APPLYING EMERGING DATA TECHNIQUES AND ADVANCED ANALYTICS TO COMBAT CYBER THREAT

A Thesis Presented to the Department of

Computer Science

African University of Science and Technology

In Partial Fulfilment of the Requirements for the Degree of

MASTER of Computer Science

By

Kohol Valentine Iornav

Abuja, Nigeria

December 2017.

CERTIFICATION

This is to certify that the thesis titled “APPLYING EMERGING DATA TECHNIQUES AND ADVANCED ANALYTICS TO COMBAT CYBER THREAT” submitted to the school of postgraduate studies, African University of Science and Technology (AUST) Abuja, Nigeria for the award of the Master's degree is a record of original research carried out by KOHOL VALENTINE IORNAV in the Department of Computer Science.

APPLYING EMERGING DATA TECHNIQUES AND ADVANCED ANALYTICS TO COMBAT CYBER THREAT

By

KOHOL VALENTINE IORNAV

A THESIS APPROVED BY THE COMPUTER SCIENCE DEPARTMENT

RECOMMENDED: _____

Supervisor, Prof Ekpe Okorafor

_____

Co-supervisor

_____

Head, Department of Computer Science

APPROVED: _____

Chief Academic Officer

_____

December 9th, 2017

ABSTRACT

Cyber threats are currently on the rise, which has caused individuals, industrial control systems (ICSs), critical infrastructures (CIs), and nations to be subjected to attacks with great losses. Among the cyber threats used for these attacks is the advanced persistent threat (APT) which tends to use highly sophisticated tools to attack targeted organizations or a nation's critical infrastructure. The capabilities of big data can be leveraged in conducting advanced analytics by gathering intelligence from potential security events and network activities to make timely reports and predictions of intrusions. In this work, big data technology is proposed; a Hadoop Ecosystem was integrated to a honeypot to collect massive data from network activities and attackers' behaviour for forensics. A decision tree classification algorithm was built in modelling a predictive model for network intrusion detection. An accuracy of 92.46% was recorded, showing its capability of giving low false positive alarm rates.

Keywords: Cyber threat, Cyberattacks, Big data, Honeypot, Hadoop Ecosystem, Predictive model for network intrusion detection

ACKNOWLEDGEMENT

I want to thank my supervisor, Prof Ekpe Okorafor for his immense support and guidance throughout this scholarly work. I also express my sincere appreciation to the Head of Department, Prof Amos David for his stellar leadership in the department along with all the faculty members spread across the globe that taught me during this program.

In indebtedness, I unreservedly thank African Development Bank (AfDB) and the entire management of the African University of Science and Technology (AUST), for sponsoring and awarding me the scholarship to obtain this quality education freely at this prestigious University. My gratitude also goes to the Pan African Materials Institute (PAMI) for their grant, which supported my education, and for also making me their scholar.

Unforgettably, I would love to thank my entire family for their unflinching support in all aspects of my life. Thank you all for your altruism. My appreciation will be incomplete without mentioning my host of friends, quintessential mates of AUST Class 2016/2017 and the entire membership of the AUST Catholic Family for their mutual collaboration, support and love.

I would love to thank all of you that I have come across in life for adding value to my life. Finally, I doff my hat in reverence to acknowledge all the scholars whose work I have used as a pedestal for accomplishing my research work.

DEDICATION

I dedicate this piece of work to God Almighty for all His countless blessings upon my life.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF APPENDICES

LIST OF ABBREVIATIONS AND ACRONYMS

ACD	Active Cyber Defence
APT	Advanced Persistent Threat
BDA	Big Data Analytics
C&C	Command and Control
CI	Critical Infrastructures
CIA	Confidentiality, Integrity and Availability
DDoS	Distributed Denial of Service
DNS	Domain Name System
DoS	Denial of Service
HDFS	Hadoop Distributed File System
HTTP	Hypertext Transfer Protocol
ICMP	Internet Control Message Protocol
ICS	Industrial Control Systems
IDS	Intrusion Detection System
IOC	Indicators of Compromise
IP	Internet Protocol
KDD	Knowledge Discovery in Databases
OS	Operating System
POSIX	Portable Operating System Interface
R2L	Remote to Local
U2R	User to Root
UDP	User Datagram Protocol

CHAPTER ONE
INTRODUCTION

1.1 Background

In recent years, society has become dependent on computers and computer networks. We are getting more connected with ubiquitous technologies than ever before (Koutský, 2014; Mohsen et al., 2017; RSA, n.d.). Consequently, securing the systems and networks on which we are dependent becomes increasingly important for individuals' safety, economic security, and national defence as this is now unavoidable (Tech Georgia, 2016).

With digitization, most critical infrastructures (CI) like the financial sector, energy supply, government services and healthcare depend on information technology networks for daily operations and activities. Society invariably depends on these infrastructures (Brasso, 2016). By definition, "critical infrastructure is a complex system of components that ensure transport, safety, health, communication, production, and any other activities necessary for a nation's needs" (Wang & Alexander, 2015).

Due to these dependencies, when critical infrastructures are disrupted with its concomitant downtime, it affects activities and the well-being of the users socially and economically. This may affect a nation entirely (European Commission, 2013; Grottke, Sun, Fricks & Trivedi, 2008). Attackers or adversaries carefully target critical infrastructures after exploring its vulnerabilities and infiltrating the control systems and are indeed ready to incur greater costs and time to gain expertise in order to accomplish their goals (Hosburgh, 2016; Virvilis, Serrano & Dandurand, 2014).

The terms cyberattack and cyber threat are sometimes used. Cyber threats have the capability of damaging and gaining unauthorised access to computers, computer networks and information systems (Gloag, n.d.). Wang & Alexander (2015) mentioned that “cyber threats include targeted attacks, malware, spam, system privilege abuse, classified information leakage, vulnerabilities exposed by poor maintenance, user indiscretions (unintentional information leaking), and web defacements (misinformation/discredit), etc.”

Statistics have shown that these infrastructures have been witnessing an alarming increase in the number of attacks (Mainone Cable, 2017; PandaLabs, 2016) and attack scenarios are also varying. PandaLabs (2016) reported that about 18 million new types of malware were recorded in the third quarter of 2016. Between April 2016 and March 2017, the number of ransomware victims increased by 11% compared to the previous twelve months (April 2015-March 2016). This means about 2,315,931 to 2,581,026 users around the world have fallen victim to these attacks (Kaspersky Lab, 2017).

Emerging threats are coming up yearly, among them are distributed denial of service (DDoS), advanced persistent threat (APT), ransomware, social engineering attacks while there has been a high increase in others, like adware, phishing attacks and Trojans (Boehmer, 2014; FBI, n.d.; Michael, 2017; US Government, n.d.). In fact, in 2014 Virvilis saw that the astuteness, complexity and number of cyber threats and cyberattacks have increased steadily over recent years.

Ciaran Martin warned that “cyber threats will continue to evolve, which is why the countries must work together at the pace to deliver hard outcomes and ground-breaking innovation to reduce the cyber threat to critical services and deter would-be attackers," (National Crime Agency (NCA), 2017).

This implies that there are no fixed ways of mitigating some of these attacks as the trend is constantly evolving in scale and sophistication (Ernst & Young, 2015; Javaid et al., 2016).

Steve Langan reported that in 2016 "cybercrime cost the global economy over $450 billion, over 2 billion personal records were stolen and in the U.S. alone over 100 million Americans had their medical records stolen" (Graham, 2017). A report has shown that the US Government alone will be investing over $19 billion for cyber defence and security in the 2017 fiscal year's budget (Steve, 2016).

The emergence of big data platform and machine learning techniques have provided a good move in knowledge discovery and data science that can be leveraged in tackling these cyber threats. "Big data analytics is defined as enabling organizations to discover previously unseen patterns and to develop actionable insights about their businesses and environments, including cyber defence. Cyber analytics applies big data tools and techniques to capture, process and refine network activity data, applies algorithms for near-real-time review of every network node and employs visualization tools to easily identify anomalous behaviour required for fast response or investigation" (Ponemon Institute, 2013).

Having seen an overview of the dangers of cyber threats, this thesis examines cyber threats and finds a novel solution to combat cyberattacks, leveraging emerging data techniques and advanced analytics.

1.2 Problem Statement

Critical infrastructures are currently experiencing a high rate of cyberattacks using highly sophisticated techniques.

A lot of effort has been employed in cyber defence to combat cyber threats yet, with the emergence of new threatscapes, such controls and traditional tools are circumvented by crafty attackers. Ernst & Young (2015) reported that the security advancement in the industry has not maintained the pace with today's diverse set of threat actors. This now leads to a research question: How can security professionals combat these cyber threats leveraging the capabilities of big data in analysing enormous and large data sets from disparate data sources (potential security events)?

This problem has spurred research which seeks to use emerging data techniques and advanced analytics to combat cyber threats (advanced persistent threat).

1.3 Research Objectives

1. The basic objective of this research work is to combat cyber threats which are used in well-orchestrated cyberattacks. This work will be achieved by the goals outlined below:

2. State-of-the-art research efforts conducted in the use of big data analytics and advanced prediction in investigating cyber threats for intrusion detection.

3. Use big data analytics to significantly enhance the detection capabilities of defenders, enabling them to detect APT activities that are passing under the radar of traditional security solutions.

4. Leverage big data methods and explore new detection algorithms capable of processing significant amounts of data from diverse data sources.

5. Generate a predictive analytical model for the prediction of cyberattacks.

1.4 Technologies for Implementation

1. In order to achieve this work, the following technologies or platforms were used:

2. Honeypot: used for active defence by deception, detection and network forensics.

3. Hadoop: used for dynamic data collection, consolidation and correlation of data from any number of diverse data sources, such as network traffic and event data (e.g., network devices, IDS).

4. Predictive analytics tools: machine learning algorithms for classification and prediction of network attacks.

5. Apache Spark: used to analyse streaming data for the classification (prediction) of intrusions.

1.5 Scope of the Work

The scope of the research work includes the following:

1. To integrate a honeypot system into Hadoop to capture data for advanced analytics, threat intelligence and forensics.

2. To develop an intrusion detection system that will predict and classify network intrusions and attacks with emphasis on the advanced persistent threat.

3. To explore existing technologies, tools and use KDD Cup 99 Datasets for training a supervised machine learning model.

1.6 Document Road Map

This master's thesis report is organised into chapters. Chapter One is the introduction of the thesis report. Chapter Two is the review of related literature detailing the underlying and fundamental concepts of the research as well as the state-of-the-art advances made by different authors.

Chapter Three is the proposed methodology for the implementation, while Chapter Four is the implementation, experimentation/simulation and analysis of the results of the system. And finally, Chapter Five is the conclusion and future work discussion.

CHAPTER TWO
LITERATURE REVIEW

2.1 Cyber threats

According to the National Institute of Standards and Technology (NIST) (as cited in Johnson, Badger, Waltermire, Snyder & Skorupka, 2016), a cyber threat is “any circumstance or event with the potential to adversely impact organizational operations (including mission, functions, image, or reputation), organizational assets, individuals, other organizations, or the Nation through an information system via unauthorized access, destruction, disclosure, or modification of information, and/or denial of service”. Wert (n.d.) also defined cyber threat as “any malicious act that attempts to gain access to a computer network without authorization or permission from the owners”. Cyber threats are the capabilities leveraged by adversaries by exploiting the vulnerabilities of an infrastructure or network of a victim. The motives behind cyberattacks vary. Attacks may be for power, fame, skills for employment, entertainment, control, exploitation (hacktivism), revenge, financial gain (ransomware) and espionage. Additionally, cyber threats may have other goals like damaging the reputation and image of a company or a person by information leaks, stealing product designs and patents for the unauthorised use of the brand, and influencing political and governmental outcomes and events (ElevenPaths, 2017; Totah, 2016).

2.1.1 Cyber threat Terminologies

2.1.1.1 Cyberattack

A cyberattack is an attack, via cyberspace, targeting an enterprise’s use of cyberspace for the purpose of disrupting, disabling, destroying, or maliciously controlling a computing environment/infrastructure; or destroying the integrity of the data or stealing controlled information.

2.1.1.2 Vulnerability

Vulnerability is a susceptibility or weakness of software, hardware, internal controls, system security procedures or online service that can be exploited or triggered by an adversary or attacker/threat. An exploitation of vulnerabilities results in a disruption of the confidentiality, integrity, or availability (CIA) of the ICT system or related information assets, which may cause a breach of data privacy, interruption of operation of mission-critical systems, and so on (ISO, 2014; NIST, 2013).

2.1.1.3 Attack Vector

An attack vector is a means or route by which a hacker can gain access to a computer or network server in order to deliver a payload or malicious outcome in a nefarious manner. Attack vectors help hackers or adversaries gain access and exploit system or network vulnerabilities, including the human element (Rouse, 2012).

2.1.1.4 Adversary

An adversary is an individual, group, organization, or government that conducts or has the intention to conduct detrimental activities (CNSS, 2015; U.S. DHS, 2008).

2.1.1.5 Indicators of Compromise (IOC)

Indicators of Compromise (IOC) are pieces of an artefact or forensic data observed on a network, such as data found in system log entries or files, remnants of intrusions that identify potentially malicious activity on a system or network. IOCs help security professionals in detecting and combating malware infections, data breaches and other threat activities (Gragido, 2012; Lord, 2017; Rouse, 2015).

2.1.2 Cyber Threat Categories

There are numerous cyber threats ranging from malicious software (malware), social engineering attacks, command and control (C&C), advanced persistent threats (APTs), local to host attacks, denial of service attacks (DoS), distributed denial of service (DDoS), worms, Trojans, viruses, spyware, spoofing, botnets, buffer overflows, SQL injections and others. In this section, some of these cyber threats will be discussed.

2.1.2.1 Malware Threats

Malware is short for *malicious software*. Malware is a broad term that encompasses a whole lot of hostile and intrusive software. This software includes computer viruses, worms, Trojan horses, ransomware, spyware, adware, scareware, and other malicious programs.

Dooley and Rooney (2017) observe that “malware has grown to become a menacing force in enterprise networks. In earlier days, malware consisted of malicious software that stealthily installed itself on a device to perform a pre-programmed form of attack. Unfortunately, this static form of malware is a rarity today, and malware is growing increasingly more sophisticated so as to hide itself on host systems, operate stealthily to avoid detection and remediation, and contact external command and control (C&C) centres for new software and instructions”. This malware, when exploited by the black-hat attacker, manipulates the host machine into a botnet and replicates itself among the other machines forming a botnet which can be used by the attacker for achieving his/her goals.

2.1.2.2 Social Engineering Cyber Threats

Social engineering is a proliferation technique through which threat vectors are exploited.

Andress & Winterfeld (2011) see social engineering (SE) as “the act of influencing someone’s behaviour through manipulating their emotions, or gaining and betraying their trust to gain access to their system. This can be done in person, over the phone, via an email, through social media, or a variety of other methods”. SE is the art of manipulating people using a ruse with bait to trick them into performing actions or divulging confidential information. This technique is different from other attacks because here humans are used as the threat vector to gain valid confidential credentials to carry out an attack, rather than breaking in or using technical techniques. SE includes but is not limited to phishing, spear phishing, vishing, email hacking or spamming, baiting, pretexting, pharming and others.

2.1.2.3 Denial of Service Attacks (DoS)

A major threat, *denial of service* (DoS) or its variant *distributed DoS* (DDoS) attacks, is prevalent today and has been plaguing Internet-based resources for years. It is an intentional attempt to disrupt or degrade the availability of network resources to legitimate users requesting it (Kadir, 2013). Practically, DoS attacks are usually DDoS where host computers are attacked and seized as zombies making a botnet to remotely command and control, to massively attack web servers, mail servers, FTP servers, and other public-facing components of a company or an organizational infrastructure (Andress & Winterfeld, 2011).

Figure 2.1 overleaf shows a view of a Distributed Denial of Service (DDoS).

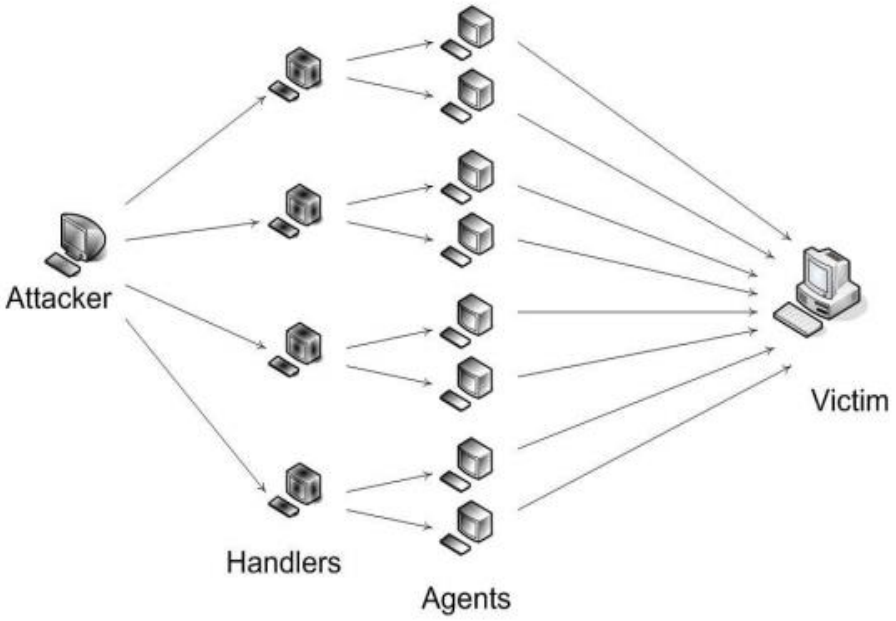


Figure 2.1: A view of a Distributed Denial of Service (DDoS)

2.1.2.4 Advanced Persistent Threat (APT)

An advanced persistent threat (APT) is a targeted, long-term and organized attack against a high-value asset or a physical system. It is specially designed to infiltrate and stealthily gain access to information. This kind of attack takes a more active role in gathering information than any other known attack and operates in a low-and-slow mode making such intrusion unnoticeable to the victim (Andress & Winterfeld, 2011; Wang & Jones, 2017). NIST (as cited in Seculert, 2014) defines an APT as a type of computer attack that meets three criteria:

- It pursues its objectives repeatedly over an extended period.
- It adapts to defenders' efforts to resist it.

- It is determined to maintain the level of interaction needed to execute its objectives.

The following are the unique characteristics that differentiate APTs from other traditional cyber threats or -attacks (Virvilis et al., 2014):

- APTs make frequent use of zero-day exploits or modify/obfuscate known ones and thus, are able to evade the majority of signature-based endpoints and network intrusion detection solutions. In addition, the attacks are generally spread over a wide period and, as a result, are often outside the limited detection/correlation window of these systems.
- Attackers focus on a specific target and are willing to spend a significant amount of time and explore all possible attack paths until they manage to subvert its defences.
- Based on the analysis of the major APT attacks it is evident that some perpetrators are supported by nation-states that have significant enabling capabilities (intelligence collection, manufacturing, covert physical access) for cyberattacks.
- APTs are highly selective. Only a small and carefully selected number of victims are targeted, usually in nontechnical departments of an organization as they are less likely to identify and report an attack.

2.2 The Stages/Phases of an Advanced Attack

Seculert (2014) established seven stages that may be involved in any APT attack namely; (1) Reconnaissance, (2) Weaponization, (3) Delivery, (4) Exploitation, (5) Installation, (6) Command and Control (C&C), and (7) Action on Objectives.

2.2.1 Reconnaissance Phase

Reconnaissance is like information engineering, it is a stage whereby the attacker tries to research, identify and profile victims and their entire network structure/topology. They figure out the defensive mechanism used by the victims' establishment, the network vulnerabilities and their emails addresses and instant messaging links or handles (Irwin & Northcutt, 2014; Seculert, 2014).

2.2.2 Weaponization Phase

In this stage, the attacker develops malware suited for infiltration and reconnaissance. "Lockheed Martin reports that client application data files such as Adobe Portable Document Format (PDF) or Microsoft Office documents increasingly serve as weaponized deliverables. Threat actors with deliberate intent to harm may use a combination of compromised JavaScript, PDF files, or Microsoft Office files that are attached to a Phishing email and sent to a targeted user or group of users in the organization (Hutchens et. al., 2011)" (Irwin & Northcutt, 2014).

2.2.3 Delivery Phase

This is the stage of the attack where the attacker delivers the malware into the targeted network using an APT campaign, typically via spear phishing email, private messages on social media, downloads off the Web, fake login pages, spoofing, compromised USB drives, etc. Malware may be downloaded through these campaigns which run on the victim's machine, infecting it and granting access into the target's environment or sending authentication credentials to the attacker for exploitation (Irwin & Northcutt, 2014; Seculert, 2014).

2.2.4 Exploitation Phase

The delivered malware finds a vulnerable system (operating system or applications) and exploits that vulnerability to execute code on the victim system.

“This may allow the attacker to execute code, such as command and control code, which will enable the malware to connect to the attacker’s command and control servers and download more code” (Hutchens et. al, as cited in Irwin & Northcutt, 2014).

2.2.5 Installation Phase

For malware to successfully exploit a system, it must be installed first. This phase deals with the installation of the malware codes with most at times remains inactive for a while to evade abrupt detection on the compromised system (Irwin & Northcutt, 2014; Seculert, 2014).

2.2.6 Command and Control (C&C) Phase

Once the threat actor has successfully installed the required malicious code, the weaponized malware will usually attempt to stealthily establish communication back to the threat actor’s command and control master server for remote control of compromised system i.e. the host or network. After a successful communication session with the control server, the threat actor can send commands remotely to further compromise and control the infected host and network (Irwin & Northcutt, 2014; Seculert, 2014).

2.2.7 Action on Objectives Phase

This last phase of APTs is the final goal of the attacker. The threat actor/attacker uses the valid authentication credentials stolen to further gain access to other systems involved and escalates the privileges. Finally, the black-hat intrudes and exfiltrates the

desired data or information all at once or piecemeal over time and installs backdoors to remain undetected and even to silently control the entire system (Irwin & Northcutt, 2014; Seculert, 2014).

2.3 Emerging Data Techniques and Advanced Analytics

Emerging data techniques include artificial intelligence (AI), machine learning (ML) and big data techniques used for mining massive data. Advanced analytics techniques include text analytics, predictive analytics (which build models for forecasting customer behaviour and other future developments), statistics, machine learning (which tap algorithms to analyse large datasets), data mining (which sift through data sets in search of patterns and relationships), natural language processing, deep learning, advanced visualisation and so on (IBM, n.d.-b; Rouse, 2017; Russom, 2011).

These techniques help with storage, processing and analysis of data for useful knowledge and insights that were previously untapped by traditional methods and business insights (BI). Most of this data in the discussion is fast-moving and diverse data, unstructured in nature (Ali et al., 2016).

2.3.1 Big Data as Emerging Data Technique

Beyer & Laney (2012) at Gartner defined big data thus: “Big data is high-volume, high-velocity and/or high-variety information assets that demand cost-effective, innovative forms of information processing that enable enhanced insight, decision making, and process automation”. There has been confusion about what big data really is, however, Gartner’s definition is the most widely accepted definition. This definition clears the misconception of seeing big data as massive volume, unstructured, semi-structured and structured data. Big data is generated most often in real time and on an extremely large scale from sensors, devices, video/audio, networks, log files, transactional

applications, web, and social media (IBM, n.d.-b). Jain (2016) of IBM asserts that rather than having only three distinct dimensions (volume, velocity and variety) as others like DXC Technology (2015) see it, big data has instead five. He called it the “5 Vs of data” which include: (1) Volume; (2) Velocity; (3) Variety; (4) Variability and; (5) Value. However, DeVan (2016) defined big data with 7 Vs: Volume, Velocity, Variety, Variability, Veracity, Visualization, and Value which gives a complete sense of what exactly big data is.

Hashem et al (as cited in Oseku-Afful, 2016) categorised big data into five different aspects viz.: data sources, content format, data stores, data staging and data processing

2.3.1.1 Volume as a Dimension of Big Data

This generally describes how much data we obtain (the scale of data) from datasets with sizes varying from gigabytes (GB) to terabytes (TB) to zettabyte (ZB) and now to yottabytes (YB). It is seen as the capability of processing the massive or vast amount of data (DeVan, 2016). “This also refers to the proliferation of data that is produced, from social media to transactional records, from system logs to the Internet of Things. The greater the volume, the more data points are available to enable systems to identify patterns and predict future outcomes based on historical record” (DXC Technology, 2015).

2.3.1.2 Velocity as a Dimension of Big Data

The velocity of big data has to do with the throughput of data and the latency of data storage and reporting. “Velocity of big” data refers to the analysis of data in motion (streaming data). It is also the speed of data processing, taking into consideration the arrival of the data collected in the database and its mining time (IBM, n.d.-a).

2.3.1.3 Variety as a Dimension of Big Data

Variety refers to the many sources (domains of origin of data) and types of data (data types) both structured, semi-structured and unstructured.

2.3.1.4 Variability as a Dimension of Big Data

"Variability refers to changes in the data flow's velocity, which for cost-effectiveness leads to the automated spawning of additional processors in cloud systems to handle the load as it increases, and release the resources as the load diminishes" (Janssen & Grady, 2013).

2.3.1.5 Veracity as a Dimension of Big Data

Veracity refers to the trustworthiness of the data. Normandeau (2013) supported this by saying that veracity is concerned with the removal of noise, biases and abnormality in the data. This calls for tools for curating data cleanly to avoid accumulating badly in the system, in order to provide reliable and trustworthy insights after mining.

2.3.1.6 Visualization as a Dimension of Big Data

After collection of data, the data needs to be presented in a clearly readable and accessible manner. This is when visualization comes in, be it for business insights (BI) or decision making. Big data requires visualizations of complex variables in dozens, which is beyond the mere x-axis and y-axis charts and spreadsheets. Implementing this feature of big data needs trained skills in order to make visualizations meaningful as required, using the complex relationship between variables and big data's velocity and variety. It also needs high-performance computing resources (Firican, 2017).

2.3.1.7 Value as a Dimension of Big Data

This dimension or characteristic of big data is the last of the “Vs” discussed above. Just collecting data is not enough. Lying data in databases is virtually worthless if no rigorous and accurate analysis is carried out on it to provide information and gainful insights.

2.3.2 Big Data Analytics (BDA)

According to IBM (n.d.-b) “big data analytics is the use of advanced analytic techniques against very large, diverse datasets that include different types, such as structured/unstructured and streaming/batch, and different sizes from terabytes to zettabytes”. This definition is further supported by one given by Ambreen, Nadeem & Dubey (2016). They described big data analysis as “a technology that searches useful information such as a relation rule, were hidden value from huge data. Big data analysis uses various existing analysis techniques, data analysis and etc.”. Big data analytics help organisations harness their data and extract opportunities and value from it. Big data platform brings high-performance analytics to users with speed and efficiency that cannot be obtained from the traditional analytics.

2.3.3 Machine Learning (ML)

Machine learning is a sub-category of artificial intelligence (AI). Machine learning capabilities are limitless. It is a field of study in which computers use algorithms to study by analysing several examples of particular problems in order to perform a specific task. Gronlund (2017) at Azure said that “machine learning is a data science technique that allows computers to use existing data to forecast future behaviours, outcomes, and trends. Using machine learning, computers learn without being explicitly programmed”. Machine learning is basically divided into two: supervised and unsupervised.

Supervised learning occurs when an agent is given a set of training examples made up of input-output pairs (labelled datasets), learns a function that maps from input to output and then predicts the output of a new input.

Whereas in unsupervised machine learning an agent is fed unlabelled datasets and it then finds a model with the structure and distribution to learn more about the data itself (Poole & Mackworth, 2010; Russell & Norvig, 1996).

2.4 Combating Cyber threats

In this section, methods of militating against insider or external cyber threats shall be discussed briefly.

2.4.1 Types of Cyber Defence

Wang & Alexander (2015) classified cyber defence into two methods. There is *active cyber defence* (ACD) and *passive cyber defence* (PCD).

2.4.1.1 Passive Cyber Defence

Passive cyber defence is countermeasures or systems that are used to mitigate threats by using reliable defence or giving insights without consistent human interaction (Lee, 2015). The methods used in passive defence include firewalls, virus detection (anti-virus software), threat detection technologies, or patches. Wang & Alexander (2015) outlined a four-step model of the passive defence as (1) locate invading code; (2) unplug affected systems; (3) thwart particular attacks using security patches and solutions, and (4) use the patches and solutions system-wide.

However, in as much as they do not need constant interaction from the personnel of the security team, this approach fails when sophisticated attacks are carried out by determined and well-resourced adversaries. This calls for the emergence of an

alternative strategy, namely active cyber defence or proactive defence (Lendvay, 2016).

2.4.1.2 Active Cyber Defence (ACD)

Lee (2015) defined Active Cyber Defence (ACD) as, “the process of analysts monitoring for, responding to, learning from, and applying their knowledge to threats internal to the network”. This definition is also supported by Dewar (2014), he asserted that ACD is “a method of achieving cyber security predicated upon the deployment of measures to detect, analyse, identify and mitigate threats to and from cyberspace in real-time, combined with the capability and resources to take proactive or aggressive action against threat agents in those agents’ home networks”. According to Lendvay (2016), and Wang & Alexander (2015), ACD is categorised into three, viz: (1) detection and forensics; (2) deception, and (3) attack termination. Highly trained security personnel are needed for analysis in the cybersecurity team. This analysis helps in identifying the attacker, deterring future attacks and offensive knock off-line of infiltrators.

Active cyber defence uses honeypots for detection and forensics, as these are used to lure adversaries and get them identified along with their behavioural pattern. In some cases, denial of service attack is launched to counter and terminate attackers when infiltrations are witnessed.

2.4.2 Honeypots/Honeynets

A honeypot is a computer system configured as a decoy (server-bait) to lure an adversary into attacking it. Honeypots help the security team/officer get a better knowledge of a black-hat’s technique and tools used in an intrusion. vArmour (2016) defined a honeypot as “a real or synthesized computer system, application, or service

that appears to be legitimate, but is in fact only utilized for luring, identifying, and analysing adversaries. A honeypot may be a stand-alone system, an individual service on a workload, or a local software agent to name a few possibilities".

A collection of several honeypots that intercommunicate is termed a honeynet. Honeypots are commonly classified into two categories, viz.: by purpose and by the level of interaction.

Going by classification by purpose, Danani & Jani (2012) identify two categories of honeypots: production honeypots and research honeypots. The production honeypots are a tool used by organisations having real production networks while the research honeypots are those used for getting as much information as possible about the black-hats. The research honeypots are not intended to secure production networks directly but rather for studies of the black-hat community (Spitzner, 2002). On the level of interaction and involvement with adversaries, there are low-, medium- and high-interaction honeypots.

Low-interaction honeypots, according to Spitzner (2002), "are the easiest to install, configure, deploy, and maintain because of their simple design and basic functionality". Low-interaction allows a minimum level of interaction with the malware or attacker, hence it lacks the capacity to lure sophisticated attackers for a long time since all services are predesigned. However, they can detect the initial probe or intrusion of the attacker. They are a simple emulation of vulnerable services. It has the advantage of potentially trapping an attacker without revealing the system's information, like its operating system functionality, to the attacker.

With medium-interaction honeypots, on the other hand, attackers are provided with more interaction with the honeypot compared to a low-interaction honeypot, however, the functionalities are fewer than high-interaction honeypots. High-interaction honeypots are the recommended technology for novel attacks. Here actual vulnerable services or software are deployed on virtual or physical devices.

High-interaction honeypots are difficult to set up but interestingly have the potential to capture zero-day attacks. In addition, they collect a massive amount of information about the attacker's techniques without allowing such adversary to unmask it. However, high-interaction honeypots may be risky since they grant access to live systems (Diebold, Hess & Schäfer, 2005; Veysset & Butti, 2006).

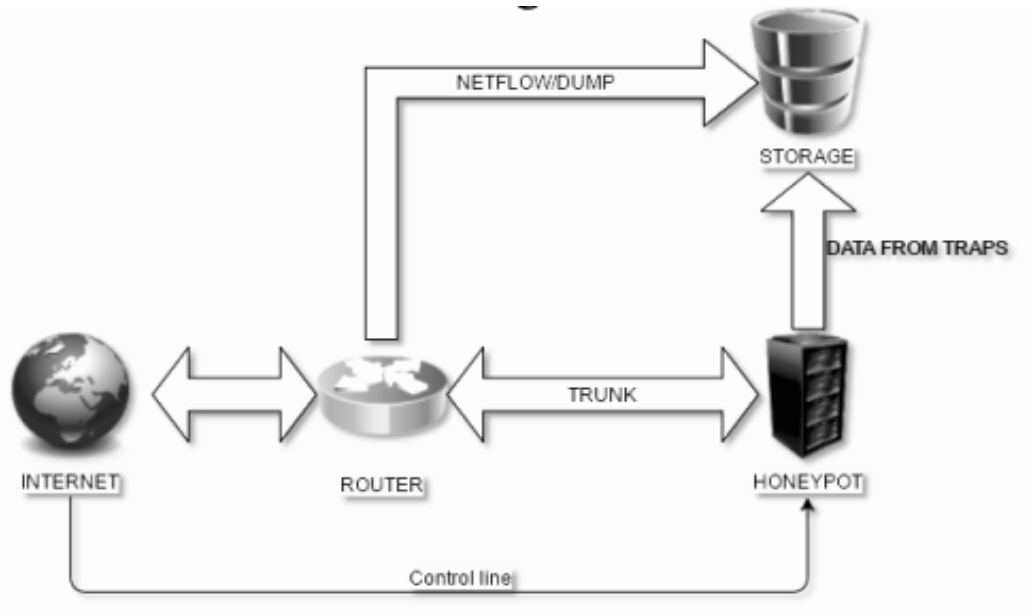


Figure 2.2: The Common Structure of Honeypot-system
(Source: Egupov, Zareshin, Yadikin & Silnov, 2017)

2.4.3 Intrusion Detection Systems

To understand what intrusion detection means, let us look at the definition given by Esposito, Mazzariello, Oliviero, Romano & Sansone (2006). They defined intrusion detection as “the art of detecting inappropriate, incorrect or anomalous activity within a system, be it a single host or a whole network”.

2.4.4 *Classifications of Intrusion Detection Systems (IDS)*

There are two types of intrusion detection systems (IDS) which are classified according to their approach. The first is based on the place where an IDS is being placed and the second is based on analysis of the intrusion technique employed.

In the first classification, Wang & Jones (2017) categorised intrusion detection systems (IDSs) into three types:

i. Network-based intrusion detection systems (NIDS).

ii. Host-based intrusion detection systems (HIDS).

iii. Hybrid-based intrusion detection systems (hybrid IDS).

The second classification of IDS, based on the intrusion approach, Javaid et al. (2016) asserted that we have:

i. Signature (misuse) based NIDS (SNIDS); and

ii. Anomaly detection based NIDS (ADNIDS).

2.4.4.1 Network-based Intrusion Detection System (NIDS)

“A NIDS is an independent platform that identifies intrusions by examining network traffic and monitoring multiple hosts. NIDSs gain access to network traffic by connecting to a network hub, network switch configured for port mirroring, or network

tap" (Beigh & Peer, 2012). A NIDS is designed and installed with sensors to receive all packets on a particular network segment in order to detect malicious traffic. They operate in a promiscuous mode with a network interface card (NIC) along with a management interface. Figure 2.3 shows a Network-Based Intrusion Detection System.

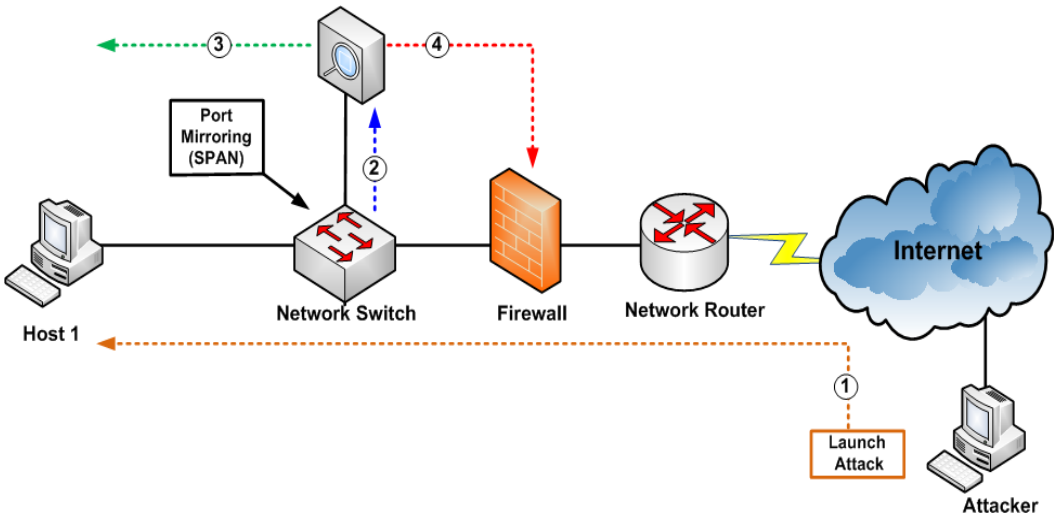


Figure 2.3: Network-Based Intrusion Detection System
(Source: www.hackthis.co.uk).

2.4.4.2 Host-based Intrusion Detection System (HIDS)

Host-based intrusion detection is a software application (agent) installed on a host. HIDS monitors only the individual workstation it has been mounted on and not an entire network. These host-based applications run in the background with the operating system and function by analysing log files and generating alarms. These log files

include “system calls, application logs, file-system modifications (binaries, password files, capability databases, access control lists, etc.) and other host activities and state” (Beigh & Peer, 2012; Esposito et al., 2006). Figure 2.4 gives a diagrammatic representation of this system.

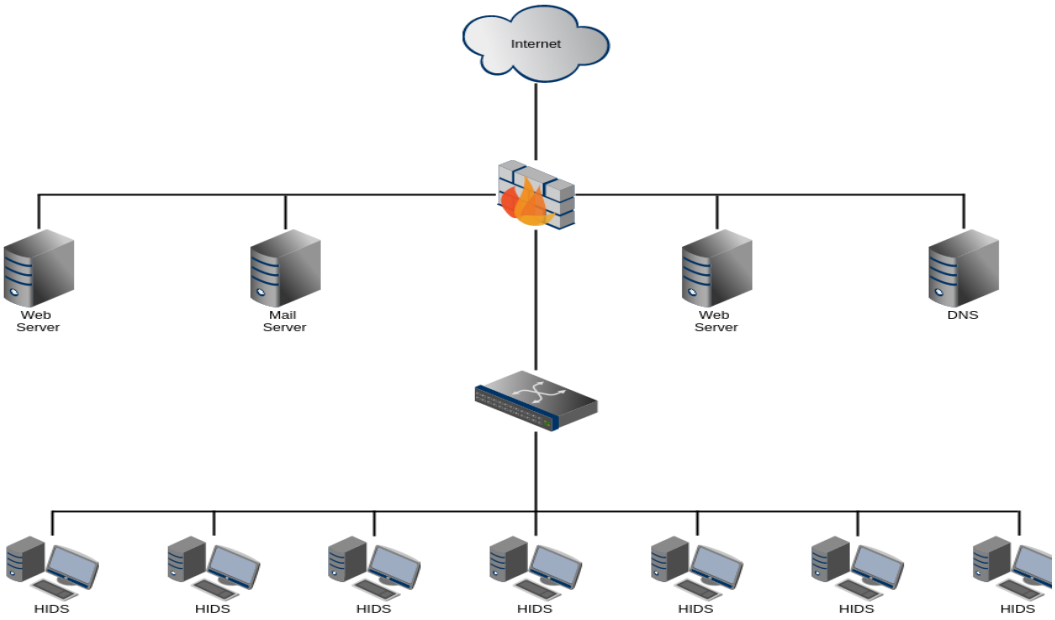


Figure 2.4: A Host-Based Intrusion Detection System
(Source: https://commons.wikimedia.org)

2.4.4.3 Hybrid-based Intrusion Detection System (Hybrid IDS)

This class of IDS is a combination of NIDS and HIDS and has been discussed by many authors. Other scholarly works advocate the use of both misuse and anomaly-based IDS for attack prevention and thus point to this as an hybrid-based IDS (Aydın, Zaim & Ceylan, 2009; Rizvi, Labrador, Guyan & Savan, 2016).

Altogether the combination of these approaches discussed herein helps secure information systems better than using just one method. This is because threat actors who are determined to attack one's network are always ready to adapt to various methodologies that can crack the defence of their victim.

2.4.4.4 Signature (misuse) Based NIDS (SNIDS)

This is the most common technique in use for abnormal network traffic detection. This approach focuses on the identification of known bad patterns and relies on a set of rules to discover attacks in network traffic. As with every system that uses a blacklist approach, it is vulnerable to attacks from sources with an unknown signature, such as zero-day exploits or use of encoding, packing or obfuscation techniques (Virvilis et al., 2014).

2.4.4.5 Anomaly Detection Based NIDS

This detection strategy consists of monitoring system activity to determine whether an observed activity is normal or anomalous, using a heuristic or statistical analysis to detect unknown attacks. It has the primary advantage of its intrinsic capability to detect novel attacks, unlike the signature-based approach. However, despite significant research efforts, such techniques still suffer from a high number of false positives. Some malicious communication through channels like a Secure Sockets Layer/Transport Layer Security (SSL/TLS) connection may become evasive as IDS analysis may classify these channels as having legitimate traffic (Esposito et al., 2006; Virvilis et al., 2014).

2.5 Review of State of the Art

In recent years, the world has been witnessing a lot of cyberattacks on countries, industrial control systems, cyber-physical infrastructures and lots more due to the

sophistication of advanced persistent threats (APTs). These APTs usually “elude traditional/conventional legacy defences such as firewalls, IDS/IPS, web filters, breach detection systems, etc.” (Seculert, 2014). There have been copious efforts by researchers and security firms/teams that have contributed greatly to the protection of networks from malicious adversaries using APTs.

In this section, I shall review the state of the art with regard to leveraging emerging data techniques and advanced prediction in combating cyber threats.

Cyberattacks are increasing because security systems are unable to detect them. Several authors have outlined how emerging data techniques can be of great help in combating these attacks. Virvilis, Serrano & Dandurand (2014) worked on how big data analytics (BDA) could be used for sophisticated cyberattacks. In their research, they surveyed and highlighted the importance of BDA for combating sophisticated attacks (APT). They pointed out two factors that make an APT so prolonged. They mentioned that firstly, after an initial foothold has been gained, a significant amount of time is required in order for the attackers to try to explore the network and its subnets, and locate the information of interest and get it exfiltrated as stealthily as possible to avoid detection. Secondly, attackers usually wish to maintain their access and continue to exfiltrate data in the future. They opined that the correlation of events across large timescales and from multiple sources, like analysis of network traffic, event logs and operating system/application artefacts, is crucial for the detection of sophisticated attacks. BDA helps in tackling these aforementioned problems through anomaly detection, based on the correlation of recent and historical events; and supports dynamic and managed collection, consolidation and correlation of data from any number of diverse data sources.

Wang and Jones (2017) further support the importance of BDA in the work they titled “Big Data Analytics for Network Intrusion Detection: A Survey”. They also affirmed that the use of data mining, machine learning and BDA can correlate multiple information sources into a coherent view, identify anomalies and suspicious activities, and finally achieve effective and efficient intrusion detection.

In their work, they investigated the state-of-the-art methods and techniques in network intrusion detection, and the advances and challenges of big data analytics in intrusion detection in order to explore new techniques that aid in intrusion detection analysis. In their review, principal component analysis (PCA) can be used for feature selection in machine learning for effective network intrusion detection. Methods of clustering and classification are leveraged to obtain valuable information of network intrusion through analysing the network data. They came up with the following ways BDA facilitates the identification of APTs in support of Virvilis et al. (2014):

- Anomaly detection based on the correlation between historical and recent events. For example, an increased volume of domain name system (DNS) traffic from a system in a short period can be due to legitimate users’ behaviours. But such a pattern indicates covert data exfiltration if it is also detected in historical traffic over a period of days. Furthermore, this kind of correlation helps reduce the false positive rate (FPR) of alerts. Big data analytics increases the scope and quantity of data that can compute correlation.
- Managed and dynamic capture, integration and correlation of data from heterogeneous data sources like network traffic, event data (e.g., IDS, network devices), and operating system artefacts help defenders correlate sporadic low-severity events as the result of ongoing intrusion or attack behaviour.

> Compared with SIEM systems, big data or big data analytics does not have a limited-time window within which the correlation is performed.

Cloud Security Alliance (2013) also advocated the use of BDA but stated that, despite the overwhelming benefits of BDA in combating APTs, there are a number of challenges that must be overcome to realize its true potential.

They pointed out data provenance (i.e. authenticity and integrity of data used for analytics), security of big data stores, human-interaction (the ability of humans to interpret the analysed data accurately) and privacy as some of the challenges to be addressed.

Since attacks have been evading current solutions by using an impressive sophisticated arsenal, Virvilis-Kollitiris (2015) advocated the redesign of defence systems and technology so that the focus is more on the detection of APTs than on prevention. In his work, he surveyed the current APTs by analysing the most common techniques, tools and attack paths that attackers are using, and highlighted the shortcomings of current security solutions. He developed a novel APT detection model, implemented and evaluated it with a deception technique (honeypot) for attack detection. His results indicated a high level of efficacy in the attack detection model with a very low false positive rate.

Similarly, Rizvi, et al.'s (2016) study proposed an improvement in the framework to current Host IDPS/Network using signature and anomaly-based methodologies by implementing a hybrid VMM-based Honeypot into a theorized self-healing hybrid IDPS to further boost their advantages in efficiency and accuracy.

Having reviewed the existing literature, this research intends to incorporate the existing detection algorithms with a honeypot system on a big data platform and use machine learning and data mining techniques to model a prediction model for the novel advanced attack.

CHAPTER THREE
RESEARCH DESIGN AND IMPLEMENTATION

This chapter presents the research design and approach to the implementation and achievement of the stated aims and objectives in Chapter One. Basically, this section discusses the big data platform, advanced analytics (machine learning) and the deployment of a honeypot. Experimental and simulation approaches were used as the research design.

3.1 The Big Data Platform

The big data platform is leverage in this research due to the fact that in production networks and critical infrastructure, a large amount of data is generated accumulatively from network events. The Hadoop software was chosen for the implementation of the proposed work. This is because of the ability of Hadoop to reliably store data in dependable distributed storage using commodity hardware with high-speed streaming analysis of a big data dataset using the MapReduce programming model. More so, the Apache Foundation software used in this research is freely open and available for use as an open source project.

Leveraging the big data platform, APTs can be tackled by getting every bit of the behavioural analysis of users and attackers on a network, monitoring their actions and patterns to gain useful insight using anomaly detection and prediction. Unlike the traditional or conventional approaches that use relational databases to collect records with predefined fields or labels, big data houses all data coming into the system and the data is curated to extract useful insights. This enables security professionals to trap the black-hat guys by correlating all successive moves which take a prolonged time to stealthily subvert the beefed-up system security.

Figure 3.1 gives an overview of the Hadoop Ecosystem.

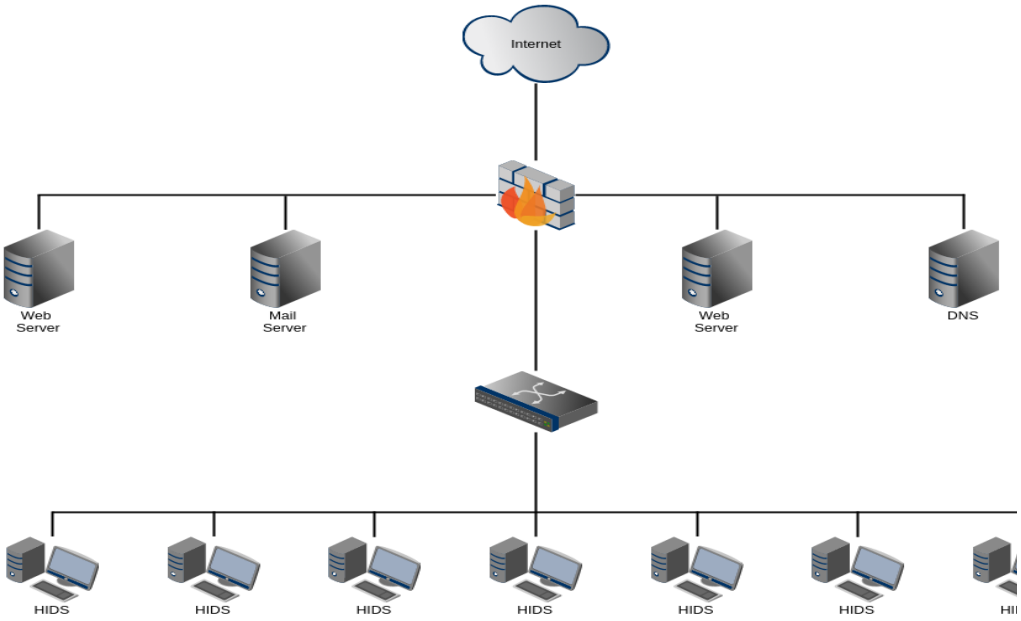


Figure 3.1: An overview of the Hadoop Ecosystem
(Source: Adapted from Edureka, 2016)

Some of these frameworks in the Hadoop Ecosystem used in this research will be discussed in the section below.

3.1.1 The Apache Hadoop

The Apache Hadoop is open source software for reliable, scalable and distributed computing in big data. It is used to reliably manage large volumes of structured, semi-structured and unstructured data. “The Apache Hadoop software library is a framework that allows for the distributed processing of large data sets across clusters of computers using simple programming models.

It is designed to scale up from single servers to thousands of machines, each offering local computation and storage. Rather than rely on hardware to deliver high-availability, the library itself is designed to detect and handle failures at the application layer, so delivering a highly-available service on top of a cluster of computers, each of which may be prone to failures" (The Foundation Apache Software, 2014).

The Hadoop core comprises the following components/modules:

i. Hadoop Common: This was formerly known as Hadoop core. Hadoop common contains common utilities and libraries that support the other Hadoop modules.

ii. Hadoop Distributed File System (HDFS): A distributed file system that provides high-throughput access to application data on commodity machines.

iii. Hadoop YARN: A framework for users' applications/job scheduling and cluster resource management.

iv. Hadoop MapReduce: A YARN-based system for parallel processing of large datasets.

The basic two key aspects of Hadoop; Hadoop Distributed File System (HDFS) and MapReduce which both run on the same systems, is briefly discussed in the section below.

3.1.1.1 The Hadoop Distributed File System (HDFS)

The HDFS was inspired by a project carried out by Google Technologies on the Google File System (GFS). GFS was carried out by the Google Research and Development team in 2003. The Hadoop Distributed File System (HDFS) is a distributed file system designed to run on commodity hardware. It has many similarities with existing distributed file systems. There are now several organisations offering similar services on the cloud platform.

However, the differences from other distributed file systems are significant. HDFS is highly fault-tolerant and is designed to be deployed on low-cost hardware. HDFS provides high throughput access to application data and is suitable for applications that have large data sets. HDFS relaxes a few POSIX requirements to enable streaming access to file system data. HDFS uses POSIX-like commands like cat, chgrp, -R, chmod, cp, du, ls, mkdir, mv, rm, stat, tail, chown, etc.

HDFS works on a master/slave architecture, having the NameNode as the master and the DataNodes as the slaves. HDFS persists metadata and application data separately. It has a single NameNode and many DataNodes (but one DataNode in a single node cluster). The NameNode is the master server that manages the file system namespace, stores metadata and regulates access to files by clients and maps data blocks to the DataNode. “HDFS metadata represents the structure of HDFS directories and files in a tree. It also includes the various attributes of directories and files, such as ownership, permissions, quotas, and replication factor” (Nauroth, 2014).

The DataNode stores the application data. It manages the data stored in each node and periodically reports its status to the master node (NameNode). Each file is split into blocks and each block is, in turn, replicated redundantly across nodes in the local rack. Figure 3.2 gives an overview of the Hadoop Distributed File System Architecture

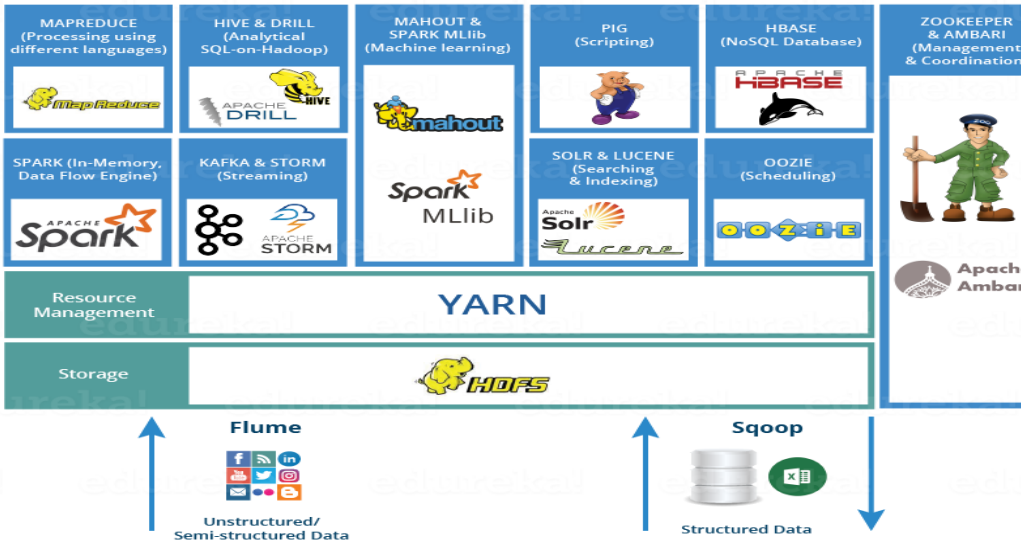


Figure 3.2: An overview of the Hadoop Distributed File System (HDFS) Architecture
(Source: Apache Foundation, 2013)

3.1.1.2 MapReduce

MapReduce is the second core component of the Hadoop common which is built based on Java. The Apache Software Foundation (2017) explains that a “Hadoop MapReduce is a software framework for easily writing applications which process vast amounts of data (multi-terabyte data-sets) in-parallel on large clusters (thousands of nodes) of commodity hardware in a reliable, fault-tolerant manner”. Just like the HDFS, the MapReduce framework also has master/slave architecture. There is one primary resource manager/master, the job tracker, and each node will have its own node manager, the task tracker. The master is responsible for scheduling the jobs' component tasks on the slaves, monitoring them and re-executing the failed tasks. The slaves execute the tasks as directed by the master.

The MapReduce is a combination of two components, Map and Reduce with distinct and separate functionalities. The map job component takes a set of data and converts it into independent chunks which are processed by the map tasks in a completely parallel manner, where individual elements are broken down exclusively into tuples (<key, value>).

The framework performs the sorting of the output from the map task then forwards it as input into the *reduce task*. The reducer stage of this process comprises the shuffle and reducer component. The output at this stage is aggregated and persisted in the file system back into the Hadoop server. These processes are monitored by the framework starting from the assignment of tasks, verification of task completion down to the copying of data across the clusters between the nodes.

Figure 3.3 on the next page provides an overview of MapReduce

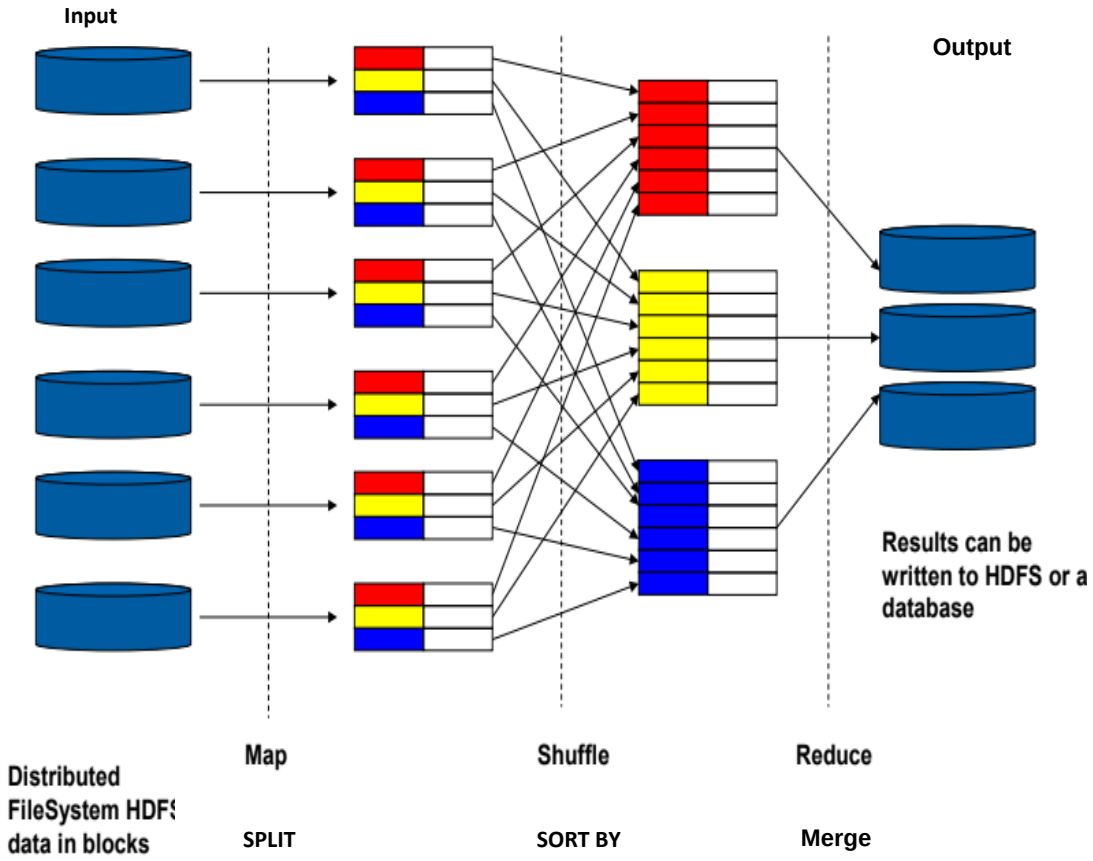


Figure 3.3: An overview of MapReduce

(Source: Adapted from IBM Corporation, 2013)

3.2 Apache Spark

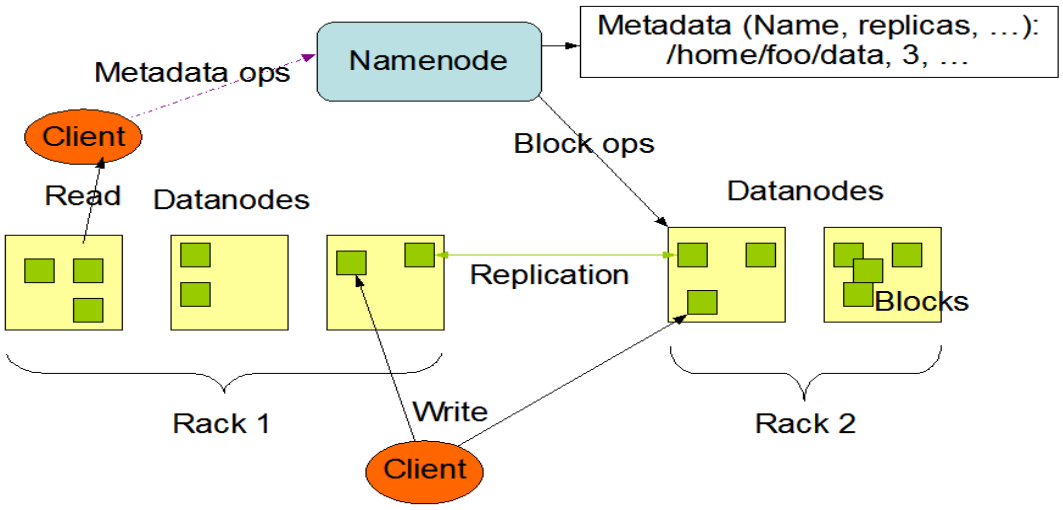


Figure 3.4: Overview of Apache Spark Libraries

Apache Spark is also one of the Apache Software Foundation's open-source software is built on top of the Hadoop Distributed File System (HDFS) platform, although it can be used as a standalone program. The Apache Spark framework is a fast and general engine for large-scale data processing used for real-time analysis of datasets in a distributed computing environment in big data. Spark comes packed with high-level libraries, supports writing applications quickly in Java, Scala, Python and R. Spark, runs programs 100 times faster than Hadoop MapReduce in memory or 10 times faster on disk. This tackles the challenges that are inherent to Hadoop MapReduce and removes all the abstractions.

At the time of writing this research report, Spark powers a stack of libraries and projects including; Spark Streaming, Spark SQL and DataFrames for structured data, Machine Learning Library (MLlib) for machine learning and GraphX (Graph-Based System) for visualization. It has support for using all these libraries in an application (The Apache Software Foundation, n.d.).

The Spark's core abstraction for working with data is termed the resilient distributed dataset (RDD) as shown in Figure 3.5 below. The RDD was introduced at the University of California at Berkeley in 2011. RDD is a fault-tolerant abstraction for in-memory cluster computing, which is an immutable distributed collection of objects. RDDs are created in two approaches, by either parallelizing an existing collection in the driver program, or by referencing a dataset in an external storage system, such as a shared file system, HDFS, HBase, or any data source offering a Hadoop input format. The potential of Spark should be utilised in combating cybersecurity challenges in this research on the big data platform.

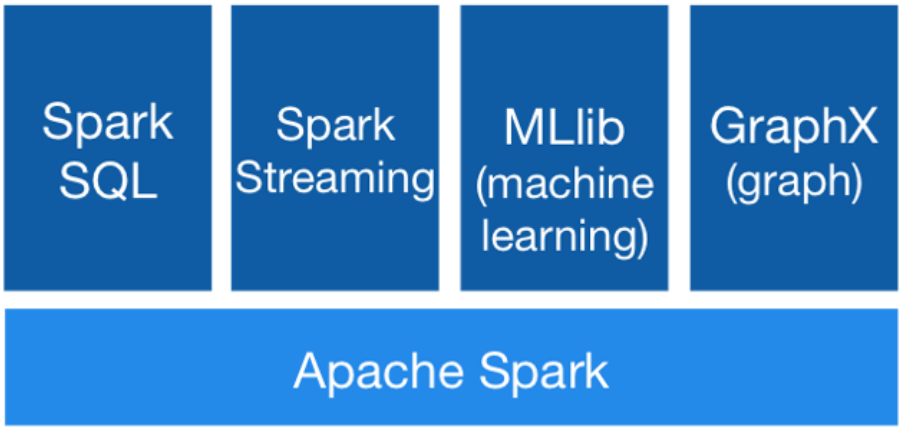


Figure 3.5: Overview of Apache Spark Libraries

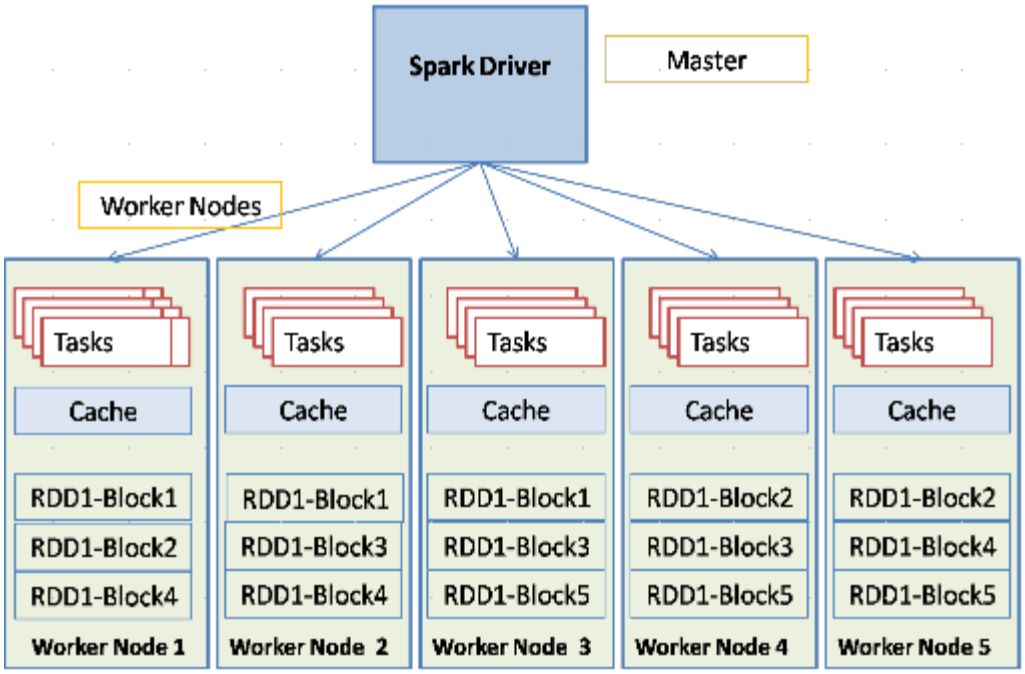


Figure 3.6: Spark's Components

3.3 The Predictive Model for Intrusion Detection

I propose to use a supervised machine learning algorithm from the Sparks MLlib. A decision tree algorithm, which is a classification-based technique, will be used in modelling a prediction model on the training dataset and will be tested on the test dataset. Both the training and test dataset are labelled datasets.

A decision tree is a decision support tool that uses a tree-like graph or model of decisions and their possible consequences, including chance event outcomes, resource costs, and utility. Decision tree classification leverages decision trees in modelling predictive models. Here observations are mapped to the class of a targeted value (Xu, Zong & Yang, 2013). A decision tree of a pair (x, y) represents a function that takes the input attribute x (Boolean, discrete, continuous) and outputs a simple Boolean y.

Decision tree classification is selected from among many other algorithms in this research due to the simplicity attached to interpreting it. In addition, it handles categorical features and extends to the multi-class classification setting. Decision tree classification does not require feature scaling, and is able to capture non-linearities and feature interactions.

3.3.1 Performance Metrics of the Trained Model

Using the decision tree classifier discussed above, which is a binary classification model, we shall be discussing some of the evaluation metrics to check the performance of the predicted intrusions as either normal connection (labelled “0”) or bad connection (labelled “1”). The predicted data points will be assigned to one of four categories:

Table 3.1: Categories of the Predicted Data Points

	Attack		
Attack		No [Normal (0)]	Yes [Attack (1)]
	No [Normal (0)]	True Negative (TN) alarm	False Negative (FN) alarm
	Yes [Attack (1)]	False Positive (FP) alarm	True Positive (TP) alarm

a. True Positive (TP): Connection is an attack and prediction is also an attack connection.

b. True Negative (TN): Connection is normal and prediction is also a normal connection.

c. False Positive (FP): Connection is normal but prediction is an attack connection.

d. False Negative (FN): Connection is an attack but prediction is a normal connection.

The following evaluation metrics will be used for the prediction performance:

Table 3.2: Trained Classifier Model Evaluation Metrics Definitions

Metric	Definition
Precision (Positive Predictive Value)	$PPV = \frac{TP}{TP + FP}$
Recall (True Positive Rate)	$TPR = \frac{TP}{P} = \frac{TP}{TP + FN}$
F-measure (Accuracy)	$F(\beta) = (1 + \beta^2) \times \left(\frac{PPV \times TPR}{\beta^2 . PPV + TPR}\right)$
Receiver Operating Characteristic (ROC)	$FPR(T) = \int_T^{\infty} P_0(T)dT$ $TPR(T) = \int_T^{\infty} P_1(T)dT$
Area Under ROC Curve	$AUROC = \int_0^1 \frac{TP}{P} d\left(\frac{FP}{N}\right)$
Area Under Precision-Recall Curve	$AUPRC = \int_0^1 \frac{TP}{TP + FP} d\left(\frac{TP}{P}\right)$

The classification model will output performance “score” (probability) for each class (normal connection or bad connection), the higher score indicates higher likelihood.

3.4 Dataset for the Detection Algorithm

The KDD Cup 99 Dataset which is the benchmark dataset for intrusion detection was used in training the model. It is the dataset that was prepared for the International Knowledge Discovery and Data Mining Tools Competition. The dataset was retrieved via http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html from the UCI Machine Learning Archive. “The competition task was to build a network intrusion detector, a predictive model capable of distinguishing between “bad” connections, called intrusions or attacks, and “good” normal connections.

A connection in this work means a sequence of TCP packets starting and ending at some well-defined times, between which data flows from a source IP address to a target IP address under some well-defined protocol. This database contains a standard set of data to be audited, which includes a wide variety of intrusions simulated in a military network environment" (UCI, 1999). This dataset was downloaded, extracted and stored in the HDFS for pre-processing, feature engineering, training, testing, prediction and validation by the Hadoop/Spark setup. The database, reduced ten percent, was used in this work. Features of this dataset are fully tabulated in Appendix C.

3.5 Big Data Architecture for the Security Framework

In this section, I present the proposed architecture of the set up for the experiment, the big data platform for security analytics. This proposed framework is not in any way a replacement of any security defence one has built, but is believed to be an added layer of the security measures to be considered. This framework will leverage the potential of big data and machine learning algorithms for advanced analytics to combat APTs. Most previous platforms have been unable to trap and analyse entire logs due to their variety, variability and volume.

Figure 3.7 shows big data security analytics architecture used for the experiment

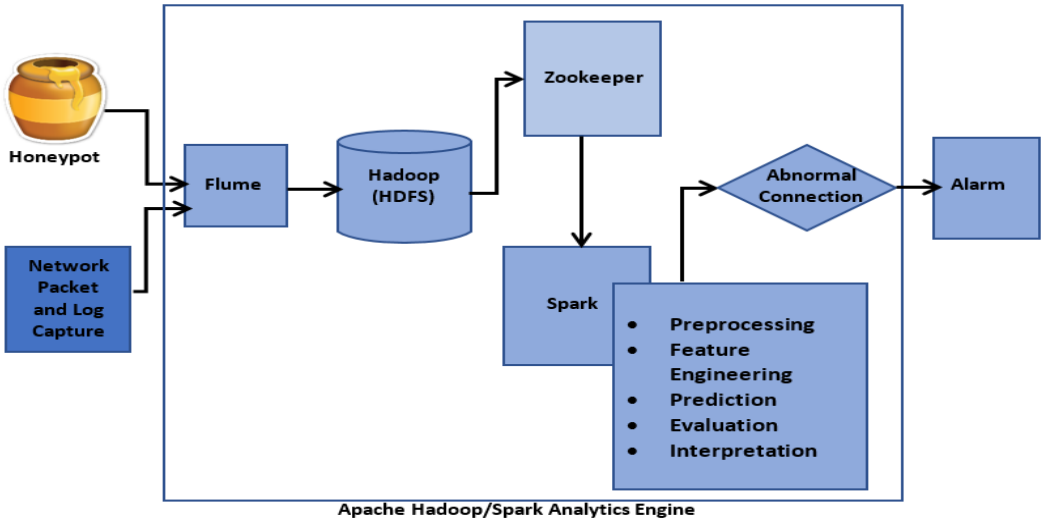


Figure 3.7: Big Data Security Analytics Architecture for the Experiment

Figure 3.6 is the schematic view of the framework for the proposed security analytics. Data is expected to be sourced from the network where necessary firewalls and port mirroring might have been configured and also from the honeypot where the behaviour of attackers can be logged into the Flume sink to be persisted in the Hadoop File System for storage. The data from the honeypot can be stored in the HBase or directly into the HDFS which is structured in a columnar format. All network events are stored unstructured into the HDFS for advanced analytics and exploration.

The streaming data from the network logs and honeypot are then analysed by Apache Spark and coordinated by the Apache ZooKeeper. The ZooKeeper is aimed at coordination and timing of services to avoid a race condition. The network logs are analysed by the Spark using the MLlib and StreamingContext library, however, because the experiment was set up on a single node, the programs are not written in the distributed format. The behavioural pattern of users through the honeypot was not analysed automatically but had to be monitored by the security expert or team.

After advanced analytics, if abnormal or illegitimate traffic is observed then the alarm will be raised. The IDS set-up in this experiment does not include prevention; hence the security expert will come in to the tackle the intrusion.

3.5.1 Set-up of Hadoop and Other Components

The implementation for the intrusion detection was done on an 8GB RAM and 500GB HDD PC with an Ubuntu 16.04.3 LTS operating system. A virtual machine was installed and assigned 4GB RAM and 80GB Local Disk memory to be used for the honeypot service. Apache Hadoop 2.7.4 was installed, as well as *Secure Shell (SSH)* and Oracle Java 8 for this experiment. The Hadoop was configured as a standalone cluster. All services were started for a test with an HDFS directory created on the DataNode. A preview of the services running can be seen in Appendix A for the Hadoop Cluster, NameNode and DataNode.

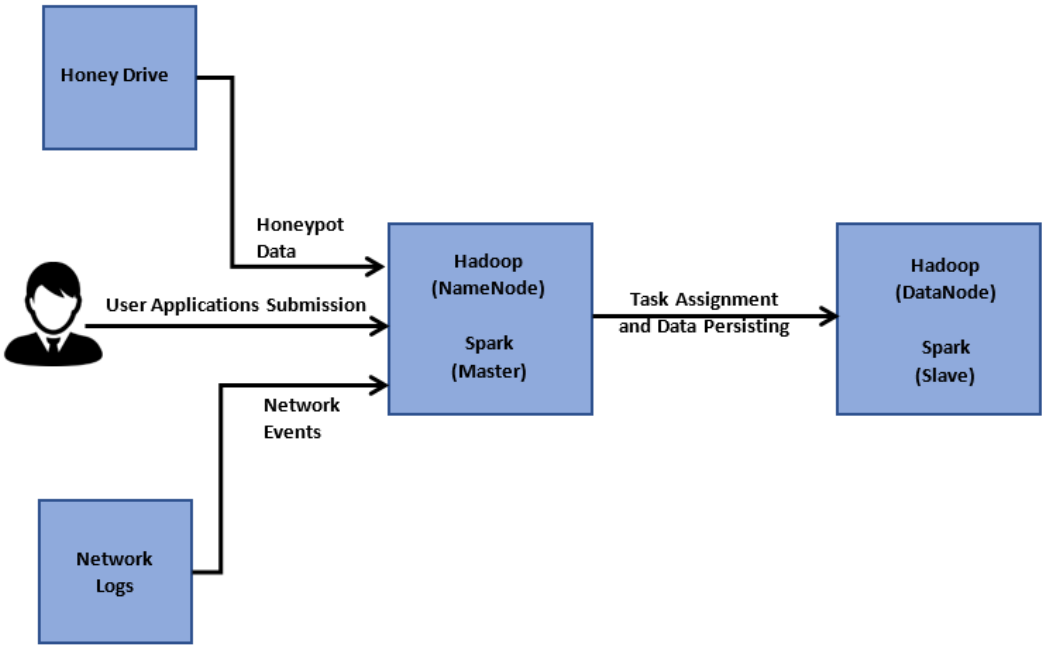


Figure 3.8: Use Case Diagram of the Hadoop/Spark Environment

Apache Spark 2.2.0 was installed and configured on a single node cluster and integrated with Hadoop and the other components of the ecosystem.

3.5.2 Experimental Set Up of the Honeypot: HoneyDrive

We chose HoneyDrive, a project by the BruteForce Lab. This was chosen among many other tools because of its simplicity of use, having preinstalled honeypots configured and installed already. It is an OVA file which comes as a VM with a Xubuntu OS. These honeypots include Kippo SSH honeypot, Dionaea and Amun malware honeypots, Honeyd low-interaction honeypot, Glastopf web honeypot and Wordpot, Conpot SCADA/ICS honeypot, Thug and PhoneyC honeyclients among others. Out of these pre-configured honeypot software packages, Kippo and Honeyd were tested.

The Kippo has a web interface where attackers' activities are analysed and visualized; and it supports processing of the data it captures via Kippo-Graph. Other scripts included in HoneyDrive are Honeyd-Viz, DionaeaFR, an ELK stack and many more. Lastly, almost 90 well-known malware analysis, forensics and network monitoring related tools are also present in the distribution. Meanwhile, the Honeyd has the advantage of setting up a honeynet with the varying OS. These honeynets are assigned to open ports. The *ifconfig* command was used to display the honeypot's IP address. A pictorial overview of these honeypots can be seen in Appendix B.

CHAPTER FOUR
EXPERIMENTAL RESULTS AND EVALUATION

This chapter of the research presents the results of the analysis and experiment carried out as discussed in Chapter Three in compliance with the outlined aims and objectives of the research work.

4.1 Findings from the Honeynet

As described in Chapter Three, Section 3.4.2, several honeypots from HoneyDrive were kicked running and several attempts were simulated. It turns out interestingly that one of the honeypots configured, Kippo, graphically displayed the behavioural patterns and actions of the attacks simulated. These were accessed through the web URL via http://localhost/kippo-graph/index.php on my localhost where the HoneyDrive was installed. The logs generated were stored unstructured in the HDFS.

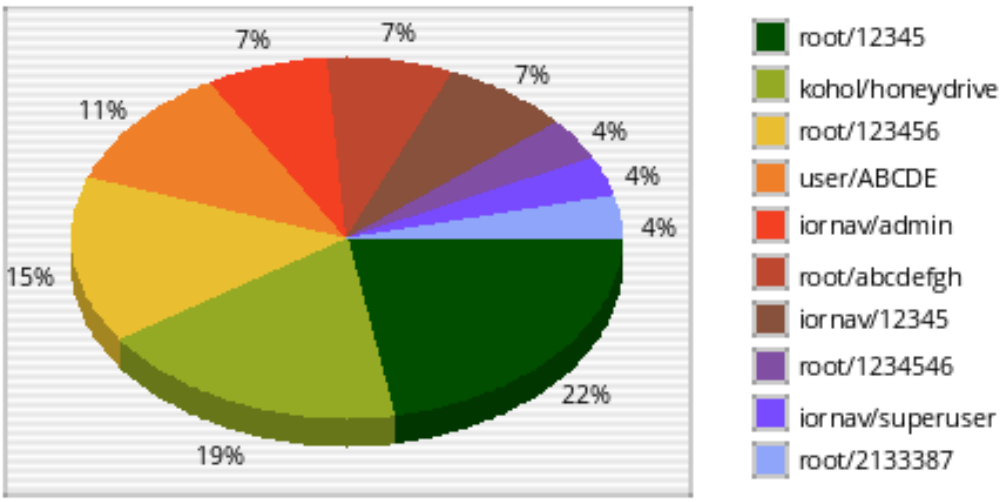


Figure 4.1: Top 10 Username-Password Combinations Generated From Kippo

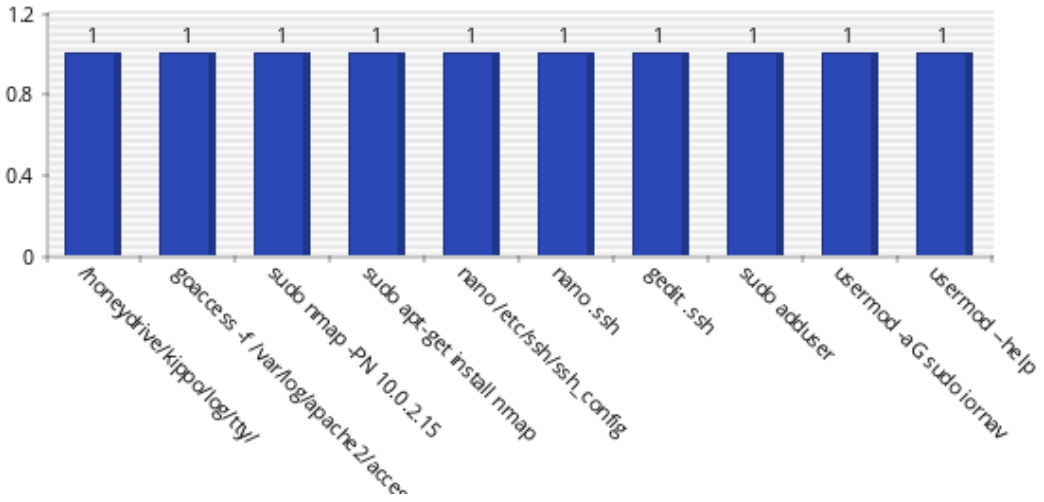


Figure 4.2: Top 10 failed input

Interesting insights were gleaned from the configured ssh honeypots, like the commands executed by successful logins from users, usernames and passwords attempted for login and lots more. Records of successful and failed commands were all recorded with their respective statistics. IP addresses of the communications with the server were also recorded as shown in figure 4.1 and figure 4.2 above.

4.2 Evaluation of the Intrusion Classification Model

Analysis and findings from the KDD Cup 99 Dataset showed that there were 97,278 'normal' interactions and 396,743 'bad ‘interactions in the dataset for training, out of the total 494,021 connections in the reduced dataset of ten percent. In the full dataset of 4,898,431 connections, there were 972,781 'normal' interactions and 3,925,650 'bad ‘interactions.

Sub categories of attacks in the training dataset showed the following statistics:

Table 4.1: Sub Categories of Attacks in the Dataset

Attack	Count Value	Attack	Count Value
Smurf	280790	buffer_overflow	30
Neptune	107201	Land	21
Normal	97278	warezmaster	20
Back	2203	Imap	12
Satan	1589	Rootkit	10
Portsweep	1040	loadmodule	9
Warezclient	1020	ftp_write	8
Teardrop	979	Multihop	7
Pod	264	Phf	4
Nmap	231	Perl	3
guess_passwd	53	Spy	2

The classification model was built with a binary classification of 2: two classes “good” or “bad”, maximum depth of 4, maxBins as 100 and impurity to be ‘gini’. In the predictive model in the decision tree, “1” represents an attack (bad) and “0” represent a normal (good) connection.

Using the reduced dataset of 10 percent for the training data, the model was built and trained. It resulted in the learned decision tree with 25 nodes as shown in Appendix D.

From the above learned model, it can be observed that features in the table below are predominant out of the 42 features in determining the category of the connections.

Table 4.2: Features Generated in the Decision Tree Model

Feature Name/type	Feature Index
logged_in: symbolic	11
src_bytes: continuous	4
hot: continuous	9
wrong_fragment: continuous	7
dst_bytes: continuous	5
error_rate: continuous	26
service: symbolic	2
count: continuous	22

The model was again trained with the full dataset to ascertain the level of accuracy of the prediction and the summary of the performance is presented in Table 4.3 below:

Table 4.3: Brief Statistics of the Dataset used for Training and Testing

	Training Dataset	Test Dataset
Total Size of connection	494,021	311,029
Size of normal connections	97,278	60,593
Size of bad connections	396,743	250,436

Table 4.4: Summary of Evaluation Metrics

Metric	Value
Area under PR	1
Area under ROC	0.8614
Precision of False Normal Connection	98.315%
Precision of True Attacked Connection	100%
Recall of False Normal Connection	72.7162%
Recall of True Attacked Connection	99.5544%
Test Error	7.5147%
The Test Accuracy	92.4853%

Table 4.5: Confusion Matrix for the Predicted Connection

	Normal Connection	Bad Connection
Normal Connection	59,572	22,352
Bad Connection	1,021	228,084
Total	60,593	250,436

From Table 4.5 above, the model predicted 59,572 out of 60,593 normal connections correctly and 228,084 out of 250,436 bad connections correctly; while 1,021 and 22,352 respectively were wrongly classified.

As shown in Table 4.4 above, there was an accuracy of 92.4853% from the prediction, which shows the ability to predict accurately, however, the statistics are not enough to evaluate the true performance of a binary classification model. Therefore the value of Area under PR (AUPR) and Area under ROC (AUROC) had to be checked. The AUPR value of 1, and AUROC of value 0.8614, shows that the prediction model is highly accurate.

CHAPTER FIVE
LIMITATIONS, CONCLUSION AND FUTURE RESEARCH

This is the concluding chapter of the research work carried out. The summary, limitations and future work are presented.

5.1 Limitations

This experiment lacked real-time data from a production or live network. This limited the extent to which data could be collected from varying sources; from the network logs and packets, and honeypots. The big data platform, Hadoop ecosystem was configured on a single node cluster thereby limiting the performance of the setup instead of fully deploying it in a distributed environment.

5.2 Summary and Conclusion

Advanced persistent threats are getting stealthier than before with a lot of sophistication, exploring zero-day vulnerabilities and leveraging social engineering campaigns against victims among many other techniques. Activities of an APT can only be used to gain insights when they are well collected, and proper data mining tools are explored to exploit the data. This experiment showed that, when large amounts of such data is collected, patterns can be derived from a big data platform to model prediction.

The honeypots showed that the astuteness of attackers and their mode of attack can be captured and known when proper forensics is carried out. From the modelled classifier, the accuracy level showed that intrusions can be predicted with a lower error rate with a streaming data with low.

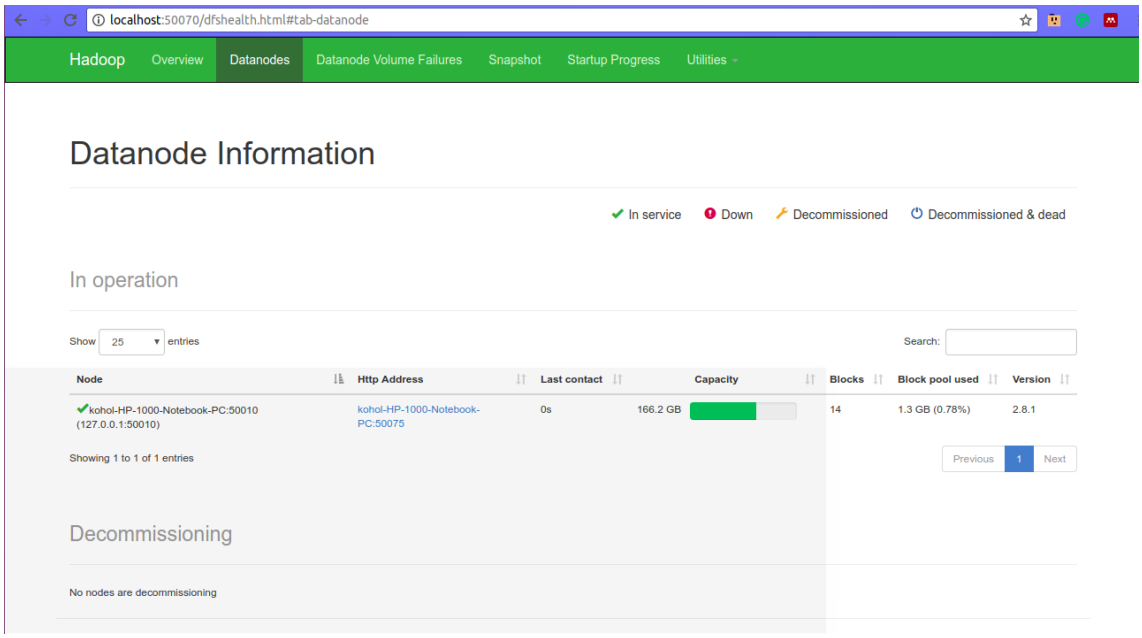
In conclusion, this experiment shows that big data analytics for cybersecurity have the ability to detect and analyse trends in network activities and that they can be put in place to accurately identify and prevent attacks that can compromise an organization's network in the early stages.

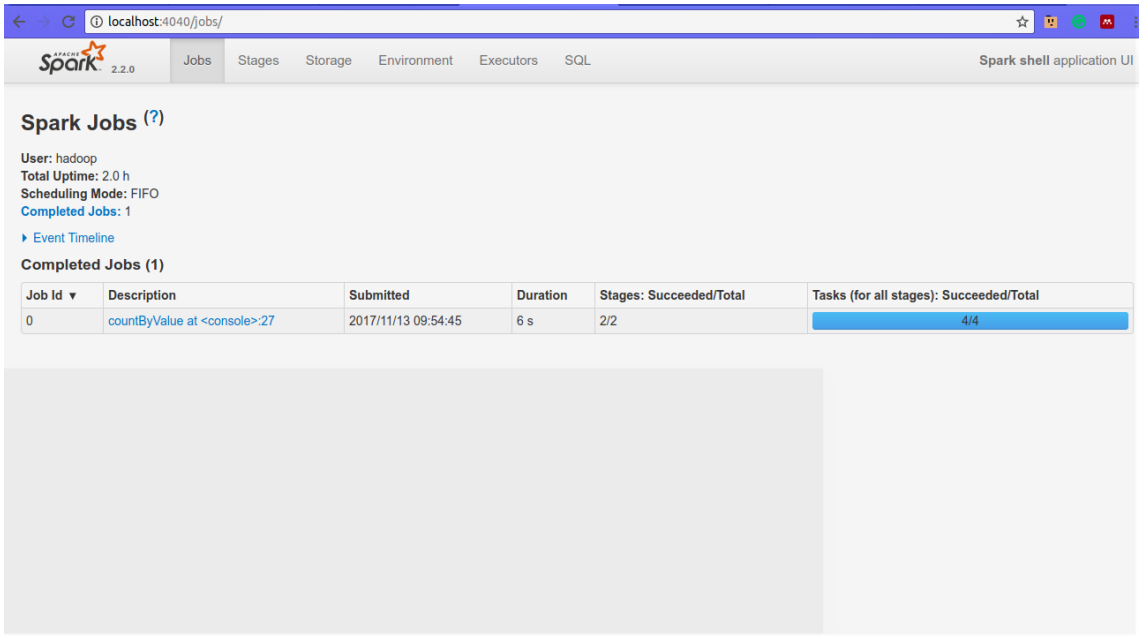
5.3 Future Research

Future work on the big data platform for security analytics should consider using unsupervised models as there would be no labels in the real-time data streamed. Trying out a multiclass prediction, predicting sub-categories of attacks/intrusion specifically and integrating it along with intrusion prevention will be very useful in helping security personnel to obtain 360 degrees security from cyberattacks. This will save a lot of time and would not need human intervention to monitor alarms triggered to proffer solutions to the intrusions, instead appropriate actions and decisions will be taken by the system to combat any breach of the security checks.

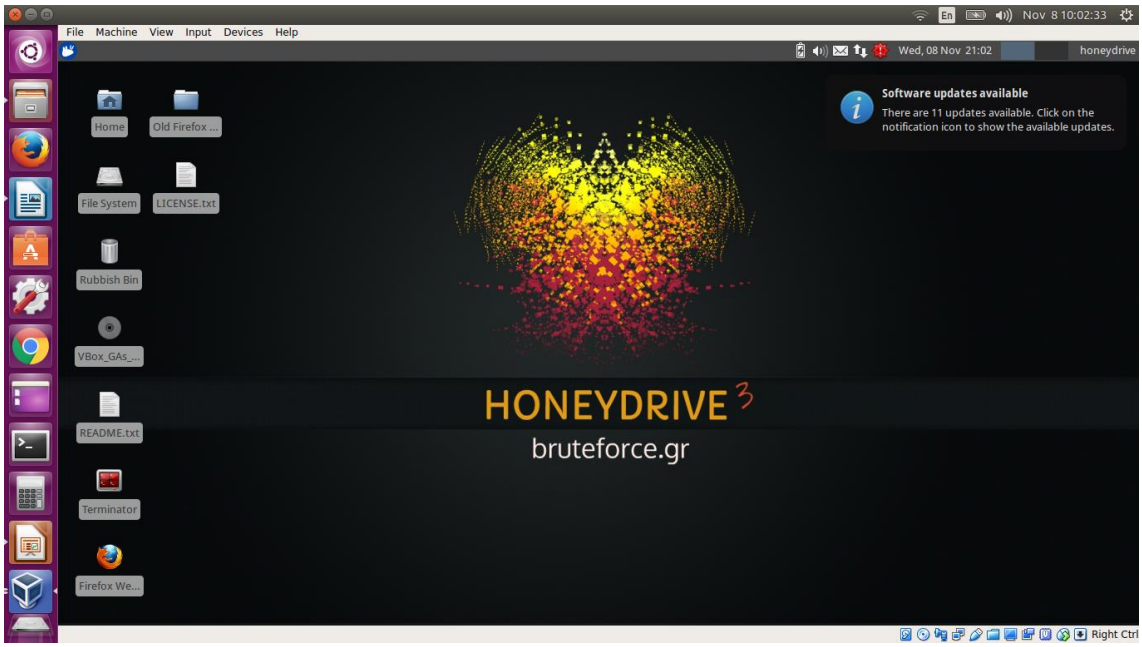
APPENDICES

Appendix A: Screen Capture of the Configured Standalone Hadoop Ecosystem





Appendix B: Overview of the Installed Honeypot: Honeydrive



Appendix C: KDD Cup 1999 Dataset Features

Feature Name	Description	Type
duration	length (number of seconds) of the connection	continuous
protocol_type	type of the protocol, e.g. tcp, udp, etc.	discrete
service	network service on the destination, e.g., http, telnet, etc.	discrete
src_bytes	number of data bytes from source to destination	continuous
dst_bytes	number of data bytes from destination to source	continuous
flag	normal or error status of the connection	discrete
land	1 if connection is from/to the same host/port; 0 otherwise	discrete
wrong_fragment	number of ``wrong'' fragments	continuous
urgent	number of urgent packets	Continuous
	Basic features of individual TCP connections	
hot	number of ``hot'' indicators	continuous
num_failed_logins	number of failed login attempts	continuous
logged_in	1 if successfully logged in; 0 otherwise	discrete
num_compromised	number of ``compromised'' conditions	continuous
root_shell	1 if root shell is obtained; 0 otherwise	discrete
su_attempted	1 if ``su root'' command attempted; 0 otherwise	discrete
num_root	number of ``root'' accesses	continuous
num_file_creations	number of file creation operations	continuous
num_shells	number of shell prompts	continuous

num_access_files	number of operations on access control files	continuous
num_outbound_cmds	number of outbound commands in an ftp session	continuous
is_hot_login	1 if the login belongs to the ``hot'' list; 0 otherwise	discrete
is_guest_login	1 if the login is a ``guest'' login; 0 otherwise	discrete
	Content features within a connection suggested by domain knowledge	
count	number of connections to the same host as the current connection in the past two seconds	continuous
	Note: The following features refer to these same-host connections.	
serror_rate	% of connections that have ``SYN'' errors	continuous
rerror_rate	% of connections that have ``REJ'' errors	continuous
same_srv_rate	% of connections to the same service	continuous
diff_srv_rate	% of connections to different services	continuous
srv_count	number of connections to the same service as the current connection in the past two seconds	continuous
	Note: The following features refer to these same-service connections.	
srv_serror_rate	% of connections that have ``SYN'' errors	continuous
srv_rerror_rate	% of connections that have ``REJ'' errors	continuous
srv_diff_host_rate	% of connections to different hosts	continuous

Sub Categories of Attacks

back	Dos

buffer_overflow	u2r
ftp_write	r2l
guess_passwd	r2l
imap	r2l
ipsweep	Probe
land	Dos
loadmodule	u2r
multihop	r2l
neptune	Dos
nmap	Probe
perl	u2r
phf	r2l
pod	Dos
portsweep	Probe
rootkit	u2r
satan	Probe
smurf	Dos
spy	r2l
teardrop	Dos
warezclient	r2l
warezmaster	r2l

Appendix D: Learned Classification Tree Model

```
DecisionTreeModel classifier of depth 4 with 25 nodes
  If (feature 22 <= 84.0)
   If (feature 4 <= 28.0)
    If (feature 2 in {0.0,10.0,56.0,42.0,24.0,25.0,52.0,14.0,20.0,46.0,57.0,61.0,6.0,28.0,38.0,21.0,33.0,9.0,53.0,1
3.0,41.0,2.0,34.0,45.0,64.0,17.0,22.0,44.0,59.0,27.0,12.0,54.0,49.0,7.0,39.0,35.0,48.0,18.0,50.0,16.0,43.0,26.0,55.
0,23.0,8.0,58.0,36.0,30.0,51.0,19.0,47.0,15.0,62.0})
     If (feature 37 <= 0.55)
      Predict: 0.0
     Else (feature 37 > 0.55)
      Predict: 1.0
    Else (feature 2 not in {0.0,10.0,56.0,42.0,24.0,25.0,52.0,14.0,20.0,46.0,57.0,61.0,6.0,28.0,38.0,21.0,33.0,9.0,5
3.0,13.0,41.0,2.0,34.0,45.0,64.0,17.0,22.0,44.0,59.0,27.0,12.0,54.0,49.0,7.0,39.0,35.0,48.0,18.0,50.0,16.0,43.0,26.
0,55.0,23.0,8.0,58.0,36.0,30.0,51.0,19.0,47.0,15.0,62.0})
     If (feature 11 <= 0.0)
      Predict: 1.0
     Else (feature 11 > 0.0)
      Predict: 0.0
   Else (feature 4 > 28.0)
    If (feature 9 <= 0.0)
     If (feature 7 <= 0.0)
      Predict: 0.0
     Else (feature 7 > 0.0)
      Predict: 1.0
    Else (feature 9 > 0.0)
     If (feature 4 <= 1107.0)
      Predict: 0.0
     Else (feature 4 > 1107.0)
      Predict: 1.0
  Else (feature 22 > 84.0)
   If (feature 5 <= 0.0)
    If (feature 11 <= 0.0)
     Predict: 1.0
    Else (feature 11 > 0.0)
     Predict: 0.0
   Else (feature 5 > 0.0)
    If (feature 2 in {1.0,33.0,9.0,34.0,35.0,4.0})
     If (feature 26 <= 0.0)
      Predict: 0.0
     Else (feature 26 > 0.0)
      Predict: 1.0
    Else (feature 2 not in {1.0,33.0,9.0,34.0,35.0,4.0})
     Predict: 1.0
```

Appendix E: Pyspark Code for the Analysis and Classifier Model

```
# Displaying the dataset with few statistics
print("Below are the top 5 rows of the TRAINING DATASET: \n{}".format(train_data.take(5)))
print("\nBelow are the top 5 rows of the TEST DATASET: \n{}".format(test_data.take(5)))
training_data_count = train_data.count()
print("\nThere are {} records in the training data set of the KDD Cup 99 10 percent dataset"\
      .format(training_data_count))
print("\nWhile we have {} records in the test data set of the KDD Cup 99 dataset"\
      .format(test_data.count()))

# Checking the number of normal interactions in the dataset
normal_train_data = train_data.filter(lambda x: 'normal.' in x)
attacked_train_data =train_data.subtract(normal_train_data)
normal_count = normal_train_data.count()
attack_count = attacked_train_data.count()
normalTestData = test_data.filter(lambda x: 'normal.' in x)
attackedTestData = test_data.subtract(normalTestData)
normalCountTest = normalTestData.count()
attackCountTest = attackedTestData.count()
testDataCount = test_data.count()
print("Found {} 'normal' interactions and {} 'bad'interactions in the dataset for training,\
out of the total size of {} records".format(normal_count, attack_count, training_data_count))
print("\nWe also found {} 'normal' interactions and {} 'bad'interactions in the test dataset,\
out of the total size of {} records".format(normalCountTest, attackCountTest, testDataCount))

# Sample of the data with numerical values
train_data_sample = train_data.sample(False, 0.1, 1234)
sample_size = train_data_sample.count()
total_size = train_data.count()
print("Sample size is {} of {} in the taining dataset".format(sample_size, total_size))
train_data_sample_items = train_data_sample.map(lambda x: x.split(","))
sample_normal_tags = train_data_sample_items.filter(lambda x: "normal." in x)
sample_normal_tags_count = sample_normal_tags.count()
sample_normal_ratio = sample_normal_tags_count / float(sample_size)
print("The ratio of 'normal' interactions is {} in the training dataset".format(round(sample_normal_ratio,3)))

# The types of protocols in the connections in the dataset
protocols_types = trainningDataFormatted.map(lambda col: col[1]).distinct()
print("We have the following distinctive protocols in the dataset\n{}".format(protocols_types.collect()))
```

```
import os
import sys
import string
# import time
import datetime
# spark runtime
from pyspark import SparkConf
from pyspark import SparkContext
sc = SparkContext.getOrCreate()
# spark mllib
from pyspark.mllib.regression import LabeledPoint
from pyspark.mllib.stat import Statistics
from numpy import array
from math import sqrt
# import classifier
from pyspark.mllib.tree import DecisionTree, DecisionTreeModel
#import classifier evaluation metrics
from pyspark.mllib.evaluation import BinaryClassificationMetrics
from pyspark.mllib.evaluation import MulticlassMetrics
# The Workbench DataSet used in this analysis http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html
"""
# To load the data from the online database of UCI KDD Repository
import urllib.request
from gzip import GzipFile
url = "http://kdd.ics.uci.edu/databases/kddcup99/kddcup.data.gz"
url1 = "http://kdd.ics.uci.edu/databases/kddcup99/corrected.gz"
training_file = urllib.request.urlopen(url, "kddcup.data.gz")
test_file = urllib.request.urlopen(url1, "corrected.gz")
"""
programStartTime = datetime.datetime.now()
# Loading the reduced 10 percentage dataset from the hdfs into the spark for training
training_file = "hdfs://localhost:9000/home/hadoop/hadoopdata/Store/kddcup.data_10_percent_corrected"
#training_file = "hdfs://localhost:9000/home/hadoop/hadoopdata/Store/kddcup.data.corrected" #full data set

# Loading the test dataset from the hdfs into the spark for testing the predictive model
test_file = "hdfs://localhost:9000/home/hadoop/hadoopdata/Store/corrected"

# Preparing the data for preprocessing
train_data = sc.textFile(training_file).cache()
test_data = sc.textFile(test_file).cache()
trainningDataFormatted = train_data.map(lambda x: x.split(","))
testDataFormatted = test_data.map(lambda x: x.split(","))
```

```
# Get the statistics of the sub attacks in the
print("\nThe sub-category of attacks in the dataset with their count value are: ")
keyValueData = trainningDataFormatted.map(lambda x: (x[41], x))
subAttacksKey = keyValueData.countByKey()
print(subAttacksKey)

# Getting fields with symbolic/categorical data
protocols = trainningDataFormatted.map(lambda x: x[1]).distinct().collect()
services = trainningDataFormatted.map(lambda x: x[2]).distinct().collect()
flags = trainningDataFormatted.map(lambda x: x[3]).distinct().collect()

"""
Parsing through the dataset for features with numerical values in the training dataset and leaving out
symbolic(categorical) features in [1]=protocol_type, [2]=service, [3]=flag and [41]=categoric_of_interaction
"""
def create_labeled_point(line_split):
    clean_line_split = line_split[0:41]
    try:
        clean_line_split[1] = protocols.index(clean_line_split[1])
    except:
        clean_line_split[1] = len(protocols)
    try:
        clean_line_split[2] = services.index(clean_line_split[2])
    except:
        clean_line_split[2] = len(services)
    try:
        clean_line_split[3] = flags.index(clean_line_split[3])
    except:
        clean_line_split[3] = len(flags)
    # converting label to binary label
    intrusion = 1.0
    if line_split[41]=='normal.':
        intrusion = 0.0
    return LabeledPoint(intrusion, array([float(x) for x in clean_line_split]))

trainingData = trainningDataFormatted.map(create_labeled_point)
testData = testDataFormatted.map(create_labeled_point)
# Building the training model
start_time = datetime.datetime.now()
model = DecisionTree.trainClassifier(trainingData, numClasses=2, \
                                     categoricalFeaturesInfo=\
                                     {1: len(protocols), 2: len(services), 3: len(flags)},
                                     impurity='gini', maxDepth=4, maxBins=100)
```

```
end_time = datetime.datetime.now()
timedelta = round((end_time-start_time).total_seconds(), 2)
print("A predictive model: Decision Tree Classifier was trained with the training dataset in {} \
seconds".format(timedelta))
predictions = model.predict(testData.map(lambda p: p.features))
labelsAndPredictions = testData.map(lambda p: p.label).zip(predictions)

# Model Evaluation Statistics: # Evaluate model on test instances and compute test error
print("Summary of Evaluation Metrics")
testErr = labelsAndPredictions.filter(lambda lp: lp[0] != lp[1]).count() / float(testData.count())
biMetrics = BinaryClassificationMetrics(labelsAndPredictions)
print("Area under PR = {}".format(round(biMetrics.areaUnderPR),4)) # Area under precision-recall curve
print("Area under ROC = {}".format(round(biMetrics.areaUnderROC,4))) # Area under ROC curve
def printEvaluationMetrics(predictions_and_labels):
    metrics = MulticlassMetrics(labelsAndPredictions)
    print ('Precision (Positive Predictive Value) of Normal Connection is {} percent\
    '.format(round(metrics.precision(0)*100,4)))
    print ('Precision (Positive Predictive Value) of bad Connection) is {} percent\
    '.format(round(metrics.precision(1))*100,4))
    print ('Recall (True Positive Rate) of Normal Connection is {} percent\
    '.format(round(metrics.recall(0)*100,4)))
    print ('Recall (True Positive Rate) of bad Connection is {} percent\
    '.format(round(metrics.recall(1)*100,4)))
    print ('Confusion Matrix\n', metrics.confusionMatrix().toArray())
printEvaluationMetrics(labelsAndPredictions)
print('Test Error = {} percent'.format(round(testErr*100,4)))
print("The Test Accuracy is {} percent".format(round((1-testErr)*100,4)))

#  Printing the classifier (decison tree classifier)
print('Learned classification tree model:')
print(model.toDebugString())

# Save and load model
model.save(sc, "/home/hadoop/hadoopdata/DecisionTreeClassifierModel")
sameModel = DecisionTreeModel.load(sc, "/home/hadoop/hadoopdata/DecisionTreeClassifierModel")
sc.stop()

#Entire program execution time
programEndTime = datetime.datetime.now()
timedelta1 = round((programEndTime-programStartTime).total_seconds(), 2)
print("The entire program execution time is {} seconds".format(timedelta1))
```

REFERENCES

Ali, A., Qadir, J., Rasool, ur R., Sathiaseelan, A., Zwitter, A., & Crowcroft, J. (2016). Big data for development: applications and techniques. *Big Data Analytics*, *1*(1), 2. https://doi.org/10.1186/s41044-016-0002-4

Ambreen, R., Nadeem, S., & Dubey, S. (2016). Review of APT attacks: How big data fights back. *International Journal of Engineering Sciences & Research Technology*, *5*(10), 501–513. https://doi.org/10.5281/zenodo.154215

Andress, J., & Winterfeld, S. (2011). *Cyber warfare techniques, tactics and tools for security practitioners*. Waltham, MA: Syngress.

Aydın, M. A., Zaim, A. H., & Ceylan, K. G. (2009). A hybrid intrusion detection system design for computer network security. *Computers and Electrical Engineering*, *35*(3), 517–526. https://doi.org/10.1016/j.compeleceng.2008.12.005

Beigh, B. M., & Peer, M. A. (2012). Intrusion detection and prevention system: Classification and quick review. *ARPN Journal of Science and Technology*, *2*(7), 661–675.

Beyer, A. M., & Laney, D. (2012). The Importance of “Big Data”: A Definition. Retrieved August 30, 2017, from https://www.gartner.com/doc/2057415/importance-big-data-definition

Boehmer, W. (2014). Towards analysis of sophisticated attacks, with conditional probability, genetic algorithm and a crime function. *Cd-Ares*, *8708*, 250–256. Retrieved from http://dblp.uni-trier.de/db/conf/IEEEares/cd-ares2014.html#Boehmer14

Brasso, B. (2016). Cyber attacks against critical infrastructure are no longer just theories « Executive Perspective. | FireEye Inc. Retrieved July 13, 2017, from https://www.fireeye.com/blog/executive-perspective/2016/04/cyber_attacks_agains.html

Cloud Security Alliance. (2013). *Big data analytics for security intelligence*. Cloud Security Alliance. https://doi.org/10.1145/2666652.2666664

CNSS. (2015). *Committee on National Security Systems (CNSS) glossary*. Retrieved from https://cryptosmith.files.wordpress.com/2015/08/glossary-2015-cnss.pdf

Danani, J., & Jani, J. (2012). Honeypot - a tool to trap website hackers. *Proceedings Published in International Journal of Computer Applications® (IJCA)*, (December), 8–13.

DeVan, A. (2016). The 7 V's of big data | Impact Radius. Retrieved August 30, 2017, from https://www.impactradius.com/blog/7-vs-big-data/

Dewar, S. R. (2014). The "Triptych of cyber security": A Classification of active cyber defence. In P. Brangetto, M. Maybaum, & J. Stinissen (Eds.), *6th International Conference on Cyber Conflict* (Vol. 3, pp. 3–17). Tallinn: NATO CCD COE Publications. https://doi.org/10.3233/978-1-60750-060-5-3

Diebold, P., Hess, A., & Schäfer, G. (2005). A Honeypot architecture for detecting and analyzing unknown network attacks. *Informatik Aktuell*, 245–255. https://doi.org/10.1007/3-540-27301-8_20

Dooley, M., & Rooney, T. (2017). Malware and APTs. In *DNS Security Management* (First Ed., pp. 195–210). JohnWiley & Sons.

DXC Technology. (2015). Five industries where big data is making a difference | DXC Technology. Retrieved August 30, 2017, from https://www.dxc.technology/analytics/insights/140512-five_industries_where_big_data_is_making_a_difference

Egupov, A. A., Zareshin, S. V, Yadikin, I. M., & Silnov, D. S. (2017). Development and implementation of a honeypot trap. *IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering*, 382–385.

ElevenPaths. (2017). CyberThreats. Retrieved August 19, 2017, from https://www.elevenpaths.com/technology/cyberthreats/index.html

Ernst and Young. (2015). *Using cyber analytics to help you get on top of cybercrime - Third-generation security operations centers*. Retrieved from http://www.ey.com/Publication/vwLUAssets/ey-third-generation-security-operations-centers-2015/$FILE/ey-third-generation-security-operations-centers-2015.pdf

Esposito, M., Mazzariello, C., Oliviero, F., Romano, S. P., & Sansone, C. (2006). Real time detection of novel attacks by means of data mining techniques. *Enterprise Information Systems*, 197–204.

European Commission. (2013). Commission staff working document. *Group*, 1–3.

https://doi.org/10.2903/j.efsa.2015.4206.OJ

FBI. (n.d.). Cyber crime — FBI. Retrieved July 14, 2017, from https://www.fbi.gov/investigate/cyber

Firican, G. (2017). The 10 Vs of big data - TDWI upside. Retrieved August 31, 2017, from https://upside.tdwi.org/Articles/2017/02/08/10-Vs-of-Big-Data.aspx?Page=2

Gloag, D. (n.d.). Cyber threats: Assessment & analysis | Study.com. Retrieved July 13, 2017, from http://study.com/academy/lesson/cyber-threats-assessment-analysis.html

Gragido, W. (2012). Understanding indicators of compromise (IOC) Part I - Speaking of security - The RSA Blog. Retrieved August 28, 2017, from https://blogs.rsa.com/understanding-indicators-of-compromise-ioc-part-i/

Graham, L. (2017). Cybercrime costs the global economy $450 billion: CEO. Retrieved July 14, 2017, from http://www.cnbc.com/2017/02/07/cybercrime-costs-the-global-economy-450-billion-ceo.html

Gronlund, J. C. (2017). What is machine learning on Azure? | Microsoft Docs. Retrieved September 6, 2017, from https://docs.microsoft.com/en-us/azure/machine-learning/machine-learning-what-is-machine-learning

Grottke, M., Sun, H., Fricks, R. M., & Trivedi, K. S. (2008). Ten fallacies of availability and reliability analysis. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *5017 LNCS*, 187–206. https://doi.org/10.1007/978-3-540-68129-8_15

Hosburgh, M. (2016). How to target critical infrastructure: The Adversary return on investment from an industrial control system. *SANS Institute InfoSec Reading Room*, 33.

IBM. (n.d.-a). Big data analytics | IBM Analytics. Retrieved August 31, 2017, from https://www.ibm.com/analytics/us/en/big-data/

IBM. (n.d.-b). What is big data analytics? – IBM Analytics. Retrieved September 1, 2017, from https://www.ibm.com/analytics/us/en/technology/hadoop/big-data-analytics/

Irwin, S., & Northcutt, S. (2014). Creating a threat profile for your organization. *Creating a Threat Profile for Your Organization*, 32.

ISO. (2014). *ISO/IEC 29147 Information Technology — Security Techniques — Vulnerability Disclosure* (1st Ed.). Switzerland: International Organization for Standardization. Retrieved from www.iso.org

Jain, A. (2016). The 5 Vs of big data - Watson Health Perspectives. Retrieved August 30, 2017, from https://www.ibm.com/blogs/watson-health/the-5-vs-of-big-data/

Janssen, T., & Grady, N. (2013). Big data for combating cyber attacks. *CEUR Workshop Proceedings*, *1097*, 158–161. Retrieved from http://ceur-ws.org/Vol-1097/STIDS2013_P1_JanssenGrady.pdf

Javaid, A. Y., Niyaz, Q., Sun, W., Alam, M., Javaid, A. Y., & Alam, M. (2016). A Deep learning approach for network intrusion Dtraining_file.map(_.split(',').last).countByValue().toSeq.sortBy(_._2).reverse.foreach(println)etection System. *Proceedings of the 9th EAI International Conference on Bio-Inspired Information and Communications Technologies (Formerly BIONETICS)*, (January). https://doi.org/10.4108/eai.3-12-2015.2262516

Johnson, C., Badger, L., Waltermire, D., Snyder, J., & Skorupka, C. (2016). Guide to cyber threat information sharing. *NIST Special Publication*, 800–150. https://doi.org/10.6028/NIST.SP.800-150

Kadir, M. F. H. A. (2013). *Early detection and prevention of application-layer DoS attacks against web servers*. (Master's thesis). University of Bath, Internet Systems and Security, Bath, United Kingdom.

Kaspersky Lab. (2017). *KSN Report: Ransomware in 2016-2017*. Retrieved from https://securelist.com/ksn-report-ransomware-in-2016-2017/78824/

Koutský, O. (2014). *Monitoring and Analysis of Cyber Attacks*. (Master's thesis). Masarykova Univerzita, Faculty of Information, Brno, Czech Republic. Retrieved from http://is.muni.cz/th/359295/fi_m/thesis.pdf

Lee, R. (2015). *The Sliding Scale of Cyber Security*.

Lendvay, L. R. (2016). *Shadows of Stuxnet: Recommendations for U.S. policy on critical infrastructure cyber defense derived from the Stuxnet attack*. (Master's thesis). Naval Postgraduate School, Monterey, California.

Lord, N. (2017). What are Indicators of Compromise? | Digital Guardian. Retrieved August 28, 2017, from https://digitalguardian.com/blog/what-are-indicators-compromise

Loukas, G. (2006). Defence against denial of service in self-aware networks.

Mainone Cable. (2017). Increased cyber attacks highlight need for managed security services. Retrieved July 14, 2017, from https://techpoint.ng/2017/06/13/mainone-combating-cyber-attacks-nigeria/

Michael, M. (2017). Few victims reporting ransomware attacks to FBI. Retrieved July 14, 2017, from https://threatpost.com/few-victims-reporting-ransomware-attacks-to-fbi/126510/

Mohsen, M., Nasaruddin, F., Gani, A., Karim, A., Hashem, I., Siddiqa, A., & Yaqoob, I. (2017). Big IoT data analytics: Architecture, opportunities, and open research challenges. *IEEE Access*, *5*, 5247–5261. https://doi.org/10.1109/ACCESS.2017.2689040

National Crime Agency (NCA). (2017). National Crime Agency - New assessment warns industry that cyber criminals are imitating nation state attacks. Retrieved July 14, 2017, from http://www.nationalcrimeagency.gov.uk/news/1043-new-assessment-warns-industry-that-cyber-criminals-are-imitating-nation-state-attacks

Nauroth, C. (2014). HDFS metadata directories explained. Retrieved October 29, 2017, from https://hortonworks.com/blog/hdfs-metadata-directories-explained/

NIST. (2013). *Glossary of key information security terms*. *National Institute of Standards and Technology Interagency or Internal Report* (Vol. NISTIR 729). https://doi.org/10.6028/NIST.IR.7298r2

Normandeau, K. (2013). Big data volume, variety, velocity and veracity. Retrieved August 31, 2017, from https://insidebigdata.com/2013/09/12/beyond-volume-variety-velocity-issue-big-data-veracity/

Oseku-Afful, T. (2016). *The use of big data analytics to protect critical information infrastructures from cyber-attacks*. Luleå University of Technology.

PandaLabs. (2016). Cybercrime reaches new heights in the third quarter. Retrieved July 14, 2017, from http://www.pandasecurity.com/mediacenter/pandalabs/pandalabs-q3/

Ponemon Institute. (2013). *Big data analytics in cyber defense*. Retrieved from http://www.ponemon.org/library/big-data-analytics-in-cyber-defense

Poole, D. L., & Mackworth, A. K. (2010). *Artificial intelligence - Foundations of*

Computational Agents. *Artificial Intelligence*. New York: Cambridge University Press. https://doi.org/10.1016/j.artint.2010.12.004

Rizvi, S., Labrador, G., Guyan, M., & Savan, J. (2016). Advocating for hybrid intrusion detection prevention system and framework improvement. *Procedia Computer Science*, *95*, 369–374. https://doi.org/10.1016/j.procs.2016.09.347

Rouse, M. (2012). What is attack vector? - Definition from WhatIs.com. Retrieved August 20, 2017, from http://searchsecurity.techtarget.com/definition/attack-vector

Rouse, M. (2015). What is Indicators of Compromise (IOC)? - Definition from WhatIs.com. Retrieved August 28, 2017, from http://searchsecurity.techtarget.com/definition/Indicators-of-Compromise-IOC

Rouse, M. (2017). What is big data analytics? - Definition from WhatIs.com. Retrieved September 2, 2017, from http://searchbusinessanalytics.techtarget.com/definition/big-data-analytics

RSA. (n.d.). *Intelligence driven threat detection*.

Russell, S., & Norvig, P. (1996). *Artificial intelligence - A Modern approach*. *The Knowledge Engineering Review* (Third Edit, Vol. 11). Pearson. https://doi.org/10.1017/S0269888900007724

Russom, P. (2011). Big data analytics. *TDWI Best Practices Report*, 1–35. https://doi.org/10.1109/ICCICT.2012.6398180

Seculert. (2014). *How to find and remove the attacker that has already passed through your traditional defenses*.

Spitzner, L. (2002). *Honeypots: Tracking hackers*. Addison Wesley. https://doi.org/10.1128/AAC.03728-14

Steve, M. (2016). *Donald Trump: "Cyber theft is the fastest growing crime in the United States by far" | CSO Online*. Retrieved from http://www.csoonline.com/article/3139973/cyber-attacks-espionage/donald-trump-cyber-theft-is-the-fastest-growing-crime-in-the-united-states-by-far.html

Tech Georgia. (2016). *2017 Emerging cyber threats, trends & technologies report*. Georgia. Retrieved from http://www.iisp.gatech.edu/sites/default/files/documents/2017_threats_report_final blu-web.pdf

The Apache Software Foundation. (n.d.). Apache Spark™ - lightning-fast cluster computing. Retrieved October 30, 2017, from https://spark.apache.org/

The Apache Software Foundation. (2013). HDFS architecture guide. Retrieved October 29, 2017, from https://hadoop.apache.org/docs/r1.2.1/hdfs_design.html

The Apache Software Foundation. (2017). Apache Hadoop 2.8.2 – MapReduce Tutorial. Retrieved October 30, 2017, from https://hadoop.apache.org/docs/stable/hadoop-mapreduce-client/hadoop-mapreduce-client-core/MapReduceTutorial.html

The Foundation Apache Software. (2014). Welcome to Apache™ Hadoop®! *Innovation*. Retrieved from http://hadoop.apache.org/index.pdf!

Totah, D. (2016). What is a Cyber Threat: How to Explain Cyber Threats to Your CEO. Retrieved August 19, 2017, from https://www.threatconnect.com/blog/how-to-explain-what-is-a-cyber-threat/

U.S. DHS. (2008). *DHS [Department of Homeland Security] Risk Lexicon*.

UCI. (1999). KDD Cup 1999 Data. Retrieved November 1, 2017, from http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html

US Government. (n.d.). *Protecting your networks from ransomware*. Retrieved from https://www.fbi.gov/file-repository/ransomware-prevention-and-response-for-cisos.pdf

vArmour. (2016). *Beginner's Guide to cyber deception*.

Veysset, F., & Butti, L. (2006). Honeypot technologies 2006 First Conference / tutorial. In *France Telecom R&D First Conference*.

Virvilis-Kollitiris, N. (2015). *Detecting advanced persistent threats through deception techniques*. Athens University of Economics and Business, Greece.

Virvilis, N., Serrano, O., & Dandurand, L. (2014). Big data analytics for sophisticated attack detection. *ISACA Journal*, *3*.

Wang, L., & Alexander, C. A. (2015). Big data in distributed analytics, cybersecurity, cyber warfare and digital forensics, *1*(1), 22–27. https://doi.org/10.12691/dt-1-1-5

Wang, L., & Jones, R. (2017). Big data analytics for network intrusion detection : A Survey. *International Journal of Networks and Communications*, *7*(1), 24–31. https://doi.org/10.5923/j.ijnc.20170701.03

Wert, M. (n.d.). Cyber threats: Definition & Types | Study.com. Retrieved August 18, 2017, from http://study.com/academy/lesson/cyber-threats-definition-types.html

Xu, G., Zong, Y., & Yang, Z. (2013). *Applied data mining*. https://doi.org/10.1007/s13398-014-0173-7.2