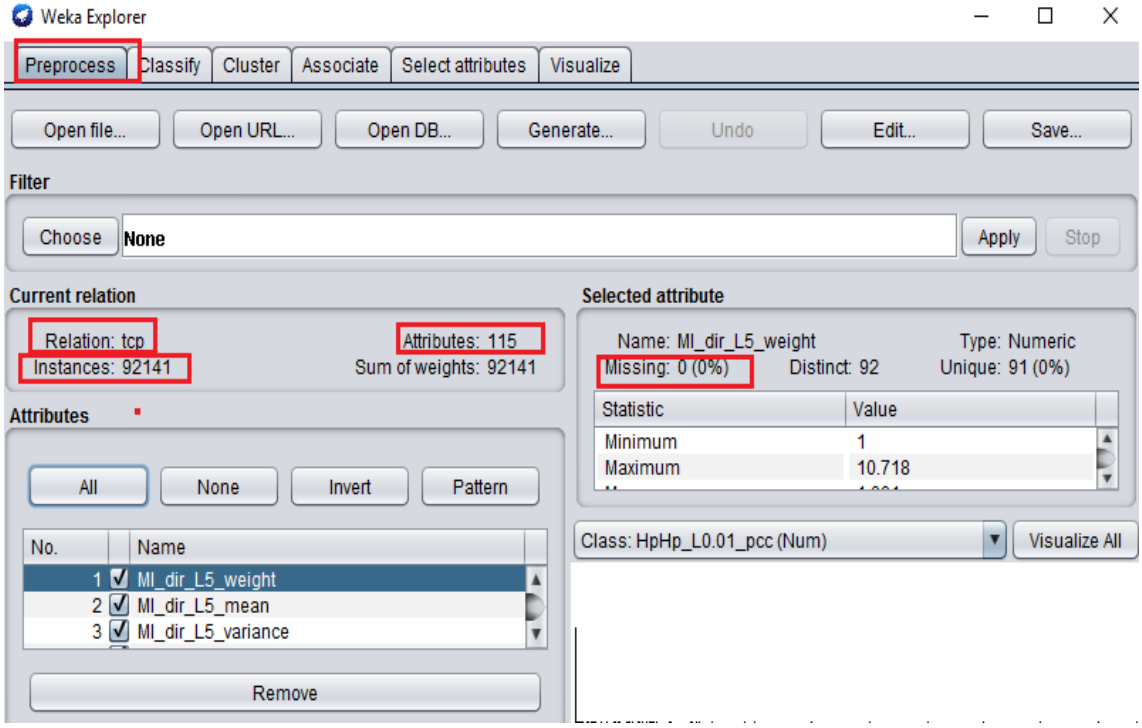


Figure A.4: TCP (Danmini)

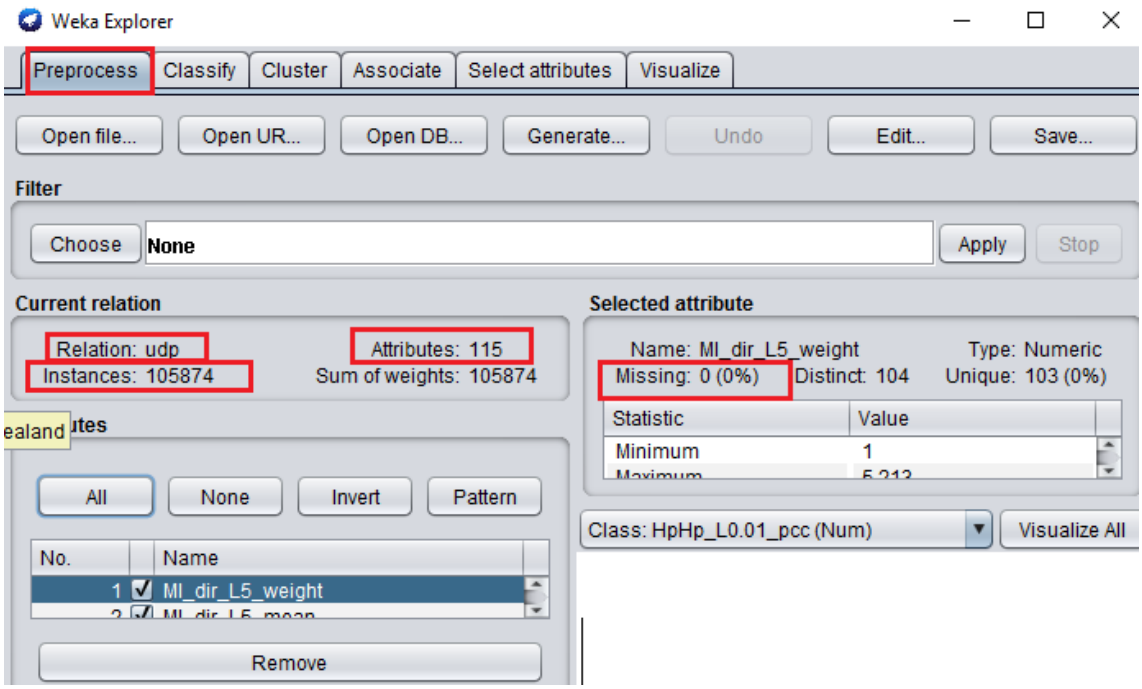


Figure A.5: UDP (Danmini)

A.2 Botnet dataset (Ecobee)

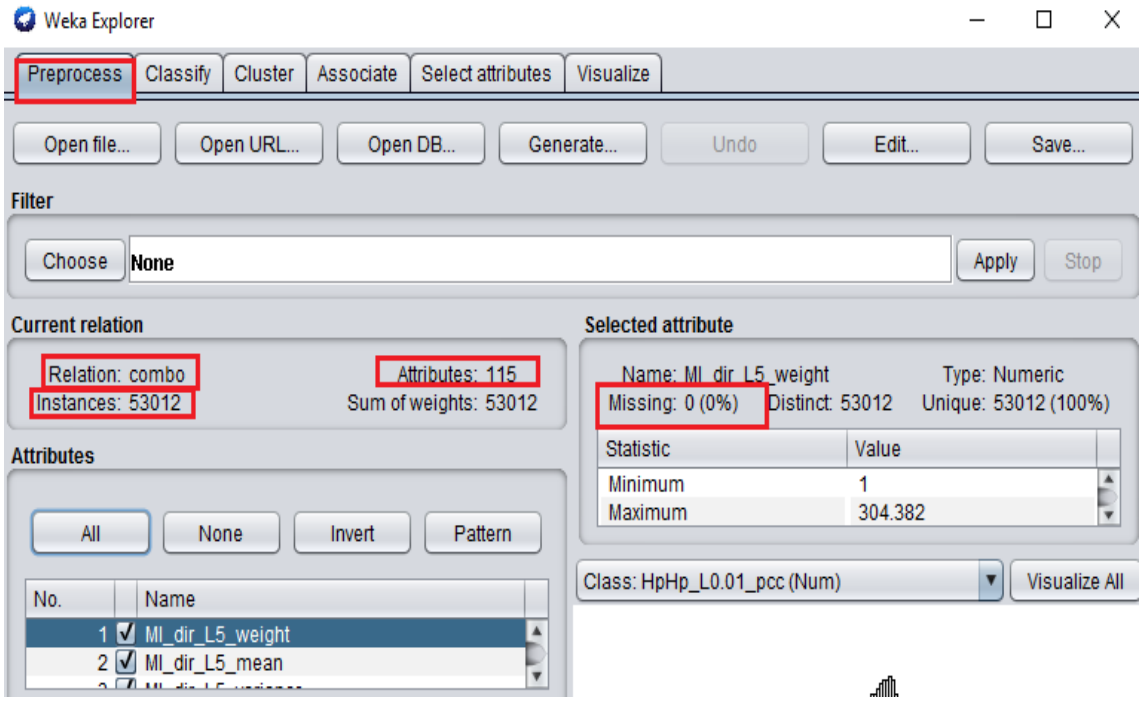


Figure A.6: Combo (Ecobee)

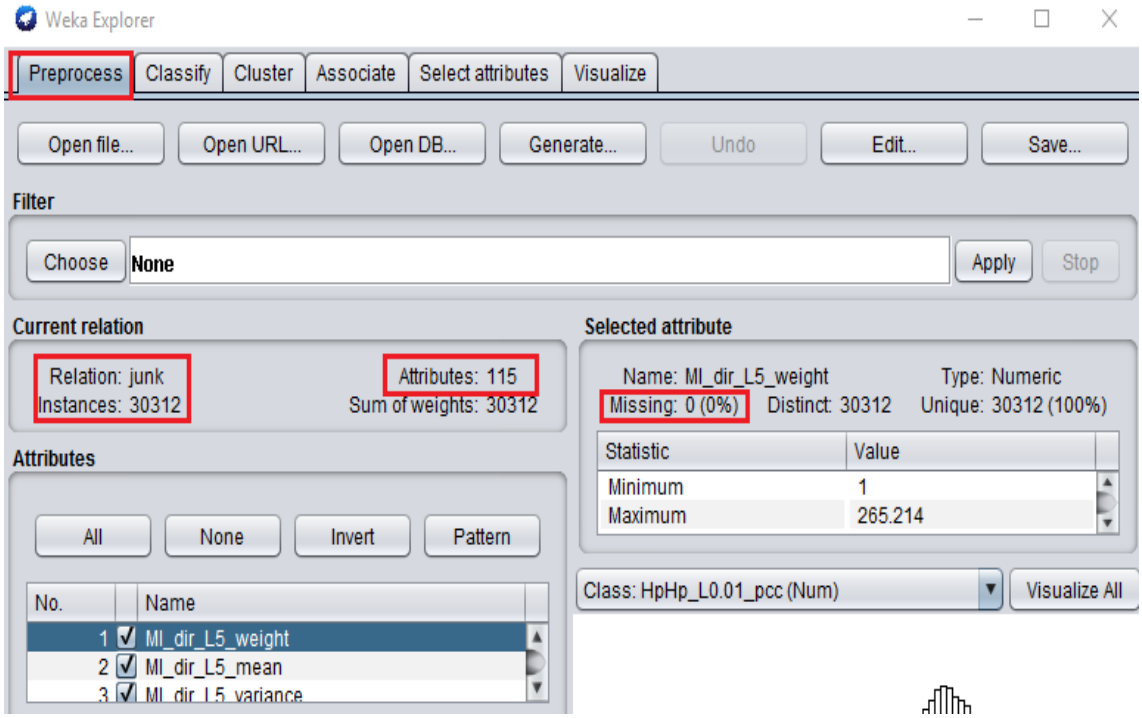


Figure A.7: Junk (Ecobee)

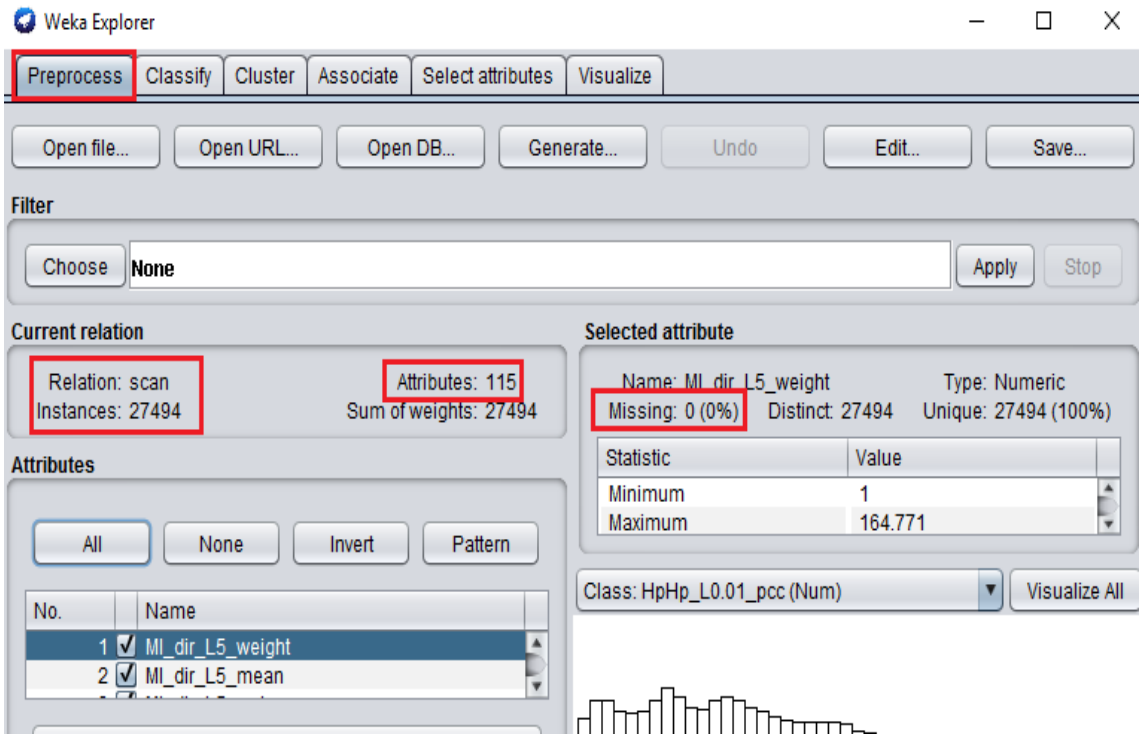


Figure A.8: Scan (Ecobee)

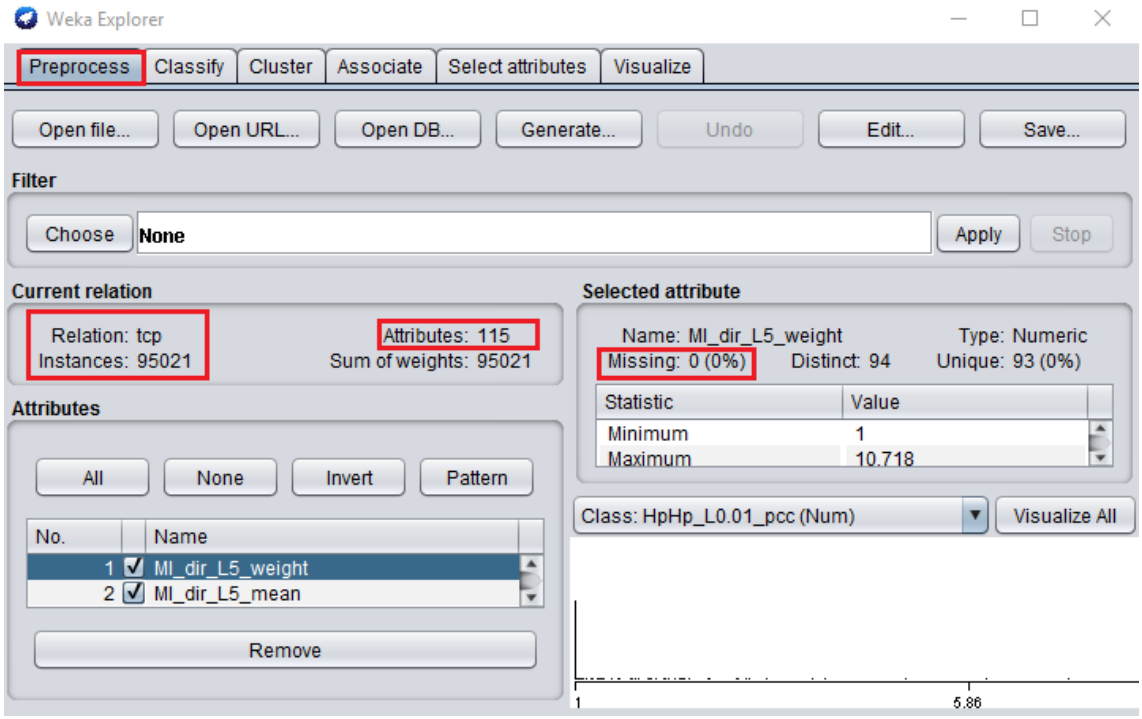


Figure A.9: TCP (Ecobee)

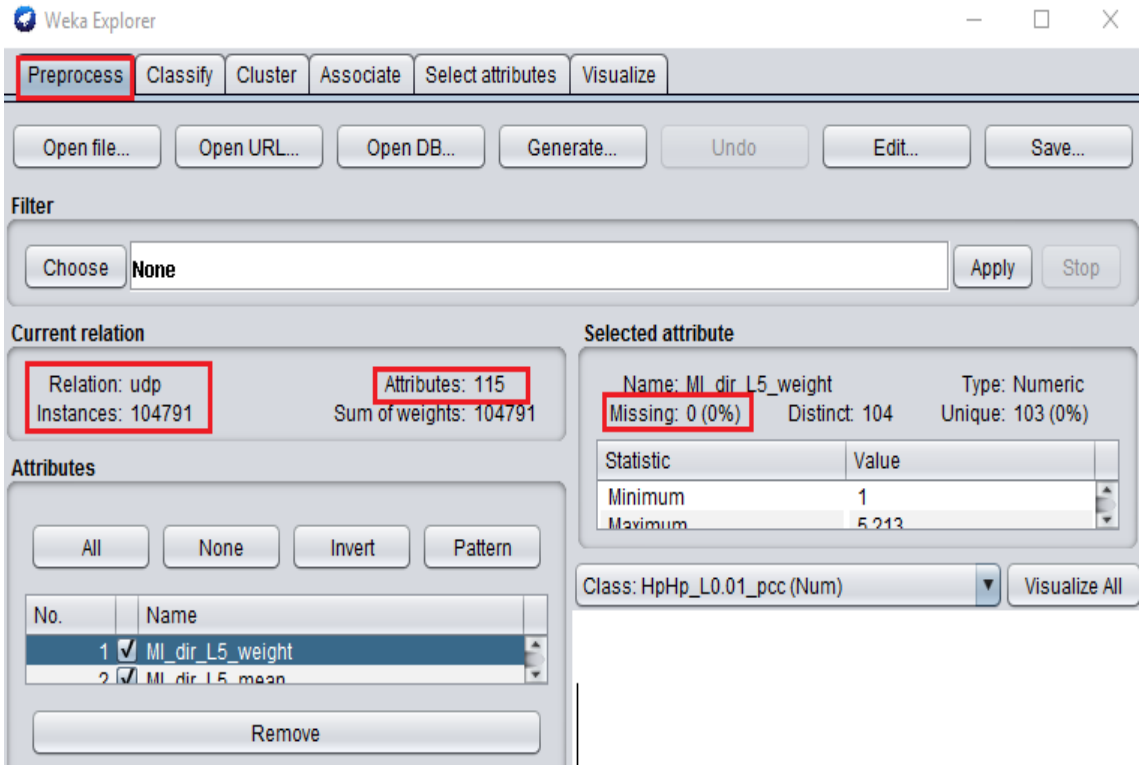
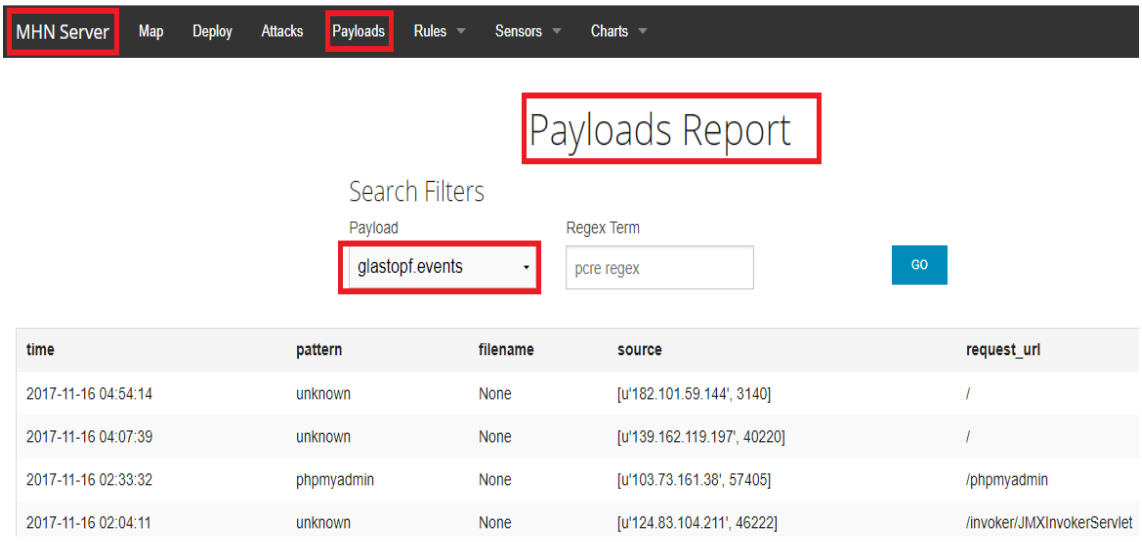


Figure A.10: UDP (Ecobee)

A.3 Honeypot (Payload Report)



time	pattern	filename	source	request_url
2017-11-16 04:54:14	unknown	None	[u'182.101.59.144', 3140]	/
2017-11-16 04:07:39	unknown	None	[u'139.162.119.197', 40220]	/
2017-11-16 02:33:32	phpmyadmin	None	[u'103.73.161.38', 57405]	/phpmyadmin
2017-11-16 02:04:11	unknown	None	[u'124.83.104.211', 46222]	/invoker/JMXInvokerServlet

Figure A.11: Payloads report of Glastopf.events

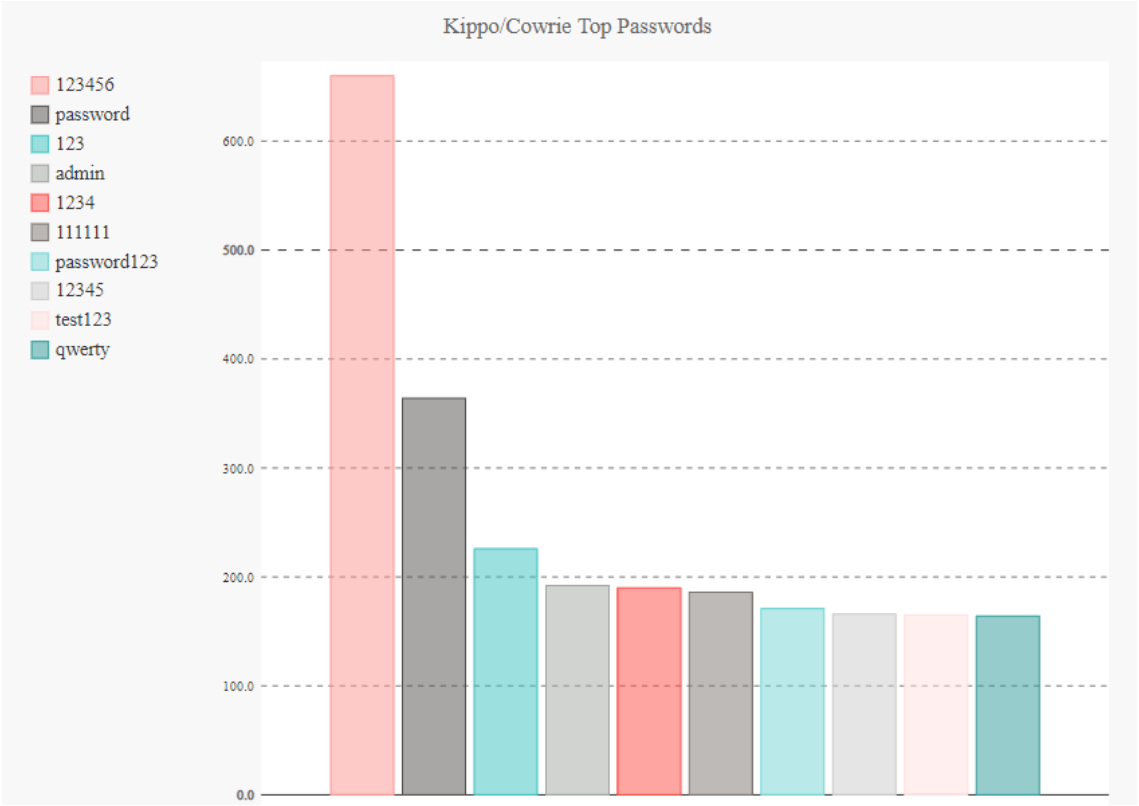


Figure A.12: Kippo top passwords

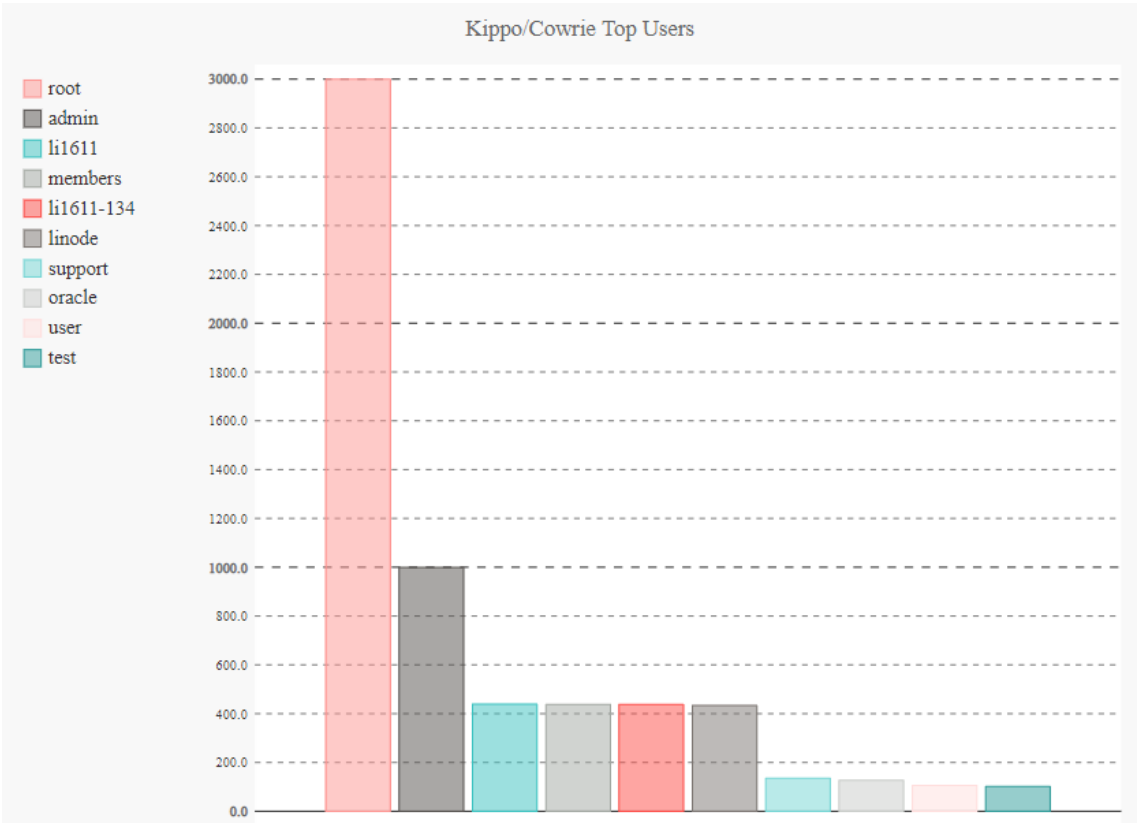


Figure A.13: Kippo top usernames

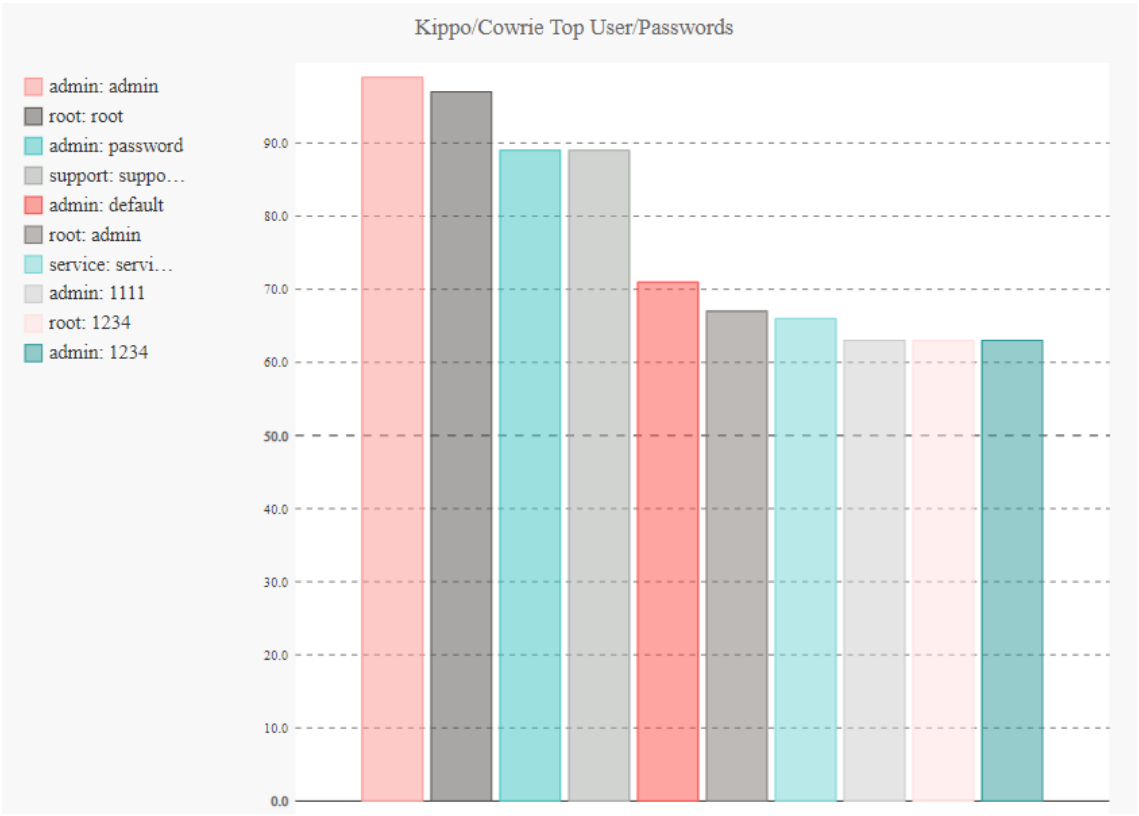


Figure A.14: Kippo top usernames/passwords

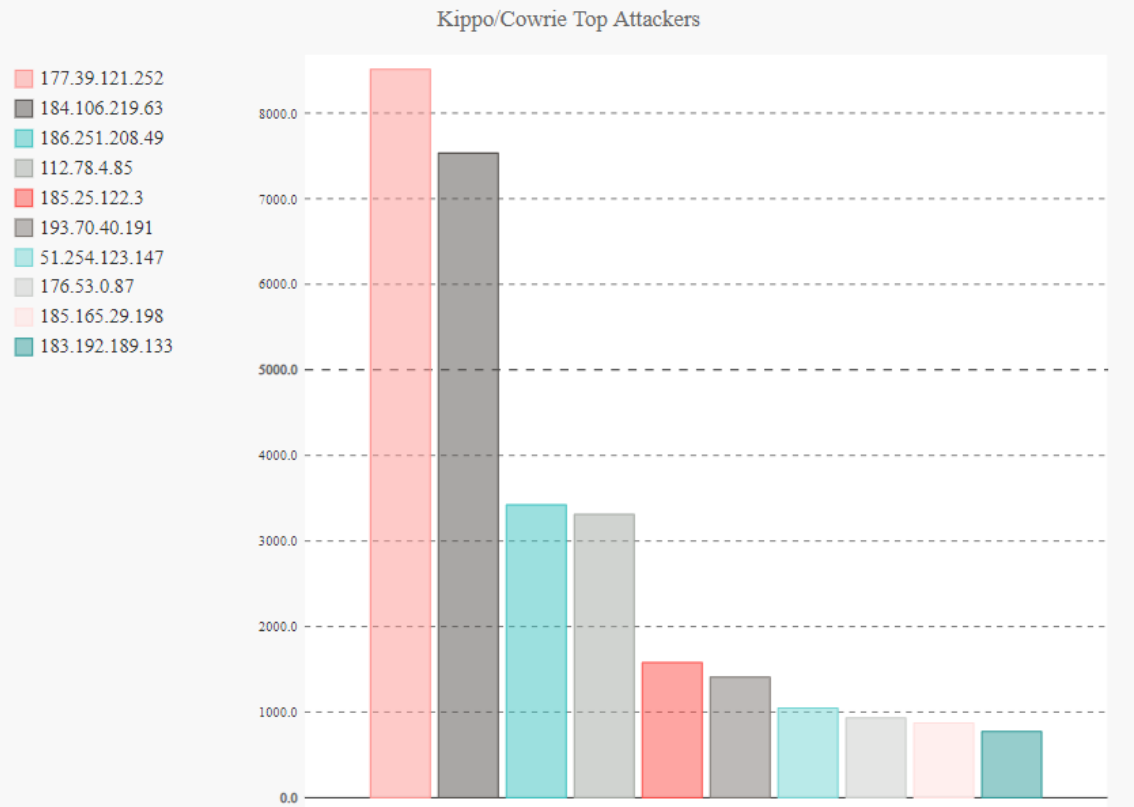


Figure A.15: Kippo top attackers

A.4 Honeypot (Deploy MHN)

Sudo apt-get install git

```
root@ubuntu: ~
Local time is now:      Tue Oct 17 11:15:54 NZDT 2017.
Universal Time is now:  Mon Oct 16 22:15:54 UTC 2017.

root@ubuntu:~# date
Tue Oct 17 11:16:00 NZDT 2017
root@ubuntu:~# sudo apt-get install git
Reading package lists... Done
Building dependency tree
```

Figure A.16: The first command for deploying MHN server

- Sudo (superuser do): use if the user of the system is a normal user who does not have all privileges on the system. This allows a normal is to build a more secure account to avoid being controlled by an attacker.
- Apt-get (advanced packaging tool): used to accept packages.
- Install: used to install the required packages.
- Git: assists with downloading the required program from the internet.

Cd /opt:

```
root@ubuntu:~# cd /opt
```

Figure A.17: The second command for deploying MHN server

- Cd (change directory): used to move to another directory in the system.
- Opt (optional directory): used to add software packages.

Sudo git clone https://github.com/threatstream/mhn.git:

```
root@ubuntu:/opt# sudo git clone https://github.com/threatstream/mhn.git
Cloning into 'mhn'...
remote: Counting objects: 6567, done.
remote: Compressing objects: 100% (88/88), done.
remote: Total 6567 (delta 71), reused 75 (delta 37), pack-reused 6442
Receiving objects: 100% (6567/6567), 3.53 MiB | 1.30 MiB/s, done.
Resolving deltas: 100% (3413/3413), done.
Checking connectivity... done.
```

Figure A.18: The third command for deploying MHN server.

- Clone: used in the command line to generate a local copy of the required code.

Cd./mhn/:

```
root@ubuntu:/opt# cd ./mhn/
root@ubuntu:/opt/mhn# ls
flags-LICENSE.txt  install.sh  LICENSE  README.md  scripts  server
```

Figure A.19: The fourth command for deploying MHN server.

- MHN: the MHN directory.

Appendix B

Additional Results for Chapter 5

B.1 Calculation of page rank feature

Error! Reference source not found., by dividing the number of phishy behaviour by the total number of attacks:

- Total number of phishy behaviour = 8,201
- Total number of attacks = 11,055
- 8,201/ 11,055 = 0.7418
- 0.7418 * 100 = 74.18 %

B.2 Calculation of average phishy features

Figure 5.2, by combining the percentage results of the following features:

- Having IP address = 34%
- Request URL = 40%
- Web traffic = 23%
- Page rank = 74%
- 34 + 40 + 23 + 74 = 171, then divided the result by 4 (the number of features), 171 /4 = 42.75%

B.3 Calculations for the phishy and botnet datasets

- Phishing occurrence = 20.14, and the total number is 11,055.
  20.14/11,055 = 0.0018
  0.0018 × 100 = 0.18%
- The average of botnet data with test analysis (Danmini and Ecobee).
  Ecobee occurrence = 98.23%
  Danmini occurrence = 94.8%
  Average = 98.23 + 94.8 = 193.03/2 = 96.515
  69.515/680.786 = 0.10
  0.10 × 100 = 10%
- The difference is 9.82% (10% – 0.18%), Figure 5.10

B.4 Calculations for the botnet and honeypot datasets

- The average of botnet data with test analysis (Danmini and Ecobee).
  Danmini occurrence = 79.84%
  Ecobee occurrence = 66.43%

Average = 79.84 + 66.43 = 146.27/2 = 73.135

73.135/ 680.786 = 0.10

0.10 × 100 = 10%

- Honeypot occurrence = 8.25, and the total number is 80.462.

  8.25/80.462 × 100 = 0.0102%

- The difference is 9.98% (10% – 0.0102%), Figure 5.11

B.5 The average of TCP and UDP in the honeypot and botnet datasets

- The average of TCP

  Botnet (Danmini) = 79.84%

  Botnet (Ecobee) = 66.43%

  Honeypot = 8.25%

  Average = (79.84 + 66.43 + 8.25) = 154.52

  154.52 /3 = 51.50%

- The average of UDP

  Botnet (Danmini) = 58.25%

  Botnet (Ecobee) = 38.53%

  Honeypot = 5.29%

  Average = (58.25 + 38.53 + 5.29) = 102.07

  102.07/3 = 34.02%

Figure 5.11 and Figure 5.12