



A CONCEPTUAL FRAMEWORK FOR MITIGATING THE INCIDENCE OF CYBERATTACKS ON FINANCIAL INSTITUTIONS

MOHAMED KAILONDO BANYA

18/PGC/SCI01/1007

**DEPARTMENT OF MATHEMATICAL AND PHYSICAL SCIENCES
(COMPUTER SCIENCE PROGRAMME),
COLLEGE OF SCIENCES,
AFE BABALOLA UNIVERSITY, ADO-EKITI.**

ABSTRACT

This study observed that there is a rising incidence of cyber-attacks against financial institutions and all institutions in general. The study also observed that subsisting frameworks (Computer architecture), for mitigating or checking this rising incidence of attacks has some drawbacks. To address these drawbacks, this study analyzed some of these frameworks and isolated some of the observed drawbacks. The study then developed a conceptual framework that addressed the core problems of data preprocessing, data training (encoding) and visualization issue into consideration. The testing of the proposed framework showed that the core drawbacks observed were addressed and the proposed framework was easy to use and produces valid results.

CHAPTER ONE INTRODUCTION

1.1 Background to the Study

The Internet has revolutionized society with more and more people connecting every day, and it is fast becoming a necessity of daily life and a mainstay for conducting day-to-day business. The continued growth in both network access and speed of network connectivity has facilitated a wide-spread adaptation by the world at large. While the growth of the Internet continues to enable breakthrough innovations and life-changing benefits to society, it's also opens the possibility for adversaries to conduct malicious activities in the digital arena. These activities are made possible through the availability of widespread free protocol information, increased general technology awareness, and the current global security situation, has in itself made cyber-attacks easier and more likely to be launched (Nazir, et al., 2018).

While this is true, the aim of most of these attacks is to leverage the connectivity of societies and institutions through the Internet to carry out a malicious goal as evidenced by the steadily increasing number of cyber related cases in recent years. These goals can vary from theft of intellectual property, denial of service, disruption of business, theft of Personally Identifiable Information (PII) or Payment Card Information (PCI), financial fraud, demanding a ransom (i.e., ransomware), destruction of physical property (e.g., Conflicker Warm), and other nefarious purposes (Cyber Security Breaches Survey, 2017).

Due to the opportunity existing for bad actors to conduct malicious activities, it is imperative that all cyber infrastructures be secure and protected from misuse. Among the many cyber infrastructure systems that exist (e.g., critical infrastructure, cyber-physical systems, Supervisory Control and Data Acquisition (SCADA) systems, etc.). This thesis focuses mainly on the protection of networks maintain and operated by financial institutions which has over the years become a lucrative target for hackers. So, in order to minimize this trend, a wide array of approaches have been developed over the years by researchers in order to stay one step ahead of their adversaries. Interestingly, the best overall approach for tackling such problems as promulgated by some researchers consists of a defense-in-depth strategy, whereby various security tools, techniques, and mechanisms are employed throughout an organization's ecosystem (Park and Calif, 2016).

It is commonly understood that there is no such thing as 100% security in any infrastructure. The aim instead is mostly channeled towards managing risk and reducing the surface area available for attack. Security is an intractable problem, and it is impossible to think of all the possible ways an attacker may break through the defense mechanism of financial institutions. As such, a preferred strategy as suggested by MIT Computer Science and Artificial Intelligence Laboratory (CSAIL), is to minimize the attack surface, and manage risk by employing a defense-in-depth strategy, (Pant et al; 2016). To this end, various techniques can be employed to reduce surface area that are mostly vulnerable to attack (e.g., by enforcing access control, multifactor authentication, network segmentation, and continuous patching), as well as reducing risk by deploying tools at various stages, from the exterior-facing network to the interior network, and down to the individual host-level workstations on the network.

Generally, as stated earlier, cyber-attacks are on the rise and financial institutions seems to be the trust of most of these attacks. This narrative, coupled with the interconnectedness of individual players, and the introduction of novel technologies further heighten the risk of largescale cyberattacks on financial institutions. Over the past decade, series of cyber related attacks on financial institutions has led to the loss of millions of dollars. According to a report published in the Journal of Internet Banking and Commerce by Tariq, (2018), he outlined a range of cyber related attacks on financial institutions which has resulted in the loss of millions of dollars. For e.g., in 2014, USA Today reported, as captured by Tariq, (2018), that “Federal officials warned companies that hackers had stolen more than 500 million financial records over the past 12 months, essentially breaking into banks without ever entering a building. Also, in 2016, another news reports that Forty-six major financial institutions were targeted with distributed denial of service (DDoS) attacks in which hackers gained remote control of hundreds of computers and servers and use them to flood a target’s server with data, clogging it up so that it can’t receive legitimate traffic. In another instance, a news report by Crime Russia, as highlighted by Tariq, (2018), assert that hackers from the Lurk team, which created the banking Trojan of the same name, were able to steal more than 1.7 billion rubles (\$28.3m) from the accounts of Russian Banks before been detained by the Interior Ministry. In the same vein, Kaspersky Lab further asserts that, in one way or another, the criminals from the Lurk team, stripped each victim bank of \$2.5 million to the tune of 10

million dollars. In 2017, Hongkong and Shanghai Banking Corporation Limited (HSBC), which happens to be one of the largest banks in the world suffered from a cyberattack in early 2017. A report from The Week Newsletter, as captured by Tariq, (2018), stated, “HSBC customers were unable to access online banking services for the second time in a month, in the wake of an apparent cyberattack. Similarly, in February 2016, media report suggested that hackers had breached the network of the Bangladesh Central Bank and sent thirty-five fraudulent transfer requests to the Federal Reserve Bank of New York, totaling nearly \$1 billion (Herman, 2016), and even though only four of the fraudulent requests succeeded, the hackers were able to transfer \$81 million to an account in the Philippines, representing one of the largest bank thefts in history.

Similarly, SC Magazine UK recently reported that the Russian Central Bank revoked the licenses of three Russian Banks in 2015 because an investigation which uncovered evidence that current and former employees of the bank had engaged in various forms of cyber related attacks to withdraw money from the accounts of their own clients, as well as cover up other crimes and violations committed by the bank (Gerden, 2016). The Russian Central Bank reported that in the last quarter of 2015 alone, more than \$20 million was stolen from the accounts of clients with what the central bank suspects was the knowledge or direct participation of the banks themselves. The central bank also reported that the hacks were likely the result of huge financial cuts that left disgruntled former employees of the bank willing to collaborate with hackers in carrying out their malicious goal.

Some of these trends are not completely new because where the money is has always been the focus of most of these attacks even before the online regime of banking came into being. The revolution of online banking just made it a whole lot easier and simpler, and the attack on financial institutions continue to escalate. For this reason, according to Lafleur et al., (2015), over a period of three months, more than a dozen men posing as employees of a synthetic grass company dug approximately 656-foot-long tunnel, 13 feet under Fortaleza, Brazil, in order to access the vault of the local branch of the Brazil Central Bank. The thieves transferred over 7,700 lbs. of cash worth \$81.9 million dollars out of the bank through their elaborate electrically-lit, air-conditioned, and structurally reinforced tunnel. Similarly, in 1976, several men under the leadership of Albert Spaggiari, a French army paratrooper-turned-criminal, as reported by Lafleur et al.,

(2015), finished a two-month job of tunneling 60 feet from the city sewers into the underground vault of the Société Générale Bank in Nice, France. In a heist lasting 36 uninterrupted hours, the criminals, including an appraiser to identify the most valuable items, stole \$40.4 million in contents from 400 of the 4,000 safe deposit boxes within the bank's vault, as well as from the bank's own supply of cash and gold. Furthermore, on the weekend of June seventh 1986, a group of unknown size criminals completed a month-long goal as they successfully broke into the First Interstate Bank of Los Angeles vault. The robbers, guesstimated by police to probably consist of only two men, entered into the vault via a tunnel they had created by drilling underneath through the ground above the Los Angeles sewer system. The vault was time locked, and could not be accessed quickly by any bank employees without drilling into the vault themselves. The bank manager, apparently in the building as the robbery was ongoing, could only feebly hold the phone to the vault door as he tried to show the men at the security offices that something strange was happening. Powerless to stop whatever was making the suspicious noise, the bank was looted to the tune of \$172,000 (Lafleur et al., 2015).

Attempts at resolving some of these crises has brought about the use of different approaches by researcher, and one of the most effective ways to protect the confidentiality, integrity, and availability of information and enterprise systems once an attacker has compromised its defense mechanism is to deploy Intrusion Detection Systems (IDS). Intrusion Detection Systems as defined by the National Institute of Technology (NIST) is a "software or hardware systems that automates the "process of monitoring the events occurring in a computer system or network and analyze them for signs of possible incidents, which are violations or imminent threats of violation of computer security policies, acceptable use policies, or standard security practices". These systems use techniques to measure the difference between input events and signatures of known bad intrusions. If the input event shares patterns of similarity with known bad intrusions, then the systems flag those events as malicious. (Lundy, 2018; Niyaz et al; 2015).

However, according to Dong and Wang, (2016), one of the major challenges in network security is the provision of a robust and effective Intrusion Detection System (IDS). Despite the significant advances in NIDS technology, the majority of solutions still operate using less-capable signature-based techniques, as opposed to well advanced and elaborate anomaly detection techniques. There are several reasons for this, including the high false error rate (and associated costs), difficulty in obtaining reliable training data, longevity of training

data and behavioral dynamics of the system. The current situation will reach a point whereby reliance on such techniques leads to ineffective and inaccurate detection. The specifics of this challenge are to create a widely-accepted anomaly detection technique capable of overcoming the limitations induced by the ongoing changes occurring in modern networks.

Interestingly, thank God we are now in the era of data revolution, and it seems to hold some promise especially with the advent of data mining, deep learning and machine learning – that designate the process of extracting useful information from large databases; possible solutions to the problem seems possible. Alan Turing in his seminal 1950 paper “Computing Machinery and Intelligence” (Turing, 1950) concluded that general-purpose computers could learn and be capable of originality. This opened questions of whether computers could learn on their own to perform specific tasks - can computers learn rules by looking at data, instead of having humans input the rules manually? These questions gave rise to the subfield of machine learning. It is worth noting that, machine learning algorithms are learning algorithms that learn and adjust from data, and instead of manually programming the computer and telling it explicitly what to do, machine learning algorithms enables the program to learn what output to produce based on examples and data. By learning based on examples and data, this allows the computer to make decisions and perform tasks on new inputs it has never seen before. This approach through supervised or unsupervised learning mechanism has the capability to detect erroneous network traffics and filter them out accordingly (Deng, 2014).

While this process is true, the concept of deep learning has gained popularity in recent years, and different models for network intrusion detection has been developed and are continuously being tested in different areas in trying to mitigate the incidence of cyber related attacks on network infrastructures. According to Goodfellow et al., (2016), Deep learning, is a subfield of machine learning that excels in generalizing to new examples when the data is complex in nature and contains a high level of dimensionality and it emerged from cognitive and information theories and human neurons with learning processes along with strong interconnection structures between neurons. A key features of computing neurons and neural network models is - it can be applied to generic neurons of any type of data and then learn from it comprehensively. Hence, deep learning is considered as a promising avenue of research as it is capable of

automatically identifying complex features at a high level of abstraction. It is all about learning multilevel representations and abstraction, which is useful for data, such as image, text and sound. One of the exceptional deep learning characteristics is its capability of using unlabeled data during training process (Schmidhuber, 2015). Fundamentally so, each neuron with activation function is considered as the single logistic node, which is connected to the input in the next layer of its loss function and is calculated to modify the weights at each neuron and optimize it to make it suitable for input data. Each layer of a neural network inputs multiple neurons that initiates with dissimilar weights and try to learn on the input data concurrently. Thus, in multiple layers with multiple nodes, each node learns from the output of the previous layers, and gradually decreases the approximation of the real input data to provide accurate output representation set. Another interesting aspect of deep learning is that it uses machine learning to discover not only the mapping from feature representations to output but also the feature representations themselves, and it does this by learning successive layers of increasingly meaningful features (Nielsen, 2015).

Deep learning is such a novel methodology currently receiving much attention, and it describes a family of learning algorithms rather than a single method that can be used to learn complex prediction models, (LeCun et al., 2015). For this reason, deep learning models has been successfully applied to several application areas in trying to mitigate the incidence of cyber related attacks on network infrastructure systems. For instance, Gao et al., (2014) used a Deep Belief Network (DBN) to develop an intrusion detection system, and after testing their parameters with four hidden layers (six layers total), beating other Support Vector Machines (SVM) and DBNs with fewer layers, they were able to obtain an accuracy of 93.49%, with a True Positive Rate (TPR) of 92.33%. Nguyen et al., (2018) also achieved similar accuracy using a similar architecture. Alrawashdeh and Purdy, (2015) performed a similar experiment with a four-hidden-layer of DBN and achieved an accuracy of 97.9%. Alom et al., (2015) built a similar model, but were able to achieve 97.5% accuracy, training on 40% of the data. This outperformed previous SVMs and DBNs followed by an SVM classifier. Dong and Wang used a Restricted Boltzmann Machine (RBM), with a less sophisticated architecture than that used by Gao et al., (2014), and Alom et al., (2015) and achieved worse results although they found that the ability to detect attacks was highly dependent on the type of attack, ranging from 82% to 41% accuracy. Li et al., (2015) used an autoencoder to reduce the

dimensionality of the data, followed by a DBN with RBM layers that achieved a TPR of 92.2% with a False Positive Rate (FPR) of 1.58%. Yousefi-Azar et al., (2017) also used an autoencoder with four hidden layers, followed by a Gaussian naive Bayes classifier and achieved an accuracy of 83.34%. Alom and Taha (2017) implemented both autoencoder and RBMs to perform dimensionality reduction on the KDD-1999 dataset, reducing it to nine features, and then performed K-means clustering on the data, achieving detection accuracies of 91.86% and 92.12% accuracy, respectively.

Wang et al., (2017) built an intrusion detection algorithm using raw network traffic data from two existing datasets: the CTU-13 dataset and the IndeX-based Integration Approach (IXIA) dataset (that the authors called the USTC-TFC 2016 dataset), which contained 10 types of normal data and 10 types of malicious data and appeared to be relatively balanced between malicious and normal. A preprocessing step took the raw network traffic data and converted it into images, which were then fed into a Convolutional neural network (CNN) with a similar architecture to the well-established CNN LeNet-5, LeCun et al., (2015) and because there was no engineering of the preprocessing stage that produced the images, this method handled the raw data directly. The classification was done in two different ways. The first method involved a 20-class classifier, and the goal was to identify which type of normal or malicious the traffic was. The second was a binary classifier which fed into one of two CNNs trained to identify the type of malicious traffic or binary traffic. The 20-class classifier achieved an accuracy of 99.17%. The binary classifier achieved a 100% accuracy whereas the 10-class normal classifier achieved 99.4% and the 10-class malicious classifier achieved 98.52%.

Javaid et al., (2015) proposed a deep learning based IDS using sparse autoencoder layers, followed by several supervised SoftMax layers to develop two different models. The first model classifies network traffic as either normal or malicious. The second model classifies the network traffic as either normal, or one of four attack types. The sparse autoencoder part of the model contains two hidden layers. The output of this was fed into three SoftMax layers, and the result is a Deep Neural Network (DNN) with four hidden layers. The two-class classifier outperformed the five-class classifier: 88.4% accuracy versus 79.1% accuracy.

1.2 Problem Statement

Cyber-attacks are growing exponentially and poses a serious threat to the stability of the overall financial sector. These attacks continue to increase in number, scope, and sophistication, thus, making it almost impossible to predict their total impact. The Herjavec Group has predicted that the global annual cost of cybercrime is estimated to increase to around USD 6 trillion by 2021, from USD 400 billion in early 2015 (Park and Calif, 2016). In a similar report, it is estimated by organizations such as Juniper Research and the World Economic Forum that a single global cyberattack could result in damages of as much as USD 121 billion (Kass, 2017), and beyond financial loss, cyberattacks can disrupt business, financial markets and contribute to a broader loss of confidence.

As the modern world is rapidly becoming more digitalized, reliant on data and the Internet has become increasingly interconnected and sophisticated, and the impact of cyber related attacks has gradually touched on every facet of life. This was particularly evidenced during the global WannaCrypt ransomware attacks in May 2017 that affected more than 200,000 computers in at least 150 countries, including those found within hospitals, utilities, railways, telecommunications and automobile companies; as well as the June 2017 Petya ransomware that impacted computers within 64 countries. It is worth knowing that though the impact of the recent ransomware attacks on financial institutions has been mostly minimal, the financial services sector has traditionally been the largest target due to both the attractiveness of financial gain and access to confidential financial data. According to a report by IBM, the financial sector in 2016 alone was attacked 65% more often than any other sector, resulting in more than 200 million records been breached, a 93.7% increase over 2015 when just under a million were breaches (Boer and Vazquez, 2017).

Over recent years, the types of perpetrators of cyber-attacks have expanded and their skills and sophistication have significantly increased. These perpetrators could belong to hacking groups or to criminal gangs but they may also be state-sponsored as part of a broader and more powerful attempt to destabilize other jurisdictions by infiltrating their systems, i.e., communications systems and most especially financial systems. One such example is North Korea, which is alleged to have sponsored several attacks (Boer and Vazquez, 2017), including both the already mentioned WannaCrypt ransomware and the attempted heist of USD 1 billion from the Bangladesh Central Bank in 2016. Regardless of the actor, the tools used or the motives they

might have, any attack on critical components or services of the financial system, could have either direct or indirect impacts that could threaten the stability of the system, or of its respective participants.

During the last decade, series of different variants of deep learning and machine learning models have been employed in an attempt to solving some of the most devastating cyber related attacks on network infrastructures especially the financial sector. To date, the incidence of cyberattacks on these institutions remains an excruciating problem, and the need to mitigate it continue to grow among researchers. For this study, a framework for mitigating the incidence of cyberattacks on financial institutions using deep learning would be developed and hopefully helps to reduce the menace of cyber related attacks even before they occur.

1.3 Aim and Objectives

The aim of the study is to develop a conceptual framework for mitigating the incidence of cyber-attacks on financial institutions using deep learning (Autoencoder).

This would be achieved through the following objectives:

- i. Some subsisting frameworks were analyzed, and their drawbacks were subsequently highlighted.
- ii. A conceptual solution that addressed the challenges of denial-of-services was designed.
- iii. Using appropriate software tools and pertinent datasets, the design in 1.2 (ii) was implemented using Python framework.
- iv. The developed framework was evaluated on denial-of-service datasets.

1.4 The Scope of the Study

It is observed that the core of the attacks on financial institutions are always preceded by denial of service-related attacks. Denial of Service (DoS) and its variant, Distributed Denial of Service (DDoS), as opined by Aamir and Zaidi, (2013), are possible threats which exhaust resources to make it unavailable for the legitimate users, thereby, violating one of the security components – Availability. Consequently, the proposed system in this study would be limited to the detection of these denial-of-service attacks that are often used as a preload to other attacks.

1.5 Research Methodology

The study will adopt the following set of research tools:

i. Literature Search

Literature search will be used to understand the various types of cyber-attacks on financial institutions and the mechanism for perpetrating such attacks. Literature search will also be used to understand the current solutions to mitigating such attacks till date and the various drawbacks of each of the approaches. The design and implementation of the proposed system will also rely extensively on the knowledge gained from the literature.

ii. Consultations

Consultations would be held extensively with experts in financial related cyberattacks and in cyber-attacks in general. The approaches to be used to achieve the said objective of the study would be discussed extensively with appropriate experts (network experts, system analysts and system developers).

iii. Training

Relevant topics related to this research area will be undertaken (Deep Learning). This is an emerging area and different tools, and applications are available for design and implementation. These various tools and how they are to be used to achieve such objectives will be acquired through extensive training.

iv. Framework Development/Simulation

Preliminary investigation on acquiring the data for the proposed system clearly indicates that real-life data would be hard to come by. For this reason, based on the knowledge gained, the proposed system would be developed using the acquired datasets and the selected system development environment (Autoencoder). It is this data that would now be used to train the proposed system using deep learning (Autoencoder). The trained system would then be tested for its capability to perform the intended functionality by subjecting the various schemes incidental to financial institutions.

1.6 Limitation to the Study

The study could not access real-life data owing to the policies of the institutions (financial institution).

1.7 Organization of the Study

The report is organized in five chapters following the introductory one which comprises of the core concepts surrounding the topic being discoursed. In Chapter 2, a review of related work together with the explanation of the technical background is presented, thus, the reading is suggested to the understanding of the core concepts and definitions in the field of cybersecurity and deep learning. Chapter 3 is dedicated to the outline and deeper explanation of the methodology followed during the research work, serving as a more detailed description of the various tools and approaches that were adopted in the design and implementation choices. Chapter 4 is dedicated to the detailed description of the system implementation, testing and discussions of the core frameworks and processes adopted during the research. Finally, the last chapter, Chapter 5, is dedicated to conclusive considerations and recommendations of this study.



CHAPTER TWO

LITERATURE REVIEW

2.0 INTRODUCTION

This chapter contains an exposition on the field of cybersecurity, artificial intelligence, and deep learning and how they relate to the problem of network intrusion detection. In addition, neural networks and the various training mechanism that are mostly adopted in mitigating the incidence of cyber-attacks is discussed. Lastly, autoencoder, which is used as the core design and implementation tool, with the general application metrics acceptable in the field of intrusion detection is discussed.

2.1 The Concepts of Cyberspace and its Definitions

Within the realm of security studies there is still significant debate over the definition of cyberspace. Instead, the aim within the field is to attempt to create uniformity in the definition so it can relate to other key definitions such as land, sea, air or space power. The challenge in defining cyberspace is that cyberspace today is fundamentally different to what was experienced when it was first created (Sheldon, 2013; McCarthy, 2013).

The idea of cyberspace began when the United

States Department of Defense started the early computing networks which became known as the Advanced Research Projects Agency Network (ARPANET) in 1968. This was later the network which underpinned the internet. However, since 1968 there have been repeated attempts to provide a definition for cyber space as the term's meaning has gradually expanded over time (Singer and Friedman, 2014). In order to demonstrate the complexity of the term, this research study will present and then discuss three definitions.

In the latest attempt to define cyberspace, the Pentagon released the following definitions:

- ✦ Cyberspace is the global domain within the information environment consisting of the interdependent network of information technology infrastructures, including the internet, telecommunication networks, computer systems and embedded processors and controllers (US Department of Defense, 2013).
- ✦ In Information Warfare and Security, Denning (1999) defines cyberspace as “the information space consisting of the sum total of all computer networks”.
- ✦ Lastly, a popular definition is the one provided by Kuehl (2009) stating that cyberspace is: A global domain within the information environment whose distinctive and unique character is framed by the use of electronics and the electromagnetic spectrum to create, store, modify, exchange, and exploit information via interdependent and interconnected networks using information-communication technologies.

In order to incorporate these ideas, cyberspace can be defined as a virtual information environment where digital data can be stored, created and shared (Rudner, 2013; Giles and Hagestad, 2013). However, cyberspace also encompasses actual hardware such as the computers that store the data and the actual

infrastructure that can connect these computers together such as wired and wireless networks (Rosenzweig, 2013). The notion of virtual space is important because while information within the cyber domain can be stored on physical hardware, it can also be held in a virtual non-physical space which is highly interconnected. As Denning (1999) defines it, it is the “sum total of all computer networks”. Significantly, cyberspace also comprises of a human element, the actual users behind the computers. This is important as it is human action which shapes cyberspace. Thus, when considering all these different elements, it is easy to understand how cyberspace is constantly evolving because it is made up of the nexus of technology and human interaction which changes at a rapid rate (Borghard and Lonergan, 2017; Tamkin, 2017).

2.2 Cybersecurity Concepts and its Definitions

Despite the threat of viruses and malware almost since the dawn of computing, awareness of the security and sanctity of data with computer systems didn't gain traction until the explosive growth of the internet, whereby the exposure of so many machines on the web provided a veritable playground for hackers to test their skills – bringing down websites, stealing data, or committing fraud; something we now call cybercrime. Since then, and with internet penetration globally at an estimated 3.4 billion users (approximately 46% of the world's population), the opportunities for cybercrime have ballooned exponentially (ACS, 2016). Thus, combating this is a multidisciplinary affair that spans hardware and software through to policy and people – all of it aimed at both preventing cybercrime occurring in the first place, or minimizing its impact when it does. This is the practice of cybersecurity (ACS, 2016).

According to Donnelly, (2018), cybersecurity is a constantly evolving, constantly active process just like the threats it aims to prevent. While what frequently makes the news are breaches of user accounts and the publication of names and passwords – it's often financial gain, or the theft of critical business or government intelligence, that drives the cyber underworld. One fact remains clear: it's only going to increase. As we integrate technology further into our lives, the opportunities for abuse will keep growing. So too, then, must the defenses we employ to stop them through the education and practice of cybersecurity.

Cybersecurity as a concept only really emerged in the 1990s in the post-Cold War era as computer advances in technology began to have serious geopolitical consequences. Initially, the term only highlighted

the challenges facing network computers and how hardware could be corrupted. Later it moved away from this technical understanding as scientists argued that the threats emerging through new technologies could be harnessed in such a way that they could have serious societal impact and cause direct harm to institutions, and processes (Rosenzweig, 2013; Donnelly, 2018)

Over the years, cybersecurity has played a crucial role in the way government, the military, corporate, financial, and medical organizations collect, process, and store unprecedented amounts of data on computers and other devices. A significant portion of such data can be sensitive information, whether that be intellectual property, financial data, personal information, or other types of data for which unauthorized access or exposure could have negative consequences (Yang et al., 2017). Organizations transmit sensitive data across networks and to other devices in the course of doing businesses, and cyber security describes the discipline dedicated to protecting that information and the systems used to process and store it. Thus, cyber security is both about the insecurity created by and through this new place, and or space and about the practices or processes to make it more secure. Cybersecurity is “the process of implementing and operating controls and other risk management activities to protect information and systems from security events that could compromise them and, when security events are not prevented, to detect, respond to, mitigate against, and recover from those events in a timely manner” (Perlroth et al., 2017; CISCO, 2017). According to Yang et al., (2017) in their report opined that cyber security involves activities or processes, ability or capability, or state whereby information and communications systems and the information contained therein are protected from and/or defended against damage, unauthorized use or modification, or exploitation. The same source also asserted with a more elaborate definition of cyber security as the “Strategy, policy, and standard regarding the security of and operations in the cyberspace, and encompassing the full range of threat reduction, vulnerability reduction, deterrence, international engagement, incident response, resiliency, and recovery policies and activities, including computer network operations, information assurance, law enforcement, diplomacy, military, and intelligence mission as they relate to security and stability of the global information and communications infrastructure” (Yang et al., 2017).

Furthermore, the International Telecommunication Union (ITU) defined cyber security as the collection of tools, policies, security concepts, security safeguards, guidelines, risk management approaches, actions, training, best practices, assurance and technologies that can be used to protect the cyber environment and organization and user's assets including connected computing devices, personnel, infrastructure, applications, services, telecommunications systems, and the totality of transmitted and/or stored information in the cyber environment (ITU, 2011). Cybersecurity strives to ensure the attainment and maintenance of the security properties of organizations and user's assets against relevant security risks in the cyber environment (ITU, 2011). Li et al., (2019) further elucidated cybersecurity as a means not only of protecting and defending society and its essential information infrastructures but also as a way of prosecuting national and international policies through information-technological means. This highlights cybersecurity's ontological and processual characteristics and its contingent relations with information technologies, particularly the Internet (Li et al., 2019).

Cybersecurity in general has attracted much attention in recent years. Both the general public and the business world are concerned about the growing cybercrimes that expose sensitive personal information, cause business disruptions, or steal trade secrets. Span et al., (2018) reports that the average number of detected cyber incidents increased 38% and the theft of "hard" intellectual property increased 56% in 2015 compared to 2014. More than 20% of the breached firms experienced substantial loss of revenue, customer base, and business opportunities, and most of the breached firms spent millions of dollars improving defense technologies and expanding security procedures following the attacks (Valeriano and Maness, 2015; CISCO, 2017).

Cybersecurity and information security are often used interchangeably. The Cybersecurity Working Group of the Assurance Services Executive Committee AICPA (AICPA, 2018), defines cybersecurity as "the process of implementing and operating controls and other risk management activities to protect information and systems from security events that could compromise them and, when security events are not prevented, to detect, respond to, mitigate against, and recover from those events in a timely manner." The committee further defines cybersecurity compromise as "a loss of confidentiality, integrity, or availability of information,

including any resultant impairment of (1) processing integrity or availability of systems or (2) the integrity or availability of system inputs or outputs, which have a negative effect on the achievement of the entity's business objectives and commitments, as well as the laws and regulations related to cybersecurity risks and the cybersecurity program." The underlying premise is that "all firms that operate in cyberspace will suffer a security event or breach at some point in time." The assumption is supported by Ransbotham and Mitra (2018), who provide empirical evidence that all systems are potential victims of cyberattacks. Firms not intrinsically attractive to attackers are not immune from attacks.

2.2.1 Cybercrime and its Impacts

According to Le et al., (2018), "Crime is not only observed in most societies of a particular species, but in all societies of all types" concluding that "a society exempt from it is utterly impossible". In modern times, a crime is the violation of a criminal statute. These statutes can be found in criminal and other legal codes and usually cover offenses against persons, property, public safety, and financial arrangements. These can be legal codes for an entire nation or for a section of a nation such as a state within the United States or prefecture within Japan (Le et al., 2018; Valeriano and Maness, 2015).

Cybercrime is not only committed in the financial institution and that has given it broad definition to include the diversity of criminal activities related to information communication technologies. One of such definition was asserted by, Patri et al., (2018), who argues that, cybercrime are abuses and misuses of computer systems or computers connected to the Internet, which result in direct or indirect losses.

The ECT Act, (2002), describe Cybercrime as unauthorized access to, interception of or interference with data, computer-related extortion, fraud and forgery, aiding and abetting cybercrime. It is an increasingly information traverses within cyber arteries powered by information and communication technologies. With the rise of the Internet as a platform to share information and conduct business online, the world has never been as connected as it is today (Rosewarne, 2013; Hoffman, 2013; Borghard and Lonergan, 2017).

Cybercriminals all over the world use computers and internet as apparatus to commit financial and other crimes, Toppa, (2017) defined cybercrime as computer mediated crime conducted through global

electronic networks which are either illegal or considered illicit by certain parties. Wall (2001) cited in Boateng et al., (2011), argue that, cybercrimes are committed in the banking sectors using online technologies to illegally remove or transfer money to different account.

According to Anderson et al., (2012), Cybercrime can be categorized into four major divisions, cyber-deceptions, cyber-pornography, cyber-violence and cybertrespass, they added that, banking frauds are sub-category in cyber-deception which can be defined as immoral activities including stealing, credit card fraud, and intellectual property violations, like Automatic Teller Machine frauds and Cyber Money Laundering. Cybercriminals exploit numerous vulnerabilities in cyberspace to commit these acts. Rosewarne (2013), claim that, cybercrime is commonly understood as involving an attack on the confidentiality, integrity and accessibility of an entity's online computer presence or networks. Academic and policy experts around the globe have defined cybercrime in a variety of ways.

- ✦ The Federal Chancellery of the Republic of Austria (2013), defined cybercrime as illegal attacks from cyberspace on or through ICT (information and communication technology) systems, which are defined in penal or administrative laws. The term therefore covers all criminal offences committed with the aid of information technologies and communications networks and also encompasses Internet crime.
- ✦ Qatar: Misconduct or crime committed using technology. Examples of cybercrime may include illegal access to systems or information, fraud, identity theft, or content-related offenses such as spam (Ministry of Information and Communications Technology, 2014).
- ✦ New Zealand: Cybercrime is a criminal act that can only be committed through the use of ICT or the Internet and where the computer or network is the target of the offence. This is regardless of what the criminal goal is – whether political or financial gain, espionage or any other reason. It was further buttressed that cyber-enabled crime is any criminal act that could be committed without ICT or the Internet, but is assisted, facilitated or escalated in scale by the use of technology (New Zealand Department of the Prime Minister and Cabinet, 2015).
- ✦ United Nations: Cyber-dependent crime requires an ICT infrastructure and is often typified as the creation, dissemination and deployment of malware, ransomware, attacks on critical national

infrastructure...and taking a website offline by overloading it with data (a DDOS attack). Cyber-enabled crime is that which can occur in the offline world but can also be facilitated by ICT (United Nations Office on Drugs and Crime n.d.).

- ✦ Council of Europe: Described cybercrime as any action directed against the confidentiality, integrity and availability of computer systems, networks and computer data as well as the misuse of such systems, networks and data (Council of Europe, 2001).

These definitions suggest a wide variety of types and levels of cybercrime. Although cyberspace can facilitate conventional, offline crimes such as identity theft, transaction fraud, and illegal drug sales, there are a number of common cybercrimes that occur only or primarily because of infrastructure in the cyber world. In the most basic form of cybercrime, unauthorized intrusions include “hacking,” accessing a computer system to gain knowledge about how the system works with no intent to cause damage or disruption, and “cracking,” accessing a computer system with the intent to cause damage or disruption. Ransomware is a form of extortion that uses malware to make a computer or its contents inaccessible until the victim pays a ransom. Digital piracy involves stealing computer software and other intellectual property stored in digital form such as movies and music. Cyber industrial or economic espionage targets firms’ trade secrets for the benefit of a foreign entity or anyone other than the owning firms. Online child sexual exploitation takes place in social media, chat rooms, child pornography websites, online groups, peer-to-peer file sharing, and bulletin board systems (Melendez, 2017; Toppa, 2017; Rutenberg, 2017; Saxena and Soni, 2018; Yang et al., 2017; Zhou et al., 2018).

Cybercrime has gained momentum over the years and has led the disruption of businesses, which translates to the loss of millions of dollars. In 2017, hackers exploited a website vulnerability to steal private information - including addresses, birthdates, social security numbers, driver’s license numbers, and tax IDs - on more than 145 million individuals from Equifax, a credit reporting agency (Fung, 2018; Singer and Brooking 2018). That same year, hackers infected the UK’s National Health Services (NHS) computer system with ransomware named “WannaCry.” The malware was delivered by email and locked down as many as 300,000 NHS computers with the hackers demanding

money to unlock the infected units. In 2015, one group of hackers stole private information that yielded more than \$900 million from more than 100 banks around the world using malware and impersonating banking staff to make fraudulent transfers. Another group stole personal information from millions of J. P. Morgan Chase customers and sold it to a network of criminal associates. This group also accessed company related information that allowed them to trade profitably in the financial markets on what amounts to insider information. In 2014, hackers believed to be associated with North Korea stole sensitive and proprietary data from Sony Pictures and in an attempt to blackmail the entertainment company into cancelling the release of a movie that cast the country's leader in a negative light (Valja et al., 2017; Sayfayn and Madnick, 2017; Siegel and Tucker, 2018).

There are some contention and ambiguity around exactly what activities fall under the cybercrime definition with respect to banks, Bosco (2012), propagate that, the term cybercrime includes traditional crimes conducted through the Internet such as Identity theft (Personal Information), Financial fraud (Financial Information), Hacking (E-commerce, e-banking, Credit Processing Cents, Spam, Counterfeiting Gambling (Money laundering), Pornography, Cyber stalking, Cyberbullying, Cyber grooming.

Therefore, the deduction from the literature suggests that, cybercrime is a computer mediated crime conducted through internet to manipulate Personal Information, to enable funds to be transfer from customer account to another account without the knowledge of accounts holder. Cybercrime Many also implies any conduct proscribed by legislation or jurisprudence that:

- i. Is directed at computing and communications technologies at the financial institution themselves;
- ii. Involves the use of digital technologies in the commission of the offence in the banks; or
- iii. Involves the incidental use of computers with respect to the commission of other crimes.

2.2.2 Current Cybersecurity Defense Mechanism

The internet is inherently insecure, and the cause of the problem can be tracked down to three issues. First, the internet's creators failed to fully contemplate security. Vinton Cerf, one of the original designers of

the internet, admitted, “We didn’t focus on how you could wreck this system intentionally” (Timberg, 2015).

To the internet’s architects, the dominating security principle was survivability in the event of military action, not the most crucial defense policy

“Confidentiality, Integrity and Availability” sometimes referred to as the CIA triad (Chia, 2012). The technologies to build in security were not available or sufficiently mature, in part due to limits in computational power available at the time and export controls on enabling technologies, i.e., encryption. As the designers defined the seven layers of the TCP/IP network stack, the primary objective was reliability. Initial engineering efforts were focused on getting the technology working, not the assurance of the CIA triad. This is why it is often said security was “bolted on” to the internet after the fact (Timberg, 2015).

Second, vulnerabilities are routinely introduced into every layer of the cyber ecosystem, and can never be entirely eliminated. Software bugs are defects in how a program was designed to operate, resulting in software behaviors that were not anticipated by the designer, and hackers seek to exploit bugs to actively circumvent how a program was designed to operate. In a Department of Homeland Security (DHS)- funded analysis, Coverity, Inc. found an error rate of 0.434 defects per thousand lines of code in a broad range of open-source software projects. Each of the TCP/IP layers requires programming, whether implemented in software or hardware. Bugs are fixed over time, whether before or after programs are released. However, it is unlikely that programmers will be able to radically reduce future introduction of bugs in new programming efforts. Two factors drive this: the pace of competition to produce and sell new features, and rapid advancement of underlying hardware upon which programs run (Timberg, 2015; Zhou et al., 2018).

The third fundamental internet security issue is that cyberattacks are a cat-and-mouse game, they are perpetrated and defended by humans, acting and counteracting each other so that one can gain an advantage over the other. Those involved are characterized as “black hats,” “white hats,” and “gray hats.” Black hats work to compromise the CIA triad with malicious intent, while white hats work to ensure the CIA triad, particularly for devices they are charged to protect. Gray hats refer to those who actively work to compromise the CIA triad, perhaps to include conducting activities that have been criminalized, but without malicious intent. Ultimately, the human adversarial dynamic makes it difficult to predict the manifestation of future

exploits, which makes it more difficult to defend against them (Timberg, 2015; Zhou et al., 2018; Yang et al., 2017).

2.2.2.1 Defensive Models

Network defenders have designed various models, tools, and techniques to help mitigate a hostile environment in which vulnerabilities are exploited by cyber threats actors. The most prevalent models are presented in this section. These models are not mutually exclusive, and are often used in combination. No model can fully protect a network. Thus, the goal of the defender is to reduce risk to an acceptable level at an acceptable cost (Schlein, 2014).

i. Network Boundary Control

According to Schlein, (2014), the network boundary control model is built upon the assumption that a cyberattack could originate from outside the defended network. Therefore, the simplest mechanism to protect the network would be one that enforces a secure border separating the internal network from the external internet, i.e., outbound network traffic. In this model, devices with external network connections are identified, and robust security controls are placed on them. This puts the cybersecurity focus on any and all devices reachable from the internet. The model does not, however, defend devices and services within the network that are not directly accessible from the internet. Such an approach has been described as being like M&M candies, hard on the outside and soft on the inside.

Cybersecurity professionals have largely discounted this approach as a standalone model. A major flaw of this approach is the basic assumption that threats originate from outside an organization. The model is particularly vulnerable to insider threats; those who have authorized access to the defended network but exceed their scope of permissions for an unauthorized purpose.

Once a malicious actor breaches the external defenses, the interior network is left unguarded and vulnerable. Another challenge to this model is that modern devices frequently combine plug-and-play configuration with multiple network interfaces, potentially opening holes in the wall unknown to defenders. Despite these flaws,

the network boundary control model remains commonly used as a building block within network security architectures to reduce the network surface area directly accessible from the internet (Schlein, 2014).

ii. Defense-in-Depth

Defense-in-depth was promulgated to overcome the single point of failure of the network boundary control model, and was conceptualized using the military principle of weakening an adversary by delaying an attacker's advance through the ceding of defended territory. The cyber correlation is to deploy multiple forms of layered defenses, each requiring time and effort for an adversary to defeat, and to give defenders more time to recognize and then mitigate an attack. It provides for information assurance by making cost-effective security investments focused on people (users), the technology of the system, and the system's operation (Schlein, 2014).

This approach raises the cybersecurity bar by acknowledging malicious events could originate from anywhere, even inside the organization. It also introduced a strategy toward managing security solutions over time, and is considered a best practice by many security professionals. However, the approach has been criticized because historic military advantages from defense-in-depth have not been realized within cyberspace. Specifically, attackers have not been weakened; rather, they are attacking and succeeding at higher rates, and delays from defensive layering have not significantly increased the amount of effort necessary for an attack to succeed (Hirschmann, 2014; Schlein, 2014).

iii. Continuous Monitoring

Schlein, (2014), opined that, continuous monitoring model assumes a network will not remain in a healthy state and it is therefore necessary to continuously review for faults. It further assumes devices and programs are designed to provide robust diagnostic information that can be logged and analyzed. Investment is focused on collecting and analyzing information from critical systems and network segments, identifying concerns, and alerting for further review and potential remediation.

A monitoring and logging guide authored by CREST, a U.K.- based non-profit cybersecurity organization, outlined a framework and process toward implementing continuous monitoring. The framework below

illustrates common logging mechanisms within a network and how they are analyzed (Creasey, 2015). Such logs exist at the server, network, application, and security suite level. The seven-step process is to: 1) develop a monitoring and logging plan; 2) identify and address your cybersecurity posture outside of logging and monitoring; 3) identify sources of security indicators; 4) develop people, processes, and practices to monitor and log; 5) buy or build monitoring and logging solutions; 6) integrate solutions into security architecture; and 7) maintain the capability (Creasey, 2015).

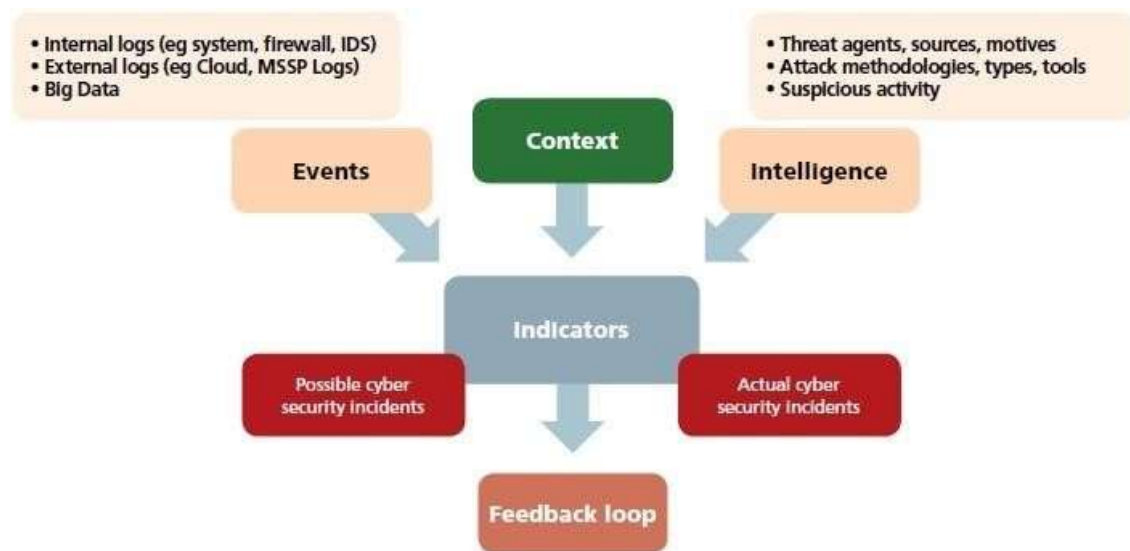


Figure 2.1: CREST Monitoring and Logging Framework (Creasey, 2015).

The main drawback of continuous monitoring is managing the complexity of information collected, and the level of effort required to synthesize the collected data into digestible information. The model also assumes devices and applications will generate sufficient logs to identify a fault. Finally, the model does not make explicit the remediation of incidents (Creasey, 2015).

iv. Intelligence Driven

Schlein, (2014) further asserts that, the intelligence driven model attempts to understand cyberattacks using information about the aggressors. Motivation and attack life cycle are analyzed to assess points of vulnerability and defensive gaps. Lockheed Martin's Cyber Kill Chain framework implemented this at the operational level and outlined how it basically works. It consists of seven stages through which an attacker must successfully progress, with the belief that a defender can disrupt the attacker at any of the seven stages.

Adversary intelligence is collected and evaluated for each step toward identifying attack mitigation strategies (Hartaud et al., 2014).

The seven stages of the Cyber Kill Chain are:

- i. Reconnaissance: the attacker identifies the victim's assets and potential vulnerabilities to exploit;
- ii. Weaponization: the attacker develops tools and scripts to conduct the cyberattack;
- iii. Delivery: the attacker deploys the tools and scripts previously developed, most likely remotely, toward the target;
- iv. Exploitation: the attacker uses the tools and scripts to take advantage of a vulnerability in the attacked system;
- v. Installation: once unauthorized access is obtained; the attacker moves additional tools and scripts to the victim system(s) to further exploit the compromised network;
- vi. Command and Control (C2): the attacker remotely controls the tools and scripts, allowing him or her to further exploit the compromised network;
- vii. Actions on Objectives: whether through exfiltration or destruction, the attacker affects his or her original goal for attacking the victim.

Deloitte cyber intelligence model introduced a more strategic model, geared toward processing of data and information into actionable intelligence. Key elements include the collection of diversified information feeds from within and outside the defending organization, integrated with expert analysis in a relevant contextual framework, and used to inform both technical and business decision processes.

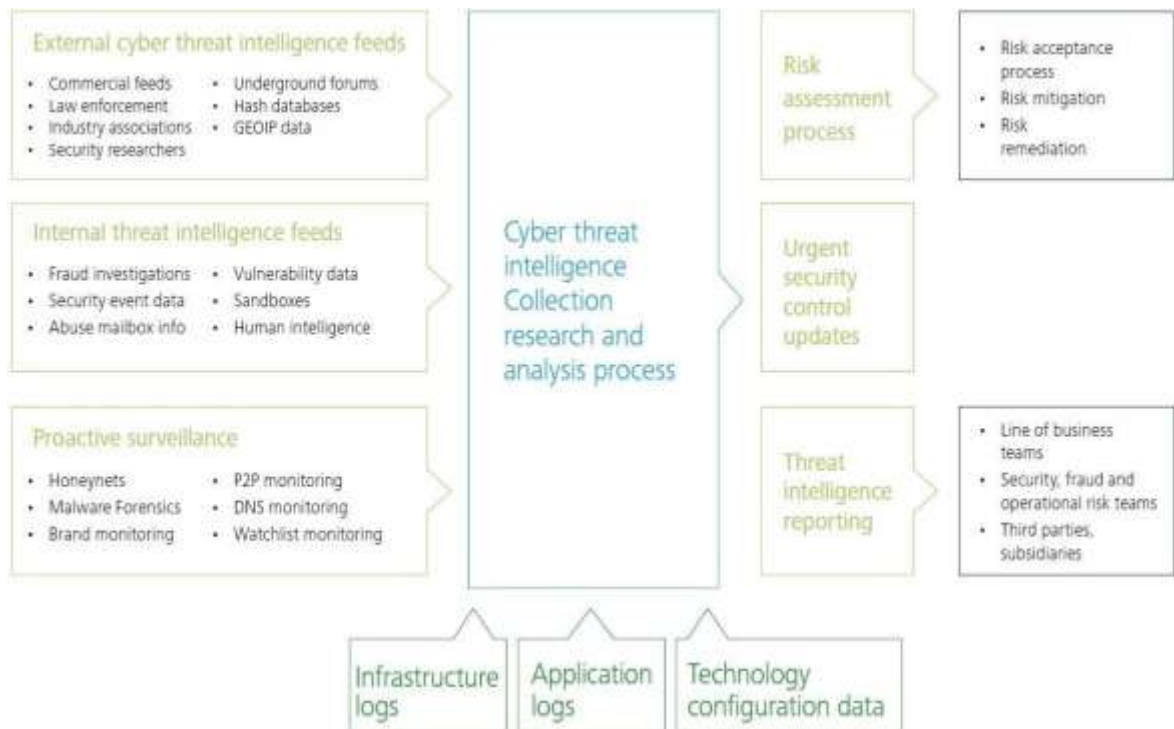


Figure 2.2: Deloitte Cyber Intelligence Model (Hartaud et al., 2014)

Despite its popularity, the Cyber Kill Chain has been criticized for focusing on malware and neglecting insider threat, social engineering, and non-malware remote access methods as the vector of intrusion. Another critique of Deloitte's model is that it relies upon considerable analytical resources for intelligence production, almost certainly far greater than available to most organizations (Schlein, 2014).

2.3 Cyber-Attacks

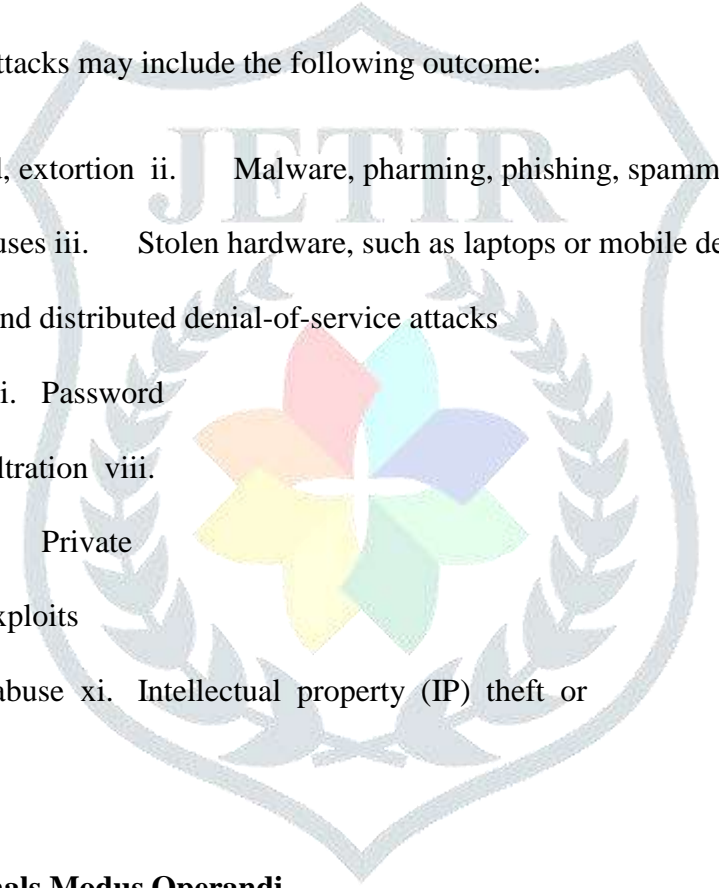
A cyber-attack is an attack initiated from a computer against a website, computer system or individual computer (collectively, a computer) that compromises the confidentiality, integrity or availability of the computer or information stored on it (Vince et al., 2011).

Cyber-attack is any form of assault or retreat operation engage by individuals or organizations that focus on computer information systems, infrastructures, computer networks, and/or personal computer devices by various means of malevolent acts usually originating from an unidentified source that either steals, alters, or destroys a specified target by hacking into a susceptible system it (Vince et al., 2011). These can be labeled either as a Cyber campaign, cyber warfare or cyber terrorism in different context. Cyber-attackers operate at a different range; sometimes can install a spyware on a Personal Computer so as to try to destroy the

infrastructure of entire nations. Cyberattacks have become increasingly sophisticated and dangerous as the Stuxnet worm recently demonstrated (Karnouskos, 2011).

More importantly, a cyber-attack is defined to mean ‘deliberate exploitation of computer systems, technology-dependent enterprises and networks. Cyber-attacks use malicious code to alter computer code, logic or data, resulting in disruptive consequences that can compromise data and lead to cybercrimes, such as information and identity theft. In addition, cyberattack is also known as a Computer Network Attack (CNA)’ (Karnouskos, 2011).

Further to this, the Cyberattacks may include the following outcome:

- 
- The logo is a shield-shaped emblem with a laurel wreath border. Inside the shield, the word 'JETIR' is written in a large, serif font. Below the text is a stylized flower with five petals in different colors: red, cyan, blue, green, and yellow.
- i. Identity theft, fraud, extortion
 - ii. Malware, pharming, phishing, spamming, spoofing, spyware, Trojans and viruses
 - iii. Stolen hardware, such as laptops or mobile devices
 - iv. Denial-of-service and distributed denial-of-service attacks
 - v. Breach of access
 - vi. Password sniffing
 - vii. System infiltration
 - viii. Website defacement
 - ix. Private and public Web browser exploits
 - x. Instant messaging abuse
 - xi. Intellectual property (IP) theft or unauthorized access

2.3.1 Basic Cyber Criminals Modus Operandi

The Organization for Economic Cooperation and Development (OECD) report (2007), categorized financial related cybercrime into five types:

- i. Innovators: who seek to find security holes in the system to overcome protection measures adopted by the banks.
- ii. Amateur: who are beginners and their expertise is limited to computer skills.
- iii. Insiders: who are working within the bank to leak out important information to order for a revenge.

- iv. Copy cats: they are interested in recreating simple tasks.
- v. Criminals: highly organized and very knowledgeable who may use all the above-mentioned stakeholders for their own profit.



Table 2.1: show types of techniques used by cybercriminals. (Marshall, 2014).

TYPES OF ATTACK**DETAILS**

Virus and worms	Viruses and worms are computer programs that affect the storage devices of a computer or network, which then replicate information without the knowledge of the user.
Spam emails	Spam emails are unsolicited emails or junk newsgroup postings. Spam emails are sent without the consent of the receiver - potentially creating a wide array of problems if they are not filtered appropriately.
Trojan	A Trojan is a program that appears legitimate. However, once run, it moves on to locate password information or makes the system more vulnerable to future entry. Or a Trojan may simply destroy programs or data on the hard disk.
Denial-of-Service (DoS)	DoS occurs when criminals attempt to bring down or cripple individual websites, computers or networks, often by flooding them with messages.
Botnet attacks	Compromising of a group of computers by a 'hacker', who then uses the computers to carry out an attack over the internet.
Scare ware	Using fear tactics, some cyber criminals compel users to download certain software. While such software is usually presented as antivirus software, after some time these programs start attacking the user's system. The user then has to pay the criminals to remove such viruses.
Malware	Malware is software that takes control of any individual's computer to spread a bug to other people's devices or social networking profiles. Such software can also be used to create a 'botnet' - a network of computers controlled remotely by hackers, known as 'herders,' - to spread spam or viruses.
Fiscal fraud	By targeting official online payment channels, cyber attackers can hamper processes such as tax collection or make fraudulent claims for benefits.

Table 2.1 Cont.: show types of techniques used by cybercriminals. (Marshall, 2014).

Phishing	Phishing attacks are designed to steal a person's login and password. For instance, the phisher can access the victims' bank accounts or assume control of their social network.
----------	--

Website defacement	Changing the visual appearance and usability of a webpage – usually through breaking into the web server and replacing the original hosted website with a replacement.
Spoofing	Changing the identity of a remote computer to the identity of a computer with special access privileges on a particular network.
Carders	Stealing bank or credit card details are another major cybercrime. Duplicate cards are then used to withdraw cash at ATMs or in shops.
State cyber-attack	Experts believe that some government agencies may also be using cyberattacks as a new means of warfare. One such attack occurred in 2010, when a computer virus called Stuxnet was used to carry out an invisible attack on Iran's secret nuclear program. The virus was aimed at disabling Iran's uranium enrichment centrifuges.

2.3.2 Example of Cyber-Attacks in some Jurisdictions

Recently, there are cyber-attacks in some of the countries across the globe and most of the attack affects the country as a matter of national security. In Russia, the Russian presidency's website was attacked on several times on by Moscow hackers and that subsequently brought down the Central Bank website. It was reported that a powerful cyber-attack is under way on the (Kremlin) site and it continued unabated with no clue as to the source of the attackers (Zee News, 2014). In addition to this, there was also reports that Ukraine's Prime Minister blamed Russian intelligence for an attack against German government websites, for which a pro-Russian group claimed responsibility (Zee News, 2015).

Also, in Brussels and London a group of Hackers brought down several public North Atlantic Treaty Organizations (NATO) websites, the alliance reports, in what appeared to be the latest escalation in cyberspace over growing tensions over Crimea. "It doesn't impede our ability to command and control our forces. At no time was there any risk to our classified networks," another NATO official said (Zee News, 2015). The so-called "distributed denial of service" (DDoS) attack, in which hackers bombard websites with requests causing them to slow down or crash, also hit the site of a NATO-affiliated cybersecurity center in Estonia. A group calling itself "cyber berkut" said the attack had been carried out by patriotic Ukrainians angry over what they saw as NATO interference in their country (Zee News, 2015).

In India, the city of New Delhi received a second-most cyber-attack on mobile devices prone country with a major chunk of these intrusions designed for phishing and stealing banking details, a report by security software maker Kaspersky said (Zee News, 2014).

Also, in China, the cities of Shanghai and Beijing, faced an Internet outage that rerouted millions of users to a U.S. website of a company which helps people get around Beijing's censorship remained a mystery, but experts weighed the possibility of a cyber-attack. Users were redirected to a site run by a company tied to the Falun Gong; a spiritual group banned in China which has been blamed for past hacking attacks (Nee News, 2014).

The state-run China Internet Network Information Center (CNNIC) said in a micro blog post that the outage, which lasted for several hours, was due to a malfunction in China's top-level domain name root servers. Chinese Internet users were rerouted to a U.S.-based website run by Dynamic Internet Technology (DIT), a company that sells anti-censorship web services tailored for Chinese. A mistake made by the Chinese government could be at fault for the outage. "Instead of targeting a small list of websites the (Chinese Internet censorship systems) malfunctioned and targeted any domain. The malfunction is a result of a Domain Name Service (DNS) hijacking, said Bill Xia, founder of DIT, where even if people tried to go to a non-existing website, they would be redirected to DIT's (Zee News, 2014).

In Israel, the city of Jerusalem: Hackers attacked Israeli computers including one used by the defense ministry department dealing with civilians in the occupied West Bank region, an Israeli data protection expert confirmed (APF, 2014). "At the beginning of this month a number of mails were sent to a number of companies in Israel, including security organizations," (Nee News, 2014). "There was an attachment... and whoever opened it was infected with a virus, a Trojan Horse, which allowed the attackers to control those computers. One of the computers belonged to the Civil Administration". And he further said that the virus allowed the attacker "complete control of the infected computers, the attackers could carry out any operation within that network." (Zee News, 2014).

The Sony cyber-attack in 2014, which targeted personally identifiable information's about Sony Pictures Entertainment (SPE) employees and their dependents, e-mails between employees, information about executive salaries at the company, copies of unreleased Sony films, and other information, was obtained and released by a hacker group going under the moniker "Guardians of Peace" or "GOP". The identities of the hackers are currently unknown. Currently, the available information is that whether or not individuals at SPE assisted in the system compromise (Zee News, 2014). But that still is yet to be verified. Still the motives for the hack have yet to be revealed, the hack has been tied to the planned release of the film, The Interview, which depicts an assassination

attempt on North Korean leader Kim Jong-un, with the hackers threatening acts of terrorism if the film were to be released (Zee News, 2014). Further to this development, investigators in the U.S. looking into the Sony Pictures hacking believe that North Korea hired hackers to wage the cyberattack (Tech Times, 2014). According to a Reuters report, it is believed that the hackers were hired from outside North Korea to aid in the cyberattack on Sony Pictures on November 24th 2014. The cyber-attack was likely "contracted out" as North Korea itself "lacks" the ability to carry out some of the elements required in the attack. The officials investigate whether North Korea recruited contractors from outside to hack Sony. To date, the FBI stands firm on its stance that the country was the orchestrator of the Sony Pictures attack (Tech Times, 2014).

In January 2015, the United State Twitter and YouTube accounts were hacked on Monday 12th by people claiming to be sympathetic towards the Islamic State militant group being targeted in American bombing raids, the account centrally oversees the United State military command operations in the Middle East (Zee News, 2015). United State officials acknowledged that the incident in which the accounts were "compromised" for about 30 minutes was embarrassing but played down the impact, and the FBI said it was still investigating the matter. Also, the Pentagon spokesman Army Colonel Steve Warren said the Defense Department "views this as little more than a prank, or as vandalism." (Zee News, 2015). The hacked accounts displayed a post that "In the name of Allah, the Most Gracious, the Most Merciful, the CyberCaliphate continues its CyberJihad," Furthermore, the Twitter account published a list of generals and addresses associated with them, titled "Army General Officer Public Roster (by rank) 2 January 2014" (Nee News, 2015).

In a press briefing from the White House spokesman Josh Earnest said that the hacking was "something that we take seriously." But Earnest added, "There's a pretty significant difference between what is a large data breach and the hacking of a Twitter account." Even as the hacking was taking place, President Barack Obama announced new proposals aimed at bolstering American cybersecurity after high-profile hacking incidents including one against Sony Pictures Entertainment that U.S. officials blamed on North Korea (Zee News, 2015). In a review of some of

the documents by Reuters released by the hackers could not immediately identify any that appeared to contain information that compromised national security. Some of the documents were easily found using Google searches. After the hacking, the heading of the Central Command Twitter account showed a figure in a black-and-white head scarf and the words "Cyber Caliphate" and "I love you ISIS" (Zee News, 2015).

2.3.3 Identifying the Tools use in Cyber-Attacks

According to Graham and Howard (2010), cyber attackers always have diversity of tools at their disposal which can be deployed to disrupt, damage computer systems or capture confidential information. Highly sophisticated cyber-attacks will often utilize a variety of different tools in order to maximize their effectiveness.

Phishing is one of the most common terms associated with cybersecurity. Phishing usually occurs via an email which takes on the appearance of official correspondence from a trusted source such as a bank, employer or a known entity. The aim is to redirect the user to a website where they then enter account details or hand over personal data unknowingly, this can then be used by the initiators of the attack to attempt to defraud you or use or details for a criminal purpose. Thus, attackers' bait or "phish" for a users' confidential information. Attackers can take it a step further by doing a targeted "spear phishing" attack which involves gaining prior knowledge about an individual before then trying to capture their details (Singer and Friedman, 2014). Another type of attack with goals to phishing is known as a Structured Query Language (SQL) injection which is a common way in which websites and companies are attacked. SQL is a programming language which assists in the management of data and is often used to store and capture information. Attackers will instead of entering the requested personal details on a website chose to enter specific command codes which allow them access to the website's database as a whole. Essentially the computer has interpreted the data as an actual instruction and in doing so, compromised itself. The access can be so far reaching that it allows an attacker to take control of an entire web server (Singer and Friedman, 2014; Sochor and Zuzcak, 2014).

Attackers have also designed malware which is a malicious software that aims to target a specific vulnerability in a computer. These are referred to as viruses or worms and they often have detailed instructions attached to them which guide them through a system that has been compromised. Computer viruses embed themselves in a particular file by making a copy of themselves. They then prevent files from been able to execute properly. Viruses generally requires some kind of human involvement or interaction (unknowingly) in order to replicate. Worms operate independently and are able to replicate themselves as they move from one computer system to another and don't require human involvement. They spread across a network of computers often overwhelming IT infrastructure as they can attempt to replicate files and slow down computer functions. Some worms are designed to capture personal data while others are designed to focus on destroying data on a target's computer. A Trojan horse refers to computer software that has been designed to conceal harmful code. It hides itself as a useful program and then activates when a user unwittingly executes a particular aspect of the software and begins hampering computer systems. Spyware, refers to a type of malware which is designed to track and monitor a computer user's data and send it on to an unauthorized third party (Sun, 2016; Sharma, 2010).

The use of malware becomes layered as attackers may sometimes seek to create an army of computers by creating networks of compromised "zombie" computers which can be used to coordinate an operation on a target. This is what is referred as bonnets and it can allow millions of machines to be controlled by a single user. Botnets are commonly used for distributed denial of service (DDoS) attacks which aims to attack the systems that allows connection to internet (Sachdeva et al., 2016). It is based on the fact that data transfers consume computational and bandwidth resources. As such it is a way in which whole system become overwhelmed and ultimately crash. The ability to block a single attacker is easy, however trying to stop millions is impossible, as the attack often involve a number of computers, they are generally hard to trace. This means they can often be used as a diversion in some cases while attackers have a more targeted attack, they want take place (Singer and Friedman, 2014) It is important to make the distinction between a Denial of Service (DoS)

attack and a Distributed Denial of Service (DDoS) attacks. A DoS attack refers to a coordinated attack from a distributed system of computers

(Sachdeva et al., 2016; Sharma, 2010).

According to Aamir and Zaidi, (2013), a DoS attack, is an attack which is launched to make networks and systems resources unavailable for the legitimate users so that no one else can access it. Hackers can create a situation in which an organization comes to a grinding halt. The main targets of these attacks are web servers, default gateways, personal computers, etc.

Most of the hackers usually keeps three things in mind. One is to explore a way through which they can get the secret information. This is to compromise the confidentiality. Second is get access to the confidential information to change or modify it. This involves the compromising of integrity. Third is to compromise the availability. The two options are usually not enjoyed by novice attackers because it is not easy to gain an unauthorized remote access to a system. Thus, they try to target the availability for which they do not need any administrative privilege on the target systems. Most DoS attacks depend upon the weaknesses in TCP/IP stack protocols, and some of the classical examples of DoS attacks currently available are TCP Syn Flood, UDP Flood, ICMP flood, Smurf and Incomplete HTTP Request and a host of others.

DoS attacks either make use of a single computer or multiple computers to perform the attacks. However, the usage of multiple computers to perform the attack is known as Distributed Denial of Service (DDoS) attack. With DoS attacks and its associated variants, systems are first compromised by using Trojans, worms, etc. These compromised machines are named as zombies while the controller machine is considered as master. This master-zombie relationship works somewhat similar to client-server architecture. It can be very difficult to detect the DDoS attacks because the zombies may be situated across the globe. As a result, they cannot be differentiated from the legitimate traffic (Aamir and Zaidi, 2013).

The explanations highlighted above aim to demonstrate the different tools available to cyber attackers who are looking to try and compromise a system. One of the key concerns which has shaped cybersecurity in relation to data management and integrity has been how cyber-attacks can threaten or disrupt critical banking infrastructure and render them completely useless and vulnerable to attacks. These financial systems represent a high value to potential attackers because if their systems are compromised, they can cause wide spread harm. The base worst-case scenario for such attacks is that hackers would launch a computer virus which targets a financial institution's mainframe and critical network infrastructure. The attack would lead to root compromise which occurs when the virus allows the hacker access to main administration settings of the computer mainframe. Once this is achieved hackers can run their own programs, change how the system works and then try and hide traces of their intrusion. Particular malware can be deployed which allows a predefined set of instructions once the mainframe has been compromised (Grobler and Jansen van Vuurn, 2012; Smith, 2012; Sachdeva et al., 2016).

2.3.4 History of Cyber-Attacks on Financial Institutions

Cyber-attacks on critical financial institutions are a major threat, potentially causing significant damage and disruption to the financial sector and the larger economy. The complexity of the financial services industry, the interconnectedness of individual players, and the introduction of new and innovative technologies further heighten the risk of large-scale cyber-attacks on the financial sector (Pandy, 2016).

The cyber incidents describe below include defacement of websites, Distributed Denial of Service (DDoS) attacks, and intrusions using more sophisticated malware. In many cases, it is difficult to know with certainty who perpetrated the attack, but the suspected attackers range from criminals and hacking groups, and/or state sponsor actors.

1. The 2016 DDoS Attacks Targeting Russian Financial Institutions

On December 2, the Russian Federal Security Service announced that it had discovered pending cyber-attacks intended to impact “a range of major Russian banks” starting from December 5.30, Servers and command centers purportedly to be used in these attacks were located in the Netherlands and owned by a Ukrainian hosting company named BlazingFast. Its director, Anton Onoprichuk, said he had no information about the asserted attack and that his company was unable to find any malicious data. The Dutch Ministry of Security and Justice said that it was aware its infrastructure could be used for cyberattacks elsewhere, and in a statement noted that “in case . . . a cyberattack does occur, then it is up to the Russian authorities to decide whether to start an investigation. If desired, they can ask the Dutch investigating authorities for assistance.” (Kottasova, 2016).

Within the same period, Rostelecom, Russia’s telecom operator, said in a statement that it had blocked DDoS attacks against the five biggest banks and financial institutions in Russia. They reached a peak volume of 3.2 million packets per second, which is low compared to the volume of other recent DDoS attacks, and the longest lasted a few hours. The statement further noted that part of the DDoS attacks involved a botnet similar to that used in prior weeks against Germany’s Deutsche Telekom and Ireland’s Eircom, exploiting a vulnerability in home routers (Kottasova, 2016).

There was no identification of state actors or perpetrators of the attack, though the Russian Federal Security Service claimed that it was being organized by “foreign intelligence services” and speculation remained that due to the servers’ location and ownership, this had been an action on behalf of Ukraine. The Russian Federal Security Service stated that it expected the DDoS attacks to be accompanied by text messages, agitating social network publications, and blog statements about a “crisis in the Russian credit and financial system, bankruptcy and withdrawal of licenses of leading federal and regional banks,” and that “the campaign would be directed against several dozen Russian cities.” Presumably, this would be an attempt to create a run-on Russian bank, initiating a financial crisis. No evidence exists that such action, complementary to the DDoS attacks, was attempted (Kottasova, 2016).

2. The 2016 Bangladesh Central Bank Heist

In February 2016, media reported that hackers had breached the network of the Bangladesh central bank and sent thirty-five fraudulent transfer requests to the Federal Reserve Bank of New York, totaling nearly \$1 billion (Herman, 2016). Four of these fraudulent requests succeeded and the hackers were able to transfer \$81 million to accounts in the Philippines, representing one of the largest bank thefts in history. A fifth request for \$20 million to be sent to an account in Sri Lanka was stopped when misspelling of the recipient's name, "Shalika Fandation" rather than "foundation," raised suspicions. The remaining transfers, which totaled somewhere between \$850 and \$870 million, were also stopped before they could be completed (Herman, 2016).

The hackers had introduced malware onto the Bangladesh central bank's server and deployed keylogger software that allowed them to steal the bank's credentials for the SWIFT system. The hackers also custom-designed a malware toolkit that compromised Swift's Alliance Access system and was designed to cover their tracks. This toolkit allowed them to delete records of transfer requests, bypass validity checks, delete records of logins, manipulate reporting of balances, and stop attached printers from printing transaction logs. Although the malware was custom-designed for the theft, the toolkit could potentially be used against other banks in the SWIFT system running Alliance Access software (Herman, 2016).

The cybercriminals had monitored the bank's routine activity in order to create money transfer requests that appeared genuine and timed the thefts over the weekend in Bangladesh when the Federal Reserve reached out to confirm the transactions, and then it was the weekend in New York when the Bangladesh central bank employees instructed the Federal Reserve to cancel the transactions (Herman, 2016).

3. The 2016 Belgian National Bank DDoS Attack

On February 22, a hacking group called DownSec Belgium shut down the website for

Belgium's National Bank for most of the morning using DDoS attacks (Cerulus, 2016). Little information has been reported about the attack, but it followed similar DDoS attacks by the same group against the websites for the Belgian Federal Agency for Nuclear Control, the country's Crisis Center, and Belgium's federal cyber emergency team. DownSec Belgium claims to fight against corrupt government abuses.

4. The 2015 Russian Banks' Thefts from the Banks' Own Customers

There's little information available on this incident currently, but SC Magazine UK recently reported that the Russian Central Bank revoked the licenses of three Russian banks in 2015 because an investigation uncovered evidence that current and former bank employees had been using cyberattacks to withdraw money from the accounts of their own clients, as well as to cover up other crimes and violations committed by the banks (Gerden, 2016). The Russian Central Bank reported that in the last quarter of 2015 alone, more than \$20 million was stolen from the accounts of clients with what the central bank suspects was the knowledge or direct participation of the banks themselves. The central bank also reported that these hacks were likely the result of huge cuts to the financial industry in Russia over the preceding year, and these cuts had left disgruntled former bank employees willing to collaborate with hackers and left the banks unwilling or unable to shoulder the cost of upgrading their cybersecurity.

5. The 2015 Malware Currency Manipulation through Russian Bank

Russian-language hackers used a virus called the Corkow Trojan to hack into the computer systems of Russian-based Energobank starting in September 2014 (Cluley, 2016). They were able to harvest credentials, launch their own trading software, and, on February 27, 2015, they placed more than \$500 million in orders at nonmarket rates that caused the exchange rate to swing with extreme volatility between 55 and 66 rubles per dollar for a period of fourteen minutes.

Interestingly, it doesn't appear that the hackers made any significant profit directly from the operation itself, although it's possible that they took advantage of their insider knowledge to profit in other markets. It's also possible that this attack was a pilot exercise for future attacks.

Energobank has claimed losses of \$3.2 million due to the trades.

6. The 2015 Metel Malware Attack on Russian Banks

A group of cybercriminals used the previously discovered Metel banking Trojan to steal directly from banks rather than end users. The criminal gang which is believed to consist of fewer than ten members used spear phishing emails or browser vulnerabilities to hack into parts of the banks' systems that had access to money transactions, such as the computers used by call center operators or the banks' support teams. Once inside, the Metel malware automated the rollback of ATM transactions. This allowed the criminal group to use cards from the compromised banks to withdraw a virtually unlimited amount of money, because after each transaction the balance on the account automatically reset to the same amount. No infections of this kind have been detected outside of Russia (Kochetkova, 2016).

7. The 2015 Ukrainian Ministry of Finance Data Breach

In May, the pro-Russian hacktivist group CyberBerkut claimed to have hacked into the network of the Ukrainian Ministry of Finance (Southfront, 2015). The group posted what it claimed were documents stolen from the network, demonstrating that Ukraine was unable to service its external debt. The veracity of the group's claims and the means by which they allegedly gained access to the ministry's network remain unknown.

8. The 2014 Warsaw Stock Exchange Breach

In October, a group claiming to be affiliated with the so-called Islamic State hacked the internal networks of the Warsaw Stock Exchange and posted dozens of login credentials for brokers online. The means by which the group gained access to the exchange's networks are unknown, but they were reportedly able to infiltrate an investment simulator and a web portal for managing the stock exchange's upgrade to a new trading system, as well as render the exchange's website unavailable for two hours. Exchange employees say that the trading system itself was not breached. NATO officials later indicated privately that they believed that the hacking group's claim of being affiliated with Islamic militants was a false flag operation, and that in fact the breach was conducted by APT, a group

widely believed by security researchers to be affiliated with the Russian government (Riley and Robertson, 2015).

9. The 2014 Ukrainian Bank Data Breach

In July, the Pro Russian group called CyberBerkut hacked into PrivatBank, one of Ukraine's largest commercial banks, and published stolen customer data on VKontakte, a Russian social media website. The means by which they gained access to the data is unknown. It is believed that they targeted PrivatBank because the bank's co-owner, Igor Kolomoisky, had offered a \$10,000 bounty for the capture of Russian-backed militants in Ukraine. CyberBerkut warned PrivatBank customers to transfer their money to state-owned banks. CyberBerkut may have connections to the Russian government, but the relative lack of sophistication of their attacks has led some experts to conclude that official links are unlikely (Gertz, 2014).

10. The 2013–2015 Carbanak Malware Attack on Various Banks

A group of criminals used Carbanak malware to attack financial institutions including banks and electronic payment systems in nearly thirty countries. The malware installed a RAT (remote access tool) that allowed the criminals to surveil the banks' daily operations using video feeds and photos over a period of months. The group was then able to order ATMs to dispense cash at terminals and impersonate bank officials to order fraudulent transfers. However, the largest amounts of money were stolen when criminals impersonating bank officers hacked into the banks' accounting systems and manipulated account balances so as to inflate the amount of money available and then transfer the additional money, so that the balance then returned to the original amount. The targeted countries included Australia, Brazil, Bulgaria, Canada, China, the Czech Republic, France, Germany, Hong Kong, Iceland, India, Ireland, Morocco, Nepal, Norway, Pakistan, Poland, Romania, Russia, Spain, Switzerland, Taiwan, Ukraine, the United Kingdom, and the United States (Kaspersky Lab, 2015).

11. The 2013 Malware Attack on South Korean Banks

This was an attack on March 20 that used what's known as Dark Seoul malware against the computer networks of three South Korean banks Shinhan, Nonghyup, and Jeju resulting in data deletion and disruptions to ATMs and mobile payment systems. Shinhan Bank's internet banking servers were temporarily blocked for part of the day, leaving customers unable to perform online transactions, while operations at some branches of Nonghyup and Jeju were paralyzed for two hours after the virus erased files on the infected computers. A fourth bank, Woori, reported hacking but suffered no damage. Several Korean media organizations were also hit by the attacks: their computers were frozen but they were able to maintain normal broadcasts. South Korea attributed the attack to North Korea (Kwon, 2015).

12. The 2012–2015 Crime Ring Responsible for JPMorgan Data Breach

In August 2014, JPMorgan reported a massive data breach in which hackers had gained access to contact information for over 80 million account holders, representing the biggest data breach of a U.S. financial institution in history (O'Toole, 2014). Although there was initial speculation that the Russian government had been involved, federal authorities indicted four men in November 2015 for the data breach, which they said was part of a huge operation that involved hacking into other financial institutions, a stock-pumping scheme, and online gambling operations that in total had netted them \$100 million. The criminals used the email addresses they gained through the JPMorgan hack to run a stock price manipulation scheme and also hoped to set up their own brokerage firm using the stolen data to contact potential customers. Although the JPMorgan hack was their biggest, the crime ring had also hacked six other financial institutions, Scottrade, E-Trade, Dow Jones (the parent company that owns the Wall Street Journal), another financial news organization, and several online stock brokerages.

13. 2012 and 2014 DDoS Attacks Against Brazilian Banks

In January 2012, the hacker group Anonymous used DDoS attacks to take down the websites of some of the country's biggest banks, which they said was intended to protest corruption and

inequality in Brazil. The attacks, which they dubbed #OpWeeksPayment, shut down the websites for Banco do Brasil, Itaú Unibanco, and Bradesco, among others, for hours at a time (Israel, 2014.)

In June 2014, Anonymous launched another series of DDoS attack, this time to protest the World Cup. The attacks, called #OpHackingCup, took down several Brazilian websites including the Bank of Brazil. Other websites that were targeted included Brazilian government websites, Hyundai Brazil, and the official World Cup site (Cooper, 2014).

14. 2012–2014 Malware Attack on Brazilian Payment System

Cybercriminals used “man-in-the-browser” malware to target Boletão Bancário, a popular Brazilian payment system. The payment system allows businesses to issue paper or online boletões (tickets) with a barcode that customers can use to remit money at a bank. The malware injected itself into browsers on nearly 200,000 infected computers, where it was able to intercept and alter legitimate boletões so as to route payments into the hackers’ own accounts. The attack compromised \$3.75 billion in transactions, although it is unclear how much of that money the criminals were able to successfully deposit into their own accounts (BBC, 2014).

15. 2012–2013 DDoS Attacks on U.S. Financial Institutions

These were two coordinated waves of DDoS attacks against U.S. financial institutions’ websites, the first in September–October 2012, and the second in took place between December and January 2013. An Islamic hacktivist group called the Izz ad-Din al- Qassam Cyber Fighters claimed responsibility for the attacks, which they dubbed Operation Ababil, but U.S. government officials have privately indicated to media that they believe Iran is actually responsible. The scale of the attacks was unprecedented in the number of financial institutions hit and the amount of traffic flooding the sites, with one security researcher commenting that “there have never been this many financial institutions under this much duress.” Although the group announced the attacks and the targets in advance both times, the banks were unable to defend themselves and access to the websites of many U.S. financial institutions was disrupted, including Bank of America, Citigroup, Wells Fargo, U.S. Bancorp, PNC, Capital One, Fifth Third Bank, BBandT, and HSBC. Defensive and remedial

measures have cost the banks millions of dollars to date. Izz ad-Din alQassam Cyber Fighters announced two more waves of cyberattacks in 2013, but they appear to have been less effective (Schwartz, 2013).

16. The 2012 Possible Manipulation of the Shanghai Stock Exchange

On June 4, the Shanghai Composite Index opened at a figure of 2,346.98, and fell exactly 64.89 points by close. June 4 is the anniversary of Beijing's infamous 1989 crackdown on student-led protests in Tiananmen Square, prompting many in China to speculate that both figures may have been intended to represent the anniversary of the tragedy (Bradsher, 2012). The number 2,346.98 can be read backwards as the year, month, and date, followed by to represent that 2012 marked the twenty-third anniversary of the protests. Similarly, many observers in China speculated that the 64.89 points that the stock market fell that day also represented 6/4/89. The apparent coincidence led to widespread, but unproven, speculation that the index may have been hacked and manipulated in order to produce those numbers. Numerology is very significant in Chinese culture, and Chinese citizens have been known to use numbers as a subtle form of protest in the past.

17. 2011–2012 Gauss Virus Infecting Lebanese Banks

On August 9, 2012, the Russian security firm Kaspersky Lab announced the discovery of the Gauss virus, which is designed to steal data from Lebanese banks, including the Bank of Beirut, EBLF, BLOM Bank, ByblosBank, Fransabank, and Credit Libanais, as well as from users of Citibank and PayPal. Kaspersky's experts concluded that the virus is state-sponsored malware designed by the creators of Stuxnet, Flame, and the Duqu collection of espionage Trojans. More than 2,500 computers belonging to Kaspersky customers have been infected in twenty-five different countries 1,660 of those in Lebanon although the security firm cautions that the total number of infected machines may number in the tens of thousands (Zetter, 2012).

Once a PC has been infected, the Trojan steals detailed information, including browser history, passwords, cookies, system configurations, and online banking account credentials, and also installs a special font called Palida Narrow, the purpose of which is unknown. Most interestingly, Gauss

contains an encrypted payload that security researchers have been unable to decipher, indicating the presence of a significant exploit that the virus's creators clearly considered important to protect. Given that Lebanon serves as a banking hub for the entire Middle East and that the opacity of the country's banks has often been a concern for financial regulators seeking to disrupt terror financing and money laundering, it seems likely that the virus may be designed to monitor and/or disrupt money flows deemed threatening to the sponsor state's national security (Goodin, 2015).

18. The 2011 Malware Targeting a South Korean Bank

This incident targeting the banking operations of Nonghyup, a South Korean agricultural cooperative, began on April 12. The malware initially infected Nonghyup's systems in September 2010 when a subcontractor inadvertently downloaded it onto a laptop, which the attackers used to spread the malware throughout the bank's networks. The attack destroyed the records of some credit card customers and caused a three-day service outage affecting ATMs, online and mobile banking, and credit card usage. South Korea attributed the attack to North Korea (Yonhap, 2011).

19. The 2010 NASDAQ Intrusion

The intrusion of NASDAQ's networks was first reported in an exclusive Bloomberg Business exposé in 2014 (Riley, 2014). In October 2010, the FBI detected an intrusion into NASDAQ's computer servers. The intrusion utilized two zero-day vulnerabilities and resembled malware previously designed by Russia's main intelligence agency, the Federal Security Service. The malware first entered through NASDAQ's Directors Desk, a system that hundreds of companies use to share confidential financial information among board members. NASDAQ's own statement at the time reported that the incursion was limited to that system alone, although Bloomberg's reporting indicated that, in fact, the incursion may have spread more widely through the stock exchange's networks while never accessing the trading platform itself.

The NSA initially believed the malware was capable of causing widespread disruption to NASDAQ's computer networks and of possibly wiping the entire exchange. There were also indications that a large cache of data had been stolen, although investigators had little proof of what exactly had been taken. The CIA later argued that the malware was less destructive than originally believed, and that while it couldn't completely wipe a computer system it could take over certain functions and use them to disrupt the network. The investigators ultimately concluded that the intrusion was primarily designed to steal critical proprietary technology for Russia to imitate or incorporate into its own stock exchanges as part of a push to turn Moscow into a global financial hub. The malware has not been publicly analyzed and Bloomberg's reporting included few details, so further technical information about the malware and its capabilities is unavailable in open-source literature (Riley, 2014).

20. The 2008 Website Defacement during the Russo-Georgian War

Offensive cyber operations against targets in Georgia began on July 20, prior to the outbreak of the war itself, and continued until mid-August when the conflict ceased. This was the first ever combination of offensive cyber operations with kinetic war and was allegedly carried out by the Russian government or Russian hacktivists with ties to the government. On the day that the kinetic war began, websites sprang up with lists of websites to attack, precise instructions, and survey forms for hackers to report their actions after the fact, demonstrating a telling degree of advance preparation and foreknowledge of the beginning of the conflict. The operations consisted of website defacements and DDoS attacks, with targets including the Georgian president's website and other government sites. The only impact on the financial sector was the defacement of the National Bank of Georgia's website (Markoff, 2008).

2.4 Concept of Electronic Banking Fraud

Numerous definitions of fraud have been advanced in the crime literature. Wells (2014) defined fraud as unlawful gain through deception. Taylor (2011) argued, in line with Wells, that fraud

is stealing, disguising and obtaining personal gain from another person or a group of persons through deception. Curt's (2013) noted that fraud contains the acquisition of property or monetary advantage by way of deception, either concealment or misrepresentation. According to Graycar and Smith (2002), frauds usually encompass the transaction, falsification or forgery of financial documents and unlawful endorsement.

Fraud occurs when a person or a group of persons of authority and responsibility refuse standard and violate the rules for the benefit of self-interest at the expense of others. Fraud is misrepresentation of financial records by an individual or group of individuals among employees in the management of an entity or third parties (KPMG, 2000; Silverstone and Sheetz, 2007; CIMA, 2009; Mirjana Pejic-Bach, 2010).

Furthermore, from the above definitions, fraud can be described as a deception and a false channel for converting another person's (legal owner) financial and non-financial property/assets for personal interest illegally. It can also be explained as a misrepresentation of financial statement which is intentionally done by internal or external stakeholders of an organization for personal motives. Bank fraud, then, involves the deceitful use of one's position without or within the bank for self-enrichment by intentionally misappropriating the bank's financial means, properties, or other resources held by the bank and collecting funds from bank accounts of customers (Graham, 2008; Finch, 2010; Taylor, 2011).

2.4.1 Methods of Perpetrating Electronic Banking Fraud

Literature is satiated with distinctive styles of frauds. This view has been debated amongst scholars, and Hamilton et al., (2012) in the study titled, "styles of fraud are usually not exhaustive as fraudsters are forever devising new methods." Therefore, as societies and businesses are expanding as progressive techniques of committing frauds are sophisticated and classy. The growth of businesses and rapid increase of the fraud perpetration are as a result of the development and expansion of the communication and information technology (Silverstone and Sheetz, 2007; Pedneault et al., 2012).

Furthermore, fraud could be seen in two ways, viz. bite and nibble frauds. When an individual takes asset and disappears in order to not be detected is known as bite fraud. This style of fraud usually involves stolen of large assets or huge amount of money and can be easily detected.

To escape being detected and tracked down, the fraudster breaks out into a protected colony. Bite fraud can occur in the form of electronic frauds particularly, stealing of hardware devices or backup devices of a computer system. While, an individual or fraudster involves in taking assets in piecemeal or small unit in order to not be detected easily is called nibble fraud. This style of fraud is very difficult to be detected at an early stage, hence, this kind of fraud occurs every day and is common in electronic banking frauds, particularly, fraud through ATM, PoS, credit card and web banking fraud (Udoayang and James, 2004; Wilhelm, 2004).

Moreover, Alao (2016), grouped fraud styles into two, internal fraud and external fraud. When fraud is perpetrated by the individual employee or group of employees of an organization or a bank using computer, point-of-sales, automated teller machine and internet inform of phishing, vishing and counterfeited or forged smart card is recognized as internal fraud. While, fraud committed with the use of bank financial records, customer's financial information and electronic devices such as ATM, internet, mobile app, mobile phone, pocket picking machine by the outsiders, such as service providers, bank customers, suppliers and unknown party is called external fraud (Hansen et al., 1996; Sydney, 1996; Adewumi, 1986). Iwuagwu (2000) argued that, fraud perpetration can involve combination of both internal and external which in itself known as outsider-employee fraud. This kind of fraud is difficulty to detect because the insider-fraudster is supplier of financial information needed and bank security information carrier to the outsider fraudster who is an operator of fraudulent acts.

Additionally, ASSOCHAM, (2015) argued that fraud against a business organization can be perpetrated either externally by vendors, customers and other related parties such as individual or managers, employees, officers, and owners of the organization. The author categorized frauds into three basic categories which are: external frauds, internal frauds and frauds against individuals.

External frauds are kinds of frauds committed by outsiders or third parties by compromising electronic bank account through personal information about the victims. This fraud could happen through pharming, phishing and vishing. While, internal frauds also known as occupational frauds, can be explained as a means of using one's profession or occupation for self or personal gain through intentional misuse or misappropriation of the company's resources. This kind of fraud happens when the executives, managers and other employees perpetrate frauds against their employer (Anderson et al., 2012; ENISA, 2014).

Agbada and Osuji (2013) further argued that, fraudsters are progressing in the use of technologies and innovative approaches for concealment and perpetration of internal fraud schemes. While, fraud against individuals is a type of fraud in which many perpetrators have designed systems to defraud individuals such as identity theft systems, phishing systems, advanced fee crimes are just a few of the methods the fraudsters have discovered to defraud unsuspecting victims.

On the same vein, Adeyemo, (2012) opined that, fraud has been categorized in diverse ways and using various methods such as management and employee frauds, customer and non- customer frauds and stakeholder and non-stakeholder frauds. Management Frauds are electronic fraud perpetrated by the top management of the organization. These frauds can be committed through electronic financial statement and the group of sufferers of these kinds of frauds are creditors and investors (ASSOCHAM, 2015). This electronic banking fraud can be perpetrated through the creating of more investment from potential and current shareholder of the organization, Doctor of Financial statement or window dressing of account statement and can occur by painting the bank in better light in the eyesight of the regulatory authorities using electronic systems ((Perlroth and Gelles, 2014; Finkle and Hosenball, 2014).

While, Association of Certified Fraud Examiners (2012) concord that management frauds happen through timing differences, fictitious revenues, improper asset valuation, inadequate disclosure and concealed liabilities and expenses. Employees' Fraud is generally known as nonmanagement fraud. It is a kind of fraud committed by the non-management staffs or employees

of the organization through forgery of customers' signatures, stealing of customer's passwords, PIN codes and electronic cheques for illegal withdrawal of money from the customers' accounts, creating and operating of fictitious electronic bank account, fund diversion, lending fictitious borrowers and other related computer's fraud or internet frauds (Adeyemo, 2012).

Furthermore, customers and non-customers' frauds occurred through the act of performing the primary functions of money deposit Banks, which connects capital deficit customers with the capital surplus customers in the money market (Association of Certified Fraud Examiners, 2012). In the process of this, bankers come in connecting or interacting with both non-customers and customers and this leads to the risk of frauds. These types of frauds may be through counterfeit securities, opening of the fictitious electronic bank account, forged electronic cheque, fraudulent electronic money transfer because of a request made sole and solemnly through email, telephone, fax, telex, and other electronic means, and skimming card data (Regha, 2015).

While, stakeholders' and non-stakeholder frauds is the kind of fraud perpetrated through the collaboration of insiders and outsiders, employees and non-employees, staffs and non-staffs of the organization. Before this type of fraud to succeed, there must be an insider or internal fraudster that will be providing financial information while, the outsider fraudsters or external fraudsters will be carrying out the instruction given Adewunmi (1986). However, majority of banking functions are now electronically base activities including transaction of business such as funds, registration of new customers, collection of customers' personal data and preparation of financial statements, particularly in this era of cashless system, then, all types of fraud mentioned above are now electronic banking related frauds and there is need to discover the best way for detecting and perfecting of these menaces (Manyika and Roxburgh, 2011; Norse, 2014).

2.4.2 The Contributing Factors for E-Banking Fraud Increase

With the global use of progressively advanced internet technology, electronic banking is developing as a great medium or network for banking businesses (Chanson and Cheung, 2001; Park,

2015). However, electronic banking fraud perpetrations are becoming more sophisticated, unbearable, greatly intimidating the security and trust of electronic banking activities. Agwu, (2012) viewed electronic banking fraud as an epidemic disease in the banking industry, which has become a great challenge to both management and customers of the industry. E-banking fraud has become a global and provocative issue that produces debate amid quite a few authors like (Saleh, 201; Pandey, 2010) asserted that electronic banking fraud is a worldwide problem and is persistent to be overpriced to both banking sectors and customers.

To corroborate their views, Usman and Shah, (2013) opined that frauds in electronic banking services happen as a product of several concessions in security extending from inadequate internal controls to feeble substantiation systems. Electronic banking fraud is now a thoughtful matter of financial crime management in all financial institutions. The development and advancement of challenging of electronic banking frauds such as ghost website, phishing scams and malware have coursed a massive loss in the banking industries worldwide (Wei et al., 2013). Therefore, there is need to examine the causes of electronic banking frauds. A lot of factors contribute to the menace of fraud perpetration are grouped into: technological challenges and nontechnological challenges (Uchenna and Agbo, 2013; Ojo, 2008; Idowu, 2013).

Hence, for the benefit of this study, the factors that contribute to the increase of e-banking fraud were grouped into technological factors and non-technological factors which is explained below:

2.4.2.1 Technological Factors

The introduction of electronic banking has come with its risks and challenges, starting from electronic banking adoption to financial transaction and the adaptation of advanced technologies (Usman and Shah, 2013; Abu-Shanab et al., 2019). Kovach and Ruggiero, (2011) discovered that a lot of electronic banking accounts are usually defrauded by only one fraudster, which involves a small amount of money transaction with a total amount of money larger than one account. The author

concluded that many frauds occur as a result of increased number of password failures which give opportunities to fraudulent behaviors. Correspondingly, in a survey carried out in Australian banks on electronic banking frauds, the finding showed that almost electronic banking frauds have the following challenges and characteristics; ineffective real time detection, weak forensic evidence, dynamic fraud behavior, imbalance large datasets and diverse behaviors patterns of customers (Wei et al., 2013; Anderson et al., 2012).

Jassal and Sehgal, (2013) in their study titled “electronic banking security flaws” aimed at finding diverse types of faults in the security of electronic banking that leads to loss of money by customers and banks. The authors discussed the reasons of security breaches and the involvement of both banks and customers in giving a chance to crackers and fraudsters to have access into their networks through web-browser installed on their customer’s personal computer which give opportunities to unauthorized persons to have access to their personal identification information and financial information. Usman and Shah (2013) viewed electronic banking fraud as a global issue which is persistent to prove costly to both banking sector. Electronic banking frauds happen because of different concessions in security started from feeble authentication systems with inadequate internal controls (Usman and Shah, 2013; Nor et al., 2008).

Electronic banking fraud could be from bank website, such as cross site scripting through malicious and SQL statement entered by attackers into the web page of the bank (Schneier, 2011). Omar et al, (2011), argued that most are stakeholders willing to use electronic banking services because of its convenience, cost effectiveness, speed and easy accessibility. European Central Bank, (2014) reported that card fraud payment is one of the major means of fraud such as counterfeit card, card not received, lost and stolen cards. The author further elucidated that the contemporary mobile devices with their operating system were not intentionally produced with the security of financial payment, unlike traditional payments, mobile payments expose include extra actor in the signal transmission such as mobile network operators and also, the general public may not have adequate

awareness of the associated information security risks attach to use of mobile devices and internet desktops or laptop for payment at home (Anderson et al., 2012).

Correspondingly, Adedipe (2016) in the study internet fraud, findings show that, the external fraud is fundamentally direct outcome of hackers' activities which include unauthorized access to electronic bank account information which are accomplished through pharming attacks, phishing attacks, session hijack, skimming attack, eavesdropping hijack, brute force attacks. These are emanated through bank staff and customers' ignorance and unawareness of common social engineering techniques, negligence in displaying PIN code and accounting information, and carelessness disposal of computer devices and related software.

Deloitte (2015) in the study of “India Banking Fraud Survey” discovered that there is increase in the electronic fraud occurrence in the banking sector because of lack of the tools and technology to discover the potential red flags. Likewise, Regha (2015) concord that difficulties in preventing electronic banking frauds could be influence by the following factors which involve: ineffective monitoring of electronic banking channels such as ATM terminals, internet banking, telephone banking, personal computer banking and card teller banking, non-existence of camera such as CCTV at e-banking transaction terminals, absence or inadequate of system base solution to trace and to report suspicious transactions and compromised accounts, lack of compliance to Know-your-customer and best practice procedures of e-banking management, no segregation of transaction limits, failure of incorporating string validation test of security, ineffective encryption key management, inadequate control to restricted environment and availability of ex-staff with active login pin to e-banking management system data base (Park, 2015).

Equally, Odediran (2014), findings in the study titled “holistic approach to electronic channels fraud management” shows that, the factors that influence the rising cases of electronic banking frauds globally are Ignorance of cardholders on card usage security, inadequate monitoring of electronic payment terminals and lack of adequate management of electronic bank production services. Gates

and Jacob (2009) have pointed that the factor that contributes to increase of e-banking fraud is the mismanagement of technology in the banking industry which comprises use of technology for illegal activities, sharing of confidential data, banking access for over-payments to sellers. European Central Bank (2014) in the survey of cards fraud, further elucidated that the contemporary mobile devices with their operating system were not intentionally produced with the security of financial payment. Banking services and other financial industries experience losses annually through fraud incidences such as internet banking frauds, cheques and cards frauds (Adams, 2010). Therefore, these obviously signify that fraudsters are exploiting electronic banking channels.

It is easy to construe that automated teller machine (ATM) fraud incidents occurred most in the day time. Agwu, (2012) reported that some banks have no provision for reporting of fraud incidences and there is no enough orientation for the customers on how to operate e-banking devices such as automated teller machine, POS and the similar, neither provision of Fair and Accurate Credits Transactions Act (FACTA) or Automated Teller Machine Manuals for the ATM users (Moore and Clayton, 2009; Wang et al., 2011).

Moreover, technological factors, universally, the costs of managing e-banking fraud risk and the number of electronic banking fraud incidents are always increasing due to the sophisticated techniques used by electronic banking criminals (CIFAS, 2009). Symantec Security Response (2015) found that internationally, on average, 116 e-fraud attacks occurred each day in 2012 through social engineering and customized malware, obtaining unauthorized contact with sensitive information, as against 82 attacks per day in 2011. Likewise, phishing, card skimming, Trojans, spyware and adware, website cloning, cyber stalking, lack of sophisticated antivirus software and weak passwords contribute to the rapid increase of electronic banking fraud. Therefore, the importance of examination of electronic banking fraud prevention and detection cannot be over emphasized (Brar et al., 2012; Omariba et al., 2012; AvinashIngole and Thool, 2013).

2.4.2.2 Non-Technological Factors

Regardless of religion, ethnicity, culture and other factors, there are individuals that are still being motivated to perpetrate electronic frauds. Irrespective of technology, The American Institute of Certified Public Accountants (AICPA), and the Association of Certified Fraud Examination (ACFE, (2015), found that the financial pressure to make means is paramount to some individuals.

Usman and Shah (2013) discovered that inadequate staff education, customer education and internal control are other areas which need to be addressed to minimize electronic banking fraud. Moreover, diverse behavior patterns of customers, electronic banking customers perform different transactions in diverse ways for various purposes. This is a challenge as it results to variety of genuine customer transactions that would be imitated by fraudsters who change their behavior regularly to contend with advances in fraud detection, thus makes it hard to characterize fraud behavior from genuine behavior. BIS, (2012) viewed the cause of electronic banking fraud beyond electronics. The finding of these authors indicates that exploited staffs, lack of training, low compliance level and competition are the major reasons for electronic banking frauds. Therefore, there is need for banks to observe the rising graph of electronic banking frauds seriously and make sure that there is no slackness in internal control mechanism (Wei, et al 2013; Peotta et al., 201).

In conjunction with the above, Choplin and Stark, (2013) in an investigation conducted, the finding was that, the banking customers are vulnerable to electronic banking frauds and lack of education and demographics have impacts on consumers' vulnerability. Zimucha et al., (2012) reaffirmed this, to prevent payment card fraud, consumer education on personal information protection is essential. Masocha et al., (2011) discovered that the causes of electronic banking frauds are insecurity, limited internet access, cultural barriers and poor legislation. While, Agboola and Salawu (2008) concurred with this, security is paramount issue of the effectiveness of electronic banking services. El-Guindy (2008) asserted that most banks are investing on development of electronic banking, but not on its security. Although modern financial institutions have invested a lot on information technology

infrastructures, most banks and e-businesses in across the globe still lack cognizance of the significant of security in electronic banking (Omariba et al., 2012; Brar et al., 2012).

While, Sullivan, (2014) argued that, financial institutions bear huge losses yearly through electronic frauds such as card fraud, Automated Teller Machine frauds, misused of private passwords and negligent of the customers to their private transaction data. This signifies that fraudsters are taking advantages of electronic banking system. Then there is a need for substantial strategies for prevention and detection of fraud. Furthermore, Ganesan and Vivekanandan, (2009) warned that the manner of opening an internet account on the internet and transaction security on the internet must be paramount to both internet bank account holders and the bank managements. This was also corroborated by Regha, (2015) as evident from historical lesson learnt where insufficient security measures caused fraud in retail payment systems. This was backed up by example of cloning that led to losses of almost \$600 million from the store's value card encryption (Kinkela and Harris, 2014).

2.5 The Concepts of Deep Learning

Deep learning, a subfield of machine learning, excels in generalizing to new examples when the data is complex in nature and contains a high level of dimensionality (Goodfellow et al., 2016). In addition, deep learning enables the scalable training of nonlinear models on large datasets (Goodfellow et al., 2016). According to Deng (2014), deep learning is a class of machine learning techniques that exploit many layers of non-linear information processing for the supervised or unsupervised feature extraction and transformation, and for pattern analysis and classification.

Deng (2014), further asserts deep learning concept as a technique where many layers of information processing stages in hierarchical supervised architectures are exploited for unsupervised feature learning and for pattern analysis and or classification. Zhang et al., (2016), opined that deep learning is an artificial intelligence function that mimic the workings of the human brain in processing data and creating patterns for use in decision making. He further asserts that deep learning is a subset of machine learning in artificial intelligence (AI) that has networks capable of learning unsupervised from data that is unstructured or

unlabeled which is also known as deep neural learning or deep neural network. In conjunction with the above, it is delineated that deep learning is a subfield of machine learning that is concerned with algorithmic inspired structure and functions of the brain called artificial neural network (Goodfellow et al., 2013; Deng and Chen, 2014; Brownlee (2019).

However, a more elaborate definition of deep learning was promulgated by Nielsen, (2015), as the application of multi-layer neural network with multiple neurons at each layer to perform the desired tasks like classification, regression, clustering and others. Fundamentally, each neuron with activation function is considered as a single logistic node, which is connected to the input in the next layer. Each layer of a neural network has multiple neurons initiated with dissimilar weights and try to learn on the input data concurrently. Thus, in multiple layers with multiple nodes, each node learns from the output of the previous layers, and gradually decreases the approximation of the real input data to provide accurate output representation set. This leads to lot of complexity between multiple interconnected neurons (Deng et al., 2013; Dong and Wang, 2016; Zhag et al., 2016).

The idea of Deep Learning is derived from the structure and functionality of the human brain and the processing of signals through neurons in the human mind. Deep learning is considered as a promising avenue of research as it is capable of automatically identifying complex features at a high level of abstraction. It is about learning multilevel representation and abstraction, which is useful for the data, such as image, text and sound. One of the exceptional deep learning characteristics is its capability of using unlabeled data during the training process. This is important in the domain of network intrusion detection because not only is it dealing with a large amount of data, but the model generated by the deep learning system will need to be capable of generalizing to new forms of attacks not specifically represented in the currently available labeled data. Ideally, the model could generalize and be effective in new, never-before seen network environments, or at a minimum be leveraged in a machine learning pipeline as part of a transfer learning step when used with data from a different computer network environment (Schmidhuber, 2015; Goodfellow et al., 2016; Shashikumar et al., 2017).

2.5.1 Brief History of Deep Learning

While deep learning has gained popularity in recent years, it has been around for a long time and its origin dates back to the 1940s (Goodfellow et al., 2016). Through its history, it has gone by different names such as “cybernetics” in the 1940-1960 timeframe, and “connectionism” in the 1980-1990s, to what it is known by today as “deep learning” with renewed interest starting back up in 2006. Some of the early algorithms in deep learning were biologically inspired by computational models of the human brain, thereby popularizing algorithms with names such as artificial neural networks (ANNs) and by describing computational nodes as *neurons*. While the neuroscientific perspective is considered an important source of inspiration for deep learning, it is no longer the primary basis for the field - there simply does not yet exist a full understanding of the inner workings and algorithms run by the brain. This is an active and ongoing area of research being conducted within the field of “computational neuroscience.” While models of the brain such as the perceptron and neuron have influenced the architecture and direction of deep learning over the years, it is by no means a rigid guide. Modern deep learning instead is based more on the principle of multiple levels of composition (Goodfellow et al., 2016).

One of the catalysts in the resurgence of deep learning in the 2000s was due to a combination of the increase in computational power, along with the increase in available data. Deep learning excels when there exists a large amount of data for which the algorithm can learn from. According to (Goodfellow et al., 2016), the general rule of thumb as of 2016 is that supervised deep learning algorithms will achieve good performance with at least 5,000 labeled examples per category. They will also exceed human performance when they are trained with a dataset that has at least 10 million labeled examples. In the field of network intrusion detection, the most common benchmark datasets that have been used in the past such as the NSL-KDD ‘99 dataset are smaller in size, containing a total of 148,517 training examples, with 77,054 being benign, and 71,463 being attack. The newer benchmark datasets used in this work such as ISCX IDS 2012 and CIC IDS 2017 are much larger. The ISCX IDS 2012 dataset contains over 2.54M examples, with over 2.47M being benign, and

68,910 being malicious. Similarly, the CIC IDS 2017 dataset contains over 2.83M examples with over 2.27M being benign, and 557,646 being malicious. These larger datasets have many more examples for the neural network to learn from, and therefore can be used to experiment on the effectiveness of using deep neural network architectures for classifying flows as benign or malicious.

Goodfellow et al., (2016) further stressed that while these datasets don't quite have 10M examples, they are much larger than any IDS datasets used in the past, and can be used to experiment and determine the effectiveness of deep learning architectures and algorithms as applied to the domain of network intrusion detection. The amount of data available in practice in an enterprise network is enormous and highly dimensional, often not only containing raw PCAP and/or network flow data, but also including application event logs, host-based logs, security event data, and a myriad of other log data from workstations, servers, sensors, and other appliances spread throughout the network. Furthermore, there exists expert human analysts which can provide ongoing feedback to a learning-based system. This immense amount of data is suited well for utilization by deep learning technologies to help find malicious activity buried within the haystack of network traffic and log data on an enterprise network.

As described earlier, the underlying technology and algorithms in deep learning are based on the utilization of neural network architectures consisting of multiple layers of neurons. In the next section, we will provide some background on neural networks, then describe distinctions inherent within deep neural network architectures (Goodfellow et al., 2016).

2.5.2 Deep Learning Architecture and its Applications

There are two major considerations while working with deep learning: Processing is multilayer non-linear and learning form can be supervised or unsupervised. The most popular architectures in use today to create deep learning models are Convolution Neural Network, Deep Belief Network and Recurrent Neural Network and more recently Autoencoders. For general classification problems,

Deep Belief Network is widely used. Convolution Neural Network is one of the most popular deep learning architectures used for classification of image, text and sound.

Moreover, Recurrent Neural Network is used when data is more in the form of sequential (Bengio, 2019).

2.5.2.1 Convolutional Neural Network (CNN)

Usually, deep neural networks are trained by using large amount of data and learn features directly from data without manual extraction of features. Convolutional Neural Network (CNN) is one of the most widespread deep neural network models that owns the capability of learning features automatically from input data. It is a special class of feed forward neural network and as it eliminates manual feature extraction, it is mostly used for image classification and makes the deep learning highly accurate. Unlike machine learning, manual extraction of features from images are not required and it is capable to perform an entire task like classification of images without manual intervention (Zhang et al., 2018; Nie et al., 2016).

The figure below illustrated an architecture of CNN that composed of convolutional layers, pooling layers and fully connected layers.

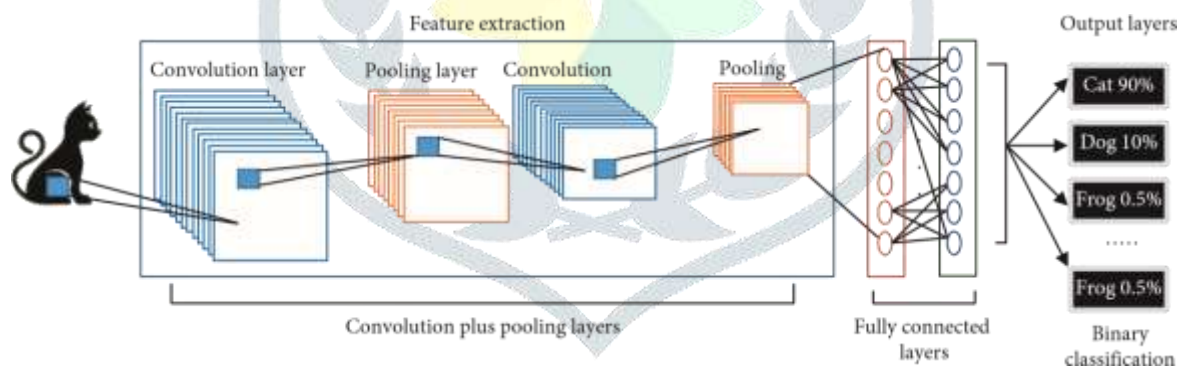


Figure 2.3: Structure of Convolutional Neural Network (Zhang et al., 2018)

2.5.2.2 Deep Belief Network (DBN)

Deep belief network (DBN) is made up with multiple layers of restricted Boltzmann machine (RBM) and represents as a graphical model. It contains many layers of hidden variables, having binary values and called hidden units or feature detectors (Kang and Kang, 2016), and typically represents as a stack of RBMs. There exists a symmetric connection between the top two layers and forms an associative memory. However, the lower layers consume top-down and directed connections. Two

distinct Neural Network types are contained by DBN - they are Belief Networks and RBM. The training to the DBN occurs in two stages: unsupervised pretraining with unlabeled samples and then supervised fine-tuning with labeled samples. The noticeable use of Deep Belief Networks (DBN) is in the field of Natural Language Processing (NLP). It is complex task to process unstructured mass of text collected from Web. DBN has been applied for extracting attributes of entities with accuracy and with marginal manual interference (Jones, 2013; Kang and Kang, 2016; Hebbo and Kim, 2013; Sarikaya et al., 2014; Zhong et al., 2016).

The figure below displays the general architecture of DBN.

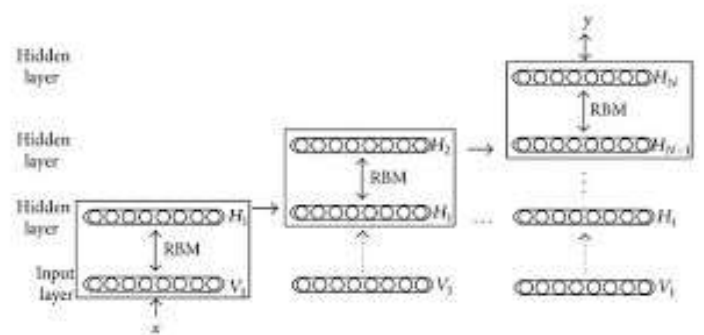


Figure 2.4: Architecture of Deep Belief Network (Hebbo and Kim, 2013).

2.5.2.3 Recurrent Neural Network (RNN)

Recurrent Neural Network (RNN) is an architecture of deep learning that is mostly used to process sequential data. With it, unidirectional cycle is formed between units and unlike traditional feed-forward neural network, it shows better performance in the area of Natural Language Processing, text and speech recognition. RNN can be constructs as depth as the size of the input data series. RNN is capable of accumulating past information as it possesses “memory” and the widely used type of RNN is LSTM (Long short-term memory) (Lee et al., 2017).

The figure below represented the architecture of RNN.

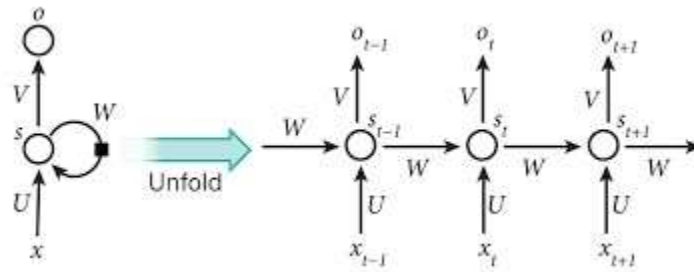


Figure 2.5: Architecture of Recurrent Neural Network (Lee et al., 2017).

2.5.3 How Deep Learning Works and its Application

According to Chollet, (2018), the specification of what a layer does to its input data is stored in the layer’s weight, which in essence are a bunch of numbers. In technical terms, the transformation implemented by a layer is parameterized by its weights. In this context, learning means finding a set of values for the weights of all layers in a network, such that the network will correctly map example inputs to their associated targets. Interestingly, a deep neural network is capable of handling tens of millions of parameters and finding the correct value for all of them may always seem like a daunting task, especially given that modifying the value of one parameter would likely affect the behavior of all the others.

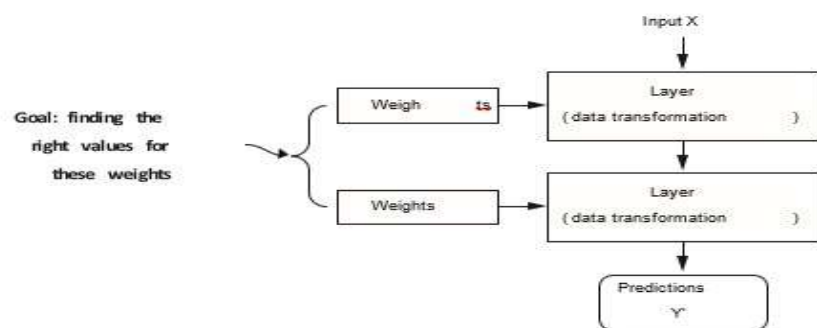


Figure 2.6: Neural Network Parameterized by its Weight (Chollet, 2018)

To control something, first you need to be able to observe it. To control the output of a neural network, you need to be able to measure how far this output is from what you expected. This is the job of the loss function of the network, also called the objective function. The loss function takes the predictions of the network and the true target (what you wanted the network to output) and computes a distance score, capturing how well the network accomplish a given task (Source: Chollet, 2018).

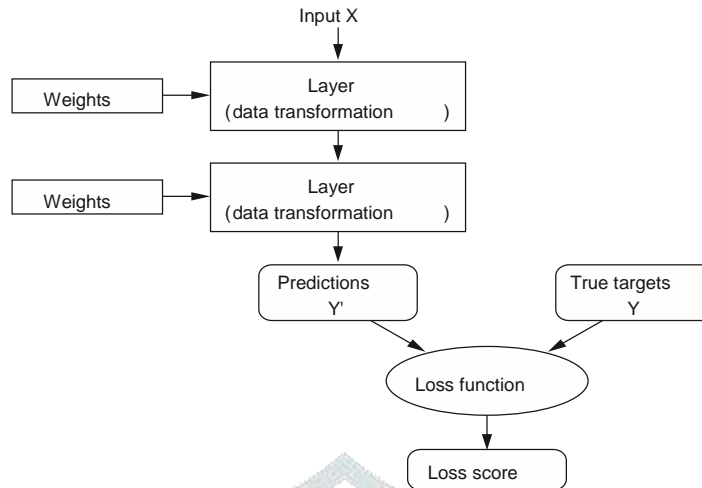


Figure 2.7: A Loss Function Measures the Quality of the Network’s Output (Chollet, 2018)

The fundamental trick in deep learning is to use this score as a feedback signal to adjust the value of the weights a little, in a direction that will lower the loss score for the current step. This adjustment is the job of the optimizer, which implements what’s called the Backpropagation algorithm: the central algorithm in deep learning (Source: Chollet, 2018).

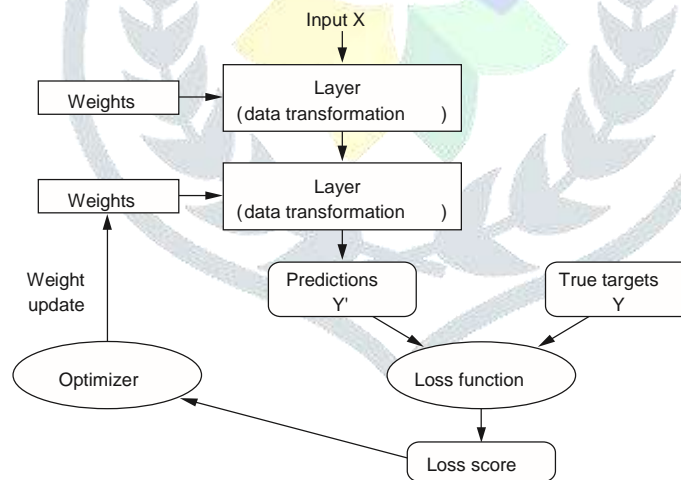


Figure 2.8: The Loss Score is used as a Feedback to Adjust the Weights (Chollet, 2018)

Initially, the weights of the network are assigned random values, so the network merely implements a series of random transformations. Naturally, its output is far from what it should ideally be, and the loss score is accordingly very high. But with every example the network processes, the weights are adjusted a little in the correct direction, and the loss score decreases. This is the training

loop, which, repeated a sufficient number of times (typically tens of iterations over thousands of examples), yields weight values that minimize the loss function. A network with a minimal loss is one for which the outputs are as close as they can be to the targets: a trained network. Once again, it's a simple mechanism that, once scaled, ends up looking like magic (Source: Chollet, 2018).

2.5.4 Deep Learning Toolkit and Libraries

There is not a single criterion for determining the best toolkit for deep learning. Each toolkit is designed and built to address the needs perceived by the developer(s) and also reflects their skills and approaches to problems (Kalogeiton et al., 2016). Therefore, in this research, we will attempt to objectively assess the various toolkit using a range of different criteria.

1. Caffe

Caffe is one of the most mature toolkits, and was developed by Berkeley Vision and Learning Center. It is modular and fast and supports multiple Graphics processing units (GPUs) with little extra effort. It uses JavaScript object notation (JSON)-like text file to describe the network architecture as well as the solver methods. Also has model zoo, which is a website where you can download Caffe models as well as network weights. This can help you get going very quickly with examples. However, tuning hyperparameters is more tedious than other toolkits, in part because a different solver and model file needs to be separately defined for each set of hyperparameters (BAIR, 2018).

2. TensorFlow

TensorFlow is a rather new library that was developed by Google, but already has strong adoption. Performance is good, and supports multiple GPUs and CPUs. Some view it as more difficult to use directly, but there are currently tools available online to address this challenge.

TensorFlow provides tools for tuning a network and monitoring performance like Tensorboard (TensorFlow 2018). In this research, TensorFlow would be used and discourse in detail in the subsequent chapter.

3. Theano

Theano is a tool for creating networks using symbolic logic, and is written in Python, but takes advantage of the efficient code base of NumPy, which improves performance over standard Python. The symbolic approach may be a challenge for some to learn, but Theano is good for building networks, but more challenging to create complete solutions. Theano includes computation of the gradients used in learning as a “free” byproduct of net creation, which may be useful for those wishing to focus more on network architecture than gradient computations.

Documentation quality is fair (Erickson et al., 2017).

4. Keras

Keras is a library written in Python that is mostly utilizes as backend either Theano or Tensorflow. It is easier to build complete solutions, and is easy to read, in that each line of code creates one layer of a network. This toolkit seems to have the greatest selection of state-of-the-art algorithms (optimizers, normalization routines, activation functions). Although Keras supports both Theano and Tensorflow backends, the assumption for the dimension of the input data is different and careful design is needed in order for the code to be able to work using both back ends. The project is well documented and a set of examples aiming at a wide variety of problems is provided. Pretrained models of commonly used architectures for transfer learning implementation are also provided (Keras 2018). Keras will also be used extensively throughout the design and implementation phase of this research.

5. MXNet

MXNet is a deep learning framework written in C++ with many language bindings, and supports distributed computing, including multi-GPU. It provides access to both lower-level constructs as well as higher/symbolic level API. Performance is considered to be on par with other good systems, including Tensorflow, Caffe (Chen et al. 2016) etc.

6. PyTorch

PyTorch is a Python library for GPU-accelerated Deep learning (PyTorch 2018). The library is a Python interface of the same optimized C libraries that Torch uses. PyTorch is written in Python, C and CUDA. The library integrates acceleration libraries such as Intel MKL and NVIDIA (cuDNN, NCCL). At the core, it uses CPU and GPU Tensor and NN backends (TH, THC, THNN, THCUNN) written as independent libraries on a C99 API. PyTorch supports tensor computation with strong GPU acceleration, and DNNs built on a tape-based autograd system. It has become popular by allowing complex architectures to be built easily.

7. Chainer

Chainer is a bit different from other toolkits because it builds the network as part of its computation. The authors describe it as “Define-then-run” which means you define the architecture and then run it. Chainer attempts to build and optimize its architecture as part of the learning process, or as they call it “Define-then-run.” Chainer stores its computations rather than the programming logic. This allows it to fully leverage the power of Python (Chainer 2018).

8. Cognitive Network Toolkit (CNTK)

Microsoft Cognitive Toolkit (CNTK) is a commercial grade distributed Deep learning framework with large-scale datasets from Microsoft Research (CNTK 2018). It implements efficient DNNs training for speech, image, handwriting and text data. Its network is specified as a symbolic graph of vector operations, such as matrix add/multiply or convolution with building blocks (operations). CNTK supports FFNN, CNN, RNN architectures and implements stochastic gradient descent (SGD) learning with automatic differentiation and parallelization across multiple GPUs and servers.

For this study, Tensorflow keras, which has been merged together would be used extensively as the core tool for the design and implementation of the main framework.

2.5.5 Artificial Intelligence and Deep Learning

Artificial Intelligence (AI) is a commonly employed appellation use to refer to the field of science aimed at providing machines with the capacity of performing functions such as logic, reasoning, planning, learning, and perception. AI has been described as software that behaves in some limited ways like a human being (Goertz, 2014). The word artificial comes from the Latin root words *facere arte* which means “make something”; thus, AI translates loosely to manmade intelligence. AI has been defined in many ways and one of such definitions refers to it as a simulation of human intelligence in machines that are programmed to think like humans and mimic their actions. The term may also be applied to any machine that exhibits traits associated with a human mind such as learning and problem-solving (Baum et al., 2011; Brézillon, 2011; Goertz, 2014).

While artificial intelligence techniques have only recently been introduced in the financial industry, they have a long history of application in other fields. Experience to date across a wide range of non-financial applications has been mixed. Patrick Winston, a leading AI researcher and the head of MIT’s AI Laboratory, conceded that the traditional AI methods such as search methods, predicate calculus, rule-based expert systems and game-playing, have achieved little progress (Goertz, 2014). Probably, over the years the problem domain area that traditional AI methods seem to fail in is in the trivial and common sense-type of tasks that humans find easy, such as recognizing faces, identifying objects and walking. Nowadays, there has been much progress with the introduction of deep learning and algorithmic applications that helps AI models to capture, learn and store in memory information that helps them to recognize and identify previously contacted objects (Bostrom and Shulman, 2016).

As a result, many of the contemporary artificial intelligence tools developed in the natural sciences and engineering field have successfully found their way into the commercial world. These include wavelet transformations and finite impulse response filters (FIR) from the signal processing/electrical engineering field; genetic algorithms and artificial neural networks from the biological sciences; and, chaos theory and simulated annealing from the physical sciences. These revolutionary techniques fall under the AI field as they represent ideas that seem to emulate

intelligence in their approach to solving commercial problems (Goertz, 2014). All these AI tools have a common thread in that they attempt to solve problems such as the forecasting and explanation of financial data by applying physical laws and processes. Bostrom and Yudkowsky (2011) state that these novel modes of computation are collectively known as soft computing as they have the unique characteristic of being able to exploit the tolerance imprecision and uncertainty in real world problems to achieve tractability, robustness, and low cost. They further state that soft computing is often used to find an approximate solution to a precisely (or imprecisely) formulated problem. These contemporary tools are often used in combination with one another as well as with more traditional AI methods such as expert systems to obtain better solutions (Bostrom and Shulman, 2016).

2.5.6 Application Areas of Artificial Intelligence and Deep Learning

Artificial Intelligence and Deep Learning have moved well beyond science fiction into cutting edge of internet and enterprise computing. Access to more computational power, advancement of sophisticated algorithms, and the availability of funding are unlocking new possibilities in the field of AI. After decades of experiencing a slow bum, artificial intelligence innovation has caught fire to become the hottest item on the agendas of the world's top technology firms (Stafford, 2010). This is partly due to the recent onslaught of data and the technological improvement of systems that learns and adapts to the field of AI. This flurry of AI advancement wouldn't have been possible without the confluence of three factors that combined to create the right equation for AI growth: the rise of big data combined with the emergence of powerful graphics processing unit (GPUs) for complex computations and the re-emergence of a decade -old AI computation model – deep learning (Stafford, 2010).

2.5.6.1 Deep Learning in Robotics

The field of robotics is at a very exciting point. Owing partly to advances in algorithms and increasing computational power, robots are poised to move out of the lab and other special purpose applications such as manufacturing, and into our everyday lives. This can be seen especially from the

recent upsurge in interest of wide array of human-like walking and running, teaching by demonstration, mobile navigation in pedestrian environments robots to collaborative automation, automated bin/shelf picking, automated combat recovery and automated aircraft inspection and maintenance, and disaster mitigation and recovery robots. Just like how a human brain processes input from past experiences, current input from senses and any additional data that is provided, deep learning models helps robots execute tasks based on the input of many different opinions.

The growth of deep learning models has accelerated, and more so created more innovative applications (Hanson, 2018). For example, in the field of robotics there are currently powerful variations in place using deep learning models. Such innovations include and not limited to Hanson Robotics' most advanced human-like robot, Sophia. Sophia's AI combines cutting-edge work in symbolic AI, neural networks, expert systems, machine perception, conversational natural language processing, adaptive motor control and cognitive architecture among others. Robot Sophia personifies the dreams for the future of AI. As a unique combination of science, engineering, and artistry, Sophia is simultaneously a human-crafted science fiction character depicting the future of AI and robotics, and a platform for advanced robotics and AI research. The character of Sophia captures the imagination of global audiences. She is the world's first robot with citizenship status of Saudi Arabia and the first robot Innovation Ambassador for the United Nations Development Programme. Robot Sophia is now a household name in Saudi Arabia and probably the world at large. Sophia is a framework for cutting edge robotics and AI research, particularly for understanding human-robot interactions and their potential service and entertainment applications. For example, she has been used for research as part of AI deep learning project, which seeks to understand how robots can adapt to users' needs through intra and interpersonal development (Hanson, 2018).

According to Makoto, (2008), another distinct concept in the field of robotics is the Robot Assistance to the Elderly in Japan. Japan is known for being a technology-driven country. Since the Meiji restoration (1868-1912), change has been connected with technology. During the period of rapid economic growth, the labor shortage was mainly compensated for by the implementation of industrial

robots. The recent robot development in the field of care covers the following areas: care assistance, interaction and therapy. Technologies for care assistance aim to reduce the burden for the nursing staff and improve the quality in care through e.g., lifting systems like the polar bear robot RIBA. Robots for interaction are developed to entertain or communicate with humans e.g., are NAO, Papero or KOBIAN-R. Technologies for medical purposes like rehabilitation and therapy are one very promising field in robotics. Through the utilization of robots for therapy or rehabilitation by using e.g., the robot seal Paro or the robo-skeleton HAL, psychical and physical conditions can be improved.

Similarly, one of the most famous humanoid robots is ASIMO made by Honda. ASIMO is said to be one of the world's most advanced robots ever and is the research result of over two decades. The latest ASIMO is 130 cm tall and weighs 54 kg. Furthermore, he can walk, ride a bike and transport things and has 36 degrees of freedom (Honda, 2013). Through extensive travels by ASIMO to different countries, it has become a perfect PR ambassador for Japan and its advanced technologies. Another Japanese robot that enjoys great media attention is the humanoid robot RIBA-II. Already his predecessors RI-MAN and RIBA were equipped with visual, olfactory, auditory and tactile sensors (Mukai et al., 2009) and able to lift and carry people. RIBA-II is currently used in hospitals and nursing homes. The robot has successfully relieved the physical burden of the nursing staff by moving people out of bed and into wheelchairs, and vice versa. The project is a collaboration between RIKEN and Tokai Rubber Industries, who together established the RIKEN-TRI Collaboration Center for Human-Interactive Robot Research. The most noticeable difference to its predecessor is that its design was not inspired by a human but by a polar bear (Sato et al., 2012).

2.5.6.2 Deep Learning Approaches used in Medicine (Radiology)

In medical imaging, machine learning algorithms have been used for decades, starting with algorithms to analyze or help interpret radiographic images in the mid-1960s. Meyers et al., (1964) Computer-aided detection/diagnosis (CAD) algorithms started to make advances in the mid-1980s, first with algorithms dedicated to cancer detection and diagnosis on chest radiographs and

mammograms, (Chan et al., 1987) and then widening in scope to other modalities such as computed tomography (CT) and ultrasound. CAD algorithms in the early days predominantly used a data-driven approach as most DL algorithms do today. However, unlike most DL algorithms, most of these early CAD methods heavily depended on feature engineering. A typical workflow for developing an algorithm for a new task consisted of understanding what types of imaging and clinical evidence clinicians use for the interpretation task, translating that knowledge into computer code to automatically extract relevant features, and then using machine learning algorithms to combine the features into a computer score. There were, however, some notable exceptions. Inspired by the Neocognitron architecture, a number of researchers investigated the use of CNNs or shift-invariant ANNs in the early and mid-1990s, and massively trained artificial neural networks (MTANNs) in the 2000s for detection and characterization tasks in medical imaging.

These methods all shared common properties with current deep CNNs (DCNNs (Chan et al., 1987).

With the advent of DL, applications of machine learning in medical imaging have dramatically increased, paralleling other scientific domains such as natural image and speech processing. Investigations accelerated not only in traditional machine learning topics such as segmentation, lesion detection, and classification (Greenspan et al., 2016), but also in other areas such as image reconstruction and artifact reduction that were previously not considered as datadriven topics of investigation

Deep learning for reconstruction mostly relies on data distribution to learn a function that maps input to output. Without a doubt, the most time-consuming process in iterative reconstructions for accelerated Magnetic Resonance (MR) data reconstruction is calculating the gradient of the object function with respect to the variables (Zhu et al., 2018). In the deep learning approach, it is not necessary to calculate the gradient for the forwarding step; the gradient calculation is only done for the back-propagation step in network training. Additionally, the number of deep learning parameters is designed to be much smaller than the number describing an arbitrary mapping from images to images. This dimension reduction can be attributed to the rationale of manifold approximation. A

recent study demonstrated the manifold property of human brain images and proposed a neural network that transforms arbitrarily encoded k-space into images (Zhu et al., 2018). This manifold approximation was additionally validated in other studies. Another group proposed a concept of residual labeling to facilitate manifold learning and explained the principles of manifold learning in a theoretical manner (Ye et al., 2018). Denoising is one of the most important aspects of image quality improvement. In practice, generally, the MR signals are always perturbed by various unwanted noises, and image denoising can be considered as an inverse problem of finding the values of the signal minimizing noise contaminations. The conventional image denoising methods are model-based methods, followed by sparse coding, effective prior, and low-rank approaches. Combined with the knowledge of the conventional methods, the deep learning methods are reported to show superior performances (Ye et al., 2018).

Furthermore, the segmentation of MR images is an essential step for the quantitative assessment of various applications such as identifying the margins of a lesion for surgical planning or measuring the volumes of the organs for a population study. The typical image segmentation algorithm depends on the spatial properties of image intensity values. Specifically, discontinuity and similarity are the key properties for the segmentation of a specific object. However, it is difficult to establish a generalized method for the intensity-based segmentation, because the image intensities of most MR images are not quantitative and are largely influenced by environmental factors such as imaging hardware, protocol, and noises. Although several successful automatic segmentation algorithms have been developed and used in specific applications, such as brain segmentation from T1- weighted images (Ye et al., 2018), these methods involve complex and extensive processing steps (which are sensitive to input variations) and often require manual interventions for abnormal cases. In addition, there is still a lack of robust algorithms available in many areas requiring segmentation tasks. Further, advances in deep learning architectures such as U-net (99) or DeepLab (9) and large amounts of image data are expected to solve the limitations of traditional methods and improve the performance in MR image segmentation applications.

Recently, deep learning methods have shown the best performances in most contests dealing with MR image segmentations, such as the brain tumor segmentation challenge. Recent methods for MR image segmentations have mostly used 3D operations to reflect the object's spatial context in 3D space (Ye et al., 2018).

Lesion detection of abnormalities (including tumors and other suspicious growths) in medical images is a common but costly and time-consuming part of the daily routine of physicians, especially radiologists and pathologists. Given that the location is often not known a priori, the physician should search across the 2D image or 3D volume to find deviations compared to surrounding tissue and then to determine whether that deviation constitutes an abnormality that requires follow-up procedures or something that can be dismissed from further investigation. This is often a difficult task that can lead to errors in many situations either due to the vast amount of data that needs to be searched to find the abnormality (e.g., in the case of volumetric data or whole slide images) or because of the visual similarity of the abnormal tissue with normal tissue (e.g., in the case of low-contrast lesions in mammography). Automated computer detection algorithms have therefore been of great interest in the research community for many years due to their potential for reducing reading costs, shortening reading times and thereby streamlining the clinical workflow, and providing quality care for those living in remote areas who have limited access to specialists (Zhu et al., 2018).

Traditional lesion detection systems often consist of long processing pipelines with many different steps. Some of the typical steps include preprocessing the input data, for example, by rescaling the pixel values or removing irrelevant parts of the image, identification of locations in the image that are similar to the object of interest according to rule-based methods, extraction of handcrafted features, and classification of the candidate locations using a classifier such as SVM or RF. In comparison, DL approaches for lesion detection are able to avoid the time-consuming pipeline design approach (Zhu et al., 2018).

2.5.6.3 Natural Language Processing (NLP)

According to Young (2018), understanding the complexities associated with language whether it is syntax, semantics, tonal nuances, expressions, or even sarcasm, is one of the hardest tasks for humans to learn. Constant training since birth and exposure to different social settings help humans develop appropriate responses and a personalized form of expression to every scenario. Natural Language Processing through Deep Learning is trying to achieve the same thing by training machines to catch linguistic nuances and frame appropriate responses. Document summarization is widely being used and tested in the Legal sphere making paralegals obsolete. Answering questions, language modelling, classifying text, twitter analysis, or sentiment analysis at a broader level are all subsets of natural language processing where deep learning is gaining momentum. Earlier logistic regression or SVM were used to build time-consuming complex models but now distributed representations, convolutional neural networks, recurrent and recursive neural networks, reinforcement learning, and memory augmenting strategies are helping achieve greater maturity in NLP. Distributed representations are particularly effective in producing linear semantic relationships used to build phrases and sentences and capturing local word semantics with word embedding (Pumsirirat and Yan, 2018).

2.6 Neural Networks (NN)

Neural networks, or more precisely artificial neural network (ANN), are an AI technique that takes inspiration from and mimics the method and functioning of the human brain. It relies on a modular system composed of interconnected basic computational units called neurons that performs a simple mathematical operation to produce an output given a set of inputs. The outputs of each neuron can be either used as output of the network, or be employed as input for other neurons, and the relationships among neurons are adjusted through weights that, together with the architecture of the network, define a learned model (Fausett, 1994).

There is no universally accepted definition of a neural network. But majority agrees that a neural network is a network of many simple processors (units), each possibly, having a small amount of

local memory. The units are connected by communication channels (connections) which usually carry numeric (as opposed to symbolic) data, encoded by any of various means. The units operate only on their local data and on the inputs, they receive via the connections. Most neural networks have some sort of “training” rule whereby the weights of connections are adjusted on the basis of data. In other words, NNs “learn from examples and exhibit some capability for generalization beyond the training data (Haykin, 1994).

According to Topping and Khan, (1993), a neural network is a system that is composed of many simple processing elements operating in parallel, whose function is determined by network structure, connection strengths, and the processing performed at computing elements or nodes. A comprehensive working definition is found in the book “An Introduction to Neural Networks” by Gurney (1997), in his book, he delineates that a neural network is an interconnected assembly of simple processing elements, units or nodes, whose functionality is loosely based on the animal neuron. The processing ability of the network is stored in the interunit connection strengths, or weights, obtained by a process of adaptation to, or learning from, a set of training patterns.

With the above definitions, it is worth summarizing that Artificial Neural Network (ANN) is an information processing paradigm that is inspired by the way biological nervous systems, such as the brain, process information. The key element of this paradigm is the novel structure of the information processing system. It is composed of a large number of highly interconnected processing elements (neurons) working in unison to solve specific problems. ANNs, like people, learn by example. An ANN is configured for a specific application, such as pattern recognition or data classification, through a learning process.

There are distinct types of neural networks, and they can be categorized by the following (Fausett, 1994);

- i. Architecture, or pattern by which neurons are connected
- ii. Learning algorithm, or the way in which values for weights on the communication links are determined.

iii. Activation function(s) used by the individual layers of the neural network.

Each individual neuron maintains its own internal state, which is determined by the activation function applied to its inputs. The neuron sends its activation to all the other neurons to which it is directly connected to downstream in the next layer of the network. Common activation functions for a neuron include the sigmoid function, and more recently the rectified linear unit (ReLU) function. Before analyzing possible network architectures and properties, it is necessary to analyze the network's more basic component: the neuron, of which, the perceptron is the most common model.

The perceptron, introduced by Rosenblatt as a statistical modelling of the functioning of the brain (Rosenblatt, 1958), is a simple unit that performs an input/output mapping, capable to learn linearly separable decision boundaries when combined with a learning procedure inspired by the Hebbian rule.

When referring to an ANN, the concept of layers plays an important role and corresponds to a set of neural units that share the same distance (in terms of computational steps) from the input and from the output of the model, and are not connected between them by weights. Each layer, exception made for the output layer that is trained to learn the final mapping, is employed to transform the output of the previous one in a new different feature space, being equivalent to consecutive applications of learned feature transformations. A simple example of artificial neural network, is composed by the input layer, a number of perceptron units, and finally the output layer. All the layers that are not direct expression of either the input or output of the network are referred to as hidden layers. The number of layers composing a network is used to define the concept of depth of the network (Fausett, 1994).

Layers in a neural network can be stacked, in order to learn feature transformations for the input that allows it to better approximate the desired target function. Depending on the architecture of the network, different kinds of features can be extracted, resulting in different tasks for which the neural network becomes more suitable. Examples of different kind of classes of network are; feedforward neural networks, in which the layers are connected without loops and links only appear

from neurons of a layer to neurons of the next one, convolutional neural networks and recurrent neural networks (both networks have been explained earlier under deep learning architecture). Feedforward neural networks are the basic structure that a multi-layer neural network can assume and are mathematically interpreted as the consecutive application to the input of learned feature transformations, thus, being equivalent to the application of a composite function. Each layer is equivalent to the application of an affine transformation to its input, in which the non-linearity is introduced by the activation function employed in the neurons (Fausett, 1994).

There are four distinct current areas of research in ANNs according to Gurney (1997);

- i. Using ANNs to model the biological networks in order to gain understanding of the human brain and its functions. This area is of particular interest to psychologists and researchers in neuroanatomy.
- ii. Using ANNs as an educational tool in order to gain understanding on how to solve complex tasks that traditional AI methodologies and computer algorithms have had difficulty in solving. Researchers in this area include computer scientists and engineers, who are mainly interested in constructing better computer algorithms by studying the problem-solving process of an ANN.
- iii. Using ANNs to solve real-world types of problems in various commercial applications. Many researchers in this area have backgrounds in areas other than those related to ANN. The attraction of using an ANN is the simplicity in using it as a tool and the reported ANNbased commercial application successes. There are many ANN software packages that are user-friendly enough for new users to start using without requiring them to have an indepth knowledge of the ANN algorithms. This is unlike conventional computer techniques which require a user to thoroughly understand the algorithm before writing a program to apply it. In the case of ANNs, all a user needs to know is how to present the problem at hand in a form that an ANN can understand.
- iv. Improving ANN algorithms. Researchers in this field are interested in constructing better

ANN algorithms that can ‘learn’ or model more efficiently; i.e., quicker training times and/or more accurate results.

2.6.1 Applications of Neural Networks

Gurney (1997) further argue that most ANN applications fall into the following three categories:

1. Pattern classification,
2. Prediction and financial analysis, and
3. Control and Optimization

In practice, their categorization is ambiguous since many financial and predictive applications involve pattern classification. A preferred classification that separates applications by method is the following:

1. Classification,
2. Time Series, and
3. Optimization.

2.6.1.1 Classification

Classification problems involve either binary decisions or multiple class identification in which observations are separated into categories according to specified characteristics. They typically use cross sectional data. Obviously, the interdependence of the observations needs to be considered as most real-world problems may contain observations that do not fall directly into the defined categories. Solving these problems entails ‘learning’ patterns in a data set and constructing a model that can recognize these patterns. Commercial artificial neural network applications of this nature include:

- i. Credit card fraud detection being used by Eurocard Nederland, Mellon Bank, First USA Bank.

- ii. Reducing credit card application processing time by processing scanned images of handprinted data to machine readable text.
- iii. Prediction of vehicle driving comfort to assist engineers in the design process.
- iv. Assisting targeting donors in fund-raising activities by predicting long term profitability of each donor on an organization's house file.
- v. An automated antivirus system for computer networks called Immune System, developed by IBM.
- vi. Identifying property sites for retail outlets.
- vii. Optical character recognition (OCR) utilized by fax software such as Calera Recognition System's FaxGrabber and Caere Corporation's Anyfax OCR engine that is licensed to other products such as the popular WinFax Pro and FaxMaster.

2.6.1.2 Time Series

In time-series problems, the ANN is required to build a forecasting model from the historical data set to predict future data points. Since the sequence of the input data in this type of problem is important in determining the relationship of one pattern of data to the next, they require relatively sophisticated ANN techniques. This is known as the temporal effect, and more advanced techniques such as finite impulse response (FIR) types of ANN and recurrent ANNs are being explored and developed to deal specifically with this type of problem.

2.6.1.3 Optimization

Optimization problems involve finding a solution for a set of very difficult problems known as Non-Polynomial (NP) complete problems. Polynomial problems are problems, if given an appropriate algorithm, can be solved in some polynomial time. On the other hand, solutions to NPcomplete problems are not easy to find, with no known mathematical formula or algorithm to determine the solutions. According to the Neuralware Manual (1991), the known solutions to this class of problems have computation time which increases exponentially with the number of inputs. In other words, the time needed to solve the problem cannot be bounded by a polynomial function of the inputs.

Combinatorial-type problems can be either Polynomial or NP-complete, depending on the problem at hand.

Examples of problems of this type include the traveling salesman problem, job scheduling in manufacturing and efficient routing problems involving vehicles or telecommunication. An AI technique, called heuristic, is often used to find ‘good’ or ‘acceptable’ solutions for such problems. Determination of optimal solutions for this class of problems is almost always impossible. An example of a heuristic rule in the case of the traveling salesman problem, is to visit the nearest unvisited city from the current city. This rule is also commonly known as a variant of the “greedy algorithm”. The ANNs used to solve such problems are conceptually different from the previous two categories (classification and time-series), in that they require unsupervised networks, whereby the ANN is not provided with any prior solutions and thus has to ‘learn’ by itself without the benefit of known patterns (Gurney, 1997).

2.6.2 Biological Background

According to Aggarwal, (2018) ANNs were inspired by the biological sciences, particularly the neurological sciences. However, ANNs resemblance to their biological counterparts are limited to some borrowed concepts from the biological networks, mainly for their architecture. They are still far from resembling the workings of the simplest biological networks, due to the enormous complexity of the biological networks.

The cells found in the human brain and nervous system are known as neurons. Information or signals are transmitted out unidirectional through connections between neurons known as axons. Information is received by a neuron through its dendrites. The human brain consists of around 100 billion neurons and over 10¹⁴ synapses. Neurons communicate with each other through synapses which are gaps or junctions between the connections. The transmitting side of the synapses release neurotransmitters which are paired to the neuroreceptors on the receiving side of the synapses. Learning is usually done by adjusting existing synapses, though some learning and memory functions are carried out by creating new synapses. In the human brain, neurons are organized in clusters and

only several thousand or hundreds of thousands participate in any given task. Figure below shows a sample neurobiological structure of a neuron and its connections.

The axon of a neuron is the output path of a neuron that branches out through axon collaterals which in turn connect to the dendrites or input paths of neurons through a junction or a gap known as the synapse. It is through these synapses that most learning is carried out by either exciting or inhibiting their associated neuron activity. However, not all neurons are adaptive or plastic. Synapses contain neurotransmitters that are released according to the incoming signals. The synapses excite or inhibit their associated neuron activity depending on the neurotransmitters released. A biological neuron will add up all the activating signals and subtract all the inhibiting signals from all of its synapses. It will only send out a signal to its axon if the difference is higher than its threshold of activation. The processing in the biological brain is highly parallel and is also very fault tolerant. The fault tolerance characteristic is a result of the neural pathways being very redundant and information being spread throughout synapses in the brain. This wide distribution of information also allows the neural pathways to deal well with noisy data.

A biological neuron is so complex that current super computers cannot even model a single neuron. Researchers have therefore simplified neuron models in designing ANNs.

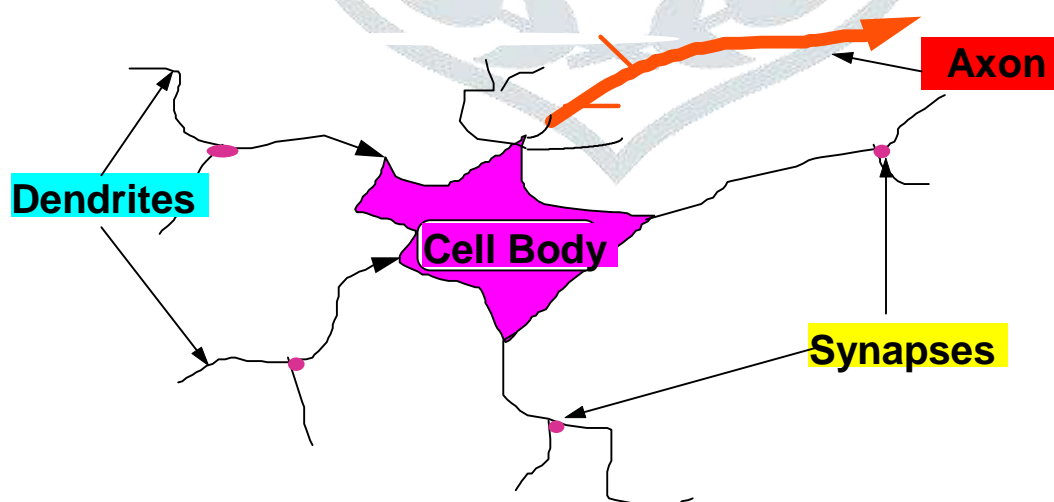


Figure 2.9: A Typical Biological Neuron (Aggarwal, 2018)

2.6.3 Strengths and Weaknesses of Artificial Neural Networks (ANN)

Aggarwal, (2018), highlights the strengths and weaknesses of Artificial Neural Networks are described below:

ANNs are easy to construct and deal very well with large amounts of noisy data. They are especially suited to solving nonlinear problems. They work well for problems where domain experts may be unavailable or where there are no known rules. ANNs are also adaptive in nature.

This makes them particularly useful in fields such as finance where the environment is potentially volatile and dynamic.

They are also very tolerant of noisy and incomplete data sets. Their robustness in storing and processing data, earned them some applications in space exploration by NASA, where fault tolerant types of equipment are required. This flexibility derives from the fact that information is duplicated many times over in the many complex and intricate network connections in ANNs, just like in the human brain. This feature of ANNs is, in contrast to the serial computer¹⁴ where if one piece of information is lost, the entire information set may be corrupted.

The training process of an ANN itself is relatively simple. The pre-processing of the data, however, including the data selection and representation to the ANN and the post processing of the outputs (required for interpretation of the output and performance evaluation) require a significant amount of work. However, constructing a problem with ANNs is still perceived to be easier than modeling with conventional statistical methods. There are many statisticians who argue that ANNs are nothing more than special cases of statistical models, and thus the rigid restrictions that apply to those models must also be applied to ANNs as well. However, there are probably more successful novel applications using ANNs than conventional statistical tools. The prolific number of ANN's applications in a relatively short time could be explained by the universal appeal of the relatively easy methodology in setting up an ANN to solve a problem. The restrictions imposed by many equivalent statistical models is probably less appealing to many researchers without a strong statistical background. ANN software packages are also relatively easier to use than the typical statistical

packages. Researchers can successfully use ANN's software packages without requiring full understanding of the learning algorithms. This makes them more accessible to a wider variety of researchers. ANN researchers are more likely to learn from experience rather than be guided by statistical rules in constructing a model and thus they may be implicitly aware of the statistical restrictions of their ANN models.

The major weakness of ANNs is their lack of explanation for the models that they create. Research is currently being conducted to unravel the complex network structures that are created by ANN. Even though ANNs are easy to construct, finding a good ANN structure, as well as the pre-processing and post processing of the data, is a very time-consuming process.

2.6.4 Basic Structure of an Artificial Neural Network (ANN)

According to Aggarwal (2018), the basic structure of an ANN consists of artificial neurons (similar to biological neurons in the human brain) that are grouped into layers. The most common ANN structure consists of an input layer, one or more hidden layers and an output layer. A modified simple model of an artificial neuron is shown below.

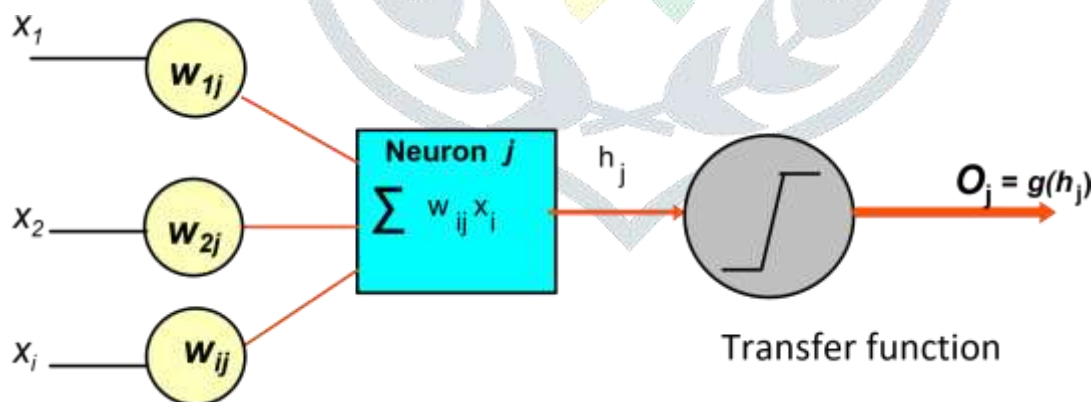


Figure 2.10: An Artificial Neuron (Aggarwal, 2018).

In the human brain, neurons communicate by sending signals to each other through complex connections. ANNs are based on the same principle in an attempt to simulate the learning process of the human brain by using complex algorithms. Every connection has a weight attached which may

have either a positive or a negative value associated with it. Positive weights activate the neuron while negative weights inhibit it. Figure above shows a network structure with inputs

(X_1, X_2, \dots, X_i) being connected to neuron j with weights $(W_{1j}, W_{2j}, \dots, W_{ij})$ on each connection. The neuron sums all the signals it receives, with each signal being multiplied by its associated weights on the connection (Aggarwal, 2018).

This output (h_j) is then passed through a transfer (activation) function, $g(h)$, that is normally non-linear to give the final output O_j . The most commonly used function is the sigmoid (logistic function) because of its easily differentiable properties, which is very convenient when the back-propagation algorithm is applied.

The back-propagation ANN is a feedforward neural network structure that takes the input to the network and multiplies it by the weights on the connections between neurons or nodes; summing their products before passing it through a threshold function to produce an output. The back-propagation algorithm works by minimizing the error between the output and the target (actual) by propagating the error back into the network. The weights on each of the connections between the neurons are changed according to the size of the initial error. The input data are then fed forward again, producing a new output and error. The process is reiterated until an acceptable minimized error is obtained. Each of the neurons uses a transfer function and is fully connected to nodes on the next layer. Once the error reaches an acceptable value, the training is halted. The resulting model is a function that is an internal representation of the output in terms of the inputs at that point (Aggarwal, 2018).

2.6.5 Constructing the Artificial Neural Network (ANN)

Aggarwal (2018) opined that setting up an ANN is essentially a six-step procedure.

First, the data to be used need to be defined and presented to the ANN as a pattern of input data with the desired outcome or target. Second, the data is categorized to be either in the training set or validation (also called test and out-of-sample) set. The ANN only uses the training set in its learning

process in developing the model. The validation set is used to test the model for its predictive ability and when to stop the training of the ANN. Third, the ANN structure is defined by selecting the number of hidden layers to be constructed and the number of neurons for each hidden layer. Fourth, all the ANN parameters are set before starting the training process. Next, the training process is started. The training process involves the computation of the output from the input data and the weights. The backpropagation algorithm is used to 'train' the ANN by adjusting its weights to minimize the difference between the current ANN output and the desired output.

Finally, an evaluation process has to be conducted to determine if the ANN has 'learned' to solve the task at hand. This evaluation process may involve periodically halting the training process and testing its performance until an acceptable result is obtained. When an acceptable result is obtained, the ANN is then deemed to have been trained and ready to be used.

As there are no fixed rules in determining the ANN structure or its parameter values, a large number of ANNs may have to be constructed with different structures and parameters before determining an acceptable model. The trial-and-error process can be tedious and the experience of the ANN user in constructing the networks is invaluable in the search for a good model.

Determining when the training process needs to be halted is of vital importance in obtaining a good model. If an ANN is over trained, a curve-fitting problem may occur whereby the ANN starts to fit itself to the training set instead of creating a generalized model. This typically results in poor predictions of the test and validation data set. On the other hand, if the ANN is not trained for long enough, it may settle at a local minimum, rather than the global minimum solution. This typically generates a sub-optimal model. By performing periodic testing of the ANN on the test set and recording both the results of the training and test data set results, the number of iterations that produce the best model can be obtained. All that is needed is to reset the ANN and train the network up to that number of iterations.

2.6.6 A Brief Description of ANN Parameters

A brief introductory non-technical description of the ANN parameters is described below.

2.6.6.1 Learning Rates

The learning rate determines the amount of correction term that is applied to adjust the neuron weights during training. Small values of the learning rate increase learning time but tend to decrease the chance of overshooting the optimal solution. At the same time, they increase the likelihood of becoming stuck at local minima. Large values of the learning rate may train the network faster, but may result in no learning occurring at all. The adaptive learning rate varies according to the amount of error being generated. The larger the error, the smaller the values and vice-versa. Therefore, if the ANN is heading towards the optimal solution it will accelerate.

Correspondingly, it will decelerate when it is heading away from the optimal solution (Aggarwal, 2018).

2.6.6.2 Momentum

The momentum value determines how much of the previous corrective term should be remembered and carried on in the current training. The larger the momentum value, the more emphasis is placed on the current correction term and the less on previous terms. It serves as a smoothing process that 'brakes' the learning process from heading in an undesirable direction (Aggarwal, 2018)

2.6.6.3 Input Noise

Random noise is used to perturb the error surface of the neural net to jolt it out of local minima. It also helps the ANN to generalize and avoid curve fitting (Aggarwal, 2018).

2.6.7 Training and Testing Tolerances

The training tolerance is the amount of accuracy that the network is required to achieve during its learning stage on the training dataset. The testing tolerance is the accuracy that will determine the predictive result of the ANN on the test dataset (Aggarwal, 2018).

2.6.8 Determining an Evaluation Criterion

It is not always easy to determine proper evaluation criteria in designing an ANN model to solve a particular problem. In designing an ANN to solve a particular problem, special attention needs to be taken in determining the evaluation criteria. This can be done by careful analysis of the problem at hand, the main objective of the whole process and the ANN role in the process (Aggarwal, 2018).

2.9 Artificial Neural Network Models

According to Aggarwal (2018), there are infinitely many ways to organize a neural network although perhaps only two dozen models are in common usage. A neural network organization can be described in terms of its neurodynamics and architecture. Neurodynamics refer to the properties of an individual artificial neuron that consist of the following:

1. Combination of input(s);
2. Production of output(s);
3. Type of transfer (activation) functions; and
4. Weighting schemes, i.e., weight initialization and weight learning algorithms.

These properties can also be applied to the whole network on a system basis. Network architecture (also sometimes referred to as network topology) defines the network structure and includes the following basic characteristics:

- ✦ Types of interconnections among artificial neurons (henceforth referred to as just neurons);
- ✦ Number of neurons and
- ✦ Number of layers
- ✦ Neurodynamics

2.9.1 Inputs

The input layer of an ANN typically functions as a buffer for the inputs, transferring the data to the next layer. Preprocessing the inputs may be required as ANNs deal only with numeric data. This may involve scaling the input data and converting or encoding the input data to a numerical form that can be used by the ANN (Aggarwal, 2018).

2.9.2 Outputs

The output layer of an ANN functions in a similar fashion to the input layer except that it transfers the information from the network to the outside world. Post-processing of the output data is often required to convert the information to a comprehensible and usable form outside the network. The post-processing may be as simple as just a scaling of the outputs ranging to more elaborate processing as in hybrid systems (Aggarwal, 2018).

2.9.3 Transfer (Activation) Function

According to Aggarwal, (2018), the transfer or activation function is a function that determines the output from a summation of the weighted inputs of a neuron. The transfer functions for neurons in the hidden layer are often nonlinear and they provide the nonlinearities for the network. For the example figure 2.10 shows the output of neuron j , after the summation of its weighted inputs from neuron 1 to i has been mapped by the transfer function f can be shown as:

$$O_j = f_j \left(\sum_i w_{ij} x_i \right)$$

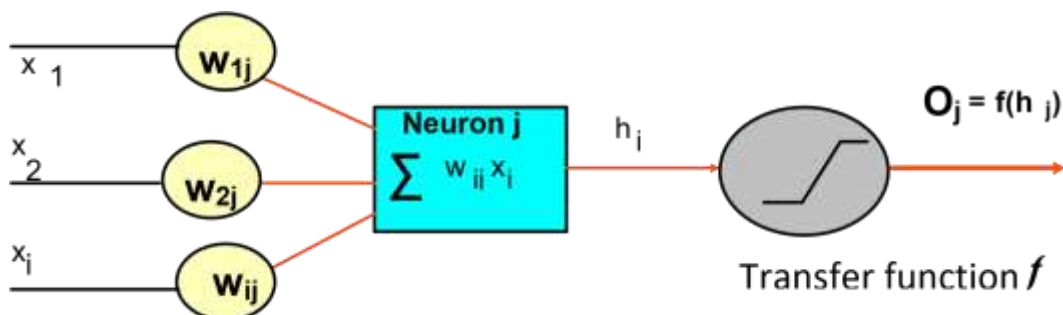


Figure 2.11: Diagram of the Neurodynamics of Neuron j (Aggarwal, 2018)

A transfer function maps any real numbers into a domain normally bounded by 0 to 1 or 1 to 1. Bounded activation functions are often called squashing functions (Aggarwal, 2018). Early ANN models, like the perceptron used, a simple threshold function (also known as a step-function, hard-limiting activation or Heaviside function):

○ Threshold: $f(x) = 0$ if $x < 0$, 1 otherwise.

The most common transfer functions used in current ANN models are the sigmoid (S-shaped) functions. Aggarwal (2018), in his book “Neural Networks and Deep Learning” loosely defined a sigmoid function as a ‘continuous, real-valued function whose domain is the reals, whose derivative is always positive, and whose range is bounded’. Examples of sigmoid functions are:

✦ Logistics: $f(x) = \frac{1}{1 + e^{-x}}$

✦ Hyperbolic tangent: $f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$

The logistic function remains the most commonly applied in ANN models due to the ease of computing its derivative:

$$f'(x) = f(x)(1 - f(x))$$

The output, O_j of the neuron x_j of the earlier example in figure 11 above if the function f is a logistic function becomes:

$$O_j = \frac{1}{1 + e^{-\sum_i W_{yj} x_i - O_j}}$$

Where O_j is the threshold on unit j

If the function f , is a threshold function instead, the output, O_j will be:

$$o_j = \{1, \sum_i w_{yi} x_i > \theta_j\}$$

0, else

However, Sutskever et al., (2014) argue that the hyperbolic tangent function is the ideal transfer function. According to Aggarwal (2018), the shape of the function has little effect on a network although it can have a significant impact on the training speed. Other common transfer functions include:

✦ Linear or identity: $f(x) = x$ normally used in the input and/or output layer.

✦ Gaussian: $f(x) = e^{-x^2/2}$

Sigmoid functions can never reach their theoretical limit values and it is futile to try and train an ANN to achieve these extreme values. Values that are close to the limits should be considered as having reaching those values. For example, in a logistic function where the limits are 0 to 1, a neuron should be considered to be fully activated at values around 0.9 and turned off at around 0.1. This is another reason why ANNs cannot do numerical computation as well or as accurate as simple serial computers; i.e., a calculator (Aggarwal, 2018).

2.10 Types of Interconnections between Neurons

A network is said to be fully connected if the output from a neuron is connected to every other neuron in the next layer. A network with connections that pass outputs in a single direction only to neurons on the next layer is called a feedforward network. Goodfellow et al., (2016) define a feedback network as one that allows its outputs to be inputs to preceding layers. They call networks that work with closed loops as recurrent networks. They also mention networks with feedlateral connections that would send some inputs to other nodes in the same layer. Feedforward networks are faster than feedback nets as they require only a single pass to obtain a solution. According to Goodfellow et al.,

(2016) recurrent networks are used to perform functions like automatic gain control or energy normalization and selecting a maximum in complex systems.

Most ANN books, however, classify networks into two categories only: feedforward networks and recurrent networks. This is done by classifying all networks with feedback connections or loops as recurrent networks. Fully connected feedforward networks are often called multi-layer perceptron's (MLPs) and they are by far the most commonly used ANNs.

2.10.1 The Number of Hidden Neurons

Hidden neurons are required to compute difficult functions known as nonseparable functions. The number of input and output neurons are determined by the application at hand (Goodfellow et al., 2016). However, there are no standard rules or theories in determining the number of neurons in the hidden layers although there are some rules of thumb suggested by various ANN researchers:

Goodfellow et al., (2016) suggested that the network topology should have a pyramidal shape; that is to have the greatest number of neurons in the initial layers and have fewer neurons in the later layers. He suggested the number of neurons in each layer should be a number from mid-way between the previous and succeeding layers to twice the number of the preceding layer. The examples given suggest that a network with 12 neurons in its previous layer and 3 neurons in the succeeding layer should have 6 to 24 neurons in the intermediate layer.

A rough guideline based on theoretical conditions of what is known as the VapnikChervonenkis dimension, recommends that the number of training data should be at least ten times the number of weights. He also quoted a theorem that suggests a network with one hidden layer and $2N+1$ hidden neuron is sufficient for N inputs.

Goodfellow et al., (2016) gives the following formula for determining the number of hidden neurons required in a network:

number of hidden neurons = training facts ´ error tolerance.

Note: training facts refers to in-sample data while the error tolerance refers to the level of accuracy desired or acceptable error range.

2.11 The Multilayer Perceptron

Multilayer Perceptron (MLP) also called a multilayer feedforward network, is an extension of the perceptron model with the addition of hidden layer(s) that have nonlinear transfer functions in the hidden neurons. Usually, having MLPs with one hidden layer is a universal approximator, and is capable of learning any function that is continuous and defined on a compact domain as well as functions that consist of a finite collection of points. According to Aggarwal (2018), the MLPs can also learn many functions that do not meet the above criteria; specifically, discontinuities can be theoretically tolerated and functions that do not have compact support (such as normally distributed random variables) can be learned by a network with one hidden layer under some conditions. Aggarwal states that in practice, a second hidden layer is only required if a function that is continuous has a few discontinuities. He further states the most common reason for an MLP to fail to learn is the violation of the compact domain assumption, i.e., the inputs are not bounded. He concludes that if there is a problem learning in an MLP, it is not due to the model itself but to either insufficient training, or insufficient number of neurons, insufficient number of training samples or an attempt to learn a supposed function that is not deterministic.

2.11.1 Learning

Learning is the weight modification process of an ANN in response to external input. There are three types of learning:

1. Supervised Learning
2. Unsupervised Learning, and
3. Reinforcement Learning

2.11.1.1 Supervised Learning

It is by far the most common type of learning in ANNs. It requires many samples to serve as exemplars. Each sample of this training set contains input values with corresponding desired output values (also called target values). The network will then attempt to compute the desired output from the set of given inputs of each sample by minimizing the error of the model output to the desired output. It attempts to do this by continuously adjusting the weights of its connection through an iterative learning process called training. As mentioned earlier, the most common learning algorithm for training networks is the back-propagation algorithm (Aggarwal, 2018).

2.11.1.2 Unsupervised Learning

It is sometimes called self-supervised learning and requires no explicit output values for training. Each of the sample inputs to the network is assumed to belong to a distinct class. Thus, the process of training consists of letting the network uncover these classes. It is not as popular as supervised learning (Aggarwal, 2018).

2.11.1.3 Reinforcement Learning

It is a hybrid learning method in that no desired outputs are given to the network, but the network is told if the computed output is going in the correct direction or not (Aggarwal, 2018).

2.12 Autoencoders

An autoencoder is a type of neural network that is trained in such a way that it aims to copy its input to its output (Goodfellow et al., 2016). An autoencoder is designed so that it reproduces its input at the output layer. One of the main differences between an autoencoder and a multilayer perceptron, or deep neural network, is that the number of input neurons is equal to the number of output neurons. In addition, instead of generating an output of say, y (i.e., representing a class of benign or malicious), the output normally will generally be X' , a reconstruction of the original input X . An autoencoder's primary utility is to find a lower dimensional, latent space representation of the input data. It does this

in a non-linear fashion, unlike other popular dimensionality reduction techniques such as Principal Component Analysis (PCA). Therefore, a key application of autoencoders is to use them for pretraining a neural network, and thus use them to improve the performance of supervised and/or unsupervised learning paradigms. In addition, they can be used to power an anomaly detection system.

Autoencoders are special kinds of artificial neural networks that employ a supervised learning algorithm for unsupervised tasks. As a matter of fact, as all other artificial neural networks, the learning of the weights is based on the error between the target and the output of the network. Although, in the case of autoencoders, the target of the network corresponds to the input of the network itself, making of them an unsupervised technique. Formally speaking autoencoders are artificial neural networks trained in a setting in which the target function corresponds to the identity function (function).

The key idea behind autoencoders is to take an input vector and map it to a latent space (encode) of lower dimensionality, then from that latent space, map it back to an output (decode) using the low dimensional latent space as input. The module that is capable of taking this latent space of lower dimensions and mapping it back to an output of with the same higher dimensions of the original input is called a decoder. This technique is commonly used for image generation; however, this study looks at applying this technique for the purposes of network intrusion detection, and classifying network flows as either malicious or benign based on reconstruction error (Goodfellow et al., 2016).

Figure 2.11 shows an example standard autoencoder neural network architecture, where the number of input neurons is equal to the number of output neurons. The way an autoencoder works is that it takes an input vector and maps it to a latent vector space using an encoder module, then decodes back to an output vector using a decoder module. The decoded output has the same dimensions as the original input vector. Then, the neural network is trained using target data that is the same as the original input vector, so that it learns how to reconstruct the original inputs. Using this process, and by placing certain constraints on the encoded output, the autoencoder can learn interesting latent representations of the original input data.

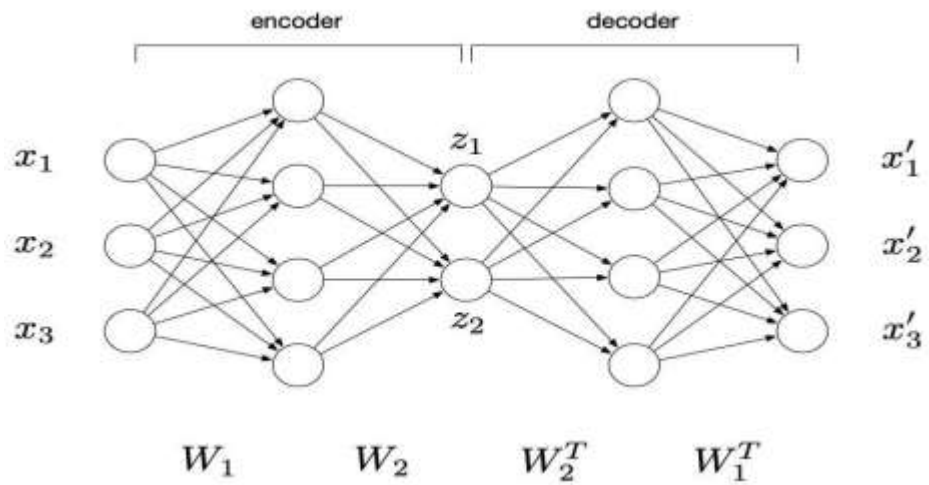


Figure 2.12: Example of a Neural Network Structure for Autoencoder (Goodfellow et al., 2016)

2.12.1 A General Autoencoder Framework

According to Baldi and Hornik, (1988), to derive at a fairly general framework, an $n/p/n$ autoencoder is defined by a t-uple $n, p, m, F, G, A, B, X, \Delta$ where:

1. F and G are sets.
2. n and p are positive integers. In this case, we consider primarily the case where $0 < p < n$.
3. A is a class of function from G^p to F^n .

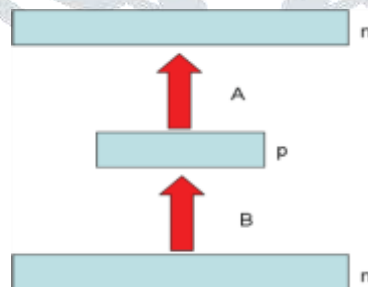


Figure 2.13: An $n / p / n$ Autoencoder Architecture (Baldi and Hornik, 1988)

4. B is a class of functions from F^n to G^p .
5. $X = \{ x_1, \dots, x_m \}$ is a set of m (training) vector in F^n . When external targets are present, we let $Y = \{ y_1, \dots, y_m \}$ denote the corresponding set of target vectors in F^n .

6. Δ is a dissimilarity or distortion function (e.g., L_p norm, hamming distance) defined over

F^n .

For any $A \in A$ and $B \in B$, the autoencoder transforms an input vector $x \in F^n$ into an output vector $A \circ B(x) \in F^n$. The corresponding autoencoder problem is to find $A \in A$ and $B \in B$ that minimize the overall distortion function:

$$\min E(A, B) = \min_{A,B} \sum_{t=1}^m E(x_t) = \min_{A,B} \sum_{t=1}^m \Delta(A \circ B(x_t), x_t)$$

In the non-auto-associative case, when external targets y_t are provided, the minimization problem becomes:

$$\min E(A, B) = \min_{A,B} \sum_{t=1}^m E(x_t, y_t) = \min_{A,B} \sum_{t=1}^m \Delta(A \circ B(x_t), y_t)$$

We note that $p < n$ corresponds to the regime where the autoencoder tries to implement some form of compression or feature extraction.

Obviously, from this general framework, different kinds of autoencoders can be derived depending, for instance, on the choice of sets F and G , transformation classes A and B , distortion function Δ , as well as the presence of additional constraints, such as regularization. To the best of our knowledge, neural network autoencoders were first introduced by the PDP group as a special case of this definition, with all vector components in $F = G = R$ and A and B corresponding to matrix multiplication followed by non-linear sigmoidal transformation is L^2_2 error function. For regression problems, the non-linear sigmoidal transformation is typically used only in the hidden layers (Baldi and Hornik, 1988; Baldi et al., 2011).

2.13 Network Traffic Analysis (NTA)

NTA is the process of detecting, recording and analyzing communication patterns in order to detect and respond to security menace, even when messages are encrypted. Traffic analysis is primarily performed to find out the data type, the traffic flowing through a network as well as data sources. However, it is used by attackers to discover communication patterns, and break in data over the

network. NTA solutions allow network administrators to collect data and monitor download/upload speeds on the traffic that flows through the network. The NTA tools have been commonly used to analyze and identify network security and performance issues. The traffic statistics from network traffic analysis helps in understanding and evaluating networks utilization, download\upload speeds and type, size, origin and destination and content of data (Taylor, 2017; Kim, 2019; Shafiq, 2016).

2.13.1 Traffic Pattern Analysis Purposes

Traffic analysis is important in network management and operations for the several purposes (Choudhury and Bhowal, 2015):

- ✦ To detect unknown threats: There are many techniques for detecting threats, but among this group, traffic pattern analysis has been proven to be the most effective tool. It can be used to detect unknown threats. Furthermore, using this tool, security administrators can configure the IP address range of each server within their local network, thus, identifying external IP address.
- ✦ To detect a malware communication with trusted sources: For example, every time you open a web browser, malware registered in your Gmail account. In this case, with the help of traffic pattern analysis, security administrators can monitor the high network traffic of email hosts, which is a starting point for further network inspection.
- ✦ To detect malware communications via HTTPS: This is performed through traffic patterns to identify changes.

2.13.2 Some Mechanisms in Traffic Analysis

Attackers are always inventing new methods and modifying methods to avoid detection by the network defense systems. Traffic analysis products have emerged in response to ongoing updates that provide ways to combat these attackers (Shafiq, 2016), Such as:

- ✦ Wireshark: Wireshark is the world's foremost and widely-used network protocol analyzer.

It lets you see what's happening on your network at a microscopic level and is the de facto (and often de jure) standard across many commercial and non-profit enterprises, government agencies, and educational institutions ((Choudhury and Bhowal, 2015).

- ✦ Jflow: JFlow is a software-based networking technology that facilitates the monitoring and recording of data packets flowing between configured devices, particularly routers and switches. JFlow is configured by default in routers and switches developed by Juniper networks. It records all network activity from the enabled port and saves statistical information on network usage retrievable through its programmatic interface (Parsaei, 2017).
- ✦ Self-Similarity and TES: Use Industrial Access Control & Security Systems for the analysis of communication system and Discover attacks.
- ✦ Wireless Sensor Networks WSN: It is a technology that can be used in large systems such as commercial applications, where security is vital for their applicability. Classify attacks in wireless sensor networks to explore patterns and possible countermeasures is widely used to deal with these attacks (Sommer and Paxson, 2010).
- ✦ Flow Analysis: Flow analysis is used to identify anonymity networks. It has a great accuracy in identifying encrypted anonymity networks. Flow analysis used in three main categories, Identification of anonymity networks, determine network traffic within encrypted and Profiling applications using flow analysis (Shahba and Zincir-Heywood, 2018).
- ✦ User Intention-Based Traffic Dependence Analysis: It uses existing algorithms and frameworks that analyze user actions and network events on a host according to their credentials. This can lead to detect relationships, identify anomalies, and conduct empirical assessments of the accuracy, security, and efficiency of algorithms (Zhang, 2012).

2.13.3 Network Traffic Monitoring System and its Challenges

Network monitoring has been used extensively for the purposes of security, forensics and anomaly detection (Shafiq, 2016). However, recent advances have created many new obstacles for NTAs. Some of the most pertinent issues include:

- ✦ **Volume:** The volume of data both stored and passing through networks continues to increase. It is forecast that by 2020, the amount of data in existence will top 44ZB (IDC, 2014). As such, the traffic capacity of modern networks has drastically increased to facilitate the volume of traffic observed. Many modern backbone links are now operating at wirespeeds of 100Gbps or more. To contextualize this, a 100Gbps link is capable of handling 148,809,524 packets per second (Juniper Networks, 2015). Hence, to operate at wirespeed, a NTAs would need to be capable of completing the analysis of a packet within 6.72ns. Providing NTAs at such a speed is difficult and ensuring satisfactory levels of accuracy, effectiveness and efficiency also presents a significant challenge.
- ✦ **Accuracy:** To maintain the aforementioned levels of accuracy, existing techniques cannot be relied upon. Therefore, greater levels of granularity, depth and contextual understanding are required to provide a more holistic and accurate view. Unfortunately, this comes with various financial, computational and time costs (Naseer et al., 2018).
- ✦ **Diversity:** Recent years have seen an increase in the number of new or customized protocols being utilized in modern networks. This can be partially attributed to the number of devices with network and/or Internet connectivity. As a result, it is becoming increasingly difficult to differentiate between normal and abnormal traffic and/or behaviors (Naseer et al., 2018).
- ✦ **Dynamics:** Given the diversity and flexibility of modern networks, the behaviour is dynamic and difficult to predict. In turn, this leads to difficulty in establishing a reliable behavioral norm. It also raises concerns as to the lifespan of learning models (Sekharan and Kandasamy, 2017).

- ✦ **Low-frequency attacks:** These types of attacks have often thwarted previous anomaly detection techniques, including artificial intelligence approaches. The problem stems from imbalances in the training dataset, meaning that NTAs offer weaker detection precision when faced with these types of low frequency attacks (Zhang, 2016).
- ✦ **Adaptability:** Modern networks have adopted many new technologies to reduce their reliance on static technologies and management styles. Therefore, there is more widespread usage of dynamic technologies such as containerization, virtualization and Software Defined Networks. NTAs will need to be able to adapt to the usage of such technologies and the side effects they bring about (Shen et al., 2018).

To resolve some of these challenges, traffic anomalies detection algorithms need to be designed and they mostly provide solutions based on categories. The first category detects flow outliers including statistical approaches, similarity approaches and pattern mining approaches. The second category derive trajectory outliers including online processing. Imperatively, intrusion detection systems, which are software applications or devices that observes a system or network for malicious activity appears to be a preferable solution to mitigate cyber threats (Djenouri et al., 2019; Kim, 2019).

2.14 Related Works on the Application of Deep Learning in Neural Network Security

Deep learning is garnering significant interest and its application is being investigated within many research domains, such as: healthcare (Liang et al., 2014; Shashikumar et al., 2017), automotive design (Falcini et al., 2017; Luckow et al., 2016), and manufacturing (Lee et al., 2017).

There are also several existing works within the domain of Network Intrusion Detection Systems (NIDS). In this section, we will discuss the most current notable works.

Dong and Wang undertook a literary and experimental comparison between the use of specific traditional NIDS techniques and deep learning methods (Dong and Wang, 2016). The authors concluded that the deep learning-based methods offered improved detection accuracy across a range

of sample sizes and traffic anomaly types. The authors also demonstrated that problems associated with imbalanced datasets can be overcome by using oversampling for which, they used the Synthetic Minority Oversampling Technique (SMOTE).

Alrawashdeh and Purdy (2016) proposed using a RBM with one hidden layer to perform unsupervised feature reduction. The weights are passed to another RBM to produce a DBN. The pre-trained weights are passed into a fine-tuning layer consisting of a Logistic Regression classifier (trained with 10 epochs) with multi-class soft-max. The proposed solution was evaluated using the KDD Cup '99 dataset. The authors claimed a detection rate of 97.90% and a false negative rate of 2.47%. This is an improvement over results claimed by authors of similar papers.

The work by Kim et al., (2017) aspired to specifically target advanced persistent threats. They propose a Deep Neural Network (DNN) using 100 hidden units, combined with the Rectified Linear Unit activation function and the ADAM optimizer. Their approach was implemented on a GPU using TensorFlow, and evaluated using the KDD data set. The authors claimed an average accuracy rate of 99%, and summarized that both RNN and Long Short-Term Memory (LSTM) models are needed for improving future defenses.

Javaid et al., (2016) propose a deep learning-based approach to building an effective and flexible NIDS. Their method is referred to as self-taught learning (STL), which combines a sparse auto-encoder with softmax regression. They have implemented their solution and evaluated it against the benchmark NSL-KDD dataset. The authors claim some promising levels of classification accuracy in both binary and 5-class classification. Their results show that their 5-class classification achieved an average f-score of 75.76%.

Potluri and Diedrich, (2016) propose a method using 41 features and their DNN has 3 hidden layers (2 auto-encoders and 1 soft-max). The results obtained were mixed, those focusing on fewer classes were more accurate than those with more classes. The authors attributed this to insufficient training data for some classes.

Cordero et al., (2016) proposed an unsupervised method to learn models of normal network flows. They use RNN, auto-encoder and the dropout concepts of deep learning. The exact accuracy of their proposed method evaluated is not fully disclosed.

Similarly, Tang et al., (2016) also propose a method to monitor network flow data. The paper lacked details about its exact algorithms but does present an evaluation using the NSL-KDD dataset, which the authors claim gave an accuracy of 75.75% using six basic features.

Kang and Kang, (2016) proposed the use of an unsupervised DBN to train parameters to initialize the DNN, which yielded improved classification results (exact details of the approach are not clear). Their evaluation shows improved performance in terms of classification errors.

Hodo et al., (2017), have produced a comprehensive taxonomy and survey on notable NIDSs approaches that utilize deep and shallow learning. They have also aggregated some of the most pertinent results from these works.

In addition, there is other relevant work, including the DDoS detection system proposed by Niyaz et al., (2015). They propose a deep learning-based DDoS detection system for a software defined network (SDN). Evaluation is performed using custom generated traffic traces. The authors claim to have achieved binary classification accuracy of 99.82% and 8-class classification accuracy of 95.65%. However, we feel that drawing comparisons with this paper would be unfair due to the contextual difference of the dataset. Specifically, benchmark KDD datasets cover different distinct categories of attack, whereas the dataset used in this paper focuses on subcategories of the same attack.

Wang et al., (2016), propose an approach for detecting malicious JavaScript. Their method uses a 3-layer SdA with linear regression. It was evaluated against other classifier techniques, showing that it had the highest true positive rate but the second best false positive rate.

Lee et al., (2017), propose a deep-learning approach to fault monitoring in semiconductor manufacturing. They use a Stacked denoising Autoencoder (SdA) approach to provide an unsupervised learning solution. A comparison with conventional methods has demonstrated that throughout different use cases the approach increases accuracy by up to 14%. in different use cases. They also concluded that among the SdAs analyzed (1-4 layers) those with 4 layers produced the best results.

Wang et al., (2017), built an intrusion detection algorithm using raw network traffic data from two existing datasets: the CTU-13 dataset and the IXIA dataset (that the authors called the USTCTFC2016 dataset), which contained 10 types of normal data and 10 types of malicious data, and appeared to be relatively balanced between malicious and normal. A preprocessing step took the raw network traffic data and converted it into images, which were then fed into a CNN with a similar architecture to the well-established CNN LeNet-5 (LeCun et al., 1995). Because there was no engineering of the preprocessing stage that produced the images, this method handled the raw data directly. The classification was done in two different ways. The first method involved a 20class classifier, and the goal was to identify which type of normal or malicious the traffic was. The second was a binary classifier which fed into one of two CNNs trained to identify the type of malicious traffic or binary traffic. The 20-class classifier achieved an accuracy of 99.17%. The binary classifier achieved 100% whereas the 10-class normal classifier achieved 99.4% and the 10-class malicious classifier achieved 98.52%.

Yu et al., (2017), developed a network intrusion detection algorithm using dilated convolutional autoencoders (DCAEs) to identify normal and malicious traffic. Dilated convolutions are similar to regular convolutions, but there are gaps in between the applications of the kernel. This can be very useful because the receptive field can grow more quickly and spatial information can be merged much more aggressively. Using a data preprocessing module, raw network traffic data (.pcap) is converted into 2D numeric vectors. Unlabeled data is then used to train the DCAE, which is an autoencoder that uses convolutional layers instead of fully connected layers. Like standard autoencoders, these can be

stacked to make deep networks. However, Yu et al., (2017) found that adding more than one hidden layer did not significantly improve performance, which they found was the best with one convolution layer, and a fully connected layer with an ReLU, followed by a classification layer. Using two datasets, CTU-UNB (CTU-13; UNB-ISCX 2012; Shiravi et al., (2012) and Contagio-CTU-UNB, Yu et al., (2017), were able to achieve accuracies exceeding 98.5% on two-, six-, and eight-class problems.

The findings from our literature review have shown that despite the high detection accuracies being achieved, there is still room for improvement. Such weaknesses include the reliance on human operators, long training times, inconsistent or average accuracy levels and the heavy modification of datasets (e.g., balancing or profiling). The area is still in an infantile stage, with most researchers still experimenting on combining various algorithms (e.g., training, optimization, activation and classification) and layering approaches to produce the most accurate and efficient solution for a specific dataset. Hence, we believe the model and work presented in this paper will be able to make a valid contribution to the current pool of knowledge.

2.15 Issues with Sourcing Research Data

A primary and ongoing challenge in the field of network intrusion detection is the lack of publicly available, labeled datasets that can be used for effective testing, evaluation, and comparison of techniques (Niyaz et al., 2015; Shiravi et al., 2012) Often times, the most useful datasets for network intrusion detection are those containing captures of real network environments. These datasets are not easily shared with the public, as they contain details of an organization's network topology, and more importantly sensitive information about the traffic activity of the users on the respective network. Furthermore, the effort required to create a labeled dataset from the raw network traces is an immense undertaking. As a consequence, researchers often resort to suboptimal datasets, or datasets that cannot be shared amongst the research community. Granted, publicly labeled datasets are available, such as CAIDA (Young et al., 2011), DARPA/Lincoln Labs packet traces (Lippmann et al., 1999), KDD '99 Dataset (KDDCUP, 1999), and Lawrence Berkeley National Laboratory (LBNL) and ICSI Enterprise

Tracing Project (Niyaz et al., 2015); however, these datasets are mostly anonymized and do not contain valuable payload information, making them less useful for research purposes (Shiravi et al., 2012). While these datasets have proven useful, there are some 26 arguments as to the validity of using them in present day research - they may be better suited for the purposes of providing additional validation and cross-checking of a novel technique (Sommer and Paxson, 2010).

This work focuses on using newer benchmark datasets that have recently become available to the research community, specifically the University of New Brunswick Canadian Institute of Cybersecurity Distributed Denial-of-Service 2019 dataset (UNB CIC DDoS 2019) datasets, which was sourced from the uniform resource locator (URL) <http://unb.ca/cic/datasets/ddos-2019.html/>.

JETIR

CHAPTER THREE

METHODOLOGY

3.0 INTRODUCTION

The approach adopted for the proposed systems development methodology is discoursed in this chapter. Three existing frameworks (architectures) were reviewed extensively and used as part of the basis for the development of the proposed architecture. Owing to the fact that the proposed framework is software based, its development was conducted using appropriate system development paradigms, system development approach based on an acceptable research approach.

First, the baseline frameworks were presented and analyzed then the conceptual solutions to the proposed framework was discoursed, and the conceptual framework was then developed.

The details of the tasks undertaken in this chapter are presented.

3.1 Research Approach

There are four basic approaches to empirical research. These are Survey, Case study, Experiment and Quasi-Experiment (Boshoff, 1980). This study adopted the quasi-experimental approach which typically comprised of the three other approaches, for e.g., the literature review took into

consideration the survey aspect of the study while the adoption of the incidence of attacks on financial institutions represents the case study aspect. The design, implementation, testing, and the fixing of bugs in the proposed system depicted the experiment aspect of the study.

3.2 Review of Existing Frameworks

3.2.1 AutoIDS: Autoencoder Based Method for Intrusion Detection System.

Gharib et al., (2019), in their work autoencoder Based method for intrusion detection systems developed a semi-supervised deep learning method to precisely detect anomalous traffic in communication systems. The main contribution of their work was focused on proposing a novel, effective and accurate solution for detecting abnormal traffic with an acceptable time complexity. More specifically, their contributions were majorly finetuned towards proposing a semi-supervised deep learning method that detect anomalies with higher performance, in terms of accuracy, compared to the other state-of-the-art solutions, and evaluating the proposed method under realistic circumstances and also showing its superiority.

Gharib et al., (2019) developed a model that consists of two detectors, D_1 and D_2 that includes a sparse autoencoder and a simple straight forward autoencoder, respectively. In their model, D_1 criterion for distinguishing normal from anomalies was sparsity and the sparse autoencoder was trained only on normal traffic. Accordingly, their work was optimized to provide a sparse representation from normal packet flows.

The framework (architecture) of how anomalous traffic was detected and classified using sparse autoencoder and simple autoencoder is illustrated in Figure 3.1. The drawback and other compromising factor in their work is highlighted in the subsequent section.

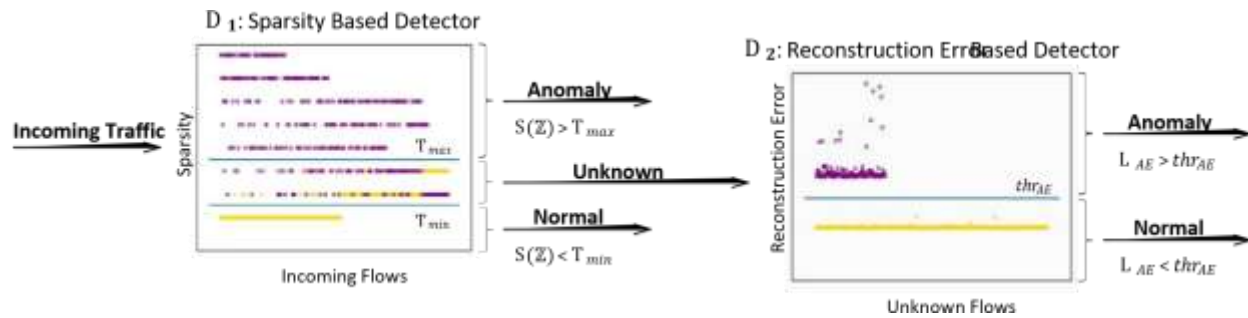


Figure 3.1: A framework of how anomaly detection in AutoIDS is processed using two phases (Gharib et al., 2019)

3.2.2 Intrusion Detection and Classification with Autoencoded Deep Neural Network

Rezvy et al., (2018) in their work focused on exploring low latency models while maintaining high accuracy by proposing a hybrid deep neural network that includes an unsupervised pre-training using autoencoders to make the model more adaptive to the changes in network traffic. To increase efficiency, they used a dedicated supervised dense neural network structure for the final classification.

In their design, they ensured that the memory or processing power to train and execute machine learning models were within the capability of the routers processing power. Figure 3.2 illustrates the framework (architecture) of how the model was developed.

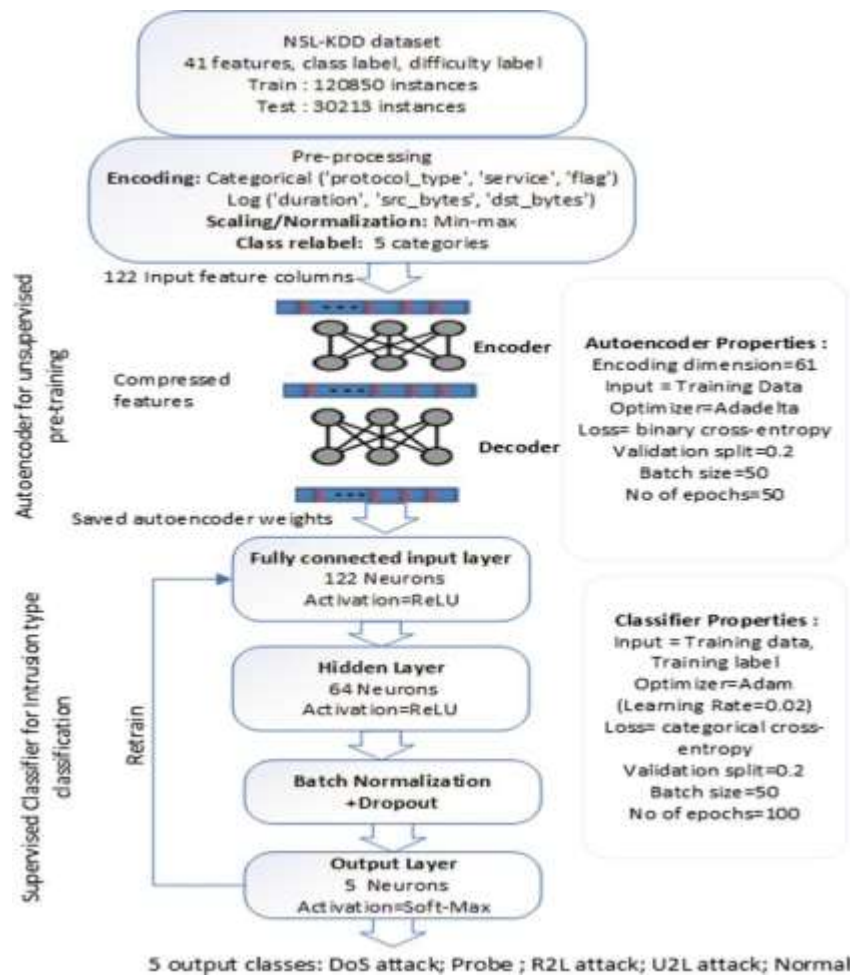


Figure 3.2: The architecture of an autoencoded dense neural network (Rezvy et al., 2018)

3.2.3 Kitsune: An Ensemble of Autoencoders for Online Network Intrusion Detection

Mirsky et al, (2018) developed Kitsune, a plug and play Network Intrusion detection system that was designed to learn and detect network attacks on a local network without supervision in an efficient way. The core algorithm of Kitsune (KitNET) was designed to use autoencoder as the key model for training and classification. The main objective of this was to design Kitsune in such a way that it will seamlessly collect and differentiate between normal and abnormal traffic patterns. Their model was designed with feature extraction framework which was able to efficiently track the patterns of every network channel.

Though Kitsune was able to achieve 75% of its design purpose, a major drawback was its failure to efficiently classify benign traffics while in train-mode. Figure 3.3 is a framework (architecture) of Kitsune and how it was designed.

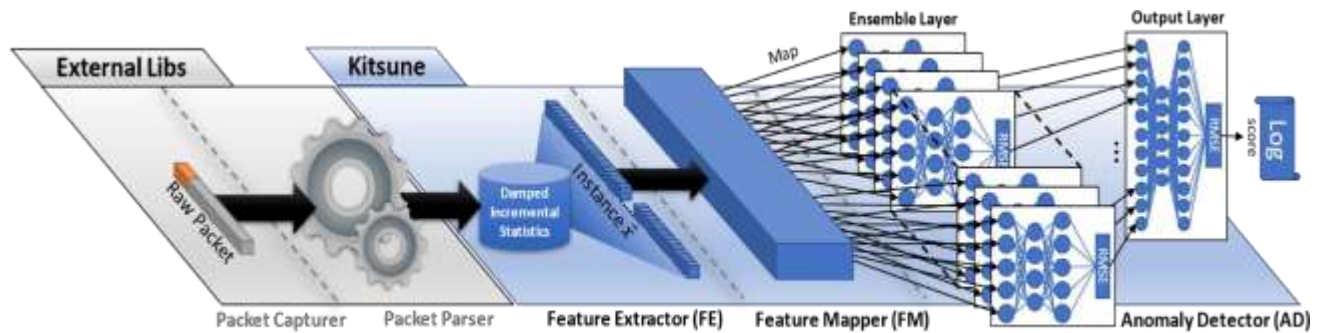


Fig. 3: An illustration of **Kitsune's** Architecture.

Figure 3.3: An illustration of Kitsune's Architecture (Mirsky et al., 2018)

3.3 Overall Drawbacks of Existing frameworks (Architecture)

From the foregoing, it is apparent that these architectures need to be improved upon for reliable solution to mitigating these attacks.

Consequently, the proposed framework is developed to address these concerns and present veritable added-on capabilities. To this end, the conceptual solutions offered by the proposed framework are presented.

1. The proposed framework is designed to handle the process of data preprocessing. Ordinarily other frameworks required preprocessing data. For e.g., some require the use of applications for pruning and/or cleaning data before they can be fed into the system. This is a major problem because most data traffic are heavy and comes in gigabits.
2. Some existing architectures were designed to handle specific forms of packets (attacks) and they do not have the capability to be retrained with ease to handle other traffic types (attacks). In the proposed framework passing other traffic types through the autoencoder automatically equips the framework with the capability to handle other traffic types.
3. The proposed framework contains enough visualization capabilities that makes it easy to use and monitor even by novices.

Figure 3.4, presents the overall conceptual architectural diagram of the proposed work.

The detailed processes and procedures entailed in the design framework are discoursed in the subsequent sections spanning the framework design methodology which adopted the Agile methodology.

3.2 Research Paradigm

This study adopted the top-down refinement systems design paradigm. The system study already had a conceptual architecture of the proposed framework, and to achieve the integral component of this framework, features, and functionalities, the study decomposed the individual integral components that comprise the architecture into sub modules that where amenable to easy design and coding.

3.3 Dataset

In the course of this study, obtaining the requisite data required for the training of the proposed model was a major challenge and this as we all know, is due to the fact that organizations,



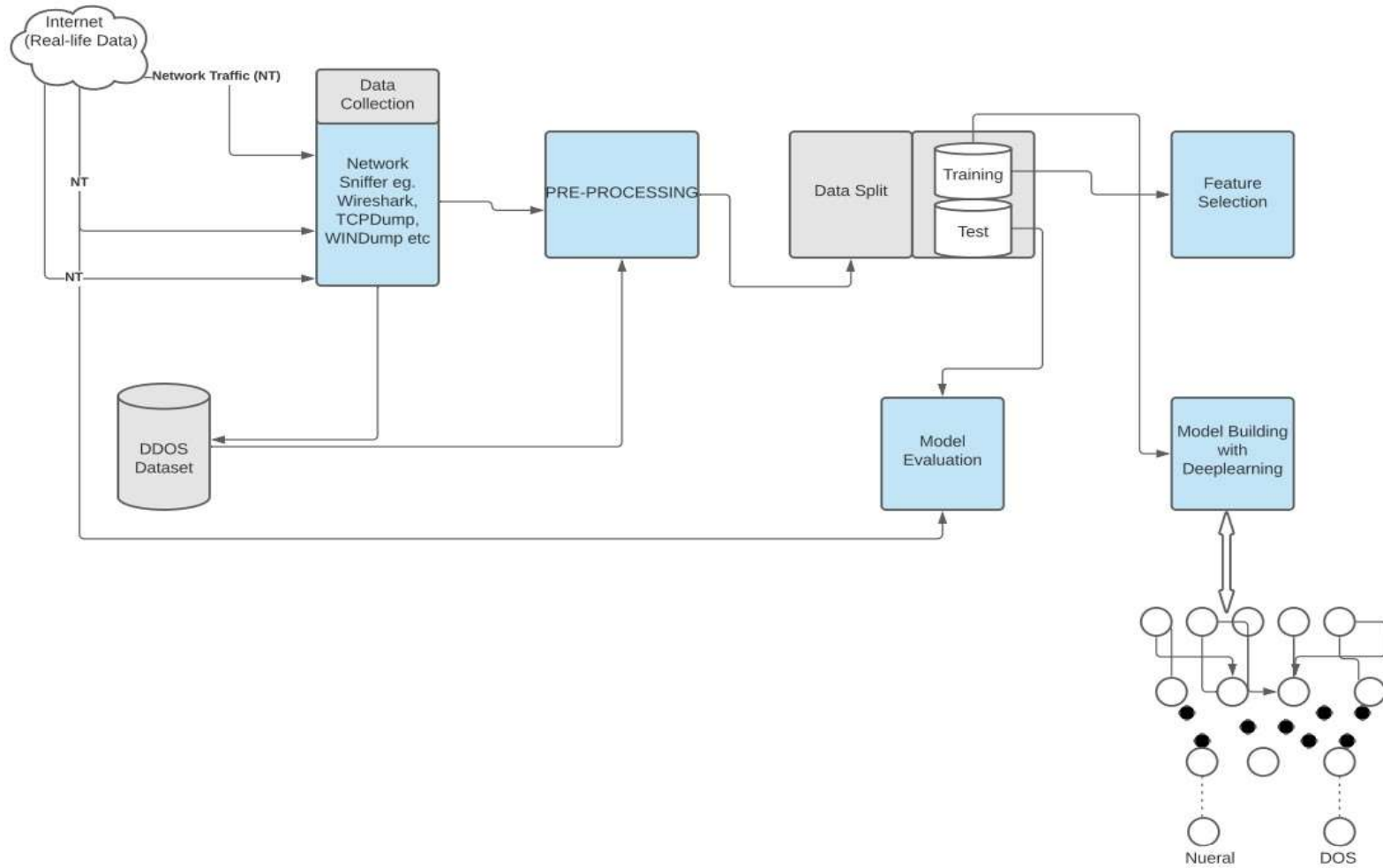


Figure 3.4: Architectural Framework for DDoS

especially financial institutions hardly disclose data breaches, let alone allow public access to the data traffic employed in such breaches.

This claim can be alluded to categorially because one of the researchers (in this study has about Eight (8) years working experience as a network administrator (with CISCO Certified Network Associate (CCNA) and CISCO Certified Network Professional (CCNP)) in a reputable financial institution. And the board of directors in financial institutions has it as a policy that forbids disclosure of attacks or publicizing attack related information of any sought.

This policy which tends to cut across virtually all organization especially financial institutions made access to the different data types difficult. For e.g., this study could only access reliable data for denial-of-service related attacks, attempts to obtain ransomware (malware) Manin-the-middle related attacks, Identity theft related attacks and other forms of attacks could not be sourced. This study builds the proposed framework using the available data source that could be obtained. However, the framework was developed in such a way that it could easily accept other data traffic types for its training and function effectively. To achieve this specific design, the study adopted an approach that allows any data source to be preprocessed and organized into a specific format that can serve as input into the framework. The framework now train itself with the input data such that it can recognize subsequent attacks that has the characteristics of the trained data.

Detail of how this was achieved is discussed in chapter four.

3.4 Proposed System Design Methodology

Agile Method was selected as the major design methodology. This choice was informed by the fact that the methodology is quick and iterative and often results in good designs. It is highly suitable for applications of this nature because they require shorter development life cycles, have frequently changing requirements characterized by frequent updates. Figure 3.5, shows the Agile system development life cycle.

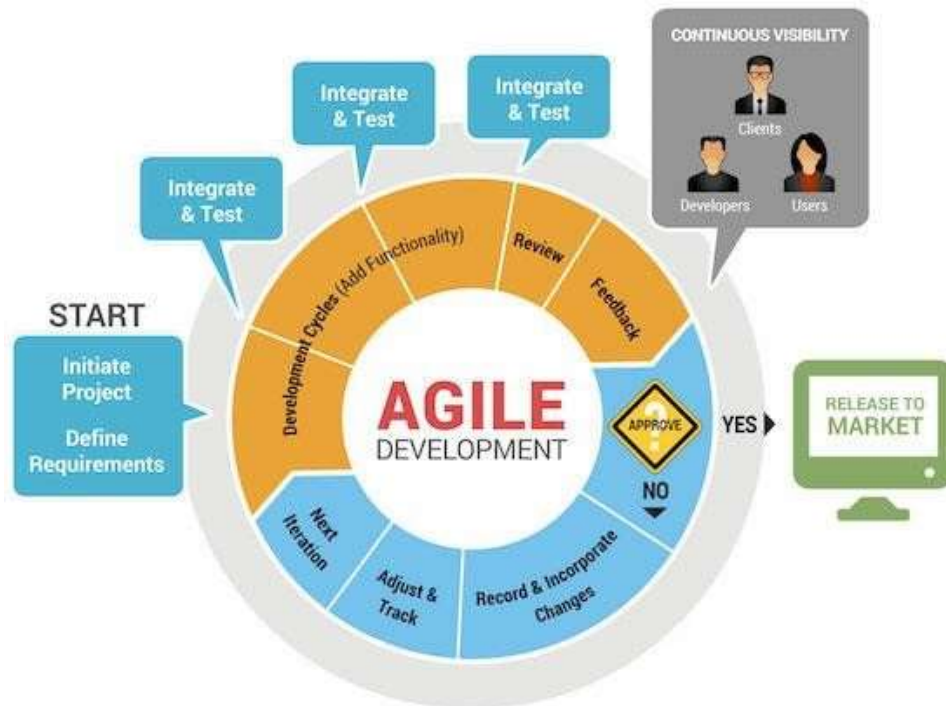


Figure 3.5: Agile Development Life Cycle (TechAheadCorp, 2016)

The specific Agile methodology adopted in this study was eXtreme Programming. Its basic activities were followed tenaciously in this order: Designing; Coding; Testing and Listening in the course of the systems development and testing.

A hybrid system design methodology was adopted in this study. These are Agile and Object-Oriented analysis and design methodology. Object oriented analysis and design was used to augment the Agile methodology to cater for the structuring of the proposed implementation. The specific object-oriented analysis and design methodology approach used was some unified modelling language tools of activity diagram and class diagrams. Figure 3.6 shows the activity diagram of the proposed system while figures 3.7, 3.8 and 3.9 shows the class diagrams.

The activity diagram shows the interaction of the entities in the proposed framework while the class diagram shows the different modules implementing the activities require for the system to function.

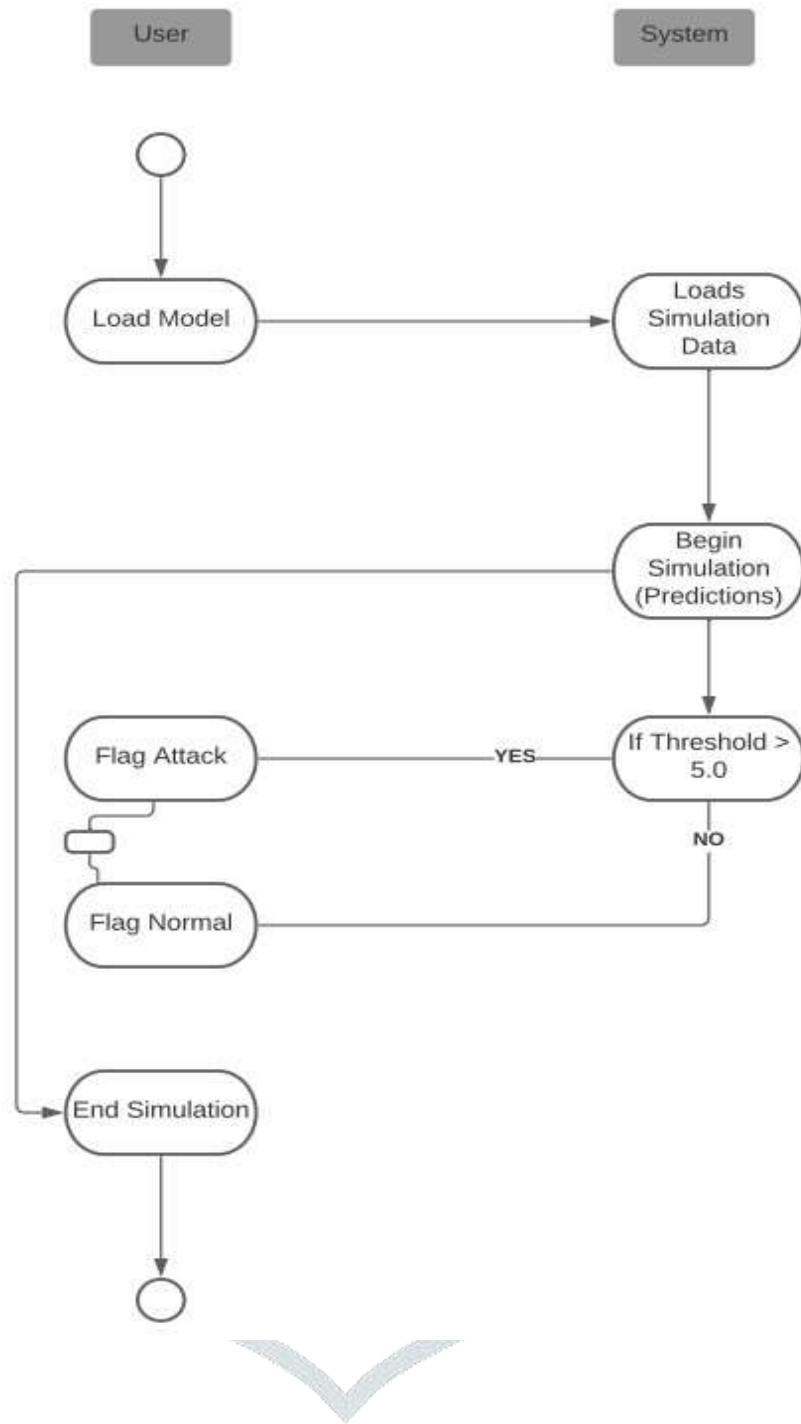


Figure 3.6: Shows Activity Diagram for the Simulation System

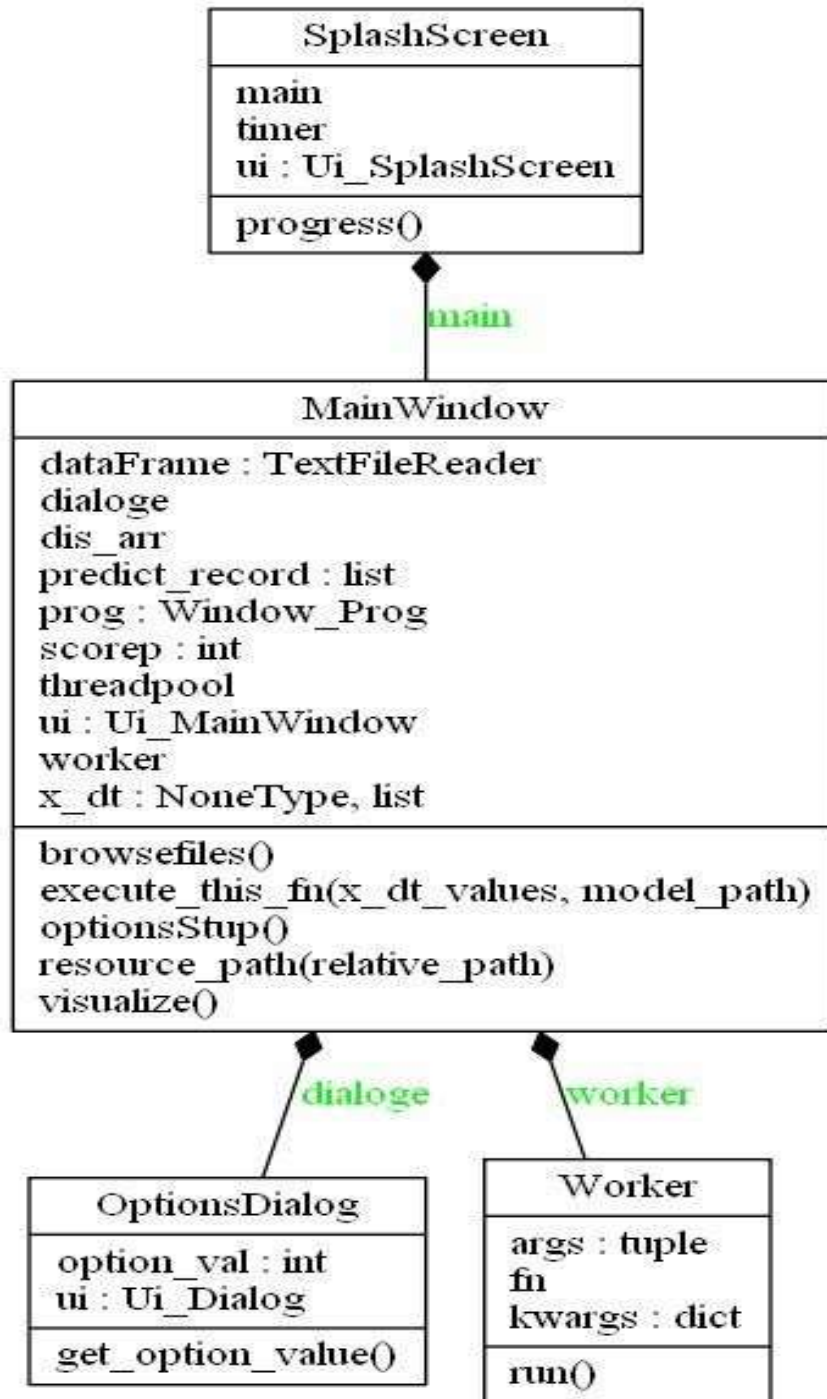


Figure 3.7: Shows a Class Diagram for the Simulation System

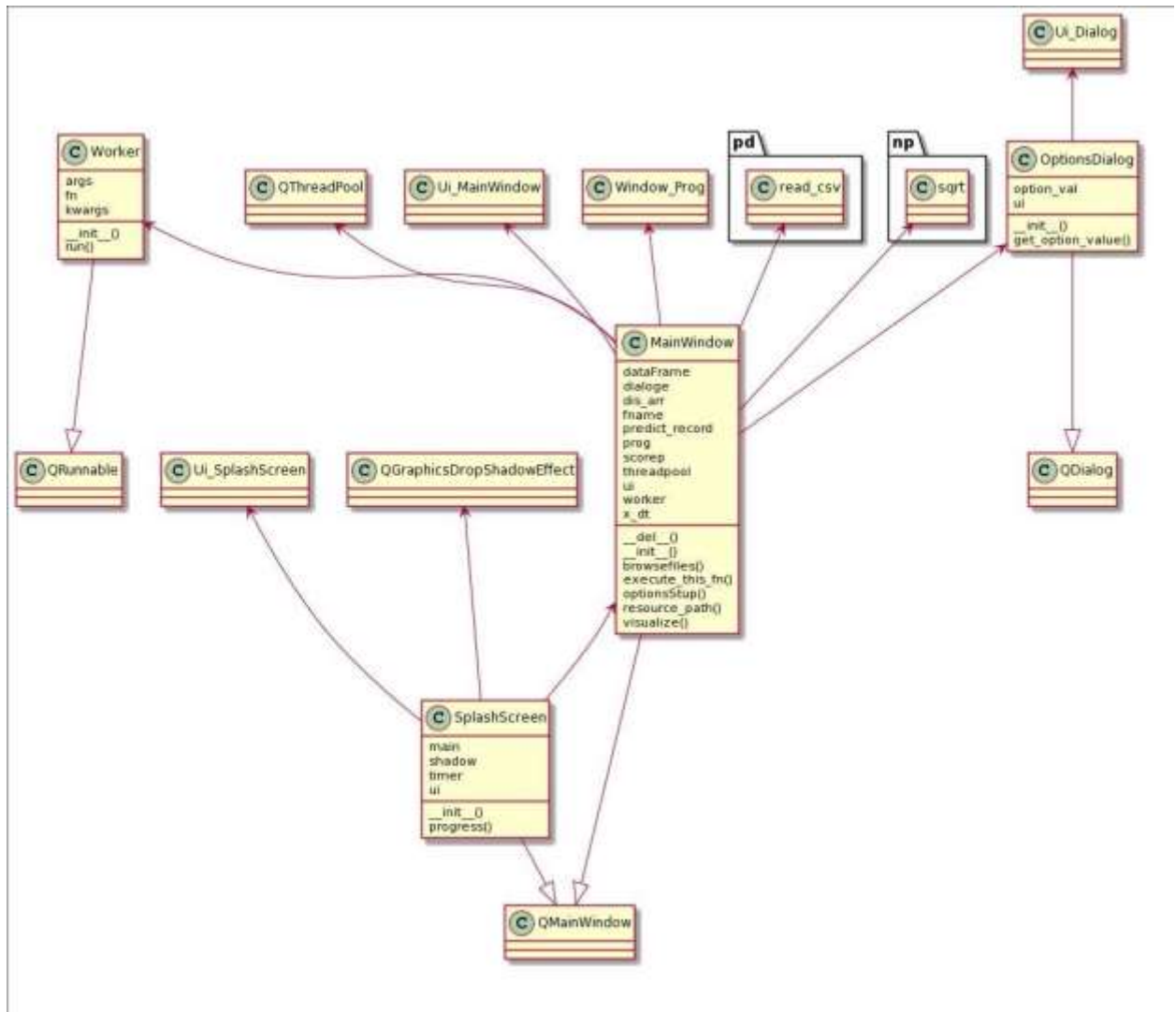


Figure 3.8: Shows a Class Diagram for the Simulation System

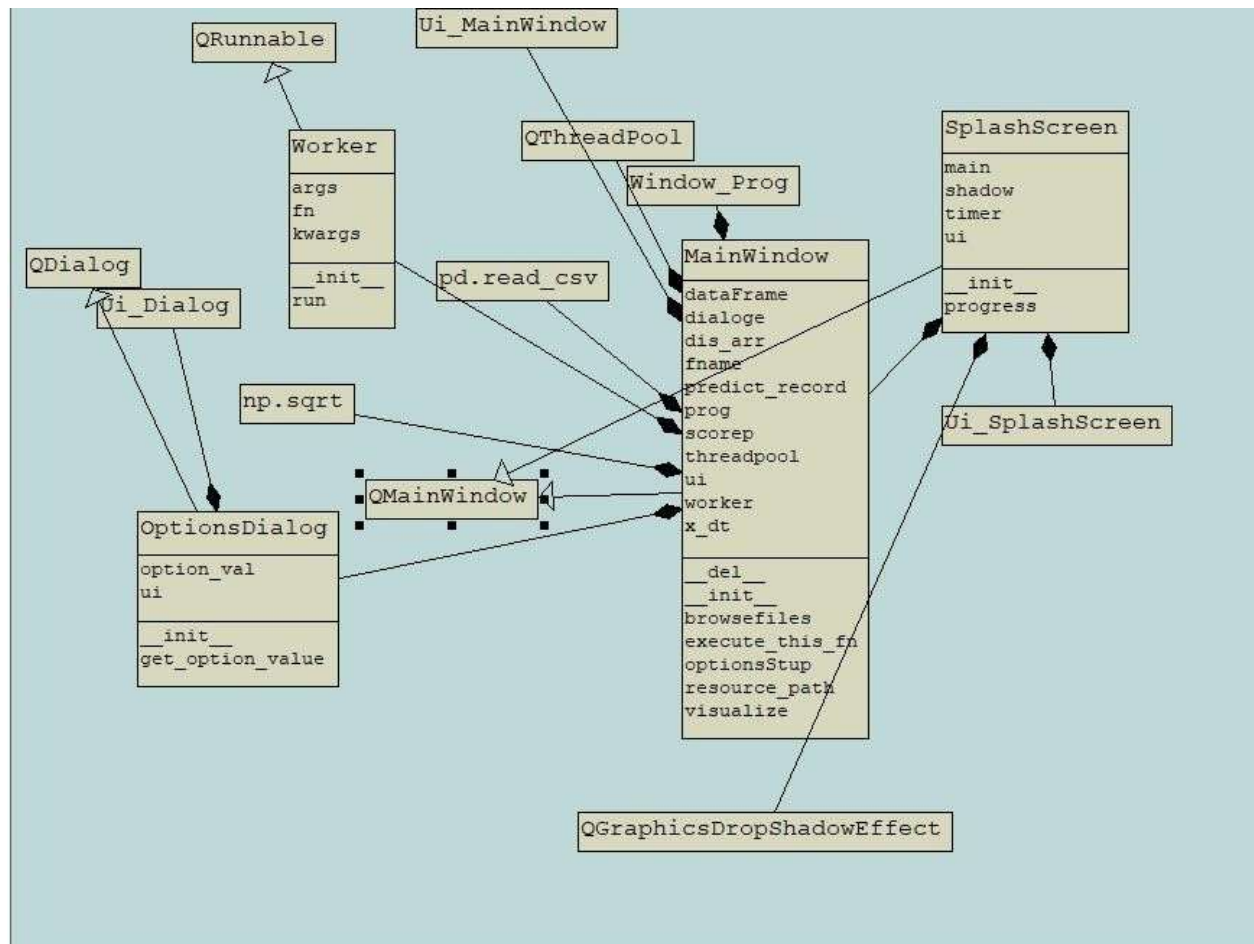


Figure 3.9: Shows a Class Diagram for the Simulation System

3.5 Development Environment Setup

To develop this proposed system the development environment was setup using the appropriate set of tools assemblage following copious research, consultation, and training. The specific tools that were eventually found appropriate are listed as follows:

- ‡ Python
- ‡ Matplotlib
- ‡ Sklearn
- ‡ NumPy
- ‡ Pandas
- ‡ Tensorflow keras
- ‡ Keract

The details of these software development tool, and the appropriate aspect of the proposed framework that they used to design and implement are presented in chapter four.

The development environment setup provided a rich and comprehensive set of software development tools.

The Python development language provided tools such as

a debugger, libraries, documentation, sample code, and tutorials. It also provided facility for accepting and packaging codes in different file format

In addition to the comprehensive list of available software development tools supported by Python development language, several other tools from the library where very useful in the proposed framework development. For e.g., utilities for computing various statistics and other desired values such as Root Mean Square Error (RMSE), Mean Square Error (MSE), Mean Absolute Error (MAE), and Confusion Matrix from available data were called straight from the libraries in the course of the framework development.

3.5.1 The Proposed System Simulation Interface

The proposed system simulation interface has facilities to pass-in incoming traffic as input into the system, conduct the required preprocessing, pass the data through the neural network for analysis and threat detection training and then generate outputs in the form of graphs and information list to advice against possible threats of attacks. The simulation interface has facilities that allows for selection of traffic type to test for the validity of the system. For e.g., traffic with real threat can be selected and the system will indicate that there's actually a threat. Also, normal traffic could be selected to test the same system to know if the framework developed is working appropriately. And as its conventional, a traffic comprising both normal and attack threats can be run on the system too to show that the system is functioning effectively.

3.6 Proposed System Design, Coding, Testing and Listening

As already explained, the specific Agile steps listed above are discoursed as it applies to this study.

3.6.1 System Design

By convention, the system design was strictly coding base, testing and debugging based. And because the system was dealing with elaborate system development a software development process was followed. This particular software development process was closer to the time tested, code, and fix model which aligned properly with the Agile design method. Having assembled the tools, the various aspects of the module were Coded and Tested iteratively before being integrated into the entire architectural framework.

3.6.2 Coding, Testing, Listening Activities

After designing and testing of the individual modules and their integration into the overall architectural framework, another set of coding, testing, and listening activities were performed successively to achieve an overall effective result required from the proposed system. This next step entailed integration, coding, testing to see that the integration was okay, and listening to ensure that the desired features and functionalities were achieved. For e.g., the system needs to accept dataset, conduct the required preprocessing, pass it through the neural network for training of the framework, the general appropriate output, and then test the system with different sets of sample data to see how well the system is working.

Application of this Agile system design method following the code and fix software development model is presented in chapter four in the system implementation. A code snippet is used to highlight some important designs.

CHAPTER FOUR

SYSTEMS IMPLEMENTATION, TESTING AND DISCUSSION

4.0 INTRODUCTION

This chapter presents an overview of the proposed system implementation and testing. The chapter commences with a description of the various software tools used and the various aspects of the proposed framework that they were used to develop. The system which is essentially code intensive, uses code snippets to highlight the accomplishments of the various modules, of the proposed framework. The codes

that constitute the overall framework are exceptionally bulky and are available for perusal if the need arises. The chapter also contains the presentation showing the outputs generated by the proposed framework in the course of testing the proposed framework.

4.1 System Implementation Tool Description

The various system implementation tools of the study are discussed as follows:

4.1.1 Python

Python provides a standard framework for developing deep learning related systems. It is an interpreted, high-level and general-purpose programming language and it was selected as the overarching development language for this study. It has vast libraries and it provide support for the easy integration of other applications in machine learning and deep learning. It meets the requirements and the design objectives of this study because it accommodated the various other tools that were employed in this study.

These tools are described as follow;

4.1.2 Pandas

As a result of the vast volume of data, it was imperative to automate the data cleaning and representation process. And to achieve this, Pandas was programmed to help to represent the requisite traffic data type and present it in a format that could be used in the study.

Pandas is a fast, powerful, flexible and easy to use open-source data analysis and manipulation tool, built on top of the Python programming language. It was primarily used in this study to manipulate and create tables. In the course of this study, the datasets employed were in csv file format and as such, to manipulate and visualize the dataset, Pandas framework was used to read in those csv files and it was also used to append the dataframes together, then the study assigned all appended dataframe to the variable df_init, that way, which made it possible to have just one reference to all the data. Pandas was used to display and visualize the raw dataset in table format. As a key framework, Pandas was used to group labels together

based on their values and then display them. Below is a sample code snippet of how pandas' tool was used to append and group the datasets together (Sample pandas code snippet).

Sample pandas code snippet

```
# appends all dataframe to the df_DrDos_DNS dataframe and assigns this dataframe to
df_init
# this way there is just one reference to all dataframes df_init =
df_DrDoS_DNS.append(df_DrDoS_LDAP,ignore_index=True, sort=False)\
.append(df_DrDoS_MSSQL,ignore_index=True, sort=False)\
.append(df_DrDoS_NetBIOS,ignore_index=True, sort=False)\
.append(df_DrDoS_NTP,ignore_index=True, sort=False)\
.append(df_DrDoS_SSDP,ignore_index=True, sort=False)\
.append(df_DrDoS_UDP,ignore_index=True, sort=False)\
.append(df_PORTMAP,ignore_index=True, sort=False)\
.append(df_Syn,ignore_index=True, sort=False)\
.append(df_TFTP,ignore_index=True, sort=False)\
.append(df_UDPLag,ignore_index=True, sort=False)\
.append(df_BENING,ignore_index=True, sort=False)
```

A code snippet of how pandas was used to display the limit for rows and columns:

```
# set the display limit for rows and columns then display the table to be able
to view the data pd.set_option('display.max_columns', 15)
pd.set_option('display.max_rows', 5) display(df_init[0:5])
```

4.1.3 Matplotlib

Matplotlib was adopted as the main plotting tool in this study. It was used to plot the performance of the training process and to specifically display the graphs. Matplotlib is an opensource comprehensive library for creating static, animated, and interactive visualizations in Python.

The Mean Absolute Error (MAE) is one of the metrics used in compiling the model, alongside the Mean Squared Error (MSE), with the goal of visualizing the training process of the framework. Matplotlib was also used to plot the result of the MAE loss alongside MAE validation and also the MSE loss and its validation. To further evaluate the performance of the model matplotlib was used to show confusion matrix of a quick prediction of thirty random attack and normal data each passed to the model (Sample matplotlib code snippet).

The sample code snippet of how matplotlib was used to achieved some of the plots is shown below:

Sample matplotlib code snippet

```
# Plot history: MAE plt.plot(history.history['mean_absolute_error'], label='MAE
(training data)') plt.plot(history.history['val_mean_absolute_error'], label='MAE
```

```
(validation data') plt.title('MAE for Chennai Reservoir Levels') plt.ylabel('MAE
value') plt.xlabel('No. epoch') plt.legend(loc="upper left") plt.show()
```

```
# Plot history: MSE plt.plot(history.history['mse'], label='MSE
(training data)') plt.plot(history.history['val_mse'], label='MSE
(validation data)') plt.title('MSE for Chennai Reservoir Levels')
plt.ylabel('MSE value') plt.xlabel('No. epoch') plt.legend(loc="upper
left") plt.show()
```

4.1.4 Keract

Keract was the tool used to generate the neural network diagram from the activation map of the proposed system. The activation maps of the framework represent how the encoder compresses the input data layer by layer down to the output layer. At a basic level, activation functions help decide whether a neuron should be activated. It also helps to view the progress and performances of the training process. Keract itself is dependent on matplotlib and uses it exquisitely inside of it to display the result of some of the performances during training and to display each of the layers in the process. Below is a code snippet of how Keract imports and display activation maps

Sample Keract code snippets for displaying activation maps

```
from keract import get_activations, display_activations
activations = get_activations(model, x_normal_test[:1])
display_activations(activations, cmap="gray", save=False)
```

4.1.5 Scikit-learn (Sklearn)

Scikit-learn is a data analysis tool that allows the generation of vital information from tabular data. It is used to elicit various forms of relationships among given dataset used extensively in machine learning and deep learning. Scikit-learn was used to achieve the Root Means Square Error (RMSE) between the predictions and the actual data being predicted in this study.

Scikit-learn is a Python module for machine learning built on top of SciPy. It's a simple and efficient tool for predictive data analysis. It is currently accessible to everybody and reusable in various contexts. Sklearn is built on NumPy, SciPy, and matplotlib. Below is a sample code snippet of Scikit-learn.

Code snippet of scikit-learn

```

from sklearn import metrics

model.predict(x_normal_test) score1 =
np.sqrt(metrics.mean_squared_error(pred,x_normal_test)) pred =
model.predict(x_normal) score2 = np.sqrt(metrics.mean_squared_error(pred,x_normal))
pred = model.predict(x_attack) score3 =
np.sqrt(metrics.mean_squared_error(pred,x_attack))
# for i in range(200):
#     x_pred_test =
print(f"Out of Sample Normal Score (RMSE): {score1}")
print(f"Insample Normal Score (RMSE): {score2}")
print(f"Attack Underway Score (RMSE): {score3}")
Out of Sample Normal Score (RMSE): 0.37192343215663665
Insample Normal Score (RMSE): 0.368255965802025
Attack Underway Score (RMSE): 0.6938244145696303
the trained model is then saved to disk

```

4.1.6 Tensorflow Keras

Tensorflow was used for the design of the framework in this study. To achieve the desired design Tensorflow and its sequential model, layers and activations as well as NumPy were integrated into the overarching Python source development code to generate a module called an autoencoder. This autoencoder was used to develop the framework, assign layers and inputs shapes alongside with the algorithm (relu). The resulting design was compiled with a mean_squared_error loss. This was done to enable the framework look for outliers (anomalies). Typically, the layers were perceptron neurons that were imported from Tensorflow library. A sequential layer was first imported and subsequent layers were added to achieve ninety input layers, this was then compressed to twenty-five (25) input layers (dense) and then to three (3) hidden layers. From the hidden layers a set of twenty-five (25) output layers were regenerated which was subsequently expanded to ninety-nine output layers. Figure 3.1, shows these layers.

The Autoencoder uses the first layer to encode the input by compressing it, then it decodes the already compressed input by trying to reconstruct the encoded data as output and it does this in 100 epochs

(iteration). An activation was also done with Tensorflow which is the mathematical algorithm that helps to improve the training. For every epoch, activation was used to improve the training process. Basically, Tensorflow was used to train the model, test predictions, and in the process determine the efficiency of the trained model with its validation.

Below is a sample code snippet of Tensorflow used to generate the neural network. Sample neural network generation code snippet

```
from IPython.display import display, HTML

from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Activation
from tensorflow.keras import metrics
model = Sequential()
model.add(Dense(25, input_dim=x_normal.shape[1], activation='relu'))
model.add(Dense(3, activation='relu')) # size to compress to
model.add(Dense(25, activation='relu')) # Multiple output neurons

# model.compile(loss='mean_squared_error', optimizer='adam',
#               metrics=[metrics.mae, metrics.mse])
model.compile(loss='mean_squared_error', optimizer='adam',
              metrics=[metrics.mae, 'mse'])
#               metrics=['mae', 'categorical_accuracy'])
model.summary()
# we print out the summary of the model to have a simple quick look of it.
```

Model: "sequential_3"

(type)	Output Shape	Param #	Layer
(Dense)	(None, 25)	2500	dense_12
(Dense)	(None, 3)	78	dense_13
(Dense)	(None, 25)	100	dense_14
(Dense)	(None, 99)	2574	dense_15

=====
Total params: 5,252
Trainable params: 5,252
Non-trainable params: 0
=====

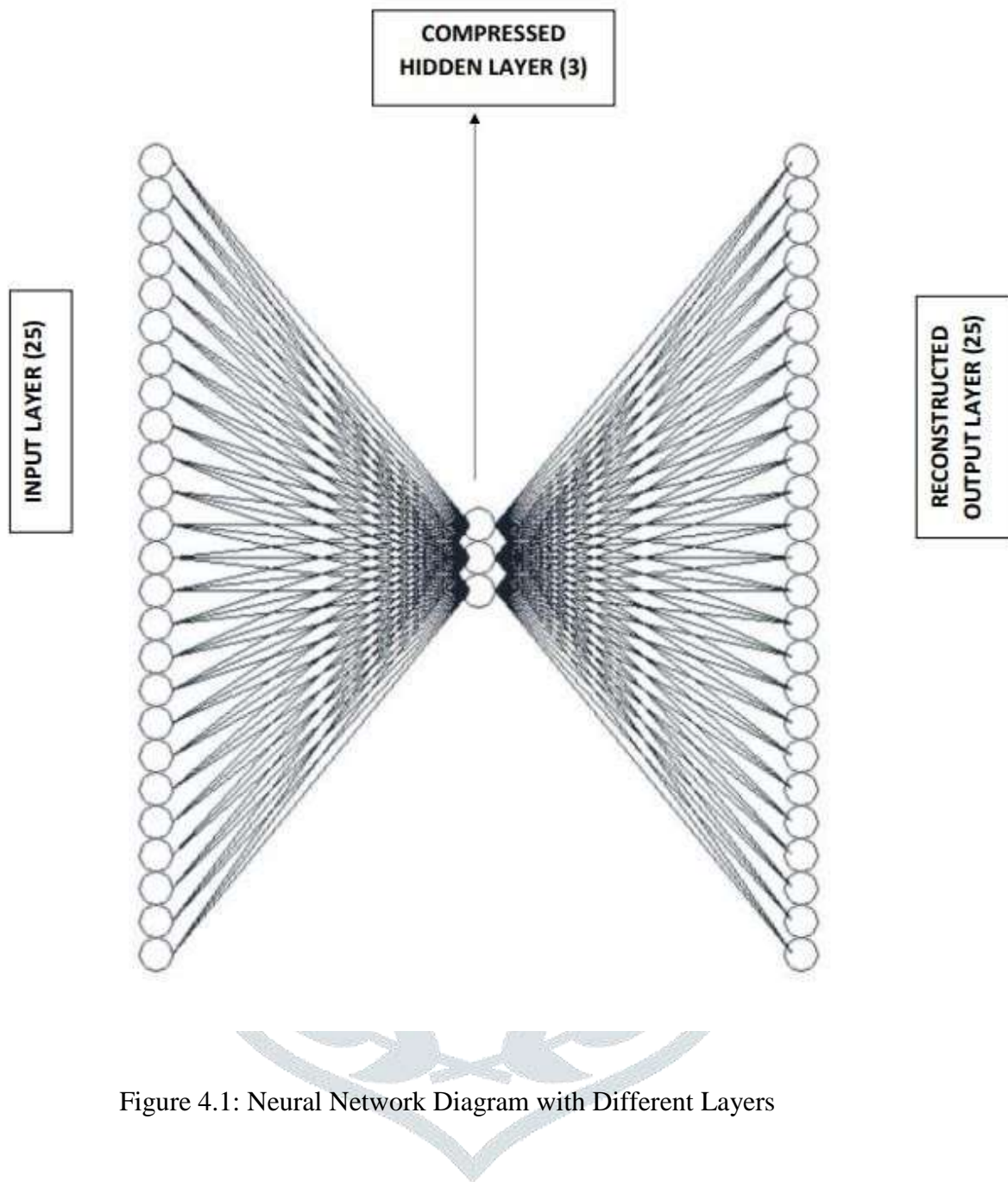


Figure 4.1: Neural Network Diagram with Different Layers

4.1.7 NumPy

Another tool that was used to facilitate the proposed framework design was NumPy. This was imperative because some computation involving the use of large dataset was undertaken in the course of the study. The most important reason for using it is because of its large number of arrays in the study

dataset. NumPy is usually used for mathematical and scientific processes, and most suitable for the processing of large datasets. With NumPy, the study was able to create a better way of handling List and array in an efficient way without having too much burden with processing of the dataset. In deriving some important useful values (Confusion matrix) used for validation in this study, the use of NumPy proves very useful. For e.g., The input(argument) for the **confusion matrix** method takes a NumPy array and for that reason the study converted the sample array (generated tabular data) into a form amenable to NumPy computation before passing it on to **confusion matrix** method for better prediction.

Below is a sample code snippet for computing the confusion matrix.

Sample code snippet for confusion matrix generation.

```
%matplotlib inline
from sklearn.metrics import confusion_matrix import
itertools
import matplotlib.pyplot as plt treshold =
np.array(treshold)
rounded_predList = [round(i) for i in pred_list] rounded_predList =
np.array(rounded_predList)
cm = confusion_matrix(y_true=treshold,y_pred=rounded_predList)
```

4.2 Proposed System Testing and Documentation

Having gotten all the modules together, integration testing was conducted to show how the proposed framework works. After that the model was then tested in units as it was trained and tested using Python programming libraries to ensure that each unit functioned properly. Later, the trained model (autoencoder) was integrated and tested in different development environment to fully assess its functionalities and features. It was then migrated to a productive standalone desktop environment for all its features and functionalities to be tested. The screen shots illustrating how the autoencoder inspect, and flag packets as either benign or malicious are discussed in this session.

Figure 4.2 basically allows you to choose the model that was trained to begin simulation while figure 4.3 clearly shows the home page that indicates that the trained model is ready to start the simulation process.

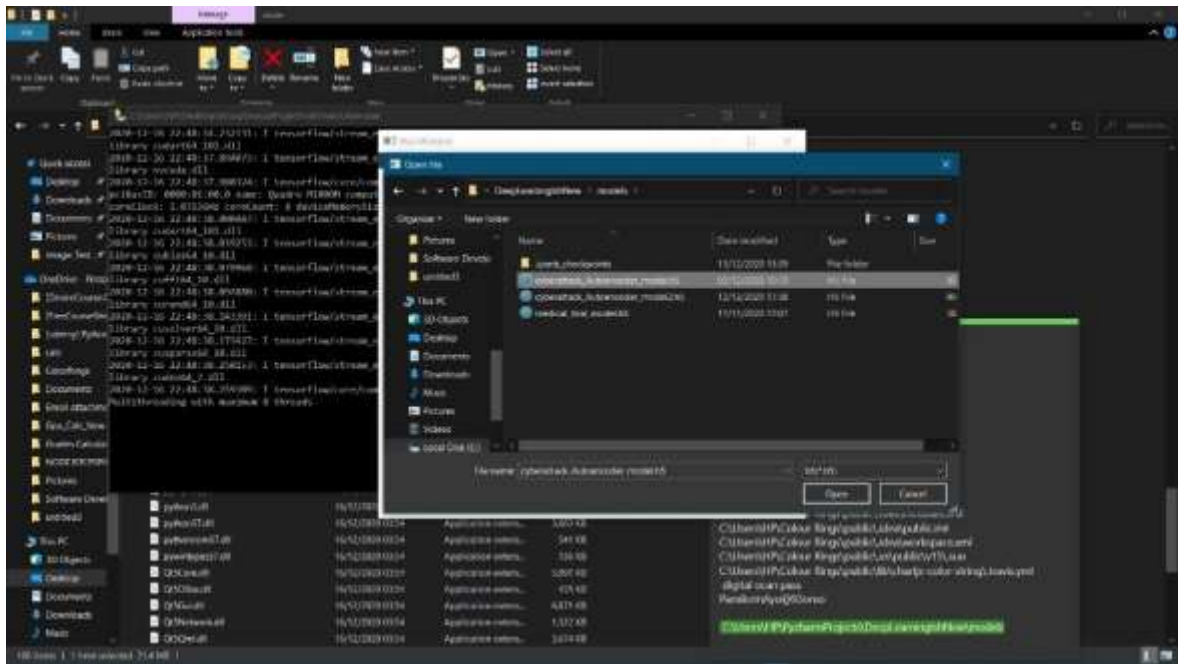


Figure 4.2 Shows a Screen that Select the Trained Model for Simulation

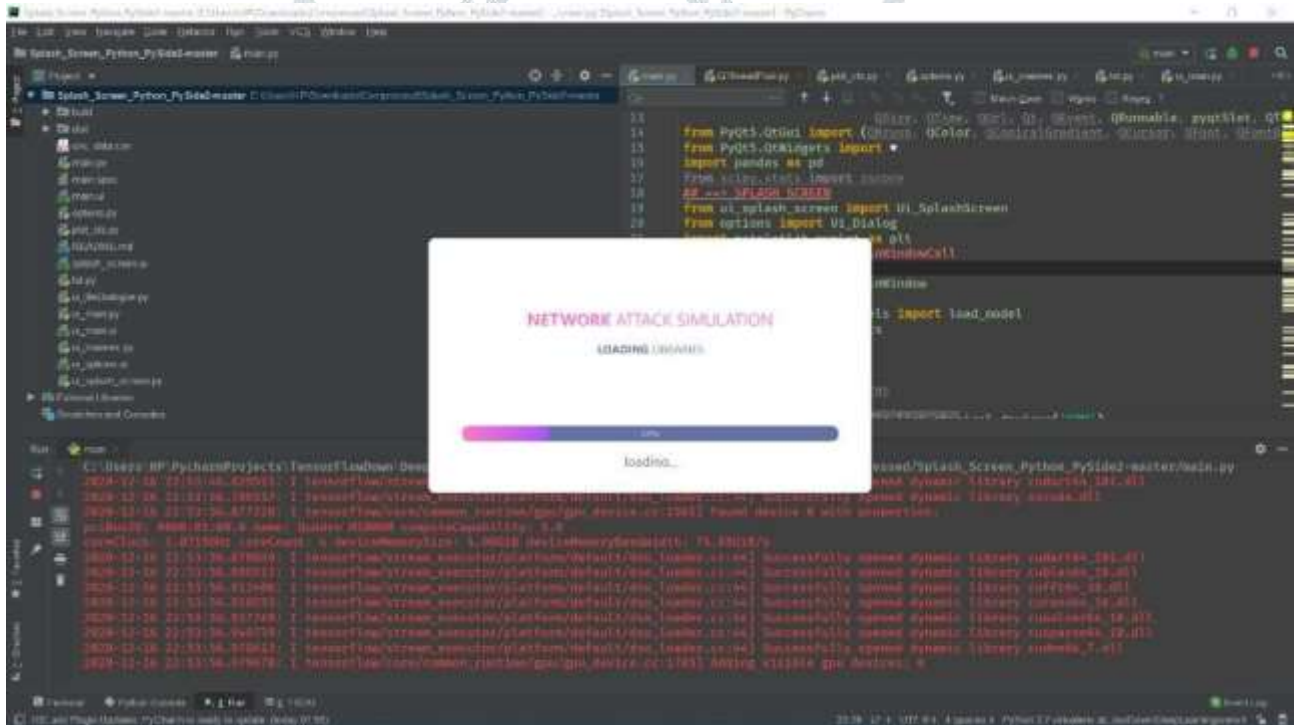


Figure 4.3: Shows Network Attack Simulation Homepage

Figure 4.4 shows the “Option” plane with three radio buttons that allows you to select the type of data to simulate with. It has an attack, normal, and both radio button. Ideally, it is preferable to select the “both” button because it allows for simultaneous simulation of both an attack and normal trained data. This makes it easier to visualize with graphs how attacks are flagged as either benign

(normal) or malicious (attack). The visualize plane also shows the Root Means Square Error (RMSE) of the simulation process. It basically indicates that if the RMSE is less than 0.5, the packet is flagged as normal. And any number that is above the trash hold which is 0.5, will be automatically flagged as an attack. The figures below highlight the process of how normal and attacks traffics are visualized.

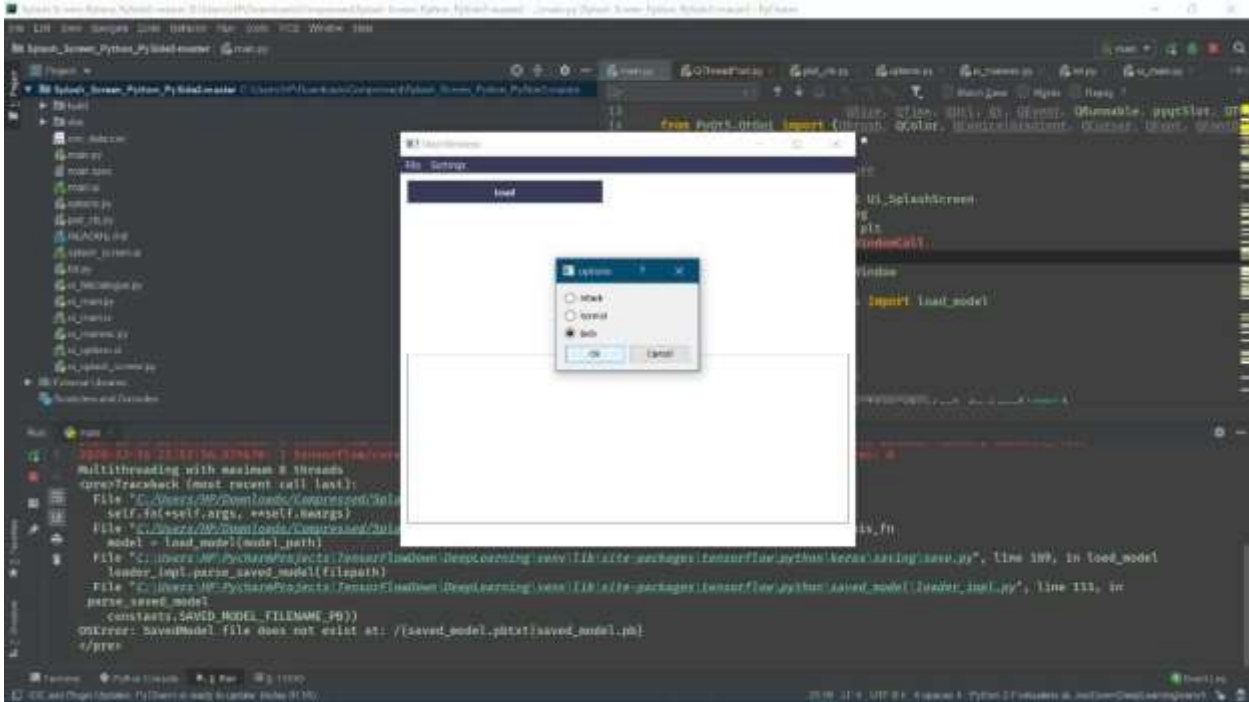


Figure 4.4: Shows the Option Plane that Indicates the Various Attack Buttons

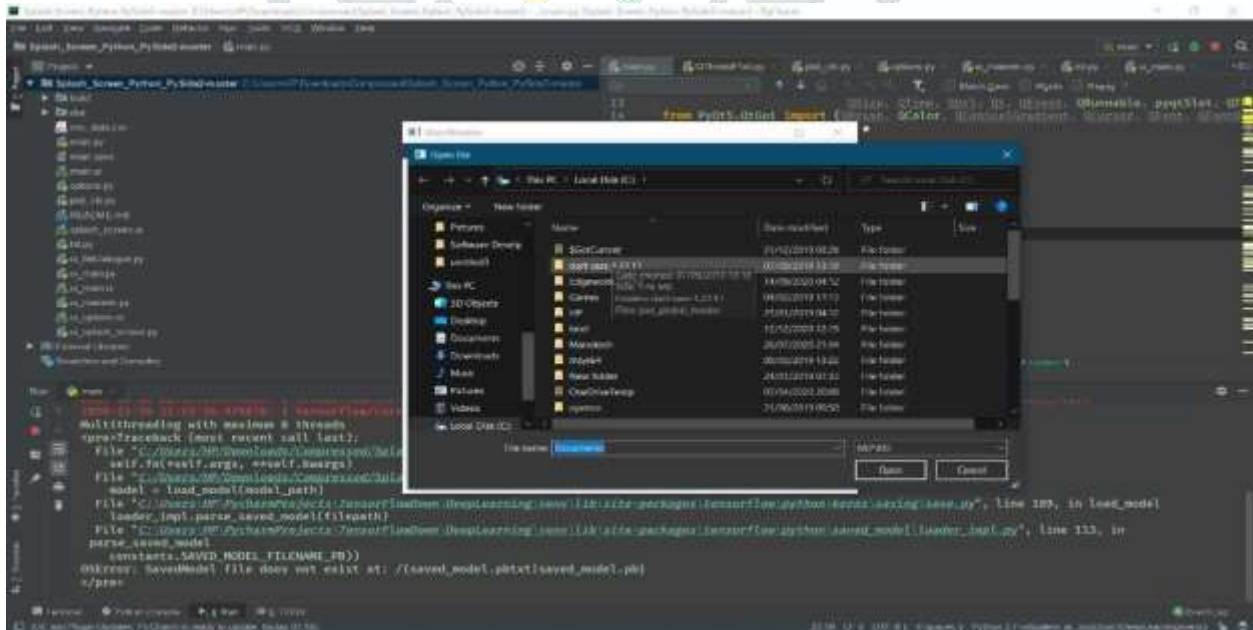


Figure 4.5: Shows the Trained Data Packets for Simulation

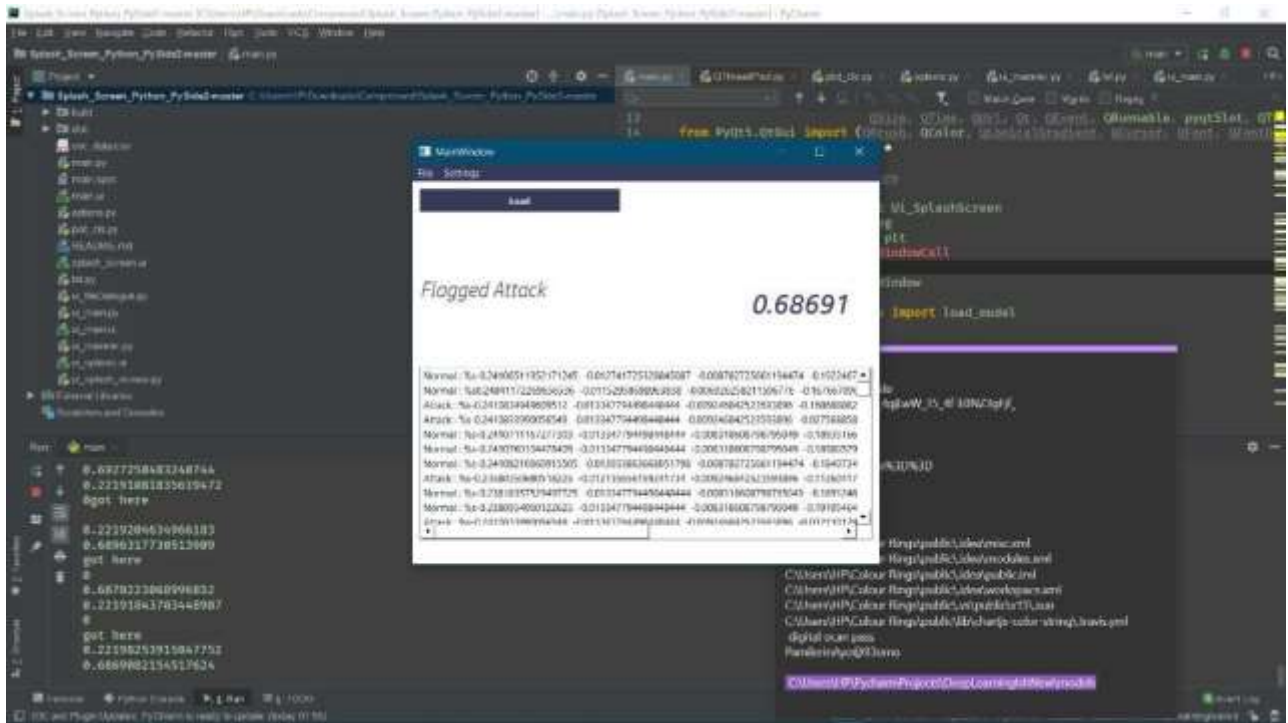


Figure 4.6: Indicates that an Attack has been Flagged Based on the RMSE Rate

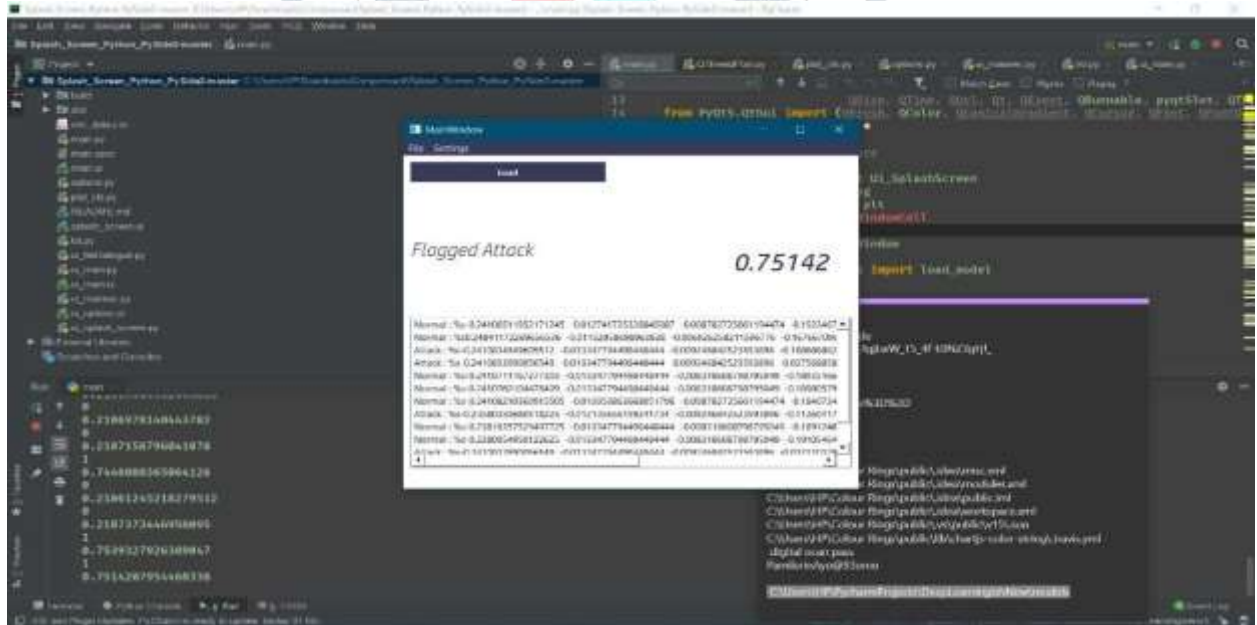


Figure 4.7: Indicates that an Attack has been Flagged Based on the RMSE Rate

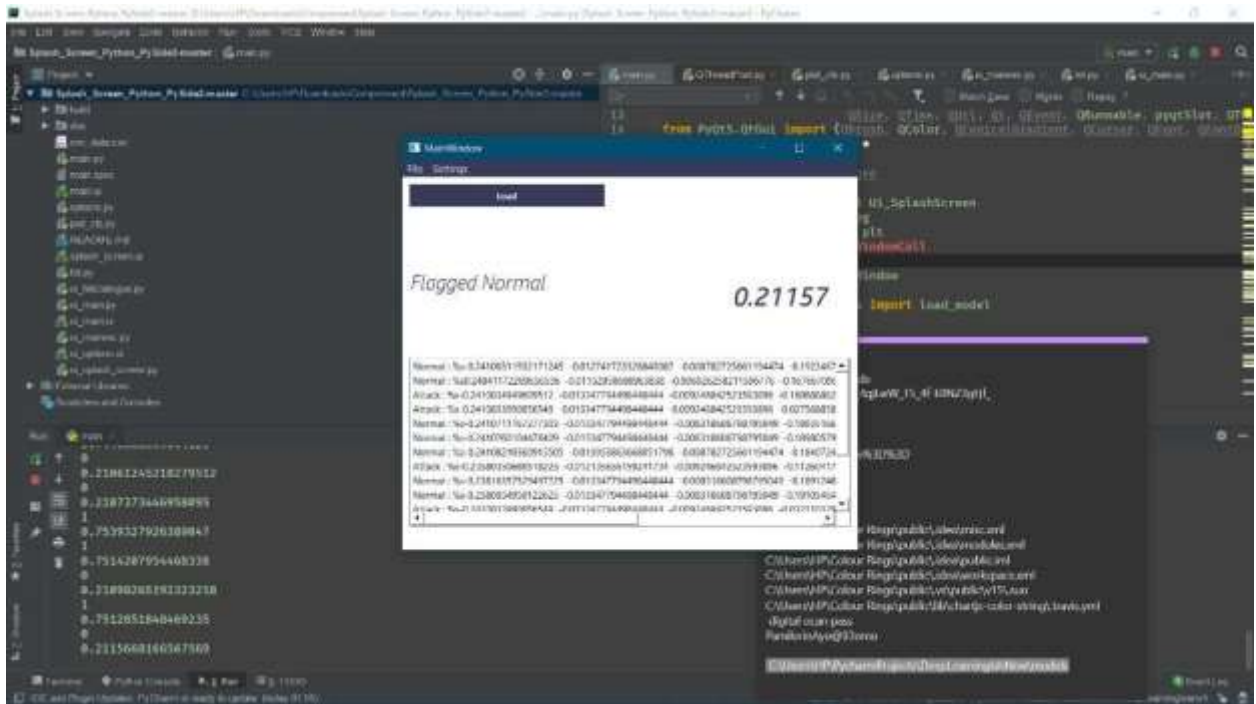


Figure 4.8: Indicates that a Packet has been Flagged as Normal Based on the RMSE Rate

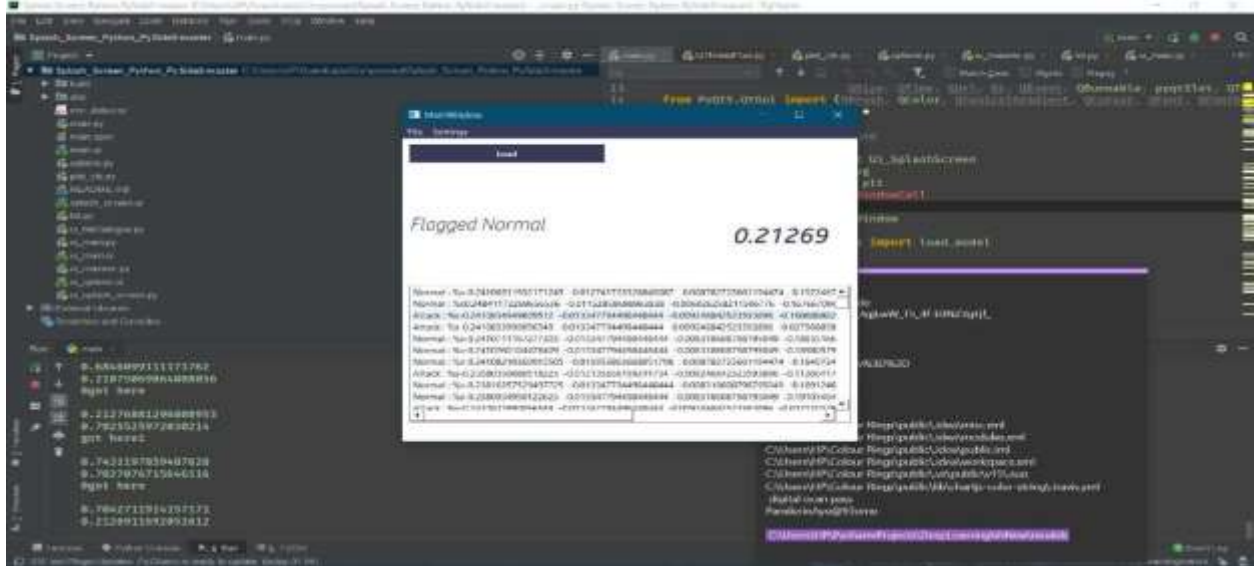


Figure 4.9: Indicates that a Packet has been Flagged as Normal Based on the RMSE Rate

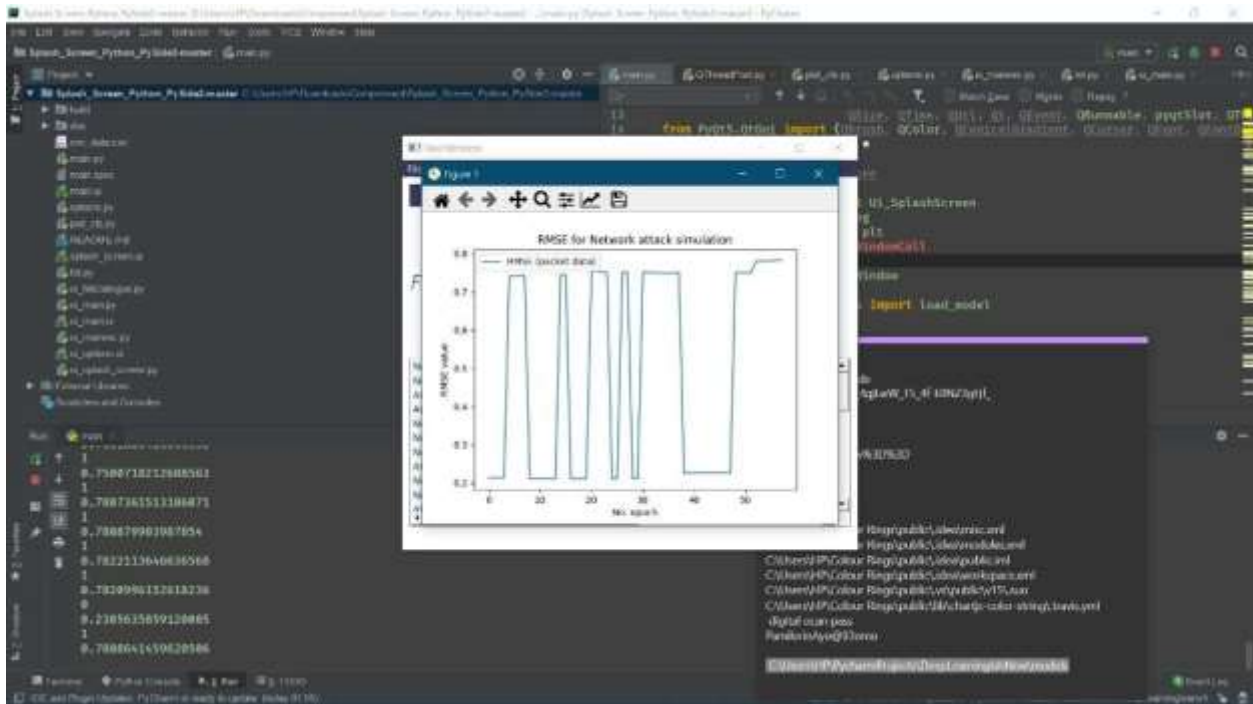


Figure 4.10: Shows a Visualization Screen with Graphs During Simulation Process

4.3 DISCUSSION

The outcome of the proposed framework testing shows that the proposed framework desired functionalities were achieved.

The testing result also showed that the conceptual features discussed such as automatic data preprocessing was achieved. The testing result also showed that the accuracy level of the framework performance took some important statistics and value into consideration. For e.g., it factored in the following statistics and values:

1. Root Mean Square Error (RMSE)
2. Mean Absolute Error (MAE)
3. Mean Square Error (MSE)
4. Confusion Matrix

4.3.1 Root Mean Square Error (RMSE)

In the study, we were able to predict the `x_normal_test` with the already trained model and find the Root Mean Square Error (RMSE). During the training process of the dataset, it was determined that the closer the value of the RMSE to 1 the larger the anomalies (outlier) and the result was then used as a trigger to

flag attacks. A threshold of 0.5 RMSE was set for normal and anything above that would be considered an attack. The closer the RMSE value to 1 defines the level of the anomalies and the dangerous the level of the attack.

The Root Mean Square Error (RMSE) is the standard deviation of residuals (prediction errors). It is frequently used measure of the difference between values predicted by a model or an estimator and the value observed.

4.3.2 Mean Absolute Error (MAE) and Mean Square Error (MSE)

The Mean Absolute Error (MAE) is one of the metrics used in compiling the model alongside the Mean Square Error (MSE). The goal of this was to be able to visualize the training process of our model. With the MAE and MSE we were able to plot the result of the MAE loss with MAE validation and also the Mean Square Error (MSE) loss and validation. Below are illustrative graphs showing the MAE and MSE during the training process.

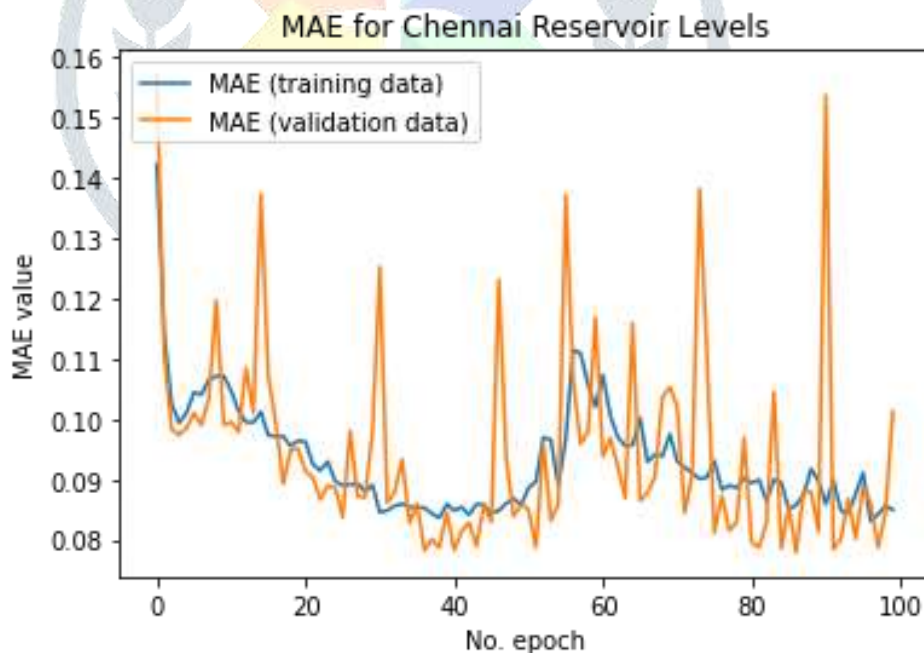


Figure 4.11: Shows the Mean Absolute Error (MAE) Level During Training Process

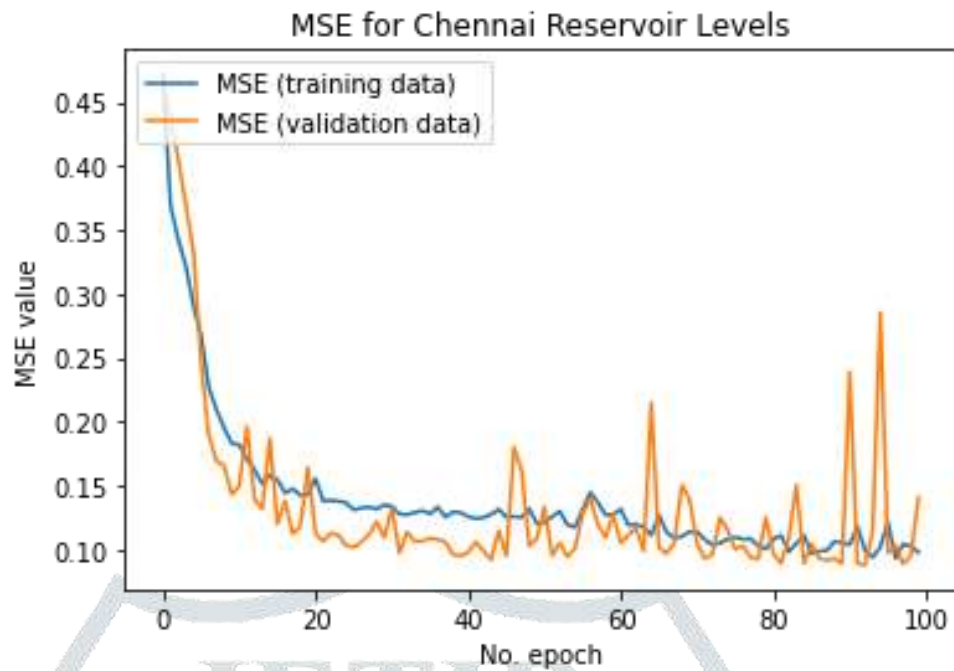


Figure 4.12: Shows the Mean Square Error (MSE) Level During Training Process

4.3.3 Confusion Metrics

Confusion matrix is used in the study to determine the amount of correct prediction during the training process. The result or output of the confusion matrix is plotted on a graph. The graph has two tables which are labeled as attack and normal. The confusion matrix shows how many attacks were predicted and how many was predicted correctly. The same conditions were determined by the number of normal flows that were predicted. Below is a graph that shows how the confusion matrix was used to predict attacks and normal flow of traffic.

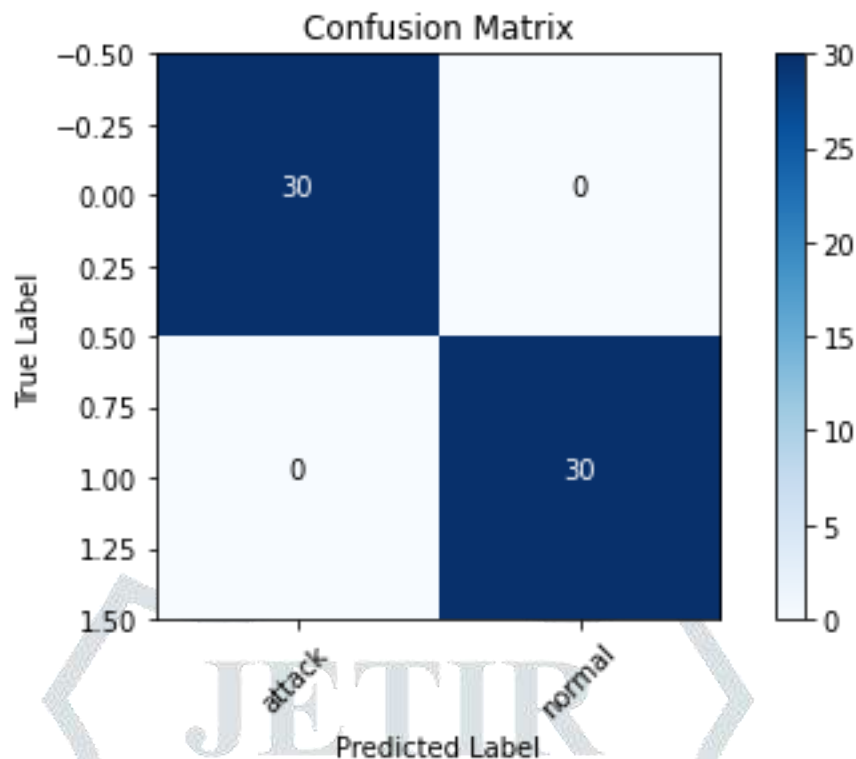


Figure 4.13: Confusion Matrix with Attack and Normal Prediction Values

The contribution of these statistics and values makes the result of the proposed framework very reliable (valid). Overall, the core objectives of the study as highlighted in the conceptual solution to the proposed framework were duly achieved in the cause of the study.

First, the proposed framework automatically handles the task of data preprocessing. Second, by using different datasets the proposed framework automatically acquired the capability to detect malicious traffics of such types. Third, the system has enough visualization tools that makes it easy for use by novice users.

CHAPTER FIVE

CONCLUSION AND RECOMMENDATION

5.1 Summary

In the first chapter of this study, the background to the study comprising of highlights of cyber-attacks, deep learning and how deep learning have been proposed to provide solutions to cyber-attacks was presented. This was followed by an elaborate articulation of the study problem statement which featured some celebrated cyber-attacks on financial institutions and the need to employ various checks and balances to mitigate these attacks using technology was mentioned. The chapter continued with the study aim and objectives indicating what the study intends to achieve and how it intended to achieve it. Next, the study defines its scope and articulated its limitations and this was then concluded with a highlight of the organization of the dissertation.

Chapter two reviewed a collection of various literature spanning cyber-attacks, online or electronic banking, deep learning covering neural networks and autoencoders and then the chapter was concluded with the studies of various authors and the difficulties associated with obtaining real life data for proper understanding of how attacks are perpetrated and for developing and testing systems for testing such attacks.

The proposed framework development methodology was undertaken in chapter three. The methodology covers the research approach, the proposed system design paradigm, the dataset used in the system training and testing, the system design methodology which was (Agile methodology) and the framework software process development model (code and fix model), and a highlight of the software development tools employed.

The implementation and testing of the system was undertaken in chapter four. The system implementation tools and the various aspects of the framework that they were used to achieve was presented using relevant code snippets to illustrate how they were used to achieve this aspect. The entire framework was then tested for its desired features and functionalities and test results were presented too.

5.2 Conclusion

This study noted that there is a dire need to find solutions to cyber-attacks in general and to financial institutions in particular. Where the money is has always been a target for attacks from time in memorial. Consequently, financial institutions remain strong center of attraction for criminals, cyber individual, gangs and groups. Incidentally, various technologies and schemes for providing veritable solutions to emerging problems are evolving. Currently the fields of machine learning and deep learning are evolving rapidly and providing reliable solutions to this escalating attacks.

A great deal of efforts is currently geared towards studies and researches in the academic communities, research institutions, and by individuals in a bid to bring this evolving area of study to full majority. This study constitutes a part that is needed towards achieving this desired majority.

Based on the firsthand experience of one of the researchers in this study, that cyber-attacks on financial institutions are a real threat, and based on the knowledge that deep learning (AI based discipline) holds a great deal of promise in combating apparently very difficult problems, this study has proposed a conceptual framework that can provide a reliable safeguard against cyberattacks.

The proposed framework employs various set of tools to design and implement a system that when properly harnessed can check the incidence of cyber-attacks in genera.

The series of testing done using the implemented system has generated results that are consistent and reliable, which implies that the proposed framework can provide a veritable solution to the problem of cyber-attacks.

5.3 Contribution to Knowledge

This study has proposed a conceptual framework that entails a substantial aspect of all that is required to develop effective safeguards against cyber-attacks which are becoming prevalence.

5.4 Recommendation

This framework was developed and tested with denial-of-service related attacks which studies show often precedes other forms of attacks. Although this framework was conceptualized and encoded to

accept various traffic types, but it was not possible to have access to other traffic types that were desired for the system training and testing. It is therefore suggested that other relevant attack data type should be sourced and used to train and test the system to know if it was possible to have an elaborate framework that can actually test the incidence of cyber-attacks.

REFERENCE

- Aamir, M., and Zaidi, M. A. (2013). "A Survey on DDoS Attack and Defense Strategies: From Traditional Schemes to Current Techniques." *Interdisciplinary Information Sciences*. 19, (2) 173-200.
- Abu-Shanab, E., and Pearson, J. M. (2019). Internet banking in Jordan: An Arabic instrument validation process. *International Arab Journal of Information Technologies*, 6(3), pp. 235-244.
- ACS, (2016). *Cybersecurity - Threats Challenges Opportunities*. Sydney NSW 2000. Vol. 1, 1st Edition, pp. 6-8. Available online: www.acs.org.au.
- Alao, A. A. (2016). Analysis of fraud in banks: Evidence from Nigeria. *International Journal of Innovative Finance and Economics Research*, 4(2), pp. 16-25.
- Alom, M. Z., and Taha, T. M. (2017). Network intrusion detection for cyber security using unsupervised deep learning approaches. In *Proceedings of the 2017 IEEE National Aerospace and Electronics Conference (NAECON)*, Dayton, OH, USA; pp. 63–69.
- Alom, M. Z., Bontupalli, V., and Taha, T. M. (2015). Intrusion detection using deep belief networks. *In Proceedings of the 2015 National Aerospace and Electronics Conference (NAECON)*, Dayton, OH, USA; pp. 339–344.
- Alrawashdeh, K., and Purdy, C. (2016). Towards an online anomaly intrusion detection system based on deep learning. *In Proceedings of the 15th IEEE International Conference Machine Learning and Applications (ICMLA)*, Miami, FL, USA; pp. 195–200.
- Adams, R. (2010). Prevent, protect, pursue: a paradigm for preventing fraud. *Computer Fraud & Security*, 2010(7), pp. 5-11.
- Adedipe, A. A. (2016). Nigerian Internet Fraud: Policy/Law changes that can improve effectiveness.
- Adeyemo K. A. (2012). Fraud in Nigerian banks: Nature, deep seated causes, aftermaths and probable remedies. *Mediterranean Journal of Social Sciences*, 3(2), pp. 279-289.
- Adewumi, O. (1986). "Fraud in banks: An overview. In *Frauds in Banks* Chartered Institute of Bankers, Nigeria.

- Anderson, R., Barton, C., Boehme, R., Levi, M., Moore, T., and Savage, S. (2012). Measuring the cost of cybercrime. Paper presented at the WEIS Conference, Berlin, Heidelberg, pp. 265-300.
- Aggarwal, C. C. (2018). *Neural Networks and Deep Learning*. International Business Machine, Yorktown Heights, NY, USA. Vol. 1:27-56.
- Agboola, A. A. and Salawu, R. O. (2008). Optimizing the use of information and communication technology (ICT) in Nigerian banks. *Journal of Internet Banking and Commerce*, 13(1), pp. 1-15.
- Association of Certified Fraud Examiners. (2015). *Report to the Nation on occupational fraud*. Austin, TX: ACFE.
- Makoto A. (2008). "Japan's Population Growth during the Past 100 Years." In *The demographic challenge: A handbook about Japan*, edited by Florian Coulmas. Leiden, Boston: Brill. pp. 5–24
- Avinash Ingole, and Thool R. C. (2013). Credit Card Fraud Detection using Hidden Markov Model and its Performance. *International Journal of Advanced Research in Computer Science and Software Engineering*, 3(6), pp. 10-14.
- BAIR (2018). Caffe deep learning framework by Berkeley Artificial Intelligence Research. Accessed 2nd June 2020. <http://caffe.berkeleyvision.org/>.
- Baldi, L., Forouzan, S., and Lu, Z. (2011). Complex-Valued Autoencoders. *Neural Networks*.
- Baldi P., and Hornik, K. (1988). Neural networks and principal component analysis: Learning from examples without local minima. *Neural Networks*, 2(1); pp. 53–58.
- Baum, S., Goertzel, B. and Goertzel, T. (2011). How long until human-level AI? Results from an expert assessment. *Technological Forecasting and Social Change*, 78(1), pp.185-195.
- BBC. (2014) "Boleto Malware May Lose Brazil \$3.75bn," BBC. Available at: <http://www.bbc.com/news/technology-28145401>. Accessed 13th July 2020.
- Bennett, M. J. (2000). A Development Approach to Training for Intercultural Sensitivity. *International Journal of Intercultural Relations*, 10, pp. 179-195 Doi:10.1016/0147-1767(86)90005-2.
- Bengio, Y. (2019). Learning deep architectures for AI. *Foundations and trends in Machine Learning*. 2: 1-127.
- BIS. (2012). *The 2011 Skills for life Survey: A Survey of Literacy, numeracy and ICT levels in England*. Department of Business Innovation and Skills.
- Boateng, R., Olumide, R., Isabalija, S., and Budu, J. (2011). Sakawa – Cybercrime and Criminality in Ghana. *Journal of Information Technology Impact*, Vol. 11, No. 2, pp. 85-100.

Boer, M., and Vazquez, J. (2017). Cyber Security and Financial Stability: How cyberattacks could materially impact the global financial system. Institute of International Finance. Available at:

<https://www.iif.com/Portals/0/Files/IIF%20Cyber%20Financial%20Stability%20Paper%20Final%2009%2007%202017.pdf?ver%3D2019-02-19-150125-767>. Bosco, F. (2012). The new cyber criminals, hackers profile project. Available online www.uncrime.if/emergingcrimes/cybercrime/. Accessed 30th May 2020.

Bostrom, N. and Shulman, C. (2016). How Hard is Artificial Intelligence? Evolutionary Arguments and Selection Effects. *Journal of Conscious Studies*, 19(7-8), pp.103-130.

Bostrom, N. and Yudkowsky, E. (2011). THE ETHICS OF ARTIFICIAL INTELLIGENCE. Draft for Cambridge Hand book of Artificial Intelligence, eds, 1(1), p.3.

Borghard, E. D., and Lonergan, S. W. (2017). "The Logic of Coercion in Cyberspace." *Security Studies* 26(3): 452-481.

Bradsher, K. (2012). "Market's Echo of Tiananmen Date Sets Off Censors," *New York Times*. Accessed 13th July 2020. Available at:

<http://www.nytimes.com/2012/06/05/world/asia/anniversary-of-tiananmen-crackdown-echos-through-shanghai-market.html/>.

Brézillon, P. (2011). From expert systems to context-based intelligent assistant systems: a testimony. *The Knowledge Engineering Review*, 26(01), pp.19-24.

Brownlee, J. (2019). What is Deep Learning: Machine Learning Mastery. Available online at <https://machinelearningmastery.com/what-is-deep-learning/>. Accessed 4th June 2020. Brunner, A. D.,

Decressin, J. W., Decressin, J., Hardy, D. C., and Kudela, B. (2004). Germany's

Three-pillar Banking System: Cross-Country Perspectives in Europe International Monetary Fraud.

Chia, T. (2012) "Confidentiality, Integrity, Availability: The Three Components of the CIA Triad," IT Security Community Blog. Available at:

<http://security.blogoverflow.com/2012/08/confidentialityintegrity-availability-the-three-components-of-the-cia-triad/>.

Cerulus, L. (2016). "Belgian Government Plagued by Hackers," *Politico*. Accessed 12th July 2020 Available at:

<http://www.politico.eu/article/belgium-government-agencies-plagued-hackers-downsec-ddos-attacks-cyber-crime/>.

CIFAS (2009). The anonymous attacker: A special report on identity fraud and account takeover. Tavistock Square London: The UK's Fraud Prevention Service.

Chainer (2018) Chainer A powerful, flexible, and intuitive framework for neural networks. (online) Available at: <https://chainer.org/index.html/>. Accessed 2nd June 2020.

Chan, H. P., Doi, K., Galhotra, S., Vyborny, C. J., MacMaho, H., Jokich, P. M. (1987). Image feature analysis and computer-aided diagnosis in digital radiography. Automated detection of microcalcifications in mammography. *Med Phys*. 14: 538-548.

Chanson, S.T., Cheung, T.W. (2001). Design and implementation of a PKI-based end-to-end secure infrastructure for mobile e-commerce. *World Wide Web*, 4(4),235-253.

Chen, T. and Guestrin, C. (2016). Xgboost: A scalable tree boosting system. In: Proceedings of the 22ND ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, pp. 785–794.

Chollet, F. (2018). Deep Learning with Python. Manning Publication Co. Vol. 2:10-15. New York, USA. ISBN 9781617294433.

Choplin, J. M., and Stark, D. P. (2013). Doomed to fail: A psychological analysis of mortgage disclosures and policy implications. *Banking & Financial Services Policy Report*, 32(10), pp. 11-19.

Choudhury, S., and Bhowal, A. (2015). “Comparative analysis of machine learning algorithms along with classifiers for network intrusion detection,” International Conference on Smart Technologies and Management for Computing, Communication, Controls, Energy and Materials (ICSTM), pp. 89–95.

CISCO. (2017). Annual Cybersecurity Report.

Cooper, P. (2014). “Anonymous Lives Up to Threats: FIFA World Cup Hacks Get Underway,” IT Pro Portal. Accessed 13th July 2020. Available at:

<http://www.itproportal.com/2014/06/13/anonymous-lives-up-to-threats-fifa-world-cup-hacks-get-underway/#ixzz41DPxOwDR>.

Cordero, C. G., Hauke, S., Muhlhauser, M., and Fischer, M. (2016). “Analyzing flow-based anomaly intrusion detection using Replicator Neural Networks,” in 2016 14th Annual Conference on Privacy, Security and Trust (PST). Auckland, New Zeland: IEEE, pp. 317–324.

Council of Europe. (2001). “Convention on Cybercrime.” Council of Europe. Accessed April 15th 2020.

Available (online) at: <https://rm.coe.int/CoERMPublicCommonSearchServices/Displa?document/>.

Cluley, G. (2014). “Corkow - the Lesser-Known Bitcoin-Curious Cousin of the Russian Banking Trojan Family,” Accessed 12th July 2020. We Live Security, Available at:

<http://www.welivesecurity.com/2014/02/11/corkow-bitcoin-russian-banking-trojan/>; and

“How malware moved the exchange rate in Russia,” We Live Security, February 12, 2016, <http://www.welivesecurity.com/2016/02/12/malware-moved-exchange-rate-russia/>.

CNTK (2018). Microsoft cognitive toolkit (CNTK), An open source deep-learning toolkit. (online) Available at: <https://docs.microsoft.com/en-us/cognitive-toolkit/>. Accessed 2nd June 2020.

Creasey, J. (2015). Cyber Security Monitoring and Logging Guide. Berkshire, UK: CREST, 2015.

Curt’s Carpet Services. (2013). Costa Mesa, United States, Costa Mesa: Experian Information Solutions. Retrieved <https://search.proquest.com/docview/1587783885?accountid=10472>.

Cyber security breaches survey. (2017). Department for Digital, Culture, Media and Sport, 19 April 2017.

<https://www.gov.uk/government/statistics/cyber-security-breaches-survey-2017>.

Deng, L., and Yu, D. (2014). Deep Learning: Methods and Applications. Foundations and Trends in Signal Processing, Vol. 7, Nos. 3-4, pp. 197-387.

Deng, L., and Chen, J. (2014). Sequence classification using the higher-level features extracted from deep neural networks. In Proceedings of International Conference on Acoustics Speech and Signal Processing (ICASSP).

Donnelly, L. (2018). “More Than 900 NHS Deaths Yearly May be Caused by IT Failings.” The Telegraph, February 6, 2018. Accessed April 30th 2020.

<https://www.telegraph.co.uk/news/2018/02/06/900-nhs-deaths-yearly-may-causedfailings/>.

Dong, B., and Wang, X. (2016). “Comparison deep learning method to traditional methods using for network intrusion detection,” 8th IEEE International Conference on Communication Software and Networks (ICCSN). Beijing, China: IEEE, pp. 581–585.

Djenouri, Y. (2019). A survey on urban traffic anomalies detection algorithms. IEEE Access, Vol. 7: pp. 12192-12205.

Dua, S., and Du, X. (2011). Data Mining and Machine Learning in Cybersecurity. Auerbach Publications, Boston, 1st edition, MA, USA.

Gao, N., Gao, L., Gao, Q., and Wang, H. (2014). An intrusion detection model based on deep belief networks. In Proceedings of the 2014 2nd International Conference Advanced Cloud and Big Data (CBD), Huangshan, China; pp. 247–252.

ECT Act, (2002). Intercom South Africa-the South African ECT. Available online www.intercomm.co.za/the-south-africa-ect-act-of-2002. Accessed 30th May 2020.

ENISA. (2014). 16 million E-identities and passwords theft. *European Union Agency for Network and Information Security*.

Erickson, B. J., Korfiatis, P., Akkus, Z., Kline, T., and Phibrick, K. (2017). Toolkits and Libraries for Deep Learning. *Journal of Digital Imaging*, pp. 1-6. Doi 10.1007/s10278-017-9965-6.

Falcini, F., Lami, G., and Costanza, A. M. (2017). “Deep Learning in Automotive Software,” IEEE Software, vol. 34, no. 3, pp. 56–63. [Online]. Available: <http://ieeexplore.ieee.org/>.

Fausett, L. (1994). Fundamentals of Neural Networks – Architecture, Algorithms, and Applications, Prentice Hall Inc., Englewood Cliffs, NJ.

Federal Chancellery of the Republic of Austria. (2013). “Austrian Cyber Security Strategy.” Accessed April 15th 2020.

https://www.bmi.gv.at/504/files/130415_strategie_cybersicherheit_en_web.pdf.

Finkle, J., and Hosenball, M. (2014). *FBI warns retailers to expect more credit card breaches*. Reuters.

Fung, B. (2018). “Equifax’s Massive 2017 Data Breach Keeps Getting Worse.” Washington Post, March 1, 2018. Accessed April 15th 2020. Available at:

<https://www.washingtonpost.com/news/the-switch/wp/2018/03/01/equifax-keeps-finding-millions-more-people-who-were-affected-by-its-massive-data-breach/>.

Ganesan, R., and Vivekanandan, K. (2009). A secured hybrid architecture model for internet banking (e-banking). *Journal of Internet Banking and Commerce*, 14(1), pp. 117.

- Gates, T., and Jacob, K. (2009). Payments fraud: Perception versus reality. *Economic Perspectives*, 33(1), pp. 7-15.
- Gerden, E. (2016). "Russian Bank Licenses Revoked for Using Hackers to Withdraw Funds," *SC Magazine UK*, Available at: <http://www.scmagazine.uk.com/>. Accessed 12th July 2020.
- Gertz, B. (2014). "Russian Cyber Warfare Suspected in Bank Attacks," *Flash//CRITIC Cyber Threat News*. Accessed 12th July 2020. Available at: <http://flashcritic.com/russian-cyber-warfare-suspected-bank-attacks-sophisticated-hackers/>.
- Giles, K., and Hagestad, W. (2013). "Divided by a Common Language: Cyber Definitions in Chinese, Russian and English." In 2013 5th International Conference on Cyber Conflict (CYCON 2013) IEEE, pp. 1-17.
- Goertzel, T. (2014). The path to more general artificial intelligence. *Journal of Experimental & Theoretical Artificial Intelligence*, 26(3), pp.343-354.
- Goodin, D. (2015). "Puzzle Box," *Ars Technica*; Kim Zetter, "Suite of Sophisticated Nation-State Attack Tools Found with Connection to Stuxnet," *Wired*. Accessed 13th July 2020.
Available at: <http://www.wired.com/2015/02/kaspersky-discovers-equation-group/>.
- Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep Learning*, MIT Press, <http://www.deeplearningbook.org>.
- Graham, J., and Howard, R. (2010). *Cyber Security Essentials*. Boca Raton, Florida, ABD: Auerbach Publications. pp.198, 199.
- Graham, J. R., Li, S., and Qiu, J. (2008). Corporate Misreporting and Bank Loan Contracting. *Journal of Financial Economics*, 89, pp. 44-61.
- Graycar, A., and Smith, R. (2002). Identifying and responding to electronic fraud risks. Paper presented at the *30th Australasian Registrars' Conference Canberra*.
- Greenspan, H., Ginneken B., Summers, R. M. (2016). Guest editorial deep learning in medical imaging: overview and future promise of an exciting new technique. *IEEE Trans Med Imaging*, pp. 1153–1159.
- Gurney, K. (1997). *An Introduction to neural networks*. UCL Press Limited, Master e-book, ISBN 0-203-456-22-X, London; pp 5-12.
- Iwuagwu, O. (2000). Corruption: A threat to democracy and national development. *Journal of National Economic Group of Nigeria*, 8(1), pp. 12-16.
- Hansen, J., McDonald, J., Messier, W., and Bell, T. (1996). A Generalized Qualitative Response Model and the Analysis of Management Fraud. *Management Science*, 42, 1022-1033.
- Hanson, D., Imran, A., Vellanki, A., Kanagaraj, S. (2018). A Neuro-Symbolic Humanlike Arm Controller for Sophia the Robot, Hanson Robotics Ltd. (online) Available at: <http://www.hansonrobotics.com/wp-content/uploads/>.

Hamilton, D. I., Justin, M., and Odinioha, G. (2012). Dimensions of fraud in Nigeria quoted firms. *American Journal of Social and Management Science*, Vol. 3, pp.112-120. Doi:10.5251/ajsms.201.

Haykin, S. (1994). *Neural Networks — A Comprehensive Foundations*, Macmillan College Publishing Co., Englewood Cliffs, NJ.

Hebbo, H., and Kim, J. W. (2013). Classification with Deep Belief Networks. (online), Available at: <https://www.ki.tuberlin.de/fileadmin/fg135/publikationen/Hebbo2013CDB.pdf>

Herman, S. (2016). “Historic Bangladesh Bank Heist Muddled in Mystery,” *Voice of America*, Available at: <http://www.voanews.com/content/historic-bangladesh-bank-heist-muddled-in-mystery/3252379.html>; Rick Gladstone

, “Bangladesh Bank Chief Resign After Cyber Theft of \$81 Million,” *New York Times*, <http://www.nytimes.com/2016/03/16/world/asia/bangladesh-bank-chief-resigns-after-cyber-theft-of-81-million.html?r=0/>. Accessed 12th July 2020.

Hirschmann, J. (2014). “Defense in Depth: A Layered Approach to Network Security.” *Security Magazine*. Available at:

<http://www.securitymagazine.com/articles/85788-defense-in-depth-a-layered-approach-to-network-security/>.

Hodo, E. Bellekens, X. J. A. Hamilton, A. Tachtatzis, C., and Atkinson, R. C. (2017). “Shallow and deep networks intrusion detection system: A taxonomy and survey,” *CoRR*, vol. abs/1701.02145, 2017. [Online]. Available: <http://arxiv.org/abs/1701.02145>.

Hoffman, C. (2013). “Hacker Hat Colors Explained: Black Hats, White Hats, and Gray Hats.” *How-To Geek*. Available at:

<http://www.howtogeek.com/157460/hacker-hatcolors-explained-black-hats-white-hats-and-gray-hats/>. Honda. (2013). “Inside ASIMO Robotics: The Technology Behind ASIMO.” Accessed July, 2020. <http://asimo.honda.com/inside-asimo/>.

Hutchins, E. M., Cloppert, M. J., and Amin, R. M. (2011). “Intelligence-Driven Computer Network Defense Informed by Analysis of Adversary Campaigns and Intrusion Kill Chains.” *Leading Issues in Information Warfare & Security Research*.

IDC. (2014). “Executive Summary: Data Growth, Business Opportunities, and the IT Imperatives - *The Digital Universe of Opportunities: Rich Data and the Increasing Value of the Internet of Things*,” MA, USA,

Idowu, A., and Adedokun, T. O. (2013). Evaluation of the effect of monitoring and control activities on fraud detection in selected Nigerian commercial banks. *Research Journal of Finance and Accounting*, .4(6), pp. 37-54.

Jassal, R. K., and Sehgal, R. K. (2013). Online Banking Security Flaws: A Study. *International Journal of Advanced Research in Computer Science and Software Engineering*, 3(8), 1016-1021.

Javaid, A., Niyaz, Q., Sun, W., and Alam, M. (2015). A deep learning approach for network intrusion detection system. In *Proceedings of the 9th EAI International Conference Bio inspired Information and Communications Technologies (Formerly BIONETICS)*, New York, NY, USA; pp. 21–26.

- Juniper Networks (2015). "Juniper Networks - How many Packets per Second per port are needed to achieve Wire-Speed?". [Online]. Available: <https://kb.juniper.net/InfoCenter/index?page=content&id=KB14737/>.
- Kalogieton, V., Lathuiliere, S., Luc, P., Lucas, T., and Shmelkov, K. (2016). Deep learning frameworks: TensorFlow, Theano, Keras, Torch and Caffe. Available online at: <https://project.inria.fr/deeplearning/files/2016/05/DLFrameworks.pdf>. Accessed 2nd June 2020.
- Kang, M. J., and Kang, J. W. (2016). "Intrusion Detection System Using Deep Neural Network for In-Vehicle Network Security," PLOS ONE, vol. 11, no. 6, pp. 155-165.
- Karnouskos, S. (2011). Stuxnet Worm Impact on Industrial Cyber-Physical System Security. In *37th Annual Conference of the IEEE Industrial Electronics Society (IECON 2011)*, Melbourne, Australia, 7-10 Accessed 20 June 2020.
- Kass, D. H. (2017). Worldwide Cyberattack Damages, Cost Estimates: Lloyd's of London Calculation. Available at: <https://cybersecurityventures.com/hackerpocalypse-original-cybercrime-report-2016/>.
- Kaspersky Lab. (2015). Global Research and Analysis Team, "The Great Bank Robbery: The Carbanak APT," Securelist (blog). Accessed 12th July 2020. Available at: <https://securelist.com/blog/research/68732/the-great-bank-robbery-the-carbanak-apt/>.
- KDD Cup Dataset. (1999). Available at: <http://kdd.ics.uci.edu/databases/kddcup99.html/>.
- Keras (2018). Keras high-level neural networks API. (online) Available at: <https://keras.io/>. Accessed 2nd June 2020.
- Kim, J., Shin, N., Jo, S. Y., and Kim, S. H. (2017). "Method of intrusion detection using deep neural network," IEEE International Conference on Big Data and Smart Computing (BigComp). Hong Kong, China: IEEE, pp. 313–316.
- Kim, J. (2019). Multivariate network traffic analysis using clustered patterns. *Computing*, 101(4): pp. 339-361.
- Kinkela, K., and Harris, P. (2014). ACFE Releases 2014 International study on Internal Fraud Investigation, Advocating Internal Audit. *Internal Auditing*, 29(5), pp. 10-14.
- Kochetkova, K. (2016). "Dozens of Banks Lose Millions to Cybercriminals Attacks," Kaspersky Lab Daily (blog), Available at: <https://blog.kaspersky.com/metel-gcman-carbanak/11236/>. Accessed 12th July 2020.
- Kottasova, I. (2016). "Russia: Foreign Hackers Are Trying to Take Down Our Banks," CNN, <http://money.cnn.com/2016/12/02/technology/russia-hack-banks-foreign/>. Accessed 12th July 2020.
- Kovach, S., and Ruggiero, W. V. (2011). Online banking fraud detection based on local and global behavior. Paper presented at *The Fifth International Conference on Digital Society*, Guadeloupe, France, pp.166-171.

KPMG. (2000). E-commerce and cybercrime: New strategies for managing the risks of exploitation. *Forensic and Litigation Services, KPMG LLP, USA.*

Kubic, T. T. (2001). Testimony Before the Committee on the Judiciary, Subcommittee on Crime, U.S.A. House of Representatives. June 12, 2001. Accessed April 14th 2020. Available at:

<https://archives.fbi.gov/archives/news/testimony/the-fbis-perspective-on-the-cybercrime-problem/>.

Kwon, K. J. (2014). "Smoking Gun: South Korea Uncovers Northern Rival's Hacking Codes," CNN. Accessed 13th July 2020. Available at:

<http://www.cnn.com/2015/04/22/asia/koreas-cyber-hacking/>.

Langner, R. (2013). "Stuxnet's Secret Twin: The Real Program to Sabotage Iran's Nuclear Facilities was far more Sophisticated Than Anyone Realized." *Foreign Policy*, November 19, 2013. Accessed 13th April 2020.

<http://foreignpolicy.com/2013/11/19/stuxnetssecret-twin/>.

Lafleur, J. M., Purvis, L. K., and Roesler, A. W. (2015). *The Perfect Heist: Recipes from Around the World*, Sandia Report, Sandia National Laboratories, Livermore, California, USA.

LeCun, Y.A.; Jackel, L.D.; Bottou, L.; Brunot, A.; Cortes, C.; Denker, J.S.; Drucker, H.; Guyon, I.; Muller, U.A.; Sackinger, E. (1995). Learning algorithms for classification: A comparison on handwritten digit recognition. In *Neural Networks*; World Scientific: London, UK, 1995; pp. 261–276.

Le, D. C., Khanchi, S., Zincir-Heywood, A. N., and Heywood, M. I. (2018). Benchmarking evolutionary computation approaches to insider threat detection. In *Proceedings of the Genetic and Evolutionary Computation Conference*; pp. 1286-1293.ACM.

Lee, H. Kim, Y., and Kim, C. O. (2017). "A Deep Learning Model for Robust Wafer Fault Monitoring with Sensor Measurement Noise," *IEEE Transactions on Semiconductor Manufacturing*, vol. 30, no. 1, pp. 23–31.

Liang, Z., Zhang, G., Huang, J. X., and Hu, Q. V. (2014). "Deep learning for healthcare decision making with EMRs," *IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, no. Cm. IEEE pp. 556–559.

Li, L., He, W., Xu, L., Ash, I., Anwar, M., and Yuan, X. (2019). Investigating the impact of cybersecurity policy awareness on employees' cybersecurity behavior. *International Journal of Information Management*, 45, pp.13-24.

Li, Y., Ma, R., and Jiao, R. (2015). A Hybrid Malicious Code Detection Method Based on Deep Learning Methods, Vol. 9, pp. 205–216.

Lippmann, R., Cunningham, R. K., Fried, D. J., Graf, I., Kendall, K. R., Webster, S. E., and Zissman, M. A. (1999). Results of the 1998 darpa offline intrusion detection evaluation. In *Proc. Recent Advances in Intrusion Detection*.

Luckow, A., Cook, M., Ashcraft, N., Weill, E., Djerekarov, E., and Vorster, B. (2016). "Deep learning in the automotive industry: Applications and tools," *IEEE International Conference on Big Data (Big Data)*. IEEE, pp. 3759–3768. [Online]. Available: <http://ieeexplore.ieee.org/document/7841045/>

Lundy, F. (2018). Cyber Threat Alert Fatigue and Reduction Methods. PhD thesis, 2017. Copy right – Database copyright ProQuest LLC; ProQuest does not claim copyright in the individual underlying works; last updated – 2018-03-02.

Manyika, J., & Roxburgh, C. (2011). The great transformer: The impact of the internet on economic growth and prosperity. *McKinsey Global Institute*.

Markoff, J. (2008). “Before the Gunfire, Cyberattacks,” *New York Times*. Accessed 13th July 2020. Available at:

<http://www.nytimes.com/2008/08/13/technology/13cyber.html>.

Masocha, R., Chilya, N., and Zindiye, S. (2011). E-banking adoption by customers in the rural milieu of South Africa: *A case of Alice, eastern cape, South Africa. African Journal of Business Management*, 5(5), pp. 1857-1863. Doi: <http://dx.doi.org/10.5897/AJBM10850/>.

McCarthy, N. (2013). “Police Officer Will Not Face Charges Over Alleged Online Abuse.” *Birmingham Live*, July 31, 2013. Accessed April 14th 2020.

McConnell, S. (1996). “Software Quality at Top Speed.” *SteveMcConnel.com*. Available at: <http://www.stevemcconnell.com/articles/art04.htm>.

Melendez, S. (2017). “Online Sleuths Are Outing Racists, But Should They?” *Fast Company*, August 19, 2017. Accessed April 14th 2020. Available at:

<https://www.fastcompany.com/40456128/ethics-of-online-sleuths-outing-racists-after-charlottesville/>.

Meyers, Nice, C. M., Becker, H. C., Nettleton, W. J., Sweeney, J. W., Meckstroth, G. R. (1964). Automated computer analysis of radiographic images. *Radiology*. 1964; pp. 1029–1034. Ministry of Information and Communications Technology. 2014. “Qatar National Cyber Security Strategy.” Accessed April 15th 2020.

http://www.motc.gov.qa/sites/default/files/national_cyber_security_strategy.pdf.

Mirsky, Y., Doitshman, T., Elovici, Y., & Shabtai, A. (2018). Kitsune: An Ensemble of Autoencoders for Online Network Intrusion Detection. *Journal of Internet Society*, San Diego, CA, USA. Available online at: <http://dx.doi.org/10.14722/ndss.2018.23204>.

Mirjana Pejic-Bach (2010). Profiling intelligent systems applications in fraud detection and prevention: *Survey of research articles*.

Moore, T., Clayton, R., and Anderson, R. (2009). The economics of online crime 23 (3), *The Journal of Economic Perspectives*, 23(3), pp. 3-20.

Mukai, T., Hirano, S., and Hosoe, S. (2009). “kaigo robotto rīman (Care Robot RI-MAN).” *Robotto, tokushū:iryō.fukushi robotto* (188); pp. 46–52.

Naseer, S., Saleem, Y., Khalid, S., Bashir, M. K., Han, J., Iqbal, M. M., and Han, K. (2018).

Enhanced network anomaly detection based on deep neural networks, *IEEE Access*, vol. 6, pp. 48231–48246.

Nazir, S., Patel, S., and Patel, D. (2018). Hyper parameters selection for image classification in convolutional neural networks. *17th IEEE International Conference on Cognitive Informatics and Cognitive Computing*, Berkeley, CA, pp. 401-407.

New Zealand Department of the Prime Minister and Cabinet. (2015). "National Plan to Address Cybercrime." Accessed April 15th 2020. <https://www.dpmc.govt.nz/sites/default/files/2017-03/nz-cyber-security-cybercrimeplan-december-2015.pdf>.

Nguyen, K.K., Hoang, D.T., Niyato, D., Wang, P., Nguyen, P., and Dutkiewicz, E. (2018). Cyberattack detection in mobile cloud computing: A deep learning approach. In *Proceedings of the 2018 IEEE Wireless Communications and Networking Conference (WCNC)*, Barcelona, Spain; pp. 1–6.

Nie D, Dong N, Li W, Yaozong G, Dinggang S. (2016). Fully convolutional networks for multi modality iso-intense infant brain image segmentation. *IEEE 13th International Symposium on Biomedical Imaging (ISBI)*.

Nielsen, M. A. (2015). *Neural Networks and Deep Learning*, Vol. 25, Determination Press, San Francisco, CA, USA.

Niyaz, Q., Sun, W., Javaid, A. Y., and Alam, M. (2015). A deep learning approach for network intrusion detection system. In *Proceedings of the 9th EAI International Conference on Bio-inspired Information and Communication Technologies (Formerly BIONETICS), BICT 15*, Vol.15, pp 21-26.

Nor, K. M., Shanab, E. A. A., and Pearson, J. M. (2008). Internet Banking Acceptance in Malaysia Based on the Theory of Reasoned Action. *JISTEM-Journal of Information Systems and Technology Management*, 5(1), pp. 3-14.

Norse, (2014). *Account Takeover: A Complex and Growing Problem* Norse Corporation.

Odediran, O. (2014). Holistic approach to electronic channels fraud management. Nigeria Electronic Fraud Forum (NeFF) 2014 Annual Report.

Ojo, (2008). Effect of frauds on banking operations in Nigeria. *International Journal of Investment and Finance*, 1(1), pp. 103.

O'Toole, J. (2015). "JPMorgan: 76 Million Customers Hacked," CNN, October 3, 2014, <http://money.cnn.com/2014/10/02/technology/security/jpmorgan-hack/?iid=EL>; Jose Pagliery, "JPMorgan's Accused Hackers Had Vast \$100 Million Operation," CNN, Available at: <http://money.cnn.com/2015/11/10/technology/jpmorgan-hack-charges/>.

Accessed 13th July 2020.

Omar, A. B., Sultan, N., Zaman, K., Bibi, N., Wajid, A., and Khan, K. (2011). Customer Perception Towards Online Banking Services: Empirical Evidence from Pakistan. *Journal of Internet Banking and Commerce*, 16(2), pp. 1-24.

Omariba, Z., Masese, N., and Wanyembi, G. (2012). Security and Privacy of Electronic Banking, *IJCSI International Journal of Computer Science*, 9(3), pp. 432-446.

Pandey, M., (2010). A model for managing online fraud risk using transaction validation. *The Journal of Operational Risk*, 5(1), pp. 49-63.

- Pandy, S. (2016). *Payment strategies: Mitigating fraud risk in the card- not Present1 environment* Federal Reserve Banks of Boston and Atlanta, Vol.1, pp. 2-8.
- Pant, B., Rus, D., and Shrobe, H. (2016). *Cybersecurity: Technology, application and policy*. Spring 2016.
- Parti, K., Tibor, K., and Koplányi, G. (2018). “Architecture of Aggression in Cyberspace: Testing Cyber Aggression in Young Adults in Hungary.” *International Journal of Cybersecurity Intelligence & Cybercrime* 1(1): 56-68.
- Park, S. (2015). *Winning the online banking war*. Blackhat USA: TrendMicro.
- Park, M., and Calif. (2016). *Hackerpocalypse: A Cybercrime Revolution*. First official Annual Crime Report. Ventures Cybercrime Magazine, Cybersecurity Venture. Available at:
<http://www.cybersecurityventures.com/hackerpocalypse-original-cybercrime-report-2016/>.
- Parsaei, M.R. (2017). Network traffic classification using machine learning techniques over software defined networks. *International Journal of Advanced Computer Science and Applications*. 8(7): pp. 220-225.
- Pedneault, S., Silverstone, H., Rudewicz, F., and Sheetz, M. (2012). *Forensic accounting and fraud investigation for non-experts* John Wiley & Sons.
- Perlroth, N., and Gelles, D. (2014). *Russian hackers amass over a billion internet passwords*. New York: New York Times.
- Perlroth, N., Scott, M., and Frenkel, S. (2017). “Cyberattack Hits Ukraine Then Spreads Internationally.” *New York Time*. Accessed April 30th 2020.
<https://www.nytimes.com/2017/06/27/technology/ransomware-hackers.html>.
- Peotta, L., Holtz, M., David, B., Deus, F., and Sousa Jr, R. (2011). A formal classification of interest banking attacks and vulnerabilities. *International Journal of Computer Science & Information Technology (IJCSIT)*, 3(1), pp. 186-197.
- Potluri, S., and Diedrich, C. (2016). “Accelerated deep neural networks for enhanced Intrusion Detection System,” *IEEE 21st International Conference on Emerging Technologies and Factory Automation (ETFA)*, vol. 2. Berlin, Germany: pp. 1–8.
- PyTorch (2018). *PyTorch—deep learning framework that puts python first*. (online) Available at: <http://pytorch.org/>. Accessed 2nd June 2020.
- Ransbotham, S., and Mitra, S. (2009). Choice and chance: A conceptual model of paths to information security compromise. *Information Security Research*, 20(1), pp. 121-139.
- Regha, O. (2015). *Cybercrime: A risk information Centre to the rescue*. *Nigeria Electronic Fraud Forum (NeFF) 2015 Annual Report*, 72-77.
- Reuter News, (2015). ‘Ukraine says Russia behind cyber- attack on German government’, *Reuters Report* on Thu Jan 8, 2015 2:55am EST, available at:
<http://www.reuters.com/article/2015/01/08/us-germany- cyberattack/>
- Rezvy, S., Petridis, M., Lasebae, A and Zebin, T. (2018). *Intrusion Detection and Classification with Autoencoded Deep Neural Network: 11th International Conference, SecITC, Bucharest, Romania*.

- Riley, M., & Robertson, J. (2015). "Cyberspace Becomes Second Front in Russia's Clash With NATO," Bloomberg. Accessed 12th July 2020. Available at: <http://www.bloomberg.com/news/articles/2015-10-14/cyberspace-becomes-second-front-in-russia-s-clash-with-nato>
- Riley, M. (2014). "How Russian Hackers Stole the Nasdaq," Bloomberg. Available at: <http://www.bloomberg.com/bw/articles/2014-07-17/how-russian-hackers-stole-the-Nasdaq>. Accessed 13th July 2020.
- Rosewarne, C. (2013). The South African Cyber threat Barometer. Available online www.wolfpackrisk.com/. Accessed 30th May 2020.
- Rosenblatt, F. (1958). "The perceptron: A probabilistic model for information storage and organization in the brain," vol. 65, pp. 386 – 408, 12 1958.
- Rudner, M. (2013). Cyber threats to critical national infrastructure: An Intelligence challenge. *International Journal of Intelligence and Counterintelligence*, 26(3), pp. 453-481.
- Rutenberg, J. (2017). "RT, Sputnik and Russia's New Theory of War." *New York Times Magazine*, September 13, 2017. Accessed April 14th 2020. Available at: <https://www.nytimes.com/2017/09/13/magazine/rt-sputnik-and-russias-new-theory-ofwar.html/>.
- M. Sachdeva, K. Kumar, and G. Singh. (2016). "A comprehensive approach to discriminate ddos attacks from flash events," *Journal of Information Security and Applications*, vol. 26, pp. 8–22.
- Sachdeva, M., Kumar, K., and Singh, G. (2016). "A comprehensive approach to discriminate ddos attacks from flash events," *Journal of Information Security and Applications*, vol. 26, pp. 8– 22.
- Saleh, Z. I. (2011). Improving security of online banking using RFID. *Academy of Banking Studies Journal*, 10(2), pp.1.
- Sarikaya, R., Hinton, G. E., and Deoras, A. (2014). "Application of Deep Belief Networks for Natural Language Understanding," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 22, no. 4, pp. 778-784, doi: 10.1109/TASLP.2014.2303296.
- Sato, S., Guo, S., Inada, S., and Mukai, T. (2012). "Design of Transfer Motion and Verification Experiment of Care Assistant Robot RIBA-II." *Transactions of the Japan Society of Mechanical Engineers C-Series* 78 (789): pp. 1899–1912. doi:10.1299/kikaic.
- Saxena, S., and Soni, H. K. (2018, February). Strategies for Ransomware Removal and Prevention. *In 2018 Fourth International Conference on Advances in Electrical, Electronics, Information, Communication and Bioinformatics (AEEICB)*; pp. 1-4. IEEE.
- Sayfayn, N., and Madnick, S. (2017). "Cyber safety Analysis of the Maroochy Shire Sewage Spill, Working Paper CISL 2017-09." Cybersecurity Interdisciplinary Systems Laboratory (CISL), *Sloan School of Management, Massachusetts Institute of Technology*, May 2017. Accessed April 23, 2020. <http://web.mit.edu/smadnick/www/wp/2017-09.pdf>
- Schatz, D; and Wall, J. (2017). "Towards a More Representative Definition of Cyber Security". *Journal of Digital Forensics, Security and Law*. 12 (2). ISSN 1558-7215. Archived from the original on 28 December 2017.
- Shiravi, A.; Shiravi, H.; Tavallae, M.; Ghorbani, A.A. (2012). Towards developing a systematic approach to generate benchmark datasets for intrusion detection. *Computer. Security*. 31, pp.357–374.

- Schlein, T. (2014). "The Five Tough Truths of Cybersecurity." TechCrunch. <https://techcrunch.com/2014/05/31/the-five-tough-truths-of-cybersecuritysoftware/>.
- Schwartz, M. J. (2014). "Bank Attackers Restart Operation Ababil DDoS Disruptions," Dark Reading, Accessed 13th July 2020. Available at: <http://www.darkreading.com/attacks-and-breaches/bank-attackers-restart-operation-ababil-ddos-disruptions/d/d-id/1108955>.
- Schmidhuber, J. (2015). "Deep Learning in Neural Networks: An overview," *Neural Networks*, vol. 61, pp. 85–117.
- Schneier, B. (2011). *Secrets and lies: Digital Security in a Networked world*. John Wiley & Sons.
- Sekharan, S. S., and Kandasamy, K. (2017). Profiling SIEM tools and correlation engines for security analytics. In *Proclamation of International Conference on Wireless Communications., Signal Processing Network. (WiSPNET)*, pp. 717–721.
- Shafiq, M. (2016). Network traffic classification techniques and comparative analysis using machine learning algorithms. *2nd IEEE International Conference on Computer and Communications (ICCC)*. IEEE.
- Shahbar, K. and Zincir-Heywood, A. N. (2018). How far can we push flow analysis to identify encrypted anonymity network traffic? in *NOMS IEEE/IFIP Network Operations and Management Symposium*. IEEE.
- Shashikumar, S. P., Shah, A. J., Li, Q., Clifford, G. D., and Nemati, S. (2017). "A deep learning approach to monitoring and detecting atrial fibrillation using wearable technology," *IEEE EMBS International Conference on Biomedical & Health Informatics (BHI)*. Florida, USA: IEEE, pp. 141–144.
- Sheldon, J. (2013). *The Rise of Cyberpower. In Strategy in the contemporary world*. New York. Oxford University Press, pp. 303-319.
- Shen, Y., Mariconti, E., Vervier, P. A and Stringhini, G. (2018). Tiresias: Predicting security events through deep learning. In *Proclamation of ACM CCS, Toronto, ON, Canada*, pp. 592–605.
- Sharafaldin, I., Gharib, A., Lashkari, A. H., Ghorbani, A. A. (2017). Towards a reliable intrusion detection benchmark dataset. *Software Networking*. Vol. 1; pp 177–200.
- Shiravi, A., Shiravi, H., Tavallaei, M., and Ghorbani, A. A. (2012). Toward developing a systematic approach to generate benchmark datasets for intrusion detection. *Computer Security*, 31(3): pp. 357–374.
- Siegel, A. A., and Tucker, J. A. (2018). "The Islamic State's Information Warfare: Measuring the Success of ISIS's Online Strategy." *Journal of Language and Politics* 17(2): pp. 258-280. Siegel, A. A., and Tucker, J. A. (2018). "The Islamic State's Information Warfare: Measuring the Success of ISIS's Online Strategy." *Journal of Language and Politics* 17(2): 258-280.
- Silverstone, H., & Sheetz, M. (Eds.). (2007). *Forensic accounting and fraud investigation for non-experts* (2nd ed.). New Jersey, USA: John Wiley & Sons, Inc.
- Small, P. E. (2011). *Defense in Depth: An Impractical Strategy for a Cyber World*. Bethesda, MD:

SANS Institute.

Singer, P. and Friedman, A. (2014). *Cybersecurity and Cyberwar: What everyone needs to know*: New York: Oxford University Press.

Singer, P. W., and Brooking, E. T. (2018). *Like War: The Weaponization of Social Media*. New York: Houghton Mifflin Harcourt.

Sochor, T., and Zuzcak, M. (2014). Study of Internet Threats and attacks methods using honeypots and honeynets. *In International Conference on Computer Networks*, pp. 118-127.

Sommer, R. and V. Paxson. (2010). Outside the closed world: On using machine learning for network intrusion detection. *IEEE symposium on security and privacy*. IEEE.

SouthFront (2015). "Cyberberkut Hacked the Site of Ukrainian Ministry of Finance: The Country has No Money," Accessed 12th July 2020. Available at:

<https://southfront.org/cyberberkut-hacked-the-site-of-ukrainian-ministry-of-finance-the-country-has-no-money/>.

Span, M. T., Mailloux, L. O., Grimaila, M. R., and Young, W. B. (2018). A Systems Security Approach for Requirements Analysis of Complex Cyber-Physical Systems. *International Conference on Cyber Security and Protection of Digital Services (Cyber Security)* pp. 1-8. IEEE.

Stafford, R. (2010). Constraints of Biological Neural Networks and Their Consideration in AI Applications. *Advances in Artificial Intelligence*, 2010, pp.1-6.

Sullivan, R. J. (2014). Controlling security risk and fraud in payment systems. *Economic Review - Federal Reserve Bank of Kansas City*, pp. 5-36.

Sun, Y. (2016). The Study on Network Information Security. *International Conference on Network and Information Systems for Computers*, pp. 85-88.

Symantec Security Response. (2015). "Phishing in the middle of the stream" - Today's threats to online banking. *Paper presented at The AVAR 2015 Conference, Symantec Security Response*, Dublin, pp. 1-28.

Tang, T. A., Mhamdi, L., McLernon, D., Zaidi, S. A. R., and Ghogho, M. (2016). "Deep learning approach for network intrusion detection in software defined networking," in 2016 International Conference on Wireless Networks and Mobile Communications (WINCOM). IEEE, pp. 258–263.

Tamkin, E. (2017). "10 Years After the Landmark Attack on Estonia, Is the World Better Prepared for Cyber Threats?" *Foreign Policy*, April 27, 2017. Accessed April 30th 2020. <https://foreignpolicy.com/2017/04/27/10-years-after-the-landmark-attack-on-estonia/>.

Tariq, N. (2018). Impact of cyberattacks on financial institutions. *A Journal of Internet Banking and Commerce*, Vol.23, no.2. Available at: <http://www.icommercecentral.com>.

Taylor, J. (2011). *Forensic accounting*. (1st ed.). Edinburg Gate Harlow Essex CM20 2JE, England: Pearson education.

Taylor, V.F. (2017). Robust smartphone app identification via encrypted network traffic analysis. *IEEE Transactions on Information Forensics and Security*, 13(1); pp. 63-78.

Tech Times, (2014). 'U.S. Investigators Believe North Korea Hired Hackers for Sony Cyberattack'.

<http://www.techtimes.com/articles/23774/20141230/u-s-investigators-believe-north-korea-hiredhackers-for-sony-cyberattack.htm/>.

TensorFlow (2018) TensorFlow, an open-source software library for machine intelligence. (online), Available at: <https://www.tensorflow.org/>. Accessed 2nd June 2020.

Jones, T. (2013). Deep learning architectures. (online), Available at:

<https://www.ibm.com/developerworks/library/cc-machinelearning-deep-learning-architectures/index.html>. Timberg, C (2015). “Net of Insecurity: A Flaw in the Design.” Washington Post. Available at: <http://www.washingtonpost.com/>.

Toppa, S. (2017). “Abuse in Pakistan: ‘I’m More Scared of Harassment Online Than Offline. The Guardian, August 9, 2017. Accessed April 14th 2020.

Topping, B. H. V., and Khan, A. I. (1993). Neural Networks and Combinatorial Optimization in Civil and Structural Engineering, Civil-Comp Press, Edinburgh.

Turing, A.M. (1950). Computing Machinery and Intelligence. Mind: a quarterly review. Blackwell for the Mind Association.

Uchenna, C., and Agbo J. C. (2013). Impact of Fraud and Fraudulent Practices on Performance of Banks in Nigeria. *British Journal of Arts and Social Science*, 15(1), pp. 2-5.

Udoayang, J. O., and James, F. U. (2004). Auditing and Investigation. Calabar: University of Calabar Press.

US Department of Defense. (2013). Joint Publication 3-12 (R0 Cyberspace operations. [online] Available at: <http://www.dtic.mil/doctrine/newpubs/jp313.pdf>. Accessed 14th July 2020.

United Nations Office on Drugs and Crime. n.d. “Global Programme on Cybercrime.” (E.d) United Nations. Accessed April 15th 2020. <https://www.unodc.org/unodc/en/cybercrime/global-programme-cybercrime.html/>.

Usman, A. K., and Shah, M. H. (2013). Critical Success Factors for Preventing E-banking Fraud. *Journal of Internet Banking and Commerce*, 18(2), pp. 1-15.

Vince, F. (2011). ‘Cyber Attacks: Prevention and Proactive Responses’, Practical Law Publishing Limited and Practical law Company, 2011, p. 1.

Valeriano, B., and Maness, R. C. (2015). Cyber War Versus Cyber Realities: Cyber Conflict in the International System. Oxford, UK: Oxford University Press.

Välja, M., Korman, M., and Lagerström, R. (2017). A Study on Software Vulnerabilities and Weaknesses of Embedded Systems in Power Networks. In Proceedings of the 2nd Workshop on Cyber-Physical Security and Resilience in Smart Grids; pp. 47-52.

Wang, W. Y. C., Rashid, A., & Chuang, H. (2011). Toward the trend of cloud computing. *Journal of Electronic Commerce Research*, 12(4), pp. 238.

Wang, Y. Cai, W., and Wei, P. (2016). “A deep learning approach for detecting malicious JavaScript code,” Security and Communication Networks, vol. 9, no. 11, pp. 1520–1534.

Wang, W., Zhu, M., Zeng, X., Ye, X., and Sheng, Y. (2017). Malware traffic classification using convolutional neural network for representation learning. In *Proceedings of the IEEE 2017 International Conference on Information Networking (ICOIN)*, Da Nang, Vietnam; pp. 712–717.

- Wei, W., Li, J., Cao, L., Ou, Y., & Chen, J. (2013). Effective detection of sophisticated online banking fraud on extremely imbalanced data. *World Wide Web (Internet and Web Information Systems)*, 1-29. doi:10.1007/s11280-012-0178-0/.
- Weissbrodt, D. (2013). "Cyber-Conflict, Cyber-Crime, and Cyber-Espionage." *Minnesota Journal of International Law* 22(2): pp. 347-387.
- Wells, J. (2014). *Corporate fraud handbook: Prevention and detection* (2nd ed.). London: John Wiley and Sons
- Wilhelm, W. K. (2004). The fraud management lifecycle theory: A holistic approach to fraud management. *Journal of Economic Crime Management*, 2(2), 1-38.
- Yan, W. N., and Chiu, D. K. (2007). Enhancing e-commerce processes with alerts and web services: A case study on online credit card payment notification. *Paper presented at the Machine Learning and Cybernetics, 2007 International Conference*, 3831-3837.
- Yang, L. X., Li, P., Yang, X., Wen, L., Wu, Y., and Tang, Y. Y. (2017). Security evaluation of cyber networks under advanced persistent threats. arXiv preprint arXiv:1707.03611.
- Ye, J. C., Han, Y., Cha, E. (2018). Deep Convolutional Framelets: A General Deep Learning Framework for Inverse Problems. *SIAM J Imaging Sci.*, pp. 991-1048
- Yonhap. (2011). "Prosecution Says N. Korea Behind Nonghyup's Network Breakdown," Accessed 13th July 2020. Available at: <http://english.yonhapnews.co.kr/national/>.
- Young, H., Huffaker, B., Andersen, D., Aben, E., Shannon, C., Luckie, M., and Claffy, K. (2011). The Caida ipv4 routed/24 topology dataset. (Online) Available at: URL http://www.caida.org/data/active/ipv4_routed_24_topology_dataset.Xml/.
- Yousefi-Azar, M., Varadharajan, V., Hamey, L., and Tupakula, U. (2017). Autoencoder-based feature learning for cyber security applications. *In Proceedings of the 2017 International Joint Conference Neural Networks (IJCNN)*, Anchorage, AK, USA; pp. 3854–3861.
- Yu, Y.; Long, J.; Cai, Z. (2017). Network intrusion detection through stacking dilated convolutional autoencoders. *Securing. Communication. Networks*; pp. 418-4196.
- Zee News, (2014). India 2nd in list of countries facing cyber- attack on mobile's, Last Updated and First Published Sunday, March 02, 2014, 12:53, Reuters, Published by Zee News, available at: http://zeenews.india.com/business/news/technology/india-2nd-in-list-of-countriesfacing-cyber-attackon-mobiles_95434.html/.
- Zee News (2014). Kremlin website hit by "powerful" cyber-attack', last Updated and First Published: Friday, March 14, 2014, 16:15, Reuters, Published by Zee News, available at: <http://zeenews.india.com/news/world/kremlin-website-hit-by-andquot-powerfulandquot-cyberattack.html>.
- Zetter, K. (2012) "Flame and Stuxnet Cousin Targets Lebanese Bank Customers, Carries Mysterious Payload," *Wired*. Accessed 13th July 2020. Available at: <http://www.wired.com/2012/08/gauss-espionage-tool/all/>.
- Zhang, H. (2012). User intention-based traffic dependence analysis for anomaly detection. *IEEE Symposium on Security and Privacy Workshops*. IEEE.

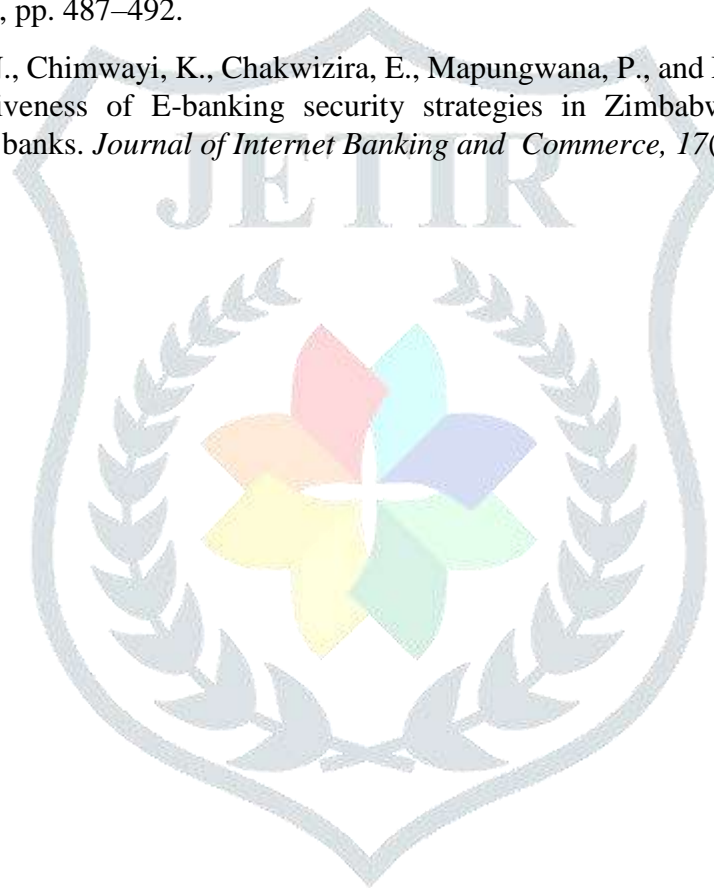
Zhang, K., Xu, J., Min, M. R., Jiang, G., Pelechris, K., and Zhang, H. (2016). Automated IT system failure prediction: A deep learning approach,” *In Proclamation of IEEE International Conference on Big Data (IEEE BigData)*, Washington, DC, USA, pp. 1291–1300.

Zhong, B., Liu, J., Du, Y., Liao Zheng Y., and Pu, J. (2016). Extracting Attributes of Named Entity from Unstructured Text with Deep Belief Network. *International Journal of Database Theory and Application* Vol.9, No.5, pp.187-196.

Zhou, H., Wu, C., Yang, C., Wang, P., Yang, Q., Lu, Z., & Cheng, Q. (2018). SDN-RDCD: A Real-Time and Reliable Method for Detecting Compromised SDN Devices. *IEEE/ACM Transactions on Networking (TON)*, 26(5), pp. 2048-2061.

Zhu, B., Liu, J. Z., Cauley, S.F., Rosen, B. R., Rosen, M. S. (2018). Image reconstruction by domain-transform manifold learning. *Nature*, pp. 487–492.

Zimucha, T., Zanamwe, N., Chimwayi, K., Chakwizira, E., Mapungwana, P., and Maduku, T. (2012). An evaluation of the effectiveness of E-banking security strategies in Zimbabwe: A case study of Zimbabwean commercial banks. *Journal of Internet Banking and Commerce*, 17(3), 1-16.



APPENDICES

APPENDIX A (Import Libraries)

The major frameworks needed for the deep learning, processing and viewing dataframes table as well as array manipulations were first imported, after which we read all CSV files for several type of DDOS attack alongside BENING data (data with no attack) and also appending them to each other so as to result in one dataframe rather than several dataframes.

```
import tensorflow as tf
import numpy as np
from tensorflow import keras
import pandas as pd

# read each csv files for ddos attacks alongside the Bening which represents normal
# packetflow(no attack) to be used for model training
df_DrDoS_DNS = pd.read_csv("C:\\Users\\HP\\Downloads\\CSV-01-12\\01-12\\DrDoS_DNSE.csv")
df_DrDoS_LDAP = pd.read_csv("C:\\Users\\HP\\Downloads\\CSV-01-12\\01-12\\DrDoS_LDAPPE.csv")
df_DrDoS_MSSQL = pd.read_csv("C:\\Users\\HP\\Downloads\\CSV-01-12\\01-12\\DrDoS_MSSQLE.csv")
df_DrDoS_NetBIOS = pd.read_csv("C:\\Users\\HP\\Downloads\\CSV-01-12\\01-12\\DrDoS_NetBIOSE.csv")
df_DrDoS_NTP = pd.read_csv("C:\\Users\\HP\\Downloads\\CSV-01-12\\01-12\\DrDoS_NTPE.csv")
df_DrDoS_SSDP = pd.read_csv("C:\\Users\\HP\\Downloads\\CSV-01-12\\01-12\\DrDoS_SSDPE.csv")
df_DrDoS_UDP = pd.read_csv("C:\\Users\\HP\\Downloads\\CSV-01-12\\01-12\\DrDoS_UDPE.csv")
df_PORTMAP = pd.read_csv("C:\\Users\\HP\\Downloads\\CSV-01-12\\01-12\\PORTMAP.csv")
df_Syn = pd.read_csv("C:\\Users\\HP\\Downloads\\CSV-01-12\\01-12\\SynE.csv")
df_TFTP = pd.read_csv("C:\\Users\\HP\\Downloads\\CSV-01-12\\01-12\\TFTPE.csv")
df_UDPLag = pd.read_csv("C:\\Users\\HP\\Downloads\\CSV-01-12\\01-12\\UDPLagE.csv")
df_BENING = pd.read_csv("C:\\Users\\HP\\Downloads\\CSV-01-12\\01-12\\BENING.csv")

# appends all dataframe to the df_DrDos_DNS dataframe and assigns this dataframe to df_init
# this way there is just one reference to all dataframes
df_init = df_DrDoS_DNS.append(df_DrDoS_LDAP,ignore_index=True, sort=False)\
.append(df_DrDoS_MSSQL,ignore_index=True, sort=False)\
.append(df_DrDoS_NetBIOS,ignore_index=True, sort=False)\
.append(df_DrDoS_NTP,ignore_index=True, sort=False)\
.append(df_DrDoS_SSDP,ignore_index=True, sort=False)\
.append(df_DrDoS_UDP,ignore_index=True, sort=False)\
.append(df_PORTMAP,ignore_index=True, sort=False)\
.append(df_Syn,ignore_index=True, sort=False)\
.append(df_TFTP,ignore_index=True, sort=False)\
```

```
.append(df_UDPLag,ignore_index=True, sort=False)\
.append(df_BENING,ignore_index=True, sort=False)
# set the display limit for rows and columns then display the table to be able
to view the data pd.set_option('display.max_columns', 15)
pd.set_option('display.max_rows', 5) display(df_init[0:5])
```

APPENDIX B (Validation Data)

Epoch 1/100

1536/1536 [=====] - 9s 6ms/step - loss: 0.4720 - mean_absolute_error: 0.1422 - mse: 0.4720 - val_loss: 0.4683 - val_mean_absolute_error: 0.1571 - val_mse: 0.4683

Epoch 2/100

1536/1536 [=====] - 9s 6ms/step - loss: 0.3689 - mean_absolute_error: 0.1141 - mse: 0.3689 - val_loss: 0.4354 - val_mean_absolute_error: 0.1107 - val_mse: 0.4354

Epoch 3/100

1536/1536 [=====] - 9s 6ms/step - loss: 0.3427 - mean_absolute_error: 0.1025 - mse: 0.3427 - val_loss: 0.4059 - val_mean_absolute_error: 0.0983 - val_mse: 0.4059

Epoch 4/100

1536/1536 [=====] - 9s 6ms/step - loss: 0.3220 - mean_absolute_error: 0.0994 - mse: 0.3220 - val_loss: 0.3698 - val_mean_absolute_error: 0.0974 - val_mse: 0.3698

Epoch 5/100

1536/1536 [=====] - 9s 6ms/step - loss: 0.2907 - mean_absolute_error: 0.1010 - mse: 0.2907 - val_loss: 0.3328 - val_mean_absolute_error: 0.0985 - val_mse: 0.3328

Epoch 6/100

1536/1536 [=====] - 9s 6ms/step - loss: 0.2675 - mean_absolute_error: 0.1044 - mse: 0.2675 - val_loss: 0.2462 - val_mean_absolute_error: 0.1010 - val_mse: 0.2462

Epoch 7/100

1536/1536 [=====] - 9s 6ms/step - loss: 0.2271 - mean_absolute_error: 0.1040 - mse: 0.2271 - val_loss: 0.1903 - val_mean_absolute_error: 0.0990 - val_mse: 0.1903

Epoch 8/100

1536/1536 [=====] - 9s 6ms/step - loss: 0.2097 - mean_absolute_error: 0.1065 - mse: 0.2097 - val_loss: 0.1693 - val_mean_absolute_error: 0.1034 - val_mse: 0.1693

Epoch 9/100

1536/1536 [=====] - 9s 6ms/step - loss: 0.1958 - mean_absolute_error: 0.1071 - mse: 0.1958 - val_loss: 0.1660 - val_mean_absolute_error: 0.1197 - val_mse: 0.1660

Epoch 10/100

9s 6ms/step - loss: 0.1832 - mea

n_absolute_error: 0.1072 - mse: 0.1832 - val_loss: 0.1431 - val_mean_absolute
_error: 0.0990 - val_mse: 0.1431

Epoch 11/100

1536/1536 [=====] - 9s 6ms/step - loss: 0.1821 - mea
n_absolute_error: 0.1046 - mse: 0.1821 - val_loss: 0.1498 - val_mean_absolute
_error: 0.0995 - val_mse: 0.1498

Epoch 12/100

1536/1536 [=====] - 10s 6ms/step - loss: 0.1711 - me
an_absolute_error: 0.1014 - mse: 0.1711 - val_loss: 0.1964 - val_mean_absolut e_error:
0.0979 - val_mse: 0.1964

Epoch 13/100

1536/1536 [=====] - 9s 6ms/step - loss: 0.1625 - mea
n_absolute_error: 0.0995 - mse: 0.1625 - val_loss: 0.1395 - val_mean_absolute
_error: 0.1085 - val_mse: 0.1395

Epoch 14/100

1536/1536 [=====] - 9s 6ms/step - loss: 0.1517 - mea
n_absolute_error: 0.0995 - mse: 0.1517 - val_loss: 0.1320 - val_mean_absolute
_error: 0.1011 - val_mse: 0.1320

Epoch 15/100

1536/1536 [=====] - 9s 6ms/step - loss: 0.1585 - mea
n_absolute_error: 0.1012 - mse: 0.1585 - val_loss: 0.1874 - val_mean_absolute
_error: 0.1372 - val_mse: 0.1874

Epoch 16/100

1536/1536 [=====] - 9s 6ms/step - loss: 0.1547 - mea
n_absolute_error: 0.0973 - mse: 0.1547 - val_loss: 0.1201 - val_mean_absolute
_error: 0.1071 - val_mse: 0.1201

Epoch 17/100

1536/1536 [=====] - 9s 6ms/step - loss: 0.1445 - mea
n_absolute_error: 0.0971 - mse: 0.1445 - val_loss: 0.1382 - val_mean_absolute
_error: 0.0994 - val_mse: 0.1382

Epoch 18/100

1536/1536 [=====] - 9s 6ms/step - loss: 0.1477 - mea
n_absolute_error: 0.0972 - mse: 0.1477 - val_loss: 0.1126 - val_mean_absolute
_error: 0.0893 - val_mse: 0.1126

Epoch 19/100

1536/1536 [=====] - 9s 6ms/step - loss: 0.1427 - mea
n_absolute_error: 0.0955 - mse: 0.1427 - val_loss: 0.1180 - val_mean_absolute _error:
0.0948 - val_mse: 0.1180

Epoch 20/100

1536/1536 [=====] - 9s 6ms/step - loss: 0.1432 - mea
n_absolute_error: 0.0964 - mse: 0.1432 - val_loss: 0.1645 - val_mean_absolute
_error: 0.0951 - val_mse: 0.1645

Epoch 21/100

1536/1536 [=====] - 9s 6ms/step - loss: 0.1555 - mea
n_absolute_error: 0.0962 - mse: 0.1555 - val_loss: 0.1126 - val_mean_absolute _error:
0.0914 - val_mse: 0.1126

Epoch 22/100

1536/1536 [=====] - 9s 6ms/step - loss: 0.1383 - mea
n_absolute_error: 0.0925 - mse: 0.1383 - val_loss: 0.1068 - val_mean_absolute
_error: 0.0903 - val_mse: 0.1068

Epoch 23/100

1536/1536 [=====] - 9s 6ms/step - loss: 0.1386 - mea
n_absolute_error: 0.0915 - mse: 0.1386 - val_loss: 0.1130 - val_mean_absolute
_error: 0.0867 - val_mse: 0.1130

Epoch 24/100

10s 6ms/step - loss: 0.1258 - me

an_absolute_error: 0.0843 - mse: 0.1258 - val_loss: 0.1059 - val_mean_absolut e_error:
0.0801 - val_mse: 0.1059

Epoch 39/100

1536/1536 [=====] - 10s 6ms/step - loss: 0.1296 - mean_absolute_error: 0.0836 - mse: 0.1296 - val_loss: 0.0964 - val_mean_absolute_error: 0.0787 - val_mse: 0.0964
Epoch 40/100

1536/1536 [=====] - 10s 6ms/step - loss: 0.1293 - mean_absolute_error: 0.0859 - mse: 0.1293 - val_loss: 0.0952 - val_mean_absolute_error: 0.0846 - val_mse: 0.0952
Epoch 41/100

1536/1536 [=====] - 10s 6ms/step - loss: 0.1258 - mean_absolute_error: 0.0849 - mse: 0.1258 - val_loss: 0.0980 - val_mean_absolute_error: 0.0783 - val_mse: 0.0980
Epoch 42/100

1536/1536 [=====] - 10s 6ms/step - loss: 0.1245 - mean_absolute_error: 0.0854 - mse: 0.1245 - val_loss: 0.1062 - val_mean_absolute_error: 0.0814 - val_mse: 0.1062
Epoch 43/100

1536/1536 [=====] - 11s 7ms/step - loss: 0.1250 - mean_absolute_error: 0.0842 - mse: 0.1250 - val_loss: 0.0992 - val_mean_absolute_error: 0.0829 - val_mse: 0.0992
Epoch 44/100

1536/1536 [=====] - 10s 6ms/step - loss: 0.1278 - mean_absolute_error: 0.0859 - mse: 0.1278 - val_loss: 0.0924 - val_mean_absolute_error: 0.0790 - val_mse: 0.0924
Epoch 45/100

1536/1536 [=====] - 10s 6ms/step - loss: 0.1317 - mean_absolute_error: 0.0858 - mse: 0.1317 - val_loss: 0.1146 - val_mean_absolute_error: 0.0858 - val_mse: 0.1146
Epoch 46/100

1536/1536 [=====] - 10s 7ms/step - loss: 0.1255 - mean_absolute_error: 0.0845 - mse: 0.1255 - val_loss: 0.0953 - val_mean_absolute_error: 0.0829 - val_mse: 0.0953
Epoch 47/100

1536/1536 [=====] - 10s 6ms/step - loss: 0.1261 - mean_absolute_error: 0.0848 - mse: 0.1261 - val_loss: 0.1804 - val_mean_absolute_error: 0.1231 - val_mse: 0.1804
Epoch 48/100

1536/1536 [=====] - 9s 6ms/step - loss: 0.1254 - mean_absolute_error: 0.0860 - mse: 0.1254 - val_loss: 0.1623 - val_mean_absolute_error: 0.0935 - val_mse: 0.1623
Epoch 49/100

1536/1536 [=====] - 9s 6ms/step - loss: 0.1321 - mean_absolute_error: 0.0869 - mse: 0.1321 - val_loss: 0.1032 - val_mean_absolute_error: 0.0839 - val_mse: 0.1032
Epoch 50/100

1536/1536 [=====] - 9s 6ms/step - loss: 0.1203 - mean_absolute_error: 0.0857 - mse: 0.1203 - val_loss: 0.1088 - val_mean_absolute_error: 0.0858 - val_mse: 0.1088
Epoch 51/100

1536/1536 [=====] - 9s 6ms/step - loss: 0.1210 - mean_absolute_error: 0.0886 - mse: 0.1210 - val_loss: 0.1336 - val_mean_absolute_error: 0.0850 - val_mse: 0.1336
Epoch 52/100

9s 6ms/step - loss: 0.1259 - mean_absolute_error: 0.0897 - mse: 0.1259 - val_loss: 0.0955 - val_mean_absolute_error: 0.0787 - val_mse: 0.0955
Epoch 53/100

1536/1536 [=====] - 10s 6ms/step - loss: 0.1302 - mean_absolute_error: 0.0970 - mse: 0.1302 - val_loss: 0.1055 - val_mean_absolute_error: 0.0961 - val_mse: 0.1055
Epoch 54/100

1536/1536 [=====] - 9s 6ms/step - loss: 0.1195 - mean_absolute_error: 0.0966 - mse: 0.1195 - val_loss: 0.0944 - val_mean_absolute_error: 0.0966 - val_mse: 0.0944

_error: 0.0832 - val_mse: 0.0944
Epoch 55/100
1536/1536 [=====] - 9s 6ms/step - loss: 0.1177 - mea
n_absolute_error: 0.0891 - mse: 0.1177 - val_loss: 0.1015 - val_mean_absolute
_error: 0.0858 - val_mse: 0.1015
Epoch 56/100
1536/1536 [=====] - 9s 6ms/step - loss: 0.1317 - mea
n_absolute_error: 0.0967 - mse: 0.1317 - val_loss: 0.1283 - val_mean_absolute
_error: 0.1371 - val_mse: 0.1283
Epoch 57/100
1536/1536 [=====] - 9s 6ms/step - loss: 0.1449 - mea
n_absolute_error: 0.1115 - mse: 0.1449 - val_loss: 0.1405 - val_mean_absolute
_error: 0.1065 - val_mse: 0.1405
Epoch 58/100
1536/1536 [=====] - 11s 7ms/step - loss: 0.1369 - me
an_absolute_error: 0.1109 - mse: 0.1369 - val_loss: 0.1195 - val_mean_absolut e_error:
0.0959 - val_mse: 0.1195
Epoch 59/100
1536/1536 [=====] - 10s 7ms/step - loss: 0.1273 - me
an_absolute_error: 0.1057 - mse: 0.1273 - val_loss: 0.1095 - val_mean_absolut e_error:
0.0978 - val_mse: 0.1095
Epoch 60/100
1536/1536 [=====] - 10s 7ms/step - loss: 0.1276 - me
an_absolute_error: 0.1021 - mse: 0.1276 - val_loss: 0.1277 - val_mean_absolut e_error:
0.1168 - val_mse: 0.1277
Epoch 61/100
1536/1536 [=====] - 10s 6ms/step - loss: 0.1318 - me
an_absolute_error: 0.1073 - mse: 0.1318 - val_loss: 0.1052 - val_mean_absolut e_error:
0.0938 - val_mse: 0.1052
Epoch 62/100
1536/1536 [=====] - 10s 7ms/step - loss: 0.1191 - me
an_absolute_error: 0.1005 - mse: 0.1191 - val_loss: 0.1119 - val_mean_absolut e_error:
0.0968 - val_mse: 0.1119
Epoch 63/100
1536/1536 [=====] - 11s 7ms/step - loss: 0.1200 - me
an_absolute_error: 0.0970 - mse: 0.1200 - val_loss: 0.1170 - val_mean_absolut e_error:
0.0922 - val_mse: 0.1170
Epoch 64/100
1536/1536 [=====] - 10s 7ms/step - loss: 0.1175 - me
an_absolute_error: 0.0955 - mse: 0.1175 - val_loss: 0.0991 - val_mean_absolut e_error:
0.0870 - val_mse: 0.0991
Epoch 65/100
1536/1536 [=====] - 10s 6ms/step - loss: 0.1115 - me
an_absolute_error: 0.0957 - mse: 0.1115 - val_loss: 0.2149 - val_mean_absolut e_error:
0.1159 - val_mse: 0.2149
Epoch 66/100
10s 6ms/step - loss: 0.0942 - me
an_absolute_error: 0.0844 - mse: 0.0942 - val_loss: 0.1141 - val_mean_absolut e_error:
0.0868 - val_mse: 0.1141
Epoch 95/100
1536/1536 [=====] - 9s 6ms/step - loss: 0.1021 - mea
n_absolute_error: 0.0875 - mse: 0.1021 - val_loss: 0.2857 - val_mean_absolute
_error: 0.0802 - val_mse: 0.2857
Epoch 96/100
1536/1536 [=====] - 10s 7ms/step - loss: 0.1217 - me
an_absolute_error: 0.0911 - mse: 0.1217 - val_loss: 0.0976 - val_mean_absolut e_error:
0.0888 - val_mse: 0.0976
Epoch 97/100
1536/1536 [=====] - 10s 7ms/step - loss: 0.0934 - me
an_absolute_error: 0.0831 - mse: 0.0934 - val_loss: 0.1032 - val_mean_absolut e_error:
0.0861 - val_mse: 0.1032
Epoch 98/100


```

1536/1536 [=====] - 11s 7ms/step - loss: 0.1045 - me
an_absolute_error: 0.0842 - mse: 0.1045 - val_loss: 0.0892 - val_mean_absolut e_error:
0.0787 - val_mse: 0.0892
Epoch 99/100
1536/1536 [=====] - 10s 6ms/step - loss: 0.1030 - me
an_absolute_error: 0.0856 - mse: 0.1030 - val_loss: 0.0952 - val_mean_absolut e_error:
0.0845 - val_mse: 0.0952
Epoch 100/100
1536/1536 [=====] - 10s 7ms/step - loss: 0.0980 - me
an_absolute_error: 0.0850 - mse: 0.0980 - val_loss: 0.1409 - val_mean_absolut e_error:
0.1013 - val_mse: 0.1409

```

APPENDIX C (Simulation Data)

```

import os import random from sys import path import sys
import platform from PyQt5 import QtCore, QtGui,
QtWidgets
from PyQt5.QtCore import (QCoreApplication, QPropertyAnimation, QDate, QDateTime, QMetaObject,
QObject, QPoint, QRect,
        QSize, QTime, QUrl, Qt, QEvent, QRunnable, pyqtSlot, QThreadPool, QTimer)
from PyQt5.QtGui import (QBrush, QColor, QConicalGradient, QCursor, QFont, QFontDatabase,
QIcon, QKeySequence, QLinearGradient, QPalette, QPainter, QPixmap, QRadialGradient) from
PyQt5.QtWidgets import *
import pandas as pd
from scipy.stats import zscore
## ==> SPLASH SCREEN
from ui_splash_screen import Ui_SplashScreen from
options import Ui_Dialog from prog_Dialogue import
Window_Prog import matplotlib.pyplot as plt # from
plot_cls import MainWindowCall
## ==> MAIN WINDOW from ui_main import
Ui_MainWindow import cglib from tensorflow.keras.models
import load_model from sklearn import metrics import
tensorflow as tf import numpy as np import time from
tensorflow import keras
# physical_devices = tf.config.list_physical_devices('GPU') #
tf.config.experimental.set_memory_growth(physical_devices[0], True)
cglib.enable(format = 'text')
## ==> GLOBALS
counter = 0

```

```
# C:\Users\HP\PycharmProjects\DeepLearningIshNew\models
```

```
StyleSheet = ""
#BlueProgressBar::chunk { background-color:
rgb(56, 58, 89); width: 10px; margin: 0.5px;
} class Worker(QRunnable): def
__init__(self, fn, *args, **kwargs):
    super(Worker, self).__init__()
    # Store constructor arguments (re-used for processing) self.fn
= fn self.args = args self.kwargs = kwargs

@pyqtSlot() def run(self):
    ""
    Initialise the runner function with passed args, kwargs.
    ""
    self.fn(*self.args, **self.kwargs) class OptionsDialog(QDialog):
def __init__(self): QDialog.__init__(self) self.ui = Ui_Dialog()
self.ui.setupUi(self)
self.ui.buttonBox.clicked.connect(self.get_option_value)
self.option_val = 1 def get_option_value(self): if
self.ui.radioButton.isChecked():
    self.option_val = 2 elif
self.ui.radioButton_2.isChecked():
    self.option_val = 1 elif
self.ui.radioButton_3.isChecked():
    self.option_val = 3
```

APPENDIX D (Application Data)

```
class MainWindow(QMainWindow): def
__init__(self):
```

```

QMainWindow.__init__(self)    self.ui = Ui_MainWindow()
self.ui.setupUi(self)
self.ui.pushButton.clicked.connect(self.browsefiles)
self.ui.actionOptions.triggered.connect(self.optionsStup)
self.ui.actionSave.triggered.connect(self.visualize)    self.threadpool =
QThreadPool()    self.dialoge = OptionsDialog()
self.predict_record = []    self.scorep = 0    self.x_dt = None
    # print("Multithreading with maximum %d threads" % self.threadpool.maxThreadCount())    #
self.fname = None
    # MAIN WINDOW LABEL
    # QtCore.QTimer.singleShot(1500, lambda: self.ui.label.setText("<strong>THANKS</strong> FOR
WATCHING"))
    # QtCore.QTimer.singleShot(1500, lambda: self.setStyleSheet("background-color: #222; color:
#FFF"))    def execute_this_fn(self,x_dt_values,model_path):
        self.predict_record.clear()    model =
load_model(model_path)
        # print("got here beg")    if(len(x_dt_values) == 2):
            self.prog.close()    for i in
range(round(len(x_dt_values[1]) / 300)):    if (i + 300 <
len(x_dt_values[1])):
                # self.ui.label.clear()    ranval =
random.randint(0,1)
                # print(ranval)    x_pred_test = x_dt_values[ranval][i:i + 300]
pred = model.predict(x_pred_test)    self.scorep =
np.sqrt(metrics.mean_squared_error(pred, x_pred_test))
self.predict_record.append(self.scorep)    self.dis_arr = x_pred_test[0][:7]
if(ranval == 1):
            item = QListWidgetItem("Attack : %s" + (" ".join([str(elem) for elem in
self.dis_arr])))    self.ui.listView.addItem(item)    else:
            item = QListWidgetItem("Normal : %s" + (" ".join([str(elem) for elem in
self.dis_arr])))    self.ui.listView.addItem(item)
self.ui.label.setText("{:.5f}".format((self.scorep)))
        print(self.scorep)
self.predict_record.append(self.scorep)    if
(self.scorep > 0.5):

```

```
self.ui.label_3.setText("Flagged Attack")           else:
self.ui.label_3.setText("Flagged Normal")
time.sleep(-time.time() % 1)           else:
self.prog.close()                               for i in
range(round(len(x_dt_values) / 300)):
if (i + 300 < len(x_dt_values)):
# self.ui.label.clear()
```

