UNIVERSITY OF KWAZULU-NATAL

The design and development of an AI based digital forensic protocol for first responders

Deepak Kumar

212561392

A thesis submitted in fulfilment of the requirements for the degree

of

Doctor of Philosophy

School of Management, IT & Governance

College of Law and Management Studies

Supervisor: Prof. Prabhakar Rontala Subramaniam

2024

DECLARATION

I ..... Deepak Kumar ..... declare that

i. The research reported in this dissertation/thesis, except where otherwise indicated, is my original research.

ii. This dissertation/thesis has not been submitted for any degree or examination at any other university.

iii. This dissertation/thesis does not contain other persons' data, pictures, graphs or other information, unless specifically acknowledged as being sourced from other persons.

iv. This dissertation/thesis does not contain other persons' writing, unless specifically acknowledged as being sourced from other researchers. Where other written sources have been quoted, then:

   a) their words have been re-written but the general information attributed to them has been referenced;
   b) where their exact words have been used, their writing has been placed inside quotation marks, and referenced.

v. Where I have reproduced a publication of which I am author, co-author or editor, I have indicated in detail which part of the publication was actually written by myself alone and have fully referenced such publications.

vi. This dissertation/thesis does not contain text, graphics or tables copied and pasted from the Internet, unless specifically acknowledged, and the source being detailed in the dissertation/thesis and in the References sections.

Signed: …… ..….. Date: 07/02/2024

ACKNOWLEDGEMENTS

I would like to express my deepest appreciation to my supervisor, Professor Prabhakar Rontala Subramaniam, for his guidance, having encouraged and helped me to complete this study. I would also like to acknowledge contributions from my previous supervisor, Professor Manoj S. Maharaj for his valuable guidance towards the completion of this study.

I'm deeply indebted to my father: Late Sh. Indraveer Singh and my mother: Shrimati Surendri Devi for their blessings and prayers throughout this journey. I sincerely believe that my late father's and my ancestors' blessings are with me always.

I would also like to extend my deepest gratitude to my wife Mrs. Pooja Kumar for her patience, prayers and support during this journey. A special thank goes out to my two lovely kids, my daughter Avantika and my son Vardaan for their support and prayers.

I also want thank my colleagues, family members and friends who supported and contributed in one way or another to this journey.

DEDICATION

I dedicate this work to-

• God Almighty (Parampita Paramatma Parmeshwar) for being with me all through.

• My parents, for their love and support they gave me.

• My life partner (my wife), for her support, prayers and encouragement during this journey.

• My kids for always believing in me.

Abstract

In today's society, access to computers and the internet has become indispensable, offering a myriad of opportunities such as online shopping, trading, banking, communication, and social media interaction. However, along with the increasing usage of the internet, there is a corresponding rise in cybercrimes, posing constant threats to organizations. Recent years have witnessed a significant surge in cyber incidents and breaches, exacerbated by emerging technologies like the Fourth Industrial Revolution (4IR) and Artificial Intelligence (AI), as well as the availability of tools such as Crimeware-as-a-Service (CaaS), anonymous technologies like Tor, and the utilization of the Darknet. In response to these challenges, cyber forensic experts and digital investigators must possess the necessary skills and expertise to effectively investigate cybercrimes, analyse electronic evidence found on digital devices, and present findings in a legally acceptable manner. To stay ahead of cybercriminals, digital forensic investigators and first responders must leverage AI and cutting-edge technologies of the 4IR era. This study addresses the evolving cybersecurity landscape by designing an AI-based digital forensic protocol tailored for first responders. Employing a design science research (DSR) methodology, the study develops a novel investigation protocol utilising AI prediction modelling. Additionally, it explores various AI models to create an efficient framework for integrating Machine Learning (ML) and predictive modelling in cybercrime data analysis of a cloud-based dataset. The design and development of Intelligent Digital Evidence Extraction Protocol or I-DEEP, a novel protocol provides a framework to make the process of cybercrime investigation more agile using triaging and quick decision making by predictive analysis. This is accomplished by development and implementation of AI and Machine Learning algorithms.

Keywords: Digital Forensics, Cybercrime, Artificial Intelligence, Predictive Modelling, Machine Learning

Table of Contents

List of Tables

List of Figures

List of Algorithms

List of Procedures

List of Python Code Snippets

List of acronyms and abbreviations

AI	Artificial Intelligence
ANN	Artificial Neural Network
CNN	Convolutional Neural Network
DF	Digital Forensics
DSR	Design Science Research
DSRM	Design Science Research Methodology
IR	Incident Response
KNN	K-Nearest Neighbors
LDA	Latent Dirichlet Allocation
LR	Linear Regression
ML	Machine Learning
SGD	Stochastic Gradient Descent
SVC	Support vector classification
SVM	Support-vector machine
URL	Uniform Resource Locator

CHAPTER 1: INTRODUCTION

1.1 Introduction

This chapter provides an overview to the digital forensics and constructs related to the field. It also highlights the problems and challenges faced by first responders and cyber forensics investigators in investigating cybercrime. Although detailed investigation into the problems pertaining to the field of digital forensics has been conducted in upcoming chapters, this chapter introduces the challenges faced by the first responders. It further emphasises the significance of this research towards providing solutions to the problem and establish the contextual argument for creating specific artifacts to provide solutions to the first responders in the field of digital forensics.

IT Forensics, Cyber Forensics, and Computer Forensics fall within the broader category of Digital Forensics, encompassing the investigation of digital evidence present in computers, portable devices, digital storage media, and network devices. The primary objective of cyber forensics is to meticulously analyse digital media in a forensically sound manner, aiming to identify, preserve, recover, analyse, and present pertinent facts hidden within data as evidence. Cyber forensics is frequently employed in a diverse array of computer-related crime investigations and e-discovery processes. Evidence from computer forensics is treated with the same level of scrutiny as other forms of physical evidence and is increasingly gaining acceptance in court proceedings.

The first event of Digital Forensic Research Workshop, also abbreviated as DFRWS, was organised in the year 2001 in Utica, New York (DFRWS, 2001), which addressed these issues of standardisation of digital forensic process. There has been major contributions to the field of Digital Forensics (DF) by various researchers and practitioners, but still universally accepted models and framework do not exist (Garfinkel, 2010; Adams et al., 2013; Bauchner, 2014).

This situation necessitates the requirement of effective interventions to prevent, contain and investigate cyber-crime. There are investigation protocols to track and trace the culprits by conducting investigation, collect electronic evidence of the crime and prosecute the cyber criminals. Due to the reasons mentioned afore, the efficacy of the tools and investigation methods used may be undermined through various confounding factors (Collie, 2018; Britz, 2009).

There is a pressing need for addressing the compliance issues that pervade the investigative process, challenges posed by vastly diverse and dynamic infrastructural implementations in computing environments (Casey et al., 2009). It has to be ensured that the methodology used in DF investigations is legally compliant to the court of law directives (Institute of Justice, 2008). DF investigative methods have to be accepted among DF investigator community and rigorous enough to stand in court of law (Kohn et al., 2013).

Therefore, the study is intended to investigate the problem domains in the field of digital forensics and propose solutions to first responders or cybercrime investigators by designing a novel investigation protocol that uses agile approach, triaging and artificial intelligence (AI) to bring greater efficacy in the process of digital investigation. The proposed digital investigation protocol intends to aid the investigators by suggesting most suitable investigative approach that also integrates all technical and methodological aspects of the investigation process.

The study further aims to develop an AI based intelligent framework and a working prototype using dataset consisting of cybercrime data extracted from 'cloud based' source. This research is aimed at designing a new protocol in order to provide assistance to the cybercrime investigators and professionals by using cutting edge technological interventions like Artificial intelligence (AI) and assisting them with cybercrime investigations, evidence extraction process, documenting the cybercrime details, effective decision making and performing predictive analysis in a methodological manner.

1.2 Background of the Study

Computer and Internet access has become indispensable part of today's life. The Internet provides unprecedented opportunities available in the domains like online-shopping, online trading, internet banking among other advantages like communication, social media interactions, social activism etc. to name a few (Moore, 2010). Along the growing trends of Internet usage, also comes the increased instances of cybercrimes that are a constant threat to organisations. Various IT companies and information security firms publishes their reports that show an increase in IT security breaches and growing cybercrime instances. The occurrence of cyber incidents or breaches has shown a significant increase in year 2023, with 91% of organizations reporting at least one incident, as opposed to 88% in the 2021 survey. Cyber concerns for organizations involve

a myriad of actors, sources, tools, and techniques. Organizations with high maturity levels exhibit more concern about cyber criminals, terrorists, phishing, malware, and ransomware attacks. Conversely, low- and medium-maturity companies express greater concern about denial-of-service attacks. Notably, organizations with low maturity levels report experiencing more substantial cybersecurity events (Deloitte, 2023).

According to Symantec Report (2018), there is a 10% increase in overall targeted attacks in 2017, and around 13% overall increase in reported vulnerabilities. More than 80% of these attacks were motivated by intelligence gathering to conduct various cyber-crimes. There is also 54% increase in new mobile malware variants in 2017 as compared to 2016. Grayware has increase 20% in 2017, and 63% of them leak device phone numbers that pose a major security risk.

A 55% increase in email spam rate, a 46% increase in ransomware variants, 600% increase in attacks against Internet of Things (IoT) devices in 2017, reports that cyber-crime is continuously increasing. In 2017, reports of ransomwares like WannaCry and Petya/NotPetya were prevalent. Another major thread that has emerged in 2017 is an increase of 8,500% in coin-mining attacks. Coin-mining attacks allow cyber-criminals to earn crypto-currencies by secretly infecting victim's computers with file or script based coin-miners (programs), while staying under the radar compared to ransomwares which are apparent (Symantec, 2018).

1.2.1 Cybercrime

To begin with, it's essential to clearly define "cyber-crime" and differentiate it from 'traditional crime'. While some computer-related crimes like fraud, forgery, theft, mischief, and defamation overlap with conventional criminal actions covered under the legal jurisdictions of many countries, the misuse of computers has also given birth to a new generation of crimes addressed by the Information Technology Acts formulated by many nations. It's not entirely accurate to describe cyber-crimes as "actions that are illegal under the Information Technology Act" since many cyber-crimes are also punishable under the Indian Penal Code, such as email cyber defamation, spoofing, and sending threatening emails. A brief yet comprehensive definition of cybercrime can be given as "unlawful activities that involve the use of a computer as a tool, target, or both" (Vadza, 2011; Navneet, 2018).

The proliferation of new technologies has led to a surge in cybercrime, presenting a formidable challenge in recent years. Cybercriminals persistently adjust and enhance their tactics to capitalize on emerging weaknesses, amplifying the threat to individuals and organizations worldwide. Contemporary cybercrime diverges from its earlier forms, incorporating sophisticated methods like distributed denial-of-service (DDoS) attacks, phishing schemes, malware campaigns, ransomware assaults, and various other malicious activities. These actions have the potential to cause significant harm and leave a profound imprint on both organizations and communities, as highlighted by Interpol (2023) in its latest published report.

While cybercrime doesn't always require cutting-edge technology, it often exploits known vulnerabilities left unattended due to carelessness. The rise of cybercrime is a growing concern as computers, smartphones, and the internet become increasingly accessible to both individuals and criminals (Symantec, 2018). Computing devices can serve as targets or tools for cybercrime. The anonymity afforded by the internet enables users to carry out criminal activities from a safe distance, away from the traditional crime scene. Common examples of personal and organizational cybercrime include hacking, cracking, intellectual property disputes, copyright and patent infringements, child pornography, and privacy violations. International cybercrime encompasses espionage, money laundering, drug trafficking, and financial crimes crossing borders. Recent years have seen increased attention on international cybercrimes such as cyber terrorism and cyber warfare

1.2.2 Cyber-Terrorism

Cyber-terrorism has emerged as a growing concern in the digital era, presenting various hacking and computer disruption scenarios that could play a significant role. In the global arena of evidence and network rivalry, cyber-terrorism has become a potent force, although there remains much confusion about its precise nature. Veerasamy (2009) posits that media portrayal has often exaggerated the potential for cyber-terrorism attacks to cause extensive destruction, but it still remains a huge threat for national security. Hybrid conflicts have supplanted traditional ones, ushering in new threats that manifest in cyberspace, now recognized as a virtual battlefield. Cyber threats, spanning cybercrimes, cyber-terrorism, and cyberwarfare, loom as significant concerns for Western governments, notably the United States and the North Atlantic Treaty Organization (NATO). The international community is increasingly regarding cyberattacks as a form of

terrorism, subjecting them to analogous measures. However, the term "terrorism" remains ambiguous and legally undefined, leading to a lack of consensus on the definition of its derivative term, "cyber-terrorism" (Marsili, 2019). Traditional terrorism has extended its reach into cyberspace, introducing new challenges to the technologically driven modern world. Cyber-terrorism instils a recent fear of how terrorists may seek to target the vulnerable and sow chaos. Additionally, cyber terrorists may target a country's critical infrastructure, including electricity and water supply systems, as well as nuclear facilities (Moore, 2014).

The use of cyber-terrorism includes the recruitment and radicalization of individuals, offering training, providing instructions and financing, and directing specific acts of terror. Critical infrastructure systems such as transportation, government services, energy, and financial institutions are prime targets due to the potential for devastating consequences if compromised.

In efforts to combat Al-Qaeda, US authorities uncovered evidence of the group's involvement in cyber operations that allowed them to gather information on vulnerabilities in key infrastructure, such as dams and bridges in the US. When influence of Al-Qaeda declined and the ISIS emerged, there has been a resurrection of cyber-terrorism. These organisations have effectively utilized the social-media to recruit new members and disseminate propaganda warfare against target nations, while also radicalising their followers to carry out attacks in their respective countries (Graham-Harrison, 2015).

In January 2015, the group carried out a significant communal attack by hacking into the YouTube and Twitter accounts of the US Central Military Command. The pages were defaced with the declaration "I love ISIS," and US employees' pictures and secret military files were sent out to show the whole world that the group has been successful in infiltrating US military organisations. Although the attack was a cyberattack called "defacement cataloguing", it established that the groups are growing in cyber capabilities and blatant tactics. As these organisations e.g. ISIS and Al-Qaeda, becomes more proficient in the cyber sphere, they are quite likely to use this strategy

to launch significant cyber-attacks, as stated by Graham-Harrison in 2015 (Graham-Harrison, 2015).

1.2.3 Cyberwarfare

Despite the lack of a widely adopted definition, it is evident that the terms "cyber war" and "cyber warfare" lack clear differentiation (Robinson et al., 2015). Cyberwarfare is a facet of information warfare characterized by politically motivated hacking aimed at sabotaging defence systems or conducting cyber-espionage. It often involves attacks on defence systems with the objective of clandestinely accessing classified information held by rival nations, governments, or military establishments, thereby seeking military, political, or economic advantages over adversaries. The current society has developed both direct and indirect dependencies on Information Technology, relying heavily on continuous connectivity, access, and linkages (Grobler & van Vuuren, 2012). A new dimension that has been added to cyberwarfare is social media exploitation by state sponsored actors to influence political establishment and creating ‘regime change’ or influencing the election process. This usually involves deploying huge workforce to create fake accounts, and peddling propaganda or disinformation by social media platforms like twitter (X) and/or Facebook.

The widespread occurrence of cyber warfare is a pervasive concern that creates anxiety among corporations, governments, and individuals alike. This contemporary form of warfare is no longer confined to developed nations only but also developing economies. It has attracted an array of actors, including insurgent groups like Hamas and Hezbollah, and hacking groups such as LulzSec, Anonymous, and others. Recent research predicts that the global market for cyberwarfare and cybercrime will be worth more than $15.9 billion by 2018 that will consist of consultation services on - Cyberwarfare, protection services, product development etc. (Shakarian et al., 2013).

1.2.4 New Trends in Cyber Attacks and Cybercrime

The cybersecurity landscape is in constant flux, particularly in an era of rapidly advancing technology. Safeguarding sensitive data, crucial infrastructure, and personal information is increasingly challenging as our world becomes more interconnected and reliant on digital technologies. This section delves into some significant security trends that are poised to influence the state of cybersecurity in 2024 and beyond.

1.2.4.1 The Rise of Automotive Hacking

Modern day vehicles are equipped with sophisticated software, facilitating seamless connectivity for drivers through features such as airbags, cruise control, door locks, and advanced driver aid systems. These vehicles establish connections via Bluetooth and Wi-Fi, exposing them to potential security issues and hacking threats (Nobles et. al., 2023). Hacking a car involves exploiting security vulnerabilities in the software, allowing unauthorized individuals to gain access to the vehicle's systems, pilfer confidential data, or pose significant threats to human life (Nair, 2023). As the prevalence of automated vehicles continues to grow in 2023 and coming years, there is an anticipation of increased attempts to gain control of these vehicles or eavesdrop on conversations within them. Consequently, it becomes paramount that holistic automotive cybersecurity solutions address both software integrity and manufacturing/supply chain security aspects (Nair, 2023).

The evolution of cybercrime into highly severe threats such as cyberwarfare and cyberterrorism underscore the growing sophistication of cybercriminals. As these threats become more complex, cyber-security and cybercrime investigation agencies must enhance their preparedness to combat cybercriminals and law offenders effectively. This necessitates continuous improvement in strategies, technologies, and collaborative efforts to stay ahead of evolving cyber threats (Reich et al., 2010). The Computer Emergency Response Teams also known as CERTs need to prepare for national emergencies that can be caused by the wider vectors of cybercrime like cyberwarfare and cyber-terrorism (Schultz, 2016).

1.2.4.2 The Potential of Artificial Intelligence (AI) Exploitation in Cyber-Attacks

Cybercriminals, well-versed in latest technological developments, are leveraging artificial intelligence to execute more intricate attacks, particularly with the increasing prevalence of Chat GPT on the internet. AI-driven attacks pose a greater challenge in terms of identification and defence, given their ability to swiftly adapt to evolving settings (Al-Khawaja & Sadkhan, 2021; Alawida et al., 2024).

AI-powered attacks manifest in various forms, including phishing emails, ransomware, and scams. For instance, AI can be employed to craft highly convincing phishing emails that appear to originate from trustworthy sources, making it easier for attackers to deceive individuals into opening malicious links or installing malware. Moreover, artificial intelligence is being harnessed

to develop sophisticated viruses and attacks capable of bypassing the latest data protection measures (Gupta et al., 2023). Organizations need to exercise caution and implement state-of-the-art cybersecurity technologies and strategies capable of detecting and mitigating these risks in real-time. This demands for the development and deployment of AI-driven cybersecurity tools capable of recognizing and thwarting attacks before they inflict damage so the first responders can have an edge over them and they can be beaten in this game (Al-Khawaja & Sadkhan, 2021).

1.2.4.3 Use of Targeted Ransomware

Ransomware constitutes a type of virus that specifically targets computer systems, encrypting files and rendering them inaccessible to the user. Subsequently, the attacker demands a ransom payment in exchange for providing the decryption key (Teichmann et al., 2023).

Ransomware attacks have become increasingly prevalent due to their lucrative nature for attackers. Compared to the expenses associated with dealing with the aftermath of a data breach or restoring systems from backups, paying the ransom is often a considerably less costly option. The consequences of a successful ransomware attack can be severe, encompassing brand damage, business losses, financial costs tied to ransom payment and system restoration, and potential misuse of sensitive data for malicious purposes (Jacob, 2023).

1.2.4.4 Advent Internet of Things (IoT) with 5G Network

The advent and expansion of 5G networks, coupled with the Internet of Things (IoT), are ushering in a new era of connectivity. The estimated 16.7 billion connected IoT devices in 2023 are expected to more than double, reaching 25.4 billion by 2025. While 5G offers distinct advantages, such as improved speed, bandwidth, and capacity compared to earlier cellular network generations, these enhancements also expand the attack surface of 5G equipment (Dayal et al., 2023).

Conventional security frameworks designed for 4G/LTE cellular networks are no longer seamlessly applicable to 5G IoT services. Security experts have underscored threats to the 5G-IoT ecosystem, including an elevated risk of proximity service (ProSe) intrusions and distributed denial of service (DDoS) attacks. The anticipated surge in both bandwidth and the number of connected devices creates heightened security vulnerabilities, making them susceptible to unauthorized access (Militano et al., 2015).

1.2.4.5 Crimeware as a Service

Crimeware-as-a-Service (CaaS) also referred as Cybercrime-as-a-Service is a term used to describe a criminal business model where cybercriminals offer various malicious services and tools to other individuals or groups, typically for financial gain. In this model, cybercriminals act as service providers, offering illicit products or services to facilitate cybercrime activities. Some common offerings of Crimeware-as-a-Service include Malware-as-a-Service (MaaS), Phishing-as-a-Service (PhaaS), money laundering services etc. (Interpol, 2023).

The popularity of Crimeware-as-a-Service is on the rise in Africa and worldwide, primarily attributed to its user-friendly interface, affordability, and the absence of repercussions stemming from inadequate legal frameworks concerning cybercrime enforcement. This platform offers perpetrators a convenient avenue to execute financially driven assaults on susceptible systems and businesses, requiring minimal exertion or technical expertise (Geldenhuys, 2023). Crimeware-as-a-Service has democratized cybercrime by lowering the barrier to entry, allowing even individuals with limited technical skills to engage in illegal activities. It poses a significant challenge to law enforcement agencies and cybersecurity professionals due to its widespread availability and ease of access (Singh & Rahman, 2023; Sood & Enbody, 2013).

This scenario calls for the cybercrime to be contained and investigated and law enforcement has to hire investigators and cyber-forensic experts who are equipped with necessary skillset and expertise to efficiently investigate cybercrime, analyse electronic evidence found on digital devices and present that in the court of law or an enquiry board so that it stands on the legally acceptable benchmarks. The forensic investigation and evidence collection should be fast, maintain data integrity and should withstand all legal jurisprudence.

1.2.5 Digital Forensics

Digital forensics, also referred to as “cyber forensics”, comprises a collection of procedures aimed at gathering, safeguarding, identifying, and documenting digital evidence which can be presented for legal proceedings. All kind of digital devices and media that has data storage, including computer-systems, servers, network devices, portable devices, and mobile phones, represent potential sources of evidence, and the primary objective is to unearth information that can be used in legal proceedings. Forensic teams ought to be equipped with the most effective techniques and

tools to deal with challenging digital-related situations. By leveraging digital forensics, forensic teams can pinpoint, preserve, and scrutinize data from various digital devices, which can subsequently serve as evidence in investigations.

1.2.6 Digital Forensics: A Brief History and Milestones

The first person to use a scientific methodology in a criminal investigation was Hans Gross (1847–1915). In 1932, FBI begins to provide forensics services to legal authorities and other field agents in USA. The “Florida Computer Crime Act” was formulated in 1978 that officially registered the first known instance of cyber-crime in Florida. It was Francis Galton, who carried out one of the first known examination of finger-prints. Academic literature first used the phrase "computer forensics" in 1992. In 1995, IOCE (International Organization on Computer Evidence) was founded. The Federal Bureau of Investigation established its first computer forensic lab at regional level in 2000. In 2002, the organisation named “Scientific Working Group on Digital Evidence” or SWGDE released its first book on digital forensics, which was titled as "Best practises for Computer Forensics". Simson Garfinkel (2010) outlined the challenges that digital investigators face in 2010.

1.2.7 Digital Forensics: Main Objectives

Digital forensics can help the investigating agency present the computer evidence and related materials by recovering, analysing, and preserving them in a legally acceptable form in a court of law. Main offender's identification and speculating about the crime's motivation is the main goal of digital forensics. Creating policies that assist digital forensic examiners to make sure the digital evidence that is collected, is untampered at a site of a possible crime is necessary.

The process of data extraction involves the collection and duplication of data, as well as the recovery of any deleted data such as files and partitions that may have been removed by perpetrators to conceal evidence (Johnson et al, 2022). This process helps to verify claims made and assists digital forensics investigators in identifying evidence and estimating the extent of the damage caused by the harmful behaviour to the victim. Additionally, a computer forensic report is generated to provide a comprehensive analysis of the investigation, and the chain of custody is maintained so that the integrity of the evidence is preserved.

1.2.8 Digital Forensics Process

The Digital Forensics Research Workshop (DFRWS) Framework outlines a set of universally accepted steps in the digital forensic process. These steps include the identification, preservation, collection, examination, and analysis of evidence, as well as the presentation of reports and evidence. DFRWS is worth mentioning as it is the de-facto standard that is universally accepted by the court of law, whole industry and digital forensic experts. All other frameworks derive and extend this framework into their implementations to a major extent.

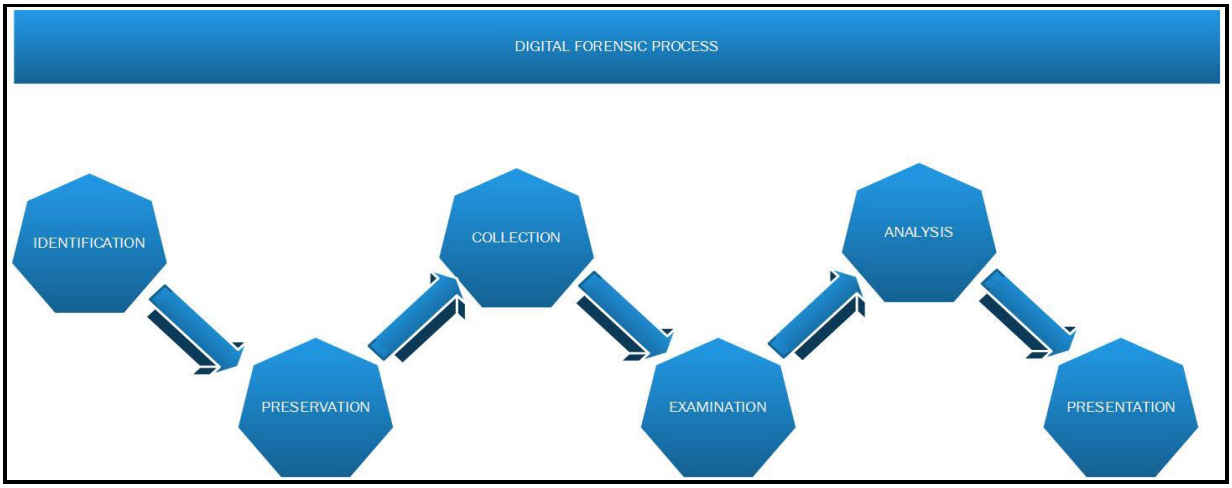


Figure 1.1: Digital Forensics Research Workshop Framework (DFRWS, 2001)

1.2.8.1 Identification of Evidence

The initial phase of the digital forensic investigation process involves answering fundamental questions such as whether evidence exists, where it can be found, and in what form it exists. This phase is based on Locard's exchange principle, which asserts that when two objects interact, they leave traces on each other. Similarly, in the digital world, when two systems interact, they exchange data and create a data trail. To illustrate this concept, consider the example of an individual browsing a website. The web-server and web-browser on the individual's machine exchange data, and the web-server logs typically contain the individual's IP address and other details such as date and time stamps. The website may also store cookies on the individual's

machine. Most computer systems have storage memory that can be used to store data (DFRWS, 2001; Johansen, 2017).

1.2.8.2 Preservation of Evidence

Once evidence is identified, it is imperative to preserve it or prevent it from being modified or deleted. In order to achieve this, the computer system have to be isolated from the network. Also, various controls whether physical (perimeter) or logical (network access control) are implemented. Users are not allowed to access the system as they can tamper with evidence. Preservation of evidence involves data segregation, data protection that is achieved in this phase. Some of technologies like system imaging or snapshotting can be utilised. It also involves access block devices to prevent tampering with digital evidence contained within the device (DFRWS, 2001; Johansen, 2017).

1.2.8.3 Collection of evidence

This involves acquiring of digital evidence by the cybercrime investigators or examiners. While digital evidence is examined, it is vital for examiners to be aware of the fact that some evidence is volatile in nature and can be lost easily. If a system is powered down or disconnected, digital evidence contained in RAM may be lost. If it is a network device or equipment, then it may loose data of active connection or data of that device. Data stored as ARP (Address Resolution Protocol) cache also constitute volatile data.

IETF (Internet Engineering Task Force) specifies certain guidelines – "Guidelines for Evidence Collection and archiving (RFC 3227)" further describing volatile digital evidence such as – registers and cache; ARP cache, routing table, RAM data; temporary system files; disk data; remote logged data; physical and network configurations and topologies; archived data. Therefore, it is very critical for investigators to be mindful of volatility of data collected.

The subsequent steps involve handling evidence, ensuring its security, documenting the collection process, and maintaining the chain of custody. The documentation process includes recording all visible data, which can assist in examining and reconstructing the crime scene. Precise documentation is crucial and involves taking photographs, creating sketches and maps, and accurately recording all aspects of the crime scene (Carrier & Spafford, 2004; Johansen, 2017).

1.2.8.4 Examination of Evidence

This phase deals with specific Digital Forensic (DF) techniques and tools utilised to extract data from the evidence that has been seized. This is also known as e-discovery, where relevant information is extracted from the device seized or image obtained during collection of data. If it is an active network, then SSH (Secure Shell) packets from the network are captured. During examination, proper data integrity and preservation has to be maintained thoroughly. If due diligence is not taken into care by examiner, then evidence can become contaminated and thus become unusable (Nortjé & Myburgh, 2019).

1.2.8.5 Analysis Phase

Once examination phase has been completed and relevant data has been obtained, it needs to be analysed. This phase mainly consists of putting together all collected evidence and analysing it thoroughly to uncover facts and judging or deriving certain conclusions. An example can be given as a host that is compromised, is having an open connection with an external host with a specific IP address. This IP address can be correlated with a network packet capture data collected from the host machine. An analyst may use this to identify any attacks vectors or any communication channels between the two hosts (Carrier & Spafford, 2004; Johansen, 2017). This should be an iterative process to achieve desired goals of crime investigations.

1.2.8.6 Presentation of Reports and Outcomes Phase

In digital forensics, it is crucial to present unbiased, clear, and concise facts in the report. The report should provide sufficient details to explain the underlying cause of the issue, and if required to testify in court, it must meet the legal standards defined by the legal system. In case the examiner possesses the requisite expertise and experience, they can act as an expert witness, which mandates the delivery of reports in the legal document format. In the final step, the process involves summarizing and interpreting the findings using appropriate abbreviations and plain language. All technical terms must include relevant details to ensure clarity (Bulbul et al., 2013), (Johansen, 2017).

1.2.9 Types of Digital Forensics

Digital forensics can be classified based on the device or environment on which it is being used. Disk-forensics pertains to the process of data extraction from different sizes of storage disks e.g. hard disks by searching for all kinds of active, modified, or deleted files. Network forensics involves monitoring and investigating activity on computer networks to gather critical data and evidence that is admissible in legal proceedings.

Wireless-forensics is a sub-branch of network-forensics that has main focus of providing the tools necessary to gather data and do a data-analysis from network traffic of a wireless network. Database-forensics involves investigating and analysing databases and their associated metadata as a subset of digital forensics. Malware forensics is dedicated to identifying harmful code and researching its payload, which includes viruses, worms, and other threats (Yaacoub et al., 2021).

Email-forensics concerns itself with email recovery and analysis, including the examination of any deleted data e.g. emails, contacts and calendars. Memory-forensics entails collection of raw data from volatile memory such as RAM, cache, and system registers, followed by carving the data from raw dump. Mobile-phone forensics primarily focuses on analysing mobile devices to retrieve data contained on phones, call history, SIM contacts, incoming and outgoing SMS and MMS, audio-files, video-files, and other forms of data (Johansen, 2017).

1.3 Research Problem

Due to rapid expansion in internet use-base, anti-social or criminal elements are inadvertently increasing, that posits potential threats to evolving internet community. The internet users become easy target to the cyber criminals due to their ignorance or lack of adopting security interventions (Vacca, 2010). Cybercrime has grown exponentially and digital computing devices are either the instruments of crime or arenas where cybercrime manifests. There is a persistent need for interventions so that perpetrators of cybercrime are brought to justice in court of law (Schultz, 2016). Cyber criminals are motivated by the fact that they are not confined to geographical boundaries and therefore are away from jurisdiction of laws. They may also exploit anonymous technologies like tor network and browser to hide their identities (torProject, 2023).

Due to aforementioned reasons, cyber-investigators/first responders are finding it difficult to conduct investigations timeously. There is a pressing need for addressing the compliance issues that pervade the investigative process, challenges posed by vastly diverse and dynamic infrastructural implementations in computing environments (Casey et al., 2009). The methodology used in DF investigations has to be legally compliant to the court of law directives (Institute of Justice, 2008). DF investigative methods have to be accepted among DF investigator community and rigorous enough to stand in court of law (Kohn et al., 2013). Therefore, the question arises that how effectively do first responders and cybercrime investigators can carry out their investigations under various challenges posed by changing technological paradigms and still adhere to prescribed methodologies as well as meet the legal benchmarks. Therefore, the main objective of this research is to establish how cybercrime investigators and first responders can be more effective in decision making while reacting to cyber-security breaches with the developed novel protocol using artificial intelligence (AI) based framework and the developed prototype based on this protocol.

1.4 Main Research Question

How effectively can cybercrime investigators and first responders react to the cybersecurity breaches with the proposed new protocol with AI based framework and newly created prototype?

1.5 Research Sub-questions

1. What are the challenges that cybercrime investigations face while conducting cybercrime investigations?

2. What digital forensic frameworks and models do cybercrime investigators and first responders adhere to in cybercrime investigations?

3. What digital forensic standards and protocols do cybercrime investigators follow internationally and in South Africa in cybercrime investigations and what are their drawbacks?

4. What digital forensic tools do cybercrime investigators use in cybercrime investigations and how can we make better use of them?

5. How effective are the present-day frameworks/models, standards/protocols, and tools available to digital forensic experts in achieving desired outcomes and what improvements can be made to increase their efficacy?

6. How does an AI based intelligent system can help investigators to be more efficient and achieve desired outcomes?

In an attempt to answer the following research questions, the research intends to design a novel digital investigation protocol using AI and machine learning and a new prototype system based on this protocol that can effectively help first-respondents/investigators to follow investigation process in a methodological manner. It is endeavoured to establish that the new protocol model and AI based system will assist first responders/investigators in triaging, doing predictive analysis and improved decision making.

1.6 Research Objectives

a. To determine the efficacy of digital forensic frameworks, models and tools used by cybercrime investigators/first respondents in cybercrime investigations.
b. To identify challenges faced by cybercrime investigators/first respondents in taking quick decisions while conducting cyber-investigations.
c. To design a new investigation protocol and framework based on AI and a working prototype that can assist first respondents in cybercrime investigations.
d. To evaluate the new investigation protocol and framework based on AI and system prototype, in order to establish whether the new artefact has achieved its goals of quick decision-making, increased efficiency, and empowering cybercrime investigators/first respondents to combat cybercrime investigation challenges through triaging and prediction analysis.

1.7 Research Rationale

Due to various reasons, like rapid internet growth and the exponential growth in data that is stored in digital devices, cyber-investigators/first responders are finding it difficult to conduct investigations timeously. There are compliance issues that result into delays in the investigative process, challenges posed by vastly diverse and dynamic infrastructural implementations in

computing environments (Casey et al., 2009). The methodology used in DF investigations are overly complex and non-practical to implement (Du et al., 2017). Therefore, the question arises that how effectively do first responders and cybercrime investigators can carry out their investigations under various challenges posed by changing technological paradigms and still adhere to prescribed methodologies as well as meet the legal benchmarks. This research hypothesize that the stated problem can be answered by incorporating AI into the investigation protocols and developing intelligent framework-based tools, can assist in overcoming the challenges faced by first responders and cybercrime investigators.

1.8 Significance of Research

This research contributes to the body of knowledge in the field of Digital Forensics by exploring the challenges that exist for the first responders in conducting digital forensic investigations and the gaps present in existing tools, protocols, models and frameworks. The research intends to fill the gaps and provides solutions by designing a novel AI based protocol for first responders which is more agile and practically implementable. This research further develops an AI based intelligent framework for predictive analysis of cybercrime data collected during investigations. The artefacts developed are demonstrated and their efficacy is evaluated by developing a prototype based on novel protocol and intelligent framework using Design Science Research (DSR) Methodology. While maintaining the accepted standards in the legal and professional field of digital forensics, the research provides compliance to established frameworks but also extends or offer improvisations to some of the most adopted existing models and frameworks to provide theoretical background to novel artefacts developed in the research.

1.9 Structure of Thesis

This thesis is structured into 10 chapters which incorporates from introduction of digital forensics, identification of the research problem, review of existing digital forensics tools and frameworks in order to find research gaps and then attempts to fill them by proposing improvements through extended frameworks and models. These extended models also form the theoretical and conceptual underpinnings of design and development of a novel AI based protocol and prototype. The evaluation of the protocol/prototype is done and finally the conclusion chapter is provided, which also contains research justification, limitations, recommendations and future scope. The chapters

are well structured to provide the progression of the research work and also align to the research methodology (Design Science Research or DSR) used as the guiding process for achieving the planned objectives of the study. The chapters and their objectives are briefly explained in forthcoming section to establish to flow and structure of the thesis.

CHAPTER 1: INTRODUCTION

This chapter offers an overview of digital forensics and its related constructs, along with spotlighting the difficulties encountered by first responders and cyber forensics investigators when addressing cybercrime. While subsequent chapters delve into specific problems within the digital forensics field, this chapter specifically addresses the challenges faced by first responders. It highlights the importance of this research in offering solutions and lays the groundwork for creating tailored artifacts to assist first responders in the realm of digital forensics. The chapter entails relevant sections like Introduction, Background of study, Research problem, Research questions, Research objectives, Research rationale, Significance of the research, Thesis structure and chapter summary.

CHAPTER 2: LITERATURE REVIEW

This chapter commences by elucidating fundamental concepts within the digital forensic domain. It extensively discusses the plethora of digital forensic investigation models and frameworks, each adhering to distinct methodologies tailored for different types of cybercrime investigations. Despite the abundance of digital forensic (DF) investigation tools catering to diverse investigation requirements and technologies such as operating systems, file systems, and mobile devices, there remains a lack of standardization. This deficiency hampers the representation of the most accepted DF processes and frameworks, consequently affecting the reliability and efficacy of forensic tools.

First responders and cybercrime investigators encounter key challenges of technological, methodological, and legal natures. Various new and established technologies pose obstacles in the digital forensic investigation process. To overcome these hurdles, it is imperative to introduce better frameworks that are more implementable and leverage cutting-edge technologies such as Artificial Intelligence (AI) and Machine Learning (ML). AI techniques can facilitate prediction analysis and augment decision-making capabilities for cybercrime investigators, thereby

enhancing the effectiveness of digital forensics investigations. Therefore, this chapter covers these aspects under sections like Introduction, Evolution of Digital Forensics, Overview of digital forensic process models and frameworks, Models and frameworks used for digital forensics, International and South African standards used in digital forensics, Limitations of the standards, digital forensic tools and their SWOT analysis. Apart from providing comprehensive analysis of digital forensic tools, frameworks and models, this chapter also proposes improvements in the models and frameworks in the form of extended models. These extended models form the theoretical underpinnings for this research and provides conceptual platform to design and develop novel protocol and prototype to fill the gaps identified in this chapter during reviewing the existing artefacts and therefore contributing towards the body of knowledge. This chapter concludes and summarises the review findings and provide pathway for solving the problems identified in this study.

CHAPTER 3: RESEARCH METHODOLOGY

This chapter introduces research methodologies and identifies the Design Science Research (DSR) methodology as suitable for designing and developing a novel protocol and prototype for this research. It discusses the research design, research outcomes planned, research process model developed for this research under relevant sections. It finally provides a summary of the chapter contents. It consists of sections like Introduction, Design Science Research (DSR) overview, Research design developed for this research, Research outcome planned, Research process model designed for this research and summary.

CHAPTER 4: INCIDENT RESPONSE - INTEGRATION WITH AGILE APPROACH

This chapter delves into Incident Response (IR) and its various aspects within the realm of digital forensic investigations, emphasizing a more agile approach for practical implementation. It highlights the key actions that organizations or their IR teams must undertake, along with the critical methodology to be followed to mitigate the damage resulting from cyber-attacks or security breaches. Central to this discussion is the concept of "Triaging," which plays a crucial role in reducing response time and facilitating swift decision-making. These objectives guide the proceedings of the chapter, emphasizing the necessity for organizations to respond systematically and efficiently to security incidents, thus minimizing the potential damage caused by cyber-attacks

and enabling effective recovery from their aftermath. The chapter consists of sections like Introduction, Implementing IR process using 'agile approach', Preserving digital evidence, Digital forensic lab requirements for conducting IR, Software tools used for IR, Discussion and suggestions for improvement of IR process and Summary.

CHAPTER 5: DESIGN OF AI BASED DIGITAL FORENSIC PROTOCOL (I-DEEP)

This chapter entails designing a novel, AI based Digital Forensic Protocol named as 'Intelligent-Digital Forensic Evidence Extraction Protocol' (I-DEEP). It also highlights critiques the background studies and models/frameworks in order to contextualise the need for this research. It has been emphasised in chapter-2 that the majority of crime investigations frameworks and protocols are not appropriate for use in scenarios where quick decision making is required. Present investigative techniques, especially those used by law enforcement, due the complexity of the cybercrime, their proliferation, and the limited amount of computing and human resources available to combat it, digital investigators are under more pressure than ever to use digital forensics and investigative techniques to find answers quickly. This research advocates for the adoption of more 'agile' methods to overcome the limitations of existing forensic instruments and effectively tackle cybercrime issues. The study emphasizes the importance of leveraging on 'Agile' approach and 'Triaging' methods to provide a design of novel and improved investigation protocol that can improve decision making capabilities of first responders. Also leveraging on Artificial Intelligence (AI) and Machine Learning (ML) technologies in order to develop sophisticated and 'intelligent' tools for better prediction and decision making. This chapter consists of sections like Introduction to chapter, Global challenges faced in the field of DF, Digital intelligence versus Intelligent Forensics, Application of AI to Digital Forensics, Intelligent Digital Evidence Extraction Protocol (I-DEEP) Design, Research Experiment Ontology and chapter summary.

CHAPTER 6: DEVELOPMENT AND DEMONSTRATION OF PROTOTYPE BASED ON I-DEEP PROTOCOL

This chapter entails the discussion around 'Demonstration stage' of DSR model adopted for this research, which is demonstrated by developing a working prototype, that is based on the design of novel protocol (I-DEEP) described in chapter-5. The research followed the Design Science

Research (DSR) Approach, more specifically DSRP model by Peffers et al. (2007) and culminated in designing a novel investigation protocol called the Intelligent Digital Evidence Extraction Protocol (I-DEEP). This protocol is designed to be agile and easily implementable, following the core methodological process of digital forensic investigation and evidence gathering. The research involved implementing the conceptual protocol model into a prototype application to collect evidential and cybercrime data. To obtain valuable insights into results, improved efficiency and perform predictive analysis, AI techniques were integrated into the protocol subsequently. This involved development of an Intelligent Framework called Digital Intelligent Forensic Framework (DIF^2) which consisted of use of AI (Machine Learning) algorithms, automation, selection of best performing algorithm. Optimisation of experiment pipelines, in order to improve efficacy and achieve better decision making in the investigative process. These implementations are discussed in detail in chapters -7 and 8, which also resulted in creation of customised algorithms for prediction modelling and image classification. Additionally, visual representations of data are generated to enhance the understanding of the characteristics of cybercrime, utilizing specific datasets extracted from the collected data using the prototype application. A web-based application interface is created to showcase the capturing and storage of data in a database. The prototype design was based on the Intelligent Digital Evidence Extraction Protocol (I-DEEP) Model, which served as the foundation for the implementation of the protocol. This was done to demonstrate that the protocol is 'agile' and can be easily implemented in practical scenarios. It also allowed for integration of AI technology into the working prototype to achieve better decision making using visual representation of cybercrime data. This chapter consists of sections like Introduction, Prototype design, Demonstration of implementation in AI in prototype using DIF^2 framework and chapter summary.

CHAPTER 7: DIGITAL INTELLIGENT FORENSIC FRAMEWORK (DIF^2)

This chapter entailed the development of novel algorithms that harnesses AI and Machine Learning for decision-making through 'triaging' and predictive modelling. By utilizing AI technologies such as machine learning and pattern recognition, the proposed DIF^2 framework-based system is aimed to elevate the efficiency and precision of digital investigation process and decision making. It demonstrates the capability to automate tasks, minimize manual labour, and expedite the analysis of bigger datasets that mostly exists in cloud computing. Through case-based reasoning, the system

learns from past cases and data, refining decision-making processes and aiding in the identification of patterns in cybercrime data. The integration of AI into computer forensics holds promise for optimizing investigative procedures, overcoming resource limitations, and enhancing outcomes. The Digital Intelligent Forensic Framework (DIF^2) is introduced as a conceptual framework first and then implemented using algorithms and procedures. The framework incorporates components like AI algorithm selection, pipeline optimization, and predictive analysis to further bolster investigative capabilities. By embracing AI-based systems and case-based reasoning, investigators can enhance their capabilities, achieve greater efficiency, and effectively uncover critical insights into the data that helps in decision making. This chapter consists of sections like Introduction, Transition from evidence extraction to analysis phase, Need for AI integration, Integration for Machine Learning algorithms for predictive modelling, Digital Intelligent Forensic Framework (DIF^2) implementation, Create a 'Baseline model' for predictive modelling, Implementation of AI modelling using 'Test Harness Method', Visual Comparison of AI model performance, Creating algorithms of 'Triage Score' calculations, Implementing Prediction Modelling and chapter summary.

CHAPTER 8: INTELLIGENT FORENSICS MODELLING - DIF^2 AUTOMATION

This chapter entails further expansion of the newly developed DIF^2 framework using implementation of TPOT and AutoML. The process followed the 'iterative and incremental' model of 'agile approach' in order to achieve 'Automation' and 'Optimisation' of framework using customised AI algorithms and auto-selection can be achieved without human intervention. This chapter also covers development of customised algorithms based on ANN and SGD ensemble model, demonstrated in Experiment-3 and a customised 'Enhanced CNN Model' for image analysis. These stage-wise enhancements, constitutes an iterative 'intelligent framework' development based on 'agile approach', which allows for AI integration and performance capabilities in the system. These newly developed artifacts not only align with the established standards of the digital forensic investigation process but also enhances their capabilities further. While existing tools can perform basic analysis of evidential data, they lack the advanced interoperable intelligence discussed earlier in this research and do not support wide AI integration possibilities. The proposed new and customised algorithms and framework enhancements offer potential system formulations aim to address the gaps identified in this study. The framework

combines classical programming (Python), artificial intelligence (AI), and machine learning (ML) techniques, leveraging on a cybercrime dataset from test data generated using newly developed prototype for training and testing (demonstration phase). As AI systems operate based on their knowledge and learned patterns, the approach is referred to as intelligent systems and intelligent forensics. Retraining is necessary for consistent results, and additional training datasets were utilized to thoroughly train the AI algorithms, following validation with test datasets for variable sizes (50 vs 100) and (50 vs 100 vs 300 records). It was demonstrated that the DIF^2 framework integration into prototype as well as other tools can empower investigators to make informed decisions in different scenarios. The integration of AI following 'iterative and incremental' agile approach and DSR methodology, description of operational ontology of the 'Automation Process' of DIF^2 framework (using TPOT and AutoML) were explained in the chapter in detail, which provides a roadmap for implementing 'Automation' and 'Optimisation' features in the AI models with minimum human intervention and also achieving best performance using hyperparameter optimisation. This implementation helped in making predictive analysis performance better, which demonstrated that automation can be effective to reduce human intervention and results into better decision-making capabilities for first responders and digital forensic investigators. This chapter incorporates sections like Introduction, Automated ML (AutoML), Implementation Specifications of AutoML and Operational Ontology, Automation of DIF2 Framework using TPOT and AutoML, Critical Analysis of AutoML, Need for developing 'Customised Algorithm' for performance enhancements using ANN+SGD ensemble and algorithm development, Creating customised and enhanced algorithm for image identification and classification (CNN), Model Extension of DIF2 framework using ANN and CNN algorithms for predictive analysis and image analysis, Discussion on DIF2 Extended framework and Summary.

CHAPTER 9: EVALUATION OF PROTOTYPE AND INTERPRETATION OF RESULTS

This chapter evaluates the prototype, discusses the experimental results and presents the performance output of experimental setup (experiments 1 to 7) which were conducted to evaluate the performance of prototype and framework, all newly created artefacts (algorithm models) and their implementations into DIF^2 Framework. This evaluation also provided the valuable observations on different AI models performance and demonstrated applicability of AI automation and optimisation using pipelines and use of AutoML and TPOT based automated frameworks.

This experimental evaluation also provided a motivation for AI integration into I-DEEP protocol and DIF^2 framework and also demonstrated the feasibility of implementation of triaging and predictive analysis of cybercrime incidents for better decision making. This chapter also evaluated prototype and DIF^2 framework performance and their assessment towards achieving better prediction modelling using automation frameworks and pipelines and testing of various artefacts. The chapter consists of sections like Introduction to chapter, Evaluation of cybercrime dataset, Evaluation of results of experiments (1 to 7) using variable datasets (50 vs 100 vs 300 records) conducted on AI models and Summary.

CHAPTER 10: CONCLUSION

This chapter provides an overview of all the novel artefacts designed, developed and evaluated in this study and contributions of this research to the body of knowledge in the field of Digital Forensics. Since the main objective of this study was to design a novel AI based Digital Forensic Protocol for the first responders, this was achieved in the form of conceptual design and further implementation of the I-DEEP protocol using a working prototype. Another main goal of the study was to develop an 'intelligent digital forensic framework' that demonstrated the technical feasibility of AI technology into the framework in order to achieve 'triaging and prediction' that could facilitate the first responders and digital forensic investigators to make quick decisions by predictive analysis of cybercrime incidents and severity cataloguing of the cybercrime data gathered during investigations. The novel protocol (I-DEEP) developed in this research is agile and easily implementable, which was demonstrated by designing a working prototype based on the newly conceptualized protocol model. To integrate AI technology into the protocol an 'intelligent framework' named as Digital Intelligent Forensic Framework (DIF^2) was also developed. Various algorithms were developed in a step-wise and systematic manner with added complexity and enhanced functionality to demonstrate the feasibility of DIF^2 framework. The chapter also discusses 'Recommendations' in terms of predictive analysis of 'realtime' and 'cloud data' emanating continuously from IoT devices, social media streams as well as realtime transactions (server logs) and scope of future research. The chapter consists of sections like Introduction, Research contribution, New Artefacts developed and their evaluation, Justification of the research, Limitation of research, Recommendations, Future scope and Summary.

1.10 Summary

This chapter introduces the major challenges faced by cybercrime investigators and first responders in the Digital Forensic (DF) domain. Due to various factors such as the rapid growth of the internet and the exponential increase in digitally stored data, cyber-investigators and first responders are encountering challenges in conducting timely investigations. Compliance issues contribute to delays in the investigative process, compounded by the complexities of vastly diverse and dynamic computing infrastructures. The methodologies employed in Digital Forensic investigations are often overly complex and impractical to implement efficiently. This raises the question of how effectively first responders and cybercrime investigators can conduct investigations amid the challenges presented by evolving technological paradigms while adhering to prescribed methodologies and meeting legal benchmarks. This research proposed that integrating artificial intelligence (AI) into investigation protocols and developing intelligent framework-based tools can help overcome the challenges faced by first responders and cybercrime investigators. The main objective of the research was to propose solutions for overcoming challenges posed by changing technological paradigms, increasing data and digital device usage, and a growing number of internet users. Cybercriminals are taking advantage of novice users who are not well-informed about existing cyber threats. In addition, the research aimed to identify gaps in the digital forensics domain caused by the rapid evolution of technology, such as cloud computing, mobile devices, storage media, and the Internet of Things (IoT). The complexity is further increased by big data and web 2.0. To address these challenges, this study intended to develop new artefacts – 'an investigation protocol with AI-based intelligent framework' to improve the efficacy of digital forensic investigation process and a working prototype to evaluate the efficacy of the artefacts. The goal was to help DF first responders and cybercrime investigators in their work with this newly developed protocol (I-DEEP) and AI based framework and finally provide contributions to the field of Digital Forensics.

CHAPTER 2: LITERATURE REVIEW

2.1 Introduction

According to Snyder (2019), a literature review is a comprehensive and unbiased assessment of academic articles, books, and other relevant sources related to a particular issue, theory, or field of study. Its objective is to analyse and summarize these works in the context of the research problem or domain being investigated. Essentially, a literature review provides an overview of the literature concerning a specific research area. In contemporary society, digital devices such as laptops, tablets, mobile phones, desktop computers and gaming consoles have become essential. As the use of these devices increases in our daily lives, there is a possibility for misuse of the data contained or obtained from them for illicit activities. Computer-based crimes such as hacking, financial fraud, murder, drug trafficking, terrorism, and forgery are frequently committed (Sadiku et al., 2017).

Digital Forensics (DF) has emerged as a vital tool in combating computer-based crimes and has been adopted by law enforcement, computer security, and national defence sectors. It has been integrated into the frameworks of law enforcement agencies, financial institutions, and investment firms to perform cybercrime investigation or provide concrete evidence of cybercrime. Initially known as Computer Forensics, Digital Forensics evolution gained momentum between 1990s and 2000s. It has gained interest from various sectors such as law enforcement, legal profession, business community, policymakers, educational institutions, and government. Digital Forensics is widely used in private investigations and criminal law, and precise standards are followed to withstand cross-examination in a court of law (Chaturvedi et al., 2020).

Digital Forensics is an all-encompassing term used for plethora of computing devices as mentioned earlier. Digital Forensics can be defined as – "The use of scientifically derived and proven methods toward the preservation, collection, validation, identification, analysis, interpretation, documentation and presentation of digital evidence derived from digital sources for the purpose of facilitating or furthering the reconstruction of events found to be criminal, or helping to anticipate unauthorized actions shown to be disruptive to planned operations" (DFRWS, 2001).

The emergence of handheld computing devices such as Personal Digital Assistants (PDAs), smartphones, GPS systems, and other mobile computing devices has expanded the scope of digital forensics beyond traditional computer systems. What makes this shift even more intriguing is the frequent connection of these devices to internet services for communication and information exchange. Consequently, forensic investigations must now encompass a broader range of devices and networks to effectively uncover digital evidence and address modern cyber threats.

Therefore, we find sub disciplines of computer forensics like mobile-forensics, network- forensics, cloud-forensics etc. Although a relatively new discipline, Digital Forensics is rapidly growing as a result of contributions from research community, software tool developers and digital forensic practitioners contributing in software tools, frameworks and evidence extraction sub-domains (Paul & Norman, 2019; Johansen, 2017).

The expansion of internet use has inadvertently led to an increase in anti-social or criminal elements, posing potential threats to the evolving internet community. Cyber criminals target internet users due to their lack of security interventions or ignorance, resulting in exponential growth in cybercrime (Vacca, 2010). Digital computing devices serve as either instrument of crime or arenas where cybercrime manifests, necessitating interventions to bring perpetrators to justice. Cyber criminals exploit the fact that they are not bound by geographical boundaries and may use anonymous technologies like the tor network and browser to hide their identities (Schultz, 2016).

As a result, cyber-investigators and first responders face difficulties in conducting timely investigations. Compliance issues pervade the investigative process, and the vastly diverse and dynamic infrastructural implementations in computing environments pose significant challenges (Casey et al., 2009). The methodology used in Digital Forensics (DF) investigations must comply with court directives and be rigorous enough to stand in court. Therefore, the effectiveness of first responders and cybercrime investigators in carrying out their investigations under changing technological paradigms while adhering to prescribed methodologies and meeting legal benchmarks is in question (Kohn et al., 2013).

One of the main objectives of this study is to explore problem areas in the digital forensics Incident Response (IR) process and provide solutions to first responders and cybercrime investigators by designing an investigation protocol that incorporates Artificial Intelligence (AI) to enhance the

efficiency of digital investigations. The proposed protocol is aimed to guide investigators by recommending the most appropriate investigative approach that integrates all technical and methodological aspects of the investigation process. Additionally, the study also aimed to create an intelligent framework based on AI and develop a working prototype based on the protocol to assist cyber investigators and professionals with methodical documentation of cybercrime details, decision-making using triaging, and predictive analysis of cybercrime incidents.

This chapter provides a literature review that examines the concepts of digital forensics and the challenges faced by digital forensic investigators in the process of Incident Response and evidence collection and analysis. It also identifies the issues and gaps that exists in the presently adopted methodologies and frameworks in the domain of digital forensics, incident response (IR) and data analytics.

2.2 Evolution of Digital Forensics

DF research is a new emerging discipline that addresses the various procedural, technological and methodological aspects of investigating computer/digital/cyber-crime (Baggili et al., 2013). Digital Forensics not only deals with collection and analysis of digital evidence but also conceptual, legal, methodological, and technological aspects involved in the process. The main aim of a digital forensic investigator is digital evidence collection that has to follow strict legal procedures outlined by the court so that the evidence stands credible (DFRWS, 2001; B. D. Carrier & Spafford, 2004).

Digital Forensics Research can be broadly categorised into two domains namely a) theoretical/conceptual research b) Developmental research. These two domains lack an integrated approach which further results into disparities that have been elaborated upon in this study (Kohn et al., 2013; Johansen, 2017).

2.3 An Overview of Digital Forensic Process

Prior to the late 1990s, Digital-Forensics was commonly referred to as 'Computer-Forensics'. The earliest experts in computer forensics were law enforcement officials who also had a passion for computers. The FBI-CART also known as “Computer Analysis and Response Team”, began their

operations in the year 1984, while in the UK, the Metro Police established a cybercrime division under an expert called John Austen. This was created within the Fraud Squad in the same year.

A major shift happened in 1990s. Technical support personnel and investigators within UK law enforcement agencies, and external experts, recognised that Digital-Forensics, just like other 'general' field of forensics investigation, required standardized skills, procedures, and processes. Apart from other 'informal' strategies, these formalities never existed and there was a need to develop them immediately. A number of seminars, mainly organized by the departments "Serious Fraud Office" and the "Inland Revenue", were held at the "Police Staff College" in city of Bramshill in the year 1994 and 1995, during which the establishment of current "British Digital Forensics Organization" was completed (Gunawardhana, 2021).

According to Burden and Palmer (2003), "Digital forensics is the use of scientifically derived and verified approaches towards the preservation, collection, validation, identification, analysis, interpretation and presentation of digital evidence derived from digital sources for the purposes of enabling or expanding the modernization of proceedings found to be felonious or facilitating to antedate the illegal activities shown to be disruptive to scheduled actions" (Burden & Palmer, 2003).

The digital forensic process is a systematic procedure that adheres to established guidelines commonly used in forensic investigations. Given that these investigations pertain to digital computing and communication devices, they are termed digital forensic investigations. In legal contexts, the seized media is referred to as an "exhibit". Various digital forensic investigation models and frameworks exist, each following distinct methodologies and proving useful in different types of investigations. Due to the vast diversity of investigation environments, the ideal digital forensic framework must be adaptable to future requirements and evolving technologies (Du et al., 2017).

2.4 Digital Forensic Frameworks and Models

In the early 2000s, digital forensics and digital forensic process models and frameworks were still in their infancy. One of the significant concerns were to establish a process framework or model that could make the entire investigation process methodical, consistent and stable. Several digital

forensic process models were proposed during this time. These frameworks and models generally described a set of essential phases in a complete investigation process, and over time, they were refined and developed (Du et al., 2017). Advanced models built upon earlier ones by incorporating additional phases or by describing sub-phases in more detail. Multiple new frameworks were also developed alongside the traditional model, and there were some inheritance relationships among these frameworks.

This research focussed on a list of some very significant frameworks that hold major relevance to this research scope and agenda, although this list may exclude some similar models and frameworks. The main focus of this research was on the process models that incorporate methodological and legal processes and covering all frameworks and models is not in the scope of this study. A detailed discussion on these DF frameworks and models is needed to establish a proper understanding of the evolution of these framework and models and subsequent developments that followed.

2.4.1 Investigative Process Model for Digital Forensic Science (DFRWS, 2001)

Digital Forensics has made remarkable advancements despite being a relatively nascent field. These developments include sophisticated software for gathering and analysing digital evidence, as well as the establishment of procedures. In digital forensic science, a process- model comprises of several stages to conduct DF investigations. Such models are typically based on prior experience. However, given the range of cases and investigative approaches, there is no one-size-fits-all workflow in digital forensics.

A main procedure for digital forensic investigation should outline a series of critical activities for the investigation, broad enough to apply to various cases, and adaptable to integrate new techniques as they emerge. An excessively minimal model may not provide sufficient guidance, while a complicated model with too many steps and restrictions may be impractical. Although creating an ideal process model which can handle all forms of investigation is virtually impossible, a well-designed framework can yield significant benefits.

Digital Forensic Research now encompasses every facet of the contemporary cyber world, given the widespread accessibility of computing and internet technologies. Whether in business and industry, military establishments, or law enforcement and judicial bodies, Digital Forensics has

grown substantially in importance. These organizations rely heavily on it due to the surge in cybercrime incidents. One of the earliest attempts to standardise digital forensic process were made in Digital Forensics Research Work Shop (DFRWS) held in Utica in 2001, where goal was set to define a framework for Digital Forensic Investigation Process.

Recognizing the necessity to categorize and develop a process model that aids practitioners in visualizing the requirements and capabilities of digital forensic technology, the initiative was taken in its first ever 'Digital Forensic Research Workshop' or DFRWS (2001). This conference\workshop resulted into experts agreeing into following a standardised process for digital forensic research. That's where the first model gets its name from as DFRWS Model which is most widely accepted model in the domain. The nucleus of DFRWS (2001) is shown in Figure 2.1, which illustrates some key focus areas in Digital Forensic (DF) research process. It was also established that academic researchers can analyse the processes to identify gaps in technology and propose to fill them by focussed research.

It was also recognized that computer forensic analysis serves to assist the law enforcement community, necessitating adherence to statutory and regulatory guidelines established for traditional forensic disciplines. Thus, it is essential for digital forensic tools and technologies to comply with sound scientific methodologies capable of producing credible digital forensic evidence admissible in the court of law (DFRWS, 2001).

2.4.2 DFRWS Linear Process Model (DFRWS, 2001)

Digital Forensic Process consists of various phases as described in figure-2.2, which illustrates a DFRWS Linear Process Model (DFRWS, 2001). This model comprises of stages from identification to preservation of evidence and finally decision stage. These phases have been discussed in detail in section 1.2.8 and a critical analysis also provided in discussion of model and its phases. The study identified the gaps and has presented an improved or 'extended model' that provides integration roadmap of new technologies like AI into the process model, which has been discussed in forthcoming section 2.5.7. Figure-2.8 provides the illustration of 'Extended Linear Process Model' which is a new artefact that provides provisions to integrate AI techniques and predictive analysis into DFRWS 'Linear Process Model'. Figure-2.2 represents the classical

DFRWS 'Linear Process Model' which represents the digital investigation phases defined and their activities mapped in the model.

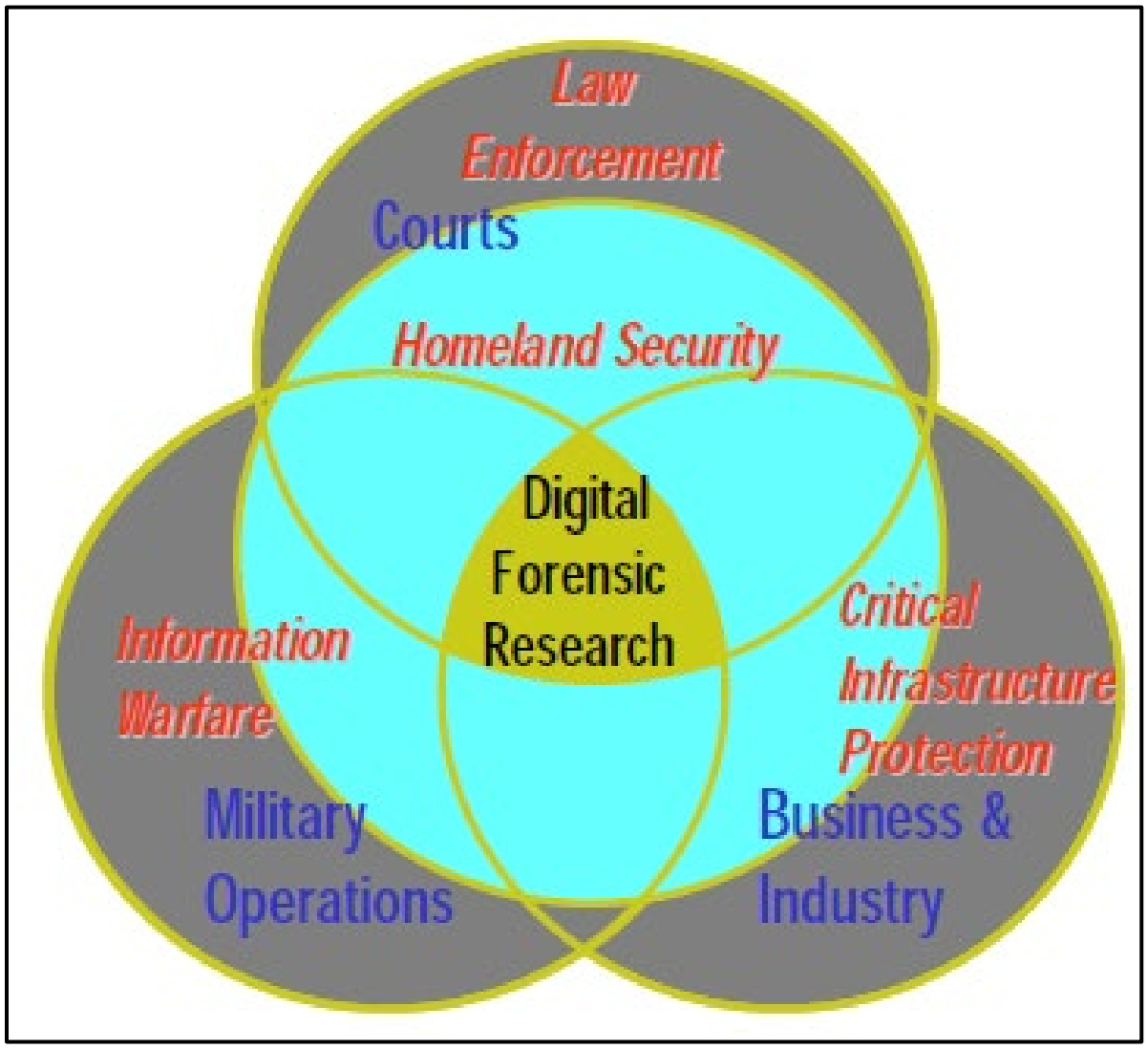
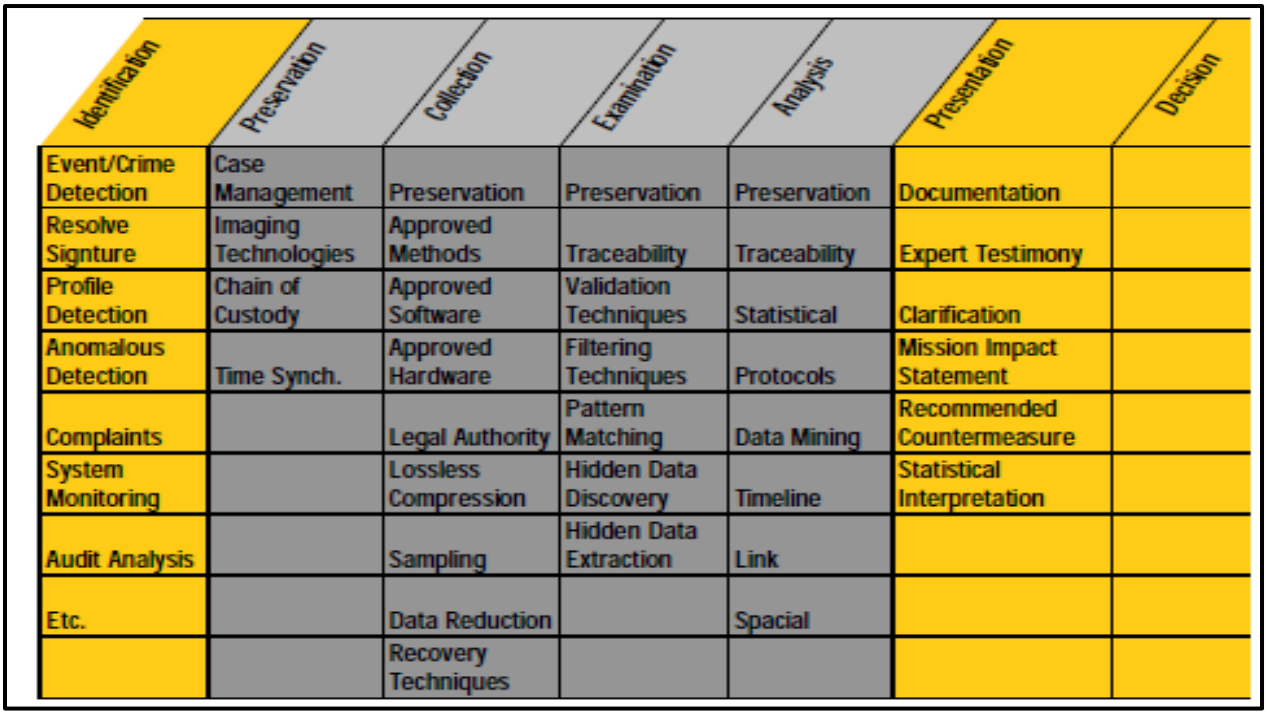


Figure 2.1: The Nucleus of Digital Forensic Research (DFRWS, 2001)).

Since the first DFRWS (2001) conference, numerous process models have been proposed and developed, as mentioned earlier, with many showcased at various DFRWS conference proceedings. DFRWS has been instrumental in the fundamental needs of practitioners and has made significant contribution in establishing core standards, refining investigation methodologies,

enhancing technological tools, and crafting digital investigation frameworks and models. This study will delve into some of these noteworthy contributions and will be reviewed in this study.



Identification	Preservation	Collection	Examination	Analysis	Presentation	Decision
Event/Crime Detection	Case Management	Preservation	Preservation	Preservation	Documentation	
Resolve Signture	Imaging Technologies	Approved Methods	Traceability	Traceability	Expert Testimony	
Profile Detection	Chain of Custody	Approved Software	Validation Techniques	Statistical	Clarification	
Anomalous Detection	Time Synch.	Approved Hardware	Filtering Techniques	Protocols	Mission Impact Statement	
Complaints		Legal Authority	Pattern Matching	Data Mining	Recommended Countermeasure	
System Monitoring		Lossless Compression	Hidden Data Discovery	Timeline	Statistical Interpretation	
Audit Analysis		Sampling	Hidden Data Extraction	Link		
Etc.		Data Reduction		Spacial		
		Recovery Techniques				

Figure 2.2: Investigative Process for Digital Forensic Science, Linear Process Model (DFRWS, 2001)

2.4.3 Building Theoretical Underpinnings for Digital Forensics Research Process

In her work, Sarah Mocas (2004) argues that while research can be pursued independently from practical application, in Digital Forensics, an applied discipline, it's essential for researchers to grasp the practical context of their work. This understanding becomes even more critical when research leads to the development of digital forensic tools, as these tools must produce evidence that withstands scrutiny, ensuring reliability for law enforcement and admissibility in court.

Moreover, Mocas (2004) emphasizes the significance of separating the investigative context from the technical environment to enhance the model's comprehension. Within the technical realm, she

further distinguishes between static scenarios, such as analysing a hard disk image, and dynamic scenarios, such as investigating a live system.

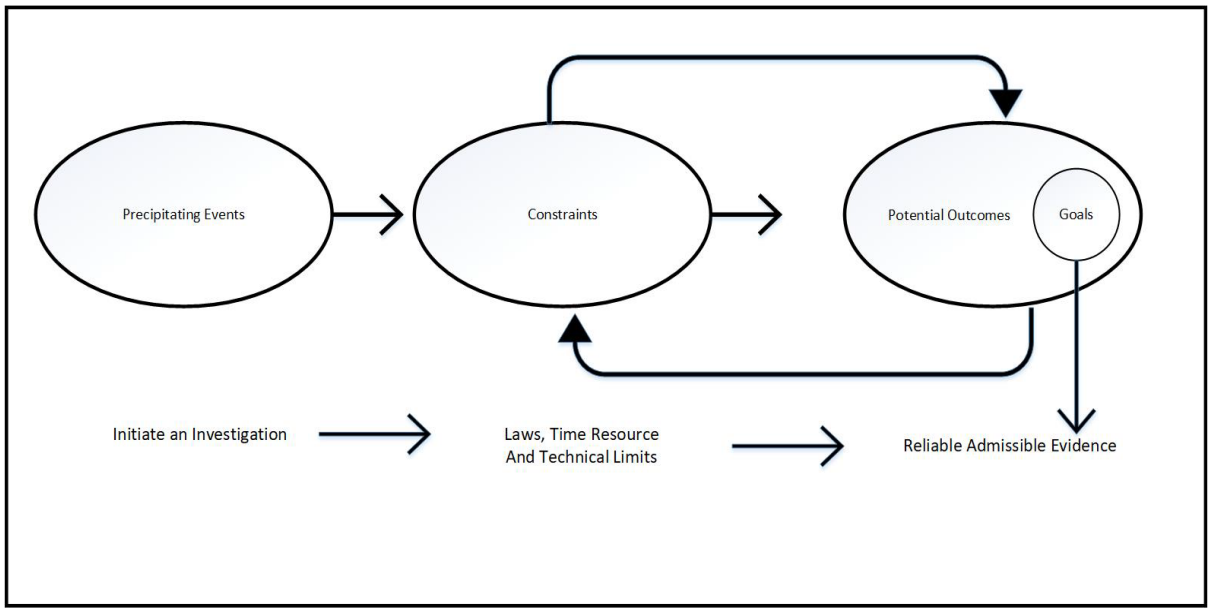


Figure 2.3: Investigative Context in a Law Enforcement Setting (Mocas, 2004).

2.4.4 An Event-Based Digital Forensic Framework by Carrier and Spafford (2004)

Carrier and Spafford (2004) introduced a framework for digital forensics during DFRWS 2004, primarily comprising an investigation process model. They emphasized that every digital device implicated in a crime should be regarded as a "digital crime scene", which is an extension of the physical crime scene where the device is located (Carrier & Spafford, 2004).

Their proposed framework is rooted in a process model that treats a computer under investigation akin to a physical crime scene. Consequently, they asserted that "An investigation is a process that develops and tests hypotheses to answer questions about events that occurred". Carrier and Spafford (2004) also drew a distinction between 'physical' and 'digital' evidence, stating that a

storage media (or device) constitutes physical evidence, while the data or files recovered from it represent digital evidence.

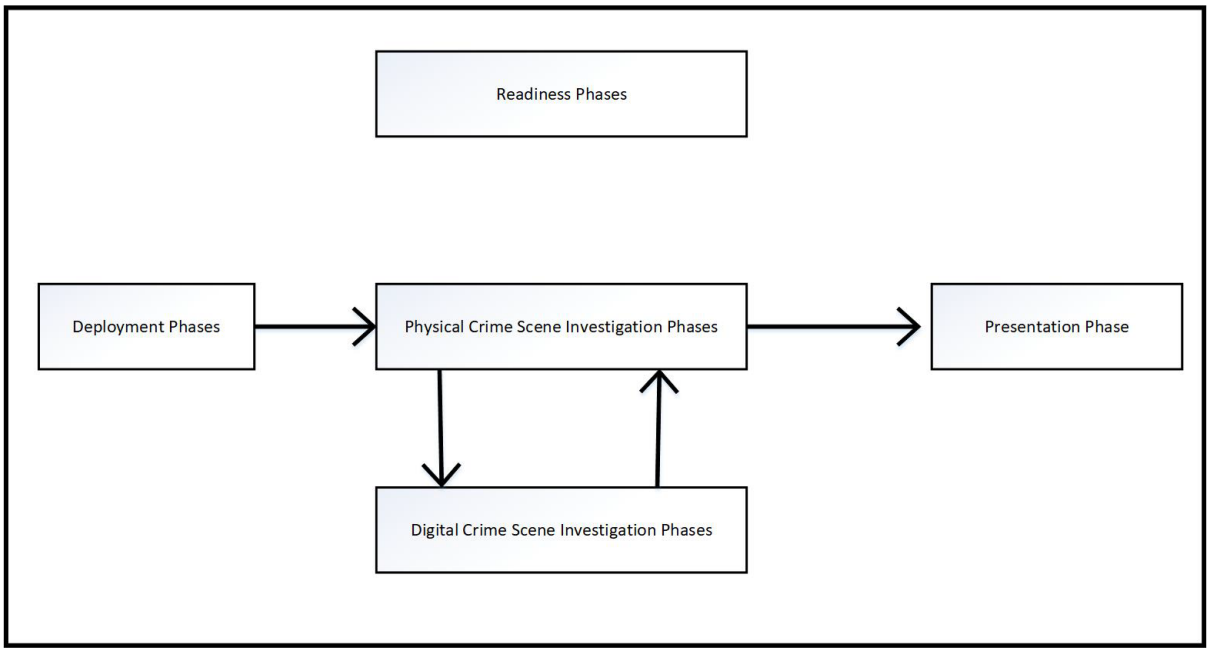


Figure 2.4: An Event Based Digital Forensic Investigation Framework (Carrier & Spafford, 2004).

In their framework, Carrier & Spafford (2004) delineate five phases, beginning with the Readiness Phases, which concentrate on ensuring the readiness of tools and operations to be used in system investigation. This involves the Infrastructure Readiness Phase, focusing on equipment configured with necessary data. The Deployment Phase comprises two key stages: the "Detection and Notification Phase", triggered when an incident is detected and reported by the victim to relevant authorities, and the "Confirmation and Authorization Phase", where investigators must obtain permission to conduct an investigation. Subsequently, the "Physical Crime Scene Investigation Phase" involves examining any physical objects, such as digital computing or communication devices.

In the 'Digital Crime Scene Investigation' Phases, the focus is on examining digital data on the physical device (digital device or media) recovered during the 'physical crime scene investigation' phase. This investigation is conducted for each independent device. The 'Presentation Phases' involve presenting facts and findings to the client or court of law. These phases include additional

sub-phases such as "System Preservation and Documentation", which involves preserving the system and documenting its state; "Evidence Searching and Documentation", which entails searching for evidence and documenting it; and "Event Reconstruction and Documentation", which involves reconstructing events and documenting them (Carrier & Spafford, 2003; Carrier & Spafford, 2004; Carrier & Spafford, 2006).

2.4.5 Integrated Digital Forensic Process Model by Kohn et al. (2013)

In their research, Köhn et al. (2013) put forward an Integrated Model aimed at standardizing the investigation process within Digital Forensics (DF). They employed Unified Modelling Language (UML) to represent the DF investigation process, utilizing Use Case and Activity Diagrams for visualization. The Integrated Digital Forensic Process Model (InteDFPM) proposed by Köhn et al. (2013), amalgamates process models from Kruse & Heiser (2001) and the U.S. Department of Justice (Institute of Justice, 2008). This model is explained through UML Activity and Use Case diagrams. Additionally, in subsequent research, Köhn et al. (2013) examined various models using Sequential Logic Notation and Process Flow Diagrams. While they were not the first to utilize standardized modelling tools and languages (e.g., UML, Sequential Logic Notation, Process Flow Charts) in digital forensic process modelling, their contributions in the comparison and refinement of DF process models is significant.

2.5 Triage based Digital Forensics Models

Triaging comes from medical field where it involves prioritizing patient's treatment based on severity of their medical condition, whereby symptoms are analysed and quick decisions are made. It has found a natural fit in DF investigation process. Use of digital triage in initial investigative stage can provide a major lead into investigations and achieving positive outcomes for the first responders. Triaging can be further classified as live triage and post-mortem triage (Jusas et al., 2017).

Digital Forensics Field Triage can provide a timely response to a cyber-crime investigation specially, where delays can cause serious life threats and a quick action becomes imperative. Digital Forensic Field Triage Process Model (DFFTPM) also referred as Computer Forensic Field Triage Process Model (CFFTPM), is a model that is based on field and evidence triaging proposed by Rogers et al. (2006) and can be used to provide fast leads. As mentioned earlier also, triaging

comes from medical field where it involves prioritising patient's treatment based on severity of their medical condition, whereby symptoms are analysed and quick decisions are made. Digital Triaging on the same concept involves taking a quick lead into investigations based on type of cyber-crime, various possibilities or scenarios at the crime scene, certain usable evidence and investigative team expertise (M. K. Rogers et al., 2006). The Field Triage model has proven to be effective for time-sensitive cases (Hitchcock et al., 2016; M. Rogers et al., 2006). By using digital forensics triage, researchers can quickly identify relevant evidence, and law enforcement can obtain leads on the cyber-criminals faster, rather than having to wait for the complete information, which could take several months or even years (Casey et al., 2009)

2.5.1 Computer Forensics Field Triage Process Model

Rogers et al. (2006) proposed the model, which emphasizes the importance of incorporating 'Field Triage' in any forensic approach. Rogers et al. (2006) noted that individuals engaged in criminal activities have been using technology to enhance their trade-craft. In their evaluation of various investigative models designed to aid law enforcement in processing digital evidence, it was observed that these models were time-consuming, mainly due to the requirement of transporting digital evidence to a central location for analysis. In situations that demand urgency, such as cases involving child abductions, kidnappings, and terrorist threats, the imperative for swift information analysis and investigative leads takes precedence over the necessity for a thorough analysis of all digital evidence. Therefore, Rogers et al. (2006) emphasised on the necessity of rapid information and investigative leads, which can be provided through Field Triage.

The "Computer Forensics Field Triage Process Model" or CFFTPM is ideal for "those investigative processes that are accompanied within the initial few hours of an investigation, that deliver information used throughout the suspect interview and quest implementation stage" (Rogers et al., 2006). The CFFTPM is designed to enable the collection of usable evidence in a short time frame, which typically involves conducting an on-site or field investigation of the computer system(s) in question. One of the key advantages of this model is the ability to quickly identify victims who are at risk and guide ongoing investigations, including identifying potential charges and accurately assessing the danger that the offender may pose to society.

While the CFFTPM is a valuable starting point, it relies on the expertise of experienced forensic specialists. Its principles can serve as a foundation for developing a model that enables non-forensic experts to complete similar tasks. Although the CFFTPM highlights the benefits of on-site investigation, it acknowledges the need for further investigation at a later time (Hitchcock et al., 2016)

As being emphasised in this study that a timely response to a cyber-crime e.g. involving paedophiles and child abductors, where delays can cause serious life threats and a quick action becomes imperative, evidence triaging can provide fast leads. Digital Triaging involves taking a quick lead into investigations based on type of cyber-crime, various possibilities or scenarios at the crime scene, certain usable evidence that can be analysed quickly to initiate response and augment investigative team expertise (Rogers et al., 2006).

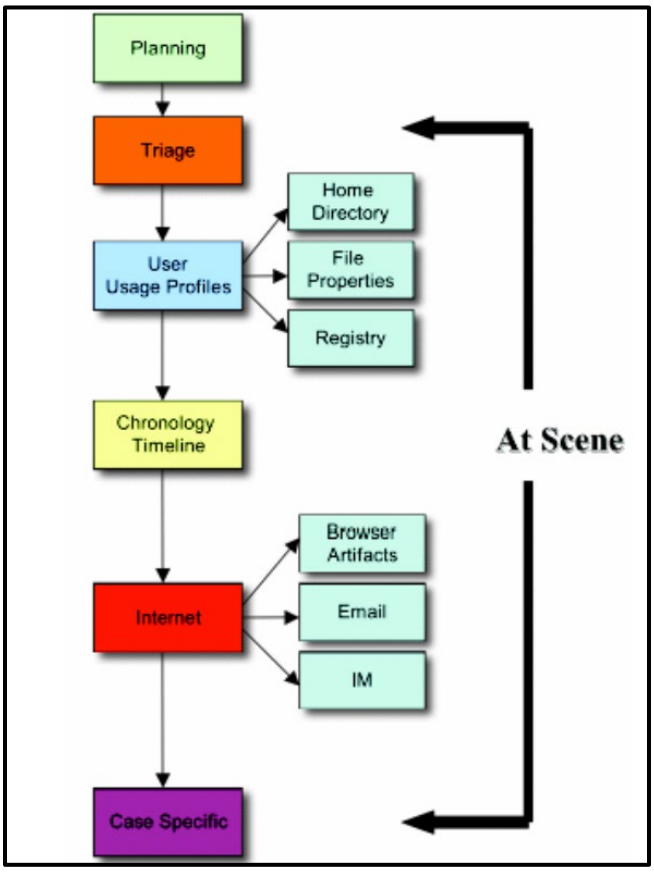


Figure 2.5: Computer Forensics Field Triage Process Model (Rogers et al., 2006)

Computer Forensics Field Triage Model (CFFTPM) can provide a timely response to a cyber-crime forensic investigation. CFFTPM is based on field and evidence triaging proposed by Rogers et al. (2006), can be effectively used to provide quick leads to initiate a response by first respondents while further evidence can be analysed. Due to the aforementioned reasons, some of the theoretical constructs of CFFTPM have been used for field triage implementation in this research.

2.5.2 Digital Field Triage Model

Hitchcock et al. (2016) introduced the Digital Field Triage (DFT) model with the intention of delegating certain key tasks within the digital forensic domain to non-digital evidence specialists. The model has two primary objectives: first, to enhance the efficiency of investigations by enabling timely access to digital evidence, and second, to alleviate the backlog of cases at forensic laboratories. The DFT model, rooted in the work of Rogers et al. (2006), comprises four levels: planning, assessment, reporting, and threshold. However, it is important to note that the DFT model is not without inherent risks, including challenges related to management, training, and the absence of supporting tools.

The effective implementation of the DFT model relies on two crucial factors - efficient management and continuous training. The management tools employed should facilitate smooth functioning of the DFT model (as illustrated in Figure 2.6), which equips non-experts in digital evidence with the necessary knowledge, skills, and capabilities to conduct limited forensic activities (M. Rogers et al., 2006). However, there exist three primary conditions that must be met for the DFT model to operate effectively. Firstly, it cannot function independently and must operate in conjunction with a parent Technological Crime Unit (TCU). Secondly, the forensic integrity of the digital evidence must be preserved at all times. Lastly, a DFT evaluation cannot substitute a thorough TCU investigation.

Evidence and Field Triaging can provide timely response to a cyber-crime (e.g. involving paedophiles and child abductors), where delays can cause serious life threats and a quick action becomes imperative. Since, Digital Triaging involves quick leads into investigations based on type of cyber-crime, various possibilities or scenarios at the crime scene, certain usable evidence and investigative team expertise, some of the aspects of the Computer Forensics Field Triage Process

Model (CFFTPM) by Rogers et al. (2006) and Digital Field Triage (DFT) Model by Hitchcock et al. (2016) has been used as a base model for implementation of triaging in the in the novel protocol model developed in this research.

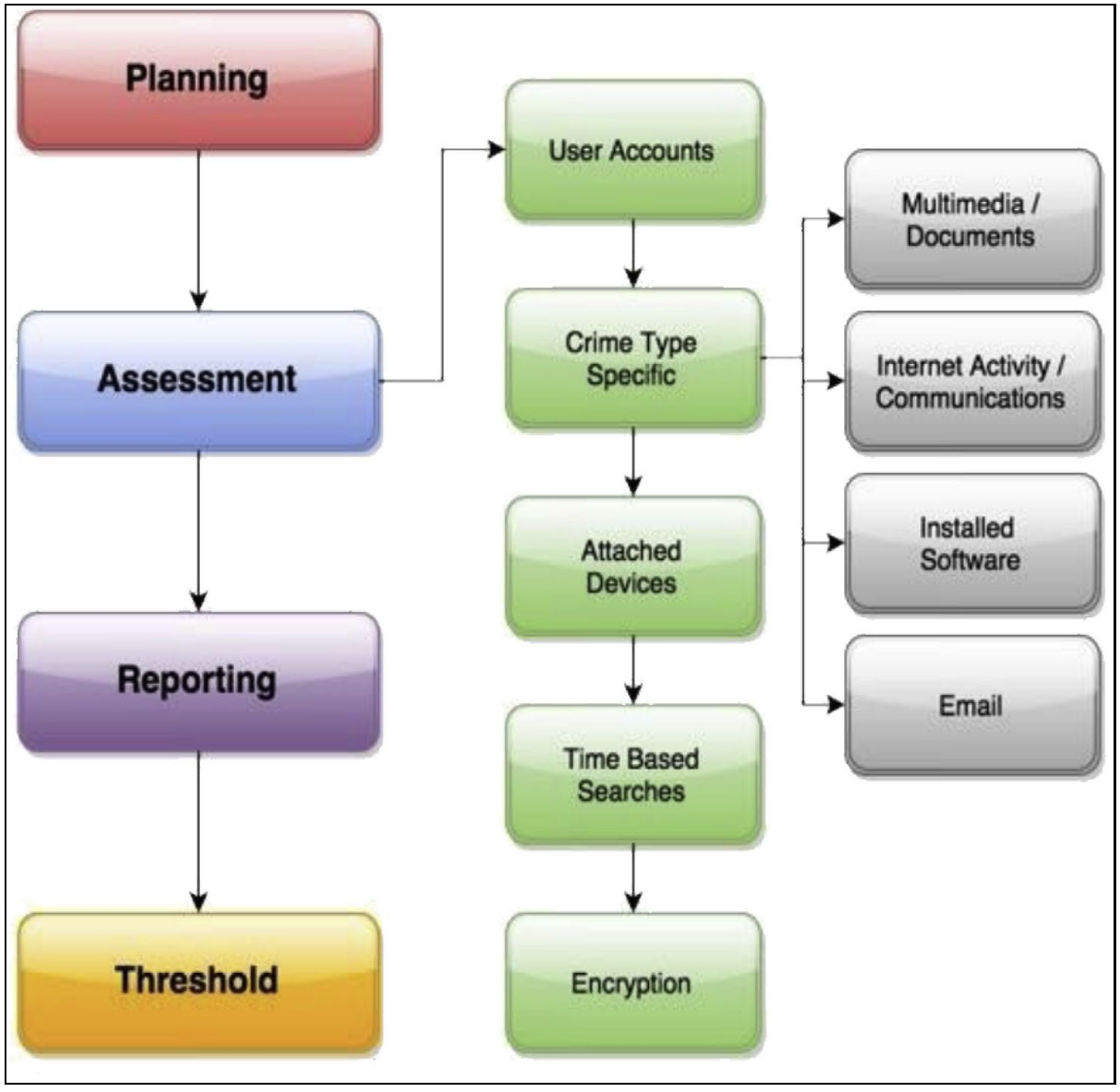


Figure 2.6: Digital Field Triage Model (Hitchcock et al., 2016).

It is to be noted that the goal of this research is not to cover all the models/ frameworks and analysis of the models is not utterly comprehensive. This study highlights key issues and reviews them in the context of their implementation by tools developers and efficacies achieved.

Shortcomings and Criticism

While the models analysed above serve various investigation processes, there isn't a universal model that can be universally applied across the discipline of Digital Forensics. The nature and requirements of investigations can vary widely depending on factors such as the type of digital evidence, the context of the investigation, legal considerations, and technological advancements.

The primary objective of these models is to delineate the typical stages of investigations, their sequential order, and the critical elements of each stage (Beebe, 2009; Burden & Palmer, 2003). However, these models have faced criticism from various researchers over time. In 2001, Henry Lee introduced the "Scientific Crime Scene Investigation" or SCSI model for digital forensic investigations (Lee et al., 2001; Lee et al., 2013). However, Ciardhuáin (2004) criticized the SCSI model, highlighting its lack of emphasis on digital crime scene investigation and its exclusive focus on physical crime scene investigation.

To address this criticism, Kohn et al. (2013) proposed a modification to the traditional physical crime scene investigation process by integrating digital crime scene investigation. Their Event-based Digital Forensic Investigation Framework distinguishes between physical and digital crime scenes, involving the collection of digital devices from the physical crime scene and the retrieval of digital evidence from storage devices (Carrier & Spafford, 2004). Casey (2009) presented a digital forensic process model, which focused on digital evidence processing and examination.

Additionally, Baryamureeba & Tushabe (2004) introduced the "Enhanced Integrated Digital Investigation Process" or EIDIP model. Building upon the Integrated Digital Investigation Process (IDIP), the EIDIP model incorporates a 'traceable phase' to address the need for restructuring observed in the IDIP. These efforts aim to refine and enhance digital forensic investigation processes to better suit the evolving nature of digital crime.

This criticism underscores the importance of considering the unique challenges and requirements of digital forensic investigations and ensuring that models adequately address these aspects.

Having discussed and critiqued various process models under Section 2.4 and current section, this section addressed research sub-question-1, while it also explained in detail various process models and framework that are used by cybercrime investigators and first responders.

2.5.3 Attempts to Refine Digital Forensic Process Models

Merely following a generic process model is often insufficient to handle the diverse range of cases typically encountered by law enforcement. Criminals could be IT professionals engaged in sophisticated cybercrimes, the investigation may involve consideration of CCTV camera footage storage, or data leakage within an organization, among other scenarios. These varying circumstances often necessitate customized approaches.

Subsequent to the description of the overall process methodology, certain researchers have delved into specific issues in greater detail. For instance, some researchers have sought to enhance a process model by improving a particular phase of the investigation, while others addressed only a particular type of case, such as network forensics or mobile devices forensics. Some researchers postulated 'triage' based models (M. Rogers et al., 2006; Hitchcock et al., 2016) that outline detailed processes for complex cases involving missing persons, child abductions, and other time-sensitive matters.

Despite numerous efforts to standardize the investigation process, leading to the proposal of several models by various researchers, none of them has achieved universal acceptance. Richard Brian Adams (2013) further emphasized this issue, contending that many forensic experts predominantly concentrate on law enforcement, overlooking other crucial areas such as incident response or commercial aspects (Adams et al., 2013).

There is lack of a consistent theoretical framework in the field of cyber-forensics. Generally, ad-hoc methods have been implemented for evidence collection using various forensic tools. The efficiency of these tools is quite limited as credibility of both tools and forensic methods can be disputed. Therefore, it is necessary to standardise and improve forensic processes and tools (Carrier & Spafford, 2004). Networks or more specifically Internet have become powerful and effective tools of digital and cyber-crime. This puts huge pressure on present digital and network forensic techniques, which fall short of preventing computer and internet crimes, conducted every

day (Mocas, 2004). Technical challenges also arise from wide disparity among platforms and application used for cyber forensic investigations. Furthermore, it has to be clearly established what type of cyber-attack has been conducted in order to launch a forensic investigation (Raghavan, 2013).

One of the bigger challenges that DF investigators face is a requirement to analyse and process vast amount of data using effective data mining techniques (Garfinkel, 2010). Some other hindrances for digital forensic experts are - data collection whilst servers and routers across the networks are in running state. To further make things difficult for forensic investigators, cyber criminals use various evading techniques to defy investigation. They deliberately create difficulties for investigation while wiping out evidences and introduce intentional doubts in investigation process by misleading the investigators (Moore, 2010; Moore, 2014).

Various anti-forensic techniques are commonly used, such as altering file extensions, utilizing swap space, employing disk wiping software, causing physical damage to media, utilizing anonymizing techniques, committing identity theft by using someone else's account, utilizing cryptography, steganography, and encryption techniques. Anonymous data storage (cloud storage) is also a significant challenge for internet forensics. Criminals may target online storage services, which can be exploited using stolen credit card data. As noted by Collie (2018), often, these illegal activities take place in countries with ineffective cyber laws.

Therefore, this discussion on challenges encountered by DF investigators in this section is an attempt to highlight the issues related to digital forensic investigation process, mainly Incident Response (IR) and data analysis and address our research sub question-2**.**

Therefore, it is evident that the challenges that are faced by cybercrime investigator and first responders can be summarised as – The field of cyber-forensics lacks a consistent theoretical framework and standardized investigation process, and current ad-hoc methods using various forensic tools have limited efficiency and credibility. There are technical challenges related to

diverse platforms/applications used for cyber forensic investigations, and processing vast amounts of data without effective data mining techniques.

Cyber criminals use anti-forensic techniques to create hindrances and introduce doubt in investigation processes, including changing file extensions, disk wiping software, anonymizing techniques, and encryption techniques. Anonymous data storage and exploitation of online storage services using stolen credit card data also pose significant challenges for internet forensics, and most cybercrimes are launched or conducted from countries with ineffective cyber laws.

2.5.4 Critical Analysis on Digital Forensic Process Models: Theoretical versus Developmental Perspectives

DF research can be categorized broadly into two basic domains based on the disparities that exits between these different frameworks and models namely 'theoretical or conceptual' and 'developmental'. Theoretical\conceptual research mainly constitute researchers in the field that have proposed numerous digital forensic models, methodologies and processes. Many of these models are inconsistent, overly complex and difficult to implement in real time scenario (Adams et al., 2013). There is no single model or investigation approach that can adequately meet all required goals (Carrier & Spafford, 2006; Kohn et al., 2013).

The 'developmental' research deal with application aspects in more detail and focuses on digital forensics tool development. This domain consists of many tools which can be categorised as commercial or open-source. Although this is more pragmatic domain, the problem with the tools and solutions is that they have been developed on ad-hoc basis and do not follow any specific conceptual model(s). Therefore, they are used discretely by the DF investigators and it is difficult to integrate them to a specific methodology of investigation.

2.5.5 Critical Analysis of DFRWS (2001)

While it is established that DFRWS (2001) and the 'Linear Process Model' described in figure 2.2 is the most popular and well adopted framework in digital forensics. It is also observed in this research that standards and guidelines put forward by DFRWS (2001) and further described in 'Linear Process Model' shown in figure 2.2, does not conform to many methodologies and tools

or techniques analysed in literature review, such as artificial intelligence (AI) and can swiftly assist cybercrime investigator and first responders in decision making.

Therefore, we posit that there is a clear gap that exist in present frameworks depicted in DFRWS (2001) Nucleus Model and Linear Process Model also represented in figure 2.1 and figure 2.2, and therefore this scenario demands some improvements. The nucleus of digital forensics investigation process (figure 2.1) does not address some key domains of digital forensics. Furthermore, it does not represent the intersecting aspects between various components described in the model accurately. The constructs are somewhat misplaced and demand a new impetus on changing paradigm of digital forensic domain.

It is also observed that components like forms of cybercrime, legal procedures, incident-response (IR) via computer emergency response teams (CERT), are missing from the model. Digital Forensic Tools, Standards and Frameworks that is applicable to all components of DF, for example 'Law Enforcement', 'Defence and Security' and 'Business and Industry', is also missing from the framework. Therefore, the space represented at the intersection of all three domains, demands to be filled by new research and development in the domain of digital forensics investigation process. This research has tried to fill this space with new proposed constructs that can complement existing model and also fill the gaps to improve the model. This introduction of new components also establishes a roadmap to justify and guide further progression of this research.

2.5.6 Proposed Improvements in Digital Forensic Investigation Process Model

As it is an observation that digital forensic investigation process models described in figure 2.1 and 2.2 have specific gaps that demands a revamp of these models. An improved model of Nucleus of Digital Forensic Research Process is provided as a new artefact in this section, which is presented in figure-2.7. New constructs like Computer Emergency Response Teams (CERT), Digital Forensic (DF) Tools, Standards and Frameworks that are applicable to all components of DF, for example 'Law Enforcement', 'Defence and Security' and 'Business and Industry', were missing from the Nucleus Model and have been added to create a new extended and updated model as shown in figure-2.7.

Therefore, the space represented at the intersection of all three domains, demands to be filled by new research and development in the domain of digital forensics investigation process. This

research has tried to fill this space with new proposed constructs that can complement existing model and also fill the gaps to improve the model. This introduction of new components also establishes a roadmap to justify and guide further progression of this research.

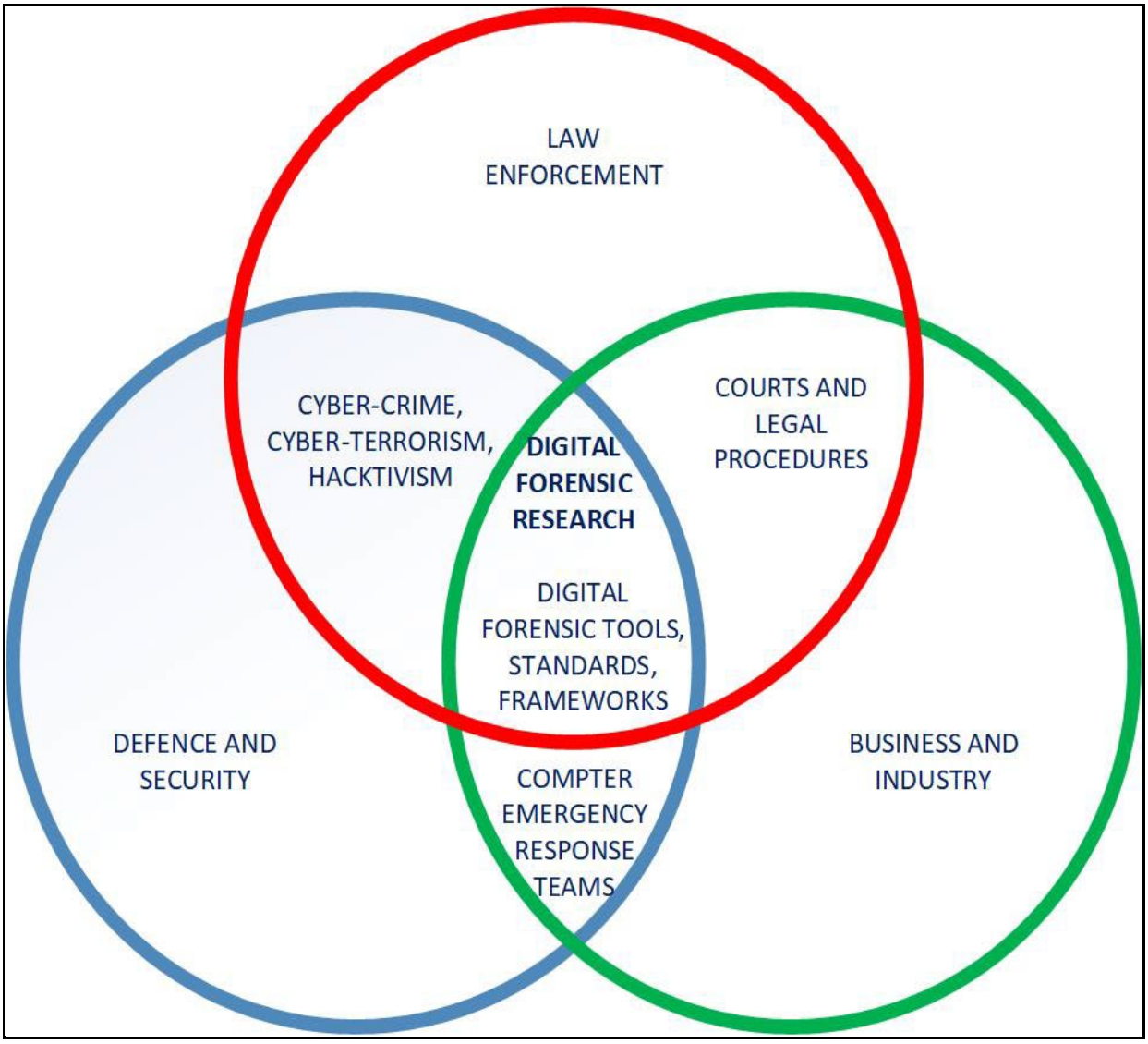


Figure 2.7: Digital Forensic Research Process Nucleus (DFRWS, 2001) Extended Model (New Artifact)

The 'Extended Nucleus of Digital Forensic Research Process' focusses on constructs like CERTs for effective Incident Response (IR), Digital forensic Standard and frameworks to standardise the process and emphasis on digital forensic research to continuously strive for improvements in the

domain of digital forensics. These new constructs also served as theoretical underpinnings of design and development of new artefacts in this research.

2.5.7 Extended Linear Process Model for Digital Forensics Investigation

As described earlier and shown in figure 2.2, a Linear Process Model adopted by DFRWS (2001) represents standardised phases and activities associated with those phases. Since it was discussed in previous sections that the model is generic in nature and has not been modified or extended to accommodate new developments in the DF investigation process domain. Therefore, this provides an opportunity to fill the gaps in existing model.

Figure 2.8 represents an 'Extended Linear Model' developed as a new artefact in this research, in order to bridge the gap and provide theoretical underpinning for design and development of a novel 'Investigation Protocol with AI based framework', which will be implemented via a working prototype to bring intelligence and efficiency to existing frameworks.

In this 'Extended Linear Model', while other phases correlate to standard DF investigation activities, the process 'Optimisation' has been extended to introduce 'Predictive Analysis' in the 'Analysis' phase of the Linear Investigation Model. Furthermore 'Visual Representation' in the form of various diagrams like scatter charts, correlation matrix plots, and whisker plots can be used to visually represent the data and results that can create a better understanding of the data and trends (Sathiyanarayanan, 2017). These theoretical underpinnings will provide the template to implement methodological and functional improvements while designing and developing new artefacts (DF investigation protocol and working prototype).

The extended model of Nucleus of Digital Forensic Research elaborated in Figure 2.7 and extended Linear Model represented in the figure 2.8 provide theoretical underpinnings for improvements in the existing models/frameworks and also provides a conceptual roadmap for this research to design and develop new artefacts to assist DF first responders. This section addressed the research sub-question-5 and investigated the challenges raised thereof.

IDENTIFICATION	PRESERVATION	COLLECTION	EXAMINATION	ANALYSIS	PRESENTATION	DECISION
Event/Crime Detection	Case Management	Preservation	Preservation	Preservation	Documentation	
Resolve Signature	Imaging Technologies	Approved Methods	Traceability	Traceability	Expert Testimony	
Profile Detection	Chain of Custody	Approved Software	Validation Techniques	Statistical	Clarification	
Anomalous Detection	Time Synch.	Approved Hardware	Filtering Techniques	Protocols	Mission Impact Statement	
Complains		Legal Authority	Pattern Matching	Data Mining	Recommended Countermeasures	
System Monitoring		Lossless Compression	Hidden Data Recovery	Timeline	Statistical Interpretation	
Audit Analysis		Sampling	Hidden Data Extraction	Link		
***Optimisation**		**Data Reduction**		**Spatial Predictive Analysis**	**Visual Representation (Scatter charts, Matrix plots, Whisker Plots)**	**Prediction & Triaging**
		Recovery Techniques				

Figure 2.8: Extended Linear Process Model for Digital Forensic Investigation (new artefact)

2.6 International Standards in Digital Forensics

Nieman (2009) asserts that the main focus of Digital Forensics (DF) is the application of forensic procedures, legal concepts, rules of evidence, precedents, and processes, rather than computers themselves. However, despite the growing importance of DF tools and technologies in this field, the absence of standardized protocols is concerning. Surprisingly, no major efforts have been made to establish or adopt protocols, standards or rules for handling digital evidence in the context of challenges posed in present day digital paradigms. Additionally, technical processes used for digital evidence are admissible in court without any formal testing requirements, as highlighted by

Scholtz (2009). As a result, the DF industry primarily relies on self-regulation, using international best practices, guidelines, case laws and frameworks developed by various industry bodies.

2.6.1 An Overview of International Standards used in Digital Forensic Investigation

Because of the constantly changing landscape of ICT (Information and Communication Technology), and differences in legal frameworks across various jurisdictions, there is a shortage of internationally recognized standards to regulate the processes and procedures utilized in digital forensic investigations. The International Organization for Standardization (2014) notes that while some standardized procedures and processes are available as guidelines, there are no set standards in this area. However, several standards could impact digital forensics, including "National Institute of Standards and Technology" also commonly referred as NIST, the "American National Standards Institute" or ANSI, and Scientific Working Group on Digital Evidence (SWGDE). As a result, DF professionals must possess a comprehensive understanding of these protocols and standards.

In the South African context, the "Good Practice Guide for Computer-Based Electronic Evidence" (1997) published by the "Association of Chiefs of Police" or ACPO is utilized as the foundation to establish standards. The "International Organization for Standardizations" also referred as ISO standards provide the framework for these standards.

2.6.2 ACPO Principles for Digital Forensic

The "Good Practice Guide for Computer-Based Electronic Evidence" was formulated by the Association of Chiefs of Police (ACPO) in 1997. These principles established a framework for digital forensic practitioners. In October 1999, an "International Hi-Tech Crime and Forensic Conference" conducted a comprehensive review of the principles. In 2001, the "13th International Criminal Organization's (Interpol) Forensic Science Symposium" formalized and approved additional protocols, with the participation of South Africa as a member (Van Dijk, 2015).

Since then, these principles formulated by ACPO, have provided a guiding framework for digital forensic investigators to develop procedures to ensure that the collection, handling, and management of evidence, is in compliance with the established principles. The guide consists of

four principles that pertain to the collection as well as efficient management of digital evidence obtained during the investigations (Association of Chief of Police Officers, 1997; Van Dijk, 2015).

The set of principles regarding the management of digital evidence includes four key guidelines. The first one emphasizes the importance of preserving data in its original form without any tampering, with the understanding that it could be used as evidence in legal proceedings in the court of law. The second principle highlights the limited circumstances under which investigators are allowed to access the original data. Moreover, investigators should possess the necessary qualifications, and the actions taken must be justifiable in terms of their relevance and implications. The third principle stresses the need for investigators to document all procedures applied in the collection and analysis of digital evidence in great detail. This documentation should be clear enough to allow third-party experts to reproduce the same results. Lastly, the fourth principle outlines the obligation of investigators to comply with all legal procedures while handling digital evidence. (*Association of Certified Fraud Examiners*, Accessed on 27-Mar-2023).

In essence, the ACPO guidelines mandate that data should remain unchanged, primary evidence must not be directly inspected, and reliable duplicates should be made for forensic analysis. The task of digital forensics should be entrusted only to competent personnel who adhere to the appropriate legal regulations. It is essential to maintain an adequate record of actions taken, uphold the chain of custody, and apply approved industry procedures and protocols. Adhering to the ACPO (1997) principles and the standards laid under ISO/IEC 27043 as well as 27037 frameworks can provide a trusted forensic and legal framework for digital forensic investigations.

Limitations and Criticism of ACPO

The guidelines provided by the principles ensure that the digital evidence that is under investigation is not altered, and that any access to original evidence is carried out only by competent individuals. An audit trail must be kept to allow for the investigation's actions to be reviewed, evaluated and assessed against the applicable legal requirements. The importance of these principles was confirmed by the South African Law Reform Commission (SALRC) in 2010, which asserted that forensic investigators' actions in accessing digital evidence are not neutral and that the volatile nature of the digital evidence makes it difficult to prove its integrity. The commission also highlighted that if procedures and protocols at crime-scene are not properly

followed, digital evidence may become inadmissible or claimed as "distorted" by the defence (Swales, 2018).

2.6.3 International Organisation of Standardization (ISO Standards)

The ISO/IEC 27043 Standard focuses on principles of investigation of incidents and processes in the field of Information Technology Security. It was approved and published in March 2015. This standard defines the various stages involved in a DF investigation and is categorised into two primary sections: 'Digital Investigation Processes' and 'Concurrent/Parallel Processes' (Valjarević et al., 2016).

Phase	Activity	Parallel Processes
Detection phase	Incidents are detected	Obtaining authorisation to investigate incidents;
First responder phase	Digital forensic investigators attend to incidents	
Planning phase	Investigations of incidents are planned	
Preparation phase and scene documentation phase	Preparation steps are taken to investigate incidents and document actions are taken on scenes	
Evidence identification phase	Potentially relevant evidence is identified	Documentation of all actions during investigations; Continual information flow between digital forensic investigators and forensic investigators; Maintaining chain-of-custody; Preserving the integrity of evidence.
Evidence collection phase	Evidence is collected	
Evidence transportation phase	Evidence is transported from scenes to digital forensic laboratories	
Evidence storage phase	Digital evidence is securely stored	
Evidence analysis phase	Evidence is analysed to determine relevance	Interaction with physical investigations
Evidence interpretation phase	Evidence is interpreted in relation to its evidential value	
Reporting phase	Evidence is reported on	
Presentation phase	Testimonies or overviews are provided regarding evidence	
Closure phase	Cases are archived	

Figure 2.9: The ISO/IEC 27043 Standard Phases & Processes (Re-modelled by the researcher)

Criticism of ISO/IEC 27043 Standard Model and Suggestions for Improvement

On analysing the model, it was observed that the parallel phases depicted in the model are not well synchronised and aligned to corresponding timelines. Therefore, the artefact represented in figure-2.9 has been re-modelled to provide a more streamlined and detailed representation of model "The ISO/IEC 27043 Standard Phases & Processes". The revised model clearly shows the timeline for when parallel processes should be initiated to ensure consistency and synchronization in the activities performed during a digital forensic investigation. In addition, specific activities and processes that should run concurrently are now color-coded with the same colour for ease of reference. Some key processes of ISO/IEC 27043 Standard are highlighted and discussed in forthcoming section.

Obtaining Authorisation

In every stage of DF investigation, it is crucial that appropriate authorization is obtained from relevant parties, including authorities of government, principals, system owners, and system custodians. The type of authorization required may vary based on the particular circumstances, such as obtaining a search warrant or a consent under 'Section 20 or 21 of the Criminal Procedure Act (51 of 1977)'. It is essential for DF professionals to be aware of the relevant requirements and up-to-date case law in such situations (Cross, 2008).

Preservation of the Chain of Custody of Evidence

As emphasised by Van der Merwe et al. (2008), the importance of maintaining a chain of custody to ensure the integrity of evidence. This involves demonstrating to the court that the digital evidence was not tampered with from the time of seizure to the time it was presented in court. Digital forensic investigators must, therefore, keep digital evidence secure while the analysis process is continuing to maintain its integrity.

The "ISO/IEC DIS 27037" Standard expanded on the chain of custody requirements, focusing on the investigator's ability to account for all evidence acquired while it was under their custody. The "chain of custody" is essentially a chronological record that documents the movement and handling of evidence.

This record should include a unique identifier; a chain of custody also includes proper records of when the evidence was accessed, where was this done, and by whom the evidence was accessed; who authorized checking the evidence in or out of storage area and why; and any modifications to the evidence, who was responsible for the modifications, and the rationale for presenting the evidence in the court (International Organisation of Standardisation, 2012). This topic is further described in detail along with sample documents in Chapter 4: Digital Forensics: Incident Response, Section – 4.5.

2.6.4 ISO 27037 Standard & Security Techniques: Guidelines to identify, collect, acquire and preserve digital evidence

The ISO 27037, also known as "Standard on Information Technology Security Techniques," gained approval and was formally published in October 2012. This standard provides guidance for “identifying, collecting, acquiring, and preserving” digital evidence. By adhering to the analysis methodologies outlined in the standard, digital forensics investigators can uphold the integrity of digital evidence during the "Collection Phase" of investigations, thereby enhancing the admissibility of evidence in legal proceedings. Kanellis (2006) underscores the pivotal role of proper evidence management to prevent degradation and potential inadmissibility in court.

The International Organization for Standardization (2012) delineates four essential principles for digital evidence collection in the ‘ISO/IEC DIS 27037 Standard’. These principles encompass minimizing the handling of original evidence, documenting actions taken, accounting for any data alterations to ensure reliability, following local rules of evidence, and not exceeding one's level of expertise (International Organization for Standardization, 2012). Additionally, the ‘ISO/IEC DIS 27037 Standard’ notes that in many jurisdictions, the digital evidence gathered is subject to seven principles, as described herein.

Relevance of Evidential Data

Digital forensic investigations adhere to the standard requirement of collecting only relevant data. This implies that the gathered data should contribute to the examination of incidents or other aspects of the case at hand, with a justifiable reason for its collection. The Criminal Procedure Act (51 of 1977), specifically Sections 28, 31, and 210, governs wrongful searches and seizures. Evidence found to be irrelevant becomes inadmissible, and items not needed for criminal

proceedings are to be returned, supporting this requirement. Digital forensic investigators must justify their procedures, validating all grounds for collecting any form of data. The admissibility of evidence should be guided by the principle that it has sufficient relevance to the matter under investigation (International Organization for Standardization, 2012).

Reliability of Evidence

The handling of digital evidence must involve processes that are auditable and repeatable. Independent parties must be able to reproduce the same results when they follow these processes. Hofman (2006, as cited in Myburgh, 2016) emphasises that digital evidence must meet the standard requirements for document admissibility, including authenticity, reliability, and originality.

Sufficiency of Evidential Data

To ensure adequate analysis, digital forensic investigators must collect all relevant information. They should also be able to justify their decisions on which data to acquire and how much of it, as well as provide a clear indication of the volume of data to be considered. (International Organisation of Standardisation, 2012).

Auditability of Evidential Data

To ensure accountability and transparency, a separate and independent team of forensic investigators or auditors must audit the actions of DF investigators. Therefore, all the procedures, processes and results should be documented to enable audits. Digital forensic investigators must be capable of explaining the methodology used during analyses and justifying the decisions made to ensure their work is credible and can withstand scrutiny.

Repeatability of the Investigation Process

Repeatability refers to the ability to obtain identical results when using the same procedures, methods, and equipment under the similar conditions, irrespective of the situation. However, achieving repeatability is not always feasible, especially when analysing volatile memory or live data. In such cases, DF investigators must ensure that the acquisition process is adequately reliable, even if they cannot achieve complete repeatability.

Reproducibility of Results

The term "Reproducibility of Results" refers to the capability of obtaining identical test outcomes while using the same process, even if different type of equipment is utilized under different circumstances. It means that the results of a test can be reproduced without much deviation from the initial results, at any time after the initial test using the same method, regardless of the equipment used or the testing conditions. This is important in digital forensic investigations because it enables other investigators to validate the results and conclusions of a particular analysis (Myburgh, 2016).

Justifiability of Digital Investigation Process

To ensure credibility, digital forensic investigators must validate every step taken while identifying, collecting, analysing, and handling any form of digital evidence. They can demonstrate that their decisions were optimal practices under the prevailing circumstances by providing justifications and showing that all possible digital evidence was obtained.

2.6.5 Limitations and Criticism of ISO/IEC DIS 27037

Although widely recognized, even the ISO standards appear to avoid establishing inflexible regulations in a digital forensic setting. The “ISO/IEC DIS 27037” Standard's scope explicitly states that it offers only "guidelines for specific activities in handling potential digital evidence; these processes are: identification, collection, acquisition and preservation of potential digital evidence" (International Organization for Standardization, 2012). As a result, there are no precise procedures or protocols specified to direct digital forensic investigators. This analysis further justifies the cause for undertaking this research and to develop a novel investigation protocol based on AI framework.

2.7 Digital Forensic Standards implemented in South Africa

There is a scarcity of local South African frameworks in cybercrime investigation field, which is evident from this literature review. This results into challenges of local adaptations of digital forensics investigation processes and also their acceptance in the South African court of law. There have been observations that there is a great deal of inconsistencies in use of terminologies of

information technology, digital forensics and their application in legal domain in South Africa. Digital Forensic investigators face challenges of standardisation and legal recognition of terminologies and concepts used in information technology and digital forensics.

According to Nortjé & Myburgh (2019), there are often loopholes in digital evidence collection and analysis and often result in overlooking of key digital evidences due to lack of expertise or the technical complexities that are posed by investigations conducted by DF investigators. There is a need for bringing international standards and principles in order to ensure that digital evidence gathered maintains its integrity, reliability and is admissible in the South African court of law.

There has not been much literature found that covered the cases involving cybercrime extensively and also not much cases where digital evidence has been thoroughly investigated. Out of the cases that have been investigated, there were a number of technical challenges for the state department to handle digital evidence in search and seizure process effectively. These challenges have been partly due to the fact that investigative processes and conventional search and seizure rules, which are applied to physical crime scene are also applied to digital crime scene arena as well but the former differs from the latter significantly.

2.7.1 An Overview of Forensic Investigation Standards Specific to South Africa

The Forensic Standard Forum, which operates under the purview of the South Africa Chapter of the Association of "Certified Fraud Examiners (ACFE SA)", has the primary goal of establishing standardized scientific methodologies for forensic investigations conducted in association with any criminal or civil legislative proceedings. These investigations cover a broad range of disciplines and practices.

The Forensic Standard Forum is crucial in taking a leadership role to establish standards for all forensic disciplines utilized in investigations. While many disciplines have established global standards, some adaptations may be necessary to address the specific circumstances in South Africa and other African countries, considering the unique environment, frameworks, legal systems and applicable legislations.

The forum should also strive to provide a common set of standards and procedures for all forensic practitioners, including digital forensic practitioners, in South Africa. These standards ensure that

forensic practitioners adopt a consistent and reliable approach to forensic investigations, enhancing the quality of evidence presented in court and ultimately fortifying the integrity of the criminal justice system.

The Forensic Standard Forum, therefore, plays an important role in promoting the credibility and effectiveness of forensic investigations in South Africa (*Association of Certified Fraud Examiners*, Accessed on 27-Mar-2023). Due to the absence of a professional body in South Africa that focuses exclusively on enhancing the standard of fraud examination without being limited to a particular profession like law or accounting and the launch of a local chapter, became necessary to cater to these objectives.

This chapter provides a community environment for local forensic examination practitioners and offers several advantages such as access to a network of highly experienced and competent professionals, a practical guidance and training framework, regular technical upgrades and ethical standards and forums to discuss local issues in the field.

2.7.2 A Critical Analysis of Standards and Frameworks in South African Context

Although ACFE-SA covers most aspects of fraud examination procedures and ethical code of conduct, it is an overly generalised framework and does not offer specific guidelines for DF investigation and cybercrime incidents. While general guidelines are applicable on DF investigator as well, there is still a need for DF Investigation Framework that is more customised to South African Context. Therefore, it is observed that there has been a scarcity of local South African frameworks in cybercrime investigation field, which is evident from this literature review. These findings are further supported by Nortjé & Myburgh (2019).

Digital forensic investigators face challenges of standardisation and legal recognition of terminologies and concepts used in Information Technology and Digital Forensics (Nortjé & Myburgh, 2019). There are often loopholes in digital evidence collection and analysis and often result in overlooking of key digital evidences due to lack of expertise or the technical complexities that are posed by investigations conducted by DF investigators. There is a need for bringing

international standards and principles to ensure reliability, integrity, and admissibility of digital evidence in the South African legal domain.

There has not been much literature found that covered the cases involving cybercrime extensively and also not much cases where digital evidence has been thoroughly investigated. Out of the cases that have been investigated, there were a number of technical challenges for the state department to handle digital evidence in search and seizure process effectively. These challenges have been partly due to the fact that investigative processes and conventional search and seizure rules, which are applied to physical crime scene, are also applied to digital crime scene arena as well but the former differs from the latter significantly as explained previously.

The study by Nortjé & Myburgh (2019) talks about the various complexities that investigators at law enforcements face globally while conducting search and seizures operations and how to manage to preserve the fundamental principles of data integrity and reliability of evidence in performing digital forensic investigations. While performing search and seizure, the "Criminal Procedure Act 51 of 1977" guides the investigator to interpret the digital evidence. However, there is a need to further standardise the laws to make digital investigations more credible and acceptable in court of law.

According to International Organisation of Standards (ISO standards), integrity and originality of digital evidence is most important. It can be ensured that digital evidence becomes more reliable and admissible in court using guidelines provided in forthcoming section.

Data Imaging to be done as original evidence should not be examined directly; Data integrity to be ensured i.e. data is not changed, altered or corrupted; Standard and protocols to be followed during data duplication and analysis; Audit trails should be established; Chain of Custody to be maintained to counter non-repudiation; Use of protocols and standards, that are recognised by industry and courts of law must be followed.

2.7.3 An Analysis of South African Legal Framework and it's Compatibility to Digital Forensic Investigation Process

The legal system and digital forensics are closely intertwined, according to US-Cert (2005, as cited in Dumont, 2010), which defines DF process as the "integration of law and computer science to

collect and analyse data from computer systems, networks, wireless communications, and storage devices in a manner that makes it admissible as evidence in a court of law".

The primary objective of DF analysis is to establish accurate and reliable facts that can withstand legal scrutiny. If the evidence fails to meet the standards of legal examination, all preceding efforts become futile. As outlined by Casey (2009), each phase of the DF process must be executed to preserve the integrity of digital evidence and ensure its admissibility in a court of law.

The State v. Ndiki (2008) case in South Africa has set a precedent that digital evidence "should be treated as real or documentary evidence and that the relevant rules of evidence should apply". The court underscored the significance of maintaining the integrity of digital evidence and advocated for a regulatory framework to ensure its authenticity and reliability. It emphasized that digital evidence must be presented in a manner that is trustworthy, accurate, and verifiable, and the evidential weight of such evidence hinges on the reliability of the methods used for its acquisition and preservation.

The legal parameters for digital evidence collection and preservation in South African courts are delineated in the "Electronic Communication and Transaction Act (25 of 2002)," as highlighted by the work of Nortjé and Myburgh (2019). Originating from the United Nations Commission on International Trade Law's (UNCITRAL) Model Law on Electronic Commerce (1996), this act gained approval from the South African Law Reform Commission (SALRC) in 2010. Sections 14 and 15 of the "Electronic Communication and Transaction Act (25 of 2002)" play a pivotal role in establishing crucial legal requirements to uphold the credibility and admissibility of digital evidence.

The statutory framework provided by the "Electronic Communication and Transaction Act" (25 of 2002) is instrumental in governing the admissibility and credibility of digital evidence in South African courts. Sections 14 and 15 of the Act serve as fundamental guidelines, particularly in cases involving search and seizure, where maintaining the originality and reliability of the evidence is of utmost importance.

Nieman (2009) explains the distinctive nature of digital evidence, emphasizing its susceptibility to easy alteration, duplication, or deletion, with the collection process itself having the potential to

significantly alter the evidence. In the South African legal context, Section 14 of the "Electronic Communication and Transaction Act (25 of 2002)" outlines requirements pertaining to the originality of data messages, while Section 15 addresses the admissibility of evidence and the evidential weight of data messages in court. These sections are designed to ensure that digital evidence is collected and presented in a manner in the courts that preserves its integrity as well as its reliability.

Section 15 of the “Electronic Communication and Transaction Act” (25 of 2002) holds particular importance regarding the admissibility of digital evidence. It explicitly states that the legal rules governing evidence admissibility should not discriminate against data messages solely because they are electronic or digital. In essence, this means that digital evidence should be treated on par with any other form of evidence concerning its admissibility, weight, and credibility.

Section 15 of the Electronic Communication and Transaction Act (25 of 2002) requires that the evidential weight of digital evidence be determined based on the reliability of how data messages were generated, stored, or communicated, as well as how their integrity is maintained. The period for assessing data reliability should be specified, beginning with data collection by forensic investigators or seizure from suspects and ending with data presentation in court.

According to international standards, digital evidence must remain unaltered, and maintaining evidence reliability is crucial. The Electronic Communication and Transaction Act (25 of 2002) Sections 14 and 15 comply with these standards. South African courts rely on these sections to determine the integrity of digital evidence. They analyse whether the evidence was modified during analysis and evaluate the methods used to collect and process digital evidence to assess its reliability.

2.7.4 Discussion and Criticism

Maintaining the originality, integrity, and reliability of digital evidence is crucial during searches and seizures. This is because the actions of police officials or investigators at the scene and their handling of digital evidence can directly affect its admissibility in court. As Nieman (2009) points

out, digital evidence is distinct from other types of general evidence and the procedure of collecting it can significantly alter it, if not followed meticulously.

Therefore, it is evident that the integration of law and computer science in digital forensics is important for collecting and analysing data from digital devices to make it admissible as evidence in court. The primary objective of analysis of digital forensic evidence is to institute reliable facts that can withstand judicial scrutiny. Hence, lack of proper intrinsic knowledge or expertise in digital evidence collection and presentation in the court of law can compromise the trial of perpetrators.

These challenges of digital evidence collection and digital forensics investigation processes and also their acceptance in the South African court of law can hinder the legal proceedings. It has also been noticed that there are many inconsistencies in use of terminologies of information technology, digital forensics and their application in legal domain in South Africa, in addition to the scarcity of expertise and professional skills among investigators in South Africa.

Therefore, Section 2.7 provide deep insight into the present situation of technical and legal shortcomings and challenges encountered by the cybercrime investigators and first responders, specifically in South African context and therefore addresses our research sub-questions 3 and 5.

2.8 Digital Forensics Tools Analysis and Shortcomings

A SWOT (Strength Weakness, Opportunity and Threats) analysis of DF tools is imperative to uncover the present state of technology and the gaps that exists thereof. Apart from providing an overview of the tools and technology available to the disposal of first responders, this information can also be integrated into the research to develop an expert system that can augment the new artifact (prototype's) ability in decision making and assisting first responders to search for the most suitable tools for their Incident Response (IR) process.

In early days of computer forensics, the computer forensic tools were developed by technical experts that aimed towards data acquisition and e-discovery. Advancements in the discipline, attempting to improve the investigation process and development of various models, the digital forensic tools have also incorporated some of these models and frameworks. Although, there have been reasonable attempts made by tool developers, there is still lot to be done as no tool can be

efficiently applied to all forms of investigations. Most of the DF tools mainly cater to examination phase of investigation whereby examiners use them to recover and view data (Mocas, 2004).

Some of the tools (e.g. EnCase, Access Data – FTK) have been widely accepted by law enforcement bodies worldwide. There is a wide range of open source or free forensic tools available to support different computing environments and file systems. A comprehensive discussion of all of them is neither possible nor in scope of this study, however a brief mention of them is provided in this section.

The digital forensics industry heavily relies on digital forensics tools, which are crucial for carrying out investigations related to cybercrime or computer crime. These tools have proven to be reliable in generating results that are reliable and have been accepted in courts. However, it is worth noting that practitioners may not receive comprehensive information on how these tools work or the methods they employ to verify that the evidence is authentic. As Carrier (2004) points out, these tools grant access to evidence but not to methods for verifying its reliability, which could pose challenges for inexperienced practitioners. Therefore, it is essential for digital forensics practitioners to be knowledgeable about the workings of these tools and the verification methods they employ to ensure that the evidence produced is reliable and also relevant to the courts (Wu & Zheng, 2020).

2.8.1 Role Based Classification of Digital Forensic Tools

Categorization of digital forensics tools can be based on their function within the digital forensics process, the targeted devices, or the supported operating system. The functions of these tools can include evidence acquisition, examination, analysis, and integration. The sections that follow, will provide a breakdown of each tool category, accompanied by some primary examples and their main features. A comprehensive review is based on functionality, use case, licensing or open source criterion. Tools utilized for digital forensics acquisition are responsible for generating an image or mirror copy of the suspect device. As part of this acquisition process, a cryptographic hash is generated, which plays a crucial role in preserving the chain of custody of the evidence. The main objective of this process is to maintain the integrity of the suspect device, which, in turn, helps in preserving the status of physical evidence from which the digital evidence is extracted, ultimately safeguarding the integrity of the evidence (Wu & Zheng, 2020).

To prevent any data being written to the drive during the acquisition process, digital forensics acquisition tools are commonly utilized with write blockers. Nevertheless, there may be concerns regarding the integrity of the image produced, such as how to verify that the image is an exact replica of the original drive and how to confirm that both the original and the copy are identical. The integration of the tools knowledge and the investigation methodology is essential in addressing these issues.

Examination and analysis tools are often integrated into a single tool to facilitate the extraction and analysis of digital evidence. Digital forensic investigators utilize these tools to retrieve and examine evidence from various devices and sources such as computers, mobile devices, and cloud services (Wu et al., 2020).

The extraction process can be classified into two main types: physical extraction and/or logical extraction. Physical extraction involves creating a bit-by-bit copy of the entire storage device, including all data and unallocated space, irrespective of the file system. On the other hand, logical extraction recovers specific files based on the device's operating system, file systems, and also its applications. A thematic analysis has been done for the DF tools on the themes or categories that can be derived on the classification based on their complexity, availability (free or licensed) and their use-case bases.

Integrated Tools for Multipurpose Investigation

Integrated digital forensic tools offer various functionalities within a single platform, including data acquisition, search, navigation, extraction, examination, analysis, and reporting. These tools are typically designed to work with specific devices or interfaces and can be used to extract and analyse digital evidence from various sources, including computers, mobile devices, and cloud services.

Encase by Guidance Software and FTK by Access Data are two examples of integrated digital forensic tools. Encase is a popular tool that supports both physical and logical data acquisition, as well as analysis and reporting of digital evidence. FTK, on the other hand, is specifically designed for digital investigations and offers similar functionalities to Encase. Both tools are widely used

by digital forensic investigators to extract and analyse digital evidence from various sources (Wu et al., 2020).

The data acquisition feature of digital forensic tools allows for the creation of a forensic image of the device or system under investigation. This process involves copying or creating a mirror image of the device's storage media, including all deleted, hidden, and unallocated files. The search functionality of these tools enables practitioners to search for specific files or data that meet certain criteria. This feature allows for efficient and accurate identification of relevant evidence in a timely manner. The navigation feature of digital forensic tools enables practitioners to explore and visualize the digital crime scene, facilitating the identification and analysis of digital evidence. This feature provides a graphical representation of the data, allowing practitioners to quickly and easily identify key pieces of evidence. The extraction feature of DF tools allows for the targeted extraction of specific data, such as internet browser history and other artifacts, which may be relevant to a digital investigation.

Finally, integrated digital forensic tools automate some aspects of the examination and analysis of digital evidence, making it easier for practitioners to gather valuable information from the data gathered. These tools can provide automated analysis and reporting, allowing for faster and more efficient digital investigations.

Integrated digital forensic tools like FTK and Encase, also referred to as “Case Management Tools”, have emerged as a popular solution to the challenges posed by ever increasing volumes of data-sets that may contain potential digital evidence to sift through. These tools provide various built in features for searching data, applying filters, and data analysis, making it easier and faster for practitioners to identify relevant evidence. Using an integrated tool for digital forensics can also help reduce costs by eliminating the need to acquire multiple tools for each function. This can save time and resources and ensure that all the required functions are available in a single tool.

Integrated tools incorporate specialized techniques that assist practitioners throughout the digital forensics process. These tools can automate tasks such as data acquisition, examination, and analysis, thereby making the process more efficient and effective. Additionally, these tools are designed to produce verifiable digital evidence, ensuring that the evidence gathered meets the standards of admissibility in court. Therefore, integrated digital forensic tools are a valuable asset

to digital forensic investigators, providing an efficient and effective means of extracting, analysing, and presenting digital evidence.

While professional digital forensics investigators typically use proprietary tools, free tools available online are often used in training (22 Popular Computer Forensics Tools, 2018). This can lead to disparities in the field. Thus, this research seeks to address various concerns within the digital forensics field, such as the classification of tools according to practitioner training or professional and industrial contexts. These classifications are highlighted further here.

2.8.2 Classification of Digital Forensics Tools based on License/Proprietary or Open Source Status

Digital forensic tools can be categorised under proprietary or commercial tools, free or open-source tools, specialised e-discovery or hacking tools.

2.8.2.1 Proprietary and Commercial Tools

These are the tools that needs a licence to be purchased for any commercial use. Some of them may be available for free for academic use only.

Encase

EnCase is a powerful digital forensics and incident response software suite that enables investigators to collect, analyse, and preserve digital evidence from a wide variety of sources, including computers, mobile devices, and network environments. The software is widely used by law enforcement agencies, government organizations, and private sector companies to investigate criminal activities, corporate fraud, data breaches, and other security incidents.

EnCase provides a comprehensive set of features for digital investigations, including data acquisition and imaging, evidence analysis and reporting, and case management and collaboration tools. The software can process and analyse data from a wide range of sources, including file systems, email archives, databases, and network traffic. It also supports advanced data carving and artifacts analysis techniques to recover deleted files and uncover hidden information.

EnCase has become one of the most popular digital forensics software suites in the world due to its powerful features, user-friendly interface, and extensive documentation and training resources.

The software is regularly updated to keep pace with the latest developments in digital forensics and cybercrime, and it has been used in some of the most high-profile investigations in recent years.

FTK

FTK (Forensic Toolkit) is a software suite developed by AccessData that is widely used in digital forensics investigations and e-discovery processes. FTK enables investigators to acquire, analyse, and index data from a wide range of sources, including computers, mobile devices, and cloud-based services.

FTK provides a comprehensive set of features for digital forensics investigations, including data acquisition and imaging, evidence analysis and reporting, and case management and collaboration tools. The software can process and analyse data from a wide range of sources, including file systems, email archives, databases, and network traffic. It also supports advanced data carving and artifact analysis techniques to recover deleted files and uncover hidden information.

COFEE

COFEE (Computer Online Forensic Evidence Extractor) is a suite of digital forensics tools developed by Microsoft exclusively for use by law enforcement agencies. The software was designed to run on Windows operating systems and provides investigators with a range of tools to acquire and analyse digital evidence from a suspect's computer system.

COFEE was first introduced by Microsoft in 2008 and was distributed exclusively to law enforcement agencies for free. The software was designed to be used in the field by law enforcement officers with minimal training and expertise in digital forensics. It includes a range of tools to acquire data from a computer system, such as password extraction, registry analysis, and internet history analysis.

The use of COFEE has been controversial in some circles, as it was not made available to the general public, leading to concerns about transparency and accountability. Microsoft has since discontinued the development and distribution of COFEE, and instead, the company now provides

a range of digital forensics tools and services to law enforcement agencies and other organizations through its Azure cloud platform.

Registry Recon

Registry Recon is a digital forensics tool designed specifically for the Windows platform. The software is used to regenerate the Windows registry from hard disk data, which can be extremely useful in digital forensics investigations and incident response scenarios.

Registry Recon is developed by Arsenal Recon, a company that specializes in digital forensics software and services. The software works by analysing the hard drive of a Windows system and reconstructing the registry from the data found on the drive. This can be useful in cases where the registry has been damaged or corrupted, or where the system has been compromised and the registry has been modified by an attacker.

Registry Recon provides a range of features for digital forensics investigators, including advanced registry parsing and analysis, keyword searching, timeline analysis, and reporting. The software is designed to be easy to use and provides a user-friendly interface that allows investigators to quickly analyse and report on registry data.

SafeBack

SafeBack is a digital forensics tool that provides investigators with the ability to create images of internal and external storage devices and analyse data from those images. The software is widely used in digital forensics investigations and serves as a valuable tool for data acquisition and backup purposes.

SafeBack is developed by ForensicSoft, a company that specializes in digital forensics software and services. The software works by creating a bit-for-bit copy of the storage device being analysed, which can then be analysed and searched for evidence using a range of other digital forensics tools.

SafeBack provides a range of features for digital forensics investigators, including the ability to create and analyse images of a wide range of storage devices, including hard drives, USB drives,

and memory cards. The software also includes advanced searching and filtering tools, allowing investigators to quickly and easily locate specific files or pieces of data within an image.

Nuix

Nuix is a digital forensics tool that is widely used in investigations to collect and analyse digital evidence from a variety of sources, including Windows, Mac OS, and Linux systems. The software is designed to be highly scalable and can handle large volumes of data, making it a popular choice for both small and large-scale investigations.

Nuix provides investigators with a range of features for evidence collection and analysis, including the ability to extract data from a wide range of sources, including emails, IP addresses, credit card numbers, and URLs. The software can also analyse data from mobile devices, cloud-based services, and other sources, making it a versatile tool for digital forensics investigations.

In addition to its core features, Nuix also provides advanced search and filtering tools, allowing investigators to quickly and easily locate specific pieces of data within a large dataset.

2.8.2.2 Open Source/ Free Tools

Free digital forensics tools available online are commonly used in training. However, this creates a potential for disparities within the field, including the tools used to train practitioners. This research seeks to address several different tools that offer potential to be used for DF investigation and also address issues within the DF industry. The tools listed below are free tools that fall under GPL or Open Source Licensing and can be used by Digital Forensic (DF) investigators without any major licensing constraints.

CAINE

CAINE (Computer Aided INvestigative Environment) is a digital forensics tool that is designed specifically for the Linux platform. The software is widely used in digital forensics investigations and is known for its ability to facilitate the examination and analysis of data in a forensically sound manner.

CAINE is developed by Nanni Bassetti, an Italian digital forensics expert, and his team. The software includes a range of digital forensics tools and features, including advanced imaging and analysis capabilities, data carving tools, and network analysis tools.

One of the key strengths of CAINE is its ability to handle a wide range of file systems and data formats, making it a versatile tool for digital forensics investigations. The software also includes a range of tools for data recovery and analysis, as well as support for a wide range of hardware and software platforms.

The Sleuth Kit

The Sleuth Kit is a digital forensics tool that provides investigators with a library of tools for evidence collection, analysis, and investigation on a range of platforms, including Unix/Linux, and Windows.

The Sleuth Kit includes a wide range of digital forensics tools, covering all major evidence collection features, such as file system analysis, data recovery, and network analysis. The software is widely used in digital forensics investigations and is known for its flexibility and ease of use.

One of the key strengths of The Sleuth Kit is its ability to handle a wide range of file systems and data formats, making it a versatile tool for digital forensics investigations. The software also includes advanced search and filtering capabilities, enabling investigators to quickly and easily locate specific pieces of data within a large dataset.

The Sleuth Kit is an open-source tool, meaning that it is freely available for download and use by anyone. The software is regularly updated and maintained by a community of developers and digital forensics experts, ensuring that it remains a powerful and effective tool for digital forensics investigations.

BackTrack

BackTrack is a Linux-based operating system that is specifically designed for digital forensics investigations and penetration testing. The software is open-source and freely available for download and use by anyone.

BackTrack includes a wide range of digital forensics tools and features, making it a valuable asset for investigators working on a wide range of cases. The software includes advanced imaging and analysis capabilities, data carving tools, network analysis tools, and a range of other features for evidence collection and analysis.

Overall, BackTrack is a powerful and widely used digital forensics tool that is known for its versatility, ease of use, and advanced capabilities. The software is popular among both digital forensics' experts and penetration testers, making it a valuable asset for anyone working in the cybersecurity field. However, it's worth noting that BackTrack is no longer actively maintained and has been superseded by Kali Linux, a similar operating system that is actively updated and maintained by the community.

Kali Linux

Kali Linux is a popular open-source operating system that is specifically designed for digital forensics investigations and penetration testing. It is a Debian-derived distribution that overtakes BackTrack. It includes a vast collection of pre-installed tools and utilities for various cybersecurity tasks, including vulnerability analysis, penetration testing, and digital forensics.

Kali Linux is widely used by cybersecurity professionals, including digital forensics investigators, penetration testers, and ethical hackers, as well as by hobbyists and enthusiasts who are interested in cybersecurity. The software is known for its ease of use, powerful capabilities, and wide range of tools and features.

One of the key strengths of Kali Linux is its ability to handle a wide range of file systems and data formats, making it a versatile tool for digital forensics investigations. The software also includes advanced search and filtering capabilities, enabling investigators to quickly and easily locate specific pieces of data within a large dataset.

Kali Linux is maintained and supported by Offensive Security, a cybersecurity company that specializes in penetration testing and security training. The software is regularly updated and maintained by the community of developers and cybersecurity experts, ensuring that it remains a powerful and effective tool for digital forensics investigations and penetration testing.

2.9 An Overview of Utility Tools: Disk Imaging, Data Extraction, Password Cracking

A detailed description of various tools used by DF investigators to perform disk imaging, conduct investigation, decryption, password cracking and collect crucial evidence is provided herein.

Ophcrack

Ophcrack is a free and open-source utility used for password cracking on Windows systems. Its main use is to decipher the hashes produced by identical Windows files, enabling users to recover lost or forgotten passwords. It is particularly useful for cracking Windows passwords, as it can bypass many security measures and recover passwords that are typically difficult to crack.

One of the advantages of using Ophcrack is that it provides a secure GUI system that is easy to use. The software can be run on various systems and supports a wide range of Windows operating systems. Additionally, Ophcrack is capable of cracking both LM and NTLM hashes, making it a versatile tool for password cracking.

Overall, Ophcrack is a powerful utility that can be used to recover lost or forgotten passwords on Windows systems. Its ability to decipher hashes and bypass security measures make it an important tool for digital forensic investigators and other security professionals.

DataDumper

DataDumper is a free and open-source tool for computer forensics that is designed to create exact replicas of disks for digital forensic examination. It is a command-line tool that is available on the UNIX operating system.

DataDumper works by creating a sector-by-sector copy of a disk, including all the data, metadata, and file system structures. This allows investigators to create a complete and accurate copy of the original disk, which can then be used for further analysis or examination.

One of the advantages of using DataDumper is that it is a reliable and robust tool for disk imaging, capable of creating bit-for-bit copies of disks. This ensures that the copy is an exact replica of the original, which is important for maintaining the integrity of digital evidence.

Md5-Sum

Md5sum is a verification tool that is commonly used in computer forensics to confirm the integrity of data that has been copied from one storage device to another.

Md5sum generates a unique checksum for a file, which can be compared to the checksum of a copy of the file to confirm that it is identical. This enables investigators to verify that data has been successfully copied to another storage device without any errors or modifications.

One of the advantages of using md5sum is that it is a fast and efficient tool for verifying data integrity. It is also widely available on various operating systems, including Windows, macOS, and Linux, and is simple to use, making it a popular choice among digital forensics investigators and other security professionals.

2.9.1 Tools used for evidence collection and e-discovery

Earlier before the advent of integrated and dedicated digital forensics tools, it was various hacking tools that were utilised for digital forensic evidence gathering and e-discovery, which included computer programs and scripts. These tools assisted hackers in identifying and exploiting vulnerabilities in servers, networks, web applications, and computer systems. They were and still are utilised by ethical hackers to test the resilience of networks commonly referred to as penetration testing. On the other hand, these same hacking tools used in ethical-hacking to investigate vulnerabilities in organisational networks are also utilised by first responders to analyse networks and diagnose vulnerabilities.

Many hacking-tools also double up as tools for evidence collection and e-discovery. It is very important for a cybercrime investigator or first responder to have a good knowledge of various hacking tools as that give them an insight of what they can be used for and the extent of damage that they can do. Also, how they can assist in investigation by cracking encryptions and password protected files to obtain crucial evidence hidden many times intentionally to cover-up their tracks.

2.9.2 Password Cracking Tools

There is quite a possibility that files recovered from evidence extraction are password protected or encrypted. The investigators may need tools to crack the passwords in order to view the contents

of the files. Certain tools known as password cracking or recovery tools can be of immence help to the digital forensic investigators. A list of password cracking or recovery tools and their description is provided here.

Cain & Abel

Cain & Abel is a software program designed to assist with password recovery on Microsoft operating systems. Its features include retrieving MS-Access passwords, exposing password fields, analysing network traffic, and utilizing various methods such as dictionary attacks, brute-force attacks, and cryptanalysis attacks to crack encrypted passwords.

Hashcat

Hashcat is a powerful password cracking tool that is often utilized in security testing and penetration testing. It is well-regarded for its speed and flexibility, supporting various password cracking methods such as brute-force attacks, dictionary attacks, and rule-based attacks on different hash types. As an open-source software, Hashcat is available on multiple platforms, making it possible to run parallel processing across different CPU or GPU systems. It also offers advanced features like session management, hash generation, and mask files. Other notable features include a built-in benchmarking system, a thermal watchdog, and optimize performance automatically. However, like any other hacking tool, Hashcat should only be used for ethical and authorized purposes.

L0phtCrack

L0phtCrack is a useful auditing tool that can recover passwords and can identify and evaluate password vulnerabilities across networks and systems. Its main features include support for multiple GPUs and multicore processors, which help optimize hardware performance. The tool is easy to customize and offers simple password loading. Additionally, it allows for the creation of complex procedures for automating enterprise-wide password management. L0phtCrack can address weak password concerns by requiring password resets or locking accounts. It also supports auditing of multiple operating systems.

John the Ripper

Also referred to as JTR, is a commonly used hacking tool for cracking password by launching dictionary-based attacks. Its primary goal is to identify vulnerabilities related to weak passwords within a network. It is used for testing system defences using rainbow attacks and brute force attacks to assist in password cracking. One of the main advantages of JTR is that it is open source and freely available. Additionally, it supports various hash and cipher types and includes a proactive module for password strength checking. Users can browse the documentation online, includes a summary of version changes, making it easier to stay up-to-date with the latest features and improvements.

Therefore, JTR is a useful tool for identifying weak passwords in a network, and its versatility makes it a popular choice among security professionals. However, it should only be used for ethical and authorized purposes, and any misuse could result in serious legal consequences.

Rainbow Crack

Rainbow Crack is a popular tool for ethical hacking and password cracking. It uses rainbow tables to crack hashes. For this, it makes use of the time-memory trade-off technique.

This tool is primarily characterized by its suite of tools that generate rainbow tables using time-memory trade-off. It supports the creation of rainbow tables for any hash algorithm and any charset, in both compact and raw file formats (.rt).

The tool also features support for multi-core processor computation, multiple GPUs that provides GPU acceleration, and is compatible with both Linux and Windows operating systems. It has a universal rainbow table file format across all supported Operating Systems and offers both a command-line interface and a user-friendly visual interface.

IKECrack

IKECrack is a software utility that is widely used by security professionals for cracking authentication through cryptography operations. It offers both dictionary and brute-force attack capabilities, making it a top tool in its category. One of the key features of this tool also used in ethical hacking, is its ability to execute cryptography operations by providing the DH public key

and packet manipulation. This feature makes it particularly useful for cracking authentication protocols used in VPNs and other secure networks. IKECrack can be downloaded for free, both for personal and business use, making it an excellent alternative for those seeking a reliable cryptography application.

Medusa

Medusa is a powerful password cracking tool commonly used for ethical hacking purposes. It employs fast and concurrent brute-force attacks, making it a top choice for security professionals. The tool is designed as a login brute-forcer that permit remote authentication and can support a wide range of services. Medusa is also known for its massively parallel and quick system, enabling thread-based parallel testing and brute-force testing. One of the key features of Medusa is its flexibility in user input, allowing for various approaches to be used in specifying the attack. Additionally, the tool's modular design ensures that each individual '.mod' file for a service module is separate, making it easy to increase the efficacy without requiring changes to the core program for brute-forcing attacks. Medusa is a valuable addition to any security professional's toolkit, offering reliable and efficient password cracking capabilities.

THC Hydra

THC Hydra is a penetration testing tool that specializes in parallelized login cracking, offering fast and flexible unauthorized access detection for security consultants and researchers. It provides several features such as rainbow table generation, "full-time memory trade-off" tool suites, conversion, sorting, and look-up capabilities. Hydra also supports most hash algorithms, character set, and file formats, and is capable of computation on multi-core processors to crack passwords using rainbow tables. The tool is available on both Linux and Windows and uses a "unified rainbow table" file format across all Operating Systems that it supports. For user convenience, Hydra supports both GUI (Graphical User Interface) as well as and command-line interfaces.

2.9.3 Network Monitoring and Diagnostics Tools

Zenmap

Zenmap provides a GUI interface to Nmap, which is a popular security scanning application. It is open-source and freely downloadable, multi-platform program that is user-friendly for beginners and offers advanced features for experienced users. This tool provides a simple view of information on a specific host or a full network scan. It is an ideal tool for administrators to monitor new services or hosts that join their networks and to keep track of any declining services.

The main features of Zenmap include interactive and graphical results visualization, the ability to create a topology map of 'found' networks, and the ability to highlight differences between two scans.

PRTG

PRTG is a tool used for monitoring networks that is highly regarded for its advanced features for management and monitoring of infrastructure. It uses multiple technologies, including WMI, Sniffing, SNMP, SQL and REST APIs, to monitor IT infrastructure effectively.

One of the key features of PRTG is its ability to scan network segments by pinging specified IP ranges. It also enables users to create customized dashboards using the latest monitoring data and a choice of layouts. Its alerting system is simple and flexible, and it provides several user interfaces to choose from. Additionally, PRTG notifies users when it detects any anomalies or unusual behaviour in the network traffic.

Traceroute NG

Traceroute NG is a useful tool for network administrators and analysts who need to troubleshoot network connectivity issues, identify potential bottlenecks, and optimize network performance. The software provides detailed information about the network path taken by packets, including the number of hops, the round-trip time, and the routers or switches involved.

The ability to create a text log file enables network administrators to store and review historical data, which can help in identifying patterns and trends. Additionally, Traceroute NG can help in identifying routing loops, misconfigured routers, and other network anomalies.

Traceroute NG's continuous probing feature can be useful for monitoring network performance over an extended period. By regularly probing the network and comparing the results, administrators can quickly identify changes in network behaviour and take appropriate action.

OWASP HTTP POST

The OWASP stands for “Open Web Application Security Project”. OWASP HTTP Post software is a tool that assists in testing the network performance of web applications. It can simulate online denial of service attacks from a single DDoS machine, allowing users to test their web applications' ability to withstand such attacks.

One of the key features of this tool is that it is available for commercial use without charge, making it accessible to a wide range of users. Additionally, the tool allows sharing of results under its license, enabling users to collaborate and share information with others. The tool is particularly useful for testing against application layer attacks and assists in determining server’s capacity to handle such attacks.

2.9.4 Penetration Testing (Pen-Test) Tools

Penetration testing tools, also referred to as Pen-Test tools are designed to help identify and detect security weaknesses within a network, server, or web application. These tools simulate hacker-like attacks to assess the system's security posture and identify potential vulnerabilities. This information is then used to address any security gaps and prevent potential security breaches.

VAPT

VAPT stands for “Vulnerability Assessment and Penetration Testing”, and these tools simulate hacker-like attacks to assess the system's security posture. Any potential security gaps found during the assessment are then addressed. The list provided here, comprises the top penetration testing (pen-test) tools that include both free and open source or paid and commercially used

software. Each tool has been detailed with a brief description of its most popular features and resources.

Metaspoilt

Metasploit is a powerful and popular framework for vulnerability and pen-testing, which is based on the concept of 'exploits' - passing a code that breaches security measures and enters a target system. Once entered, it runs a 'payload' - a code that performs operations on the target machine. This creates the perfect framework for penetration testing and is also an excellent tool to test whether an IDS can successfully prevent attacks when bypassed.

Metasploit can be used on various targets, such as networks, applications, and servers. It features a command line interface and a GUI clickable interface and is compatible with Apple Mac OS X, Linux, and Microsoft Windows. Some of its main features include a basic command line interface, third-party import, manual brute forcing, and website penetration testing.

Kali Linux Distro

Kali Linux is a top-rated distro and pen-testing tool collection that allows you to customize your backup and recovery schedule according to your preferences. This software provides an efficient way to access and update the largest security penetration testing database available. It is particularly useful for packet sniffing and SQL injection testing, and requires expertise in TCP/IP protocol and networking. Kali Linux is the successor to the popular BackTrack platform and is exclusively compatible with Linux machines.

The key features of Kali (as well as Back Track) include, support for 64-bit architecture, which enables brute force password cracking. It offers pre-installed tools for LAN and WLAN sniffing, vulnerability scanning, password cracking, and digital forensics.

It integrates with popular tools such as Metaspoilt and Wireshark, and also includes non-networking tools like pidgin, xmms, Mozilla, k3b, etc. It is compatible with both KDE and Gnome desktop environments.

Acunetix

Acunetix is an automated web application security testing tool used for penetration testing. It can accurately scan complex JavaScript, HTML5 and single-page applications and can audit web applications that have been authenticated to generate management and compliance reports for various network and web vulnerabilities.

This tool can detect common web application vulnerabilities such as Cross-site scripting (XSS), SQL Injection and more than 4500 different vulnerabilities. Acunetix is fast and scalable, capable of crawling through hundreds of thousands of web-pages continuously. It integrates with popular WAFs (Web Application Firewalls) and issue trackers to aid in the SDLC (Software Development Lifecycle) and is available as both a cloud solution and on-premises. Additionally, it can detect over 1200 vulnerabilities in WordPress core, plugins, and themes.

2.9.5 Web Security & Vulnerability Testing Frameworks

OWASP Framework

The Open Web Application Security Project (OWASP) is a non-profit organization with a global reach focused on enhancing software security. The project offers multiple pen-testing tools for various software environments, protocols and web applications.

The flagship tools of the project include the ZAP (Zed Attack Proxy), an integrated penetration testing tool; the OWASP Web Testing Environment Project, which is a collection of security tools and documentation and the OWASP Dependency Check, which scans for project dependencies and checks against known vulnerabilities.

Burp Suite

Burp Suite is a popular and powerful platform for web application security testing and penetration testing. Its various tools work together to facilitate the pen-testing process, from mapping the attack surface of an application to identifying vulnerabilities and exploiting them.

One of the key strengths of Burp Suite is its ability to detect over 3000 web application vulnerabilities, making it one of the most comprehensive security testing tools available. It can

examine both custom-built and open-source software, allowing it to identify vulnerabilities in a wide range of web applications.

The Login Sequence Recorder feature makes automatic scanning simple, allowing users to easily record and replay login sequences for authenticated testing. Burp Suite also offers integrated vulnerability management, allowing users to analyse vulnerability data and generate technical and compliance reports quickly and easily.

Burp Suite provides 100% accurate critical vulnerability detection and includes advanced scanning logic to ensure that even complex vulnerabilities are identified. Its automation features for crawling and scanning allow users to quickly and easily identify potential vulnerabilities and threats.

Samurai Web Testing Framework

This Web Testing Framework is a robust pen-testing software developed as a web penetration testing environment. It is designed to be run on VMWare and VirtualBox and comes pre-configured with all the necessary tools for web testing and attacking. This open-source tool is free to use and includes some of the best free and open-source tools for website testing. Moreover, it comes with a pre-configured wiki that can be used to set up a central information store during the pen-test, providing users with an all-in-one solution for web pen-testing.

W3af Audit Framework

The W3af Audit Framework is a powerful tool designed for auditing and attacking web applications. It is comprised of three types of tools as plugins, including discovery, audit, and attack, which work together to identify vulnerabilities within the target site. For instance, the discovery plugin in W3af is responsible for searching for different URLs to test for any hidden vulnerabilities and passing the results to the audit plugin, which uses the identified URLs to find any security weaknesses.

In addition to its auditing and attacking capabilities, W3af can also function as a Man-in-The-Middle (MITM) proxy. This feature intercepts requests, sends them to the request generator, when

then enables manual testing of web applications using defined variable parameters. The framework also has features to exploit any vulnerabilities that it discovers.

W3af offers several notable features, including the ability to add custom headers to requests, HTTP basic and digest authentication, support for proxy, HTTP response and DNS caching, uploading files using multipart, user agent spoofing and handling cookies. With its wide range of capabilities, W3af is an essential tool for any web application security testing.

OpenSSL

OpenSSL is a free and open-source project that offers a comprehensive toolkit for the SSL and TLS protocols. The project is licensed under an Apache-style license agreement and primarily written in C, but it also has wrappers for various programming languages. Its toolkit includes several tools, such as RSA private key and "Certificate Signing Request" or CSR generators, as well as the ability to create new private keys and CSRs and remove passphrases from keys. It also supports CSR file verification.

2.9.6 Intrusion Detection and Prevention Tools

The Intrusion Detection System Tools allow administrators to detect and identify various advanced threats effectively. This tool also facilitates compliance reporting for DSS and HIPAA. It continuously monitors suspicious attacks and activities to provide you with real-time logs. The primary features of this tool include reducing intrusion detection efforts, ensuring compliance through efficient reporting, and detecting malicious IPs, applications, accounts, and other potential threats.

The Intrusion Prevention System (IPS) tools are designed to prevent external intrusions and safeguard your network against known, unknown, and undisclosed vulnerabilities. These tools offer automated and inline inspections, ensuring real-time protection and providing proven network reliability and availability.

IPS tools offer several key features that make them highly effective for network security management. These tools are designed to provide centralized management, enabling security policy integration and prioritization, as well as enhanced visibility and response capabilities. IPS

tools utilize “patented machine learning” techniques to maximize real-time protection, ensuring that all known vulnerabilities and attack permutations are guarded against with minimal false positives. Additionally, IPS tools offer a policy-based operational model that enhances scalability and integrated security, allowing for fully automated and real-time protection. Therefore, IPS tools are highly capable network security solutions that can deliver comprehensive protection against a wide range of potential threats.

Security Onion

Security Onion is a software application designed to detect intrusion and monitor network security during penetration testing. Its user-friendly Setup wizard makes it easy for users to create a network of sensors across their organization. Security Onion offers several key features, such as a “distributed client-server” model, full packet capture, and both host-based and network-based Intrusion Detection Systems. It also includes a Network Security Monitoring system to observe security-related events and has an automatic data purging function to prevent storage devices from reaching their capacity.

SolarWinds Security Event Manager

This is a software utility that is designed to improve computer security by monitoring security protocols, automatically detecting threats, and protecting your network. This tool enables you to easily track log files and receive quick alerts in case of any abnormal activity.

The primary features of SolarWinds Security Event Manager include integrity monitoring for network security, control over memory stick storage, a user-friendly dashboard and interface, tools for integrated compliance reporting, a centralised log collection, and fast hazard detection and response capabilities.

Sucuri

Sucuri's technology applications are designed to defend against DDoS attacks using cutting-edge tools such as “Web Application Firewall” (WAF) and “Intrusion Prevention System” (IPS). By constantly monitoring website traffic and rankings, Sucuri improves website performance while

ensuring its safety. This tool can block DDoS attacks at layers 3, 4, and 7, while providing malware and hack protection through a Web Application Firewall (WAF).

The main features of Sucuri include: continuous monitoring of patch updates and server rules to protect your website; a Protect Page Feature that allows users to add passwords, CAPTCHA, 2FA, and other security measures to sensitive pages; and easy set-up using only web server credentials and DNS changes.

2.9.7 Digital Forensic Tools used for Live Forensics

LiveAction

One of the best 'Live Forensics' and ethical hacking tools available, it offers extensive visibility that can help fix performance problems and reduce security risks. It is regarded as one of the most powerful hacking programs and uses 'LiveAction' packet intelligence to provide faster and more accurate analysis of network issues.

This tool provides excellent network forensics solutions and offers several noteworthy features, including effective and user-friendly software, automated data collection with 'LiveAction' for speedy security alarm assessment, compatibility with various computer programs and appliances, deep analysis capabilities with packet intelligence, fast resolution of network and security issues, logical workflow that is easy to use, professional technical assistance, and on-site deployment as an appliance.

QualysGuard

Businesses may streamline their security and compliance solutions with the aid of QualysGuard. Additionally, it incorporates security into their plans for digital transformation. One of the greatest tools for detecting hacker activity when it comes to online cloud services.

This online hacking tool is widely considered as one of the best, and trusted by users worldwide. It eliminates the need to purchase or handle any hardware, making it a hassle-free solution. With its end-to-end scalability, this software can address various IT security aspects. The data it collects regarding vulnerabilities is stored securely and processed on a load-balanced server architecture

with multiple tiers. Thanks to its sensor, it can provide continuous visibility, and the data it collects can go through real time analysis. It can also attend to real time threats without any latency.

2.9.8 File Decryption Tools

EnCase includes a range of capabilities for decrypting encrypted data. It can work with a wide range of file formats. FTK Imager includes a decryption capability that can be used to recover encrypted data. Oxygen Forensic Suite is a commercial forensic tool that supports decryption of various types of encrypted data. It can decrypt data from a variety of mobile devices, as well as computers and other storage devices.

Passware Kit is a commercial tool that specializes in password recovery and decryption of various types of files, including encrypted files. It supports a wide range of encryption algorithms and can work with various file formats. Elcomsoft Forensic Disk Decryptor is another commercial forensic tool that can be used to decrypt disks and volumes encrypted with various encryption algorithms. It supports TrueCrypt, VeraCrypt, BitLocker, and other popular encryption tools.

2.10 A Comparative Analysis of Most Popular Digital Forensics Tools

Since most of the digital forensic tools mentioned earlier in section 2.8, have been developed on ad hoc basis and are well suited for different DF investigation methodology, therefore they do not strictly follow any framework guidelines. These framework guidelines were developed by researchers and practitioners whereas the forensic tools developers were either commercial organisations or open source developers.

Moreover, investigations are conducted in so widely diverse technical environments that vary with every operating system, file system and software requirements; it has led to ad hoc development in digital forensic tools that lacks uniformity and conformance to common acceptable investigation framework standards developed by researchers and practitioners. It has been investigated that present frameworks demonstrate lack of instruments for measuring efficacy of DF tools as well as interventions are required to standardise the DF investigation tools to confirm to established DF frameworks. Hence there is a requirement to develop an efficacy assessment framework that addresses the challenges broadly.

2.11 Challenges faced by Digital Forensic Investigators in using current Frameworks and Tools

After doing detailed review of various DF frameworks, models and tools under sections 2.4 to 2.9, and doing a further analysis of most popular tools in section 2.10, researcher has identified three main categories (themes) that can be posed as major challenges that intimidate the first respondents/investigators as described herein.

2.11.1 Technological Challenges

The expanding volume of data generated has resulted in the creation of more advanced and larger storage drives in computers. These drives are vital for individuals and businesses to store and process data. As the demand for data storage grows, it is probable that we will witness further developments in this domain to ensure that storage solutions keep pace with the escalating demand for data processing and storage capacity. We have seen that large volumes of data makes it hard to complete investigation in limited time-frames. Others are encryption of data; use of anonymous technology (websites using onion routing, tor browser) and cloud computing also poses huge challenge (Garfinkel, 2010). The challenges posed by advancements in technology and added sophistication in the utilisation of various techniques by the cyber criminals are categorised herein.

Growing Data and Storage Drives in Computers

In today's digital age, data is being generated at an unprecedented rate. With the proliferation of smartphones and social media, cloud computing services, and the IoT (Internet of Things), the amount of data being created each day is staggering. As a result, storage drives in computers have to keep up with the growing demand for storage capacity.

The first hard disk drives (HDDs) were introduced in the 1950s and could store just a few megabytes of data. Over time, their capacity increased, and by the 1990s, they could store several gigabytes. In the early 2000s, solid-state drives (SSDs) were introduced, offering faster read/write speeds and more reliable performance than HDDs. However, their storage capacity was limited and expensive, making them impractical for most consumers.

Today, both HDDs and SSDs have continued to evolve, offering larger storage capacities at increasingly affordable prices. HDDs can now store up to 18TB of data, while SSDs can store up to 16TB. These drives are commonly used in desktop and laptop computers, as well as servers and data centres. Cloud storage has also become a popular solution for storing data, with services such as Dropbox, Google Drive, and iCloud offering users the ability to store and access files from anywhere with an internet connection. These services are also popular among businesses, as they offer scalable storage solutions that can be accessed by employees from anywhere in the world.

Despite the increase in storage capacity, the growing amount of data being created still poses a challenge. The rise of big data, machine learning, and artificial intelligence has led to an explosion in data processing requirements. This has resulted in the development of specialized storage drives such as solid-state hybrid drives (SSHDs) and Non-Volatile Memory Express (NVMe) drives, which offer faster read/write speeds and reduced latency. Most high end commercial or proprietary tools also struggle to handle huge drives and cloud technology offers even more hurdles to run investigations in these changing paradigms.

Use of Encryption Technologies

Encryption technologies poses a big challenge to cyber-forensic investigators as cyber criminals are using sophisticated technologies to hide digital evidence in encrypted files and drives. We have discussed DF tools that can decrypt files which contain encrypted data and crack encryption passwords (e.g. EnCase, FTK Imager, Aircrack, Ettercap, Hashcat, Cain & Abel, Oxygen. Passware Kit etc.) under sections 2.8 and 2.9. Cyber forensic investigators use EnCase, FTK Imager, Oxygen Forensic Suite, Passware Kit, and Elcomsoft Forensic Disk Decryptor to decrypt encrypted data.

EnCase has a broad range of decryption capabilities and file format compatibility, while FTK Imager is equipped with a decryption feature to recover encrypted data. Oxygen Forensic Suite is a commercial forensic tool that decrypts various types of encrypted data from mobile devices, computers, and storage devices. Passware Kit specializes in password recovery and decryption of various file formats, including encrypted files, and works with a wide range of encryption algorithms. Elcomsoft Forensic Disk Decryptor can decrypt disks and volumes encrypted with popular encryption tools like TrueCrypt, VeraCrypt, and BitLocker. Still, encryption algorithms

keep on improving and require more powerful computing resources and decryption techniques. No tool can be guaranteed to be effective in breaking a strong 256-bit encryption used by ransomwares and other actors like cyberterrorists quite often these days. Therefore while quite useful, the fast improvements in encryption technology and it's widespread use in devices and applications is posing a constant challenge to DF investigators.

2.11.2 Methodological challenges

Methodical challenges arise when a DF investigation methodology is not easily adoptable or is too complex to implement. It is evident in discussion under sections 2.4, 2.5 and 2.6 that lack of acquaintance of investigation/process frameworks and protocols poses a major challenge.

Absence of clearly defined methodological process or investigation protocol can lead to poor investigations and data of poor standard that lacks integrity and reliability.

Improper investigation tools selection or lack of proper knowledge of using the technology in a methodical manner as described in details in section 2.8, also creates inefficiencies. This is further emphasised by Mocas (2004) and Garfinkel (2010) and therefore more leverage on cutting edge technologies like AI and machine learning is required in order to overcome the disadvantages faced by DF investigators.

2.11.3 Legal Challenges

Lack of knowledge or expertise on legal procedures, cyber-laws; warrant process etc. makes the evidence collection difficult as discussed in section 2.5, 2.6 and 2.7. Also, poor quality of investigation and documentation process may not stand the legal scrutiny. These challenges were further highlighted by Collie (2018), Institute of Justice (2008), Schultz (2016) and Snail (2009) highlights legal challenges in South African context.

2.12 Discussion

It has been noted that there are several digital forensics tools available, each developed by various organizations and industry groups with their own specific objectives. While these tools are driving technical advancements in the field of digital forensics, they also highlight the ad hoc manner in

which the field currently operates. This has been emphasised by Garfinkel (2010) and also evident by the analysis done in section 2.10.

The rapid advancement of technology has presented several challenges for conducting investigations effectively in digital forensics domain. It has been noted that emergence of new technologies like cloud computing, encryption-based messaging tools, anonymous technology (tor onion routing), has made evidence collection more difficult, while the proliferation of the Internet of Things (IoT) has introduced new gadgets and storage forms. Additionally, the increasing number of digital devices connected to the internet has led to the generation of vast amounts of data, further complicating investigations. To address these challenges, there is a growing focus on integrating additional tools and technologies such as data mining and Artificial Intelligence (AI). These novel concepts can help to build better solutions that can support investigations and mitigate the concerns arising from these technological advancements, as noted by Baggili et al. (2013) and Jusas et al. (2017).

2.13 Major Gaps Identified

To conclude, the researcher has identified three main challenges that intimidate first responders/investigators in digital forensics: technological (such as large volumes of data, encryption, and anonymous technology), legal (such as lack of knowledge of legal requirements and procedures in evidence gathering and cyber-laws), and methodological (such as lack of acquaintance of investigation/process model and investigative tools selection). Novel and standard technologies present different complications for digital forensics investigations, including cloud computing, Internet-of-Things, and huge volume of data that is always growing. Recent investigation on “process models” did reveal some efforts that focuses on adding new tools or updating technologies, such as “data mining” to support novel models and frameworks in order to resolve concerns triggered by new technologies, but there are substantial gaps in these attempts, as highlighted under sections 2.4, 2.5, 2.7 and 2.8 respectively. Therefore, this research lay emphasis on filling these gaps by development of new interventions in the form of a new digital forensic protocol with a newly developed AI (Artificial Intelligence) framework, which can perform prediction analysis and enhance decision making capabilities of cybercrime investigators.

Section 2.8 along with previously discussed sections 2.4, 2.5, 2.6 and 2.7 addresses research sub-question-5 and provides a motivation towards this research to fill the gaps identified in this chapter.

2.14 Summary

This chapter begins by describing various fundamental concepts in digital forensic domain. It describes in detail that there have been various digital forensic investigation models and frameworks, which adhere to different methodologies and are useful in different types of cybercrime investigations. Also, there are a vast array of DF investigation tools catering to different investigation requirements, technologies (Operating Systems, file systems, mobile devices etc.). But these tools lack standardisation and fails to represent most accepted DF process and frameworks. This affects the reliability of forensic tools and their efficacy. The key challenges faced by first responders/cybercrime investigators are technological, methodological and legal in nature. Several new and established technologies can create obstacles in digital forensics investigations process. Cloud computing, for example, can make 'evidence collection' more difficult, while the 'Internet-of-Things' or IoT introduces a whole new range of innovative gadgets and the data stored in various formats take challenges to another dimension, leading to increased data volumes and complexities in evidence collection. Additionally, the growing number of digital devices that are always connected to the internet further exacerbates this issue. These challenges can be overcome by introducing better frameworks that are more implementable and use of cutting-edge technologies like Artificial Intelligence (AI) techniques which can perform prediction analysis and enhance decision making capabilities of cybercrime investigators.

CHAPTER 3: RESEARCH METHODOLOGY

3.1 Introduction

This chapter entails a discussion on various research philosophies and methodologies used in the field of Information Systems and Technology (IS&T) and describes the most suitable research methodology adopted for this research. Research Methodology entails theoretical or conceptual framework that guides the study and warrants for the research plan ideal for the study. Many experts have given their own elucidation to this concept, some of them needs a more elaborative description here.

Research methodology (RM) refers to the systematic and scientific approach taken by researchers to investigate a particular phenomenon or problem. It involves the use of different techniques and procedures to collect, analyse, and interpret data in a way that helps to answer research questions and achieve research objectives (Saunders & Tosey, 2013). A sound research methodology is critical for producing reliable and valid research results that can be used to advance knowledge and inform decision-making. It enables researchers to design studies that are appropriate for their research questions and to use appropriate methods to collect and analyse data (Van Wyk, 2012; Walliman, (2021).

There are many different research methodologies, each with its own strengths and weaknesses. Quantitative research methodologies involve the use of numerical data, statistics, and mathematical models to analyse and interpret data. Qualitative research methodologies, on the other hand, involve the collection and analysis of non-numerical data, such as interviews, observations, and case studies (Queirós et al., 2017).

In addition to quantitative and qualitative research methodologies, there are also mixed-methods research methodologies that combine both approaches. These methodologies are becoming increasingly popular as they offer the advantages of both quantitative and qualitative research methods, allowing researchers to triangulate data and gain a more comprehensive understanding of the phenomenon being studied (Creswell, 2021).

Design Science Research (DSR) methodology offers an iterative approach that combines problem-solving and creative design to create new artefacts that solve complex problems in a particular domain. DSR is commonly used in information systems research, where it has proven to be an effective method for developing and evaluating Information Systems and Technologies. DSR involves the development of a rigorous research framework, followed by the design, implementation, and evaluation of an artifact (Hevner et al., 2004). The artifact is then tested, refined, and validated through a series of iterations until a satisfactory solution is obtained. The goal of DSR is to produce new knowledge in the form of an artifact that can be used to improve real-world systems and processes (Peffers et al., 2007).

Since the main objective of this research is to 'design and develop an AI based Digital Forensic protocol for first responders', this research is based on design science principles as it endeavours to understand and improve over the body of knowledge in the field of Digital Forensics. In order to create AI based interventions that could help first responders in cyber-investigations, the Design Science Research (DSR) approach has been streamlined to achieve the objectives of the research.

Addressing the research questions and research objectives, this chapter outlines the research process and procedures, including the research philosophy, research approach, research strategy, research design, and various techniques and procedures. These details are explained in forthcoming sections of the chapter.

3.2 Design Science Research (DSR)

Design Science is considered a research approach that is an outcome-based methodology where the outcomes are newly developed artefacts. Research projects using this approach follow specific guidelines for evaluations and iterations. Design Science approach has been quite extensively used in designing Information Systems focussing on Human/Computer Interaction (HCI), process models, algorithms etc. Conceptually this methodology inculcate Function-Behaviour-Structure (FBS) ontology propagated by (GERO & KANNENGIESSER (2006). The ontology of DSR is

evolutionary and complementary whereas epistemological process can be described as "….knowing through making" (Purao, 2013).

Purao et al. (2013) further emphasises on the knowledge creation and theory building aspects of DSR at three different levels of abstraction in figure 3.1.

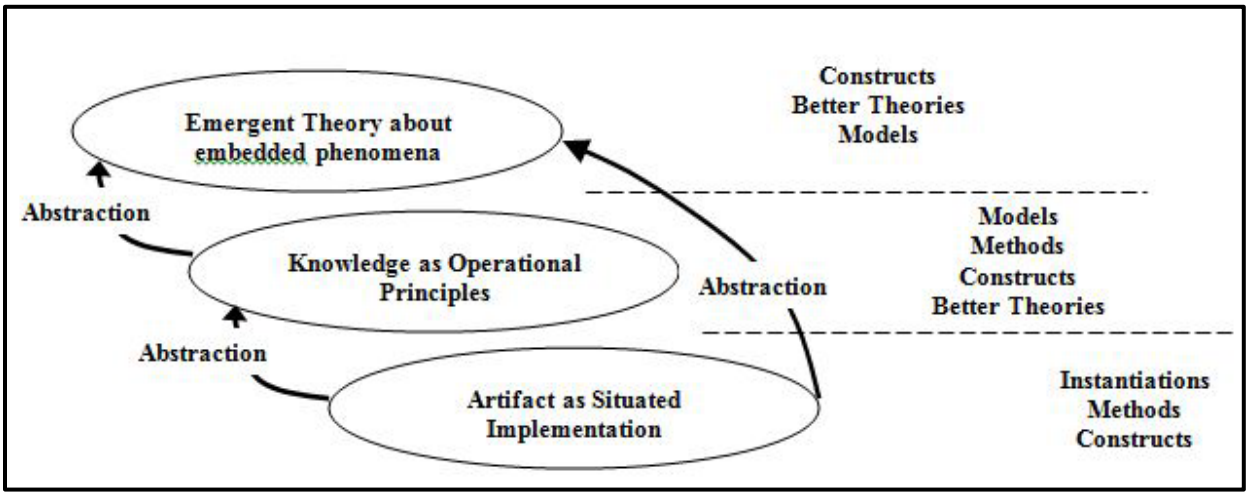


Figure 3.1: Outputs of Design Science Research (Purao, 2013).

Design Science Research (DSR), has been used in many disciplines but its use to IS discipline is ever more justified. Kuechler and Vaishnavi (2008) describe Design Science Research (DSR) methodology in the context of Information Systems research. Design Science Research is a research paradigm that focuses on the creation of innovative artefacts, such as models, methods, processes, and systems, to address and solve complex problems in a specific domain.

Some key points emphasised are "Problem Identification". DSR starts with the identification of a significant and relevant problem in a particular domain. The research process is driven by the need to develop a solution to this problem. DSR emphasizes "Artifact Creation", the creation of novel artefacts that provide solutions to real-world problems. These artefacts can take various forms, including software systems, frameworks, models, or methodologies.

DSR involves an iterative process where the researcher iteratively designs, develops, and refines the artifact. Each iteration contributes to the improvement and enhancement of the artifact. The research outcomes in DSR are often presented as problem-solution pairs. Researchers articulate the problem, present the designed artifact as a solution, and evaluate its effectiveness in addressing

the identified problem. The artefacts should be practically applicable and address real-world problems while also adhering to rigorous research principles.

The developed artifact is subjected to “Evaluation”. This evaluation can take various forms, such as performance testing, usability studies, or comparisons with existing solutions. The goal is to assess the effectiveness and efficiency of the artifact. DSR researchers “communicate” their findings not only through traditional research papers but also through the presentation of the developed artifacts. This includes detailed documentation and guidelines for implementing and using the artifact. DSR contributes to cumulative knowledge by building on existing theories and best practices. The artefacts developed in DSR projects can serve as a foundation for future research in the same or related domains. Figure 3.2 describes DSR cycle proposed by Kuechler and Vaishnavi (2008).

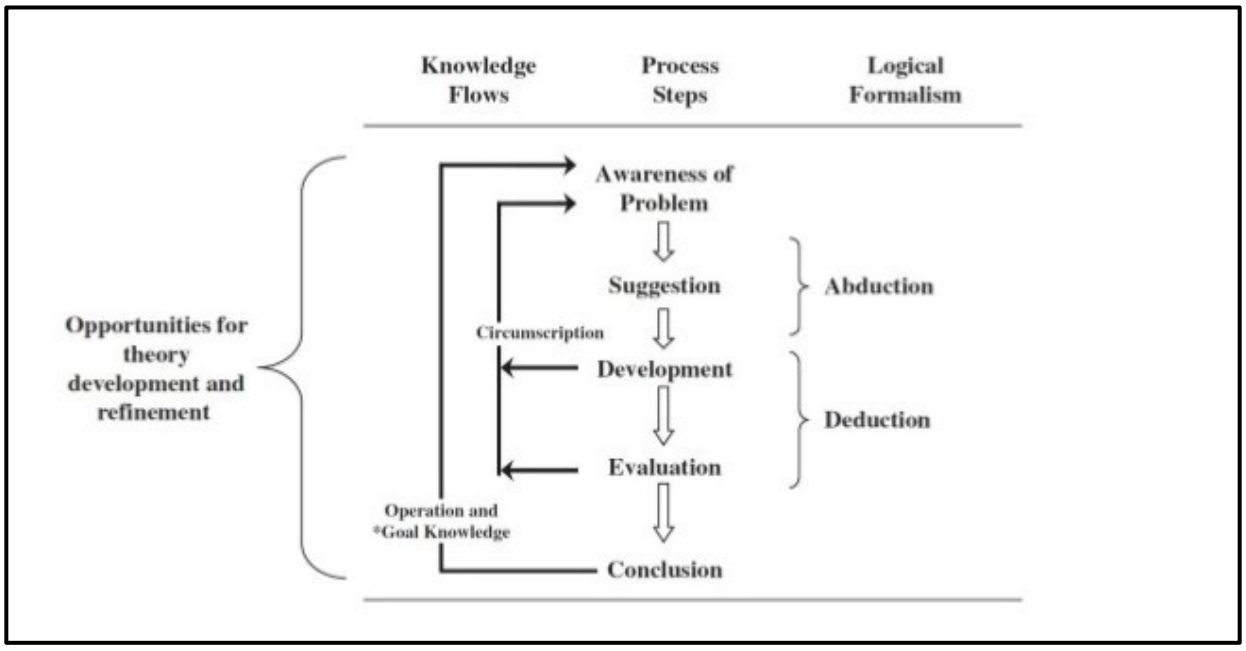


Figure 3.2: Design Science Research Cycle (Kuechler & Vaishnavi, 2008).

Peffers et al. (2007) offers a comprehensive and practical methodology for conducting Design Science Research (DSR) in Information Systems (IS) research. Peffers et al. (2007) emphasizes the need for a rigorous approach to DSR that balances creative design and problem-solving with sound research principles. The methodology proposed by Peffers et al. (2007) comprises six stages: problem identification, objective setting, design and development, demonstration, evaluation, and communication. Each stage involves a set of activities that should be completed to achieve the

desired outcome. For example, in the problem identification stage, the researcher should identify the problem or opportunity that the research aims to address, and then formulate research questions and objectives that align with this problem or opportunity.

According to Peffers et al. (2007), main objective of DSR methodology is to create a mental model. Mental models are analogous to an architect's model or a physical diagram and represents a "small scale model of reality …. [that] can be constructed from perception imagination or the comprehension of discourse" (Peffers et al., 2007). Figure 3.3 represents different phases of DSRM process model.

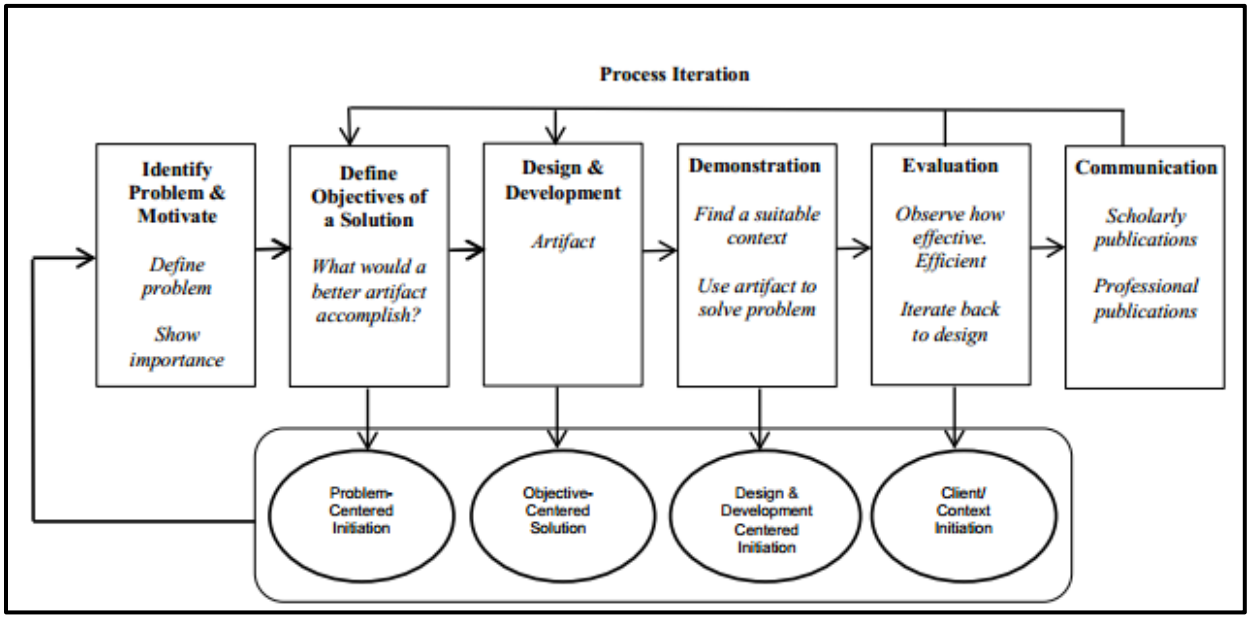


Figure 3.3: Design Science Research Methodology Process Model (Peffers et al., 2007).

3.3 Research Design

Design Science Research Model by Peffers et al. (2007) has been adopted as guiding methodology for this research. The model adopted and stages used in this research are presented in figure-3.4 and illustrates the flow of processes and specific outcomes (artifacts) expected.

The first stage is 'Identify Problem and Motivate' which has been explained in section 1.3. After the problem has been identified, the second stage 'Define Objectives of a Solution' are described under section 1.6. The main objective of the research is to create a solution (a novel protocol) that

is practically more implementable and can assist first responders in digital forensic investigation process as well as decision making.

Chapter-2 also provided extensive reviews of existing DF models/frameworks and proposed new modifications or extensions of existing ones. Chapter-4 proposed modifications in Incident Response (IR) process and emphasised on making IR process more 'agile'.

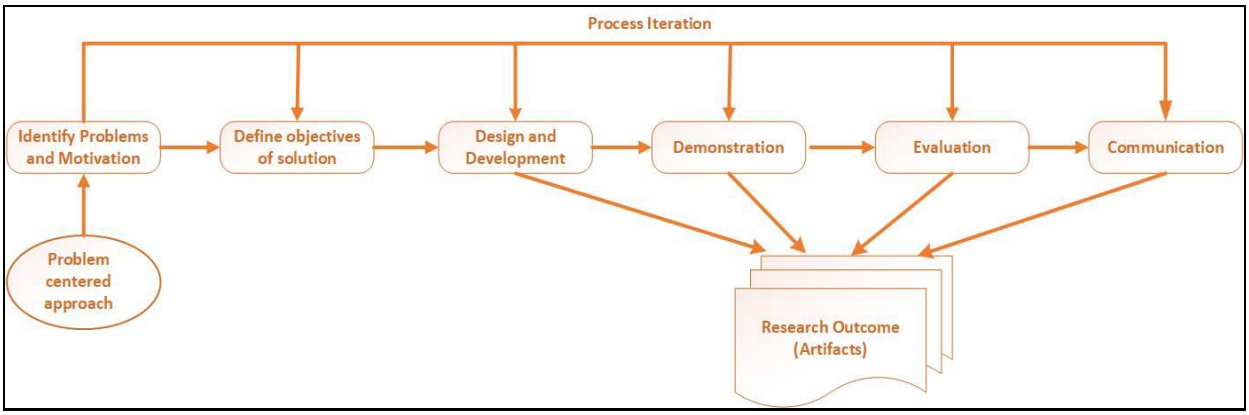


Figure 3.4: Design Science Research Methodology Model adapted for the research

The third stage leads to 'Design and Development' of new artifact, which is 'designing an AI based protocol, a novel Digital Forensic investigation protocol in this study. The motivation and need for this novel protocol have been established by critiquing the existing protocol/framework models under section 2.4. This process has been described and the design aspects explained in chapter-5 in extensive detail.

The fourth stage is 'Demonstration', which caters to demonstrate the implementation of novel protocol conceptual model into a working prototype, which is covered in chapter-6. Since the main objective of the research is to improve the efficacy of the digital forensic investigation process and assist first responders in decision making through predictive analysis of cybercrime incidents using Artificial Intelligence (AI), a novel intelligent framework is also developed and demonstrated in chapters-7 and 8 subsequently.

The fifth stage is 'Evaluation' of the artifacts, that has been designed and developed as described under chapters-5, 6, 7 and 8, are evaluated in chapter-9 and results presented. An experimental

setup is designed to perform evaluation of the artefacts developed, which constituted four experiments initially (section-5.7) and then extended to seven later in next iteration to further analyse the performance of the artifacts with larger datasets, as discussed in chapter-9. Final stage 'Communication' leads to publication of the research work in academic/industry centric journals.

This study followed DSR methodology to design and develop new artifacts based on the shortcomings found in literature review done in chapter-2. Since a detailed analysis of frameworks, standards and protocol models and forensic tools was carried in chapter-2, under sections 2.4 to 2.8, this led to identification of gaps in existing frameworks and models.

The analysis resulted into design and development of new and refined (extended) models like the 'Extended Nucleus of Digital Forensic Research' (DFRWS, 2001) described in figure 2.7, and 'Extended Linear Process Model' derived from Linear Process Model of DFRWS (2001) is represented in figure 2.8.

The new artefacts have been designed and developed and explained under forthcoming chapters. These new artifacts consist of a novel investigation protocol named as Intelligent-Digital Evidence Extraction Protocol (I-DEEP), an intelligent framework, named as Digital Intelligent Forensic Framework (DIF^2) and a working prototype based on the new protocol designed. Also, there were improvements made to existing models that were presented as 'extended models' and formed the theoretical underpinnings of new artifacts (I-DEEP and DIF^2).

3.4 Research Outcome Planned

This research intends to improve upon existing protocol models and also develop an 'AI based Digital Forensic Protocol' and an 'Application Prototype' that implements AI technology and newly developed protocol. The novel protocol adheres to the established fundamentals of digital forensic (DF) domain and the new system (prototype) is highly scalable application with an intuitive UI (User Interface). The system envisaged can assist in cybercrime diagnosis as well as provide guidance the cyber-crime investigators to take prompt action.

The new developed system thus performs the role of a 'Decision Support System'. Following the philosophy of DSR Approach**,** the research takes some guidance from AGDC model (Aggregate General Design Cycle) proposed by Kuechler & Vaishnavi (2008) and the outcomes further

mapped to Design Science Research Methodology Model (Peffers et al., 2007). The research outcomes are illustrated by the newly created research design represented in figure 3.5.

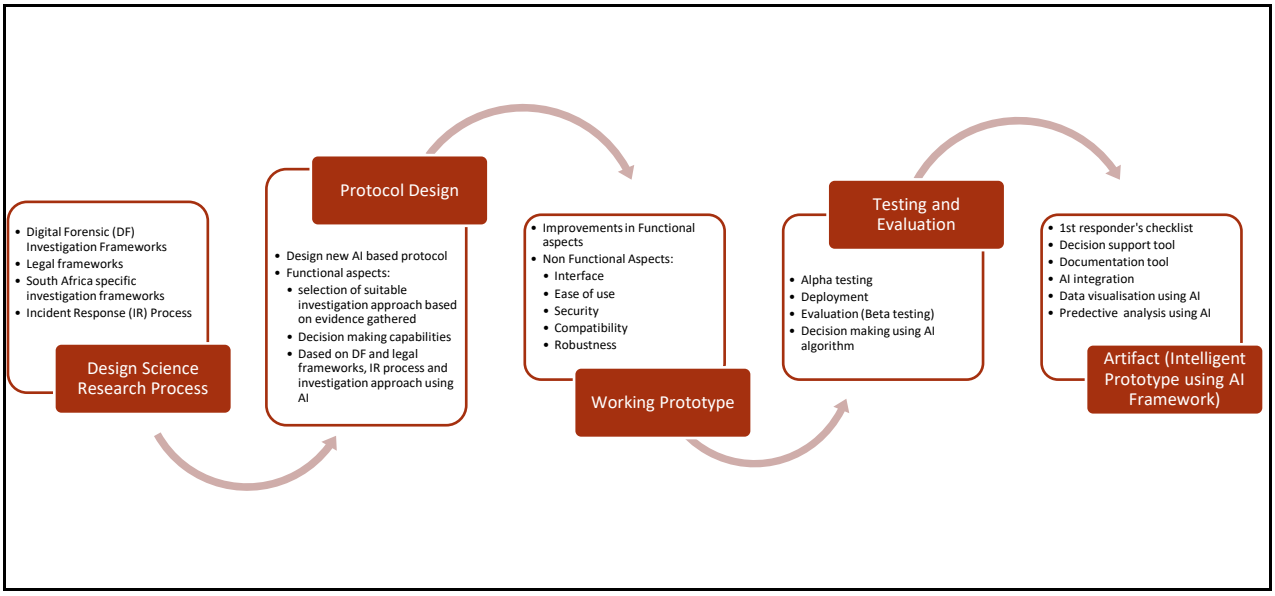


Figure 3.5: Research Design based on DSR Methodology developed for this research

Figure 3.5 shows conceptualised model of the research design, which finally result in the design and development of the artifacts- a newly designed protocol (I-DEEP) and a working prototype tool based on the newly designed protocol. This prototype is further integrated with an intelligent framework (DIF^2) which uses AI technology to provide data visualisations and predictive analysis functionalities.

A more detailed illustration of the research design is provided in figure-3.6 that guides this research and provides mapping to the DSR Model developed for this research based on Design Science Research Methodology (Peffers et al., 2007).

Figure 3.6: Research Design Flow developed to implement research goals by researcher

The 'Research Design Flow' shown in figure 3.6 shows the mapping with the DSR model adopted for this research (Peffers et al., 2007) and elaborates on the various stages of the research design

process flow and the artifacts developed at different stages. These stages and research process flow is also mapped to the research design detailed in figure-3.5 and guides the research into a methodical structure that also maps to the adopted DSR Methodology Model adopted for this research and 'Research Objectives' defined in section 1.5. The 'Research Design Flow' described in figure-3.6 provides objective mapping of the 'Research Design' and the research objectives in detail.

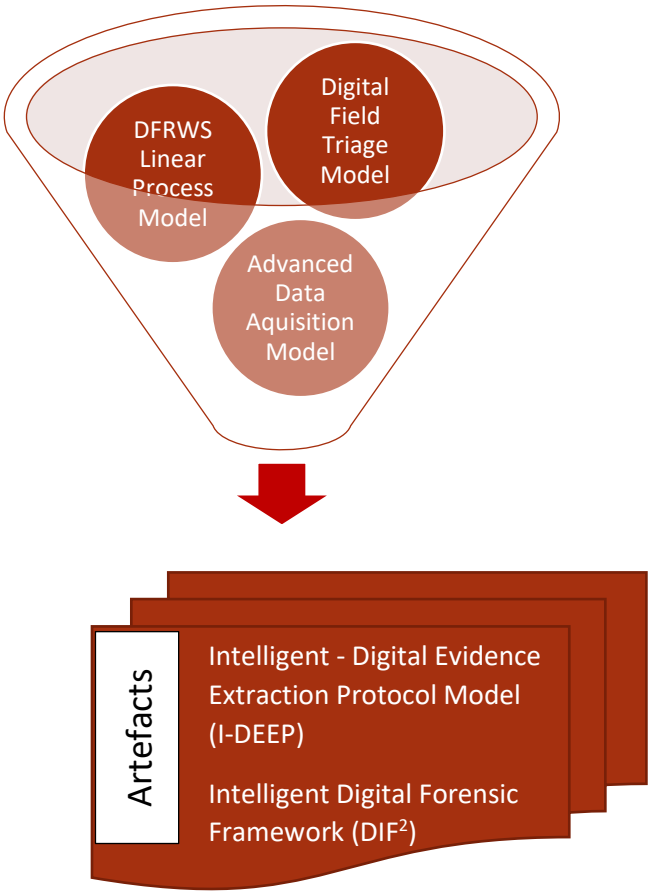


Figure 3.7: Conceptualisation of I-DEEP Model and DIF^2 Framework

The design and development of new digital investigation protocol model called Intelligent-Digital Evidence Extraction Protocol or I-DEEP has been a derivative of three basic models namely DFRWS Linear Process Model (DFRWS, 2001), Digital Field Triage Model (Hitchcock et al., 2016) and Advanced Data Acquisition Model (Adams, 2013). Figure-3.7 illustrates the convergence and conceptualisation of the process. A complete Research Process Model has been represented in Figure 3.8.

3.5 Research Process Model Designed for the Research

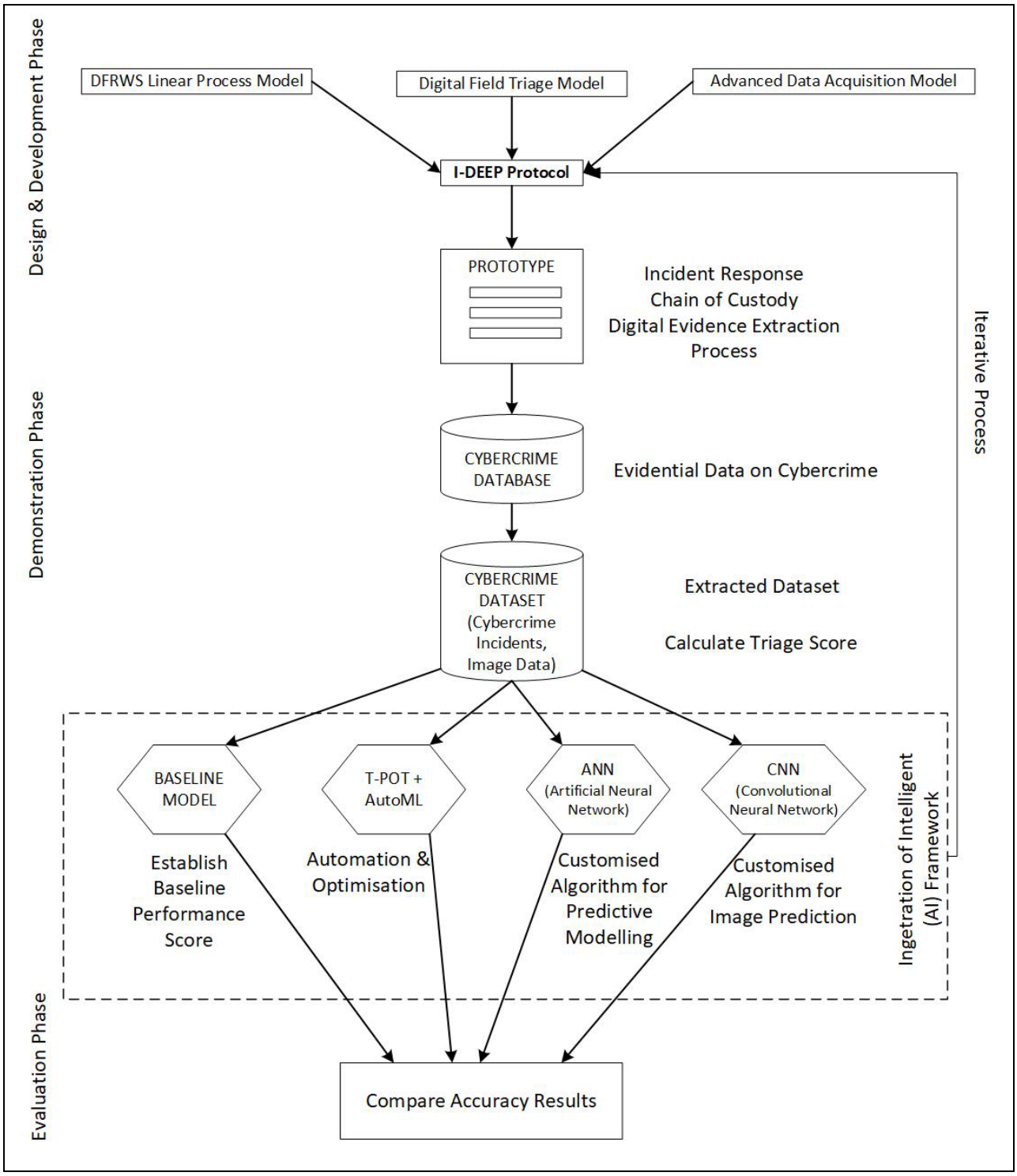


Figure 3.8: Research Process Model Conceptualised for the research

As illustrated in Figure 3.8, The newly conceptualised protocol model, Intelligent-Digital Evidence Extraction Protocol or I-DEEP is built upon theoretical underpinnings of three basic models- The DFRWS Linear Process Model (DFRWS, 2001), The Digital Field Triage Model (Hitchcock et al., 2016) and the Advanced Data Acquisition Model (Adams et al., 2013). This research specifically considers the gaps in these existing models and endeavours to improve upon them. These improvements have been projected under the 'extended models' created from the existing ones and explained under sections 2.5.6 and 2.5.7 and also delved upon in chaper-5 again to establish continuity towards development of I-DEEP protocol and DIF^2 framework in forthcoming chapters. It is imperative to develop new and improved models in the context of meeting the requirements of paradigm shift constantly happening in the field of digital forensics. The 'Research Process Model' illustrates the whole conceptual modelling of the research process represented in figures 3.5 to 3.8, and their alignment to the DSR methodological process adopted from Peffers et al., (2007) for this research.

The DFRWS 'Nucleus' and 'Linear Process Model' were redesigned by the researcher (figure-2.7 and 2.8) in order to improve and adapt them to South African context as well as new requirements arising due to changes in legal, technological and methodical spheres in digital forensics domain. These adaptations are based on requirements identified in literature review and further elucidated in forthcoming chapters, and are new contributions to the body of knowledge. In order to improve efficacy in decision making and provide predictive analysis capabilities to the first responders and digital forensic investigators, the newly developed protocol redefines the evidence extraction process and also implements Artificial Intelligence (AI) to create an intelligent tool for Incident Response (IR) and decision making. A new AI based framework is also developed and implemented by the researcher to make the artifact "intelligent" and efficient. A complete representation of Research Process Model that forms the theoretical underpinnings of this research and also guides the 'Design and Development" process is provided in figure-3.8.

The dataset used in the experiments is extracted from the cybercrime 'cloud' database, which consists of cybercrime incidents recorded in three major South African cities (Durban, Capetown, Johannesburg). Dataset contains cybercrime cases recorded under variables defined as cybercrime 'type' in these three major cities.. Therefore extracted dataset consists of four columns identifying 'cybercrime type' like Phishing, ZeroDay Attack, Ransomware Attack, Child Pornography and

the City. In order to test the models and their performance, the extracted dataset is used, which consists of 'count' of four types of cybercrime incidents namely 'Phishing, ZeroDay Attack, Ransomware Attack, Child Pornography', that occurred in three major cities 'Durban, Johannesburg and Capetown'. The cybercrime incidents extracted, represents lowest severity level (Phishing) to highest (Child Porn) and therefore provides a severity spectrum in cybercrime 'type' parameters. The dataset description is important to understand the nature of data comprising the dataset, so that a appropriate strategy can be adopted to analyse the data, and also, the output can be cross validated. This dataset is available in a public repository specified by the URL parameter in the 'cloud-based' database. The dataset description has been provided in section 6.3.1.

3.6 Summary

This chapter provided an overview of research philosophies and methodologies commonly used in the field of Information Systems and Technology (IS&T), and identified the most appropriate methodology for the current research. Research methodology refers to the theoretical framework that guides the study and determines the ideal research plan. Various experts have provided their own explanations of this concept, which were discussed in this chapter. In any research field, a systematic and scientific approach is taken by researchers to investigate a particular phenomenon or problem, which is called research methodology (RM). It involves various techniques and procedures to collect, analyse, and interpret data, to answer research questions and achieve research objectives. A sound research methodology is essential to produce reliable and valid research results, which can be used to advance knowledge and inform decision-making. It also enables researchers to design studies that are appropriate for their research questions and to use appropriate methods to collect and analyse data. Various research methodologies exist, each with their own strengths and weaknesses. Quantitative research methodologies involve numerical data, statistics and mathematical models, while qualitative research methodologies involve non-numerical data such as interviews, observations, and case studies. Mixed-methods research methodologies combine both approaches and are becoming popular as they offer the advantages of both quantitative and qualitative research methods, allowing researchers to gain a more comprehensive understanding of the phenomenon being studied. Design Science Research (DSR) is a problem-solving and creative design approach that develops new artifacts to solve complex problems in a specific domain. It is a commonly used method in Information Systems research to

develop and evaluate solutions and artifacts in Information Systems and Technology (IS&T) domain. DSR involves developing a rigorous research framework, followed by designing, implementing, and evaluating an artifact through multiple iterations until a satisfactory solution is obtained. The goal is to produce new knowledge in the form of an artifact that can improve real-world systems and processes. This research uses DSR principles to create AI-based interventions that can assist first responders in cyber-investigations. The study outlines the research process and procedures, including the research philosophy, approach, strategy, design, techniques and procedures to address the research questions. The newly developed artifacts (I-DEEP protocol, DIF^2 intelligent framework) endeavours to fill the gaps and provide interventions to achieve the defined objectives of the research. The forthcoming chapters will elaborate on the details of the new artifacts developed in stage called 'Design and Development' of DSRPM by Peffers et al. (2007) adopted for this study.

CHAPTER 4: INCIDENT RESPONSE - INTEGRATION WITH AGILE APPROACH

4.1 Introduction

This chapter discusses Incident Response (IR) and different aspects of it in the field of digital forensic investigations emphasising a 'more agile approach', thus making it more applied and implementable. What are the keys actions that an organisation or its IR teams should take and what methodology should be followed, is crucial to minimise the damage caused by a cyber-attack or a security breach? How 'Triaging' can be used to reduce response time and provide quick decision making? These are the primary objectives that guides the proceedings of this chapter. It is therefore critical for organizations to respond to security incidents in a systematic and efficient way as is in order to minimize the damage caused by potential cyber-attacks and recover from their aftermath. To achieve a structured response, companies of all sizes are integrating incident response capabilities into their current policies, processes and procedures (Kruse & Heiser, 2001).

To establish IR capability in an organization, several key concepts need to be addressed. Firstly, companies need to understand the IR process, which outlines the sequence of events that occurred during an incident and the corresponding actions required at each stage. Secondly, a team of skilled personnel should be established as the core of the IR capability, and a formal plan with all the related processes should be developed. The IR plan and the affiliated processes provide an organized strategy that organizations should follow when an incident is encountered. Finally, organizations must continually assess, test, and improve their incident response plan as new threats emerge. By adopting new frameworks conceptualised in the novel protocol, companies can prepare for the unfortunate reality of a security incident compromising their operations, an event that many organizations have already experienced (Johansen, 2017).

4.2 Implementing Incident Response in Digital Forensic Process: An Agile Approach

As it has been elaborated in section 1.2.8 that the Digital Forensic (DF) process comprises six main phases: "Identification, Preservation, Collection, Examination, Analysis, and Presentation" (DFRWS, 2001). In the Incident Response (IR) process, the 'Collection Phase' becomes very important and needs specific attention. Therefore, this phase needs further discussion in reference to this chapter. As discussed earlier that 'Collection Phase' mainly consists of gathering crucial digital evidence. It is also crucial that evidence is handled correctly and chain of custody is

properly maintained. After critiquing various digital forensic investigation models, it has been established that the IR process can be very complex and time consuming. It also may be difficult to estimate how much time should be ideal for evidence collection phase. The question that needs to be answered is to how this process can be made more agile and a quick decision-making be incorporated using 'Triaging' process. These requirements have been addressed in redefining the IR process in this chapter and further in developing the novel data extraction protocol (I-DEEP), which is elaborated in chapter-5 and its implementation in a prototype design in chapter-6.

4.3 Preserving Digital Evidence

It is critical to handle and secure evidence properly since errors during evidence acquisition can compromise its integrity and render it unsuitable for forensic purposes. If the incident involves legal proceedings, crucial evidence may be excluded from criminal or civil cases if the process is not done meticulously. To ensure proper evidence handling, there are several key issues that must be addressed, for example tampering of original evidence, documentation of the source of evidence, type of evidence (text files, images, video etc.) and maintaining a proper chain of custody (Nieman, 2009).

4.3.1 Original Evidence Tampering

Digital forensic examiners must refrain from altering the original evidence. For instance, they should avoid accessing a running system unless it is absolutely necessary. Nevertheless, certain tasks may inevitably modify the evidence, and this can be mitigated by thorough documentation and providing sound reasoning for any changes made (Institute of Justice, 2008).

4.3.2 Documentation of Evidence

The saying "if you didn't write it down, it didn't happen" commonly used in law enforcement is particularly relevant in digital forensics. It is essential to document every action taken, including detailed notes, photographs or diagrams. In the event that the integrity of the evidence is challenged, this documentation allows examiners to reference the documentation and reconstruct the series of events (Institute of Justice, 2008).

4.3.3 Maintaining Chain of Custody of Evidence

The procedure of documenting the movement of evidence from its initial possession to its disposal or return is referred to as the "chain of custody". The preservation of the chain of custody is crucial because any omissions in the documentation could result in the exclusion of the evidence from legal proceedings (Kruse & Heiser, 2001). As a result, it is essential to document the complete life cycle of the collected evidence with accuracy.

There are two key approaches to recording and preserving the chain of custody. The first method is the traditional approach, which involves the use of paper-based forms containing the essential information for initiating and maintaining a chain of custody. Although this method requires more vigilance to ensure the form's security and protection against tampering or destruction, it is a cost-effective option for smaller CERT teams that lack the resources to implement an automated system.

The second method is electronic and involves the use of specialised software to automate the process of maintaining chain of custody of evidence. Use of stickers that are barcoded uniquely are assigned to every piece of evidence, and a scanner can be used to generate an "electronic trail" by scanning these barcodes (Johansen, 2017).

Figure 4.1 represents a Chain of Custody form by "National Institute of Standards and Technology" (2013) (NIST, accessed on 02-Apr-2023) that provides a standardised template for implementing chain of custody process.

4.3.4 Implementation of Chain of Custody Process in Prototype

The first section of the form developed by NIST (2013) provides a comprehensive description of the evidence collection process and chain of custody, which may seem repetitive, but is critical in digital forensics. The detailed record-keeping is needed to ensure that the evidence collected by investigators is authentic. However, the documentation of evidence and the "chain of custody" implemented in our prototype has been streamlined and a 'thin model' has been adopted to make the 'chain of custody' documentation process more agile and a more pragmatic approach has been used for implementation in the prototype. However, some details are discussed here for the purpose of giving complete perspective of the process involved. Not all of the aspects are obligatory to

implement in our ‘case-based model’ and some aspects have been skipped to make our prototype model easily implementable as the main focus is to create a decision support tool and not a complete documentation or ‘chain of custody’ solution.

Evidence Item Number: A distinct item number should be assigned to each evidence item captured in the chain of custody form. In cases where multiple pieces of evidence are involved, it is necessary to fill out a separate form for each item.

Item Description: A brief summary of the item or item details should be incorporated, e.g. "500 GB SATA HDD."

Item Manufacturer: Having this information can be beneficial in situations where there are many different forms of evidences that may have different manufacturers.

Model Number of Item: Recording the model number of the items is important as it provides additional information about the item, considering that there are several model numbers available for different components.

Item Serial Number: When multiple items have the same make and model number, recording this information becomes crucial, especially in incidents that involve several systems. Without this information, reconstructing the chain of custody becomes challenging.

When dealing with evidence in the form of log files or images obtained during an investigation, an alternative section is utilized, which includes the following elements:

Date/Time Acquired: The precise date and time when the files were obtained.

Description: A brief summary of the acquired media.

Method: The method employed to obtain the evidence, such as a forensic tool or a simple copy.

Storage Drive: The external media used to store the files should be recorded.

Name of the File/Image: Each image or file should be assigned a specific and relevant filename.

Hash Value: Each file should have a unique hash value computed for it to ascertain its data integrity.

An item can go through various stages of investigation (or transitions in location) known as life-cycles. The subsequent section of the chain of custody form outlines each stage of the item's life cycle and records the following information:

Tracking No: A sequential number indicating the stage in the life cycle of the item.

Date/Time: The precise date and time at which the item moved from one stage to the next.

FROM and TO: The name of the person or the storage location involved in the transition of the item from one stage to the next.

Reason: A valid reason must be provided for each transition in the life cycle of the item. To ensure accountability, all individuals involved in the chain of custody must sign the form. Additionally, the date and time of each stage should be recorded to allow for a comprehensive reconstruction of the chain of custody, which is vital in any digital forensics' investigation.

The chain of custody has to be maintained meticulously throughout the lifespan of a piece of evidence, including when it is returned or destroyed. Each instance has to be documented in the "chain of custody" form, which must be preserved alongside other documentation created during the incident (NIST, accessed on 02-Apr-2023). This documentation must be included in any subsequent report produced.

Figure 4.1 and 4.2 represents a standard "Chain of Custody" Form-1 and Form-2 by National Institute of Standards and Technology (2013). However, only some critical aspects have been adopted in designing the 'protocol' and 'prototype model' as of reasons explained earlier in this section.

Property Record Number: _____

Anywhere Police Department

EVIDENCE CHAIN OF CUSTODY TRACKING FORM

Case Number: _____ Offense: _____
Submitting Officer: (Name/ID#) _____
Victim: _____
Suspect: _____
Date/Time Seized: _____Location of Seizure: _____

Description of Evidence		
Item #	**Quantity**	**Description of Item** (Model, Serial #, Condition, Marks, Scratches)

Chain of Custody				
Item #	**Date/Time**	**Released by** (Signature & ID#)	**Received by** (Signature & ID#)	**Comments/Location**

APD_Form_#PE003_v.1 (12/2012) Page 1 of 2 pages (See back)

Technical Working Group on Biological Evidence Preservation. *The Biological Evidence Preservation Handbook: Best Practices for Evidence Handlers*. U.S. Department of Commerce, National Institute of Standards and Technology. 2013.

Figure 4.1: Chain of Custody Form-1 by NIST (2013), (NIST, accessed on 02-Apr-2023)

EVIDENCE CHAIN-OF-CUSTODY TRACKING FORM
(Continued)

Chain of Custody				
Item #	Date/Time	Released by (Signature & ID#)	Received by (Signature & ID#)	Comments/Location

Final Disposal Authority

Authorization for Disposal

Item(s) #: _____ on this document pertaining to (suspect): _____
is(are) no longer needed as evidence and is/are authorized for disposal by (check appropriate disposal method)

☐ Return to Owner ☐ Auction/Destroy/Divert

Name & ID# of Authorizing Officer: _____ Signature: _____Date: _____

Witness to Destruction of Evidence

Item(s) #: _____ on this document were destroyed by Evidence Custodian _____ID#:_____
in my presence on (date) _____.
Name & ID# of Witness to destruction: _____ Signature: _____Date:

Release to Lawful Owner

Item(s) #: _____ on this document was/were released by Evidence Custodian
ID#: to
Name _____
Address: _____ City: _____State: _____ Zip Code: _____
Telephone Number: (____) _____
Under penalty of law, I certify that I am the lawful owner of the above item(s).

Signature: _____ Date: _____

Copy of Government-issued photo identification is attached. ☐ Yes ☐ No

This Evidence Chain-of-Custody form is to be retained as a permanent record by the Anywhere Police Department.

APD_Form_#PE003_v.1 (12/2012) Page 2 of 2 pages (See front)

Technical Working Group on Biological Evidence Preservation. *The Biological Evidence Preservation Handbook: Best Practices for Evidence Handlers*. U.S. Department of Commerce, National Institute of Standards and Technology. 2013.

Figure 4.2: Chain of Custody Form-2 by NIST (2013) (NIST, accessed on 02-Apr-2023).

4.4 Digital Forensic Lab Requirements

A laboratory for digital forensics is crucial for extracting potential evidence from computer systems. It requires the use of proper expertise, techniques and tools. To preserve the integrity of collected evidence and ensure privacy for forensic examiners, it is necessary to have a separate location from normal business operations. A digital forensics lab should possess specific features, including physical security and evidence lockers, to facilitate the examination process and prevent tampering or destruction of evidence.

To perform essential tasks in the laboratory, having an adequate number of computers and hardware is crucial. These tasks include creating hard drive images, processing large amounts of data, and that requires a forensics-enabled computer with software tools as well as sufficient RAM. In addition to the need for ample memory and processing power in workstations, examiners must have necessary software to review extensive amounts of data. Therefore, forensic workstations should have a primary Operating System drive capable of accommodating digital forensic software tools and a secondary drive exclusively dedicated to storing evidence collected during investigations.

Apart from workstations with forensic capabilities, examiners may need access to a computer connected to the internet for communication or reporting. However, it is crucial to ensure that the forensic workstation remains disconnected from the internet to maintain security and prevent possible evidence tampering. It is advisable to use a separate machine for online research or report writing. A physical write blocker is an essential device that facilitates a connection between the evidence hard drive and the forensic imaging machine. Using a physical write blocker instead of a direct USB drive or Thunderbolt connection offers a significant advantage, as the digital forensic examiner can be confident that there is no possibility of any data being written to the drive containing evidence (Johansen, 2020).

4.4.1 Controlled Physical Access

Strict regulations must be enforced to control the lab entry and maintain the chain of custody in the forensic lab. Entry into the lab should only be granted to individuals with a legitimate reason, and access must be tightly controlled. The lab should always remain locked, and access should

only be granted through the use of access cards or fobs managed by a centralized system that keeps track of personnel access.

The lab should also have evidence lockers that can securely store evidence when not being examined. The lockers should be locked, either via on-board or combination locks, and the keys should be kept within the lab accessible to the examiners. Ideally, each incident should have its own locker to prevent mingling of digital evidence. Moreover, the climate and humidity levels of the lab should be managed similarly to data centres to preserve evidence.

4.4.2 Lab Tools

Examiners may need to remove screws or cut wires during collection and examination of evidence in certain cases. Therefore, it would be beneficial for them to have a small set of hand tools readily available. In addition, the lab has to be equipped with secure boxes to store evidence. In case of a need for examining devices like smartphones and tablets, it is advisable to keep Faraday bags readily available. These bags are effective in isolating smartphones and tablets from the cellular network while still keeping the power source intact.

4.4.3 Lab Hardware

We have had a detailed discussion under section 2.8, where various software tools are described in a comprehensive manner. It is also imperative that specialised hardware is also carried for evidence collection. These tools are needed for imaging and securing of evidence to avoid any evidence tampering. A brief description of these tools is also provided in this section.

In order to carry out the necessary tasks in the laboratory, having an adequate number of computers and hardware is essential. These tasks involve creating hard drives images and also processing large amounts of data, which necessitates the use of a forensics enabled computer with sufficient RAM. Although personal preferences may differ, it is suggested to have 32 GB of RAM or more.

Apart from need for sufficient memory and processing power in the work-stations, examiners need to review extensive amounts of data. Therefore, forensic workstations also require a primary Operating System drive that can accommodate digital forensic software tools and a secondary drive exclusively dedicated to storing evidence collected during investigation. The recommended size for secondary drive must be at least 2 TB.

Besides workstation with forensic capabilities, the examiners need to have access to a computer that is connected to internet for research, communication or reporting. However, it is critical to ensure that the forensic workstation remains disconnected from the internet to maintain security and prevent any possible evidence tampering. It is recommended to use a separate machine for online communication or report writing.

A physical write blocker is an essential device. It facilitates a connection between the evidence hard drive and the forensic imaging machine. Using a physical write blocker offers a significant advantage as the digital forensic examiner can be confident that there was no possibility of any data been written to the drive that contains evidence (Johansen, 2020).

4.4.4 Forensic Jump Kits

When responding to incidents outside their location, CERT team members may encounter the challenge of conducting investigation off-site. This situation is particularly prevalent in larger enterprises or when CERT teams are consulting for other organizations. In such cases, teams may have to perform the entire incident response without any support from the digital forensic lab. To address this issue, CERT teams can prepare jump kits that come preconfigured with the necessary hardware and software to execute incident response tasks. The forensic jump kits should be extensive enough to support any investigation from beginning to end, including identifying secure zones at the location of incident in order to store and analyse digital evidence.

Forensic Jump kits must be portable and configured to fit into a hard-sided case that is secure to ensure quick deployment. CERT teams should keep the jump kits ready for immediate use. After every investigation, the kit should be replenished with any items that were used in the previous incident. All hardware and software tools should also be properly reconfigured to ensure they are available when needed. This helps ensure that first responders and CERT teams are always prepared to respond to an incident.

For efficient hard drive imaging, it's imperative that the jump kit comprises a forensic laptop with no less than 32 GB of RAM. Additionally, the laptop must have a forensic software installed, along with at least one forensic Linux Operating System platform such as SIFT, KALI or CAINE. Multiple CAT5 cables of varying lengths should be added to access networks or network hardware. To image any hard drives encountered, a physical write blocker is indispensable. Furthermore,

multiple USB compatible hard drives with capacity of 1 TB or 2 TB should be included to image potentially compromised systems.

To ensure collection of forensically sound evidence, it's vital to have numerous large-capacity USBs (64 GB) to capture RAM contents, offloading log-files, or any other information that needs to be captured from command-line outputs of some applications. CERT teams should also carry bootable CD/DVD or USB that contains several Linux distributions, which can be useful in specific cases. On-site, evidence bags or boxes must be readily available to secure the evidence and transport it off-site if necessary. Moreover, anti-static bags are crucial for transporting hard drives seized as evidence.

Including chain of custody printed forms or having an application-based replacement of them and a hardware toolkit in the jump kit is essential for a successful incident response. The forms allow for the proper documentation and tracking of evidence, while the toolkit provides the necessary tools to access and remove hardware. Both items are crucial for maintaining the integrity of the evidence and ensuring that the incident response is conducted efficiently and effectively. Having a small hardware toolkit with tools like pliers, different screw drivers and a flashlight is crucial in a jump kit. This allows responders or CERT teams to cut connections, remove hard drives, or access hard-to-reach areas of a data centre during an incident response.

4.5 An Overview of Incident Response Process

During the course of their lifespan, cybersecurity incidents usually follow a consistent path. When an organization has an established incident response capability, it's expected that the incident response team is equipped to handle incidents at every stage during the investigation process. The investigation typically starts when the organization becomes aware of any suspicious activity, which can be identified through an alert or notification received from the security control team of an external party or security operations centre (SOC). Upon receiving the alert, the CERT proceeds to analyse and contain the incident and to restore normal operations of the affected system. After a thorough examination, it's evident that every incident can be utilized as a learning opportunity to enhance the organization's preparedness for future incidents in the Post-Incident Activity phase. The organization then uses the experience to prepare for the next incident (Kruse & Heiser, 2001).

According to Johansen (2020), the incident response process is made up of six different phases, and each stage involves specific actions that the incident response team can take to effectively manage the incident and advise the organisation. These phases are described here.

Preparation Phase

Achieving a successful incident response requires effective preparation, as an unorganized response can potentially worsen the situation. This preparation should include developing an incident response plan, ensuring that personnel assigned to incident response tasks are fully trained in all relevant processes and procedures; have a good experience in investigation and use of various specialised DF tools. It's also important to acquire and integrate forensics hardware and software knowledge into the overall process. There should be an organisational strategy of doing regular exercises in order to familiarize the organization with the incident response process and ensure they are well-prepared to handle any incidents that may occur.

Analysis Phase

During the analysis phase, the organization's incident response team or a trusted third party typically collect evidence from various systems. This includes obtaining data from network connections, log files, running software processes, and running memory. The evidence collection phase can range from a few hours to several days, depending on the nature of the incident. Once the evidence is collected, it is examined using various digital forensic tools. Analysts use these tools to determine the specifics of the incident, such as what incident occurred, which systems were compromised, and whether there was loss of any confidential data.

The primary objective of the analysis phase is to identify cause of the incident. The next step is an activity reconstruction process to uncover all the threat-actors involved throughout from initially compromised stage to final detection stage. Through this process, analysts can determine how the incident occurred, what vulnerabilities were exploited, and what damage was caused. The analysis phase is critical in understanding the scope and severity of the incident and provides valuable insights into how to prevent similar incidents from happening in the future.

Containment Phase

After an organization gains a thorough understanding of the incident and the systems affected by the incident, the next phase is containment. In this phase, steps are taken to restrict the threat actor's ability to continue inflicting damage on the network infrastructure by compromising network resources, extraction of confidential data or communication with attacker's command and control servers. The containment measures may vary from securing a firewalls ports and IP addresses to physically disconnecting the network connections from all the infected systems. As each incident requires a unique containment approach, having multiple options at hand allows personnel to prevent further damage in case there is any security incident detection either before or during the data theft process.

Eradication and Recovery Phase

In this phase, the organization takes actions to eliminate the threat-actors from the affected networks. If there is a malware infection detected, then the organization may use advanced anti-malware software or choose to wipe and reconfigure infected machines. Compromised user accounts may also be removed or modified. Furthermore, if the organization discovers an exploited vulnerability, its incident response team will apply software updates or vendor patches. The recovery actions should align with the disaster-recovery or business-continuity plans of the organization. During this phase, fresh operating systems or applications are reinstalled, and data is restored from any older backups that exist. To ensure due-diligence and thoroughness, the organization will conduct an audit of their administrator accounts and all existing user accounts to confirm that no new accounts were created by the "threat actors". Lastly, there is a need to conduct vulnerability scan to ensure that all vulnerabilities that existed and were exploited by the threat-actor, have been removed.

Post Incident Activities

After completing the incident response process, a thorough incident review is conducted with all relevant parties. The post-incident activities entail a comprehensive assessment of all actions that are undertaken when the incident happened, with a focus on identifying successful and unsuccessful strategies. These reviews are essential to highlight specific actions or tasks that had

an impact on the incident response outcome. A written report is generated during this phase that consist of documentation of the remedial activities carried out during the incident, including potential legal ramifications. The documentation should be completed in detail and chain of events should be presented clearly with a specific emphasis on identifying the main cause of the incident. It is important to keep in mind that non-technical stakeholders may review the report, so the concepts should be explained clearly without much technical jargon.

4.6 Forensic Tools used in Incident Response, Disaster Recovery

It is vital for Incident Response (IR) teams to have a sound knowledge of various network-diagnostic tools, penetration-testing tools and Anti-Malware tools to operate and mitigate various threats posed to organisational infrastructure by attackers and also discover any vulnerabilities existing in networks or systems. Today's commercial and freeware market offers a variety of software tools for digital forensic investigations. In order to ensure effective operations, a DF lab should be equipped with variety of software tools that can perform all desired functions. Essential software for the lab includes programs capable of network-diagnostics, imaging tools of evidence drives, examining those images, analysing captured memory, and producing detailed reports (Johansen, 2020).

Digital forensic analysts also need to utilize various other types of software apart from digital forensic tools like encryption and password cracking tools, anti-malware tools and other utilities. Of course, popular software-applications, which are designed to perform various digital forensic tasks on purpose are indispensable. These applications are widely used in law enforcement, government communities, and private industry. Some of these tools have been discussed under section 2.8 and 2.9, but it will be useful to elaborate a bit more on some of them already covered, since there are specific features which become more important from ‘Incident Response’ perspective.

An overview of some useful IR tools, both proprietary and free/open source that first responders and CERT teams need to know about and are available to their discretion, is provided specifically to make the IR process easier and faster. An analysis of IR tools is also provided in this section from the point of view that first responders have a decision making capability to select the most

appropriate tools to carry IR process effectively and promptly (*27 BEST Penetration Testing (Pentest) Tools in 2023*, Accessed on 19-Mar-2023).

Another motivation for providing this specific overview and analysis of software tools essential for first responders, apart from enriching their arsenal in the fight against the cyber-threats, is also that this information can be integrated with the newly developed prototype as database or repository. This integration can provide an 'expert system' functionality to the first responders, so that they can get suggestions for optimum tools that will suite their IR investigation.

EnCase

This software which is developed by OpenText, is a comprehensive digital forensic tool used for Incident Response that performs a broad range of tasks from collecting vital evidence, mainly from storage media and drives. It is most widely used commercial tool. One of the most remarkable features of EnCase is its reporting function that enables examiners and first responders to present case-data in a digestible format. However, the application's cost can be a drawback, making it challenging for many forensic examiners on a limited budget.

In short, to ensure efficient digital forensics investigations, a digital forensics lab should have a collection of multiple software tools to meet any requirements. Forensic applications, such as EnCase, Access Data-FTK and Autopsy are commonly utilized in the industry for Incident Response (IR) and offer a range of features for analysing digital evidence.

Autopsy

This feature-rich software, developed by Brian Carrier, is an open source project that automates essential digital forensic tasks. Additionally, Autopsy has open-source modules that provide further functionality which makes it effective for evidence collection.

Rapid7's Nexpose Vulnerability Management Tool

Rapid7's Nexpose is a valuable software for managing vulnerabilities. It provides real-time monitoring of potential threats and quickly adapts to new risks by collecting and updating fresh data, enabling first responders to take remedial action promptly when necessary.

Key features of Nexpose include providing IR teams with a view of potential risks in real-time, offering solutions in most practical way to enhance productivity, pinpointing areas that require attention to identify vulnerabilities, and enhancing the overall security scenario.

Nessus

Nessus is a widely-used vulnerability assessment tool that is designed to identify security vulnerabilities and misconfigurations in computer systems and networks. It is used by security professionals and first responders to scan and analyse systems for potential vulnerabilities, including network protocols, open ports, and services, among others. Nessus is a commercial product with different pricing plans and supports multiple operating systems and architectures.

Personal Software Inspector

Personal Software Inspector also called PSI, is an open-source security application that enables identification of vulnerabilities in applications running on personal computers or servers. This powerful tool offers a range of features, including its availability in eight languages. One of its standout features is the automation of updates for programs that are insecure due to lack of manual updation.

The PSI tool can scan thousands of different computer programs and can automatically detect insecure ones, making it a reliable and convenient solution for software management. It also performs regular scans of PCs to identify any programs that are vulnerable and alerts users to programs that cannot be updated automatically. Overall, PSI is an efficient and user-friendly software management solution for ensuring the security of computer applications.

Kismet

Kismet is an Intrusion Detection System (IDS) that functions as a wireless network detector and for Wi-Fi networks and a pen-testing tool. Additionally, it supports other network types through plugins. Main features of Kismet include a standard “PCAP logging” capability, a modular configuration to support client/server architecture, and the ability to expand its main features through its plug-in architecture. It also supports capturing of multiple sources and remote sniffing

using distributed architecture through remote capture feature that is light-weight. It also supports easy integration with other DF tools using XML output functionality.

Parrot Security

Parrot Security is a versatile and powerful tool that is specifically designed for network diagnostics and penetration testing. This all-in-one portable lab is an essential tool for CERT teams as well as digital forensics professionals. The software includes a range of powerful features, including comprehensive security tools for conducting penetration tests and security audits. Additionally, it comes preinstalled with regularly updated libraries and offers access to powerful mirror servers located worldwide.

The development process of this tool is community-driven which ensures that the software remains up-to-date and efficient. Another notable feature of Parrot Security is its anonymity and crypto tools, which enable users to protect their privacy. Moreover, the tool also offers a “Cloud OS” that is tailored for servers only. Therefore, Parrot Security is an indispensable tool for first responders and security professionals that delivers robust and comprehensive security solutions.

Snort

Snort is a highly effective Intrusion Detection System that provides maximum protection from malware attacks by combining the strengths of different inspection methods like protocol-based, signature-based and anomaly-based. It is an open-source tool. Its high speed and accuracy in detecting threats have made it a popular choice for analysing security breaches.

One of Snort's main features is its flexibility in creating customized network security solutions. It allows users to tailor their security measures to meet their specific needs and offers quick protection against emerging threats. Another notable feature is its ability to test the SSL certificate and cipher acceptance of a specific URL, as well as verifying the “Certificate Signer” Authority. This feature provides an added layer of security, ensuring that users are accessing secure and reliable websites.

Additionally, Snort offers a user-friendly interface that allows users to submit list of false positives and false negatives. This feature allows users to fine-tune their security measures and improve the overall effectiveness of their network security.

TraceRoute

TraceRoute is a powerful application that enables users to analyse network paths with ease. With this software, IR teams can track IP addresses, identify hostnames, and monitor packet loss, and obtain analysis accurately. It uses a command line interface. The tool supports network path analysis using both TCP and ICMP protocols and offers several useful features to enhance the user experience.

One of the main features of TraceRoute is its “txt logfile” creation ability, which provides a detailed record of network activity. The tool can support both IP4 as well as IPV6 addresses, making it compatible with a wide range of networks. Additionally, TraceRoute can detect path changes and send notifications, ensuring that security teams are aware of any issues that may arise.

Another key feature of TraceRoute is its ability to continuously probe a network. This feature enables users to monitor network activity and identify any potential issues before they become critical. With its versatile set of features, TraceRoute is an invaluable tool for network analysis and troubleshooting.

BackBox

BackBox is an open-source project that aims to promote a security-focused culture in IT environments. The project offers two variations, BackBox Linux and BackBox Cloud, both of which are equipped with popular DF analysis tools. The project's main goal is to reduce the resource needs of companies and lower the costs associated with multiple network devices management.

One of BackBox's key features is its ability to perform scheduled automated configurations without the need for agents or network configuration. This pen-testing tool is fully automated, which saves time and eliminates the requirement of network devices tracking individually. Additionally,

BackBox provides secure access to devices and supports “configuration file encryption” and credentialing, thus enhancing overall security.

BackBox also offers IP-based access control, automatic remote storage, and self-backup for further strengthening network security. Its pre-configured commands eliminate the need to write commands, making it easy to use and saving time in the process.

SM Anywhere

SM Anywhere from Open Threat Exchange, enables professionals to track the reputation of their organization and is a freely available service. With this tool, organizations can easily monitor the public IP and domains and can be used by IR teams for vulnerability analysis.

One of the key features of Open Threat Exchange SM Anywhere is its continuous threat intelligence, which provides real-time updates on emerging threats. This allows organizations to stay ahead of potential risks and take action before they become major issues. The tool also offers the ability to monitor valuable assets in private as well as hybrid cloud, and infrastructure on their premises, providing comprehensive coverage across all environments.

Open Threat Exchange SM Anywhere also offers the excellent threat detection and incident response instructions that are actionable, making it a powerful tool for organizations of all sizes. It can be deployed quickly and easily with minimal effort, and it offers a reduced total cost of ownership compared to traditional security and vulnerability assessment solutions.

4.7 New Output Mapping Model for Investigation Tools used in Incident Response

Selamat et al. (2008) provides a framework for comparative analysis of digital forensic investigation frameworks where output mapping is used to categorise the frameworks. These mappings are adapted here to do an analysis of most commonly used investigation tools. However, it is observed that the framework is mapped to five standard phases of DFRWS (2001) model and lacks a phase that can provide integration of AI and data visualisation technologies. Therefore, there is a need to provide improved provisions to the existing model to accommodate these technologies. Therefore, an improved (extended) model has been proposed as part of this study that can be used to analyse the investigation tools based on output mapping process.

Figure 4.3 represents this 'Extended Output Mapping Model', which also provides the theoretical and conceptual framework for design and development of novel protocol and prototype for this research. An additional phase is introduced in the model to fill the gap that exist due to lack of visual representation of results, predictive analysis of evidential data, lacking in decision support functions of existing tools and lack of integration of Artificial Intelligence (AI). The modification introduces AI techniques integration into the existing DFRWS (2001) model phases and output mapping. **A new phase (*Phase 5)** have been introduced into the model, which is named as 'Predictive Analysis using AI', with output shown as 'visual representations' of data patterns and performance analysis in the form of correlation matrix plots, Box and whiskers plots, histograms etc.

Phase	Phase Name	Output	EnCase (P)	FTK (P)	COFEE (P)	SafeBack (P)	Nuix (P)	CAINE (O)	WireShark (O)	BackBox (O)	KaliLinux (O)
Phase 1	Preparation	Plan, Authorisation, Notification									
Phase 2	Collection and Preservation	Crime type, Potential evidence sources, media devices, event	✓	✓		✓	✓				
Phase 3	Examination and analysis	Log files, File, Event log, Data, Information	✓	✓	✓	✓	✓	✓	✓	✓	✓
Phase 4	Presentation and Reporting	Evidence, Report	✓								
***Phase 5**	**Predictive Analysis using AI**	**Box and whisker plots, Correlation matrix plot, histograms**									
Phase 6	Disseminating the case	Evidence Explanation, New Policies, New Investigation Procedures, Evidence Disposed, Investigation Closed									

Figure 4.3: Extended Output Mapping Model for Tools Classification (new artifact)

Figure 4.3 represents a newly created 'Extended Output Mapping Model' which provides a framework to analyse commonly used DF tools. Tools flagged with (O) are open source/free tools whereas ones flagged with (P) are proprietary. The 'evaluation rubric' clearly indicates that digital

forensic tools do not comply to all the investigation phases and the corresponding outputs are not supported by all tools. We observe that most commonly used tools used by professional investigators like Encase only covers three phases (phase 2 to 4) 'Collection and Preservation', 'Examination and Analysis' and 'Presentation and Reporting'. Other popular tools like FTK, SafeBack and Nuix only cover two phases (phase 2 and 3) that are 'Collection and Preservation' and 'Examination and Analysis'. While most of other tools like Kali, CAINE, Backbox and Wireshark only caters to one phase that is 'Examination and Analysis'. It is also evident that phase 1 and phase 6 require manual interventions and there is a lack of automation that consumes a lot of time and resources. Another drawback is that the existing DF and IR tools do not support 'triaging'.

Apparently, none of the tools possess an AI based integrations to support prediction analysis mechanism that can utilise big data and produce intelligent output for greater insight into data and support decision making by first responders/cybercrime investigators. It also makes a strong motivation to design and develop new interventions to fill this gap, which is one of the main objectives of this research. Therefore, it was realised that this model can be further extended to incorporate AI based prediction analysis (*Phase 5) to enhance efficacy mechanism. This parameter is not fulfilled by any existing tools and therefore shows the gap in present digital forensic and IR tools.

Based on analysis of popular digital forensic tools provided earlier in this section and the mapping used under Extended Output Mapping Model (figure-4.3), it is evident that current popular DF and IR tools lacks some critical functionality in terms of mapping to standard investigation phases laid down by frameworks like 'Investigative Process Model' and 'Linear Process Models' developed under DFRWS frameworks (DFRWS, 2001). Therefore, this makes a compelling case for need for improvements and integration of AI technologies to existing tools or development of new tools that can support 'predictive analysis', 'triaging' and 'data visualisations' to achieve agility and quick decision making.

4.8 Discussion and Suggestions for Improvement of IR Process

The discussion in this chapter highlights that Incident Response involves a broad range of elements, comprising both legal and scientific components. To carry out digital forensics

investigations effectively, members of Computer Emergency Response Teams (CERT) must possess an extensive knowledge of both the technical and legal aspects of the IR process. They should also have a profound understanding of the various equipment and tools required to appropriately gather, analyse, and present the evidence revealed, while conducting an investigation. It is critical to apply forensic methods to gain insights into the series of events that prompted the investigation of an incident.

Having an established Incident Response (IR) capability is crucial to effectively manage cybersecurity incidents, which typically follow a consistent path. The IR process consists of six phases: “Preparation, Analysis, Containment, Eradication and Recovery, Post-Incident Activities, and Improvement”. Preparation phase involves developing an incident response plan, training personnel, and acquiring and integrating forensics hardware and software to enhance the organization's ability to respond effectively. During the ‘Analysis’ phase, evidence is collected and examined to identify all the root causes of the incident and prevent future incidents. The ‘Containment’ phase involves implementing measures to limit the threat actor's ability to continue compromising network resources. The ‘Eradication and Recovery’ phase entails eliminating the threat actor from the affected network and restoring normal operations. ‘Post-Incident Activities’ phase involve a comprehensive review of the incident and the creation of a detailed report to document the events that occurred and any potential legal consequences.

In this chapter, the focus was to describe complete details of the IR process which also includes ‘evidence documentation’ and ‘chain of custody’. However, the prescribed NIST framework contain too much details and is a very exhaustive process to implement. In this research, the objective is to create a novel protocol that is agile and eventually a prototype that is ‘thinner’ and easier to implement as a decision support tool. For the purpose of practicality, the implementation of most critical aspects of IR is included in a more ‘agile manner’ in the designing of novel protocol and the prototype based on it. The main improvement is attained in implementing ‘agile approach’ in the IR process and adopt ‘case-based modelling’ approach in the process to achieve better efficacy. This approach also emphasise that significant time can be saved in evidence extraction phase by using the ‘triaging’ method for swift evidence analysis and prioritizing ‘collection’ and ‘analysis’ based on ‘triaging’ to identify most critical aspects. This ‘Agile Incident Response’ approach and triaging can help first responders in making quick decisions. These two approaches

(Agile IR and Triaging) have been implemented in designing of an intelligent (I-DEEP) protocol which is covered in detail in chapter-5 (section-5.6) and subsequently in developing a prototype based on this novel protocol in chapter-6. This chapter addressed our research sub-question-4 and provided fundamentals for IR improvements that eventually formed the basis for designing a novel protocol to address the shortcoming in the IR process.

4.9 Summary

This chapter focused on the practical implementation of Digital Forensics Incident Response (IR) process in a more agile manner. This chapter underscored that Incident Response involves legal and scientific components, and members of Computer Emergency Response Teams (CERT) need a comprehensive understanding of both technical and legal aspects for effective digital forensics investigations. The use of appropriate equipment and tools is essential for gathering, analysing, and presenting evidence during an investigation. Proper application of forensic methods is critical for gaining insights into the events prompting an incident investigation. Establishing an Incident Response (IR) capability is crucial for effective cybersecurity incident management, which typically follows a consistent path. The IR process comprises six phases: Preparation, Analysis, Containment, Eradication and Recovery, Post-Incident Activities, and Improvement. The Preparation phase involves developing an incident response plan, training personnel, and acquiring forensics hardware and software. The Analysis phase focuses on collecting and examining evidence to identify root causes and prevent future incidents. ‘Containment’ involves limiting the threat actor's ability to compromise network resources. ‘Eradication and Recovery’ aim to eliminate the threat actor and restore normal operations. ‘Post-Incident Activities’ include a comprehensive review and the creation of a detailed report documenting events and potential legal consequences. ‘Improvement’ is an ongoing process that enhances the organization's IR capability based on lessons learned from previous incidents. This chapter advocated for process improvement by adopting an ‘Agile approach’ in the IR process, readjusting time frames, and implementing ‘case-based modelling’. The benefits of this approach, along with the 'Triaging' method for swift evidence analysis, are highlighted. 'Triaging' aids in prioritizing the collection and analysis based on critical aspects, potentially saving significant time in the evidence extraction phase. The ‘Agile IR’ approach and 'Triaging' contribute to quick decision-making for first responders. These methodologies are incorporated in the design of the I-DEEP protocol covered in Chapter 5.

CHAPTER 5: DESIGN OF AI BASED DIGITAL FORENSIC PROTOCOL (I-DEEP)

5.1 Introduction

This chapter entails designing a novel, AI based Digital Forensic Protocol named as 'Intelligent-Digital Evidence Extraction Protocol' (I-DEEP). It also highlights some background studies in order to contextualise the need for this research. It has been emphasised in chapter-2 that the majority of crime investigations frameworks and protocols are not appropriate for use in scenarios where quick decision making is required. Present investigative techniques, especially those used by law enforcement, due the complexity of the cybercrime and their proliferation, is insufficient to combat it. Due to the limited amount of computing and human resources available, digital investigators are under more pressure than ever to use cutting edge digital forensics and investigative techniques to produce answers quickly.

This research advocates for the adoption of more 'agile' methods to overcome the limitations of existing forensic instruments and effectively tackle cybercrime issues. The study emphasizes the importance of leveraging on 'agile' approach and 'triaging' methods to improve decision making capabilities of first responders. Also leveraging on Artificial Intelligence (AI) technology in digital forensics to develop sophisticated and 'intelligent' tools for better prediction and decision making, is required. By integrating new techniques into digital investigations, it is possible to address the challenges posed by the changing paradigms in increasingly complex domains in which cybercrimes occur.

5.2 Global Challenges in Digital Forensics

The utilization of technology, techniques, and procedures in digital investigations is perceived to be inadequate in keeping up with the methods used by criminals (Casey, 2009). The continuous increase in cybercrime, storage volumes, a broad range of data sources for evidence, and enhanced computational power contribute to the "big data problem in digital forensics". This issue creates a "backlog" of digital devices that require investigation. Various studies have confirmed that the length of this backlog is constantly increasing as the cybercrime numbers are growing exponentially (EURIM, 2010; EURIM, 2022). Corresponding patterns have also been observed by Gogolin and Jones (2010) in the United States.

It is important to acknowledge that the backlog of digital investigations is not solely caused by the increase in dataset sizes, but also due to a shortage of available investigators to conduct the investigations. Furthermore, a significant challenge faced by digital investigators is the uncertainty around which sources of evidence will hold greater relevance to an investigation. At the outset of an inquiry, it is often unclear where digital evidence may be located. This can result in investigators having to photograph every device seized, as well as any potential sources of digital evidence, to ensure that no valuable information is overlooked. When conducting research that involves social media and a large number of participants, the number of data sources can significantly increase. This issue is further compounded by the rapid expansion of storage capacity, which has become more affordable over time. For instance, the storage capacity of hard discs has increased from few Gigabytes in the 1980s to Terabytes in the present times.

Cybercrime and the commission of crimes involving the use of digital devices continue to rise annually as there is a continuous increase in the volume of digital devices outreach. The Federal Bureau of Investigation (FBI) releases annual data that demonstrates an increase in forensic investigations, the volume of data being searched, and investigated per case year over year as shown in table 5.1. At the micro level, the increase of cybercrime may also be seen as a cause of this phenomenon. The data from the forensics Police e-crime section is used in the accompanying figure (Figure 5.1) as an example to illustrate the growth in cybercrime investigations. Even though this is just one instance, it demonstrates the pattern of investigative growth that is shared by numerous police agencies (EURIM, 2010; EURIM, 2022).

Table 5.1: The total number of forensic examinations conducted and amount of data investigated by the FBI from year 2007 to 2019

Year	**2007**	**2008**	**2010**	**2015**	**2019**
Number of Forensic Examinations	4634	4524	6016	6564	7629
Terabytes of Data Processed	1228	1756	2334	3086	4263
Terabyte per Forensic Examination	0.26	0.39	0.39	0.47	0.56

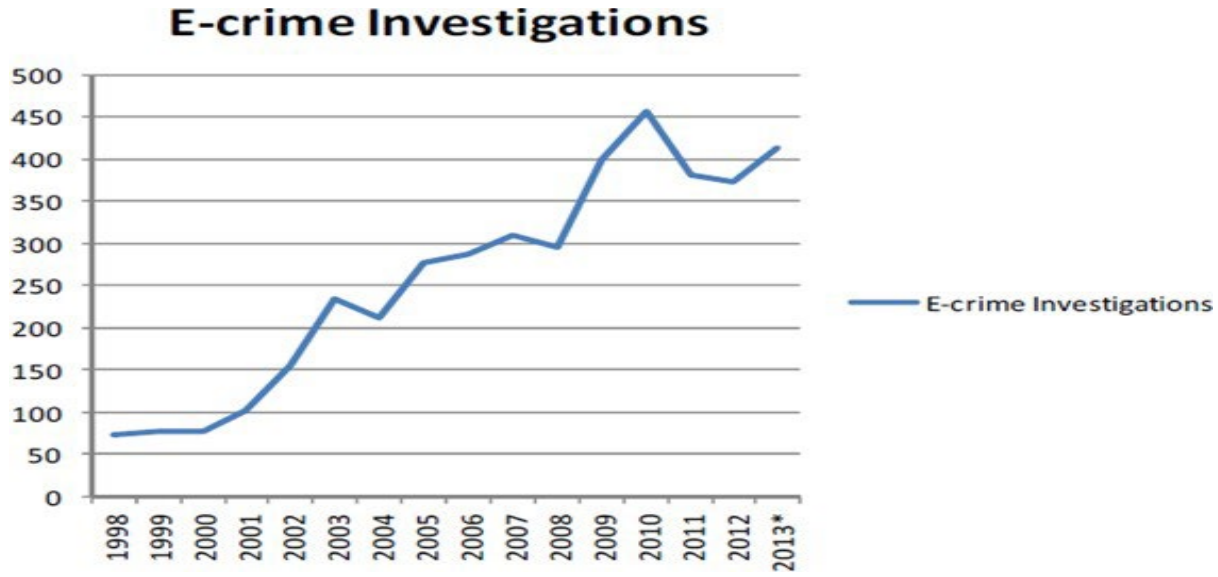


Figure 5.1: E-Crime Investigations Police E-Crime Section by FBI for years 1998 to 2013

In 2012, CART, a division that supports the FBI in searching for and seizing digital evidence, played a crucial role in over 14,000 investigations, conducted more than 133,000 digital investigations, and examined a staggering amount of data exceeding 10,500 Terabytes. Recent individual investigations have also highlighted the magnitude of the problem, as demonstrated by a notable case involving Kim Dotcom in July 2012. In that instance, the FBI was tasked with copying 150 TB of data stored on the "Mega Uploads" server, showcasing the vast volumes of data that agencies have to analyse collectively (US Department of Justice, 2022).

The aftermath of the Enron collapse in 2001 witnessed one of the most renowned and complex digital investigations. The investigation involved two distinct and independent inquiries conducted by the US Securities and Exchange Commission, also known as SEC and the Federal Energy Regulatory Commission (FERC). These investigations required the analysis of extensive email and several accounting datasets, along with a substantial volume of paperwork. Various entities, including law enforcement organizations, forensic accountants, and digital detectives, contributed their specialized expertise to unravel the complexities of the case (Healy et al., 2003).

5.2.1 Network Forensics Challenges

When delving into the realm of network forensics, the presence of extremely large log files associated with firewalls, web servers or IDS (Intrusion Detection Systems), can pose a significant challenge. For instance, a firewall log for a usual organizational subdomain that comprises of 150 IP addresses may contain around 60,000-70,000 IP entries per hour. Extrapolating this to encompass the entire network over a week, the number of entries could easily surpass 150 million.

Considering the data logs provided by Internet Service Providers (ISPs) and the vast amount of information available on social networks like Facebook (with details on friendships, relationships and locations), Flickr (containing metadata like 'name and place' tags), and YouTube (housing tagged videos), the potential dataset expands even further.

This situation also brings to light the issue of "log time correlation", which has been explored by various researchers such as Irons and Lallie (2014), Elhalabi et al. (2014), Al-Hammadi and Aickelin (2008) and Al-Hammadi and Aickelin (2010) to perform behaviour analysis using server data logs and network activity from the perspective of an Intrusion Detection System (IDS) and detection of malware in systems.

5.2.2 Cloud Investigations Challenges

The importance of conducting further research on "cloud investigation", specifically focusing on evidence gathering techniques, has been emphasized by Beebe (2009) and ENISA or "European Network and Information Security Agency", (Al-Hammadi & Aickelin, 2010). Unfortunately, the field of cloud investigation has not received the level of research attention it deserves.

Birk and Wegener (2011) conducted an investigation into the challenges associated with evaluating different cloud platforms, offering valuable insights into the issues that arise when assessing SaaS, PaaS, and IaaS (Herrerias & Gomez, 2007). Various studies on cloud investigations by Reilly et al. (2011), Taylor et al. (2011), Birk and Wegener (2011), Lallie and Pimlott (Beebe, 2009), and Grispos et al. (Lallie & Pimlott, 2012) have highlighted the difficulties in applying guidelines and acquiring data stored in the cloud. They have also questioned the feasibility of current methods and recommendations for conducting cloud investigations. Furthermore, legal issues pertaining to cloud storage ownerships and its location have been raised

by Taylor et al. (Birk, 2011) and Lallie and Pimlott (Beebe, 2009), along with the challenges associated with data collection from different cloud system deployment models.

The process of locating and then imaging the data sources is perhaps one of the largest challenges in cloud investigation. Many distinct server farms and data stores may make up a public cloud storage infrastructure, which allows for the routing and storage of data on the fly (Reilly et al., 2011). Before being able to take a snapshot of the data, the investigator must pinpoint its exact location. This provides a unique forensics challenge for the investigator and has not, if at all, been previously examined. A fresh approach to the technology and techniques that researchers utilise to capture big data stores is needed in order to image such large datasets. In a digital investigation, establishing a chronological sequence is crucial, but it becomes more challenging due to uncertainties regarding data placement. The lack of data mobility records in file metadata poses difficulties for investigators in tracking the movement of data over time.

5.2.3 Big Data Challenges

Big data generally refers to issues with processing very large datasets that are frequently obtained with limited detail and therefore demand comprehensive, occasionally difficult approaches to process the data. The term ‘big data’ is a relative one that depends on the context, what is considered huge data today may not be in the future. During the 1980s, what can be considered as a notable data challenge, was encountered with the IBM 3850 MSS (Mass Storage System) that featured a 100 GB hard drive. This system was utilized to provide academics with rapid access to the U.S. Census data from 1980 (Taylor et al., 2011).

To be considered a big data problem, an analytical challenge typically needs to meet the criteria of “volume, velocity, and/or variety”. This means that the dataset is either too large (in terms of size or number of items) in order to be processed “effectively and efficiently” (volume), the extraction of valuable data from the dataset at best of times takes an excessive amount of time (velocity), most of the times dataset consists of diverse and complex data structures, such as imagery, financial transactions, computer access logs, and/or website navigation trees which represents “variety”. In all cases, it is essential to utilize “cutting-edge” technology to effectively process such datasets (Irons & Lallie, 2014).

It is argued that the challenges faced in digital investigations, while often involving handling

large amounts of data, should be considered as "huge data" issues rather than strictly falling under the realm of big data issues. The issues of volume, velocity, and variety in digital forensics differ from those encountered in other big data fields, such as analysing CardioDX data or dealing with data from the Large Hadron Collider (Irons & Lallie, 2014). While the challenges in digital investigations are significant and require careful management, they are not insurmountable. However, it is acknowledged that there may be scenarios where the digital forensics experts encounter problems that involve datasets of such magnitude, require an impractical amount of time to produce meaningful results, or contain data formats that current investigative tools and techniques are unable to handle (Schopp & Hillmann, 2020).

5.3 Digital Intelligence verses Intelligent Forensics

In light of the discussion regarding complex and large-scale digital investigations, it is essential to explore how investigations can be expanded by incorporating various techniques to provide more comprehensive details and potentially enhance efficiency. The field of digital forensics needs to broaden its methods and tools, develop more effective approaches to data analysis, and focus on extracting 'intelligence' from evidence sources to gain insights into both user behaviour and the incident being investigated. Other researchers have also acknowledged this need. For instance, Lai et al. (2013) proposed a conceptual framework for Internet pirates profiling, while Ieong (2006) introduced the FORZA or 'Forensics Zachman' framework, which considers multiple layers within an organization during investigations to integrate and apply diverse forensic techniques for solving complex problems in the domain. This reflects the growing interest in exploring digital forensic techniques to address challenging issues.

The importance of intelligence in criminal investigations has been extensively discussed in academic literature, with a particular focus on forensic intelligence. Ribaux et al. (2003) have contributed significantly to the subject matter through their extensive writings. It is essential to distinguish between intelligence and evidence when conducting criminal investigations. Evidence refers to the collection of facts or information that can be used to establish the truth or accuracy of a claim (Ribaux et al., 2003). On the other hand, intelligence is defined as the capacity to acquire and utilize knowledge and abilities (Ribaux & Walsh, 2006; Ribaux et al., 2010).

Intelligence, in this context, refers to relevant knowledge that may or may not constitute evidence in a committed or potential crime or scenario. It possesses certain characteristics, including value, which, as described by Ribaux et al. (2003), necessitates it being timely, accurate, and usable (Lai et al., 2013).

The term "digital intelligence" seems to have multiple interpretations. According to Ribaux et al. (2010), corporate managers can gain a significant competitive advantage by possessing what is known as "digital intelligence," which enables them to understand, assess, and effectively utilize digital technology to their benefit (Ribaux et al., 2010).

Stanhope and Dickson (2012) present a slightly different perspective on digital intelligence, defining it as the process of capturing, managing, and analysing data to gain a comprehensive understanding of the digital customer experience. This understanding drives the measurement, optimization, and implementation of marketing tactics and business strategies.

Stanhope's definition takes a business-customer centric approach, emphasizing the importance of digital intelligence in analysing and comprehending consumer data which is complex in nature. In his model, various types of data, such as emails, comments, ratings, social networks, display advertising, and transactions are collected as inputs to the “digital intelligence architecture”. These inputs are then processed, stored, and subsequently analysed to inform business decisions, as depicted in Figure 5.2 (Stanhope & Dickson, 2012).

This model can be adopted for cybercrime incidents related data as well. By combining the two concepts, we can define digital forensic intelligence as follows: It refers to valuable information acquired through the “forensic analysis” and processing of “digital storage” systems, which is utilized by law enforcement and various other investigating agencies.

Digital forensic intelligence is often obtained through both regular investigations and intelligence-driven operations, and it is typically stored in databases for future reference and analysis.

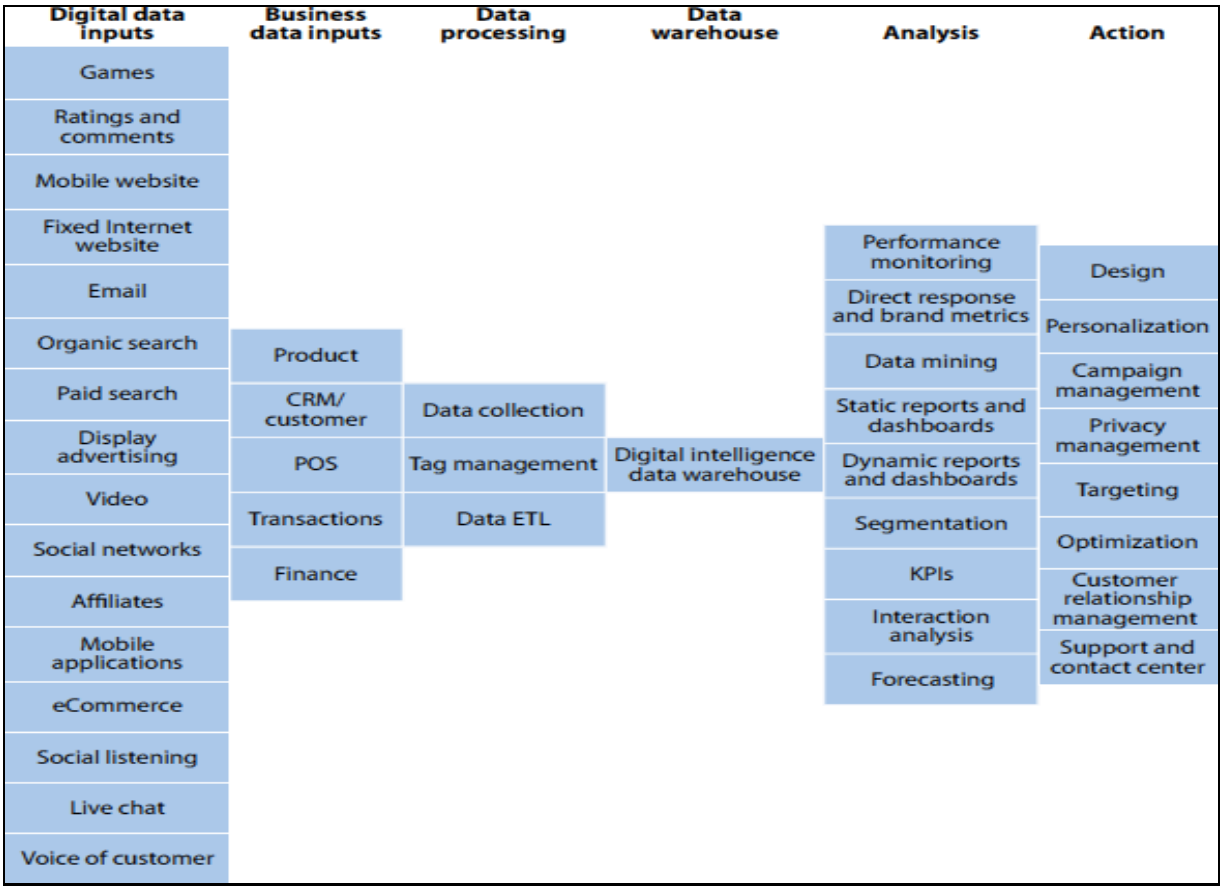


Figure 5.2: Stanhope's Digital Intelligence Architecture (Stanhope & Dickson, 2012)

5.3.1 Intelligent Digital Forensics

Intelligent digital forensics is an interdisciplinary approach that leverages technological advancements and utilizes resources in a more intelligent manner to effectively solve cases. It incorporates a range of technologies and techniques such as computational modelling, Artificial Intelligence (AI), and Social Network Analysis (SNA) within the field of Intelligent Forensics. This approach aids in streamlining digital investigations by focusing efforts and reducing the time spent in searching for digital evidence.

The proposed approach for examining incidents that are highly complex in nature is Intelligent Forensics. Intelligent Forensics can be applied in both proactive and reactive ways. Proactive implementation involves cognitive forensics to identify threats before an incident takes place. This proactive approach is utilized by intelligence and law enforcement agencies worldwide in their intelligence gathering efforts. Similarly, intelligent forensics can also be employed reactively after an incident occurs to investigate and gather evidence.

In a conventional investigation, the reactive use of intelligent forensics tools can enhance the gathering of additional intelligence, thereby aiding in the comprehensive examination of data sources. Various techniques, such as Social Network Analysis (SNA) and Artificial Intelligence (AI), can be applied during this phase. These methods offer valuable insights and contribute to the effectiveness of a digital investigation.

To tackle the complexity of extensive data sources in digital evidence, there exist viable solutions in the field of intelligent forensics. These solutions revolve around narrowing down the scope of investigation, such as using hashing techniques to identify stable or unaltered data sources. Another approach is to enhance the efficiency of investigative tools or leverage Intelligent Forensics itself.

Intelligent Forensics incorporates advanced procedures and strategies. While traditional digital forensics relied on searching for data, Intelligent Forensics retains this approach but also employs methods that enable queries to discover other queries, data to discover data, and data to discover relevant queries and patterns. This expanded approach in Intelligent Forensics allows for more comprehensive and efficient exploration of digital evidence.

Social Network Analysis

Social Network Analysis (SNA) utilizes mathematical methods, including graph theory, to analyse networks, particularly networks of people. One important metric in SNA is degree centrality, which measures the centrality of individuals within a network and provides valuable insight by examining a network structure.

A famous example is the Enron email dataset, which consists of nearly half a million retrieved emails spanning three and a half years, has been a significant area of research where the relevance of SNA in investigations has been demonstrated. Following the

dataset's public release in 2002, Mithas (2016) was able to discover hidden groups and also identified groups of individuals who were planning activities through communication channels quite discretely without explicitly stating their intentions to do so (Stanhope & Dickson, 2012). Uncovering of organizational structure and revealing the structure and relationships within the organization was also achieved. Examining of network dynamics and demonstrating how the networks of people evolve and change during critical situations was also uncovered. In the Enron case, for example, executives had formed tighter cliques, and distribution of information became less coordinated and open during the company's collapse as shown in figure 5.3 (Diesner et al., 2005). Use of AI techniques like SNA has proven to be a valuable tool in uncovering hidden patterns, understanding organizational dynamics, and studying the evolution of networks in complex investigations like the Enron case.

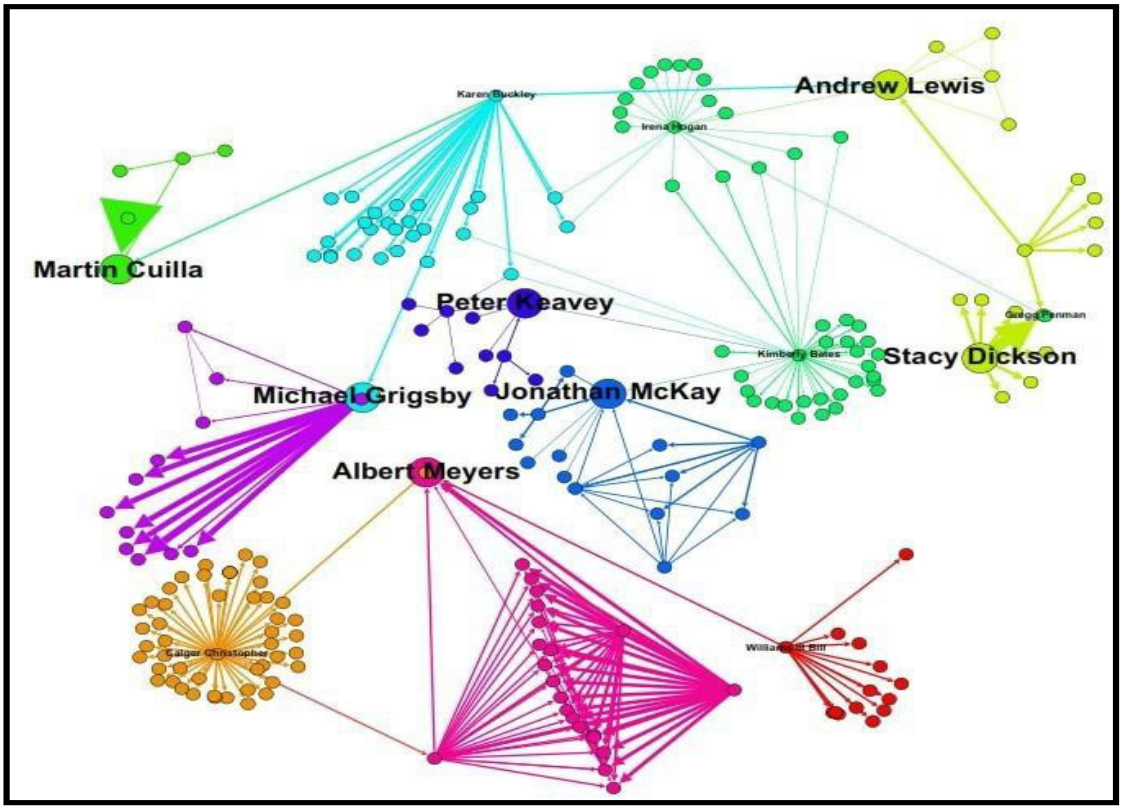


Figure 5.3: Social Network Behaviour Analysis of Enron staff emails (Diesner et al., 2005)

.

5.3.2 Artificial Intelligence and Digital Forensics

This section explores the potential application of Artificial Intelligence (AI) in computer or digital forensics, specifically focusing on how AI can enhance investigations in this field. The aim is not to delve into the debate surrounding the definition of AI but to assume that an AI system is a computer system that simulates some level of intelligence.

Intelligence plays a crucial role in various stages of computer forensics investigations, including the collection, preservation, analysis, and presentation of digital evidence. These aspects demonstrate how intelligence is employed throughout the investigation lifecycle. The competence and expertise of the computer forensics investigator are vital for the successful execution of investigations at each of these stages (Irons & Lallie, 2014).

The utilization of AI in digital forensic investigations is expected to provide investigators with valuable tools to tackle challenges related to complexity, speed, and volume of data. By leveraging AI, investigators can identify the most relevant areas for investigation while excluding areas that yield limited results. This approach has been partially employed in the past, such as using “hash algorithms” to eliminate inactive and "static" system files from investigations, aiding in streamlining the investigation process. The integration of AI technologies holds the potential to further enhance efficiency and effectiveness in digital forensic investigations (Johnson et al., 2022).

Knowledge representation plays a crucial role in digital investigations, assuming that the knowledge utilized by a digital investigator can be appropriately structured. In a similar vein, the concept of ontology from artificial intelligence can be employed when the knowledge is structured in a manner that enables reasoning.

By structuring digital forensic information, such as evidence, in a formal and organized manner, knowledge representation facilitates effective reasoning and analysis. Ontologies, which provide a formal structure for representing knowledge, can be utilized to enhance the organization and application of knowledge in digital investigations. This enables investigators to reason and draw meaningful insights from the structured knowledge, ultimately aiding in the investigation process.

One of the key challenges in applying artificial intelligence algorithms to computer forensics is the clear description of their usage. However, it can be beneficial to

conceptualize the use of artificial intelligence (AI) in computer or digital forensics as primarily focused on anomaly detection. This perspective has legal as well as computational implications (Irons & Lallie, 2014).

In the context of digital forensics, legal anomalies refer to actions that violate the laws of a specific jurisdiction, such as a case of underage drinking or driving. On the other hand, computational anomalies encompass various instances, including data that deviates from expected patterns in data streams or stored in static data storage, abnormally formatted data packets, and relational data indicating relationships that reflect unusual patterns. These anomalies can also extend to the computing machine abnormal states, such as data residing in an anomalous sector of the disk.

By focusing on anomaly detection, artificial intelligence algorithms can assist in identifying and flagging potentially significant deviations from normal patterns and behaviours, aiding in the identification of relevant digital evidence in computer forensics investigations. In the detection of anomalies, a wide range of artificial intelligence approaches can be employed. Knowledge-based systems can be developed to capture the expertise of legal professionals and detect unusual behaviour that may indicate legal violations. Neural networks have the capability to model the behaviour of users, distinguishing between appropriate and inappropriate actions, as well as identifying anomalous usage patterns for individual users. Data mining and Machine Learning (ML) techniques can be utilized to identify anomalies and uncover patterns of behaviour (Johnson et al., 2022).

To keep pace with the evolving field of digital forensics, systems can be designed to continuously learn and improve their performance. This can involve the integration of "big data analytics" and high-performance computing platforms usage. These approaches can be applied both reactively and proactively, automating certain steps in the identification, collection, preservation, and analysis of evidence. By leveraging artificial intelligence in these ways, digital forensic investigations can benefit from enhanced efficiency, accuracy, and the ability to handle the increasing volume and complexity of digital data (Du et al., 2017).

Recent developments have brought about significant changes in the landscape of cybercrime and digital investigations, necessitating the exploration of more effective and efficient processes and procedures. The evolving nature of cybercrime requires digital

forensic investigators to enhance their capabilities in “identifying, gathering, recovering, analysing, and documenting” digital evidence.

To keep pace with the changing digital landscape, it is crucial to consider the tools and techniques that are available to digital forensics investigators. Advancements in technologies, such as high-performance computing and cloud computing, as well as the widespread use of social media and mobile technologies, have created new environments for potential cybercrime. These developments present both challenges and opportunities for digital investigators, highlighting the need to adapt and leverage innovative approaches in their investigations.

By embracing new technologies, methodologies, and strategies, digital forensics investigators can enhance their ability to tackle cybercrime effectively. This includes staying updated on emerging trends, leveraging advanced tools and techniques, and continuously expanding their knowledge and skills to address the evolving challenges in digital forensic investigations.

In order to tackle the challenges posed by cybercrime, it is essential to leverage available resources effectively and surpass the limitations of current forensic tools. This study argues for the application of intelligent approaches to digital investigations, which have the potential to enhance the speed and effectiveness of investigations.

By incorporating artificial intelligence principles and practices into digital forensic intelligence and intelligent forensics, there is an opportunity to address the complexities of the larger domains in which cybercrimes take place. The application of intelligent procedures can offer valuable insights and capabilities that go beyond traditional investigative methods (Du et al., 2017).

Intelligent forensics has the potential to analyse vast amounts of data, identify patterns, detect anomalies, and provide actionable intelligence to investigators. By harnessing the power of artificial intelligence, digital investigations can become more efficient and proactive in detecting, preventing, and mitigating cybercrimes.

Embracing intelligent approaches in digital forensics can contribute to advancing the field and better equipping investigators to navigate the evolving landscape of cybercrime. It is a step towards building stronger defences and combating the ever-growing challenges posed

by cyber threats.

5.4 Application of AI to Digital Forensics

The introduction and use of "secure" technologies pose a danger to the simplicity and effectiveness of conventional methods of digital forensic inquiry. Digital investigations may take longer to complete as a result of technologies like the expansion of encryption, which might include encryption of full discs, secure processors, anonymous routing (onion routing) and secure network communication.

Given this wide range of challenges, it can be argued that it is vital to reassess the methodology that digital investigators employ to approach problems and a need for integrating intelligent techniques into the investigative process. In order to respond to developments in the illegal use of technology to cover or destroy evidences, it is necessary to re-examine established digital forensic practises and investigative procedures and also the applications of digital forensic techniques. This research attempts to accomplish this by taking into account intelligent forensics. An analysis of some of the most significant technology developments in recent years that provide unique difficulties for law enforcement organisations was required. This research examines intelligent notions and suggests that “intelligent forensics” can greatly enhance digital forensic investigations (Johnson et al, 2022).

5.5 Intelligent-Digital Evidence Extraction Protocol (I-DEEP) Modelling

Since it is noted from literature review and also analysis done in chapter-4 in context to incident response (IR), that many Digital Forensic (DF) investigation models are very complex to implement by investigators for incident response. Therefore, there was a need to design and develop a new protocol that covered all core stages of DF investigation process, mainly evidence gathering, in agile manner to facilitate first responders and speed-up the process. A novel protocol model called Intelligent-Digital Evidence Extraction Protocol (I-DEEP) has been proposed to achieve these objectives. The detailed protocol model is displayed in figures-5.5, 5.6 and 5.7, that represents all necessary steps required to carry out an investigation and evidence gathering in a more efficient and ‘agile’ way.

5.5.1 Theoretical Underpinnings of the Newly Proposed Protocol Model

Since this has been thoroughly discussed under chapter-2 that one of the earliest attempts to standardise digital forensic process were made in Digital Forensics Research Work-Shop (DFRWS) held in Utica in 2001, where goal was set to define a framework for Digital Forensic Investigation Process. It was understood that computer forensic analysis is aimed to facilitate the law enforcement community and it must follow statutory and regulatory guidelines established for traditional forensics disciplines. Therefore, it is imperative that digital forensic tools and technologies adhere to sound scientific methodologies which can generate credible digital forensic evidence and it can stand in the court of law (DFRWS, 2001).

The increasing prevalence of computing and internet technology in all domains of modern society has made digital forensic research relevant across various sectors. Whether in business and industry, military establishments, or law enforcement and judicial bodies, digital forensics has become increasingly significant. These organizations heavily rely on digital forensics to address the growing occurrences of cybercrime.

The importance of designing and developing new models/protocols for digital forensic technology was recognized at the DFRWS (2001) conference. This is eminent and depicted in Nucleus of Digital Forensic Research (figure-2.1), which highlights the main focus on 'digital forensic research'. It was also acknowledged that academic researchers can analyse the existing processes to identify technological gaps and propose focused research to fill those gaps.

It has also been noted that the Nucleus of Digital Forensic Research Process Model (DFRWS, 2001) as discussed in section-2.3 (figure-2.1), do not accurately place the key characteristics of digital forensic research in present context. These anomalies are especially visible at the intersections of the Venn diagram depicting the digital forensic domains like law enforcements, business and industry, court of law etc.

A refined and extended model was created and presented in section-2.5.6 (figure-2.7) and is also discussed in this section again. This refined and extended DFRWS Model forms the fundamental theoretical background of the novel (I-DEEP) Protocol Model developed in this research.

5.5.2 Critical Analysis of Linear Process Model (DFRWS, 2001)

There have been several process models proposed and developed since the early days of DFRWS Conference held in 2001 in Utica, New York, as explained earlier in section-2.4 and some of them have been presented at many DFRWS conference proceedings since then. DFRWS has been instrumental in addressing various needs for practitioners and made major strides in developing standards, investigation methodologies, technological advancements in research tools and development of digital investigation frameworks and models. Some of them have been reviewed in this study earlier in chapter-2 (section-2.4 and 2.5).

DFRWS 'Linear Process Model' provides the theoretical background for developing new protocol model as it is the most accepted model in the DF research arena and covers the fundamental concepts applied in DF investigation phases from 'Identification' to 'Presentation'. This model consists of phases that are universally accepted by courts of law and also widely adopted in the domain of digital forensics. Although, it has been observed in this research that present form of this model lacks critical aspect of 'decision making', importance of analysing trends in evidential data and implementation of cutting-edge technologies i.e. artificial intelligence and machine learning. As this model mainly focusses on identification and collection of evidence, analysis of evidence and its interpretation for quick decision making is left to the investigators' own personal and professional ability. If the investigator is not competent enough, this may lead to inconclusive investigations that may lead to loss of valuable time and resources. This aspect is identified in this study as a 'research-gap' and 'triaging' and 'predictive analysis' of cybercrime data has been proposed as prospective solutions to mitigate this issue and improve decision making in DF investigations. Use of Artificial Intelligence (AI) and Machine Learning (ML) have been integrated as a platform technology to achieve this goal in the new artifacts' design and development.

In order to maintain the standardisation and consistency, the fundamental concepts of existing models have been considered and also further enhanced in the form of extension of most popular models like DFRWS 'Linear Process Model'. The newly developed model is referred to as 'Extended Linear Process Model' and explained in section-2.5.7 (Figure-2.8). The main addition (extension) in this model is the introduction of the concepts of 'Optimisation', 'Data reduction', 'Predictive analysis', 'Visual representation of data' (using scatter charts, matrix plots and whisker plots) and improvements in 'Decision' phase, which involved the addition of 'Triaging' and 'Prediction' into the previous model. While answering the need for these

interventions, the ontological descriptions of these constructs and implementation is discussed comprehensively in forthcoming chapters-6 and 7.

5.5.3 Digital Field Triage (DFT) Model

Hitchcock et al. (2016) proposed the Digital Field Triage (DFT) model with the aim of delegating some of the primary tasks performed in the digital forensic domain to non-digital evidence specialists. The model has two primary objectives: (i) to improve the efficiency of investigations by providing timely access to digital evidence, and (ii) to reduce the backlog of cases at forensic laboratories. The DFT model is based on the work of Rogers et al. (2006) and is comprised of four levels: planning, assessment, reporting, and threshold. However, the DFT model comes with inherent risks, including management, training, and lack of supporting tools.

The effective implementation of the DFT model relies on two crucial factors - efficient management and continuous training. The management tools employed should facilitate smooth functioning of the DFT model (as illustrated in figure 2.6), which equips non-experts in digital investigations with the necessary knowledge, skills, and capabilities to conduct limited forensic activities (M. Rogers et al., 2006). Evidence and Field Triaging can provide timely response to a cybercrime (e.g. involving paedophiles and child abductors), where delays can cause serious life threats and a quick action becomes imperative.

Digital Triaging involves quick leads into investigations based on type of cybercrime, various possibilities or scenarios at the crime scene, certain usable evidence and investigative team expertise (M. K. Rogers et al., 2006).

5.5.4 Critical Analysis of Digital Field Triage (DFT) Model

Although DFT Model can be very effective to speed up the DF investigation process and assist in decision making but has its inherent shortcomings. There exist three primary conditions that must be met for the DFT model to operate effectively. Firstly, it cannot function independently and must operate in conjunction with a parent Technological Crime Unit (TCU). Secondly, the forensic integrity of the digital evidence must be preserved at all times, which may get affected as non-technical team is also involved. Lastly, a DFT evaluation cannot substitute a thorough TCU investigation.

5.5.5 Advanced Data Acquisition Model (Adams et al., 2013)

Adams et al. (2013) has proposed Advanced Data Acquisition Model (ADAM) which provides a conceptual framework for data acquisition and covers all digital investigation phases in substantial detail. This model elaborates on specific DF process of data acquisition. This is a generic model that builds on DF process models like DFRWS (2001), IDIP by Carrier and Spafford (2004) and other generic models. ADAM uses Unified Modelling Language (UML) as a descriptive language, although it does not strictly follow a precise syntax of UML. Main focus is to create a generic model framework for data acquisition. It is a three-stage model which is built upon underlying principles summarised here.

The preservation of the original data is a crucial principle in digital forensics. If it is necessary to make changes to the original data, the impact of these actions should be clearly documented, and the justification for such changes should be provided. A comprehensive record of all activities related to the acquisition and handling of the original data and any copies should be maintained. This includes adherence to the appropriate rules of evidence, such as maintaining a chain of custody record, and utilizing verification processes like hashing.

Digital forensic practitioners should only engage in activities that are within their expertise and knowledge. They should not attempt tasks that exceed their capabilities or understanding.

The personal safety of the digital forensic practitioner should be given utmost consideration during their work. All aspects of personal safety should be taken into account, to ensure a secure and protected working environment.

5.5.6 Critical Analysis of ADAM

ADAM is a generic model that lacks specifics. Although it covers data acquisition process in comprehensive detail and tries to build on the legacy of established models, it is not technology specific and might pose challenges in implementation. This model provides a very complex structure to practically implement all phases and does not provision any latest technological interventions to overcome the challenges posed by the need to analyse vast amounts of data, identify patterns, detect anomalies, and provide actionable intelligence to investigators. By harnessing the power of artificial intelligence, digital investigations can become more efficient and proactive in detecting, preventing, and mitigating cybercrimes. The novel protocol design conceptualised in this research uses ADAM (2013) as a reference model but provides considerable improvements over the base model to bring agility and practicality to the model.

5.5.7 Proposed Improvements in Digital Forensic Investigation Models

Since it was an observation that digital forensic investigation process models described in section-2.4, have specific gaps that demands a revamp of these models. An improved model of Nucleus of Digital Forensic Research Process is provided by the researcher in figure-2.7.

The new construct (Nucleus of Digital Forensic Research Process Extended Model) highlights that Computer Emergency Response Teams (CERT) form the core of 'Defence and Security' and 'Business and Industry' domains. Therefore, while emphasizing the central role of CERTs, it also highlights that both these domains are equally critical when it comes to cyber-security and incident response (IR) planning.

The Nucleus Extended Model also emphasize that Digital Forensic (DF) frameworks, standards and tools play an indispensable role in cybercrime investigations, law enforcements and court or legal proceedings. Therefore, they also become key focus areas of DF research.

5.5.8 Extended Linear Process Model

As discussed in section-2.5.7, a Linear Process Model adopted by DFRWS (2001) represents standardised phases and activities associated with those phases. Since we discussed earlier that the model is generic in nature and has not been modified or extended to accommodate new demands of increased efficacy, predictive analysis and decision making, therefore this provides an opportunity to fill the gaps in the existing model. An extended form of the 'Linear Process Model' discussed in detail in section-2.5.7 and represented in figure-2.8, has been created by the researcher in order to bridge the gap and also provide theoretical underpinning for design and development of a novel investigation protocol model with AI based framework. This new extended model has been incorporated in I-DEEP protocol design and its implementation is demonstrated via a working prototype to bring intelligence in digital forensic tools and efficiency into existing investigation process.

In the 'Extended Linear Model', while other phases correlate to standard DF investigation activities, the new constructs like 'Optimisation' of IR process is introduced, thus making it more agile. AI based 'Predictive Analytics' have been introduced to achieve better efficiency in the investigation process. This addition is made in the 'Analysis' phase of the 'Linear Model'. Furthermore, 'Visual Representation' of dataset and AI algorithm prediction 'outputs' in the form of scatter charts, correlation matrix plots, and whisker plots can be used to visually represent the data trends and results. These theoretical underpinnings will provide the template

to implement methodological and functional improvements while designing and developing new artifacts e.g. novel investigation protocol with intelligent framework and a working prototype based on these artifacts.

The improved or extended version of 'Nucleus of Digital Forensic Research' elaborated earlier in section 2.5.6 (figure-2.7) and re-designed 'Extended Linear Model' (figure-2.8), both show pragmatic adaptations to address new requirements arising in the field of digital forensics. Therefore, these extended and redesigned models are discussed again in this section to highlight the improvements. These changes were imperative to fill the gaps uncovered in literature review in this research and also provide theoretical background for improvements in the existing models/frameworks. Furthermore, they offer a 'conceptual roadmap' for this research to design and develop new artifacts (a novel investigation protocol with AI based framework) to assist first responders. These extended models address the research problem and maps to the issues raised in research sub-question-5.

Digital Forensic process consists of various phases as described in 'Linear Process Model', which comprises of stages from identification to preservation of evidence and finally decision stage. These phases have been discussed in detail in section 1.2.8. The main improvements introduced in this model is the 'Optimisation' process to provide agility, 'Data reduction' to perform triaging and quick decision making, 'Predictive analysis' using AI algorithms, 'Visual representation' of data using scatter charts, matrix plots and whisker plots and improvements in 'Decision' phase, with the help of 'Prediction' and 'Triaging'. These new constructs added to the existing models form the basis for the design of a novel 'digital evidence extraction protocol' and also fills the 'gaps' existing in present models.

5.5.9 Extended Digital Field Triage Model

Since, we observed that DFT model comes with inherent risks, including management, training, and lack of supporting tools. Also, the model lacks use of cutting-edge technology implementations like Artificial Intelligence (AI) and Machine Learning (ML) capabilities. Therefore, an Extended DFT Model is proposed that integrates AI predictive modelling and 'Triaging' into the existing model. In order to perform 'Triaging' effectively, a severity score is applied to cybercrime index/scale with a formula (severity_score='Severity index/level' x 'number of incidents') and thereafter determining the 'critical' or 'non-critical' status of cybercrime incidents for triaging and decision making. This formula is used in creating new customised AI algorithms described in chapter-7 (section 7.9) and further implemented in

newly designed AI based 'intelligent framework' for predictive analysis. This research while implementing some aspects of this model into the novel investigation protocol model, also introduces some functional improvements like providing 'crime severity weightage' to the cybercrime incidents and then calculating a 'Triage Score'. These two constructs have been implemented in the novel protocol (I-DEEP) and the intelligent framework (DIF^2) discussed in upcoming chapters. These constructs help in 'predictive modelling' and 'triaging' process, thus providing improved decision-making capabilities to be incorporated in the artifacts.

Some of the existing aspects of the Digital Forensics Field Triage Process Model (DFFTPM) has been used as a base model in designing I-DEEP protocol and field triage implementation in the intelligent framework (DIF^2) in this study. The extended model constructs help to build the conceptual roadmap in implementing 'Triaging' process and thus contribute in decision making improvements. Figure-5.4 illustrates these new additions to the model.

Figure 5.4: Extended Digital Field Triage Model (new artifact)

5.6 Intelligent Digital Evidence Extraction Protocol (I-DEEP) Design

The main objectives for design and development of new protocol was to make it more agile, incorporate AI technology into the DF process and drive this towards intelligent forensics. This protocol addresses major processes that are crucial to investigation such as requirement determination and gathering, setting requirement parameters, authorization, access to premises, number of locations, determining physical constraints, time constraints, data constraints in Stage-I. Stage-II of the model addresses safety of data, equipment, personnel, maintaining activity logs and charts etc. Stage-III is implementing AI based techniques to the dataset obtained from collection of evidence and various cybercrime data.

5.6.1 I-DEEP Stage-I: Identification & Planning Phase

During the initial phase of a digital forensic investigation, fundamental questions need to be addressed, such as the existence, location, and form of the evidence. This phase is guided by Locard's exchange principle, which states that when two objects interact, they leave traces on each other. In the digital realm, this principle holds true as well, where interactions between systems result in the exchange of data and the creation of a data trail.

To illustrate this concept, consider a scenario where an individual visits a website. The web-server and the web-browser on the individual's computer engage in data exchange, leaving traces of their interaction. The web-server logs typically contain the individual's IP address, along with other details such as date and time stamps. Additionally, the website may store cookies on the individual's computer, further contributing to the data trail. It is important to note that most computer systems have storage memory capable of retaining data, which can be crucial for digital forensic investigations (Johansen, 2017). This initial phase sets the foundation for subsequent steps in the investigation process, allowing investigators to identify and acquire relevant digital evidence. This phase illustrated in figure-5.5, maps to the Identification Phase of DFRWS (2001) framework. It represents the new artifact ‘Stage-I of I-DEEP Model’. Although it adheres to the ‘Identification of Evidence’ phase of the DFRWS Model, it further adds the notion of ‘Planning’ aspect and emphases this aspect in detail. This phase also lays emphasis on ‘Requirements Gathering’ and highlights the need for ‘determining requirements’, ‘planning and preparing for desired outcomes’, ‘identifying and setting requirement’ parameters. ‘Authorisation’ requirement is also very crucial where the investigators need to obtain permits and court orders. Many jurisdictions may require separate permits for site and device seizures.

Stage-I: Identification & Planning Phase

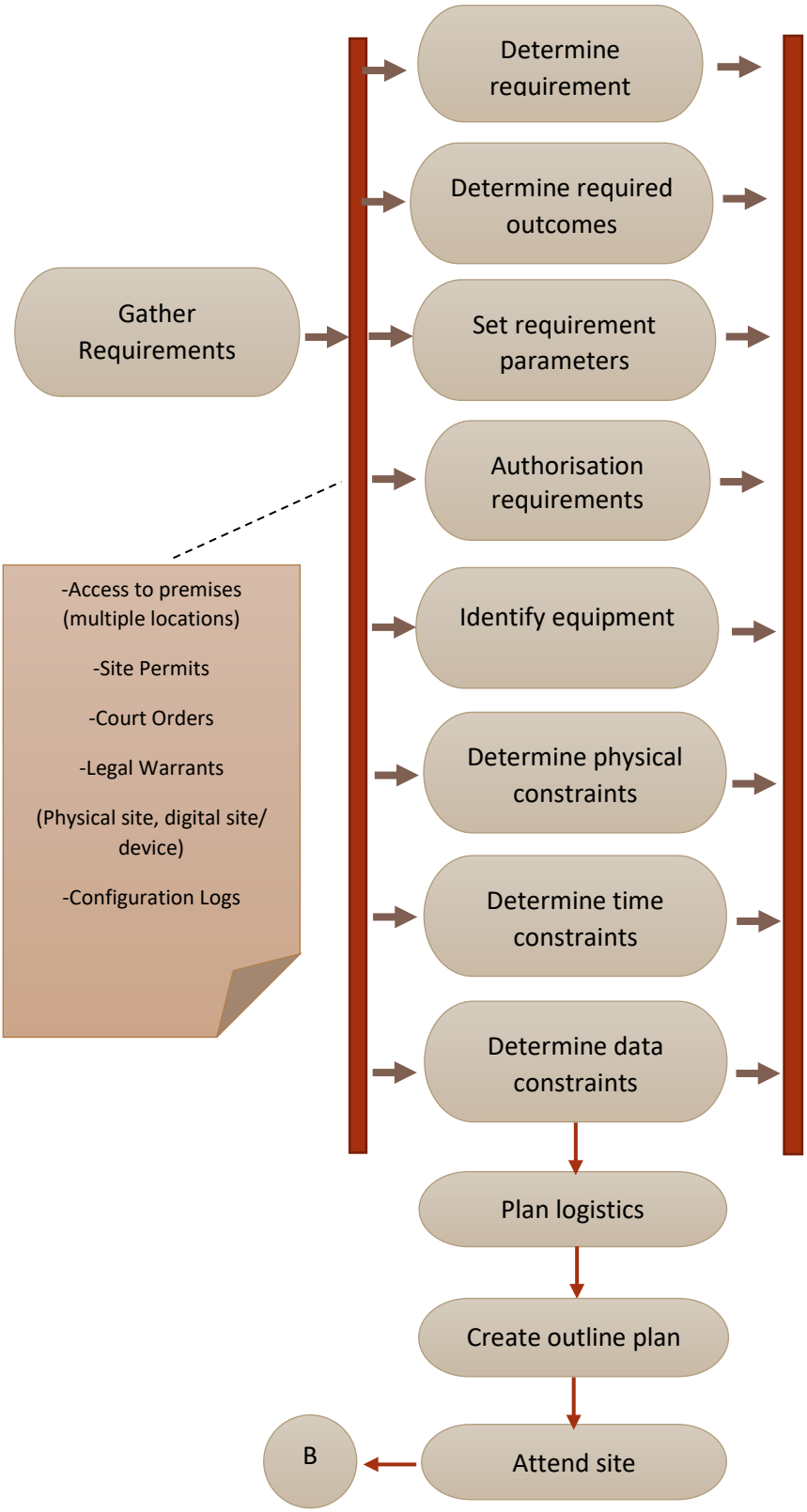


Figure 5.5: Identification and Planning Phase (I-DEEP: Stage-I)

Determine Requirements

According to Casey et al. (2009), and also emphasised by Adams et al. (2013), the acquisition of digital data involves "Authorisation and Preparation". Casey et al. (2009) emphasizes the importance of planning in cases involving computers, although he does not provide a specific plan name. In an ideal scenario, detailed knowledge of the digital environment would enable the creation of a precise plan that could be executed on-site. NIST (2023) framework and Johansen (2017) highlight the need for comprehensive information gathering about the computers, their types, operating systems, and connections before entering the scene.

Determine Requirement Parameters

However, in practice, digital forensic examiners often have limited details about the computer systems, data quantity and location, types of hard disks, or operating systems involved. This makes it challenging to create a detailed plan beyond a rough outline. Brown (2015) argues that due to the scarcity or inaccuracy of initial information, the planning stage should focus on preparing for various likely scenarios. It is crucial to have a playbook that provides a framework for investigators to work from, similar to a sports team coach's playbook containing all the planned plays.

Therefore, main parameters that can be identified as critical in digital forensic investigations are 'Identification' of evidence, 'Data Collection and Preservation', 'Legal Authorisation'. 'Chain of Custody', 'Documentation and Logging'. All these parameters have been considered and processes involving these parameters have been addressed in the Stage-1 of I-DEEP design. Since the objective is to make the investigation process more 'agile', therefore the streamlining of investigation phases is necessary and ease of implementation is the main driving force for I-DEEP design.

Determine Required Outcomes

Even when some information about the encountered computer systems is available, allowances must be made for errors or inaccuracies in that information. The planning stage plays a fundamental role in the process of acquiring digital evidence and is commonly employed across different environments where digital forensic personnel operate. Therefore, it is crucial to identify the equipment and environment in which data extraction or acquisition happens and

prepare in advance. Certain outcomes can be planned in advance that pre-empt the investigation procedure and help in creating an outcome strategy (Adams, 2013).

Authorisation Requirements

Before delving into the intricacies of the process, it is essential to address the issue of ensuring that the digital forensic practitioner possesses the necessary authority to carry out their work. This authority can be comprised of several distinct elements: internal authorization from the organization providing the services, legal authorization, and external authorization from the owner of the resources containing the material to be acquired (Johansen, 2017).

Marcella (2021) outline a comprehensive list of fundamental steps for a 'cyber investigation,' with the first step being 'Obtain proper authorization'. They emphasize the critical nature of this step and provide detailed coverage of its importance in digital crime investigation.

Determine Constraints

There might be various constraints that may impact an investigation adversely. Timing is indeed an important consideration during the planning stage of a digital forensic investigation. While some authors may not explicitly mention timing as part of the initial preparation, it is a practical and crucial aspect to consider. The choice of techniques and methods employed in the investigation should consider the time constraints associated with the case (Adams et al., 2013).

Time Constraints

In some instances, authors focusing on in-house digital forensic practitioners may not explicitly address timing aspects, possibly assuming that the practitioners are already familiar with the time constraints within their organization. However, it is important for digital forensic practitioners to consider the time limitations and factors that may impact the investigation, such as legal requirements, organizational priorities, and operational needs.

Taking timing into consideration during the planning stage helps ensure that appropriate resources, techniques, and methods are selected to effectively and efficiently conduct the investigation within the given time constraints. It allows the forensic team to allocate sufficient time for each phase of the investigation, including acquisition, analysis, and reporting, while meeting legal and organizational requirements.

Physical Constraints

Physical constraints can span multiple dimensions ranging from environmental, physical or site location to hardware and other resource limitations. Limited access to the physical hardware or devices being investigated can be a constraint. For example, if the hardware is damaged, encrypted, or located in a remote or inaccessible location, it may pose challenges for forensic examiners. Incompatibility between forensic tools and the hardware being investigated can be a constraint. The lack of suitable connectors or adapters to interface with diverse hardware configurations may hinder the extraction of evidence.

Data Constraints

Data constraints in digital forensic investigations refer to limitations or challenges associated with the nature, volume, accessibility, and integrity of digital data that investigators encounter during the investigative process. Several factors can impose constraints on the collection, analysis, and interpretation of digital data in forensic investigations like 'Data Volume'**.** The sheer volume of data that needs to be analysed can be overwhelming. Large datasets may require significant computational resources and time for examination, potentially slowing down the investigative process. Data encryption can be a significant constraint if the investigator does not have the necessary decryption keys. Encryption techniques can prevent access to the content of files or communication, hindering the examination of crucial evidence. 'Remote or cloud data' can impose significant challenges as more data is stored in the cloud and on remote servers, investigators may face challenges in accessing and retrieving digital evidence from these platforms.

Court Orders and Warrants

When conducting digital forensic investigations, court orders and warrants play a crucial role in determining the scope and timeframe of the acquisition activities. These legal documents provide authorization for the investigation and impose specific requirements and limitations.

In some cases, court orders may include strict time limits within which the acquisition activities must be carried out. These time limits define the duration during which the forensic examination can take place. It is important for digital forensic practitioners to adhere to these time limits and ensure that all necessary data is acquired within the specified timeframe. Similarly, warrants issued by a court or other legal authority may also contain restrictions and

time constraints. These warrants outline the authorized scope of the investigation, including the specific locations, devices, or individuals subject to examination. The warrants may specify the timeframe within which the acquisition activities must be completed.

Digital forensic practitioners must carefully comply with the terms and conditions outlined in court orders and warrants. Failure to adhere to these requirements may compromise the admissibility and credibility of the evidence in court. Therefore, it is essential to manage the acquisition process effectively, ensuring that all necessary data is acquired within the stipulated time limits while maintaining the integrity and chain of custody of the evidence.

Maintain Configuration Data and Logs

It is imperative to maintain configuration logs and identify specific technical requirements before starting the data extraction process. An important aspect of 'Identification and Planning' phase is to prepare the technical requirements (forensic tools, devices needed for evidence collection and preservation) that would be crucial for evidence collection and maintaining its integrity that can be presented effectively and defended legally in the court of law.

The I-DEEP Model Stage-1 supports multitasking and therefore also emphasise that some of the activities in 'Identification and Planning' phase can be carried simultaneously (in parallel) and iteratively to reduce timeframes and achieve best outcomes as illustrated in figure-5.5.

5.6.2 I-DEEP Stage-II: Digital Evidence Extraction Phase

This phase maps to the "Preservation, Collection, Examination" phases of DFRWS (2001) framework and collectively named as 'extraction' phase. Figure-5.6 represents this new artifact of I-DEEP Stage-II: Evidence Extraction Phase. This model highlights safety issues associated with the data or evidence extraction process and also issues associated with personnel. Equipment isolation strategy, personnel and data safety must be considered prior to confiscation of device and evidence extraction.

Preservation of Data

Once digital evidence is identified, it is crucial to preserve its integrity by preventing any modification or deletion. This requires isolating the computer system from the network and implementing various controls, including physical (perimeter) and logical (network access

control) measures. Restricting user access to the system is essential to prevent tampering with the evidence.

Preserving the evidence involves implementing measures such as data segregation and data protection. Technologies like system imaging or snapshotting can be utilized to capture a forensic copy of the system at a specific point in time. This ensures that the original state of the system and its data are preserved for analysis (DFRWS, 2001), (Johansen, 2017).

In addition, access to devices should be restricted to prevent any unauthorized modifications or tampering with the digital evidence stored within those devices using block writers. By implementing these preservation techniques, the integrity and admissibility of the digital evidence can be maintained throughout the investigation process.

Stage-II: Digital Evidence Extraction Phase

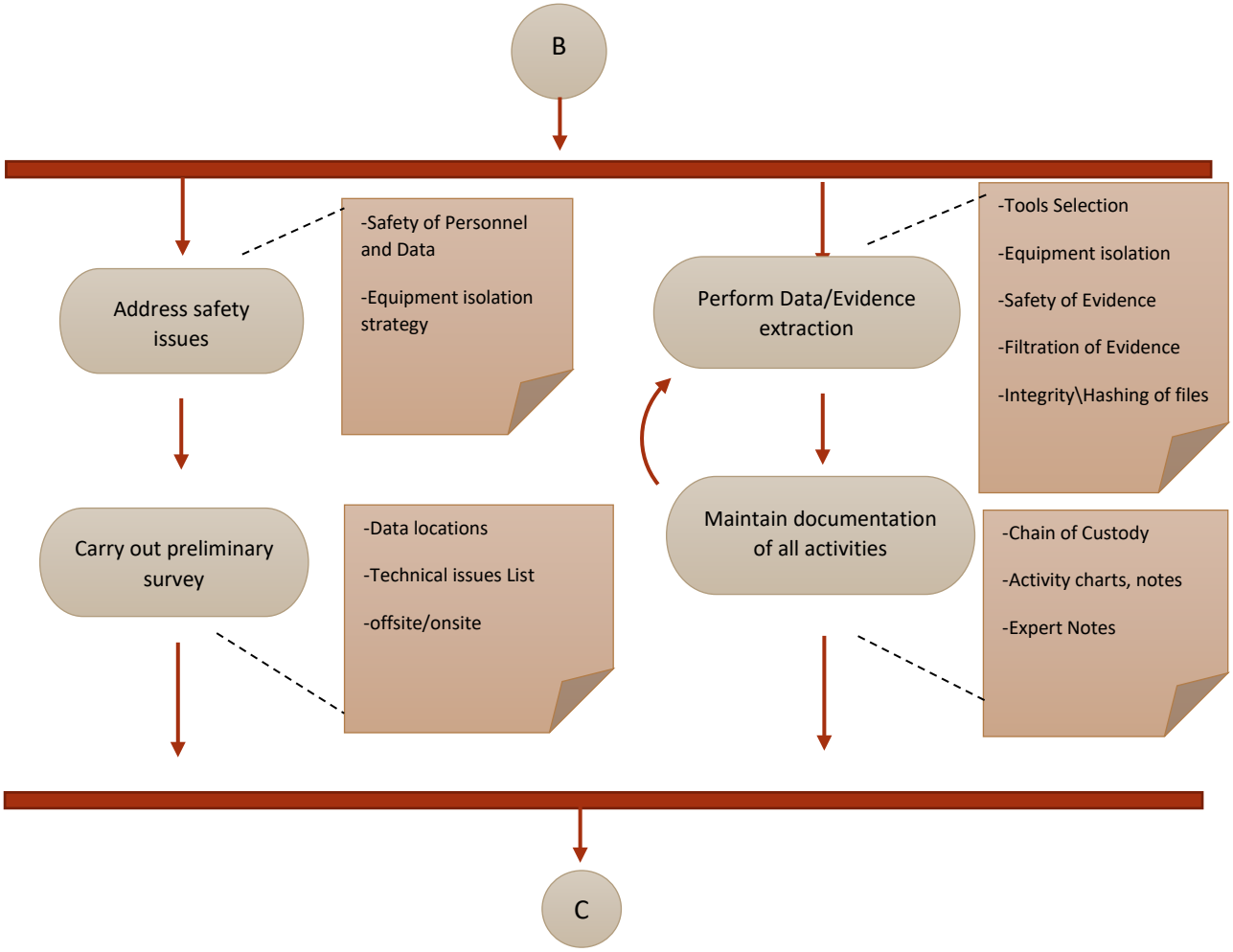


Figure 5.6: Digital Evidence Extraction Phase (I-DEEP Stage-II)

Stage-II of I-DEEP model highlights the approach of data\evidence filtration in order to obtain best results in minimum time span possible from examination process. This process must be carried out in an 'iterative manner' to achieve desired goals and should be repeated until the objectives and 'desired outcomes' of the investigation process are not met for the set benchmarks, that they are acceptable in the court of law. The selection of most suitable tools and forensic devices is imperative to achieve the objectives and 'desired outcomes' of the investigation process. Therefore, the model provides a process framework for achieving agility in the investigation process using iterative process approach.

Data Integrity: File Hashing

File hashing is a crucial technique used in digital forensics to ensure the integrity and authenticity of files during investigations. It involves generating a unique digital fingerprint, often referred to as a hash value or hash code, for a specific file. This hash value is derived from the file's content using a mathematical algorithm (Carrier & Spafford, 2004; Johansen, 2017).

The process begins by selecting an appropriate hashing algorithm, such as MD-5 (Message Digest-5), SHA-1 (Secure Hash Algorithm-1), SHA-256, or others. The chosen algorithm takes the binary representation of the file and applies a series of complex mathematical operations to produce a fixed-length hash value, typically represented as a hexadecimal string.

The generated hash value is unique to the specific file, meaning even a minor change in the file's content will result in a different hash value. This property makes file hashing a valuable tool for verifying file integrity and detecting any unauthorized modifications or tampering.

In digital forensics, file hashing serves several purposes. It allows investigators to 'Verify Data Integrity'. By comparing the hash value of a file obtained during the investigation with a previously generated hash value, investigators can determine if the file has been altered or corrupted (Johansen, 2017).

Hash values can be used to create a database of known files or known malicious files. This enables investigators to quickly identify and categorize files based on their hash values, assisting in the identification of potentially relevant evidence. Hash values can be used to verify

the authenticity of files obtained from different sources. By comparing the hash value of a file provided by a source with a known hash value, investigators can ensure the file has not been tampered with or substituted.

Hash values can also be used to identify duplicate files within a storage medium or across multiple sources. This is particularly useful in identifying duplicate files that may be relevant to an investigation, saving time and effort in reviewing identical data.

Extraction of Data or Evidence

During the process of acquiring digital evidence, cybercrime investigators or examiners must be aware of the volatile nature of certain types of evidence. Volatile data refers to information that can be easily lost or altered if proper precautions are not taken. For example, evidence stored in random-access memory (RAM) can be lost if a system is powered down or disconnected. Similarly, network devices may lose data related to active connections or log data.

To address the handling of volatile digital evidence, the Internet Engineering Task Force (IETF) has provided guidelines in RFC-3227 titled "Guidelines for Evidence Collection and Archiving". These guidelines specify various types of volatile data, including registers and cache, ARP cache, routing table, RAM data, temporary system files, disk data, remote logged data, and physical and network configurations and topologies. It is crucial for investigators to be mindful of the volatility of collected data to ensure its preservation (BREZINSKI & Killalea, 2002).

The subsequent steps in the digital forensic process involve handling the evidence, ensuring its security, documenting the collection process, and maintaining the chain of custody. Documentation is a critical aspect and involves recording all visible data to assist in examining and reconstructing the crime scene. This may include taking photographs, creating sketches and maps, and accurately recording all relevant details. Precise documentation helps establish the integrity of the evidence and provides a comprehensive record of the crime scene.

Examination of Data or Evidence

During the examination phase of digital forensics, specific techniques and tools are utilized to extract relevant data from the seized evidence. This process is often referred to as e-discovery,

where investigators extract and analyse pertinent information from the seized device or the acquired image. If the examination involves an active network, Secure Shell (SSH) packets from the network are captured and analysed for potential evidence. Throughout the examination process, it is crucial to maintain data integrity and preservation. Examiners must exercise due diligence to ensure that the evidence remains untampered and uncontaminated. Failure to adhere to proper protocols and precautions can result in the evidence becoming compromised and ultimately unusable for legal proceedings (Nortjé & Myburgh, 2019).

In order to perform data and evidence extraction, appropriate hardware and software tools are required. The devices may be confiscated or data imaging may be done if the systems cannot be removed from the premises and taken to digital forensic labs for examination.

In the laboratory, having an adequate number of computers and hardware is crucial to carry out the necessary tasks in digital forensics. These tasks include creating hard drive images and processing large amounts of data, which require the use of a forensics-enabled computer with sufficient RAM. It is generally recommended to have 32 GB of RAM or more for efficient performance.

In addition to the need for memory and processing power, forensic workstations also require specific storage configurations. A primary Operating System drive is necessary to accommodate digital forensic software tools, while a secondary drive is dedicated exclusively to storing the evidence collected during investigations. It is recommended that the secondary drive has a minimum capacity of 2 TB to handle the extensive amounts of data involved. While examiners may need access to a computer connected to the internet for certain tasks, it is essential to ensure that the forensic workstation remains disconnected from the internet to maintain security and prevent any potential evidence tampering. It is advisable to use a separate machine for online research or report writing.

Another critical device is the forensic toolkit with a physical write blocker. This device acts as a bridge between the evidence hard drive and the forensic imaging machine, ensuring that no data can be written to the drive containing the evidence. Using a physical write blocker provides the examiner with confidence that no unintended modifications are made to the evidence during the imaging process, unlike direct connections through USB drives or Thunderbolt connections. Having the appropriate hardware and devices in the forensic laboratory is essential to ensure the integrity and reliability of digital forensic examinations.

When DF investigators are required to conduct off-site investigations, they face the challenge of performing investigations without the support of a dedicated digital forensics lab. To address this challenge, investigators can prepare jump kits that contain all the necessary hardware and software for conducting incident response tasks. These kits should be portable, securely stored in hard-sided cases, and readily available for immediate use.

The jump kit should be replenished after each investigation, ensuring that all items used in the previous incident are replaced. This includes properly reconfiguring hardware and software tools to ensure their availability when needed. By keeping the jump kit up to date, investigators can ensure they are always prepared to respond to incidents.

For efficient hard drive imaging, the jump kit should include a forensic laptop with a minimum of 32 GB of RAM. The laptop should have forensic softwares installed, along with at least one forensic Linux Operating System platform such as SIFT, KALI, or CAINE. Multiple CAT-5 cables of varying lengths should be included to facilitate network access. A physical write blocker is essential for imaging hard drives encountered during investigations. Additionally, multiple USB-compatible hard drives with capacities of 1 TB or 2 TB should be included for imaging potentially compromised systems.

To ensure the collection of forensically sound evidence, the jump kit should contain several large-capacity USB drives (64 GB) for capturing RAM contents, offloading log files, and storing other relevant information. It is also beneficial to include a bootable CD/DVD or USB containing various Linux distributions, which can be useful in specific cases. On-site, evidence bags or boxes should be readily available to secure and transport evidence if necessary. Anti-static bags are crucial for safely transporting seized hard drives as evidence.

Including chain of custody forms and a forensic toolkit in the jump kit is essential for a successful incident response. The chain of custody forms are needed for proper documentation and tracking of evidence, ensuring its integrity throughout the investigation. The toolkit should include tools like pliers, screwdrivers, and a flashlight, enabling first responders or investigation teams to cut connections, remove hard drives, or access hard-to-reach areas of a data-center during an incident response. Maintaining overall chain of custody and proper documentation of the examination process is essential part of this phase.

5.6.3 I-DEEP Stage-III: Digital Intelligent Forensics Framework (DIF^2) Implementation Phase

By focusing on predictive modelling and anomaly detection, artificial intelligence algorithms can assist in identifying and flagging potentially significant deviations from normal patterns and behaviours, aiding in the identification and extraction of relevant digital evidence in computer forensics investigations.

Stage-III: Digital Intelligent Forensics Framework (DIF^2) Implementation Phase

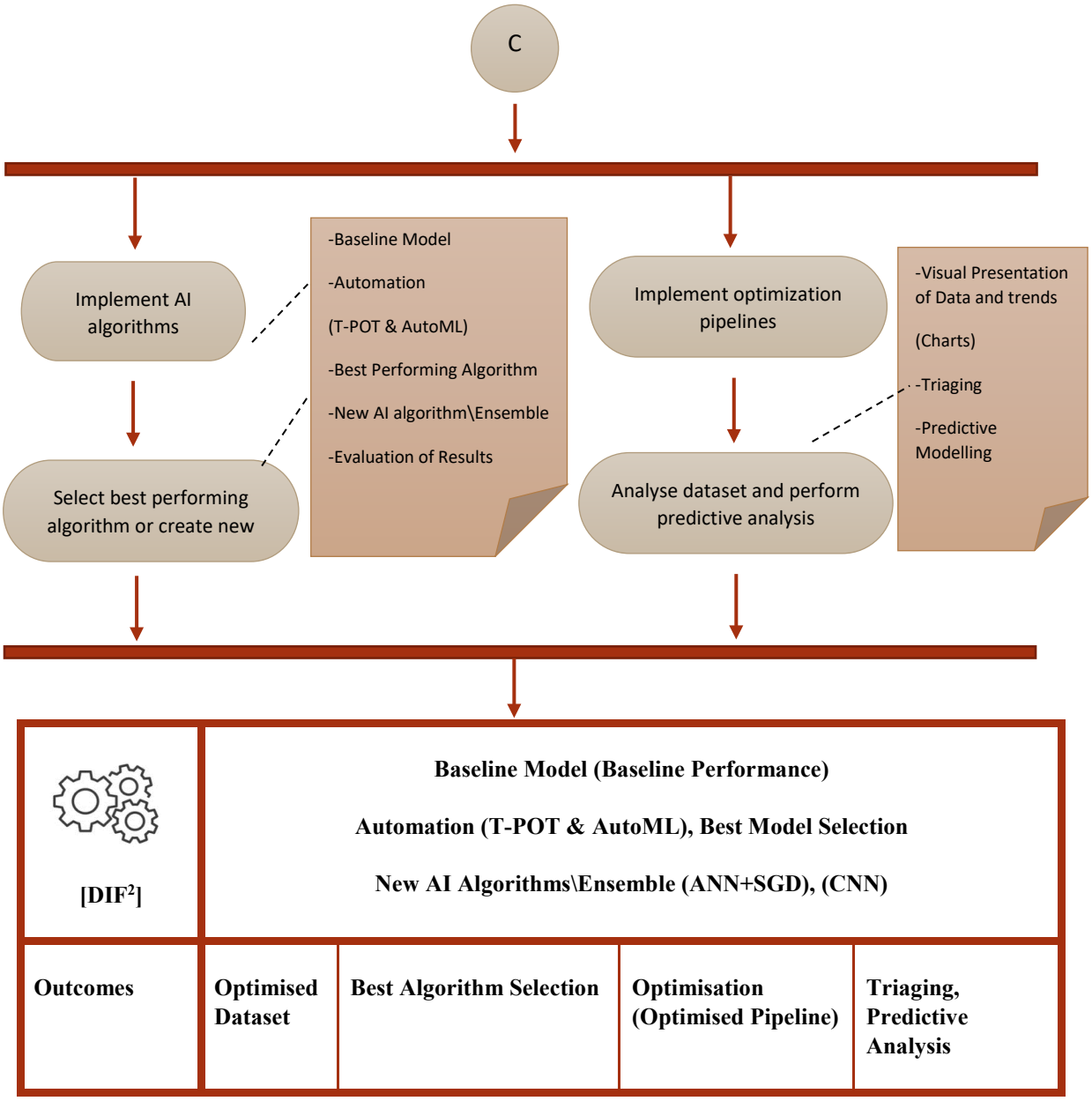


Figure 5.7: Digital Intelligent Forensics Phase (I-DEEP Stage-III)

In the detection of anomalies, a wide range of artificial intelligence approaches can be employed. Knowledge-based systems can be developed to capture the expertise of legal professionals and detect unusual behaviour that may indicate legal violations. Neural networks have the capability to model the behaviour of users, distinguishing between appropriate and inappropriate actions, as well as identifying anomalous usage patterns for individual users. Data mining and Machine Learning (ML) techniques can be utilized to identify anomalies and uncover patterns of behaviour (Johnson et al., 2022). Figure-5.7 illustrates the Stage-III of I-DEEP model.

The incorporation of artificial intelligence (AI) into computer forensics presents significant opportunities to enhance investigative procedures, alleviate resource constraints, and elevate the outcomes of forensic examinations. Through the utilization of AI-powered systems and case-based reasoning, investigators can amplify their capabilities, leading to more streamlined and proficient results in the discovery of crucial evidence. The Digital Intelligent Forensic Framework (DIF^2) is a proposed AI framework that integrates features such as Prediction Modelling using AI algorithms, optimal selection of AI algorithms, pipeline optimization, and predictive analysis into the Stage-III of this model. In forthcoming chapters (Chapter-7 and 8). Figure 7.1 illustrates the ontology for the Digital Intelligent Forensic Framework, referred to as DIF^2 in extensive details and explains its implementation in subsequent discussions in upcoming chapters-7 and 8. This model is expected to evolve as additional elements are introduced to enhance its functionality and implementation following an iterative approach.

In an overview of the AI techniques implemented in the intelligent framework ($DIF^{2)}$, an incremental approach has been used. In first stage, it consists of creating a 'Baseline Model' of AI algorithm to perform data analysis and provide a baseline result of 'accuracy' for predictive analysis. In next stage, data validation and manual selection of best performing algorithm is implemented using a test harness. This constitutes experiment-1. In Experiment-2, automation is implemented and results are compared with the stage-1 and stage-2 experiments. The automation is achieved by implementing 'TPOT and AutoML' that allows for automatic setting of hyperparameters, automatically creating experiment pipelines, auto-selection of best performing algorithms to perform predictive analysis. This stage also demonstrates the potential of using AI automation technology to minimise user intervention. Third stage of experimental setup (experiment-3) involves creating customised algorithms using Artificial Neural Network (ANN), and ensemble method (ANN+SGD) to further improve the predictive

modelling process. Fourth stage (experiment-4) is adding functionality for image predictive analysis using a customised algorithm based on Convolutional Neural Network (CNN) to implement 'image predictive analysis' capability to the framework and also in the prototype functionality. Some additional experiments (experiment-5, 6, and 7) were introduced later to compare the performance of customised algorithm models on varying size datasets (50, 100, 300) and evaluating the results using different hyperparameter optimisation configurations in chapter-9.

5.7 Research Experiment

A Research experiment has been designed to evaluate the performance of the AI algorithm models developed and implemented under DIF^2 framework. This involves parameters like – 'size' of dataset, 'cybercrime type', 'triage score' and 'predictive accuracy' of the model. Integration of AI algorithms and testing their efficacy in predictive analysis (accuracy of prediction) in both classification problems as well as regression problems is the objective of the experimental setup. While classification problems predict the occurrence of an instance of data in a specific class categorisation, the regression analysis provides a trend that any data instance follows and is predicted to follow as an outcome.

5.7.1 Baseline Model: Experiment-1 Design

The stage-1 of experimental setup involved creating a baseline model to train-test cybercrime dataset. In initial experiments (experiments-1 to 4), datasets of 50 vs 100 records were tested. In a later iteration of experiments (5, 6, and 7), datasets with 50, 100 and 300 records were used to test the AI algorithm performance and compare the results in both 'classification' and 'regression' problems. Figure-5.8 illustrates this stage of experimental setup.

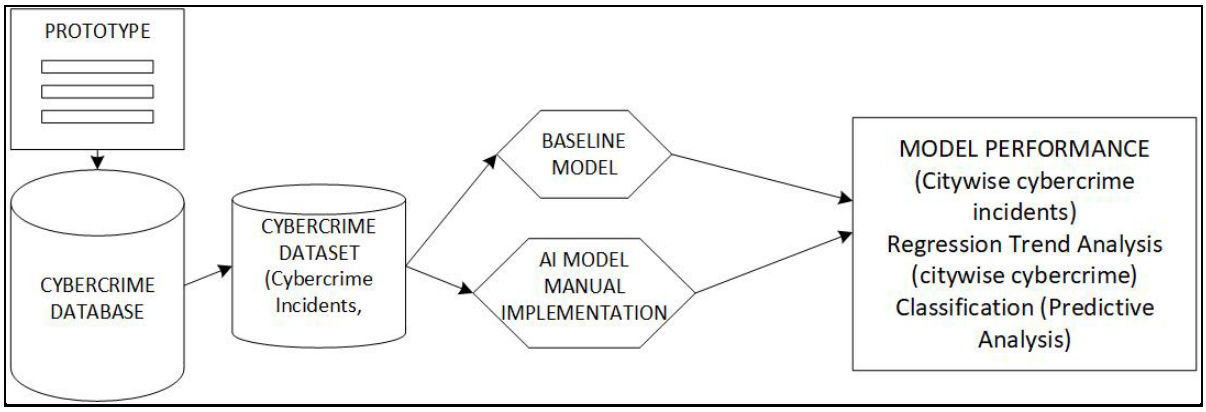


Figure 5.8: Experiment-1 Design: Stage-1 (Baseline Model); Stage-2 (Spot-check Model)

In this experiment (stage-1 of experiment-1), apart from analysing dataset parameters like mean, standard deviation, 25%, 50%, 75% distribution of dataset, it also evaluated other statistical characteristics. We also created a 'baseline model' where we tested our dataset and obtain a baseline score. A Zero Rule Algorithm was developed for this stage of experiment, which has been used for predictive analysis for regression and classification problems. Algorithm code and pseudocode has been detailed in section 7.6. This experiment was designed to compare two datasets (50 vs 100). New experiments were designed later, with three dataset sizes (50 vs 100 vs 300 records) as varying parameters and their performance metrics were compared and results evaluated in chapter-9.

Next step in this experiment-1, stage-2 involved 'manual testing' of some popular AI algorithms in order to evaluate their performance and choose best performing algorithm as shown in figure-5.8. A 'Test harness' algorithm was created for AI algorithm selection and a new algorithm is developed to implement 'Spot Check' method to test the performance of algorithms using 'stratified 10-fold cross validation method'. The performance output was compared and output data was presented in a tabular format as well as 'visual representations' were produced (via box and whisker charts, scatter diagrams). New algorithms developed in this stage and the experimental process is detailed under sections 7.7 and 7.8 in chapter-7.

5.7.2 AutoML Model: Experiment-2 Design

Experiment-2 involved adding more technological complexity into the framework and demonstrated implementation of 'Automation' and 'Auto-optimisation' using AutoML techniques. This stage involved 'implementation and evaluation' of Automation Techniques (TPOT + AutoML) to perform auto 'hyperparameter optimisation' and predictive analysis of dataset. This stage (experiment-2) demonstrated the benefits of automation like auto selection of best performing AI algorithms, 'hyperparameter' optimisation, and optimisation of pipelines with minimum human intervention.

The dataset is tested in this stage and performance of predictive modelling is evaluated. Figure-5.9 illustrates this experiment design. The experiment uses parameters 'cybercrime type, city-wise number of incidents' for predictive analysis.

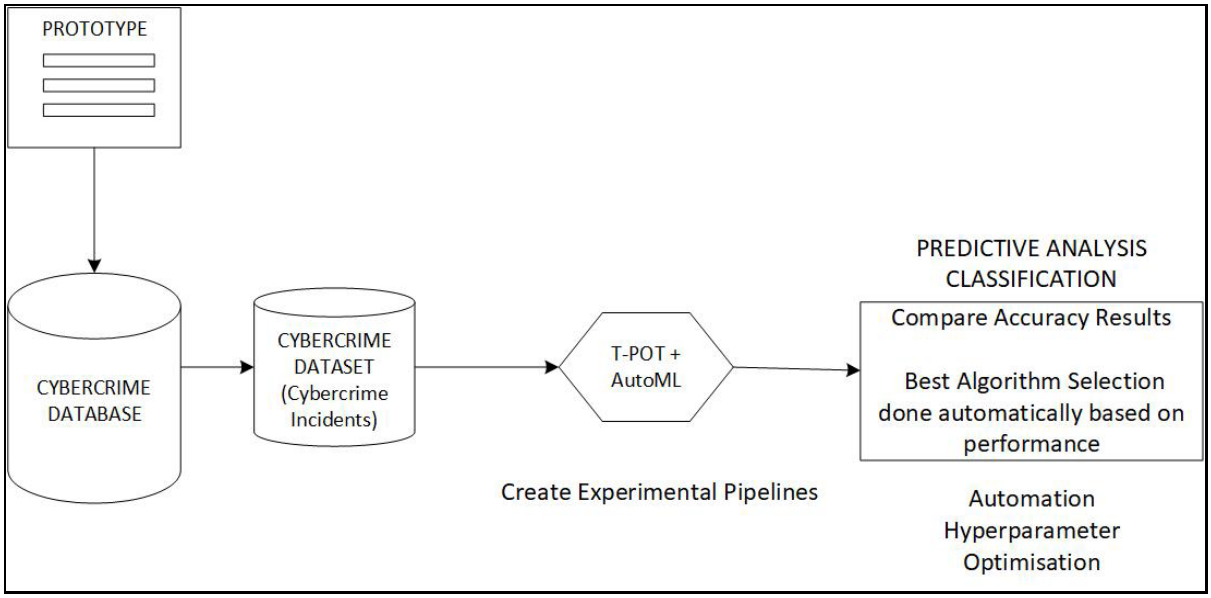


Figure 5.9: Experiment-2 Design: Automation using TPOT and AutoML

5.7.3 ANN Model: Experiment-3 Design

Experiment-3 stage involved development of a customised algorithm from scratch to perform more specific analysis using the 'predictive modelling' in order to 'predict cybercrime incident' in 'classification' problems. This customised algorithm was created by implementing the 'triaging' formula into the algorithm and using Artificial Neural Networks (ANN) and Stochastic Gradient Descent (SGD) ensemble method to achieve better performance in predicting cybercrime incidents. Figure-5.10 represents the experiment-3 design.

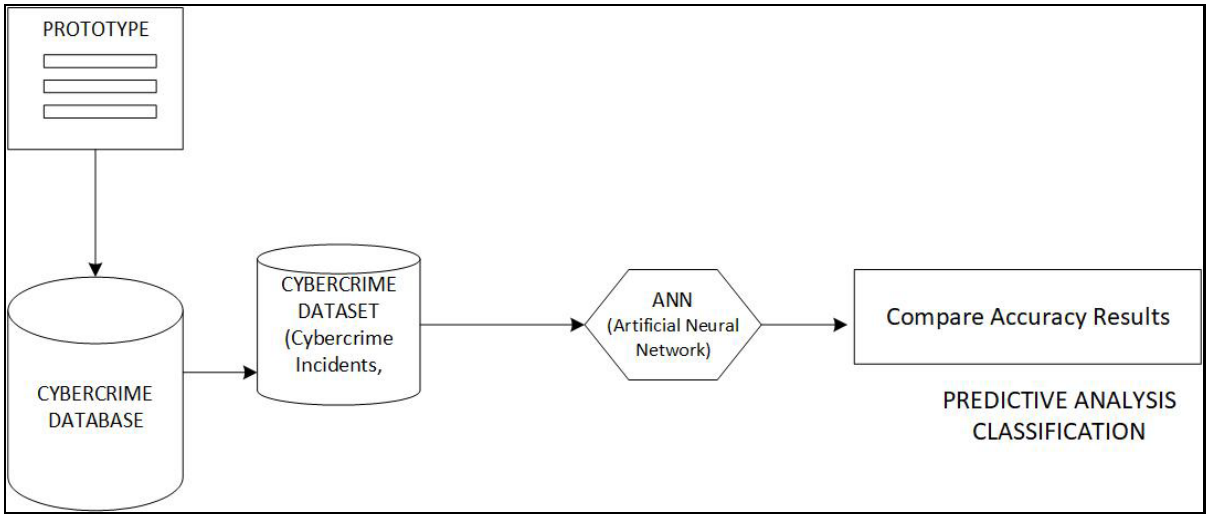


Figure 5.10: Experiment-3 Design: Custom 'ANN and SGD' Algorithm Modelling

The Parameters used for this experimental stage (Experiment-3) were dataset size of varying records. First datasets with 50 and 100 records were compared in experiment-3. Later new experiments (experiment-5, 6, and 7) were conducted with more variation on dataset sizes (50 vs 200 vs 300), severity-score for triaging were calculated and used in predicting the 'severity' status of cybercrime landscape in three different South African cities (Durban, Capetown, Johannesburg). This analysis was done to achieve classification of cities into 'critical' and 'non-critical' class and eventually training the model to achieve predictive analysis using the model to discover future trends in cybercrime incidents and plan for future incidents using 'triaging' and 'prediction'. The performance evaluation and results were discussed in sections 9.3.3 and 9.4.3 in chapter-9

5.7.4 CNN Model: Experiment-4 Design

In order to perform image data analysis, a customised AI model of Convolutional Neural Networks (CNN) was implemented and results evaluated. The choice of this algorithm is based on studies done by other researchers and found it highly effective as demonstrated by Brownlee (2021) in his experiments.

Experiment-4 was specifically designed to evaluate the performance of image analysis (identification of contraband images) from collected evidence using 'Enhanced CNN Algorithm' model. This experiment did not compare with other experiments as the algorithm developed and tested in this stage was specific for image analysis, while others were not designed to perform image analysis. Therefore, this experiment was explicit to demonstrate the 'practical capability' of the prototype to perform image analysis functionality. The results were evaluated based on parameters of 'time' taken for comparing and 'accuracy' of the results. Since it is not possible to test the model on real 'child pornography' contraband images, therefore sample dataset of cats (representing normal images) and dogs (representing contraband images was demonstrated as part of this stage of experiment. This experiment used a dataset of 25000 images consisting of cats and dogs, where cats represented 'normal' images, while dogs represented 'contraband' images. In order to train the dataset, two separate sub-directories were created, one with cats' images and another with dog images. Figure-5.11 illustrates experiemnt-4 design.

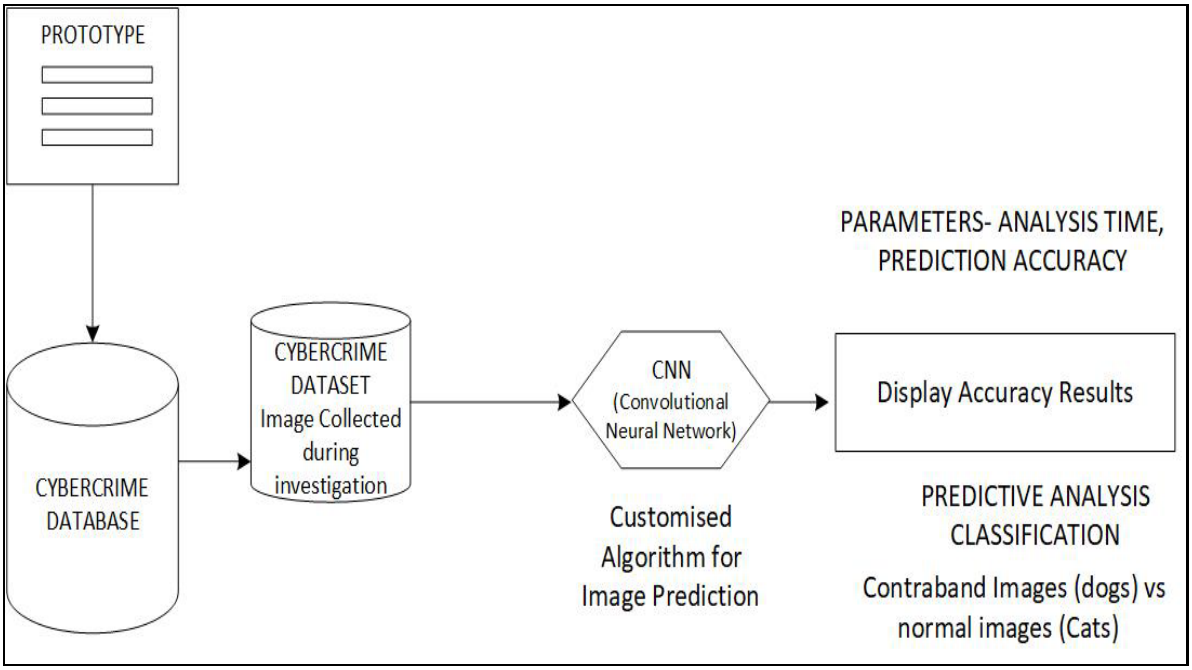


Figure 5.11: Experiment-4 Design: Image Analysis using customised 'Enhanced CNN Model'

5.7.5 Parameters Used for Experiment

The parameter used in this experiment were – 'Dataset size', 'Cybercrime type', 'Triage_score' and 'Predictive accuracy' of the AI models.

While introducing the functionality of image analysis (normal images vs contraband images) a new customised and 'Enhanced CNN Algorithm' was developed. While demonstrating 'image analysis' capability in 'Experiment-4' parameters used were 'time consumed' for search and analysis and the 'accuracy' of results.

Results were presented along with output of models and also discussion on improving the performance of model using 'Three Block VGG' optimisation under section 8.8.5 in chapter-8, and results discussed under section 9.3.4 in chapter-9. A complete diagrammatic illustration of the experimental setup consisting of experiments 1 to 4, is shown in figure-5.12.

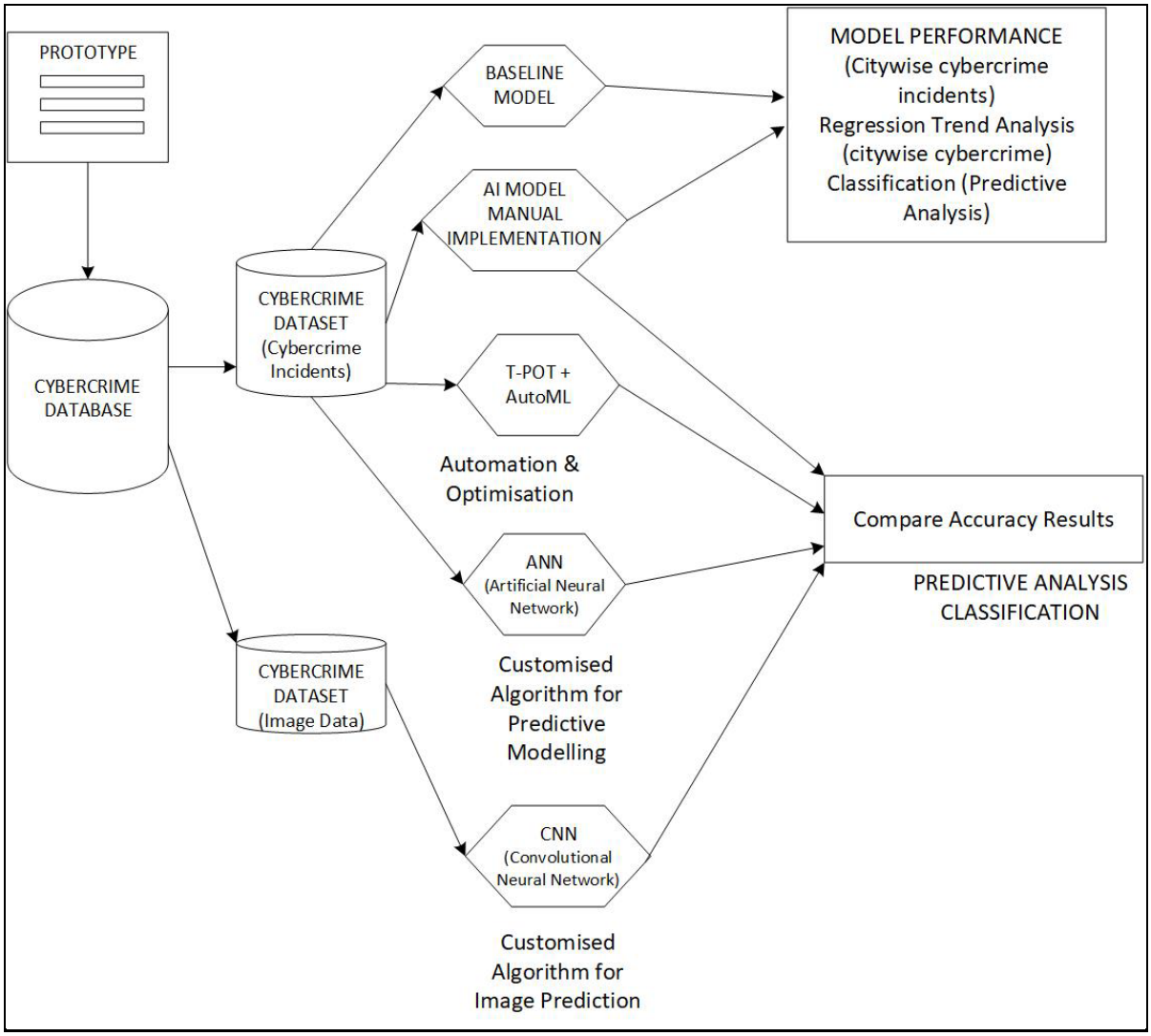


Figure 5.12: Complete experimental ontology of performance evaluation of AI models implemented in DIF^2 framework.

5.8 Experimental Results

The results of experiments have been discussed and interpretations done in chapter-9 in detail. A brief mention of results for various experiments is provided in this section with reference to the experiments conducted.

Experiment-1: On analysis of the results obtained on different split percentages, it is observed that 50% split produced best prediction accuracy scores. This explains the nature of the model (baseline model) that it is simply making predictions randomly and needs further training or feature enhancements. Experiment-1 helped to establish a baseline score for other models and their evaluations in further experiments and also helped to analyse model behaviour. The

advantage of creating the baseline model is that, it presents a formal structure to create test harness that can be further utilised to create 'train and test' data splits and also provides a template to add more complexity to the development of forthcoming models. Another significant observation was that the dataset size also played a critical role in the performance of the model as we compared the results with initially two dataset sizes (50 vs 100) in experiments-1 to 4 and then with three different sizes of datasets (50 vs 100 vs 300) in second stage of experiments (experiments-5, 6, 7). An overall performance improvement was observed when using a dataset of 300 records in second stage of experiments.

We clearly observed that dataset size impacts the performance accuracy scores as in case of 60% Train-Test split, dataset with 50 records (dataset-50) produced score of 42. While it produced 45.3 and 49.1 scores with datasets of 100 and 300 records respectively. A split % of train and test dataset that is too narrow (50-50) or a split too wide (90-10) often produced either random predictions as in the case of 50-50 split or poor accuracy, if the dataset is small (50 or 100 records). Table 9.1 shows complete set of output and observations with varying dataset record sizes (50, 100, 300) and different Train-Test dataset split percentages (50, 60, 70, 90).

Experiment-2: A fairly good efficacy score of around **64.66%** was achieved using the best-selected model, KNeighbors Classifier (KNN). In this experiment that used TPOT and AutoML piplines for model training. The actual output of various pipelines is presented in figure 8.9, showing a creditable performance. However, there is only a marginal improvement in the output score from generation-1 to generation-5.

This modest improvement observed in experimental implementation is attributed to the relatively small dataset, emphasizing the need for a more substantial training dataset to realize the full potential of the optimization process. Table-9.4 shows the output CV (Cross Validation) results of the five generation pipelines using KNN model as best pipeline. The datset size was kept fixed in this experiment (100 records) to observe the output score across generations.

Experiment-3: This experiment involved demonstration and analysis of a newly developed and customised AI algorithm based on Artificial Neural Network (ANN) using Stochastic Gradient Descent (SGD). ANNs demonstrate the capacity to autonomously extract pertinent features from raw data, alleviating the need for laborious manual feature engineering processes. ANNs exhibit effectiveness in both linear and non-linear predictive modelling. ANNs have also shown good performance with smaller datasets as ones used in the experiments in this

research. The model produced final performance accuracy of **97%** as demonstrated in the results discussed in section 9.3.3. The actual algorithm and program implementation is shown in figure-B.12 (APPENDIX-B), which was done using python code. This experiment involved hyperparameters setting – number of folds (n_folds) = 5; learning rate = 30%; number of epoch (n_epoch) = 500 and number of hidden layers (n_hidden) = 5. A higher percentage of learning rate was avoided as high learning rate might cause the optimization algorithm to skip over the minimum of the cost function. Usually high learning rates can cause optimization algorithm to converge to suboptimal or poor local minima instead of the global minimum. In order to demonstrate the functioning of the algorithm and to show how algorithm performs where 'error rate' drops gradually with 'each epoch' a test setting of parameters of number of folds (n_folds) = 2 and number of epoch (n_epoch) = 10 was also tested and the results are shown in Table-9.7 under section-9.3.3.

Experiment-4: This experiment involved evaluation of 'Customised Image Classification' Algorithm based on 'Enhanced CNN Model'. This algorithm was created and eventually enhanced as described in section-8.8.5 using 'Three Block VGG' optimisation. This enhanced model demonstrated the potential to classify contraband images containing 'child pornography' from normal images. This experiment used a dataset of 25000 images consisting of cats and dogs, where cats represented 'normal' images, while dogs represented 'contraband' images. In order to train the dataset, two separate sub-directories were created, one with cats' images and another with dog images.

When a new unseen image is presented to the 'trained' model, it can predict the image's class by using classifier as '0' which represents 'normal' image or '1' as 'contraband' image. The output of the model is evaluated using two parameters namely "Cross Entropy Loss" and "Classification Accuracy". Executing the algorithm involved initially fitting the model, followed by assessing the model's performance on the hold-out test dataset. It's important to note that the results may differ due to the stochastic nature of the algorithm or evaluation procedure, as well as variations in numerical precision. To obtain a more robust understanding, the program is executed multiple times and the average outcome compared. In this instance, an improvement in performance of approximately 5% was observed.

The baseline model achieves around 80%, while the baseline model with simple data augmentation demonstrated an enhanced performance of about **85%** as shown in figure-9.18. The enhanced model with 'three block VGG' algorithm produced accuracy results of **97%**

which is significantly improved compared to the baseline model with 85%. The output in the form line plots of cross entropy loss and accuracy curve is presented in figure-9.19 for the customised and 'Enhanced CNN Model' using 'Three Block VGG' optimisation.

5.9 Summary

This chapter (Chapter-5) focused on studying various aspects of DF research and after thorough analysis done in terms of challenges found in DF domain, based on available literature finally created a new protocol model called Intelligent-Digital Evidence Extraction Protocol Model (I-DEEP). The newly created protocol model considered various phases of DF investigation and evidence gathering in detail whilst it introduced an agile approach into the IR and evidence extraction process. This novel protocol also tends to simplify the DF investigation process and focusses on a more implementable model. This newly developed (I-DEEP) protocol provides a model framework for DF first responders and DF investigators to easily implement the model, saves time by reducing the complexity and making the model more agile to implement. It also formed a theoretical framework to develop a working prototype that can be used by DF investigators to collect evidence and cybercrime related data during investigation and maintain 'chain of custody' information for future reference. This design process was covered in detail in this chapter (Chapter 5). Next chapter (Chapter-6) further elaborates the process of implementation of 'I-DEEP' protocol as a prototype and provides a platform for integration of AI techniques to make the model 'intelligent' and the resulting intelligent framework is referred to as 'Digital Intelligent Forensic Framework' (DIF^2) described in detail in forthcoming chapters. This chapter also detailed the experimental designs to demonstrate the AI integration into the protocol model and also discussed the evaluation of performance and results of the model tested on different datasets with different parameter configurations discussed in upcoming chapters.

CHAPTER 6: DEVELOPMENT AND DEMONSTRATION OF PROTOTYPE BASED ON I-DEEP PROTOCOL

6.1 Introduction:

As the research follows DSR Approach and more specifically DSRP model by Peffers et al. (2007), the researcher has endeavoured to design a new protocol that is more agile and easily implementable named as 'Intelligent Digital Evidence Extraction Protocol' or I-DEEP. This protocol follows the core methodological process of DF investigation and evidence gathering. It further implemented the conceptual protocol model (I-DEEP) in the form of a working prototype in order to collect evidence and cybercrime data and demonstrate the technical feasibility of protocols practical implementation. This development of working prototype further provided a platform for AI integration and allowed to implement AI framework called 'Digital Intelligent Forensic Framework' (DIF^2) to demonstrate the use of AI capabilities to achieve better decision making and predictive analysis on cybercrime data.

Therefore, implementation of I-DEEP protocol and the newly developed AI based framework which also included use of AI techniques (Machine Learning) in order to obtain valuable results in predictive analysis of cybercrime incidents. This framework has been designed to be incremental using a 'gearbox model' analogy that allows to select and to integrate best performing AI models. It involved selection of best performing AI algorithm to achieve desired outcome in prediction and triaging. It also involved integrating more complex technologies like automation pipelines using TPOT and AutoML and finally creating customised Algorithms (ANN+SGD) and Enhanced CNN Models for 'Predictive Analytics' and 'Image Identification'.

Visual representation of data is also done in order to better explain the data characteristics to understand cybercrime from a specific dataset extracted from the collected data using the prototype application. The prototype described in this chapter consists of 'web-based application' interface (prototype) created in order to demonstrate that digital evidence extraction process whereby, data is captured and stored in a database. The prototype design is based on newly developed the digital forensics protocol model – Intelligent Digital Evidence Extraction Protocol (I-DEEP).

This chapter maps to the 'Demonstration' stage of DSRP model by Peffers et al. (2007) and follows newly developed 'research design conceptual model' for this research as described

under section 3.3 in chapter-3. This chapter also addresses the research objective – (c) that endeavours to design a new prototype model that is used to demonstrate the feasibility of implementation of the novel investigation protocol (I-DEEP) discussed in section 5.5 and an intelligent framework based on AI technologies (DIF^2), that can assist first respondents in cybercrime investigations.

The integration of AI has been demonstrated by first representing the logical process as a generalised algorithm in a step-wise manner. This algorithm is then encoded and represented in detail in chapter-7 and 8, as a neutral representation of pseudocode (labelled as procedure) which is language independent. Finally, the actual implementation of the algorithms and pseudocode is done in Python in the prototype.

6.2 Prototype Design

This section covers the User Interface (UI) design of the prototype designed and developed to implement I-DEEP protocol model. Main objective of this activity is to demonstrate that the new protocol model developed is easily implementable in practical scenario. Although the prototype does not extensively implement all the theoretical aspects of I-DEEP as the primary objective is to provide a 'demonstration' of the technical feasibility of the newly designed protocol model. This further provides a platform which allowed for integration of AI technologies and implementation of 'intelligent framework' into the protocol model using newly designed AI framework (DIF^2) to be integrated to achieve optimisation, automation and perform predictive analysis.

This implementation of prototype maps to the fourth phase of the Peffers et al. (2007) Design Science Research Methodological Model that is 'Demonstration' phase. Figures 6.1 to 6.11 shows UI design of the application prototype that assists DF investigators to collect evidential data and 'chain of custody' information. Following the stages discussed in I-DEEP protocol design in section 5.6 of chapter-5 in detail, this chapter is intended to demonstrate the implementation part of newly designed I-DEEP protocol. This implementation follows an 'iterative and incremental' model, which based on 'agile approach'.

6.2.1 Identification and Evidence Collection Page:

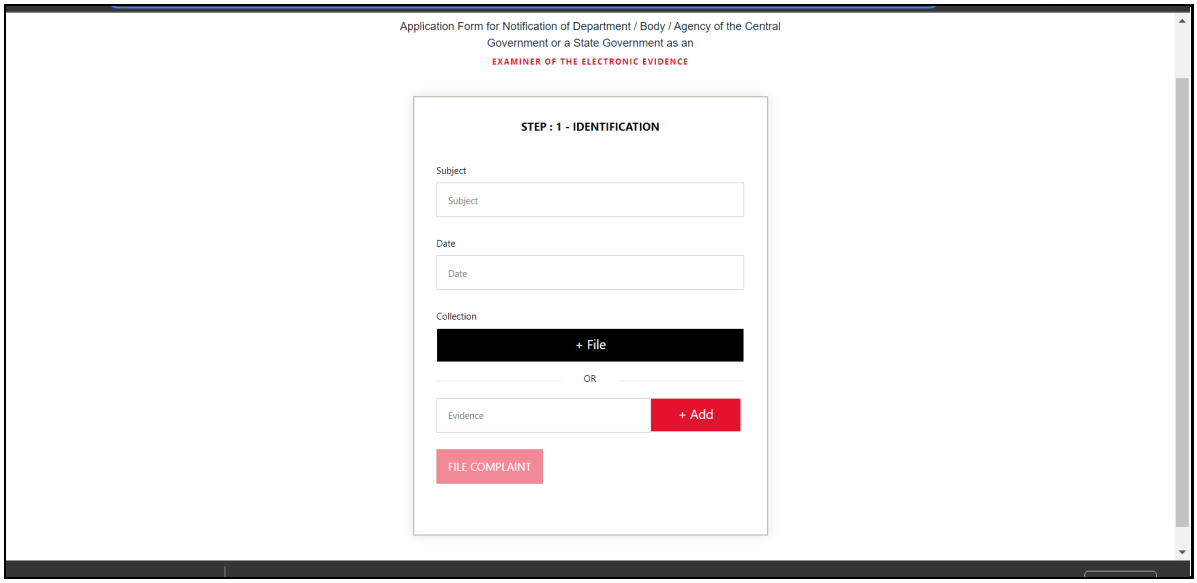


Figure 6.1: Identification and Evidence Collection page (Web UI Design)

The web user interface (UI) shown in figure-6.1 maps to "Identification Phase" of I-DEEP protocol model, which is also common in most DF investigation frameworks or models. This phase identifies the type of cybercrime and the prototype facilitates collection of crucial data regarding investigation process gathered by the investigator as shown in figure 6.1.

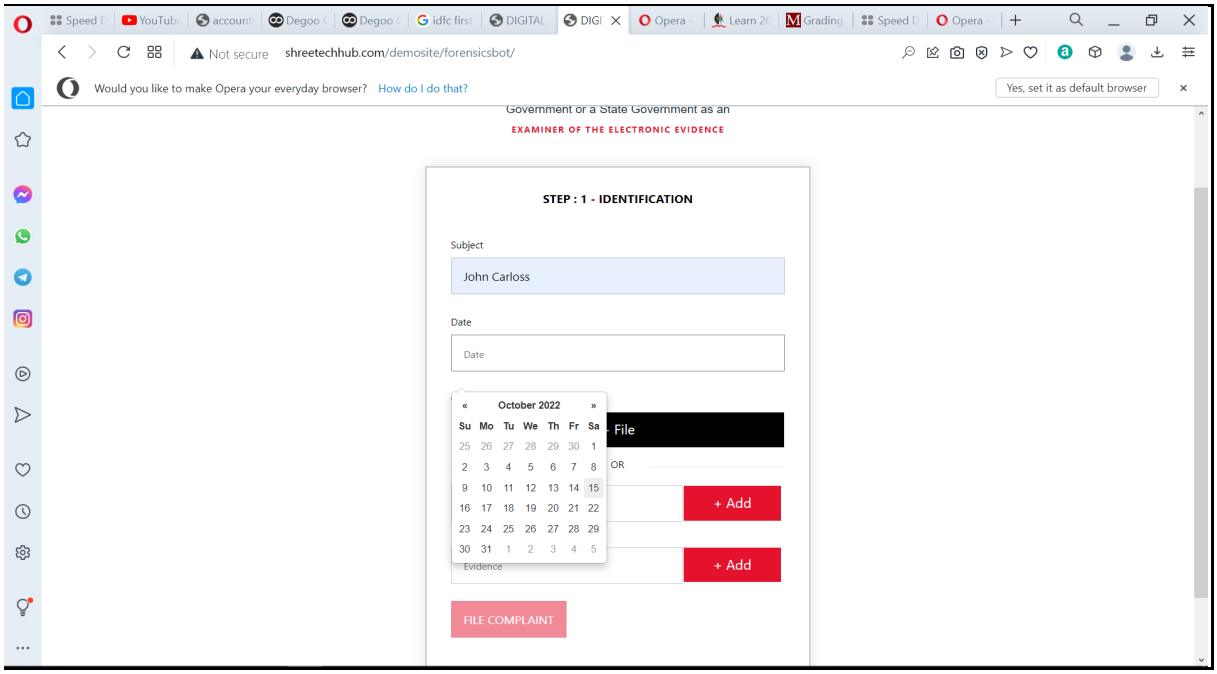


Figure 6.2: Capture Date and Time and Subject of Investigation Page (Web UI Design)

Figure 6.2 shows capturing of important details of the subject and investigation details (date of investigation). It also allows uploading evidence data in various formats (in the form or pictures, videos) and also textual data.

On clicking "+File", picture or video evidence can be added. This can include various formats of images and videos that can represent the investigation site, device or other related information. I can also store data obtained from 'search and seizure' that can hold critical evidential value for investigators.

The UI provides provisions to capture critical information that can be utilised to establish chain of custody, which is an important step in Incident Response process of cybercrime investigation process. Therefore, the prototype demonstrates the implementation of I-DEEP Model in practical terms and also is designed to adhere to 'Agile' principles. This prototype is a new artefact and is crucial in implementation of the theoretical constructs of I-DEEP protocol model and further implementation of Intelligent Framework (DIF^2) as will be described in forthcoming chapters.

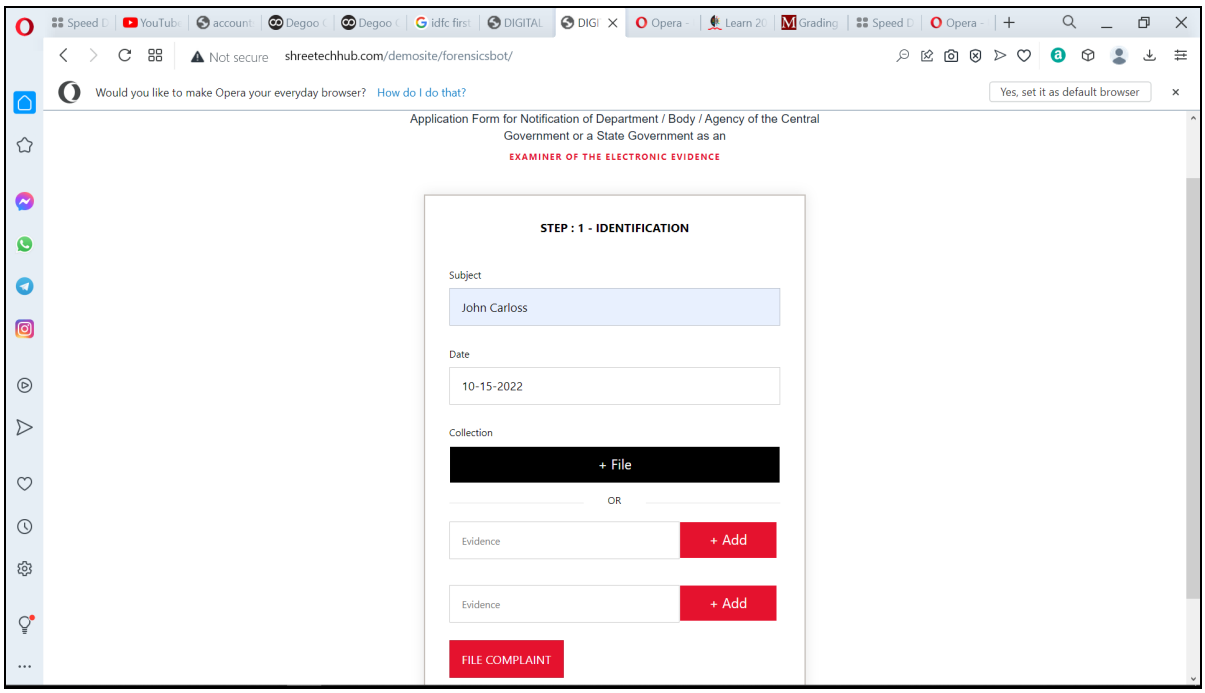


Figure 6.3: Prototype Web UI to capture media information (Media Type, File formats)

The prototype shown in figure-6.3 demonstrates web-based UI for capturing of date and/or adding various media format files for evidence collection.

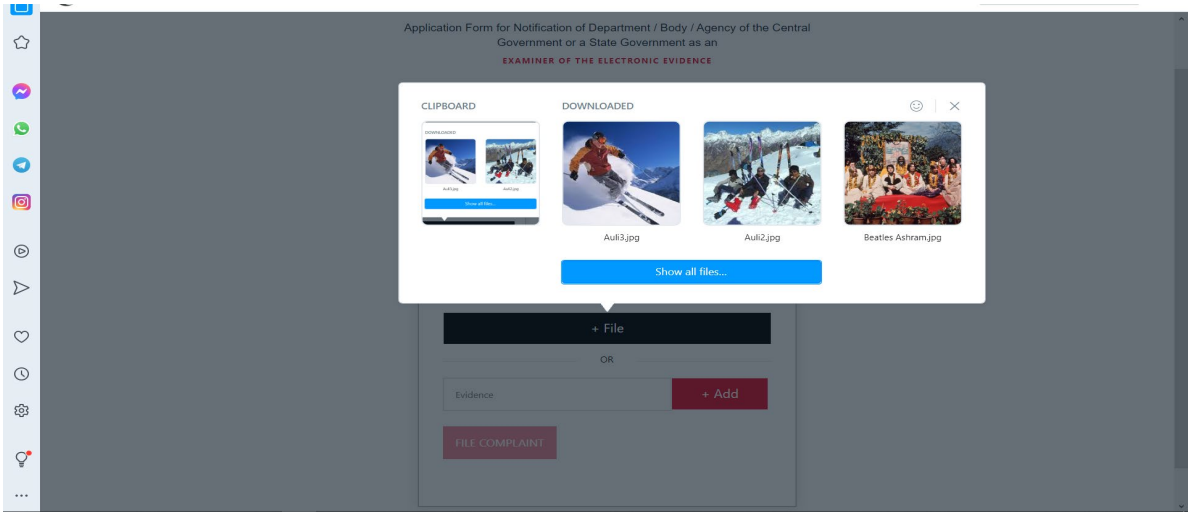


Figure 6.4: Web UI to upload image/video evidence (Pictures, videos of evidence/subject)

Figure 6.4 shows uploading of visual evidence via pictures and video files. More evidence data can be added by clicking "+ add" button using Web based User Interface (UI)

The Application prototype allows to submit evidence for the crimes conducted on a specific day under specific categories of crime-type

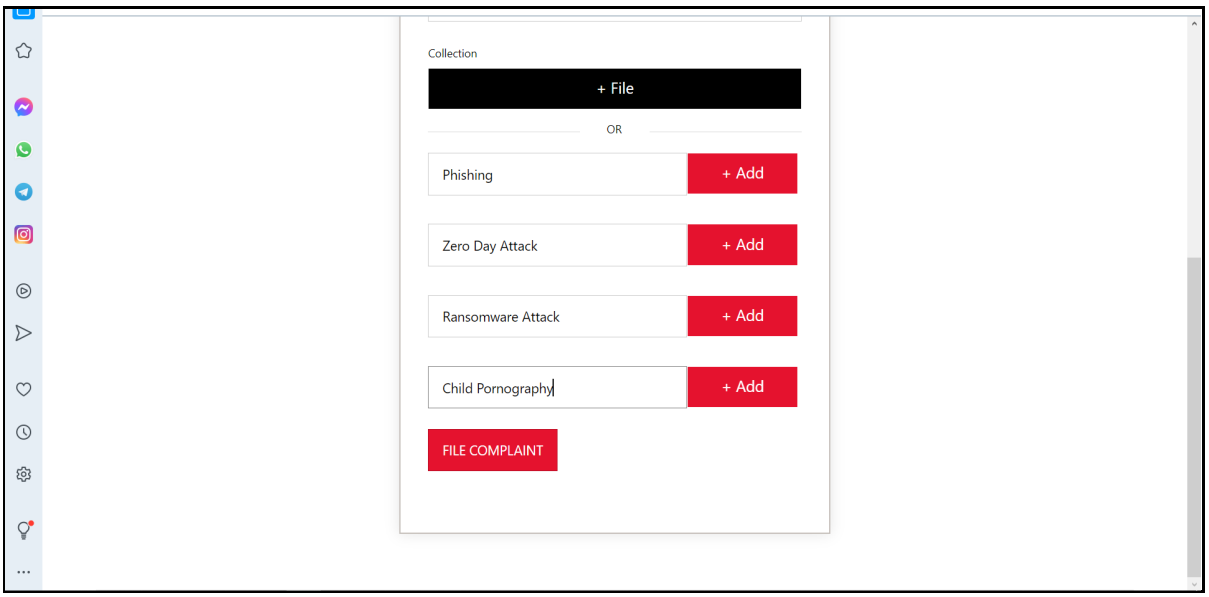


Figure 6.5: Web UI to capture cybercrime information (Cybercrime Type)

6.2.2 Preservation of Digital Evidence:

It is critical to handle and secure evidence properly since errors during evidence acquisition can compromise its integrity and render it unsuitable for forensic purposes. If the incident involves legal proceedings, crucial evidence may be excluded from criminal or civil cases.

To ensure proper evidence handling, there are several key issues that must be addressed or avoided for example tampering of original evidence, documentation of the source of evidence, type of evidence (text files, images, video etc.) and maintaining a proper chain of custody.

Digital forensic examiners must refrain from altering the original evidence. For instance, they should avoid accessing a running system unless it is absolutely necessary. Nevertheless, certain tasks may inevitably modify the evidence, and this can be mitigated by thorough documentation and providing sound reasoning for any changes made.

Tool for Documentation of Evidence

The saying "if you didn't write it down, it didn't happen" commonly used in law enforcement is particularly relevant in digital forensics. It is essential to document every action taken, including detailed notes, photographs or diagrams. In the event that the integrity of the evidence is challenged, this documentation allows examiners to reference the documentation and reconstruct the series of events. The prototype allows for capturing the details related to evidence extraction. This allows for the prototype to serve as a “Documentation Tool”.

Tool for Maintaining ‘Chain of Custody’ of Evidence

The procedure of documenting the movement of evidence from its initial possession to its disposal or return, is referred to as the “chain of custody”. The preservation of the chain of custody is crucial because any omissions in the documentation could result in the exclusion of the evidence from legal proceedings. As a result, it is essential to document the complete life cycle of the collected evidence with accuracy.

There are two key approaches to recording and preserving the chain of custody. The first method is the traditional approach, which involves the use of paper based forms containing the essential information for initiating and maintaining a chain of custody. Although this method requires more vigilance to ensure the form's security and protection against tampering or

destruction, it is a cost-effective option for smaller CERT teams that lack the resources to implement an automated system.

The second method is electronic and involves the use of specialised software to automate the process of maintaining chain of custody of evidence. Use of stickers that are barcoded uniquely are assigned to every piece of evidence, and a scanner can be used to generate an “electronic trail” by scanning these barcodes.

The prototype design follows recommendations of Chain of Custody document by “National Institute of Standards and Technology” (2013) (NIST, accessed on 02-Apr-2023) represented in figure-4.1 and detailed in section 4.3.

The first section of the form provides a comprehensive description of the item, which may seem repetitive, but is critical to digital forensics ‘preservation of evidence’ phase. The detailed record leaves no room for doubt that the evidence collected by investigators is authentic. More details can be added regarding preservation of the evidential data.

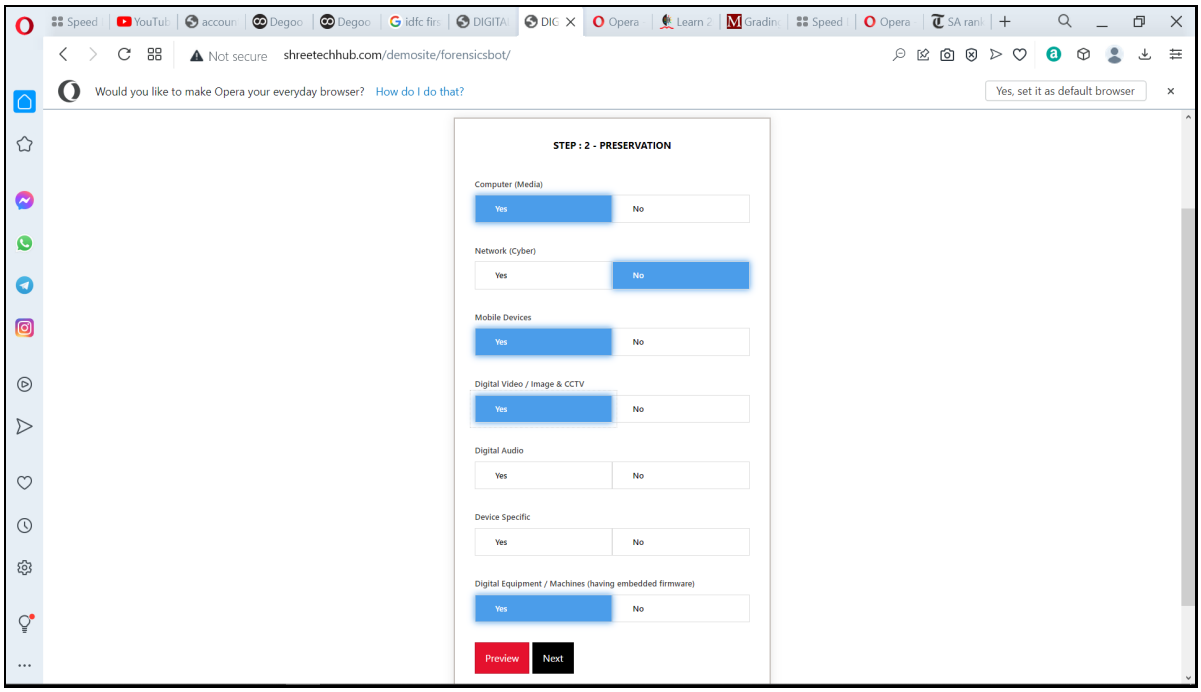


Figure 6.6: Evidence Preservation Information (Equipment type, Device details, Preservation Protocol)

6.2.3 Evidence Extraction and Collection of Evidential Data

A cloud-based hybrid data store is created to hold all types of complex evidential data like data-files, images, videos Etc. This collection can be utilised to implement big data analysis techniques later to derive vital intelligence from the data collected using the prototype application.

This provides an opportunity the demonstrate that cloud-based data is heterogenous in nature and can consist of information in various types of formats like text, images, audio and video files. The prototype is mainly designed and developed to demonstrate the practical feasibility of the conceptual model of I-DEEP protocol and AI integration via Intelligent Framework (DIF^2). These implementations provide a platform for implementation of various algorithms to be integrated into prototype to perform predictive analysis of cybercrime data.

This implementation is further extended to design a series of experiments that demonstrate the AI integration and also evaluate the performances of various AI algorithms. It provides the crucial functionality of 'triaging' and decision making in an 'agile' methodology.

.

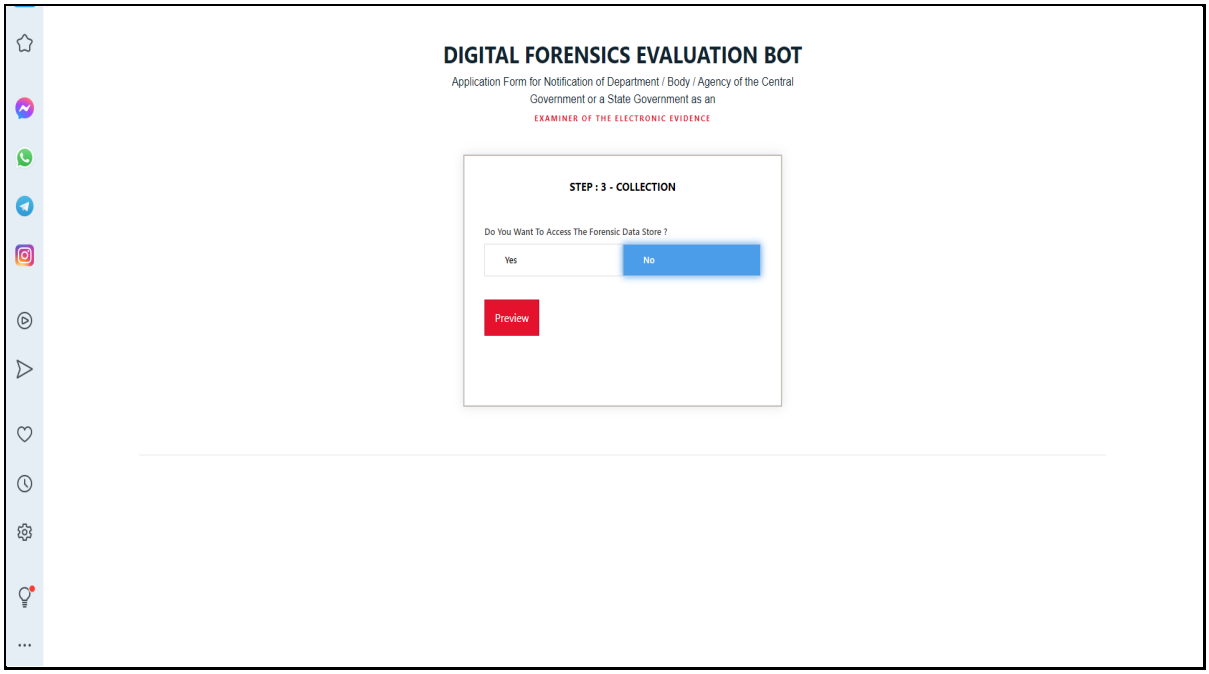


Figure 6.7: Cloud-based Evidential Data Collection and Evaluation (Web UI Design)

Chain of Custody

The procedure of documenting the movement of evidence from its initial possession to its disposal or return is referred to as the “chain of custody”. The preservation of the chain of custody is crucial because any omissions in the documentation could result in the exclusion of the evidence from legal proceedings. As a result, it is essential to document the complete life cycle of the collected evidence with accuracy.

In order to maintain proper chain of custody and evidence extraction and collection details, the details like the department name authorising the investigation and credentials of investigator are maintained. The department the person belongs to, the contact details of investigator etc. are vital details that need to be recorded by the prototype application as shown in figure-6.8 and figure 6.9.

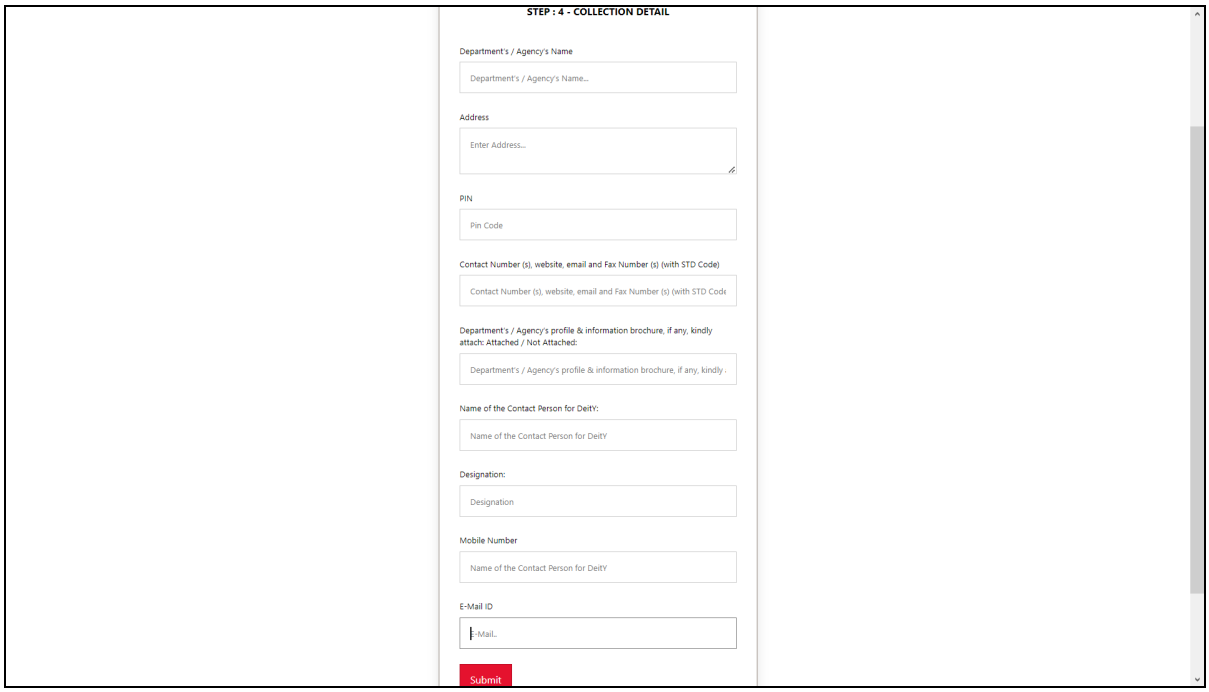


Figure 6.8: Evidence Collection Details (Department/Agency Conducting Investigation) Web UI

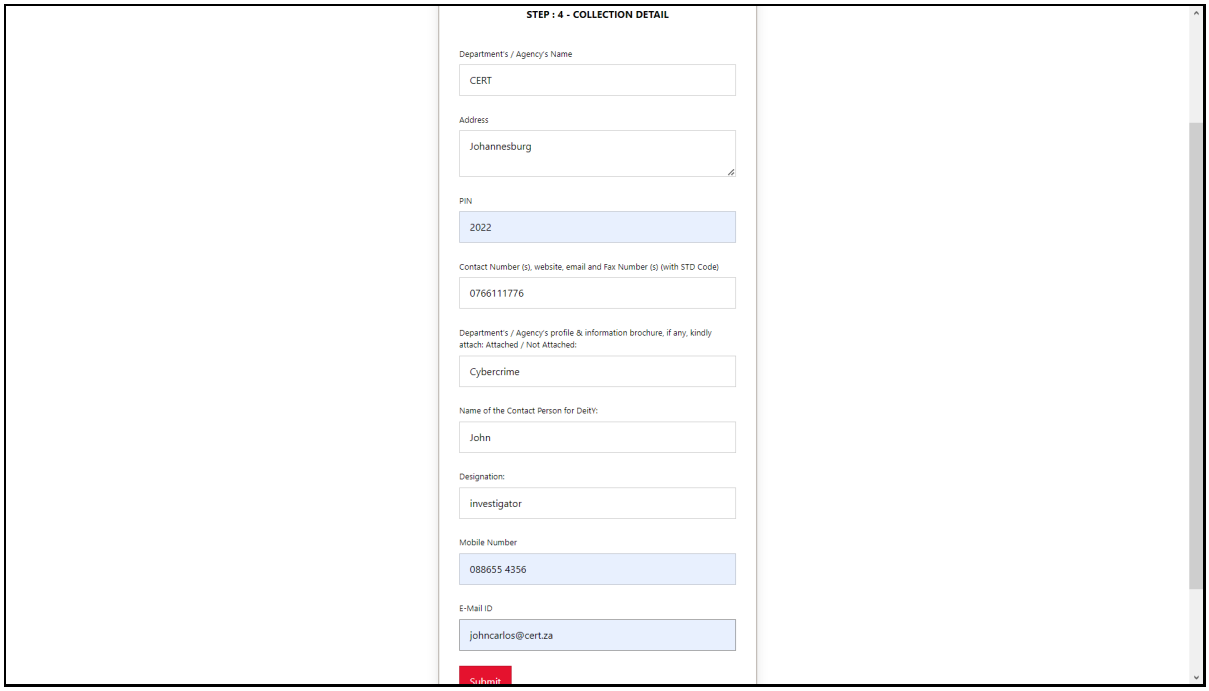


Figure 6.9: Chain of Custody details capture process (Agency\Body\Investigator Details)

Prototype application stores all the evidential data to maintain chain of custody and generates a documentation report for manual filing of evidence as shown in figure-6.10 and figure-6.11.

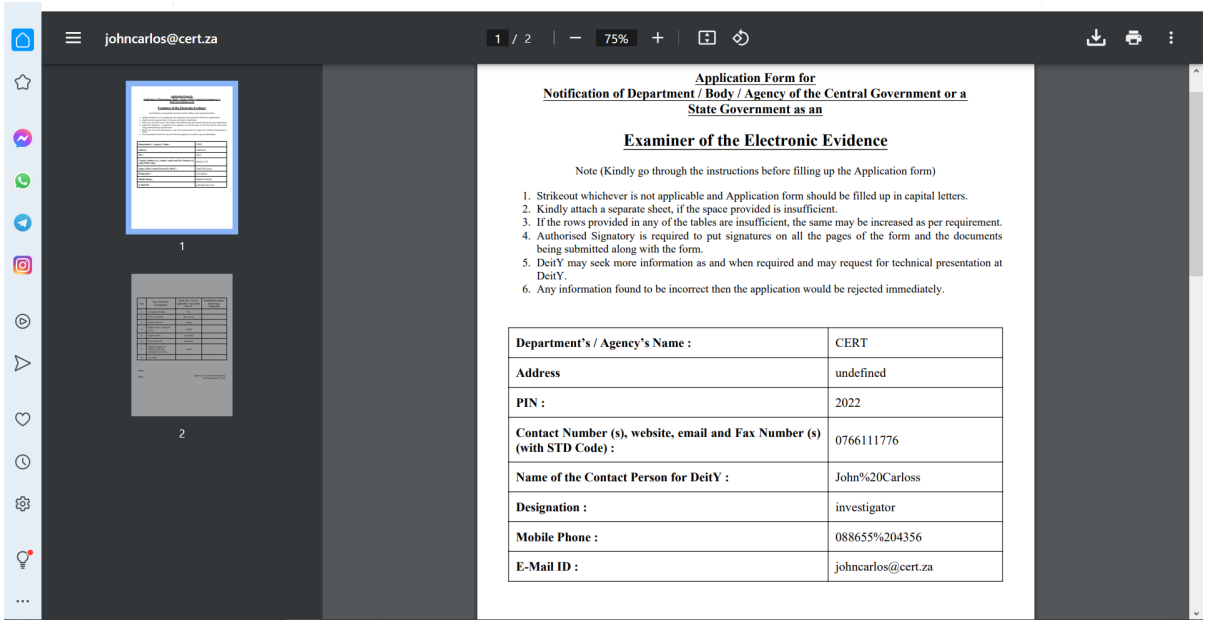


Figure 6.10: Documentation Phase: Auto Generated Report for 'Chain of Custody' Filing Process

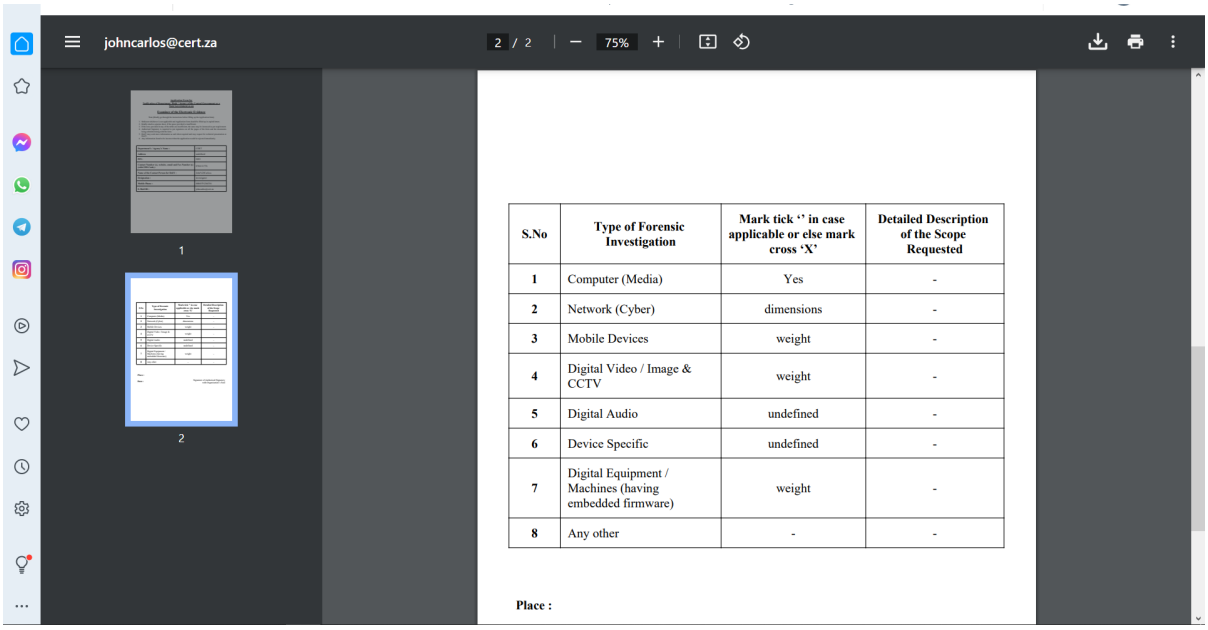


Figure 6.11: Documentation Phase (contd.) Chain of Custody and evidence filing report

6.3 Demonstration of implementation of AI in Digital Intelligent Forensics Framework (DIF^2)

In order to demonstrate the implementation of AI and Digital Intelligent Forensics Framework (DIF^2) in the newly developed prototype model, a specific dataset has been extracted from the data store. The specified dataset is based on defined criteria to investigate and analyse the type of cybercrime committed in a specific location (a city or a province/state). The objective of this exercise is to demonstrate that the newly designed protocol (I-DEEP) is agile and practically implementable in a real-world scenario. Also, this provides a mechanism to demonstrate the implementation of newly created intelligent framework (DIF^2). The methodological process involves extraction of a dataset from the data collected using prototype application in described under section 6.2.

6.3.1 Description of dataset

The dataset consists of cybercrime recorded in three major metro-cities in South Africa – Capetown, Johannesburg and Durban. This dataset is constructed to demonstrate the implementation of AI and intelligent framework (DIF^2) in analysing the data and also further selecting most appropriate (AI) model for further data analysis. This approach follows a

methodological process with the following steps -Define Problem; Prepare Data; Evaluate Algorithms; Improve Results; Present Results.

The dataset consist of test data collected using the protopype designed and described under section 6.2 “Prototype Design”. The Newly extracted cybercrime dataset is hosted on github repository with public access for further analysis using intelligent framework which is designed and implemented as a part of I-DEEP protocol model.

The URL details of the dataset are provided under ‘Data URL’ section

Dataset URL

The parameter ‘url’ provides the location of cybercrime cloud-based dataset repository. This parameter is used to ‘load cybercrime dataset’ for analysis as provided here.

url = "https://raw.githubusercontent.com/deep0025/CyberCrimeData-SA/main/cybercrime-sample-dataset.csv"

6.3.2 Dataset Features Extraction Criteria

Dataset contains cybercrime cases recorded under variables defined as cybercrime type in three major cities in South Africa. Therefore extracted dataset consists of five columns identifying ‘Cybercrime type’ like Phishing, ZeroDay Attack, Ransomware Attack, Child Pornography and the ‘City’ parameters. In order to test the models and their performance, the extracted dataset is taken that consists of ‘count’ of four types of cybercrime incidents namely ‘Phishing, ZeroDay Attack, Ransomware Attack, Child Pornography’, which occurred in three major cities ‘Durban, Johannesburg and Capetown’. The cybercrime incidents extracted represent lowest severity level (Phishing) to highest (Child Porn) and therefore provides a severity spectrum in cybercrime ‘type’ parameters. The dataset description is important to understand the nature of data comprising the dataset, so that a appropriate strategy can be adopted to analyse the data, also the output can be cross validated. This dataset is available in a public repository specified by the URL parameter.

The actual implementation uses Python language to demonstrate the functioning of newly developed algorithms used for AI framework implementation. Algorithms contains summerised logic steps in plain English and their program neutral ‘pseudocode’ labelled as ‘procedures’ are provided to explain the implementation process for AI framework. Actual

implementation of Python code is provided in APPENDIX-B with some instances included in chapters also, whenever it is felt that actual code will assist to demonstrate the functionality in proper context.

Algorithm_Data_Extraction_City (based in 'city' criteria) using parameters (Count, Mean, Standard Deviation of 25%, 50%, 75%, Max_Value)

Step 1: Access URL (location) of the dataset;

Step 2: Extract The names or type of Cybercrime attacks from the dataset based on 'City'as criteria;

Step 3: Read the URL and names of cybercrime incidents;

Step 4: Describe the dataset based on parameters such as 'count', 'mean', 'standard deviation of 25%, 50%, 75% and max values as results;

Algorithm 6.1: Data_Extraction_City based in 'City' criteria

Procedure 6.1: Pseudocode for data extraction based in 'city' criteria and description based on parameters (Count, Mean, Standard Deviation of 25%, 50%, 75%, Max_Value)

Input: Cybercrime incidents dataset with features - ('Phishing', 'ZeroDayAttack', 'Ransomeware', 'ChildPorn'); Class - ('City')

Output: Dataset Description (Count, Mean, Standard Deviation of 25%, 50%, 75%, Max_Value)

```
url = "https://raw.githubusercontent.com/deep0025/CyberCrimeData-
SA/main/cybercrime-sample-dataset.csv"
names = ['Phishing', 'ZeroDayAttack', 'RansomwareAttack',
'ChildPorn', 'City']
dataset = read_csv(url, names=names)
describe dataset
```

Procedure 6.1 is based on Algorithm-6.1 and it represents the neutral and program-independent pseudocode to demonstrate the dataset location and cybercrime data extraction based on 'City' parameter. Python code and output of the procedure is shown in figure-6.12.

```
# descriptions
print(dataset.describe())
```

```
       Phishing  ZeroDayAttack  RansomwareAttack   ChildPorn
count  100.0000      100.00000        100.000000  100.000000
mean     8.3700        5.73000          6.840000    2.100000
std      2.1161        1.73412          2.281768    1.344649
min      4.0000        3.00000          2.000000    0.000000
25%      7.0000        4.00000          5.000000    1.000000
50%      8.0000        6.00000          7.000000    2.000000
75%      9.0000        7.00000          8.000000    3.000000
max     15.0000       11.00000         13.000000    5.000000
```

Figure 6.12: Dataset Description (Python code output of the procedure 6.1)

6.3.3 Statistical Representation of Dataset

Statistical description of Dataset is important to understand how data is shaped also helps to cross validate the results after performing data analysis. In order to describe the dataset statistically, it is further arranged into 'shape' and 'head' and the distribution of cybercrime based on city is represented in Algorithm-6.2 (Algorithm_Statistical_Rep_Dataset), Python code shown in code snippet-6.1 and output of the program is shown in figure-6.14.

Algorithm_Statistical_Rep_Dataset (for Statistical Representation of Dataset)

Step1: Load dataset from location specified in URL;

Step2: Read cybercrime type as cybercrime incident 'names';

Step3: Group the data based on 'city';

Step4: Describe 'shape' of data to display 'city' group-wise cybercrime incidents;

Step5: Arrange the output in rows and columns with 'cybercrime-type' or name as header;

Step6: Show a statistical analysis of dataset as count, mean, standard deviation of minimum, 25%, 50%, 75% and maximum values;

Step7: Display summary of city-wise cybercrime incidents;

Algorithm 6.2: Algorithm_Statistical_Rep_Dataset

The implementation code in Python is shown in figure-6.3 and output is shown in figure 6.14.

```
# summarize the data
from pandas import read_csv
# Load dataset
url = "https://raw.githubusercontent.com/deep0025/CyberCrimeData-SA/main/cybercrime-sample-dataset.csv"
names = ['Phishing', 'ZeroDayAttack', 'RansomwareAttack', 'ChildPorn', 'City']
dataset = read_csv(url, names=names)
# shape
print(dataset.shape)
# head
print(dataset.head(20))
# descriptions
print(dataset.describe())
# city-wise distribution
print(dataset.groupby('City').size())
```

Code Snippet 6.1: Dataset URL and Python Code for Dataset Description

Output of the Python code shown in code snippet 6.1 is showing City-wise distribution of data and statistical representation of dataset, which is shown as output in figure-6.13. It is observed that the dataset contains 34 cybercrime cases in Capetown, 47 cases in Johannesburg and 19 cases in Durban.

Mean score represents that 'Phishing' which has a lowest severity is also highest recorded incident as compared to 'ChildPorn', which is most severe but also least occurring type of cybercrime. The minimum count of Phishing is 4 and maximum is 15. While 'ChildPorn' has a minimum count of '0' and maximum of '5'. Understanding this pattern will help us to cross validate our results and understand if the models are providing correct interpretations.

```
(100, 5)
   Phishing  ZeroDayAttack  RansomwareAttack  ChildPorn       City
0         8              4                 6          1  Capetown
1         8              4                 2          1  Capetown
2         5              3                 3          2  Capetown
3         6              4                 5          2  Capetown
4         8              6                 6          3  Capetown
5        10              7                 6          2  Capetown
6        12              7                 6          1  Capetown
7         8              6                 5          1  Capetown
8         9              9                 8          4  Capetown
9         8              5                 4          2  Capetown
        Phishing  ZeroDayAttack  RansomwareAttack   ChildPorn
count  100.0000      100.00000        100.000000  100.000000
mean     8.3700        5.73000          6.840000    2.100000
std      2.1161        1.73412          2.281768    1.344649
min      4.0000        3.00000          2.000000    0.000000
25%      7.0000        4.00000          5.000000    1.000000
50%      8.0000        6.00000          7.000000    2.000000
75%      9.0000        7.00000          8.000000    3.000000
max     15.0000       11.00000         13.000000    5.000000
City
Capetown        34
Durban          19
johannesburg    47
dtype: int64
```

Figure 6.13: City-wise distribution of data and statistical representation of dataset

The data analysis of the output of dataset is covered under chapter 9: “Evaluation of Results” in detail.

6.3.4 Data Visualisation and Visual Representation of Data

Visual representation of data provides the investigators a more comprehensive understanding of the dataset characteristics and provides insights in to the trends that the data contains. Dataset can be visually represented into box and whisker plots, histograms, scatter plot matrix to better represent the data trends and discover the underlying patterns.

The cybercrime data can be effectively visualized using Python libraries to create box and whisker plots, histograms, and scatter plot matrices. These visual representations offer a comprehensive view of the data, aiding first responders in analyzing cybercrime trends and making informed decisions. The Python code implementation serves as a prototype, demonstrating the capability to visually analyse the collected cybercrime data.

The primary objective of including this analysis functionality here is to showcase the prototype's capability in analysing cybercrime data and aligning with the visual representation stage of the 'extended Linear Model,' as detailed in section 5.5.2. The 'Reporting' feature of the prototype enables the creation of visual representations of cybercrime data.

For the actual Python implementation, refer to Figure B.1 in APPENDIX-B, which includes both the code and the corresponding output. Section 9.2.1 delves into the results and provide a thorough discussion of the findings. The integration of visual representation tools in the prototype enhances the ability to draw key insights from the cybercrime incidents and facilitates a deeper understanding of the underlying patterns in the data.

This prototype is designed to be based on the I-DEEP protocol model, which follows an ‘agile’ methodology and follows an iterative and incremental approach. It systematically integrates an intelligent framework named as Digital Intelligent Forensic Framework (DIF^2). This framework is designed to be incremental again using a ‘Gearbox Model’ analogy, that allows to selectively integrate best performing AI models. This allows for selection of best performing AI algorithms to achieve desired outcome in prediction and triaging. It also supports integrating more complex technologies like automation pipelines using TPOT and AutoML and finally creating customised Algorithms (ANN+SGD) and Enhanced CNN Models for ‘Predictive Analytics’ and ‘Image Identification’ in subsequent iterations.

6.3.5 Algorithm_Visually_Represent_Data

Step 1: Load dataset from location specified in URL;

Step 2: Read cybercrime type as cybercrime incident ‘names’;

Step 3: Group the data based on ‘city’;

Step 4: Plot the data in the form of ‘box and whisker plot’ with a layout of 2 by 2;

Step 5: Show the output;

Step 6: Plot the data as histogram;

Step 7: Show output;

Step 8: Plot the data as ‘scatter plot matrix’;

Step 9: Show the output;

Algorithm 6.3: Algorithm_Visually_Represent_Data

The detailed description of Dataset rescaling, statistical analysis and AI algorithm implementation is explained in upcoming chapters. The main purpose of this representation here is to demonstrate the actual implementation of program in prototype in order to demonstrate the applicability of the conceptual model described in I-DEEP protocol model and establish its implementation potential. This section represents ‘Demonstration’ phase of Design Science Research (DSR) Model used as research methodology in this study.

Actual implementation of the program in ‘Python Code’ is shown in Code Snippet 6.2. This implementation demonstrates the use of an extracted dataset from cybercrime database which is stored in a cloud-based repository and the ‘url’ parameter shows the location/path of the dataset. It also demonstrates the ‘visual representation’ or visualisation of data to understand the patterns using ‘box and whisker plots, histograms and scatter plot matrix shown in figures-6.16, 6.17 and 6.18 respectively.

```python
from pandas import read_csv
from pandas.plotting import scatter_matrix
from matplotlib import pyplot
# Load dataset
url = "https://raw.githubusercontent.com/deep0025/CyberCrimeData-SA/main/cybercrime-sample-dataset.csv"
names = ['Phishing', 'ZeroDayAttack', 'RansomwareAttack', 'ChildPorn', 'City']
dataset = read_csv(url, names=names)
# box and whisker plots
dataset.plot(kind='box', subplots=True, layout=(2,2), sharex=False, sharey=False)
pyplot.show()
# histograms
dataset.hist()
pyplot.show()
# scatter plot matrix
scatter_matrix(dataset)
pyplot.show()
```

Code Snippet 6.2: Python Code for visual representation of dataset in histogram, scatter-matrix, box and whisker plot

Figure-6.14 shows the output of code snippet-6.2, which represents the dataset visually as 'Box and Whisker Plots'.

Output of program showing visual representation of cybercrime dataset

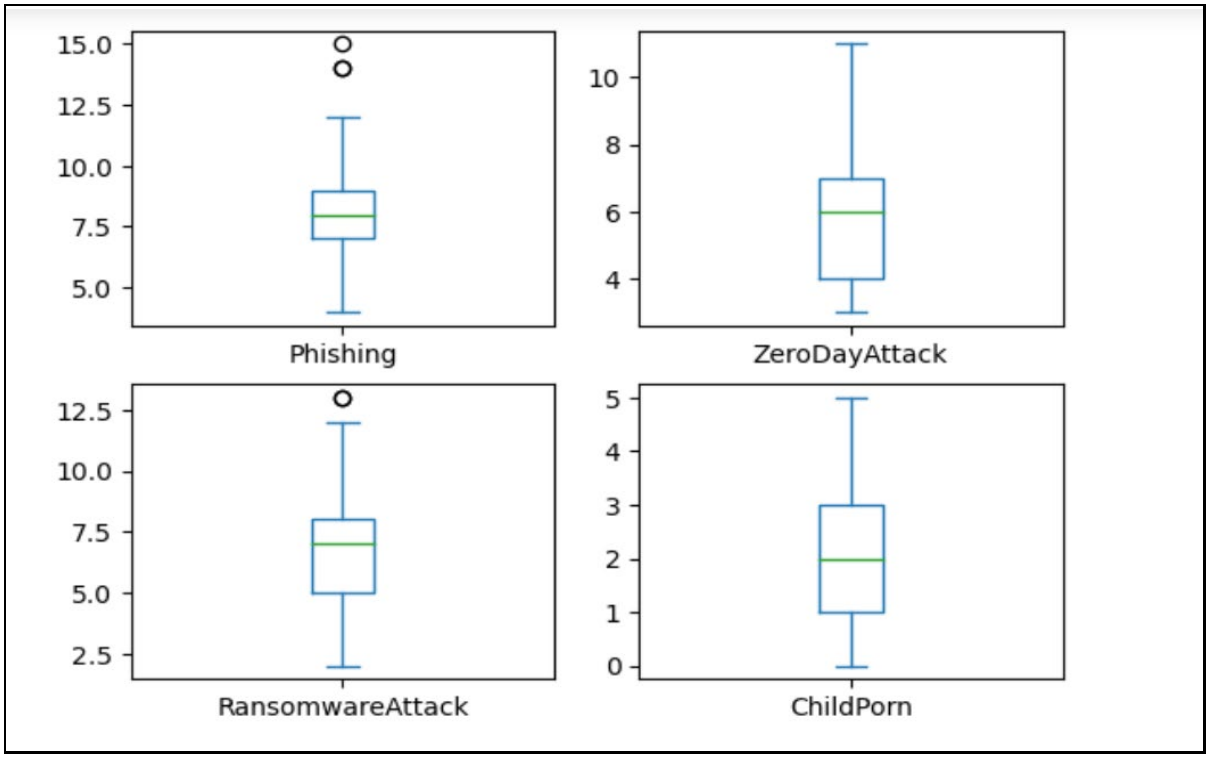


Figure 6.14: Visual Representation of Cybercrime Dataset using Box and Whisker Plots

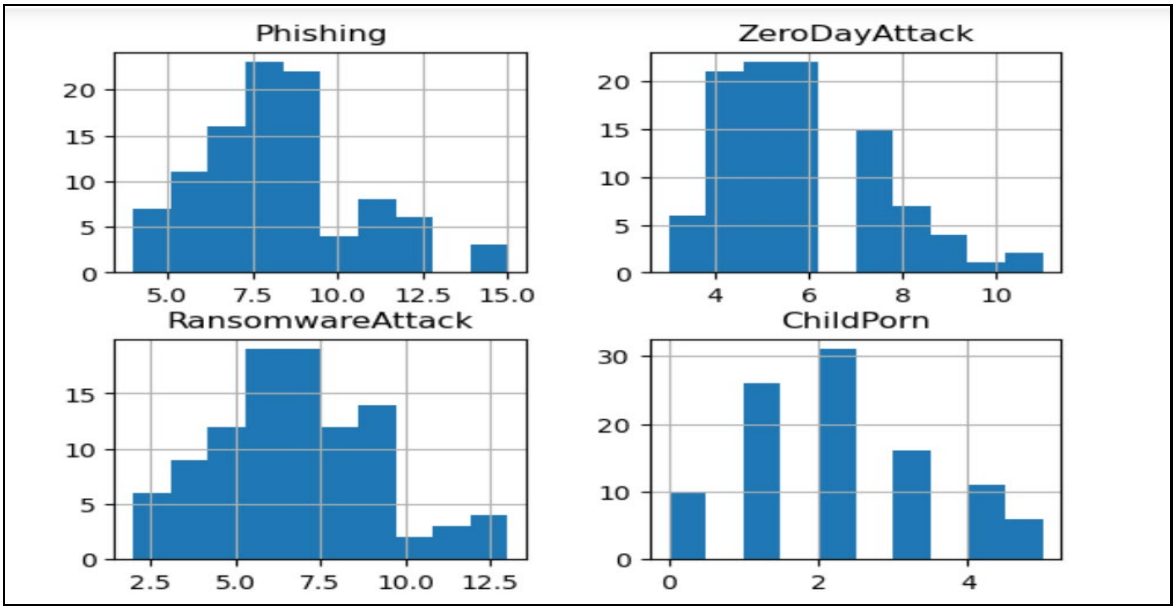


Figure 6.15: Visual Representation of Cybercrime Dataset as histogram charts

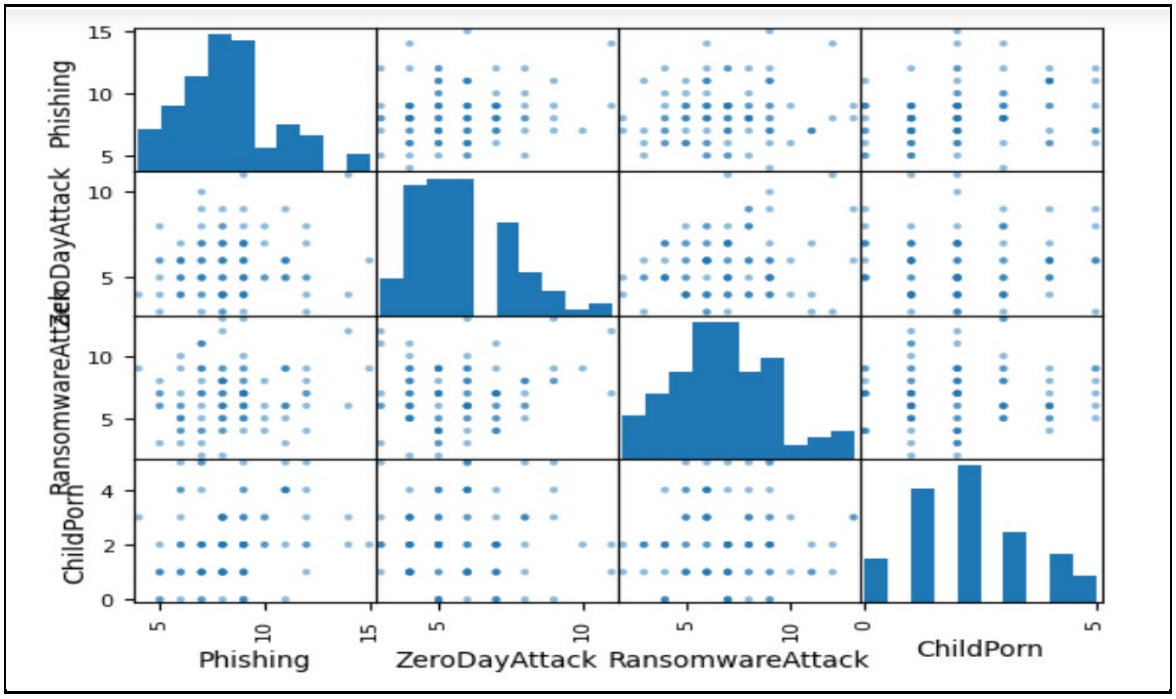


Figure 6.16: Visual Representation of Cybercrime Dataset using Scatter Plot Matrix

Figures-6.14, 6.15 and 6.16 show the program output which helps understand the data characteristics and relationships. Box and whisker plots in figure-6.14 show the data representation in the form of boxes and line 'whiskers'. X-axis show data samples as multiple box plots, while y-axis represent observation values. Boxes span from 25percentile to 75percentile distribution called "interquartile range" or IQR, while the median i.e. 50percentile is shown as a line within the box. Lines extending on outer side of boxes are called 'whiskers'

which represent expected range of sensible values calculated as (1.5 x IQR). The observations outside the whisker range are ‘outliers’ represented with small circles.

The histograms chart shows the nature of data that show “density estimates” of data. Histogram charts are useful in representing the summarised distribution of data samples. Figure-6.15 represent histogram where it is seen that ‘Phishing’ and ‘Ransomware’ cybercrime data show a bell-shaped curve or ‘Gaussian Distribution’, which shows normal distribution of data. This pattern shows a single peak at the centre of distribution where the mean or central point divides the data into two equal halves that are symmetrical. These patterns can help investigators understand a bigger picture of the trends in cybercrime incidents. Figure-6.16 shows ‘visual representation’ of Cybercrime Dataset using ‘Scatter Plot Matrix’ and the chart analysis shows a strong diagonal correlation which is represented in histogram chart. These patterns show density of each variable and its distribution.

Scatter plots are graphical representations that are particularly useful for visualizing the relationship between two continuous variables. Each point on the plot represents a unique combination of values for the two variables. They can also help to understand association or correlation between variables, which can be positive or negative. A scatter plot matrix is a grid of scatter plots that allows for the visual examination of relationships between multiple pairs of variables in a dataset. In a scatter plot matrix, each cell in the grid represents the scatter plot of two variables, and the matrix displays these plots for all possible pairs of variables in the dataset. By examining the scatter plots, patterns such as linear relationships, clusters, or trends can be identified. Correlation between pairs of variables can also be visually assessed.

Regression plot of Severity Scores for Cybercrime Data

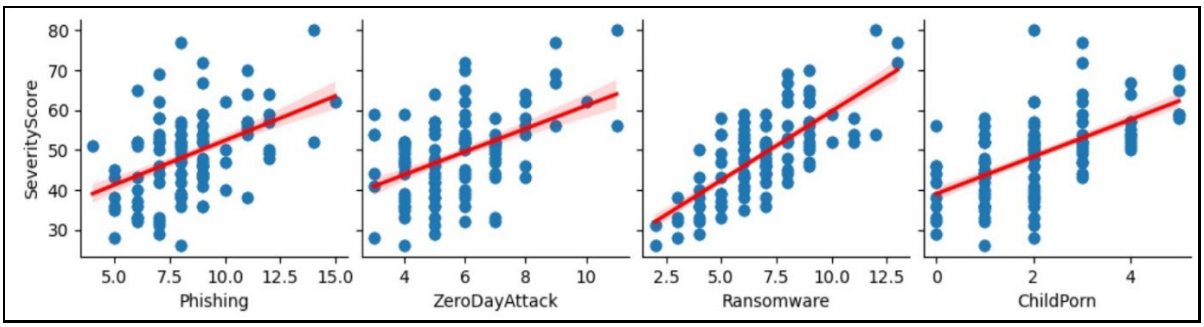


Figure 6.17: Regression plot graph showing cybercrime severity score comparison

Figure 6.17 represents ‘Regression plot’ graph of cybercrime severity score comparison. This visual representation uses ‘seaborn’ and ‘matplotlib’ libraries to plot the regression between multiple variables representing cybercrime data showing cybercrime severity score comparison. Y-axis represents the severity score that is calculated by providing weightage to cybercrimes and multiplying this value with number of incidents. X-axis represents the cybercrime incidents categorised into variables like ‘Phishing’ ‘ZeroDayAttack’, ‘Ransomware’ and ‘ChildPorn’. The plot shows regression patterns and also some correlation patterns between ‘Phishing’ and ‘Ransomeware’ attacks. It also shows least distribution between ‘ChildPorn’ and increased distribution between other three cybercrime types. This explains that the incidents like Phishing’ and ‘ZeroDayAttacks’ are more prevalent although their severity is less as compared to ‘ChildPorn’ incidents.

Pivot-Table Representation of ‘Critical’ Classification of Cybercrime

Python Code Snippet-6.3 displays the ‘Critical’ (represented by ‘1’) and ‘Non-critical’ (represented by ‘0’) instances of classification based on ‘City’. This represents ‘Johannesburg’ as the city that registered most incidents flagged as ‘Critical’, while least critical incidents in ‘Durban’. This analysis can be effective in assisting of efficient ‘traiging’ process and decision making by first responders and cybercrime investigators.

```
import seaborn as sns
import pandas as pd
import matplotlib.pyplot as plt

# Load the data
dataframe = pd.read_csv("CybercrimeDataset_City_SeverityScore_Class.csv")


print(pd.pivot_table(dataframe,index='Critical',columns='City',values='SeverityScore',aggfunc='count'))
print('------------------------------------------------------------')


# Create pairplot for specified columns
#sns.pairplot(pd.pivot_table(dataframe)

# Display the plot
plt.show()
```

```
City      Capetown  Durban  Johannesburg
Critical
0               21      15            14
1               13       4            33
------------------------------------------------------------
```

Code Snippet 6.3: Pivot Table representation of cybercrime data on class ‘Critical’

These data characteristics are discussed in detail in chapter -9 again and further analysis and interpretations are provided to derive conclusions that can assist cybercrime investigators in understanding trends and decision making.

6.4 Summary

This chapter entails the discussion around 'Demonstration stage' of DSR model adopted for this research, which is demonstrated by developing a working prototype, that is based on the design of novel protocol (I-DEEP) described in chapter-5. The research followed the Design Science Research (DSR) Approach, more specifically a research process model based on DSRP model by Peffers et al. (2007) and demonstrated a prototype based on the newly designed investigation protocol called the Intelligent Digital Evidence Extraction Protocol (I-DEEP). This protocol is designed to be agile and easily implementable, following the core methodological process of digital forensic investigation and evidence gathering. This chapter demonstrated implementing the conceptual protocol model into a prototype application to collect evidential and cybercrime data. In order to make the prototype intelligent, and obtain valuable insights into results, improved efficiency and perform predictive analysis, AI techniques were integrated into the protocol subsequently. The prototype demonstrated a "Gearbox Model' approach, which allows for iterative and incremental integration of AI based technology and permits selection of best performing algorithms. This also involved development of an intelligent framework called Digital Intelligent Forensic Framework (DIF^2), which consisted of use of AI (Machine Learning) algorithms, automation, selection of best performing algorithm. Optimisation of experiment pipelines, in order to improve efficacy and achieve better decision making in the investigative process. These implementations are discussed in detail in forthcoming chapters (chapter-7 and 8). Additionally, visual representations of data are generated to enhance the understanding of the characteristics of cybercrime, utilizing specific datasets extracted from the collected data using the prototype application. A web-based application interface was created to showcase the capturing and storage of data in a database. The prototype design is based on the Intelligent Digital Evidence Extraction Protocol (I-DEEP) Model, which serves as the foundation for the implementation of the protocol. This is done to demonstrate that the protocol is 'agile' and can be easily implemented in practical scenarios. It also allows for integration of AI technology into the working prototype to achieve better decision making using visual representation of cybercrime data. This chapter is mapped to the 'Demonstration' stage of DSRP and research design conceptual model developed for this research.

CHAPTER 7: DIGITAL INTELLIGENT FORENSICS FRAMEWORK (DIF^2)

7.1 Introduction

Digital forensic investigations rely on the efficient evaluation of large volumes of data to uncover critical evidence in criminal cases. However, the limitations of time and resources, including computational power and human expertise, hinder the effectiveness of these investigations. To overcome these challenges, there is a need to optimize the utilization of available resources and extend the capabilities of existing forensic tools. In this context, we propose the integration of Artificial Intelligence (AI) into digital forensics through the development of an AI-based framework and the application of case-based reasoning. This chapter details the integration of AI via intelligent framework named as Digital Intelligent Forensic Framework (DIF^2) which has been developed by the researcher as contribution to the study.

By harnessing AI technologies, such as machine learning and pattern recognition, the proposed system aims to enhance the efficiency and accuracy of digital investigations. It can assist in automating certain tasks, reducing the manual effort required, and enabling faster analysis of vast amounts of data. Case-based reasoning allows the system to learn from past cases and apply that knowledge to current investigations, improving decision-making and facilitating the discovery of relevant evidence.

The integration of AI in digital forensics holds great potential to enhance investigative processes, mitigate resource constraints, and improve the outcomes of forensic investigations. By leveraging AI-based systems and case-based reasoning, investigators can augment their capabilities and achieve more efficient and effective results in uncovering crucial evidence. Digital Intelligent Forensic Framework (DIF^2) is a proposed AI framework that implements features like Prediction Modelling using AI algorithms, best AI algorithm selection, pipeline optimisation and predictive analysis.

7.2 Transition from Evidence Extraction to Analysis Phase using AI Techniques

The forensic analysis of computer systems entails a number of methods to preserve, gather, and analyse evidence contained in digital storage media in order to present and use it as proof of illegal activities utilising such resources. As an illustration, we can use web server intrusions, which may result in the alteration of materials that are accessible to the public or the illegal access to sensitive information, among other

things. Digital evidence plays a critical role in various significant criminal cases, which includes document forgery, tax evasion, child exploitation, and even acts of terrorism, throughout the course of a criminal inquiry.

The digital storage media capacity has been increasing steadily, and their pervasiveness in everyone's everyday lives has led to an increase in both the volume of data that has to be examined and the need for research in this domain. In addition to this issue, the available forensic technologies are not effective to analyse a large number of pieces of evidence collected and then correlate the results timeously. Due to these reasons, the work of digital forensic experts becomes quite time-consuming. Since the majority of digital forensic tools lack distributed processing and AI capabilities, the computational resources necessary to conduct such exams also lacks efficacy (Yuki et al., 2019).

The main research goal in this research entails the creation of an innovative protocol and AI-driven framework, accompanied by the development of an AI-powered tool. This tool is designed to support Digital Forensics (DF) investigators in specialized forensic investigations, with the aim of achieving significantly enhanced outcomes compared to the existing instruments. This endeavour takes into consideration three crucial aspects: Firstly, minimizing the process of repetitive and routine analysis process while simultaneously lowering the volume of evidential material that the expert needs to analyse. Secondly correlating the evidence and thirdly, dispersing the common procedures in doing so and adopting new technologies like AI, it is possible to use computational and human resources more effectively.

7.3 Need for AI Integration

In reality, computer forensic examinations frequently involve professionals who are unable to predict in advance which pieces of data will prove to be most crucial to the case under investigation. Think of a cybercafé as an example, or other similar situations where computer networks are installed and multiple internal network devices as well as workstations use the same IP address via proxy server or sub-netting. If IP tracing is performed, it will often point to the cyber cafe rather than any specific machine. The similar challenge might be encountered while assembling fraud proof in businesses with numerous users and networked workstations.

In such cases, conducting a preliminary analysis of potentially suspicious systems would limit the quantity of machines to be investigated. This reduction in the number of machines would significantly decrease the time required to complete forensic examinations. The challenge lies in the absence of sophisticated tools that could aid forensic professionals during the preliminary analysis stage. This absence results in the unnecessary collection of numerous machines for scrutiny, including many of such machines that may not significantly contribute to the final investigative results. Consequently, this prolongs the overall time needed to conclude the digital investigation process.

The core focus of this research revolved around fulfilling the need for advanced intelligent tools and enhancing the efficient utilisation of computational resources within forensic examinations. Considering the extensive data collected and presented to forensic laboratories for scrutiny, coupled with the constraints of time and available resources, we contend that expecting digital forensic experts to perform precise individual and cross-analysis of machines in limited timeframe is impractical.

In the most straightforward scenario, experts can only inspect the elements of an isolated machine. Nonetheless, given that computers are consistently interconnected within networks, facilitating data exchange, this situation unfortunately isn't the most common. Additional complexities arise from the growing storage capabilities of portable media. Due to resource limitations that hinder a comprehensive cross-analysis, these computers and removable media necessitate separate examinations. Due to the issues raised, many potentially relevant pieces of evidence are lost during forensic investigations. This affects incident response (IR) as well as DF investigations. To overcome some of the challenges it makes a pressing need to incorporate intelligent forensics into the investigation process. Integration of machine learning can improve decision making and bring efficacy into the DF investigation process.

7.4 Integration of Machine Learning Algorithms for Predictive Modelling

The objective of predictive modelling is to construct models that can effectively predict outcomes on unseen or new data. Since access to this new data is absent during the training phase, reliance is placed on statistical techniques to estimate how well a model would perform when applied to new, unseen data. These techniques are

categorized as resampling methods, as they involve the process of resampling from the available training data to simulate the performance of the model on different subsets of data that mimic what might be encountered in the real-world application (Brownlee, 2016).

Validating a dataset, data rescaling, data standardisation, resampling of dataset are some crucial steps within the Machine Learning (ML) process that needs to be achieved before performing predictions on a given dataset. To accomplish this, a specific dataset is extracted from the data collected using prototype based on newly created I-DEEP protocol that has been discussed extensively in chapter-6, which serves as a representative subset of the entire dataset. The validation sample obtained should consist of equal distribution and representation of complete dataset.

7.5 Digital Intelligent Forensic Framework (DIF^2)

The integration of AI in computer forensics holds great potential to enhance investigative processes, mitigate resource constraints, and improve the outcomes of forensic investigations. By leveraging AI-based systems and case-based reasoning, investigators can augment their capabilities and achieve more efficient and effective results in uncovering crucial evidence. Digital Intelligent Forensic Framework (DIF^2) is a proposed AI framework that implements features like Prediction Modelling using AI algorithms, best AI algorithm selection, pipeline optimisation and predictive analysis.

Figure 7.1 provides a conceptual model for 'Digital Intelligent Forensic Framework' also referred as 'DIF^2' in further deliberations in upcoming sections and chapters. This newly developed artifact (framework) will also witness maturity as new constructs are added to extend the functionality during implementation phase. This DIF^2 model resembles a 'gearbox' analogy, where the best performing model can be selected to achieve best performance. The integration of AI has been demonstrated by three step process. First step involves creating or customising an algorithm by representing the logical process as a generalised algorithm created in a step-wise manner. In second stage, this algorithm is then encoded and represented in detail as a neutral representation of pseudocode (labelled as procedure), which is language independent. Finally, the actual implementation of the algorithm and pseudocode is done in Python language in the prototype.

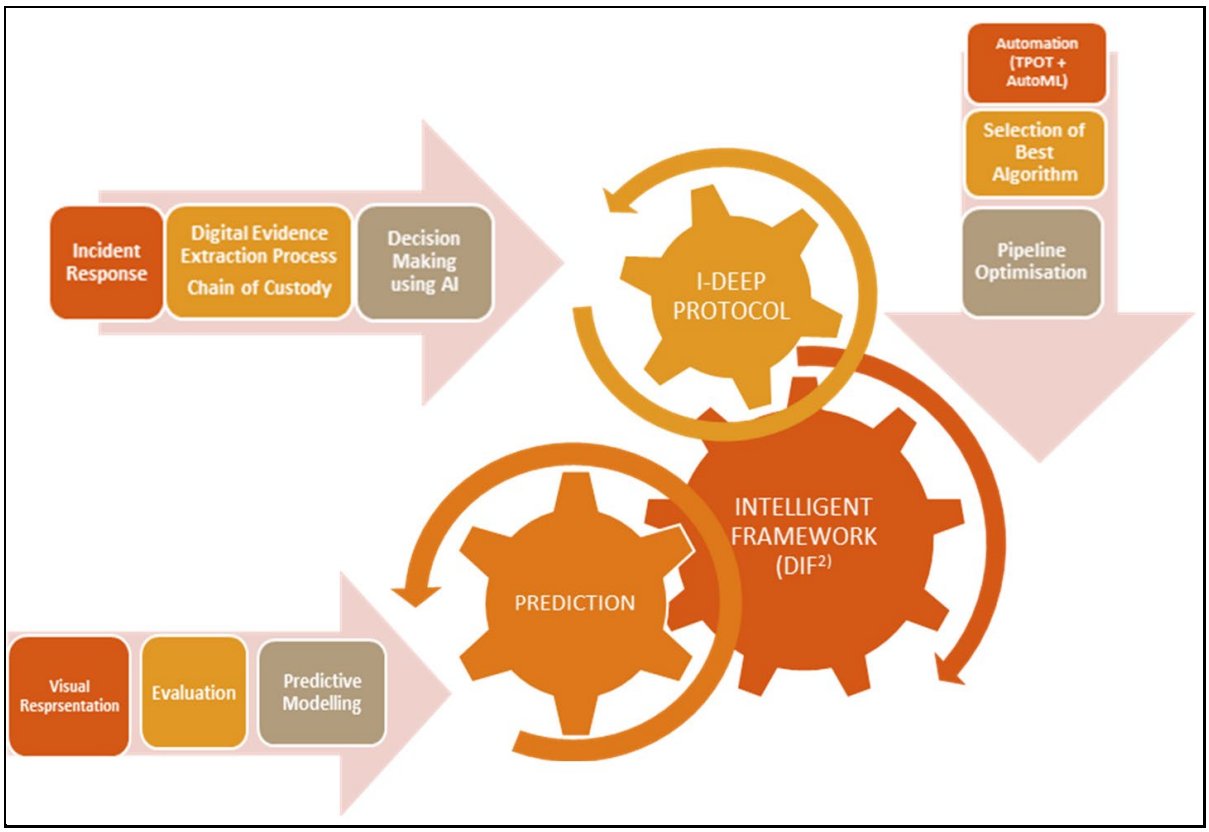


Figure 7.1: Ontology of Digital Intelligent Forensic Framework (DIF^2)

Incorporating artificial intelligence into computer forensics has significant potential to enhance investigative procedures, alleviate resource limitations, and enhance the overall outcomes of forensic investigations. Through the utilization of AI-driven systems and case-based reasoning, investigators can expand their capabilities, leading to more streamlined and successful endeavours in uncovering critical evidence. The Digital Intelligent Forensic Framework (DIF^2) represents a proposed AI framework that incorporates functionalities such as AI algorithm-based prediction modelling, the selection of the most suitable AI algorithms, pipeline optimization, and predictive analysis.

Data Rescaling

Numerous machine learning algorithms require data to be consistently scaled. When it comes to scaling your data for machine learning, two widely used methods used are 'Normalisation' and 'Standardisation'. Therefore, it is advisable to effectively rescale the data for optimal performance in machine learning (ML) tasks.

In situations where dataset consists of attributes with disparate scales, numerous machine learning algorithms can derive advantages from standardizing the attributes to share a uniform scale. Typically, this process is referred to as "normalization," wherein attributes are typically rescaled to fall within the range of 0 and 1. This normalization practice is particularly beneficial for optimization algorithms employed at the core of machine learning techniques, such as gradient descent (Brownlee, 2016).

Moreover, it proves advantageous for algorithms that assign weights to inputs, such as regression and neural networks, as well as for algorithms that employ distance metrics, like k-Nearest Neighbors. To perform normalization, it is essential to have knowledge of both the minimum and maximum values associated with each attribute. These values can either be approximated using training data or provided directly if one possesses a profound understanding of the problem domain. Estimating the minimum and maximum values for each attribute within a dataset can be accomplished by iterating through the dataset's values. To carry out this rescaling operation, the MinMaxScaler class provided by scikit-learn can be utilised (Brownlee, 2016).

Data Standardisation

Standardization is a valuable technique for converting attributes with varying means and standard deviations, assuming a Gaussian distribution, into a standard Gaussian distribution with a mean of 0 and a standard deviation of 1. This method is particularly well-suited for algorithms that rely on the assumption of Gaussian distribution in input variables and perform better with rescaled data. Some examples of such techniques include linear regression, logistic regression, and linear discriminant analysis. To achieve standardization of dataset, one can make use of the StandardScaler class provided by scikit-learn.

7.5.1 Resampling of Dataset

The objective of resampling methods is to make the best use of your training data, with the aim of accurately estimating the performance of a model on new, unseen data. Precise estimates of performance can then be employed to assist in the selection of which set of model parameters to use or which model to opt for. After a model has been selected, the final model can be trained using the entire training dataset and can then be used for making predictions.

The two popular resampling methods that can be utilised are ‘train and test split’ of dataset and ‘k-fold cross-validation’. While ‘Train and Test Spit’ is a resampling method which is easier to implement and is also widely used method (Brownlee, 2016).

Resampling using Train and Test Split Method

This method involves splitting the dataset into two parts- (i) Training dataset; (ii) Test dataset. The model is trained using the training dataset within the context of a machine learning algorithm. The test dataset, on the other hand, is deliberately withheld and is subsequently utilized to assess the model's performance. Random selection is employed to allocate rows to each dataset in order to foster objectivity in the model's training and evaluation process.

Resampling using k-fold Cross Validation Split Method

A drawback associated with the train and test split method lies in the fact that it yields a somewhat noisy estimate of algorithm performance. In contrast, the k-fold cross-validation method (also known as cross-validation) represents a resampling technique that provides a more precise assessment of algorithm performance.

This is achieved by initially dividing the data into k distinct groups or "folds”. Subsequently, the algorithm undergoes training and evaluation k times, and the performance is consolidated by computing the mean performance score. Each of these data groups is referred to as a "fold", hence the name "k-fold cross-validation”.

The procedure involves training the algorithm on k-1 of the data groups and evaluating it on the kth group held out as the test set. This process iterates, ensuring that each of the k groups takes a turn as the test set. Consequently, it is crucial that the value of k be evenly divisible by the number of rows in the training dataset, to ensure that each of the k groups contains an equivalent number of rows. When properly configured, k-fold cross-validation offers a robust performance estimate compared to other methods such as the train and test split.

Although the benchmark for assessing the performance of machine learning algorithms on new data is k-fold cross-validation. However, a drawback of cross-

validation is its potential time-consuming nature, as it necessitates training and evaluating k distinct models. This can pose challenges when dealing with extensive datasets or models that require significant training time.

In contrast, the 'train and test split' resampling method remains the most commonly employed approach. Its popularity stems from its simplicity of understanding and implementation, as well as its ability to provide a swift assessment of algorithm performance. With this method, only a single model is built and evaluated. Although it may yield a somewhat noisy or less reliable performance estimate on new data, this limitation becomes less pronounced when dealing with very large datasets.

When comparing multiple algorithms or different configurations of the same algorithm, it is imperative to employ the identical split of the dataset for both training and testing. This consistency ensures a fair comparison of performance. Achieving this consistency can be accomplished by applying the same seed to the random number generator before splitting the data or by preserving the same dataset split for use across multiple algorithms. Algorithm for the process is mentioned here.

7.5.2 Algorithm Train_Test_Split_Resampling

Step 1: Split out validation dataset into array;

Step 2: Assign dataset values to array;

Step 3: Select all rows and all columns except the last one (by specifying ':' in the rows index, and ':-1' in the columns index) in X;

Step 4: Select last column in y;

Step 5: Fetch training data in the *X_train* and *Y_train* for preparing models and a *X_validation* and *Y_validation* sets for test;

Step 6: Split the loaded dataset into 80% to train, evaluate and select models, 20% as a validation dataset;

Algorithm 7.1: Algorithm_Train_Test_Split_Resampling

Actual implementation code in Python for ‘train and test split’ resampling and creating validation dataset is shown in section-B.6 (APPENDIX-B).

7.5.3 Implementation of k-fold Cross Validation Split Method

The ‘k-fold cross-validation’ method (also known as cross-validation) represents a resampling technique that provides a more precise assessment of algorithm performance. This is achieved by initially dividing the data into k distinct groups or "folds". Subsequently, the algorithm undergoes training and evaluation k times, and the performance is consolidated by computing the mean performance score. Each of these data groups is referred to as a "fold". The procedure involves training the algorithm on k-1 of the data groups and evaluating it on the kth group held out as the test set. This process iterates, ensuring that each of the k groups takes a turn as the test set.

A stratified 10-fold cross validation is an effective technique that can be used to estimate model accuracy. This method splits the dataset into 10 parts, train on nine parts or ‘folds’, test on one fold and repeat the process for all combinations of train-test splits. Stratified means that each fold or split of the dataset will aim to have the same distribution (e.g. by ‘City’) as exist in the whole training dataset. ‘Random State=1’ (i.e. seed=1) is used to keep same folds in order to uniformly test all AI models against same samples.

7.6 Create a Baseline Model for Predictive Modelling

Establishing a baseline performance for a predictive modelling problem holds significance as it furnishes a reference point against which the more advanced methods can be compared. Hence creating a baseline model can be beneficial to provide a performance comparison between different AI based interventions implemented as part of this study.

A set of predictions can be provided by a baseline prediction algorithm, which can be evaluated using standard metrics like classification accuracy or Root Mean Square Error (RMSE), just as one would evaluate any predictions for their problem. The required point of comparison is provided by the scores obtained from these algorithms when evaluating all other machine learning algorithms for various classification or regression problems. Once it is established, comments can be made regarding the

degree of improvement achieved by a given algorithm in comparison to the naïve baseline algorithm, thereby offering context on the actual effectiveness of the method in question.

Two most common methods used for creating a baseline model, which can be used in a naïve baseline algorithm are ‘Random Prediction Algorithm’ and ‘Zero Rule Algorithm’.

7.6.1 Random Prediction Algorithm

A random outcome is predicted by the random prediction algorithm, reflecting the patterns observed in the training data. It is arguably one of the easiest algorithms to put into practice. Storing all distinct outcome values from the training data is a requirement, which can be particularly cumbersome when dealing with regression problems featuring a multitude of unique values. Given that random numbers guide the decision-making process, it is advisable to set a fixed seed for the random number generator before employing the algorithm. This ensures the consistency of the generated random numbers and, consequently, the same decision-making outcomes each time the algorithm is executed.

7.6.2 Zero Rule Algorithm

The Zero Rule Algorithm serves as a superior baseline model compared to the random prediction algorithm. It leverages a greater amount of information specific to the problem at hand to formulate a single rule for making predictions. This rule, however, varies depending on the nature of the problem being addressed. Considering the instance of classification problems, one rule is to predict a class value that occurs most frequently in a training dataset. For instance, in case of binary classification where the class value would be ‘0’ or ‘1’, a training dataset has 90 instances of ‘0’ and 10 instances of ‘1’. If ‘0’ represents negative outcome and ‘1’ represents positive outcome, then the accuracy for prediction of negative outcome is 90% while predicting positive outcome is 10% using zero rule algorithm.

In regression problems, the objective is to predict a continuous real value. A suitable default prediction strategy for real values involves forecasting the central tendency of the data. This central tendency can be represented by either the mean or the median of the target variable (Brownlee, 2021).

Equation: $$mean = \frac{\sum_{i=1}^{n} value_i}{count\ (values)}$$ Eq. (7.1)

Eq. 7.1 describes the formula for calculating Zero Rule Algorithm Regression

7.6.3 Algorithm Baseline_Modelling_Zero_Rule_Classification

Step1: Collect Training Data in a dataset with known outcomes (classes or labels);

Step2: Search for most common outcomes;

Step3: Make predictions using test data;

Step4: Display predictions;

Algorithm 7.2: Algorithm_Baseline_Modelling_Zero_Rule_Method_Classification

7.6.4 Algorithm Baseline_Modelling_Zero_Rule_Regression

Step1: Collect Training Data (each with numerical outcome);

Step2: Calculate the average (mean) by adding all outcomes and dividing my number of outcomes;

Step3: Make predictions for test data (using the mean calculated in step-2);

Step4: Display Predictions;

Algorithm 7.3: Algorithm_Baseline_Modelling_Zero_Rule_Method_Regression

7.6.5 Pseudocode for Baseline Modelling Zero Rule Algorithm in Classification Problems

Procedure 7.1: Baseline Modelling Zero Rule Algorithm in Classification Problem

Input: Cybercrime incidents dataset with features - ('Phishing', 'ZeroDayAttack', 'Ransomeware', 'ChildPorn'); Class - ('City')

Output: CV Score (Cross Validation Score Prediction)

```
Function zero_rule_algorithm_classification (train, test):
    // Initialize an empty list to store output values from training
data
    output_values = []
```

```
    // Loop through each row in the training data
    For each row in train:
        // Extract the last element (the output value) and add it to
output_values
        Append row[-1] to output_values
    // Find the most frequent class in the output_values list
    prediction = Max(set(output_values),
key=CountOccurrences(output_values))
    // Initialize an empty list for predictions
    predicted = []
    // Loop through the test data
    For each example in test:
        // Add the 'prediction' to the 'predicted' list
        Append prediction to predicted
    // Return the list of predictions
    Return predicted
// Set a random seed for reproducibility (optional)
Seed(1)
// Example training and test data
train = X [Array of input (feature) column (class)]
test = y [Array of output column (class)]
// Make predictions using the Zero Rule Algorithm
predictions = zero_rule_algorithm_classification(train, test)
// Print the predicted values
Print(predictions)
```

The procedure 7.1 describes the pseudocode for ‘Zero Rule Algorithm’ for classification problems in a way that can be implemented in various programming languages. There may be a need to adapt certain language-specific functions or syntax, such as Append, total_sum, Seed, and Print, to the conventions of the language that is being used. We have used Python while implementing these algorithms in this study. This algorithm implementation is dealing with a classification problem where we are attempting to predict ‘Critical’ and ‘Non- critical’ status in cybercrime incidents in a dataset by implementing triaging using ‘TriageScore’ calculated by creating a formula (severity x occurrence), therefore we have encoded this logic in the pseudocode (procedure) and finally implemented the customised algorithm using Python. The

actual Python implementation code along with output for 'Baseline Algorithm' for Classification Problems is represented in Section B.2 (Figure B.2 in APPENDIX-B). The output of the model showed an accuracy score of **45%** at 60-40 split into train and test datasets, which is an acceptable base score to start. The split of dataset into train and test datasets is achieved by creating a test harness, where data is compared and performance is evaluated in an iterative manner.

7.6.6 Pseudocode for Baseline Modelling of Zero Rule Algorithm in Regression Problems

Procedure 7.2: Baseline Modelling Zero Rule Algorithm in Regression Problems

Input: Cybercrime incidents dataset with features - ('Phishing', 'ZeroDayAttack', 'Ransomeware', 'ChildPorn');

Output: Severity Score Prediction

```
Function zero_rule_algorithm_regression(train, test):
    // Initialize an empty list to store output values from training
data
    output_values = []
    // Loop through each row in the training data
    For each row in train:
        // Extract the last element (the output value) and add it to
output_values
        Append row[-1] to output_values
    // Calculate the average of the output values
    total_sum = 0
    For each value in output_values:
        total_sum = total_sum + value
    prediction = total_sum / float(len(output_values))
    // Initialize an empty list for predictions
    predicted = []
    // Loop through the test data
    For each example in test:
        // Add the 'prediction' to the 'predicted' list
        Append prediction to predicted
    // Return the list of predictions
    Return predicted
```

```
// Set a random seed for reproducibility (optional)
Seed(1)
// Example training and test data with numerical outcomes
train = dataset (0);
test = [[None], [None], [None], [None]]
// Make predictions using the Zero Rule Algorithm for regression
predictions = zero_rule_algorithm_regression(train, test)
// Print the predicted values
Print(predictions)
```

Procedure 7.2 describes Baseline Model of Zero Rule Algorithm for regression problems in covered in Section-7.6.6. The pseudocode describes the Zero Rule Algorithm for regression problems in a way that can be implemented in various programming languages. Since we have created a Baseline (Naïve) Zero Rule Algorithm, we can use this as a benchmark for making predictions and evaluating the performance of other AI model interventions. The study also compares performance of most commonly used AI algorithms and select best performing algorithm and evaluate the prediction results discussed in section-9.3.

7.6.7 Implementation of Baseline Model to Cybercrime Dataset for Prediction

We observed that our Baseline Model produced an accuracy of **45%** overall for classification predictive modelling using 60-40 split method. More parameter tuning in order to improve performance has been demonstrated in section-9.3.1 where an improved performance of **48%** was observed. These baseline observations will be used to benchmark other implementations in upcoming sections. The next stage of implementation is to perform a manual evaluation of existing AI algorithms most commonly employed for predictive analysis. This implementation has been described in detail under 'Experiment-1, Stage-2' of 'Research Experiment' in section-5.7. A test harness was created to test these algorithms and compare the performance as demonstrated in upcoming sections.

7.7 Implementing AI algorithms using Test Harness Method

The main goal of this AI implementation is to identify the 'best performing' AI algorithm and use it for predictive analysis on dataset. A test harness was created using stratified 10-fold cross

validation method to ‘spot-check’ various AI algorithms, the best performing model is selected and output is evaluated.

7.7.1 Spot-Checking of AI Algorithms

Various AI algorithms can be spot-checked using the test harness created in order to select best performing AI algorithm models like Logistic Regression (LR), Linear Discriminant Analysis (LDA), KNeighbors Classifier (KNN), Decision-Tree Classifier (CART), Gaussian NB (NB) and Support Vector Machines (SVM). This process helps the experts to evaluate the performance of existing algorithms and select the best performing algorithm, which can then be used further for analysing the problem. Different algorithms perform differently depending on datasets and their types. An algorithm created to implement this spot checking is represented as algorithm 7.4 and detailed in section 7.7.2, which describes the steps involved in the process.

7.7.2 Algorithm_Spot_Check_AI_Algorithms

Step1: Create a list of AI models to spot-check algorithms;

Step2: Add all models to the list;

Step3: Create array for model names;

Step4: Create array for results;

Step5: Evaluate each model iteratively;

Step6: Initialise n_splits=10, random_state=1, shuffle=True;

Step7: Compare attribute pairs to determine cross-validation score and correlation

Step8: Capture results for ‘accuracy’ score after cross-validation for each model;

Step9: Display descriptive statistical data for results (mean, standard deviation);

Algorithm 7.4: Algorithm_Spot_Check_AI_Algorithms

```
from sklearn.neighbors import KNeighborsClassifier
from sklearn.discriminant_analysis import LinearDiscriminantAnalysis
from sklearn.naive_bayes import GaussianNB
from sklearn.svm import SVC
# Load dataset
url = "https://raw.githubusercontent.com/deep0025/CyberCrimeData-SA/main/CybercrimeDataset50.csv"
names = ['Phishing', 'ZeroDayAttack', 'RansomwareAttack', 'ChildPorn', 'City']
dataset = read_csv(url, names=names)
# Split-out validation dataset
array = dataset.values
X = array[:,0:4]
y = array[:,4]
X_train, X_validation, Y_train, Y_validation = train_test_split(X, y, test_size=0.20, random_state=1, shuffle=True)
# Spot Check Algorithms
models = []
models.append(('LR', LogisticRegression(solver='liblinear', multi_class='ovr')))
models.append(('LDA', LinearDiscriminantAnalysis()))
models.append(('KNN', KNeighborsClassifier()))
models.append(('CART', DecisionTreeClassifier()))
models.append(('NB', GaussianNB()))
models.append(('SVM', SVC(gamma='auto')))
# evaluate each model in turn
results = []
names = []
for name, model in models:
    kfold = StratifiedKFold(n_splits=10, random_state=1, shuffle=True)
    cv_results = cross_val_score(model, X_train, Y_train, cv=kfold, scoring='accuracy')
    results.append(cv_results)
    names.append(name)
    print('%s: %f (%f)' % (name, cv_results.mean(), cv_results.std()))
# Compare Algorithms
pyplot.boxplot(results, labels=names)
pyplot.title('Algorithm Comparison')
pyplot.show()
```

```
LR: 0.625000 (0.201556)
LDA: 0.525000 (0.207666)
KNN: 0.625000 (0.201556)
CART: 0.300000 (0.244949)
NB: 0.525000 (0.261008)
SVM: 0.475000 (0.134629)
```

Code Snippet 7.1: Implementation and output of Spot-check Algorithm on cybercrime dataset

Complete implementation code of the algorithm 7.4 using Python language is represented in section B.3 (Figure-B.3, APPENDIX-B) along with the output of the program on a cybercrime dataset. While the implementation using dataset with 50 records show LR and KNN as best performing algorithms (accuracy score-62.5%). The implementation with dataset of 100 records shows Linear Discriminant Analysis (LDA) has highest accuracy mean value of **67%** and a standard deviation of 17.8% as shown in figure-B.3. This shows that different datasets possess varied characteristics and therefore will show different results on evaluating the performance during predictive modelling. Also, the dataset size has a positive correlation on prediction performance. This discussion is limited to explain the implementation process and observe that different dataset sizes produces different accuracy scores whereby an increase in dataset size produces better accuracy. The detailed experimental evaluation is provided in chapter-9 (section 9.3.1) where a comparative analysis of model performance is done using variable dataset sizes initially with 50 vs 100 records and thereafter with 50, 100, and 300 records.

7.8 Visual Comparison of AI Algorithm Models

Visual representation of performance results of AI models, using various charts and diagrams for assessing the distribution of individual attributes, can be a useful approach that helps understand the other valuable characteristics and behaviour of models. This involves employing Box and Whisker Plots, commonly referred to as boxplots. Boxplots provide a

concise summary of attribute distributions by representing the median (the central value) with a line and enclosing the 25th and 75th percentiles (the middle 50% of the data) within a box. The whiskers extending from the box offer insights into the data's range, while any data points lying beyond the whiskers are indicative of potential outliers' values that exceed 1.5 times the size of the spread within the middle 50% of the data.

The algorithm performance comparison, represented as Box and Whisker Plot is shown in figure-7.3. The comparison of performance of various AI algorithm models using box and whisker plots, shows that Linear Discriminant Analysis (LDA) is the best performing among other algorithms. Implementation code in Python and output of the program is shown in section B.4 (Figure B.4, APPENDIX-B).

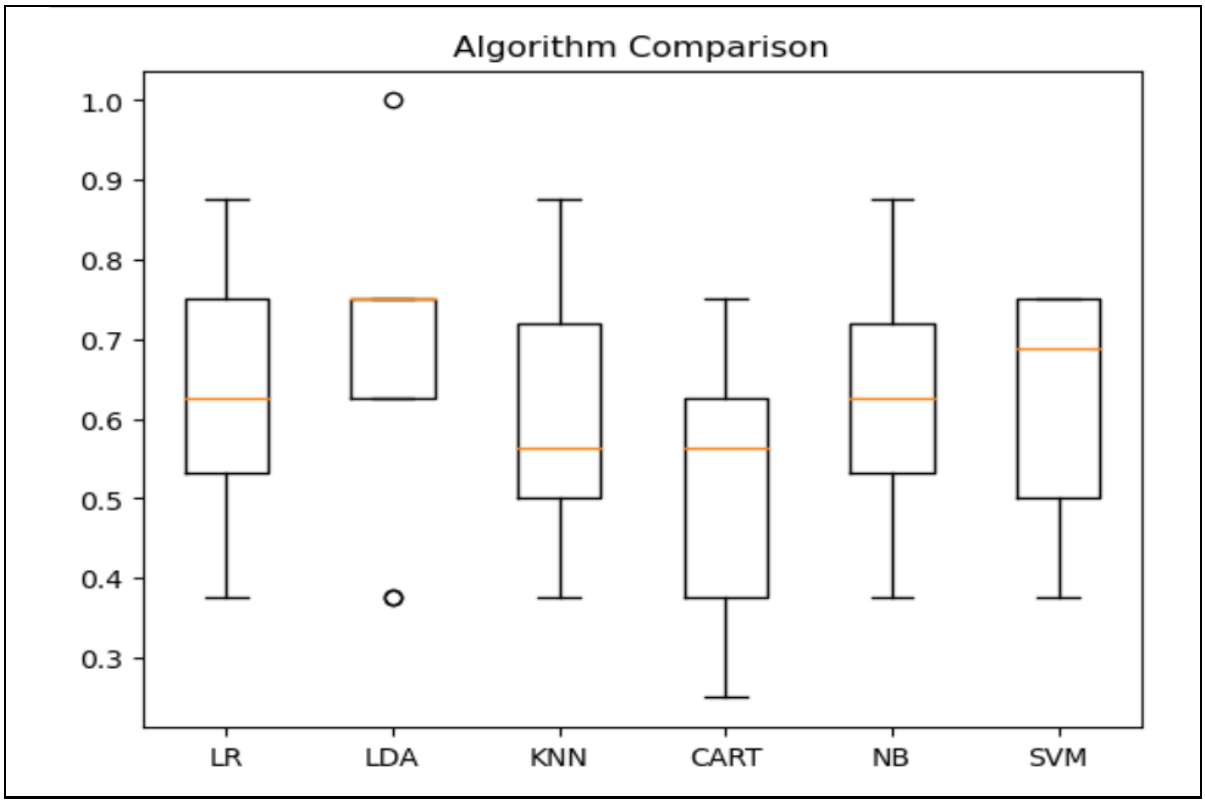


Figure 7.2: Visual comparison of AI algorithms using Box and Whisker Plot

The program loads dataset, splits out validation datasets, spot-checks algorithms, evaluates each model in turn and finally compares algorithms performance. A complete implementation of program is provided in figure B.3 (Section B.3, APPENDIX-B).

7.9 Algorithm_Calculate_Triage_Score

Step1: Normalise data

Step2: Define a function to find min and max values for each column

Step3: Define parameterised function minmax (dataset)

Step4: Create a list datatype called minmax;

Step3: Create function called normalise_dataset with parameters as dataset and minmax list;

For each row in dataset:

Column values = row (present row number) from rows in dataset;

store value_min=min(Column values);

store value_max=max(column values);

For each row in dataset:

For each column:

$$New\ scaled\ value\ = \frac{value\ -\ min}{max\ -\ min}$$

add the min and max values to minmax list;

return minmax;

Step4: Calculate Triage_Score based on crime severity index

For each column:

Crime=column_value;

If crime= ‘Phishing’

Then severity=1;

Else If crime= ‘ZeroDayAttack’

Then severity=2;

Else If crime= ‘RansomwareAttack’

Then severity=3;

Else crime= ‘ChildPorn’

Then severity=4;

$TriageScore += severity\ x\ occurance$

Add TriageScore to a new column;

If TraigeScore > 5

Then Class = Critical (C);

Else Class = Non_critical (N);

Add column_value to ‘Class’;

Algorithm 7.5: Algorithm_Calculate_Triage_Score

A new algorithm named ‘Calculate_Triage_Score’, represented as Algorithm 7.5, is created to calculate a triage score for computing the status of cybercrime severity. This algorithm is implemented in the DIF^2 framework to calculate the ‘severity score’ using cybercrime incidents. This algorithm uses a severity point-scale index based on cybercrime ‘type’ feature in dataset (Phishing=1; ZeroDayAttack=2; Ransomware=3; ChildPorn=4), multiplied by the number of cybercrime incidents, to arrive at a ‘triage score’ that is used for triaging and decision making. This score is used in prediction modelling to provide intelligence to first responders and investigators, to evaluate the severity of cybercrime landscape, in order to plan their resources and achieve effective decision making.

7.10 Implementing Prediction Modelling

In order to implement prediction modelling, some specific AI algorithms have been discussed here. These AI algorithms forms the basis of further implementation of more complex algorithms and finally creation of a custom algorithm which is developed specifically for the problem at hand i.e. prediction modelling of cybercrime incidents to perform triaging.

7.10.1 Linear Regression

Linear regression, a prediction technique with a history spanning over two centuries, serves as an excellent introductory machine learning algorithm. Its simplicity lies in the fact that it necessitates the estimation of parameters based on training dataset, making it an ideal starting point.

Linear regression operates under the assumption of a direct, linear relationship between the input variables (X) and a single output variable (y). In precise terms, it posits that the output (y) can be computed as a linear sum of the input variables (X) as represented in eq. (7.2). In cases where only one input variable is involved, this approach is termed as 'Simple Linear Regression' or SLR. Within the realm of SLR, statistical analysis of the training data can be used to determine the coefficients necessary for the model to generate predictions on fresh data points.

Equation: $\mathbf{y = b0 + b1 \times x}$ Eq. (7.2)

Here 'b0' and 'b1' are estimated coefficients from the training data available. The output values 'y' can be estimated from input values 'x' using the coefficients known using the equation. Other statistical properties of the data such as mean, variance and covariance can be calculated as well.

7.10.2 Multivariate Linear Regression

At the heart of numerous machine learning algorithms lies the concept of 'optimization'. Machine learning algorithms employ optimization techniques to discover an optimal set of model parameters based on a provided training dataset. Among the various optimization algorithms employed in the field, stochastic gradient descent (SGD) stands out as the most prevalent and widely used method.

Linear regression is a method used to make predictions for real-valued outcomes. Somewhat surprisingly, these types of problems, where the objective is to forecast a real value, are referred to as 'Regression Problems'. In linear regression, a straightforward approach is employed to model the relationship between input and output values. When dealing with more than two dimensions, this straightforward line can be envisioned as a plane or a hyperplane.

The process of prediction involves combining the input values to anticipate the output value. Each input attribute (represented as 'x') is assigned a weight using a coefficient (denoted as

‘b’). The primary objective of the learning algorithm is to uncover a set of coefficients that yield accurate predictions (represented as ‘y’). Eq. (7.3) represents the formula used in calculation.

$$\textbf{Equation: } y = b0 + b1 \times x1 + b2 \times x2 + \ldots \quad \text{Eq. (7.3)}$$

In order to find coefficient that can be used to make most accurate predictions, ‘Stochastic Gradient Descent’ can be an effective method. Gradient Descent is a process employed to minimize a function by following its slope or gradient. In the context of machine learning, a technique known as Stochastic Gradient Descent (SGD) is employed to minimize the model's error on training data by evaluating and adjusting the coefficients at each iteration.

The functioning of this optimization algorithm involves presenting each training instance to the model one at a time. The model generates a prediction for a training instance, calculates the error, and updates the model to reduce the error for subsequent predictions. This cycle is repeated for a predetermined number of iterations. This approach is employed to determine the set of coefficients in a model that yield the smallest error when applied to the training data. In machine learning terminology, the coefficients (often represented as ‘b’) are updated with each iteration of the algorithm to progressively refine the model. Eq. (7.4) represents the formula used in the calculation.

$$\textbf{Equation: } b = b - \text{learning rate} \times \text{error} \times x \quad \text{Eq. (7.4)}$$

In the context of machine learning and optimization algorithms like Stochastic Gradient Descent, the key elements in the equation are "b" representing the coefficient or weight that is being optimized or adjusted during the learning process; "Learning rate" (often denoted as "α") is a hyper-parameter that needs to be configured, typically to a value like 0.01. It governs the step size taken during each iteration of the optimization process. Parameter "error" refers to the prediction error of the model on the training data, which is attributed to the weight "b". This error quantifies how far off the model's predictions are from the actual training data; "x" represents the input value, which is part of the ‘feature’ set used by the model to make predictions. In stochastic gradient descent (SGD), these elements work together to iteratively adjust the coefficients "b" to minimize the prediction error and ultimately improve the model's performance on the training data. The “learning rate” controls the size of each step in the weight updates, and the “error” provides the feedback needed to guide these updates in the right direction.

In order to derive coefficient values for training data, stochastic gradient descent or SGD can be employed, that consist of a process requiring the specification of two parameters:

Learning Rate: This parameter constrains the extent to which each coefficient is adjusted every time an update is made.

Epochs: The number of iterations through the training data, during which the coefficients are updated.

These parameters, along with the training data, serve as arguments for the function. The function involves three nested loops: An outer loop iterates over each epoch. Within each epoch, there is a loop that traverses each row in the training data. Another nested loop is responsible for updating each coefficient for a specific row within an epoch. As this process unfolds, every coefficient for every row in the training data is updated for each epoch. These coefficient updates hinge on the model's prediction error, calculated as the discrepancy between the predictions generated using the candidate coefficients and the expected output value as shown in eq. (7.5).

$$\textbf{Equation: } \mathbf{error} = \mathbf{prediction} - \mathbf{expected} \qquad \text{Eq. (7.5)}$$

Each input results in weighing each coefficient consistently as shown in eq. (7.6)

$$\textbf{Equation: } \mathbf{b1}(\mathbf{t}+\mathbf{1}) = \mathbf{b1}(\mathbf{t}) - \mathbf{learning\ rate} \times \mathbf{error}(\mathbf{t}) \times \mathbf{x1}(\mathbf{t}) \qquad \text{Eq. (7.6)}$$

The special coefficient (b0) in the beginning is called the “intercept or bias” is updated in similar fashion but without any input as shown in eq. (7.7).

$$\textbf{Equation: } \mathbf{b0}(\mathbf{t}+\mathbf{1}) = \mathbf{b0}(\mathbf{t}) - \mathbf{learning\ rate} \times \mathbf{error}(\mathbf{t}) \qquad \text{Eq. (7. 7)}$$

7.10.3 Artificial Neural Networks - Back Propagation Method

The ‘Backpropagation Algorithm’ is a supervised learning technique applied to multilayer feedforward networks within the domain of Artificial Neural Networks. These neural networks draw inspiration from the information processing capabilities of individual neural cells, known as neurons. A neuron receives input signals through its dendrites, transmitting these electrical signals to the cell body. The axon, in turn, carries the signal to synapses, which are connections between one cell's axon and another cell's dendrites.

The fundamental concept behind the backpropagation approach is to model a given function by adjusting the internal weightings of input signals to generate an anticipated output signal. This system undergoes training using supervised learning, where the error between the system's output and a predetermined expected output is introduced to the system. This error information is then utilized to modify the system's internal configuration. Technically speaking, the backpropagation algorithm serves as a means to train the weights within a multilayer feedforward neural network. This necessitates defining a network structure comprising one or more layers, with each layer being fully interconnected to the next. A common network structure consists of an input layer, a hidden layer, and an output layer. Backpropagation can be employed for both classification and regression tasks.

In classification tasks, optimal performance is typically attained when the neural network features one neuron in the output layer corresponding to each unique class value. For instance, in a binary classification scenario with class labels A and B, the expected outputs need to be converted into binary vectors with a single column dedicated to each class value. This transformation is known as "one-hot encoding", where class A is represented as [1, 0] and class B as [0, 1], ensuring that each class is uniquely identifiable in the output layer.

In this discussion, we have found high potential in Artificial Neural Network (ANN) and Stochastic Gradient Descent (SGD) as they are very effective in prediction modelling for both classification and regression problems and provide high efficacy even if datasets are not large enough. Next chapter (chapter-8) contains detailed description of the ANN model as a customised novel algorithm and its implementation in the form of pseudocode (procedures) as well as in the form of Python code and discussion on the performance of the model is provided. Automation of AI framework (DIF^2), as well as interventions using TPOT and AutoML for automation and optimisation of the DIF^2 framework, and further improvements in performance of predictive analysis, is also discussed on upcoming chapter. In order to achieve these goals, a customised algorithm is developed and implemented using an ensemble of ANN and SGD in upcoming chapter (chapter-8). Therefore, a detailed background of ANN and SGD has been provided in this chapter to explain the functional aspects. A specialised form of ANN known as Convolutional Neural Network (CNN), which is capable of performing prediction on image data is also investigated and implemented as an ‘enhanced and customised’ model in chapter-8.

7.11 Summary

Digital forensic investigations face challenges in efficiently evaluating large volumes of data to uncover critical evidence and quick decision-making. Limited time and resources, including computational power and human expertise, hinder their effectiveness. To address these challenges, integrating artificial intelligence (AI) into digital forensics through an AI-based framework and case-based reasoning is proposed. This chapter resulted into creation of new algorithms that implements AI and Machine Learning to achieve decision making using 'triaging' and prediction modelling. By leveraging AI technologies like machine learning (ML) and pattern recognition, the proposed system aims to enhance the efficiency and accuracy of digital investigations. It can automate tasks, reduce manual effort, and enable faster analysis of extensive data. Case-based reasoning enables the system to learn from past cases data, improving decision-making and aiding in the discovery of relevant evidence. The integration of AI in computer forensics has the potential to optimize investigative processes, overcome resource constraints, and improve outcomes. The Digital Intelligent Forensic Framework (DIF^2) is introduced as a proposed framework that incorporates features like AI algorithm selection, pipeline optimization, and prediction analysis to enhance the investigative capabilities further. By adopting AI-based systems integration and case-based reasoning, investigators can achieve more efficient results, perform 'triaging' and quick decision-making. The process followed the 'iterative and incremental' model of 'agile approach' in order to integrate an intelligent framework using customised AI algorithms into the system prototype.

CHAPTER 8: INTELLIGENT FORENSICS MODELLING - DIF^2 AUTOMATION

8.1 Introduction

In this chapter, we introduce a novel architecture designed to incorporate machine learning and facilitate the development of an innovative AI-based intelligent framework, which we comprehensively discuss in this chapter and evaluate in subsequent chapters. We delve into the ontology modelling aspect of this new AI-based intelligent framework, emphasizing its potential for automation and the optimisation of pipeline techniques. Additionally, we provide an overview of their inherent capabilities in predictive modelling. Our novel framework harnesses the power of artificial intelligence (AI) and machine learning (ML) to automate critical tasks such as model selection, enabling the identification of the best-performing models for predictive modelling rooted in machine learning principles (Zhou et al., 2005).

However, it is important to acknowledge that even with automation and AI, achieving 100% accuracy and intelligence remains an elusive goal. Errors and false positives may still arise. Consequently, our AI framework will undergo rigorous testing, and we endeavour towards enhancing the accuracy and effectiveness of predictive modelling. Furthermore, we explore and evaluate the performance of several AI algorithms and investigate the possibilities to improve our predictions using customised or new algorithms and using ensemble methods.

Despite the potential for minimizing user inputs and human efforts through automation, we emphasize the importance of verification at each stage of the process to mitigate errors, especially in the case of system-generated results and reports. It is crucial that these results and reports are thoroughly verified and cross-checked with relevant data before being presented in a court of law.

8.2 Automated Machine Learning (AutoML)

Automated Machine Learning, commonly known as AutoML, encompasses the process of autonomously selecting data preprocessing steps, machine learning models, and model hyperparameters to tackle predictive modelling tasks. AutoML comprises of a set of methodologies that empower both moderately experienced machine learning professionals and individuals without extensive expertise to efficiently identify an optimal predictive model pipeline for their specific machine learning projects. This is achieved with minimal manual intervention, primarily involving the provision of a dataset and TPOT + AutoML technology framework (Feurer et al., 2015). TPOT and AutoML provides a promising start for

implementing and achieving our automation objectives towards development of an intelligent framework referred as Digital Intelligent Forensic Framework (DIF^2) represented in figure-8.1.

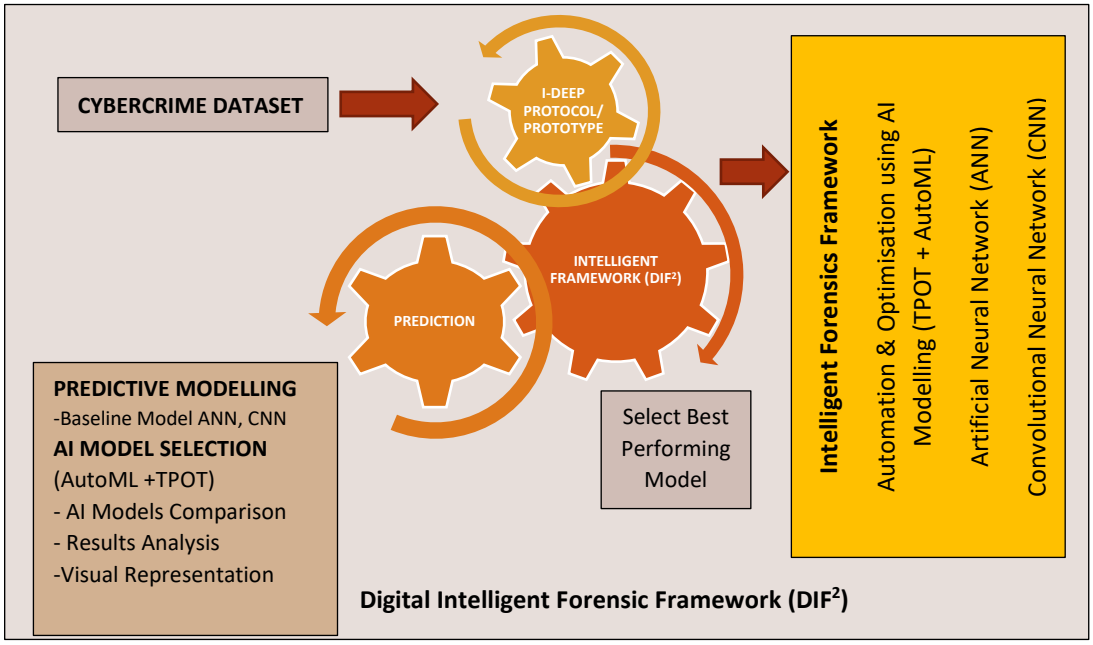


Figure 8.1: Ontology of automation of Digital Intelligent Forensic Framework (DIF^2) using TPOT and AutoML

8.2.1 Operational Ontology of Digital Intelligent Forensic Framework (DIF^2) Automation

Automation is a salient feature of AI and it is widely being utilised to create intelligent solutions that reduce user input and produce acceptable results and better productivity for users. Therefore, we conceptualise a new framework based on AI technology that has the potential to automate the process of model comparison, selection of the best performing model and then optimisation of pipelines to generate high accuracy in predictive modelling.

Figure 8.1 presents an ontological illustration of automation of newly developed Digital Intelligent Forensic Framework (DIF^2) which utilizes AutoML and TPOT implementations to achieve automation in AI algorithm comparison, selection of best performing algorithm and performing pipeline optimization.

The newly developed AI framework, referred as DIF^2, is a case-based model and along with its integration with I-DEEP protocol, it is compatible with all the elements of the digital forensic process. The majority of the already available software tools are capable of

implementing incident response and data extraction capabilities, but they lack the sophisticated interoperable intelligence as discussed in previous chapters and the proposed intelligent framework (DIF^2) and system formulation addresses this issue.

The new framework is constructed using well-known classical programming, AI and ML techniques, where pre-existing datasets from prior forensic investigations are utilised for prediction modelling. For the system to comprehend which decision to make in a given situation, certain sets of data are helpful. In the subsections that follow, AI integration at each point and operational ontology is explained. Since AI systems operate in accordance with their knowledge and what they have learned, the measures in this context are referred to as intelligent systems and thus intelligent forensics. For consistent results after training, each step needs to be retrained. Based on contextual performance measures, more training datasets have been utilised to thoroughly train the AI algorithms after the test data sets have been used to validate the learning process as also recommended by Yuki et al. (2019).

The ability to use domain knowledge and iterated learning (either by an intelligent human agent or meta learning methods) to determine what the key features of the prediction task are and how to optimize the hyperparameters of the learning algorithms for successful learning over large databases, which cannot be pruned or cleaned by a human data scientist, is one of the most crucial aspects of creating an automated prediction model. Therefore Automated Machine Learning (AutoML) refers to techniques for automatically discovering well-performing models for predictive modelling tasks with very little user involvement (Feurer et al., 2015).

8.2.2 Tree-based Pipeline Optimization Tool (TPOT)

TPOT also referred as T-POT many times, is an open-source Python library with a primary focus on automated machine learning (AutoML). It capitalizes on the widely-adopted 'scikit-learn' machine learning library, using it for tasks like data transformations and the implementation of machine learning algorithms. TPOT leverages the potency of Genetic Programming, a stochastic global search approach, to methodically and efficiently discover an ideal model pipeline that can attain remarkable performance when confronted with a specific dataset.

In the realm of TPOT, a tree-based structure model, serves as the representation for a model pipeline designed for predictive modelling tasks. This encompassing structure model includes components for data preprocessing, various modelling algorithms, and model hyperparameters.

TPOT functions as an evolutionary algorithm, suitably named the Tree-based Pipeline Optimization Tool (TPOT), which constitutes the automatic designing and optimization of ML pipelines. After data preprocessing, an optimization process is set in motion to pinpoint the most effective tree structure for achieving superior performance with a given dataset. This approach relies on a carefully designed genetic programming algorithm tailored to carry out stochastic global optimization, utilizing program representations in the form of trees (Evaluation of a Tree-based Pipeline Optimization Tool for Automating Data Science, 2016).

Within TPOT, a modified version of genetic programming is harnessed to automatically create and refine a series of data transformations and ML models. The primary goal is to optimize classification accuracy in the context of a particular supervised learning dataset.

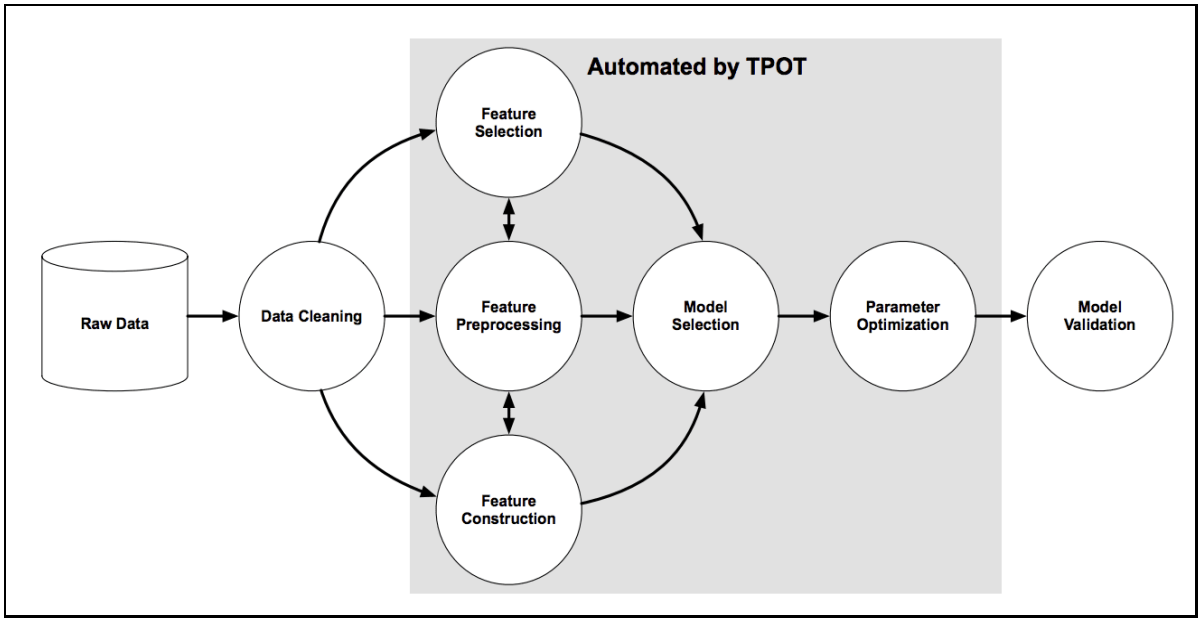


Figure 8.2: Overview of the TPOT Pipeline Search (Evaluation of a Tree-based Pipeline Optimization Tool for Automating Data Science, 2016).

Figure 8.2 illustrates the various components engaged in the pipeline search process. These components include data cleaning, feature selection, feature processing, feature construction, model selection, and hyperparameter optimization.

8.2.3 TPOT & AutoML Integration for Automation for DIF² Framework

The newly proposed implementation will make use of the TPOT and AutoML framework. This framework possesses the capability to systematically explore a multitude of possibilities in search of the optimal 'Model Fit'. As depicted in Figure 8.3, the framework accommodates the addition of new data to models that have been trained in previous runs. This functionality

enables TPOT to skilfully integrate knowledge from various learned pipelines and leverage the combined data from past user runs to identify the most effective model pipeline for predictive modelling. This process requires minimum input from user and can provide an effective framework for automation as well as for facilitating Intelligent Forensic (IF) Analysis.

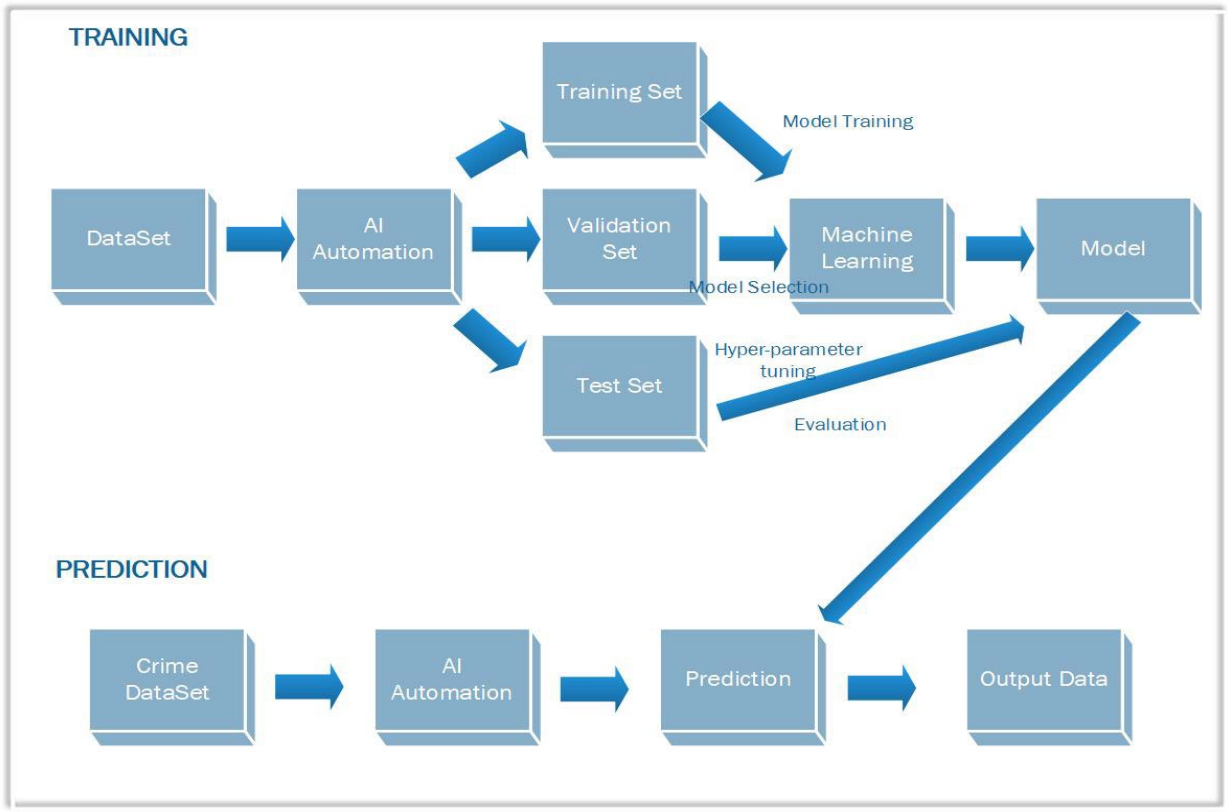


Figure 8.3: Operational Ontology of Model Training and Prediction Process implemented in DIF^2 Framework Automation, adapted from Johnson & Ananthakumaran (2021).

8.3 Operational Ontology of Digital Intelligent Forensic Framework (DIF^2) Automation

In this context, proficient AutoML pipeline implementations play a pivotal role in satisfying the core prerequisites of the DIF^2 Automation Architecture. Furthermore, open-source libraries are available, which specialize in AutoML techniques. These libraries focus on specific aspects, including data transformations, model selection, and hyperparameter tuning within the search space. They also employ various algorithms to explore and optimize the possibilities within this search space, with Bayesian Optimization variations being notably prevalent (Li, Zhi, et al., 2019).

The utilization of TPOT also opens doors to cutting-edge methodologies such as pruned decision trees, Gradient Boosting, and even tailor-made Deep Learning algorithms.

Additionally, it provides access to advanced encoding and feature preprocessing techniques. Every machine learning system comes with hyperparameters, and AutoML's fundamental role is to automatically fine-tune these hyperparameters to optimize performance. This capacity significantly reduces the manual effort required for implementing machine learning, a particularly noteworthy benefit in the realm of AutoML.

TPOT has the ability to enhance the effectiveness of machine learning algorithms by tailoring them to the specific problem at hand. This capability has led to the generation of new state-of-the-art results in various machine learning benchmarks across numerous studies as noted by Olson and Moore (2016). Moreover, it contributes to the enhancement of scientific experiment fairness and reproducibility. Automated TPOT offers a clear advantage in terms of reproducibility compared to human-based search methods. This is vital because to fairly compare multiple approaches, they need to be consistently tuned to the same degree for a given problem, simplifying meaningful comparisons. The architecture of the framework utilized in our work is outlined in Figure 8.4.

Digital Intelligent Forensic Framework (DIF^2) integrates with this automation feature to achieve best model selection and 'Pipeline Optimisation' using TPOT and AutoML. This iterative approach is expected to produce significant results for predictive modelling. These algorithms can be employed to facilitate transfer learning, merge data frames to enhance modelling, and create more comprehensive 'feature lists' by leveraging combined datasets.

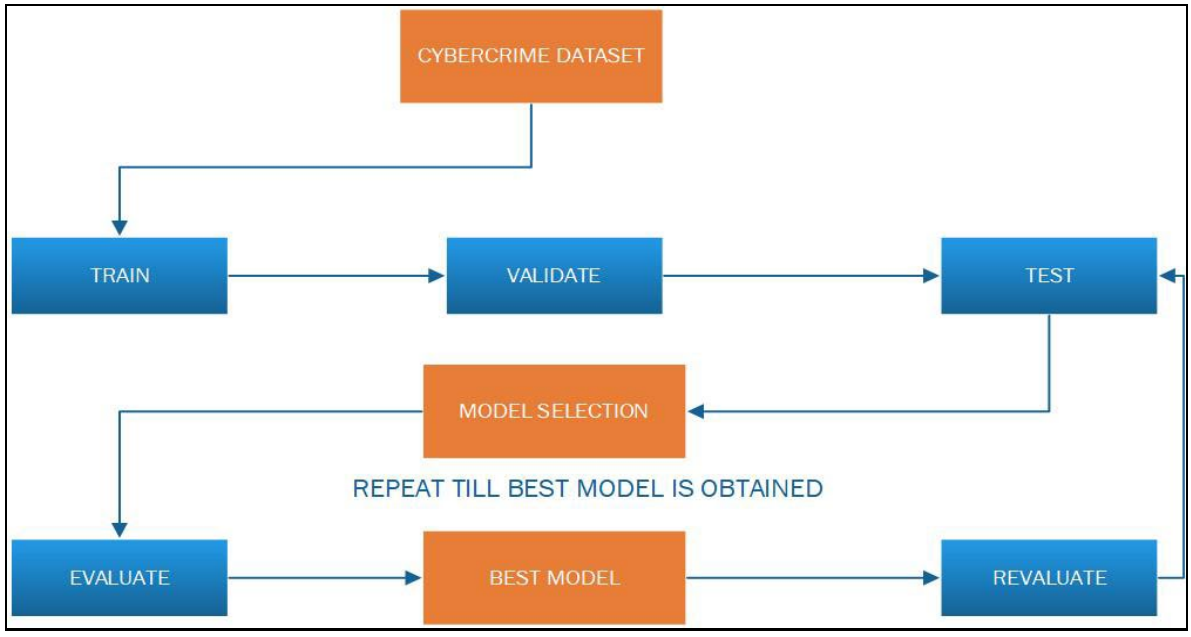


Figure 8.4: Best Model Selection and Pipeline Optimisation using TPOT and AutoML in DIF^2

8.4 Implementation Specifications

It is expected that implementing TPOT-AutoML based architecture will prove effective for Intelligent Forensics (IF) analysis and predictive analytics. By integrating this automation alongside a user-friendly interface designed in accordance with best Human-Computer Interaction (HCI) practices tailored to diverse use cases within newly developed prototype, we aim to provide digital forensic investigators with an efficient mechanism for automating tasks and making informed decisions through predictive modelling.

For the purpose of distributed data mining and forensic analysis, catering to agencies and investigators at all levels of hierarchy, including policy makers and field agents, an AI based application built upon a newly designed ontology and framework description holds good potential. While other frameworks like TensorFlow also support distributed and cloud computing applications and are backed by the AutoML computation pipeline as emphasised by Ono et al., (2021), it's important to note that achieving robust functionality in this regard typically requires substantial engineering effort and isn't readily available "out of the box".

All the capabilities offered by TPOT are accessible through JSON via HTTP, making them available for external programs or scripts through the REST-ful API for TPOT. The additional parts of the API utilize the TPOT web interface (Flow UI), R binding (TPOT-R), and Python binding (TPOT-Python).

In this particular implementation, ‘TPOT-Python’ libraries are employed to accomplish the selection of the best-performing model. Furthermore, these libraries are utilized for further optimizing and scoring pipelines, making it a versatile choice for model selection and refinement.

8.4.1 Dataset Overview

Dataset contains cybercrime cases recorded under variables defined as cybercrime ‘type’ in three major cities in South Africa. Therefore dataset consists of five columns identifying cybercrime type ‘Phishing, ZeroDay Attack, Ransomware Attack, Child Pornography’ and the ‘City’ represented as features in the prediction modelling.

Cybercrime DataSet Repository in Cloud

Dataset URL:

url = "https://raw.githubusercontent.com/deep0025/CyberCrimeData-SA/main/CybercrimeDataset.csv"

8.4.2 Experiment Pipeline

Figure 8.5 depicts the ontology of the experimental pipeline of Digital Intelligent Forensic Framework (DIF^2) automation and its implementation. In this implementation, we harnessed machine learning techniques employing TPOT-Python and AutoML to attain optimal AI model selection and scoring pipeline optimization. To facilitate performance comparison, two versions of datasets containing 50 and 100 tuples were employed. The framework effectively conducted pipeline tuning and model selection, ultimately delivering a scoring pipeline that highlights prediction accuracy scores.

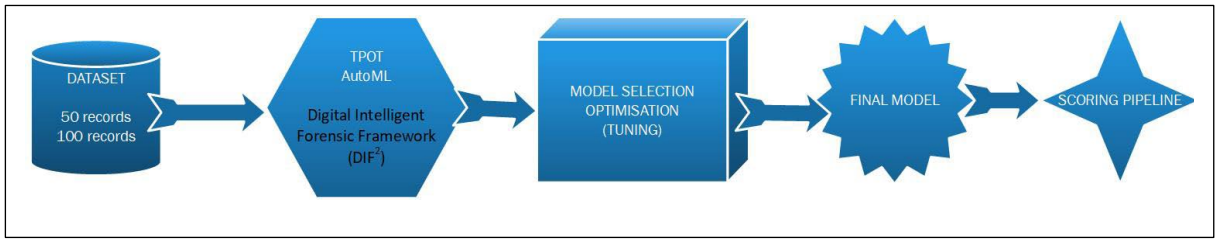


Figure 8.5: Experimental Pipeline Ontology for DIF^2 Framework using TPOT and AutoML

The experimental ontology involved in this pipeline are implemented in experiment-2 and described in more detail in section-5.7.2 and this section.

Data Ingestion: In the initial phase, ingesting the data is achieved, during which the types of columns (features) present within the dataset are separated.

Model and Feature Tuning: The subsequent stage involves a dynamic interplay of random hyperparameter tuning, feature selection, and feature generation. In each iterative cycle, features are enhanced based on their significance, as determined probabilistically from the preceding iteration. This process guides the creation of new features. Ultimately, the best-performing model and features are advanced to the next phase: feature evolution.

Model Parameter Tuning: Through the extensive training of models with varying sets of variables, identification of the optimal parameters for key model types, such as constants and decision trees is achieved (Feurer et al., 2015).

Model Selection: In the pursuit of pinpointing the most effective combination of model parameters and feature transformations for the final model, a genetic algorithm is employed. This stage systematically refines the model's configuration, ensuring that it represents the data in the most advantageous way.

Optimal Feature Representation: The quest for the finest data representation for the ultimate model culminates in the creation and assessment of two distinct features over ten iterative iterations. This thorough approach guarantees that we arrive at the most optimal feature representation.

Final Model Creation: Finally, armed with the insights gained from the feature engineering iterations and the optimized feature representation, the best-performing model is selected, ensuring that it encapsulates the refined characteristics and parameters honed through this comprehensive process.

Scoring Pipeline Generation: Following the extensive experimental implementation, a scoring pipeline is successfully crafted and exported. This pipeline serves as a powerful tool for seamlessly and efficiently evaluating the performance of AI models. Figure- 8.5 explains the ontology of this experimental setup whereas figure-8.7 represents actual implementation in Python language.

AutoML-Powered Model Training: Throughout the course of the experimental implementation, AutoML played a pivotal role. It tirelessly trained numerous models, meticulously exploring various parameters, dataset configurations, and model architectures. The objective was to identify the most advantageous combinations of these factors to arrive at the optimized final model.

$$\{X_{train}, Y_{train}, X_{Test}\} = \textbf{TPOT + AutoML \{Data Preprocessor, Feature Preprocessor, Classifier\}} = Y_{Test} \quad \text{Eq. (8.1)}$$

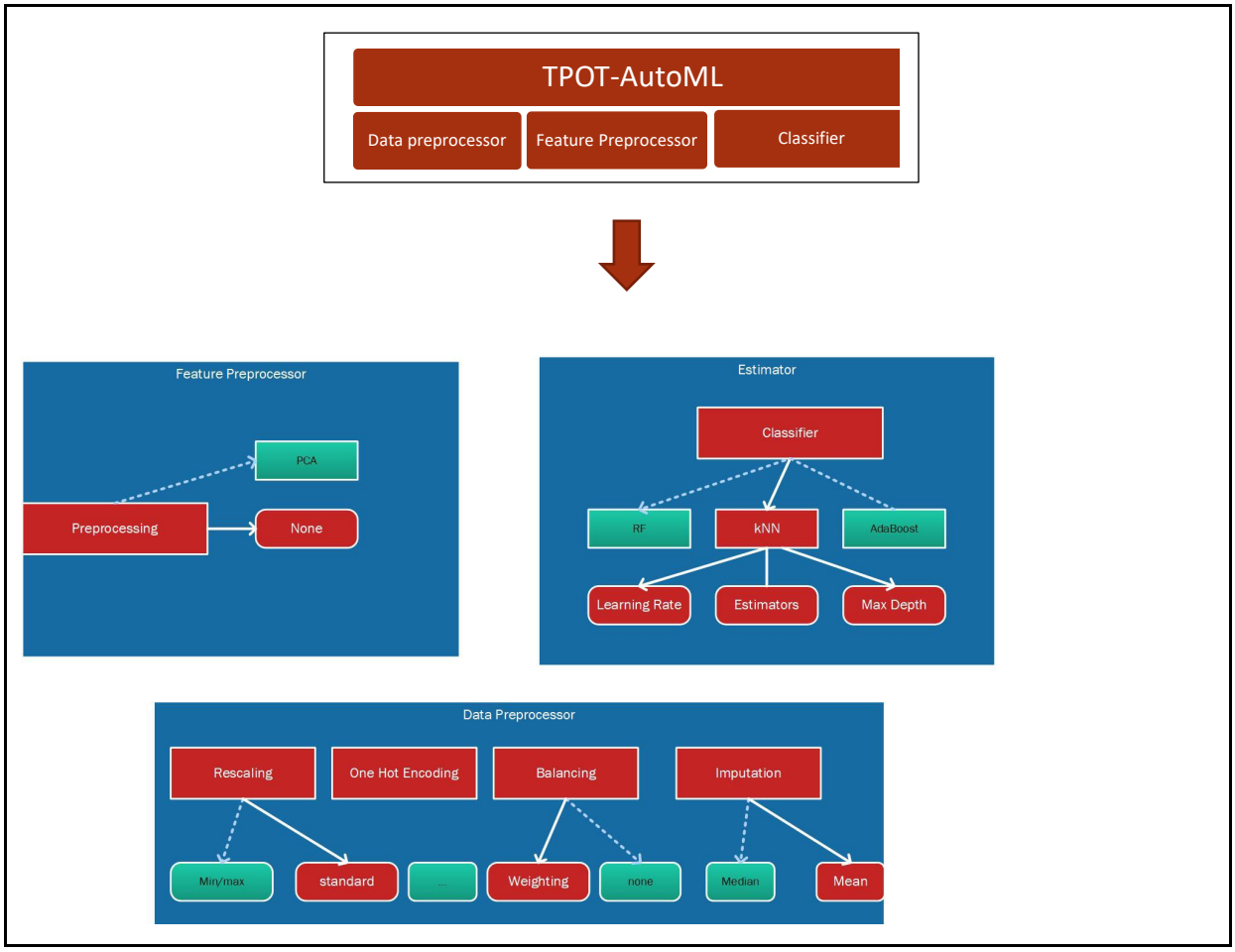


Figure 8.6: Ontological Structure Automation Framework for DIF^2 using AutoML

Ontological Structure of AutoML Automation Framework for DIF^2 is illustrated in figure 8.6. This framework represents ontological component structure that is implemented in DIF^2 using AutoML. Here squared boxes represent 'parent' hyperparameters, whereas boxes with round edges represent 'leaf' hyperparameters. The red boxes represent active hyperparameters in machine learning pipeline in AutoML modelling. Each pipeline consists of one feature "pre-processor", classifier and up to three "data preprocessors". Eq. (8.1) represents the formulation of the model implemented for automation of DIF^2.

8.5 Supported Algorithms by TPOT and AutoML

TPOT primarily leverages a genetic algorithm for its optimization procedures. It employs genetic programming algorithm, which is tailored for stochastic global optimization on programs represented as trees, to identify the most effective tree structures for a given dataset. In the context of the AutoML Framework, TPOT offers support for a diverse array of machine learning algorithms.

TPOT supports a long list of AI algorithms such as Decision Trees, Random Forest, Gradient Boosting (e.g., XGBoost, LightGBM), k-Nearest Neighbors (k-NN), Support Vector Machines (SVM), Linear Regression, Logistic Regression, Naive Bayes, Neural Networks (Deep Learning) and Principal Component Analysis (PCA).

These algorithms, among others, are part of TPOT's toolkit, enabling it to explore and optimize various machine learning techniques to discover the best model for a given dataset through its genetic algorithm-based optimization process. Some of them are described here while others have been mentioned in other sections already.

LightGBM: LightGBM, developed by Microsoft, is a gradient boosting framework renowned for its efficient utilization of tree-based learning techniques. It distinguishes itself by its ability to operate with minimal memory consumption, superior performance, and swift training capabilities. LightGBM stands as one of the premier grad ient boosting implementations, often considered on par with XGBoost. In the realm of Driverless AI, LightGBM is also applied for fitting techniques from Random Forest, DART (an experimental algorithm), and Decision Trees.

XGBoost: XGBoost is a supervised learning model that harnesses the boosting technique to construct accurate predictive models. Boosting, in this context, refers to an ensemble learning strategy that sequentially assembles multiple models, with each new iteration designed to rectify the shortcomings of its predecessor (Wen et al., 2020). In the domain of tree boosting, each fresh addition to the ensemble is a decision tree. XGBoost, often referred to as GBDT (Gradient Boosting Decision Trees) or GBM (Gradient Boosting Machines), provides parallel tree boosting, making it a go-to choice for a wide array of data science challenges.

This model stands as one of the foremost implementations in the realm of gradient boosting machines (GBMs). Driverless AI implements the DART (Dropouts meet Multiple Additive Regression Trees) techniques of XGBoost GBM and XGBoost as part of its repertoire of algorithms. These DART techniques enhance the capabilities of XGBoost by incorporating dropout regularization, contributing to more robust and effective model training within the Driverless AI framework.

A Decision Tree (DT) is a binary tree-based machine learning model that constructs its tree structure by recursively partitioning the training data into subsets. Unlike some other algorithms, Decision Trees typically do not involve random row or column sampling during

their construction. Instead, they make decisions about how to split the data based on the features and values that best differentiate the samples.

Key characteristics of Decision Trees include:

Recursive Splitting: Decision Trees start with a single root node representing the entire dataset and then recursively split it into subsets by evaluating different features and their thresholds. These splits continue until certain criteria are met, such as a maximum tree depth or a minimum number of samples per leaf.

Leaf Nodes: The final nodes of a Decision Tree are called leaf nodes or terminal nodes. Each leaf node corresponds to a subgroup of the data and represents a predicted outcome or class label.

Hyperparameters: Decision Trees are controlled by hyperparameters that determine their structure and behaviour. Two essential hyperparameters are:

Tree Depth: This parameter sets a limit on how deep the tree can grow. Deeper trees can capture more complex patterns but may also overfit the data.

Growth Strategy: Decision Trees can be grown in a depth-wise (splitting nodes by fixed depth levels) or a loss-guided (splitting nodes based on minimizing impurity or error) manner.

8.6 Automation of DIF² Framework using TPOT and AutoML

Leveraging the TPOT library within an AutoML framework allows for the development of optimized AI model pipelines. This entails the creation of an instance of either the TPOTRegressor or TPOTClassifier class, the customization of its settings to facilitate the search process, and ultimately, the exportation of the best-performing model pipeline discovered for data analysis.

The configuration of the class revolves around two primary components. The initial component pertains to the manner in which models are assessed, including elements such as the cross-validation scheme and the choice of performance metric. This entails the explicit specification of a cross-validation class with a tailored configuration, as well as the selection of an appropriate performance metric.

The implemented code example uses “RepeatedKFold” with “*neg_mean_absolute_error*” metric for regression. Alternatively, “RepeatedStratifiedKFold” for regression with “*accuracy*” metric for classification can also be used. Next step involves defining the nature of the stochastic global search procedure.

Being an evolutionary algorithm, this process entails configuring specific parameters, including the population size, the number of generations to execute, and, optionally, the crossover and mutation rates. The population size plays a crucial role in dictating the scope of the search, while the default values for crossover and mutation rates can typically be retained, especially for evolutionary search. In this particular instance, a population size of 50 and a run of 5 generations have been selected as the hyperparameter configuration.

Upon the completion of the search process, the algorithm identifies an optimal pipeline for achieving the highest performance. This pipeline can be exported as code into a Python file, allowing for customization and seamless implementation. To specify the model evaluation approach, a repeated ‘stratified k-fold’ cross-validation with three repetitions and ten folds is employed. The search is executed with hyperparameter setting of five generations, utilizing all available CPU cores by setting "n_jobs" to ‘-1’. Ultimately, the search is initiated, culminating in the preservation of the top-performing model at the conclusion of the run.

8.6.1 Optimisation of the AI Framework (DIF^2) using Pipeline

Figure-8.7 shows the actual code implementation of program (in Python) for using TPOT library and AutoML in order to obtain an optimised pipeline, while code snippet-8.1 represents the Python code implementation program. The example uses SKLearn Machine Learning libraries for general implementation. Pipelines provides effective mechanisms to prevent data leakage and produce more accurate predictive analysis.

```python
from pandas import read_csv
from sklearn.preprocessing import LabelEncoder
from sklearn.model_selection import RepeatedStratifiedKFold
from tpot import TPOTClassifier
# load dataset
url = 'https://raw.githubusercontent.com/deep0025/CyberCrimeData-SA/main/cybercrime-sample-dataset
dataframe = read_csv(url, header=None)
# split into input and output elements
data = dataframe.values
X, y = data[:, :-1], data[:, -1]
# minimally prepare dataset
X = X.astype('float32')
y = LabelEncoder().fit_transform(y.astype('str'))
# define model evaluation
cv = RepeatedStratifiedKFold(n_splits=10, n_repeats=3, random_state=1)
# define search
model = TPOTClassifier(generations=5, population_size=50, cv=cv, scoring='accuracy', verbosity=2,
# perform the search
model.fit(X, y)
# export the best model
model.export('tpot_cybercrime_data100_best_model.py')
```

```
C:\Users\kumard\Anaconda3\lib\site-packages\xgboost\compat.py:36: FutureWarning: pandas.Int64Index is
deprecated and will be removed from pandas in a future version. Use pandas.Index with the appropriate
dtype instead.
  from pandas import MultiIndex, Int64Index
```

Optimization Progress: 20% 60/300 [02:04<08:10, 2.04s/pipeline]

Code Snippet 8.1: Optimisation of the AI Framework (DIF^2) Pipeline using TPOT and AutoML

Python Code Snippet 8.1 shows the implementation and progress of optimisation process. As the model optimises itself completely, five generation of search and their optimisation score is shown in figures-8.7 and 8.8. The results are discussed in section 9.3.2 in chapter-9.

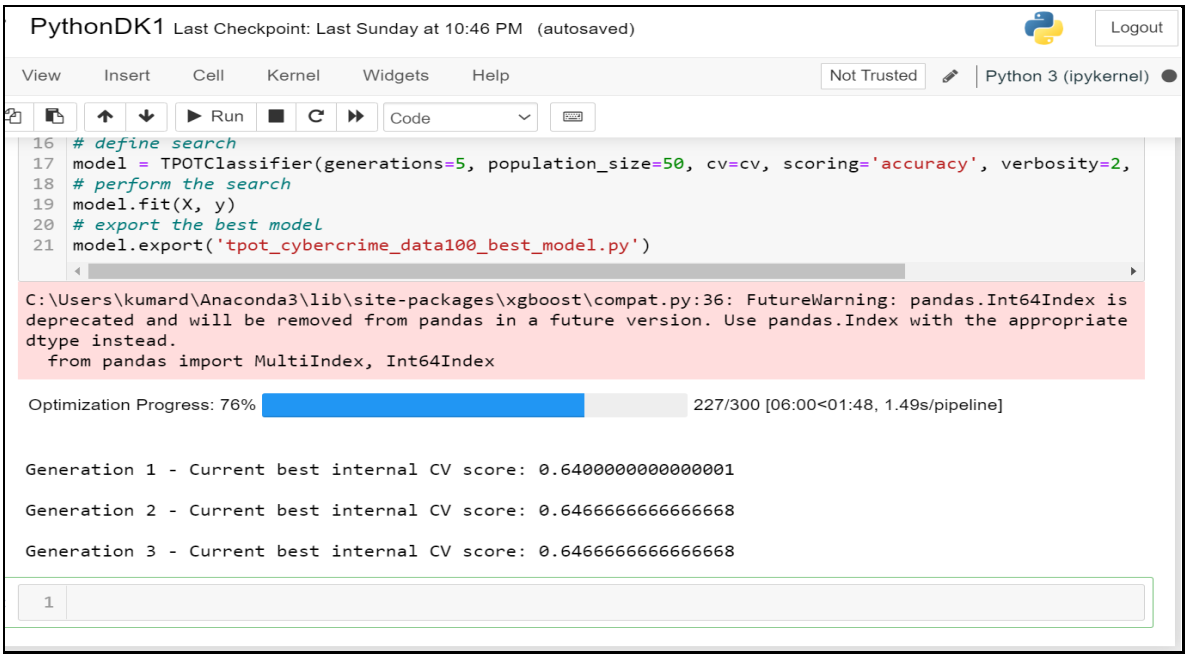


Figure 8.7: Optimisation pipelines progress showing three generation CV scores

This process takes time and the hyperparameters are used to set the number of generations for which the optimisation process will continue.

```
Generation 1 - Current best internal CV score: 0.6400000000000001

Generation 2 - Current best internal CV score: 0.6466666666666668

Generation 3 - Current best internal CV score: 0.6466666666666668

Generation 4 - Current best internal CV score: 0.6466666666666668

Generation 5 - Current best internal CV score: 0.6466666666666668

Best pipeline: KNeighborsClassifier(BernoulliNB(input_matrix, alpha=10.0, fit_prior=True), n_neighbor
s=23, p=1, weights=uniform)
```

Figure 8.8: Optimisation Pipeline Scores of DIF^2 Automation using AutoML and TPOT

8.6.2 Critical Analysis of Optimisation of DIF^2 using TPOT and AutoML

This research experiment has demonstrated the 'Automation' and 'Optimisation' of DIF^2 intelligent framework using AutoML and TPOT and also observed various benefits as discussed here and results evaluated further in section-9.3.2 (experiment-2). When employing TPOT, we gain access to state-of-the-art techniques, including pruned decision trees, Gradient Boosting, and even tailored 'Deep Learning' algorithms. Additionally, TPOT provides a range of advanced encoding and feature preprocessing methods. Every machine learning system involves hyperparameters, and the primary objective of AutoML is to automatically configure these hyperparameters to optimize performance. This capability greatly reduces the level of human effort needed for machine learning implementation, a particularly important aspect of AutoML.

8.6.3 Criticism of TPOT and AutoML Automation Framework

Although many benefits can be achieved from using automation framework like TPOT and AutoML, especially in terms of reducing human efforts and auto selection of best performing algorithms. Also, this automation framework can be very effective in predictive modelling of a range of datasets and support quick decision making which is one of the objectives we are trying to achieve in this research. However, it was observed that the implementation of AutoML and TPOT is complex and need excessive resources. 'Pipeline Optimisation' process can be very 'time consuming' and creates massive overheads if the model needs to be implemented into the prototype to be used by first responders. While the dataset size also leads to improved accuracy but limited computing resources can be difficult and detrimental to the

implementation of this technique in DIF^2 framework domain and integration in the prototype. An output of five generation pipeline in shown in figure 8.8. Although, this model can be ideal in implementation of Forensic-As-a-Service (FAAS) domain.

Although a good score in terms of efficacy of optimisation of **64.66%** is achieved by the best performing model KNeighbors Classifier (KNN) but only slight improvement in the output score from Generation-1 to Generation-5 is observed. This can be attributed to the fact that we have a relatively small dataset whereas the training dataset needs to be substantive.

The primary drawback identified in the implementation is its resource-intensive nature. This characteristic stands in contrast to the objectives of this research to develop an intelligent (AI based) framework that is easily implementable using a prototype which ideally should be fast, lightweight, and agile.

To address this drawback, it may be beneficial to explore other models, optimization techniques and alternative approaches in the implementation. Consideration should be given to streamlining the code, optimizing algorithms, and adopting more efficient data processing methods. This way, the intelligent framework and its prototype implementation can better align with its intended characteristics of speed, lightness, and agility, ensuring a smoother and more responsive user experience.

8.7 Need for developing a Customised Algorithm from Scratch (ANN+SGD Ensemble)

Since we have demonstrated under section-8.6 that 'Automation' of DIF^2 framework can be achieved using AutoML and TPOT implementation that yielded encouraging results in terms of predictive modelling. The automation is beneficial as it can cater to different use cases and provide a comprehensive framework to predictive analysis to major problems. Although, this automation has its own benefits like applicability to a diverse problems and datasets, automatic hyperparameter tuning and model selections etc., but is quite resource intensive. Main drawback is that the implementation is complex and require extensive resources. This may not fit well with research objectives of prototype implementation, which need to be fast, light-weight and agile. Another major requirement is that the dataset needs to be quite large enough to produce high accuracy rate using automation techniques like AutoML and TPOT. However, we see a scope and need to obtain better performance and contemplate that still it is possible to achieve high accuracy on relatively smaller datasets using AI techniques like Artificial Neural Networks (ANN) and performing predictive modelling using Stochastic Gradient Descent as

demonstrated by Brownlee (2021) in his machine learning experiments and also emphasised by Mijwil (2018) in their research.

8.7.1 Motivation for Creating Customised Algorithm using ANN and SGD

Artificial Neural Networks (ANNs) offer numerous advantages over alternative AI techniques, making them a favoured and powerful choice for a wide array of applications. These advantages include their capability to differentiate intricate patterns within data tasks like data classification. They are effective in 'linear and non-linear' predictive modelling. ANNs are characterized by their flexibility and adaptability, making them suitable for various tasks with minimal adjustments to their architectural design. They possess the capacity to autonomously extract relevant features from raw data, thus eliminating the laborious process of manual feature engineering.

ANNs' remarkable generalization ability enables them to make accurate predictions on new, unseen data, contributing to enhanced real-world performance. Furthermore, ANNs are proficient in modelling complex, nonlinear relationships within data, a crucial attribute for addressing problems that defy linear assumptions. Their scalability is evident as they can be expanded by adding more layers or neurons to tackle increasingly intricate tasks and accommodate vast datasets (Mohammad, 2018).

ANNs support continuous learning, allowing them to adapt to evolving data distributions over time, which proves invaluable for applications like fraud and anomaly detection. Additionally, ANNs demonstrate robustness in the face of noisy data, effectively filtering out irrelevant information, which is advantageous for tasks with imperfect input (Mijwil, 2018).

Stochastic Gradient Descent

At the heart of numerous machine learning algorithms lies the concept of optimization. Machine learning algorithms employ optimization techniques to discover an optimal set of model parameters based on a provided training dataset. Among the various optimization algorithms employed in the field, Stochastic Gradient Descent stands out as the most prevalent and widely used method (Singh, 2022). This algorithm is discussed in detail under section-7.6.1 Multivariate Linear Regression

Creating a new algorithm from scratch that is specifically developed considering the problem at hand can be an effective means to achieve desired efficacy in prediction modelling. This new

algorithm is based on ‘Ensemble Method’ that employs Artificial Neural Network (ANN) and Stochastic Gradient Descent (SGD) algorithms combination to achieve low error rate (RMSE) and higher accuracy. Therefore, we endeavour to design a new algorithm from scratch using ANN and SGD ensemble model, while also incorporating ‘triaging’ algorithm (Calculate_Triage_Score) developed and discussed in section-7.9, in order to compare the performance of various AI interventions and improve the prediction and decision-making.

8.7.2 Artificial Neural Networks Backpropagation Method

The Backpropagation algorithm is a supervised learning technique applied to multilayer feedforward networks within the domain of Artificial Neural Networks. These neural networks draw inspiration from the information processing capabilities of individual neural cells, known as neurons. A neuron receives input signals through its dendrites, transmitting these electrical signals to the cell body. The axon, in turn, carries the signal to synapses, which are connections between one cell's axon and another cell's dendrites.

The fundamental concept behind the backpropagation approach is to model a given function by adjusting the internal weightings of input signals to generate an anticipated output signal. This system undergoes training using supervised learning, where the error between the system's output and a predetermined expected output is introduced to the system. This error information is then utilized to modify the system's internal configuration.

Technically speaking, the backpropagation algorithm serves as a means to train the weights within a multilayer feedforward neural network. This necessitates defining a network structure comprising one or more layers, with each layer being fully interconnected to the next. A common network structure consists of an input layer, a hidden layer, and an output layer. Backpropagation can be employed for both classification and regression tasks (Brownlee, 2016).

In classification tasks, optimal performance is typically attained when the neural network features one neuron in the output layer corresponding to each unique class value. For instance, in a binary classification scenario with class labels A and B, the expected outputs need to be converted into binary vectors with a single column dedicated to each class value. This transformation is known as "one-hot encoding", where class A is represented as [1, 0] and class B as [0, 1], ensuring that each class is uniquely identifiable in the output layer. We will use ‘Backpropogation’ method to create and implement a ‘new algorithm’. This new algorithm

implementation will involve 6 steps as (i) Initialise Neural Network; (ii) Forward Propagate; (iii) Backpropagate Error; (iv) Train Neural Network; (v) Make Prediction using 'Triaging'.

Initialise Neural Network

This step involves creating a new network. Each neuron in Neural Network consists of a set of weights that it tries to maintain. There is one weight for each input and an additional weight is designated for the bias. During training, additional properties also need to be preserved therefore a collection object (e.g. Dictionary) can be used and properties be stored as "weights". Normally, a network consists of layers. Each row from the training dataset forms an "input layer". The next layer is the "hidden layer", which is followed by an "output layer" where each class value is stored in a separate neuron. Layers can be organised as an array of dictionaries and the whole network is represented as an array of layers. Initially the weights can be initialised to a small random number in the range of 0 and 1.

This process can be achieved by creating a function called 'Initialise Network' that accepts parameters such as the number of inputs, the number of neurons in the hidden layer, and the number of outputs.

'n_hidden' neurons store hidden layer and each neuron in the hidden layer has weight of 'n_neuron + 1' for each column in dataset and an additional one for bias. The output layer that connects to the hidden layer has 'n_output' neurons with a weight of 'n_hidden +1'. Therefore, each neuron in output layer connects to each neuron in hidden layer and carries weight for this neuron.

Forward Propagation

To obtain an output from a neural network, we perform a sequential transmission of an input signal through each layer until the final output layer produces its values. This process is referred to as "forward-propagation." It serves as the fundamental technique required for generating predictions during training, which subsequently undergo correction, and it remains essential for making predictions on new data once the network is trained. The forward-propagation process can be divided into three distinct phases: (i) Neuron Activation; (ii) Neuron Transfer; (iii) Forward Propagation.

Neuron Activation

The initial phase involves computing the activation of a single neuron when provided with an input. This input could either be a row extracted from our training dataset, as is the case for the hidden layer, or it may consist of the outputs generated by each neuron within the hidden layer when dealing with the output layer. Neuron activation is determined by calculating the weighted sum of its inputs, resembling the process in linear regression.

Equation: $\textbf{activation} = \textbf{bias} + \sum_{i=1}^{n} \textbf{weight}_i \times \textbf{input}_i$ Eq. (8. 2)

In this equation, "weight" pertains to a weight within the neural network, "input" represents an input value supplied to the neuron, "i" stands for the index associated with either a weight or an input, and "bias" serves as a unique weight devoid of an input for multiplication or the input as a constant value of 1.0. Eq. (8.2) represents the formula used in calculating activation.

Neuron Transfer

After a neuron is triggered into activation, the next step involves transferring this activation to determine the actual output of the neuron. Various transfer functions are available for this purpose. Traditionally, the sigmoid activation function has been the popular choice, but alternatively, the hyperbolic tangent (Tanh) function can be employed to transfer the outputs. More recently, the rectifier transfer function has gained popularity, particularly within extensive deep learning networks.

The sigmoid activation function exhibits an S-shaped curve and is also referred to as the logistic function. It is capable of taking any input and generating a value ranging between 0 and 1 along the curve. Importantly, it is a function for which the derivative (slope) can be computed easily, which is a critical aspect for subsequent error backpropagation.

Equation for sigmoid activation function: $\boldsymbol{output} = \frac{1}{1+e^{-activation}}$ Eq. (8. 3)

Eq. (8.3) represents the sigmoid activation function, where ‘e’ represents the base of natural logarithms also known as “Euler’s number”.

Forward Propagation

Forward-propagating an input is a straightforward process. The function systematically traverses each layer of the network, computing the outputs for each neuron. The outputs generated by one layer serve as the inputs for the neurons in the subsequent layer.

Backpropagation of Error

The term "backpropagation" in the artificial neural networks pertains to the manner in which weights are adjusted during training. The process begins by calculating the error, which quantifies the disparity between the expected outputs and the outputs obtained from forward-propagating the input through the network. These error values are then retrospectively transmitted backward through the network, starting from the output layer and moving towards the hidden layers. During this backward propagation, responsibility for the error is attributed, and weight adjustments are made accordingly. The underlying mathematics of error backpropagation are deeply rooted in calculus, but in this section, we will maintain a high-level perspective, focusing on process and results, rather than delving into the intricacies of why these specific calculations are employed. This stage can be further divided into two parts (i) Transfer Derivative; (ii) Error Backpropagation;

Transfer Derivative

A slope is calculated from the output value of a neuron. Here a sigmoid transfer function is being used to calculate 'Transfer Derivative' as shown in eq. (8.4).

Equation: $\boldsymbol{derivative = output\ x\ (1.0 - output)}$ Eq. (8.4)

Error Backpropagation

This step involves calculation of error for each output neuron. This will provide a signal for error to propagate backwards through the network. Error calculation is calculated by the formula given in eq. (8.5).

Equation: $\boldsymbol{error = (expected - output)\ x\ transfer\_derivative\ (output)}$ Eq. (8. 5)

In this context, "expected" represents the anticipated output value for a neuron, "output" corresponds to the actual neuron output, and "transfer derivative ()" computes the gradient (slope) of the neuron's output, as previously described. This error calculation is employed

specifically for neurons within the output layer, with the "expected value" being synonymous with the class value itself.

However, when it comes to neurons in the hidden layer, the process becomes more intricate. The error signal for a hidden layer neuron is computed by considering the weighted errors of every neuron in the output layer. This error signal can be imagined as traveling in reverse along the weight connections from the output layer to the neurons in the hidden layer. This accumulated, “back-propagated” error signal is then utilized to determine the error associated with each neuron in the hidden layer using formula shown in eq. (8.6).

Equation: $\boldsymbol{error = \left(weight_k \, x \, error_j\right) x \, transfer\_derivative \, (output)}$ Eq. (8.6)

In eq. (8.6), $error_j$ represents error signal from j[th] neuron in the output layer and $weight_k$ connects k[th] neuron to the current neuron, whereas $output$ is the output for the current neuron. The error signal calculated for each neuron is stored as “delta”. The iteration through the layers occurs in the reverse order i.e. starting from output and going backwards. Therefore the “delta values” for the output layer neurons are calculated first and hidden layer neurons can use it for successive iterations (Brownlee, 2016). In the newly created algorithm, the “delta” represents the “weight delta” or the change in neuron after error calculation. The “hidden layer” neuron error-signal is calculated from “outer layer” neurons.

Training of Network

The network training is done using ‘Stochastic Gradient Descent” as discussed in earlier section. The process mainly entails multiple iterations so that training dataset is exposed to network. Each row of data does “forward-propagating” of the inputs, “back-propagating” the error and thus updating the “weights” of the network. This process can also be elucidated by breaking into two steps: (i) Update Weights (ii) Train Network.

Update Weights

After error calculation is done for each neuron in the network using “backpropagation”, the weights are updated using formula shown in eq. (8.7).

$\boldsymbol{Equation: weight = weight_{bias} + learning_{rate} \, x \, error \, x \, input}$ Eq. (8. 7)

Learning rate in the formula is specified and ‘error’ is calculated using “backpropagation” function, while “input” is the input value that resulted into causing the error. The same process

is used to update the “bias” weight as there is no input or input is fixed at 1.0 value as shown in eq. (8.7).

The learning rate serves as a control mechanism dictating the extent of weight adjustments made to rectify errors within a neural network. To illustrate, consider a learning rate value of 0.1; this would result in weight updates that are 10% of the maximum possible adjustment that could be applied.

In practice, smaller learning rates are often favoured. They induce a gradual learning process that spans a considerable number of training iterations. This approach increases the probability of the network discovering an optimal set of weights across all layers, rather than hastily converging towards a set of weights that minimizes error at the expense of thorough exploration. This phenomenon is referred to as "premature convergence", where the network settles for a suboptimal solution due to overly aggressive weight updates.

Train Network

The network is trained and updated using “Stochastic Gradient Descent” as discussed previously under section-7.10.2 (Multivariate Linear Regression). This process involves iterating over a fixed number of epochs, and within each epoch, updating the network for each individual row in the training dataset. This type of learning is referred to as "online learning" because updates are made for each training pattern in a sequential manner. In contrast, if errors were accumulated across an entire epoch before updating the weights, it would be termed "Batch Learning" or "Batch Gradient Descent".

The newly developed algorithm implements the training of a pre-initialized neural network with a given training dataset, learning rate, a fixed number of epochs, and the expected number of output values. The expected number of output values is utilized to convert class values in the training data into a “one-hot encoding”, which is essentially a binary vector with one column for each class value, aligning with the network's output structure. This transformation is necessary for computing the error in the output layer. Additionally, the algorithm accumulates and prints the ‘sum of squared errors’ between the expected output and the network's output during each epoch. This helps in monitoring and visualizing the network's learning progress and improvement across epochs.

Prediction

This process mainly consists of "forward-propagation" of an input pattern to produce output and make predictions.

Figure 8.9 represents a graphical model of newly developed and customised algorithm based on ANN Back propagation and SGD ensemble in order to achieve better performance for predictive modelling.

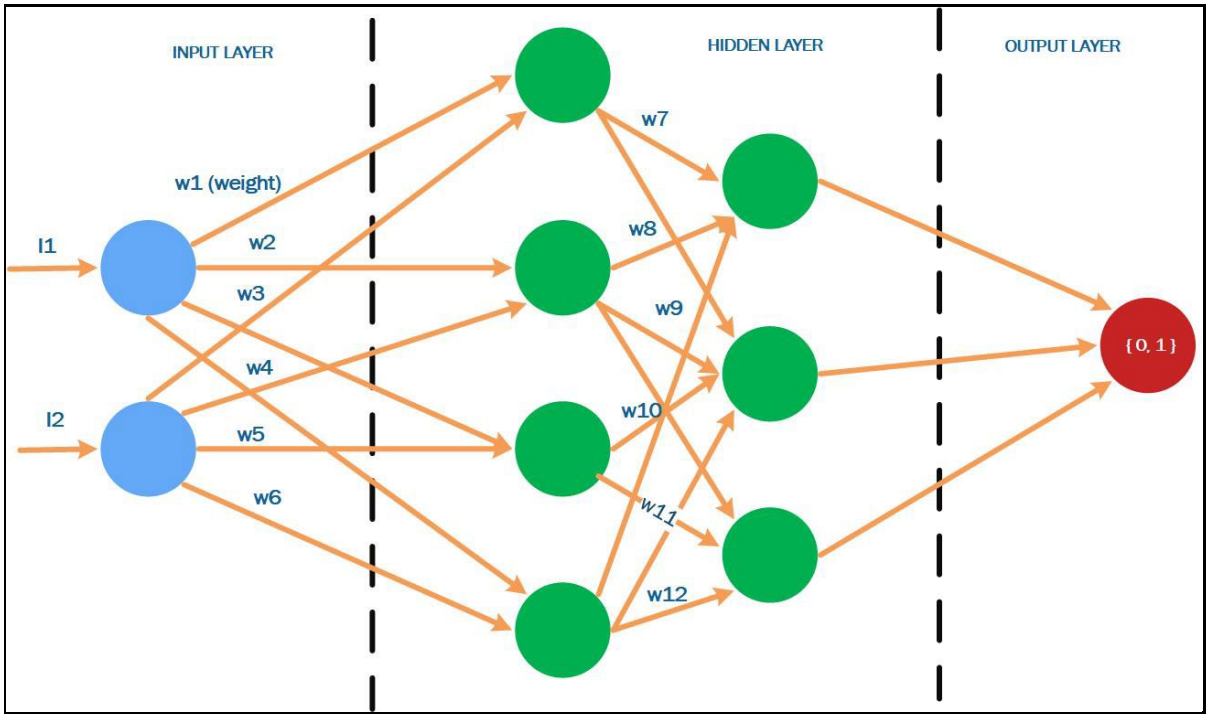


Figure 8.9: Algorithm implementation for predictive analysis using ANN and SGD ensemble

8.7.3 Algorithm Custom_ANN_Backpropagation_SGD_Ensemble

Step 1: Load Data: Load the data from a CSV file, including the input features and output classes.

Step 2: Data Preprocessing: Convert string columns to numerical values (floats and integers).

Normalize the input variables to a range of 0-1.

Step 3: Calculate TriageScore by using based on crime severity:

For each column:

Crime=column_value;

If crime= ‘Phishing’ then severity=1;

Else If crime= ‘ZeroDayAttack’ then severity=2;

Else If crime= ‘RansomwareAttack’ then severity=3;

Else crime= ‘ChildPorn’ then severity=4;

Calculate formula: $TriageScore += severity\ x\ occurance$

Add TriageScore to a new column;

If TraigeScore > 50

Then Class = Critical (C);

Else Class = Non_critical (N);

Add column_value to ‘Class’;

Step 4: Initialize the Neural Network

Create a neural network with an input layer, a hidden layer with a specified number of neurons, and an output layer.

Step 5: Train the Neural Network

Repeat for a fixed number of epochs (training cycles).

For each training epoch:

For each row in the training dataset:

Forward propagate the input through the network to compute predictions.

Calculate the error between the predictions and the expected output.

Backpropagate the error through the network to adjust weights.

Update the network's weights based on the errors and learning rate.

Step 6: Make Predictions

Use the trained network to make predictions on a test dataset.

Record the predictions for each test example.

Step 7: Evaluate Performance

Repeat the process for k-fold cross-validation (default is 5 folds).

Calculate the accuracy of predictions for each fold.

Print the accuracy scores for each fold.

Print the mean accuracy score across all folds.

Step 8: Output Results

Print the accuracy scores for each fold.

Print the mean accuracy across all folds as a final performance measure.

Algorithm 8.1: Algorithm_Custom_ANN_Backpropagation_SGD_Ensemble

It is also important to note that that calculation of 'TriageScore' is an important step towards implementation of classification of severity level as 'Critical' or 'Non-critical' to achieve predictive modelling and decision-making capabilities.

An aggregate function for calculation of "Triage Score" is presented in eq. (8.8). This "Triage Score" is calculated from "Crime Severity Score" and "occurrence" of crime incidents form the basis of Classification as 'Critical' or 'Non-critical'. If 'TriageScore' is less than '50', then the status is 'non-critical' but if it is greater than 50, then it is flagged as 'critical'. The dataset represents critical as '1' and 'non-critical' as '0' (binary classification) that algorithm treats as a 'classification' problem.

Equation: $\boldsymbol{Triage_{Score} = \sum_1^n severity_{score} X\ occurance}$ Eq. (8. 8)

This algorithm (8.1) has been implemented as separate program while implementing the procedure in prototype, whereby the 'TriageScore' is calculated based on 'Crime Severity'

Level, which is ‘1’ for Phishing, ‘2’ for ‘Zero Day Attack’, ‘3’ for ‘Ransomware Attack’ and ‘4’ (most severe) for ‘Child Pornography’ cybercrime incidents’. These scores are calculated automatically using the procedure implementation in prototype and recorded automatically into the database.

8.7.4 Pseudocode for Custom_ANN_Backpropagation_SGD_Ensemble

Procedure 8.1: Pseudocode for Customised Algorithm based on ANN and SGD Ensemble

Input: Cybercrime incidents dataset with features - (‘Phishing’, ‘ZeroDayAttack’, ‘Ransomeware’, ‘ChildPorn’); Class - (‘City’)

Output: Accuracy Scores, Mean_Accuracy

```
# Import necessary libraries and functions
- Import the 'seed' function for random number generation
- Import 'randrange' and 'random' for randomization
- Import 'reader' for CSV file handling
- Import 'exp' for mathematical calculations
# Define a function to load a CSV file
Function load_csv(filename):
    - Create an empty list called 'dataset'
    - Open the file 'filename' for reading
    - Create a CSV reader object 'csv_reader'
    - Loop through each row in 'csv_reader':
        - If the row is empty, continue to the next row
        - Otherwise, append the row to the 'dataset'
    - Return the 'dataset'
# Define a function to convert a string column to float
Function str_column_to_float(dataset, column):
    - For each row in 'dataset':
        - Convert the value in 'column' to a float and store it back
in the same column
# Define a function to convert a string column to integer
   Function str_column_to_int(dataset, column):
          - Create a list 'class_values' containing all values in
'column'
       - Create a set 'unique' to store unique values from
'class_values'
```

```
      - Create a dictionary 'lookup' to map unique values to
integers
    - For each unique value in 'unique', assign an integer index in
'lookup'
    - For each row in 'dataset', replace the value in 'column' with
its corresponding integer from 'lookup'
    - Return the 'lookup'
# Define a function to find the min and max values for each column
Function dataset_minmax(dataset):
    - For each column in 'dataset':
        - Calculate and store the minimum and maximum values in a
list
    - Return a list of lists containing the min-max values for each
column
# Define a function to calculate ‘Traige Score’ for each row in
‘dataset’
Function calculate_triage_Score(dataset)
  -For each column:
-Crime=column_value;
-If crime= ‘Phishing’ then severity=1;
-Else If crime= ‘ZeroDayAttack’ then severity=2;
-Else If crime= ‘RansomwareAttack’ then severity=3;
-Else crime= ‘ChildPorn’ then severity=4;
-TraigeScore += severity * column_value
-Add TriageScore to a new column;
     -If TraigeScore > 50
-Then Class = Critical (C);
     -Else Class = Non_critical (N);
-Add column_value to ‘Class’;
# Define a function to rescale dataset columns to the range 0-1
Function normalize_dataset(dataset, minmax):
    - For each row in 'dataset':
        - For each feature column (excluding the last one):
            - Rescale the value to the range 0-1 using the
corresponding min-max values
# Define a function to split a dataset into k folds
Function cross_validation_split(dataset, n_folds):
```

```
    - Create an empty list 'dataset_split' to store the folds
    - Create a copy of 'dataset' called 'dataset_copy'
    - Calculate the size of each fold ('fold_size')
    - Repeat the following 'n_folds' times:
        - Create an empty list 'fold' to represent one fold
        - While the size of 'fold' is less than 'fold_size':
            - Generate a random index ('index') within the range of
'dataset_copy'
            - Remove and append the row at 'index' from
'dataset_copy' to 'fold'
        - Append 'fold' to 'dataset_split'
    - Return 'dataset_split'
# Define a function to calculate accuracy percentage
Function accuracy_metric(actual, predicted):
    - Initialize a counter 'correct' to 0
    - For each element at the same index in 'actual' and
'predicted':
        - If they are equal, increment 'correct' by 1
    - Calculate the accuracy as (correct / total) * 100.0 and return
it
# Define a function to evaluate an algorithm using cross-validation
Function evaluate_algorithm(dataset, algorithm, n_folds, *args):
    - Split the 'dataset' into 'n_folds' using
'cross_validation_split' and store the folds in 'folds'
    - Create an empty list 'scores' to store accuracy scores
    - For each fold in 'folds':
        - Create a 'train_set' by combining all other folds except
the current fold
        - Create a 'test_set' by copying the current fold while
masking the output values
        - Use the provided 'algorithm' to make predictions on
'train_set' and 'test_set'
        - Extract the actual output values from the 'test_set'
        - Calculate the accuracy using 'accuracy_metric' and append
it to 'scores'
    - Return 'scores'
# Define a function to calculate neuron activation for an input
```

```
Function activate(weights, inputs):
    - Initialize 'activation' with the bias weight (the last weight
in 'weights')
    - For each input weight and input value:
        - Add the product of the weight and input value to
'activation'
    - Return 'activation'
# Define a function to transfer neuron activation using the sigmoid
function
Function transfer(activation):
    - Return 1.0 / (1.0 + exp(-activation))
# Forward propagate input to a network output
Function forward_propagate(network, row):
    Set 'inputs' to the input 'row'
    For each 'layer' in the 'network':
        Create an empty list 'new_inputs'
        For each 'neuron' in the 'layer':
            Calculate 'activation' using
'activate(neuron['weights'], inputs)'
            Set 'neuron['output']' to 'transfer(activation)'
            Append 'neuron['output']' to 'new_inputs'
        Set 'inputs' to 'new_inputs'
    Return 'inputs'
# Calculate the derivative of a neuron output
Function transfer_derivative(output):
    Return 'output * (1.0 - output)'
# Backpropagate error and store in neurons
Function backward_propagate_error(network, expected):
    For 'i' in reversed range(len(network)):
        Set 'layer' to 'network[i]'
        Create an empty list 'errors'
        If 'i' is not equal to 'len(network) - 1':
            For 'j' in range(len(layer)):
                Set 'error' to 0.0
                For 'neuron' in 'network[i + 1]':
                    Add '(neuron['weights'][j] * neuron['delta'])'
to 'error'
```

```
                Append 'error' to 'errors'
        Else:
            For 'j' in range(len(layer)):
                Set 'neuron' to 'layer[j]'
                Append '(expected[j] - neuron['output'])' to
'errors'
        For 'j' in range(len(layer)):
            Set 'neuron' to 'layer[j]'
            Set 'neuron['delta']' to 'errors[j] *
transfer_derivative(neuron['output'])'
# Update network weights with error
Function update_weights(network, row, l_rate):
    For 'i' in range(len(network)):
        Set 'inputs' to 'row[:-1]' (excluding the output column)
        If 'i' is not equal to 0:
            Set 'inputs' to '[neuron['output'] for neuron in
network[i - 1]]'
        For 'neuron' in 'network[i]':
            For 'j' in range(len(inputs)):
                Increment 'neuron['weights'][j]' by 'l_rate *
neuron['delta'] * inputs[j]'
            Increment 'neuron['weights'][-1]' by 'l_rate *
neuron['delta']'
# Train a network for a fixed number of epochs
Function train_network(network, train, l_rate, n_epoch, n_outputs):
    For each epoch in range(n_epoch):
        For each 'row' in 'train':
            Set 'outputs' to 'forward_propagate(network, row)'
            Create a list 'expected' with '0' values for each output
neuron
            Set 'expected[row[-1]]' to '1'
            Call 'backward_propagate_error(network, expected)'
            Call 'update_weights(network, row, l_rate)'
# Initialize a network
Function initialize_network(n_inputs, n_hidden, n_outputs):
    Create an empty list 'network'
    Create a hidden layer with 'n_hidden' neurons:
```

```
        For each neuron:
            Create 'weights' with 'random()' values for each input
and bias
        Append the hidden layer to 'network'
    Create an output layer with 'n_outputs' neurons:
        For each neuron:
            Create 'weights' with 'random()' values for each hidden
layer output and bias
        Append the output layer to 'network'
    Return 'network'
# Make a prediction with a network
Function predict(network, row):
    Set 'outputs' to 'forward_propagate(network, row)'
    Return the index of the output neuron with the highest output
value
# Backpropagation Algorithm with Stochastic Gradient Descent
Function back_propagation(train, test, l_rate, n_epoch, n_hidden):
    Set 'n_inputs' to the number of features in 'train[0]' minus 1
    Set 'n_outputs' to the number of unique output classes in
'train'
    Initialize the 'network' using 'initialize_network(n_inputs,
n_hidden, n_outputs)'
    Call 'train_network(network, train, l_rate, n_epoch, n_outputs)'
    Create an empty list 'predictions'
    For each 'row' in 'test':
        Set 'prediction' to 'predict(network, row)'
        Append 'prediction' to 'predictions'
    Return 'predictions'
# Test Backpropagation on the Cybercrime dataset
Set the random seed using 'seed(1)'
Load and prepare the dataset from the file 'Cybercrime_dataset.csv'
using 'load_csv(filename)'
For each feature column in the dataset, convert it to floating-point
values using 'str_column_to_float(dataset, column)'
Convert the class column to integers using
'str_column_to_int(dataset, len(dataset[0]) - 1)'
```

```
Normalize the input variables to the range 0-1 using
'dataset_minmax(dataset)' and 'normalize_dataset(dataset, minmax)'
Set 'n_folds' to the number of desired cross-validation folds (e.g.,
5)
Set 'l_rate' to the learning rate for training
Set 'n_epoch' to the number of training epochs
Set 'n_hidden' to the number of neurons in the hidden layer
Evaluate the algorithm using 'evaluate_algorithm(dataset,
back_propagation, n_folds, l_rate, n_epoch, n_hidden)'
Print accuracy_scores , mean_accuracy
```

Procedure 8.1 shows the pseudocode for customised algorithm developed, which is based on ANN and SGD ensemble. This algorithm employs ANN Backpropagation with Stochastic Gradient Descent to train a neural network for classification tasks and evaluates its performance on the Cybercrime dataset. This also incorporates the triaging algorithm (Calculate_Triage_Score) described in section-7.9 by calculating severity scores for cybercrime incidents and determining the severity status as 'critical' or non-critical'.

A complete and actual implementation of the algorithm and procedure (8.1) in Python and its output is demonstrated in section B.12 (figure-B.12, APPENDIX-B). The Mean Accuracy Prediction Score of **65%** was achieved on first iteration on test dataset of 50 records and an overall 97% was achieved on executing multiple iterations with a dataset of 100 records, which is a very good score for classification problems when compared with Brownlee (2016) experimental demonstrations using 'wheat' dataset. This experimental demonstration is described in section-5.7.3, labelled as experiment-3. It involved hyperparameters setting – number of folds (n_folds) = 5; learning rate = 30%; number of epoch (n_epoch) = 500 and number of hidden layers (n_hidden) = 5.

A higher percentage of learning rate is avoided as high learning rate might cause the optimization algorithm to skip over the minimum of the cost function. High learning rates can cause optimization algorithm to converge to suboptimal or poor local minima instead of the global minimum. In order to demonstrate the model performance, a lowered test configuration of n_folds = 2 and n_epoch = 10 is used and output shown in figure-8.10.

The performance results along with other hyperparameter testing configurations are discussed in upcoming chapter (9) 'Evaluation of Results' in detail, where many distinct configurations

are tested and results analysed to learn more about the model behaviour and identify most optimised hyperparameters of the model.

```
str_column_to_int(dataset, len(dataset[0])-1)
# normalize input variables
minmax = dataset_minmax(dataset)
normalize_dataset(dataset, minmax)
# evaluate algorithm
n_folds = 2
l_rate = 0.3
n_epoch = 10
n_hidden = 5
scores = evaluate_algorithm(dataset, back_propagation, n_folds, l_rate, n_epoch, n_hidden)
print('Scores: %s' % scores)
print('Mean Accuracy: %.3f%%' % (sum(scores)/float(len(scores))))
```

```
>epoch=0, lrate=0.300, error=36.532
>epoch=1, lrate=0.300, error=29.686
>epoch=2, lrate=0.300, error=26.065
>epoch=3, lrate=0.300, error=25.762
>epoch=4, lrate=0.300, error=25.530
>epoch=5, lrate=0.300, error=25.268
>epoch=6, lrate=0.300, error=24.966
>epoch=7, lrate=0.300, error=24.617
>epoch=8, lrate=0.300, error=24.213
>epoch=9, lrate=0.300, error=23.747
>epoch=0, lrate=0.300, error=34.900
>epoch=1, lrate=0.300, error=27.092
>epoch=2, lrate=0.300, error=26.468
>epoch=3, lrate=0.300, error=26.357
>epoch=4, lrate=0.300, error=26.250
>epoch=5, lrate=0.300, error=26.138
>epoch=6, lrate=0.300, error=26.017
>epoch=7, lrate=0.300, error=25.884
>epoch=8, lrate=0.300, error=25.735
>epoch=9, lrate=0.300, error=25.567
Scores: [48.0, 82.0]
Mean Accuracy: 65.000%
```

Figure 8.10: Output of procedure based on Custom_ANN_Backpropagation_SGD_Ensemble

8.8 Creating a Custom Algorithm for Image Identification and Classification (Enhanced CNN Model)

Identification of contraband images (depicting 'child pornography') from a gathered evidence is an important goal of the prototype model for effective digital forensic investigation. In order to achieve this Convolutional Neural Network (CNN) techniques can be utilised. A customised 'Enhanced CNN Model' is implemented, which is further optimised using '3 Block VGG' algorithm to further extend the functional capabilities of DIF^2 framework.

8.8.1 Convolutional Neural Network

A Convolutional Neural Network (CNN or ConvNet) is a specialized type of artificial neural network (ANN), that employ deep learning for processing structured grid data, most prominently images and videos. CNNs have substantially transformed the landscape of computer vision tasks, offering capabilities in image classification, object detection, image generation, and more using deep learning techniques. They excel at capturing hierarchical patterns and features inherent in visual data. At the core of Convolutional Neural Networks

(CNNs) are convolutional layers, pivotal building blocks that apply convolution operations to input data using adaptable filters or kernels. These layers excel at identifying local patterns and features. Additionally, pooling layers frequently complement convolutional layers, reducing the spatial dimensions of feature maps while preserving essential information. Activation functions like ReLU (Rectified Linear Unit) introduce non-linearity into the model, enabling the capture of intricate data relationships.

Fully connected layers typically follow the feature extraction stages, allowing CNNs to make final predictions. They connect neurons across layers. CNNs are adept at handling multi-channel inputs, such as RGB colour images with their three channels. Weight sharing is a fundamental concept, as the same set of weights (filters) is applied to different segments of the input data, reducing parameter count for efficiency. CNNs autonomously learn hierarchical features, with lower layers detecting basic features like edges and textures, while higher layers amalgamate these features to recognize more intricate objects and patterns. CNNs have achieved exceptional success in computer vision, from dominating the ImageNet challenges in image classification to enabling state-of-the-art object detection (Brownlee, 2021).

Image standardisation is needed to resize and reshape images to make it fit for pre-processing using the CNN algorithm. If the image dataset is too large (which is usually the case), Keras image processing API can be used to create training and validation datasets. The dataset images can be organised into 'train' and 'test' datasets using separate folders for images. We created a random image dataset of 75% for training dataset and use the rest 25% for test dataset.

8.8.2 Developing a Customised CNN Model

Initially to establish a baseline performance, a CNN model a single Visual Geometry Group (VGG) is implemented. This implementation serves as a foundational benchmark for evaluating the performance of other models. It establishes a minimum standard against which we can measure the effectiveness of subsequent models. Additionally, it provides us with a foundational model architecture that we can use as a reference point for further study and enhancement. A practical initial approach involves adopting the fundamental architectural principles found in the Visual Geometry Group (VGG) models implemented by that can be used to create the 'CNN Baseline Model' (Simonyan & Zisserman, 2015).

The model architectural design entails the stacking of convolutional layers employing compact 3×3 filters, which are subsequently followed by a max pooling layer. These combined layers

constitute a block, and these blocks can be iterated upon. Importantly, as the network's depth increases, the number of filters within each block also rises. For instance, in the initial four blocks of the model, the filter counts of 32, 64, 128, and 256 is achieved. Additionally, padding is applied to the convolutional layers to guarantee that the dimensions of the output feature maps align with the input dimensions in terms of height and width. The model uses Rectified Linear Unit or “ReLU activation” function and “He weight” initialiser as most commonly used parameters. This function outputs ‘input’ directly if it returns positive or ‘0’ if negative. A node or unit that employs the rectified linear activation function is commonly known as a “Rectified Linear Activation Unit”, abbreviated as ReLU. Networks that incorporate the rectifier function in their hidden layers are frequently referred to as "rectified networks”. The model will utilise Stochastic Gradient Descent (SGD) to perform binary classification (predicting 0 or 1). The model uses sigmoid activation function and “binary crossentropy” loss function. This model can be implemented as a single block, two block or three block model, with higher accuracy in results as number of blocks increase (around 80% with three block model).

8.8.3 Algorithm_Custom_CNN_Model (Image Classification)

Step1: Import necessary libraries: The code begins by importing various libraries like "sys" for system-related functions, "pyplot" for creating plots, and several components from the Keras library, which is a popular deep learning framework.

Step2: Define the CNN model: The "define_model" function creates the architecture of the CNN. It sets up layers for image processing, including convolutional layers (Conv2D), pooling layers (MaxPooling2D), and fully connected layers (Dense). These layers are designed to learn and extract features from images.

Step3: Compile the model: After defining the model, it's compiled using specific settings like the optimizer (SGD with certain parameters), loss function (binary cross-entropy), and metrics (accuracy). This step prepares the model for training.

Step4: Plot diagnostic learning curves: The "summarize_diagnostics" function is used to create two plots. The first plot shows how the loss (error) changes during training for both the training and test datasets. The second plot illustrates how the classification accuracy changes during training. These plots help visualize how well the model is learning over time.

Step5: Run the test harness: The "run_test_harness" function puts everything together. It defines the model, sets up data generators to load and preprocess images, trains the model on the training data, evaluates its performance on the test data, and prints the accuracy achieved.

Algorithm 8.2: Algorithm_Custom_CNN_Model

Overall, the algorithm creates a customised Baseline CNN model to classify images (animal images used as sample), training it on a dataset, and then evaluating its accuracy. The procedure also creates plots to help assess the model's learning progress. Algorithm 8.2 represents the summarised logic steps, while procedure 8.2 represents detailed procedural illustration and shows the functional steps of this coding using pseudocode, that can be implemented independently in any language. The baseline model achieved around **80%** accuracy in image prediction scores, while the baseline model with 'simple data augmentation' and optimisation technique demonstrates an enhanced performance of about **85%.**

8.8.4 Pseudocode for Customised Baseline CNN Model for Image Identification and Classification

Procedure 8.2: Custom_Baseline_CNN_Model

Input: Images Dataset (Cats and Dogs)

Output: accuracy_score (Image prediction)

```
# Import necessary libraries
import sys
import matplotlib.pyplot as pyplot
from keras.utils import to_categorical
from keras.models import Sequential
from keras.layers import Conv2D, MaxPooling2D, Dense, Flatten
from keras.optimizers import SGD
from keras.preprocessing.image import ImageDataGenerator
# Function to define the CNN model
function define_model():
    model = Sequential()
    model.add(Conv2D(32, (3, 3), activation='relu',
kernel_initializer='he_uniform', padding='same', input_shape=(200,
200, 3)))
    model.add(MaxPooling2D((2, 2)))
```

```
    model.add(Flatten())
    model.add(Dense(128, activation='relu',
kernel_initializer='he_uniform'))
    model.add(Dense(1, activation='sigmoid'))
    # Compile the model
    opt = SGD(lr=0.001, momentum=0.9)
    model.compile(optimizer=opt, loss='binary_crossentropy',
metrics=['accuracy'])
    return model

# Function to plot diagnostic learning curves
function summarize_diagnostics(history):
    # Plot loss
    pyplot.subplot(211)
    pyplot.title('Cross Entropy Loss')
    pyplot.plot(history.history['loss'], color='blue',
label='train')
    pyplot.plot(history.history['val_loss'], color='orange',
label='test')
    # Plot accuracy
    pyplot.subplot(212)
    pyplot.title('Classification Accuracy')
    pyplot.plot(history.history['accuracy'], color='blue',
label='train')
    pyplot.plot(history.history['val_accuracy'], color='orange',
label='test')
    # Save plot to file
    filename = sys.argv[0].split('/')[-1]
    pyplot.savefig(filename + '_plot.png')
    pyplot.close()
# Function to run the test harness for evaluating a model
function run_test_harness():
    # Define the model
    model = define_model()
    # Create a data generator
    datagen = ImageDataGenerator(rescale=1.0/255.0)
    # Prepare iterators for training and testing data
```

```
    train_it =
datagen.flow_from_directory('dataset_dogs_vs_cats/train/',
                  class_mode='binary', batch_size=64,
target_size=(200, 200))
    test_it =
datagen.flow_from_directory('dataset_dogs_vs_cats/test/',
                class_mode='binary', batch_size=64,
target_size=(200, 200))


 # Fit the model
    history = model.fit_generator(train_it,
steps_per_epoch=len(train_it),
             validation_data=test_it, validation_steps=len(test_it),
epochs=20, verbose=0)
    # Evaluate the model
    _, acc = model.evaluate_generator(test_it, steps=len(test_it),
verbose=0)
    print('> Accuracy: %.3f' % (acc * 100.0))
    # Generate learning curves
    summarize_diagnostics(history)
# Entry point, run the test harness
run_test_harness()
```

Dropout Regularisation

Dropout regularisation helps a deep learning neural network to get "regularise", i.e. removal of "dropping out" inputs probabilistically which may be input variable from the previous layers. This helps to improve robustness of the nodes and assist in simulating very large networks with varied structures and achieve optimal performance.

8.8.5 Enhancing the Customised CNN Model with Three Block VGG Algorithm (Enhanced CNN Model)

We can further improve the accuracy by using 'three VGG Block' algorithm, which uses 128 filters and can provide higher accuracies as compared to initial baseline model. The pseudocode of the updated model with 'three VGG block' and "dropout regularisation" is provided in procedure-8.3 to create optimised model and achieve increased prediction performance.

Procedure 8. 3: Enhanced_CNN_Model (with Three Block VGG Algorithm)

Input: Images Dataset (Cats and Dogs)

Output: accuracy_score (Image prediction)

```
function define_model():
 # Create a Sequential model
    model = Sequential()
 # Add a Convolutional Layer
    model.add(Conv2D(32, (3, 3), activation='relu',
kernel_initializer='he_uniform',      padding='same',
input_shape=(200, 200, 3)))
 # Add a MaxPooling Layer
    model.add(MaxPooling2D((2, 2)))
    model.add(Dropout(0.2))
 # Add another Convolutional Layer
    model.add(Conv2D(64, (3, 3), activation='relu',
kernel_initializer='he_uniform', padding='same'))
 # Add another MaxPooling Layer
    model.add(MaxPooling2D((2, 2)))
    model.add(Dropout(0.2))
 # Add one more Convolutional Layer
    model.add(Conv2D(128, (3, 3), activation='relu',
kernel_initializer='he_uniform', padding='same'))
# Add one more MaxPooling Layer
    model.add(MaxPooling2D((2, 2)))
    model.add(Dropout(0.2))
# Flatten the output
    model.add(Flatten())
 # Add a Dense Layer (fully connected)
    model.add(Dense(128, activation='relu',
kernel_initializer='he_uniform'))
     model.add(Dropout(0.5))
 # Add the final Dense Layer with sigmoid activation for binary
classification
    model.add(Dense(1, activation='sigmoid'))
 # Compile the model
    opt = SGD(lr=0.001, momentum=0.9)
```

```
    model.compile(optimizer=opt, loss='binary_crossentropy',
metrics=['accuracy'])
        # Return the compiled model
    return model
```

The model is optimised and enhanced further to improve prediction accuracy using '3 Block VGG algorithm'. The 'Enhanced CNN model' produced an overall accuracy score of **97%**. Output of the "Cross Entropy Loss" and "Classification Accuracy" of the customised and 'Enhanced CNN model' developed and optimised further, and performance results are discussed in details in section-9.3.4 (chapter-9).

Preparing for Experiment using Images Dataset (Experiment-4)

In experiment-4, order to train the dataset used in the experiment as discussed in section-5.7.4, two separate sub-directories were created, one with 'normal' images and another with 'contraband' images. For experiment demonstration purpose, a sample dataset consisting of normal images represented as 'cats' images and contraband images (child porn) as 'dogs' images. The publicly available dataset from Kaggle (https://www.kaggle.com/c/dogs-vs-cats/data) was used in the experiment, which consisted of around 25000 images. A 75-25 % split between train and test dataset was done. The 'train' dataset consisted of 18697 images while 'test' dataset had 6303 images. The images were further classified and arranged into subdirectories/subfolders named as cats and dogs. Procedure-8.3 represents the pseudocode, while actual implementation code on the 'Enhanced CNN Model' is represented in section B.14.

Prediction of Normal vs Contraband Images

When a new unseen image is presented to the 'trained' model, it can predict the image's class by using classifier as '0' which represents 'normal' image or '1' as 'contraband' image, which is represented by cat and dog images respectively in experiment-4. This 'Enhanced CNN Model' implementation can be easily integrated into DIF^2 framework for image analysis and newly developed prototype, which is based on I-DEEP protocol, as well as other DF evidence collection tools, to assist first responders/investigators to accurately classify and flag images as contraband (child pornographic) or normal images.

A dataset of images is used to train the model, where the dataset is split into 'train' and 'test' dataset. The 'test' dataset is used to validate the model and test its performance accuracy. Once the model is trained it can be used to predict the status of unseen data (images) and distinguish them into 'normal' or contraband' images. The description of this process is provided under sections 5.7, dataset details and evaluation of performance are provided in section-9.3 in chapter-9.

8.9 Model Extension of DIF^2 Framework using ANN and CNN Algorithms for Predictive Analysis and Image Analysis.

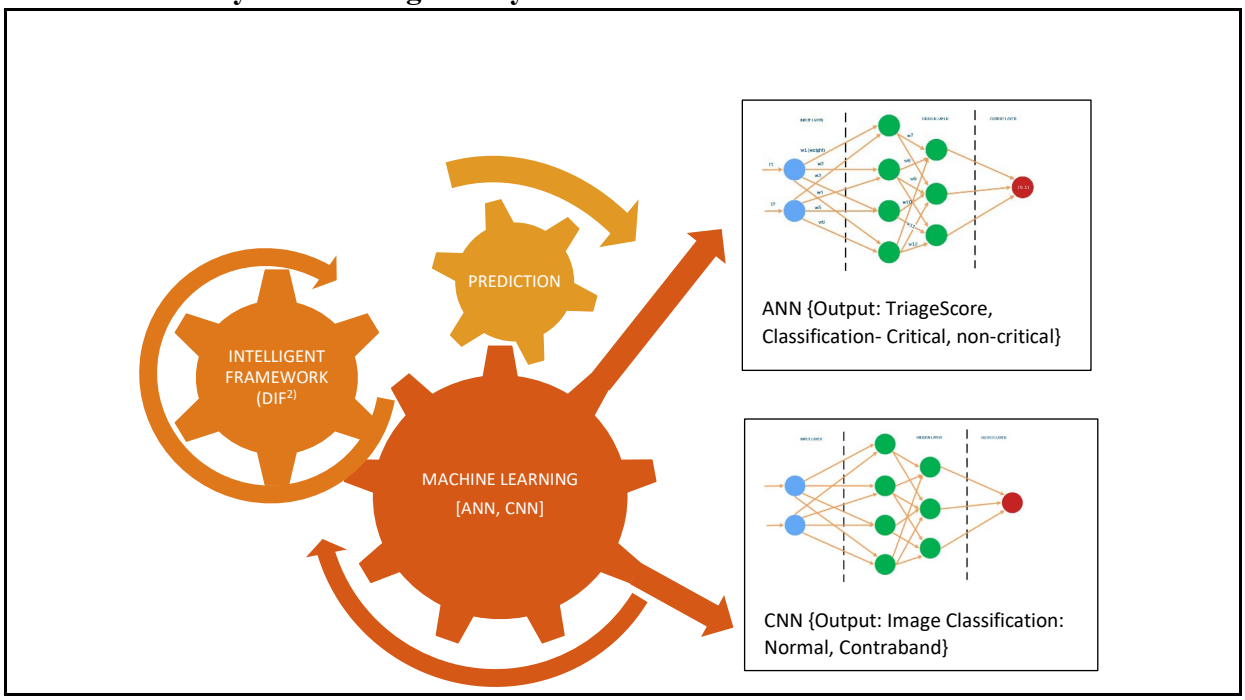


Figure 8.11: Model extension of DIF^2 using ANN and CNN Algorithms Integration (Gearbox Model)

8.10 Discussion on Model Extension and DIF^2 Extended Framework

In the implementation discussed in section-8.9, we have found high potential in Artificial Neural Network (ANN) and Stochastic Gradient Descent (SGD) as they are very effective in prediction modelling for both classification and regression problems and provide high efficacy even if datasets are not large enough. Therefore, a detailed explanation of ANN and SGD implementation has been provided in this chapter under section-8.7. A customised algorithm based on ANN and SGD ensemble was developed and integrated into DIF^2 Framework to further enhance its capabilities and offer more options for conducting predictive analysis. A specialised form of ANN known as Convolutional Neural Network (CNN), which is capable

of performing prediction on image data was also developed and integrated into the framework thus enhancing its capabilities. This model was further enhanced using '3 Block VGG' optimisation. These additional AI and ML integrations into the system provided enhanced options to the newly developed prototype and offer investigators to choose best performing options and tools and use them for better decision making.

The integration is based on 'gearbox model' philosophy of I-DEEP design and DIF^2 framework, where options (tools) providing most optimum functional performance can be chosen. The process followed the 'iterative and incremental' model of 'agile approach' in order to achieve 'Automation' and 'Optimisation' in DF investigations, using customised AI algorithms. Auto-selection of best performing models (automation) and optimisation can be achieved without human intervention using TPOT and AutoML, was discussed in section-8.2.3 and also demonstrated in experiment-2. It was established in section-8.7, that high performance in predictive analysis is achievable with 'smaller datasets', using newly developed 'Custom_ANN_SGD' ensemble model and triaging for better decision making is feasible, as demonstrated in experiment-3. Finally, section-8.8 covered image analysis integration into the DIF^2 framework and prototype using '3 VGG Enhanced CNN' model to extend their functionality as demonstrated in experiment-4.

8.11 Summary

This chapter described extension of the newly developed DIF^2 framework using implementation of TPOT and AutoML in order to achieve 'Automation' and 'Optimisation' of AI Models without human intervention. It further delved into development of customised algorithms based on ANN and SGD ensemble model. These enhancements, in conjunction with the I-DEEP protocol constitutes an 'intelligent framework' development that can provide additional AI integration and performance capabilities. These newly developed artefacts not only align with the established standards of the digital forensic investigation process but also enhances their capabilities further. While existing tools can perform basic analysis of evidential data, they lack the advanced interoperable intelligence discussed earlier in the research and do not support wide AI integration possibilities. The proposed algorithms and framework enhancements offer potential system formulations aim to address this gap. The framework combines classical programming, artificial intelligence (AI), and machine learning (ML) techniques, leveraging on a cybercrime dataset from test data generated using newly developed prototype for training. As AI systems operate based on their knowledge and learned patterns,

the approach is referred to as intelligent systems and intelligent forensics. Retraining is necessary for consistent results, and additional training datasets can be utilized to thoroughly train the AI algorithms, following validation with test datasets. The DIF^2 framework when integrated in prototype can empower investigators to make informed decisions in different scenarios. The integration of AI using 'iterative and incremental' agile approach, description of operational ontology of the 'Automation Process' of DIF^2 framework using TPOT and AutoML, are explained in the chapter in detail, which provides a roadmap for implementing 'Automation' and 'Optimisation' to the AI models with minimum human intervention and also achieving best performance using hyperparameter optimisation. This implementation helped in making predictive analysis performance better, which demonstrated that automation can be effective to reduce human intervention and results into better decision-making capabilities for first responders and DF investigators.

CHAPTER 9: EVALUATION OF PROTOTYPE AND INTERPRETATION OF RESULTS

9.1 Introduction

The main purpose of the research was to develop an AI based 'novel' protocol for first responders using an intelligent framework. This chapter discusses the experiments designed to evaluate the efficacy of the algorithm models developed and described in chapters 7 and 8 and perform analysis of the results obtained. It also compares the efficacy of results and evaluates the performance in terms of implementation of AI framework (DIF^2). The AI framework was developed to demonstrate the potential of AI integration into the novel protocol designed. The framework DIF^2 was developed in stages with incremental technological complexity and the experimental setup was also designed to test the efficacy of the evolutionary prototype in performing prediction modelling of cybercrime incidents and collected evidence (images). The experimental setup comprises of conducting performance evaluation of AI algorithm models developed and testing the performance of these models. To achieve this objective, creating a novel protocol with AI based intelligent framework for first responders and implementing these artefacts as a working prototype was imperative and also included in design process of this research. The results are represented and a comparative analysis is made to interpret the efficacy of the framework in this chapter.

9.2 Evaluation on Cybercrime Incidents (Dataset Extracted from Cybercrime Database)

Statistical description of Dataset is important to understand how data is shaped also helps to cross validate the results after performing data analysis. Dataset and results can be visually represented to better analyse the data and discover the underlying patterns. This helps the first responders to analyse the cybercrime data so that they can take informed decisions and also gain key insights into the trends in cybercrime incidents. After comparing the results obtained from dataset extracted, the observation of dataset is shown in Table-9.1. It displays the statistical analysis based on parameters – count, mean, standard deviation, minimum, 25%, 50%, 75%, maximum. The actual program code implementation and output is displayed in Figure-B.1 in APPENDIX-B.

Table 9.1: Descriptive Analysis of Results (Cybercrime Incidents)

Cybercrime Type	Phishing	ZeroDay Attack	Ransomware Attack	Child Pornography
Mean	8.3700	5.7300	6.8400	2.1000
Standard Deviation	2.1161	1.7341	2.2817	1.3446
Minimum	4	3	2	0
25%	7	4	5	1
50%	8	6	7	2
75%	9	7	8	3
Maximum	15	11	13	5

The descriptive statistical analysis of dataset as shown in figure-9.1, highlights that the most common occurrence is 'phishing' incident and forms the bulk of cybercrime activity in any city. It also shows that the least recorded incident is 'child pornography'. The standard deviation is highest in 'Ransomware Attack' while it is least in 'child pornography' incidents. This data is important because it will provide us insights into the prediction efficacy of AI models that we have implemented as demonstrated as part of various experimental implementations.

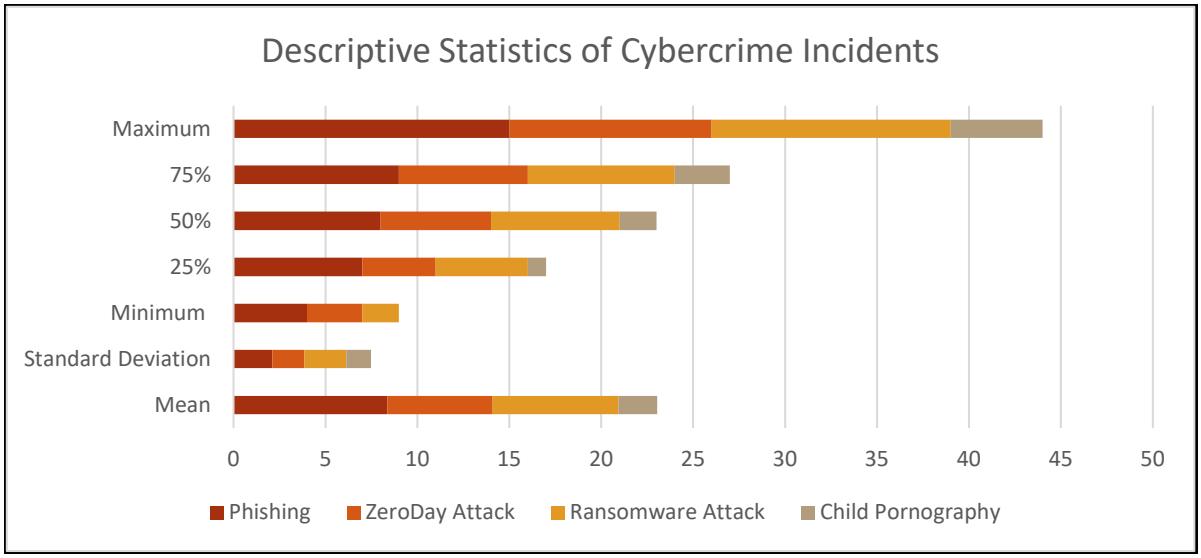


Figure 9.1: Descriptive Statistical Analysis of Cybercrime incidents (Cybercrime Dataset)

9.2.1 Visual Representation of Cybercrime Dataset using Python

Dataset and results can be visually represented into box and whisker plots, histograms, scatter plot matrix to better represent the data and discover the underlying patterns using python libraries and code. This helps the first responders to visually analyse the cybercrime data so that they can take informed decisions and also gain key insights into the trends in cybercrime incidents. The python code implementation provides a 'demonstration' of this using prototype.

The main reason to provide the results (output) here is to establish the functionality of prototype in analysing the cybercrime data collected and confirms to 'visual representation' stage of the 'Extended Linear Model' discussed in section-5.5.2. The 'Reporting' feature of prototype provides the functionality of creating visual representation of cybercrime data, which aligns to the 'demonstration' phase of DSR model adopted for this research. Actual Python implementation is shown in figure B.1 (APPENDIX-B) with code and the output. This section and upcoming sections provide visual representation and discussion on the results obtained.

9.2.2 Box and Whisker Plots of Cybercrime Incidents

Another useful way to review the distribution of each attribute (cybercrime) is to use Box and Whisker Plots or boxplots for short. Boxplots summarize the distribution of each attribute, drawing a line for the median (middle value) and a box around the 25th and 75th percentiles (the middle 50% of the data). The whiskers give an idea of the spread of the data and dots outside of the whiskers show candidate outlier values (values that are 1.5 times greater than the size of spread of the middle 50% of the data). The 'Reporting' feature uses Python program implementation in prototype to visually display cybercrime data as 'Box and whisker plots' of the cybercrime incidents for the four types selected in test data as shown in figure-9.2.

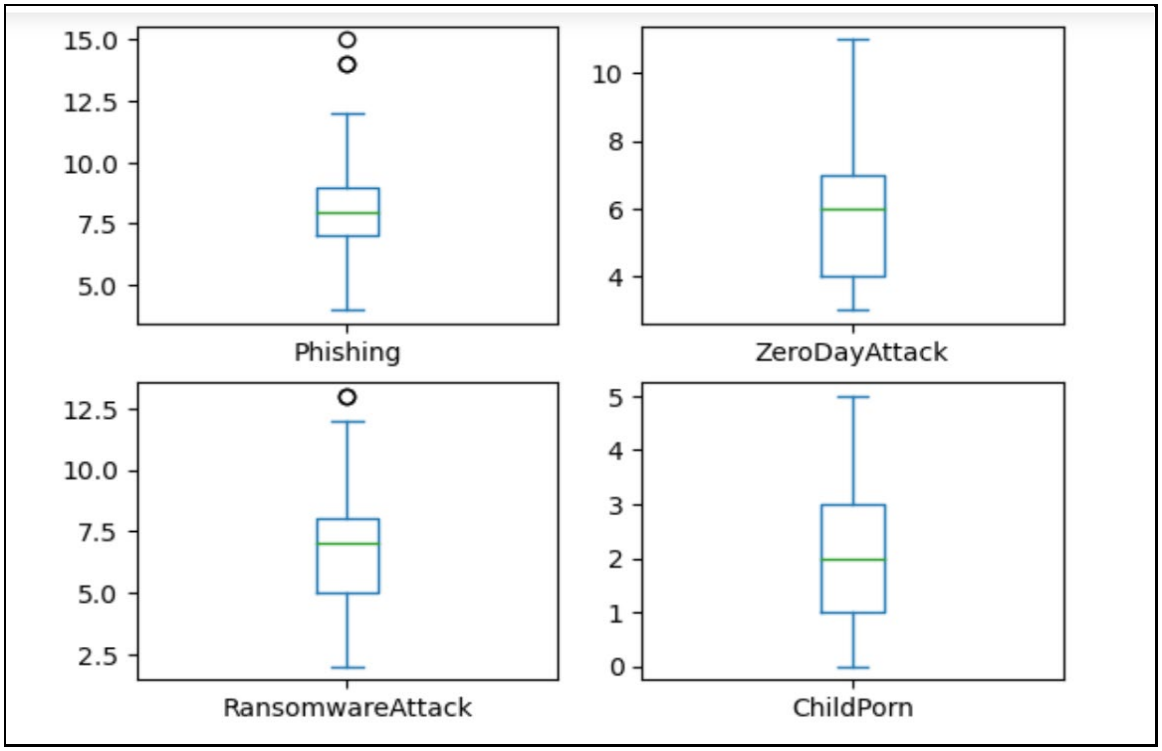


Figure 9.2: Dataset analysis of cybercrime incidents displayed as Box and Whisker Plots

9.2.3 Cybercrime Incidents displayed as Histograms

Histograms can be used to depict the cybercrime incidents on a numeric scale and therefore can be effective to analyse the severity of the cybercrime landscape in the dataset that is analysed. Figure-9.3 shows cybercrime incidents as Histograms and highlights that 'Phishing' is the most common cybercrime reported on all the cities. The histographic analysis show that the 'Child pornography' also had some isolated cybercrime incidents in all three cities.

9.2.4 Correlation Matrix Plot of Cybercrime Incidents

Correlation gives an indication of how related the changes are between two variables. If two variables change in the same direction they are positively correlated. If they change in opposite directions together (one goes up, one goes down), then they are negatively correlated. Correlation can be calculated between each pair of attributes. This is called a correlation matrix. The correlation matrix can be plotted to get an idea of which variables have a high correlation with each other. This is useful to know, because some machine learning algorithms like linear and logistic regression can have poor performance if there are highly correlated input variables in the data.

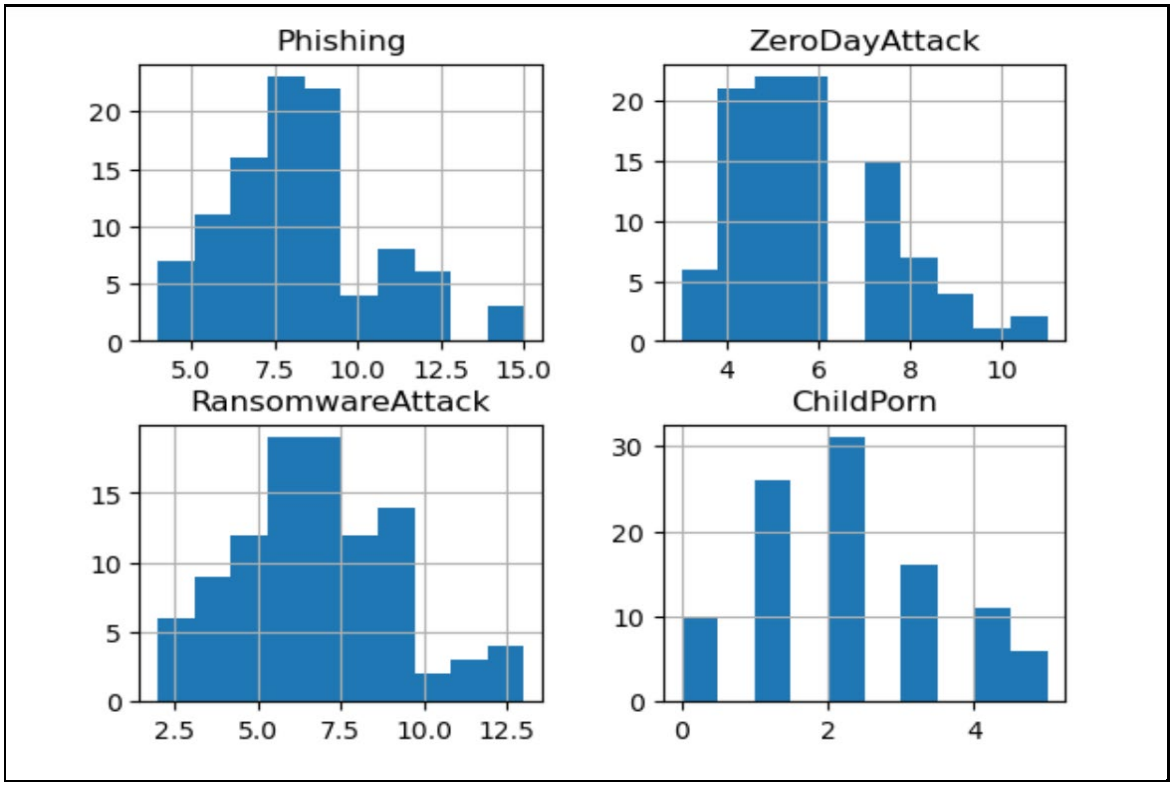


Figure 9.3: Analysis of cybercrime incidents displayed as Correlation Matrix plot

9.2.5 Scatter plot matrix of Cybercrime Incidents

Scatter plot matrix in figure-9.4 shows a distinct pattern of diagonal arrangement of histographic data and scatter plot matrix. This pattern explains why the results show strongest correlation with same 'type' cybercrime incidents in the training dataset when compared with the test dataset, while it shows distinct patterns or correlation with other incidents. New patterns can emerge as the training dataset and test dataset use different 'split methods' as discussed in other experiments (experiment-2) where test harness is created and different models are manually tested.

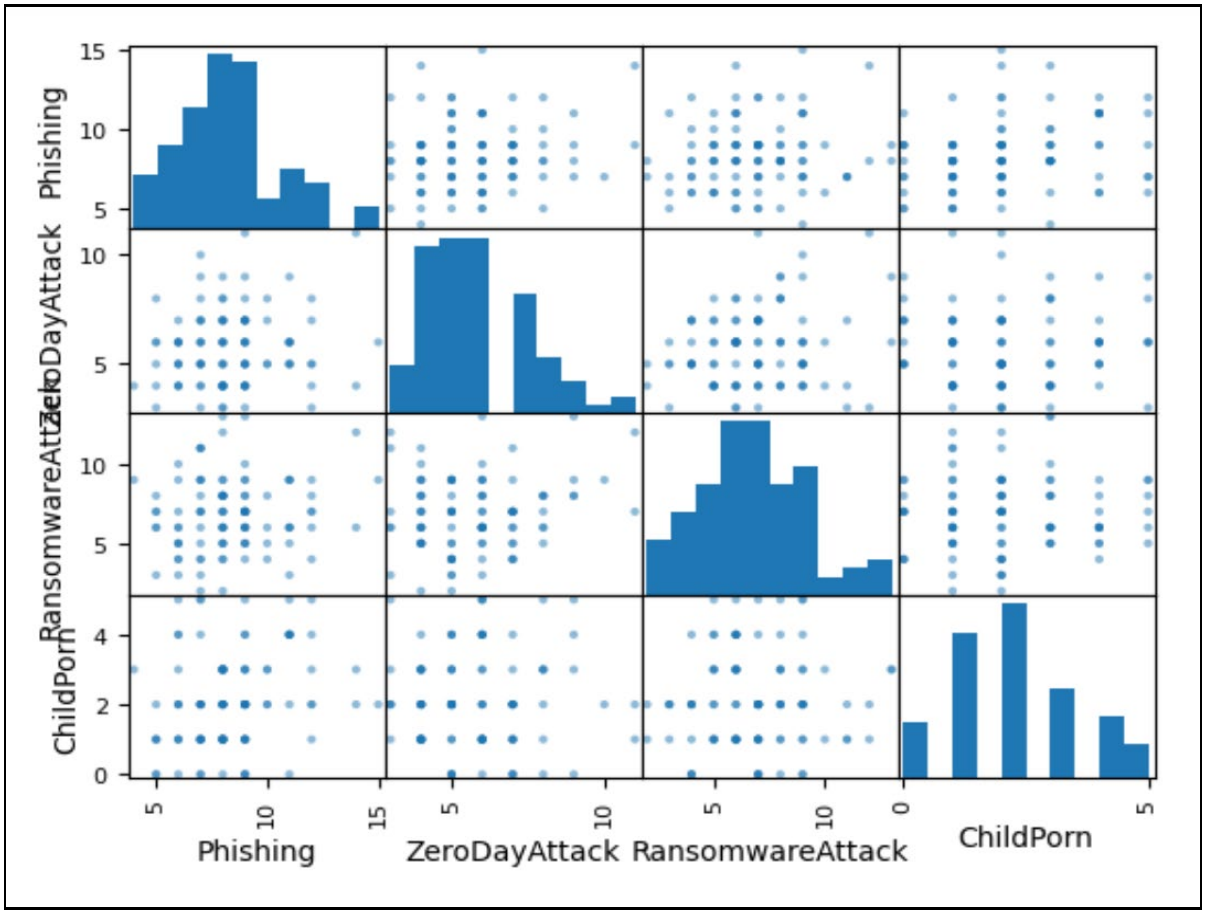


Figure 9.4: Dataset Analysis of Cybercrime Displayed as Scatter plot matrix

Figure 9.5 represents a bar graph to show the classification of cybercrime data as the 'Critical' and 'Non-critical' based on 'City' as a class parameter. This represents 'Johannesburg' as the city that registered most incidents flagged as 'Critical', while least 'Critical' incidents in 'Durban'. This analysis can assist in efficient 'triaging' process and decision making by first responders and cybercrime investigators. The graph is based on pivot table that uses 'Critical' class as index and 'City' as variable for classification of cybercrime incidents.

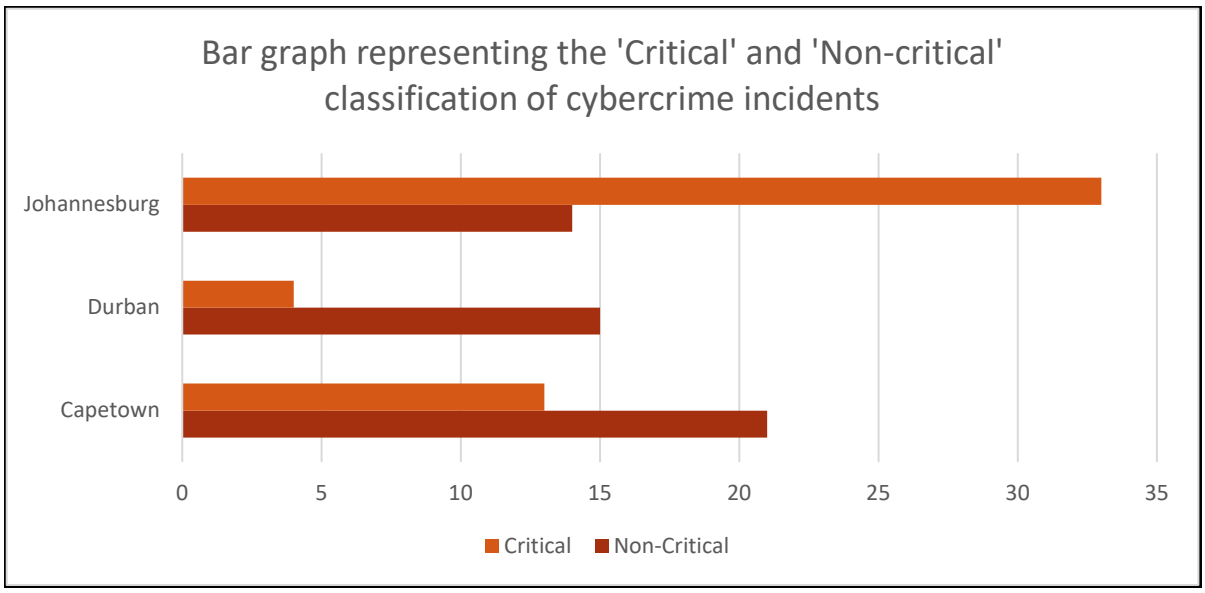


Figure 9.5: City-wise representation of cybercrime data classified as ‘Critical’ or ‘Non-critical’

9.3 Experiment Results Evaluation and Discussion

Several research experiments were conducted to assess the effectiveness of AI algorithm models created and deployed within the DIF^2 framework. The evaluation considered various parameters such as dataset size, cybercrime type, triage score, and the predictive accuracy of the models. The study focused on testing the efficacy of AI algorithms in predictive analysis, measuring accuracy in both classification and regression problems. In classification problems, the aim was to predict the occurrence of a data instance within a specific class, while regression analysis aimed to identify trends that data instances follow and predict their outcomes. This experimental setup consisted of four experiments initially, as discussed in detail in section 5.7 and the next iteration included adding three more experiments. The evaluation of results of the experiment and discussion on the efficacy of the models is presented in this section.

Dataset contains cybercrime cases recorded under variables defined as cybercrime incidents in three major cities in South Africa. Therefore extracted dataset consists of four columns identifying ‘cybercrime-type’ like Phishing, ZeroDay Attack, Ransomware Attack, Child Pornography and the City. In order to test the models and their performance, the extracted dataset consisted of ‘count’ of four types of cybercrime incidents namely ‘Phishing, ZeroDay Attack, Ransomware Attack, Child Pornography’, which occurred in three major cities ‘Durban, Johannesburg and Capetown’. The cybercrime dataset consisting of count of incidents, extracted on ‘type’ represent lowest severity level (Phishing) to highest (Child Porn)

and therefore provides a severity spectrum in cybercrime 'type' parameter. The dataset description is important to understand the nature of data comprising the dataset, so that a appropriate strategy can be adopted to analyse the data, also the output can be cross validated. This dataset is available in a public 'cloud based' repository specified by the URL parameter.

9.3.1 Experiment-1: Stage-1 Observations and Discussion

This research experiment tested the baseline algorithm model developed in order to establish a baseline score for testing and evaluation of the models. This model was tested on datasets with 50 and 100 records to compare the performance. The experiment was extended further to train and test on variable dataset size (50 vs 100 vs 300 records) labelled as experiments-5,6,7 and results compared.

Another parameter tested in this experiment was the training and test data splits percentage. Table-9.2 shows the results of the experiment. Figure-9.6 demonstrates the results via a histogram chart. The prediction accuracy scores provide evidence that the efficacy in results shows correlation with the dataset size, although it is not very apparent considering the fact that the dataset size does not remarkably differ. Also, AI algorithm models also have tendency of getting saturated if dataset is too big as discussed by Brownlee (2016) in his experiment. System resources and configuration parameters also play an important role in running the experiments and model testing. Another parameter tested in this experiment was the dataset split into 'training and test' datasets. This is implemented by creating a test harness to implement the model testing.

Table 9.2: Table showing 'Prediction Accuracy' scores for Zero Rule Algorithm Baseline model

Dataset Train-Test Split %	**Accuracy Scores Dataset (50 Records)**	**Accuracy Scores Dataset (100 Records)**
Split 50-50%	47	48
Split 60-40%	42	45
Split 70-30%	38	40
Split 80-20%	33	35
Split 90-10%	27	30

The train-test split serves as a straightforward resampling technique for assessing machine learning algorithms. Consequently, it serves as a favourable initial step in establishing a test harness that can be further used for testing much advanced models later. This process calls for partitioning a dataset into training and testing sets and for evaluating the accuracy of a set of

predictions. The optimal split percentage is to be determined in order to achieve better results. However, better results do not always guarantee that the model is learning from the data patterns and predictions can be random also. The AI models' parameter tuning is an essential step towards achieving optimal performance.

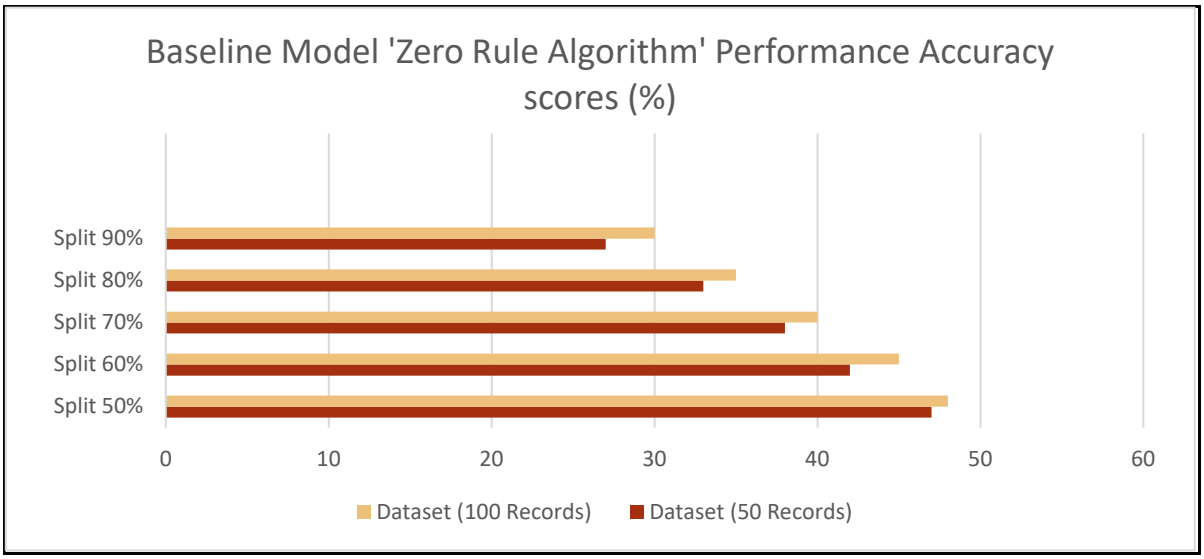


Figure 9.6: Baseline Model 'Zero Rule Algorithm' performance accuracy scores

On analysis of the results obtained on different split percentages, it is observed that 50-50% split produced best prediction accuracy scores. This explains the nature of the model (baseline model) that it is simply making predictions randomly and needs further training or feature enhancements. This experiment helps to establish a baseline score for other model evaluations in further experiments and also helps to learn about model behaviour. The advantage of creating the baseline model is that, it presents a formal structure to create test harness that is used to create 'train and test' data splits and also provides a template to add more complexity to the development of forthcoming models.

Experiment-1: Stage-2 Observations and Discussion

This experimental implementation (Experiment-1, Stage-2) comprised of creating a test harness to test existing AI models for predictive modelling for classification and regression problems. This involved steps like creating a test harness, loading the datasets, loading the models, spot-checking the models using k-fold validation process, and performance comparison of various AI algorithms. The existing models are spot-checked using split validation technique. This setup tested the performances using two datasets (dataset of 50

records and 100 records) and the results were compared. Code Snippet-9.1 shows the python code implementation and output of the experimental results.

```
# Load dataset
url = "https://raw.githubusercontent.com/deep0025/CyberCrimeData-SA/main/CybercrimeDataset50.csv"
names = ['Phishing', 'ZeroDayAttack', 'RansomwareAttack', 'ChildPorn', 'City']
dataset = read_csv(url, names=names)
# Split-out validation dataset
array = dataset.values
X = array[:,0:4]
y = array[:,4]
X_train, X_validation, Y_train, Y_validation = train_test_split(X, y, test_size=0.20, random_state=1, shuffle=True)
# Spot Check Algorithms
models = []
models.append(('LR', LogisticRegression(solver='liblinear', multi_class='ovr')))
models.append(('LDA', LinearDiscriminantAnalysis()))
models.append(('KNN', KNeighborsClassifier()))
models.append(('CART', DecisionTreeClassifier()))
models.append(('NB', GaussianNB()))
models.append(('SVM', SVC(gamma='auto')))
# evaluate each model in turn
results = []
names = []
for name, model in models:
    kfold = StratifiedKFold(n_splits=10, random_state=1, shuffle=True)
    cv_results = cross_val_score(model, X_train, Y_train, cv=kfold, scoring='accuracy')
    results.append(cv_results)
    names.append(name)
    print('%s: %f (%f)' % (name, cv_results.mean(), cv_results.std()))
# Compare Algorithms
pyplot.boxplot(results, labels=names)
pyplot.title('Algorithm Comparison')
pyplot.show()
```

```
LR: 0.625000 (0.201556)
LDA: 0.525000 (0.207666)
KNN: 0.625000 (0.201556)
CART: 0.300000 (0.244949)
NB: 0.525000 (0.261008)
SVM: 0.475000 (0.134629)
```

Code Snippet 9.1: Spot-check Algorithm and Output (CV scores and Standard Deviation)

The Results provided in table-9.3, show mean values of Cross Validation (CV) scores and standard deviation for spot-checked algorithm using dataset of 50 records. This indicates the performance of the models during spot-check validation. These results provide a more reliable estimate of a model's performance compared to a single 'train-test' split, offering insights into its generalization ability and stability across different subsets of the data. Figure-9.7 shows the performance comparison of spot-check algorithms as a histogram chart with 50 records.

Table 9.3: Spot-check Algorithm results shown using a dataset of 50 records

Model Name	**Mean Value**	**Standard Deviation**
Logistic Regression (LR)	0.625000	0.201556
Linear Discriminant Analysis (LDA)	0.525000	0.207666
K-NeighborsClassifier (KNN)	0.625000	0.201556
Decision Tree Classifier (CART)	0.300000	0.244949
GaussianNB (NB)	0.525000	0.261008
SVM	0.475000	0.134629

This experiment was repeated with a dataset of 100 records. Results of the Spot-check algorithm validation process on a dataset with 100 records is provided in Table-9.4, which consists of mean value scores (CV results) and standard deviation.

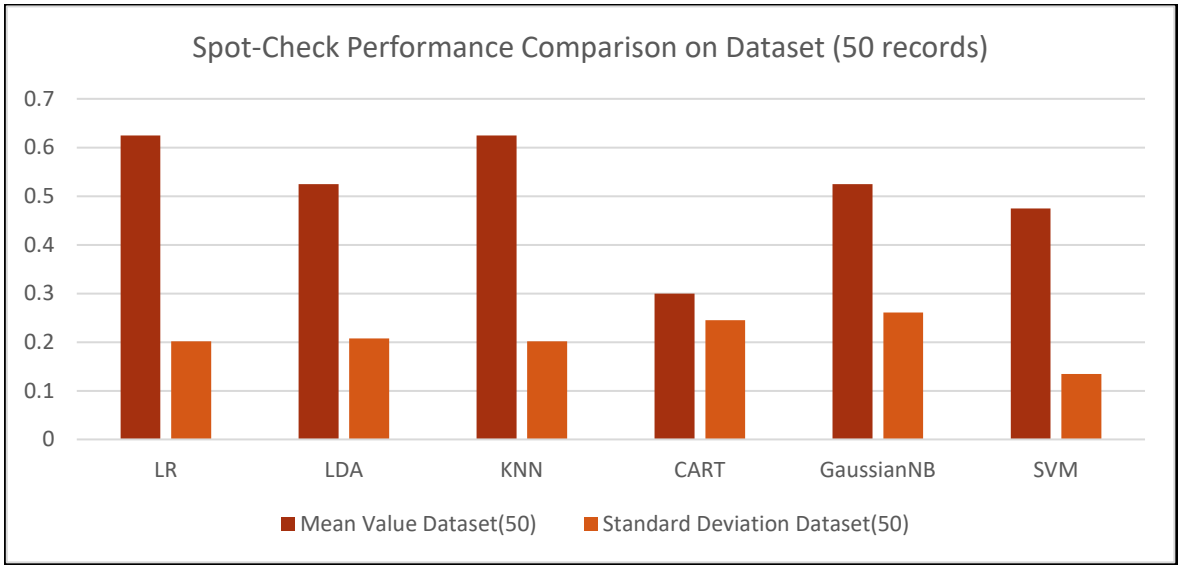


Figure 9.7: Spot Check comparison of algorithms scores on dataset (50 records)

The Mean value scores and standard deviations of spot checking of different AI algorithms on a dataset of 50 records are represented in figure-9.7. These parameters provide us with an indication of which algorithms are more suitable for predictive analysis of a given dataset. We observe that K-NeighborsClassifier (KNN) and Logistic Regression (LR) both provide an equal performance in terms of mean value of predictive accuracy scores of 62.5% as also represented it the table 9.4 and also shown in Code snippet-9.1 program output. Both algorithms show a standard deviation of 20% which again represents a high variation.

Table 9.4: Spot-check Algorithm results shown using a dataset of 100 records

Model Name	**Mean Value**	**Standard Deviation**
Logistic Regression (LR)	0.637500	0.171847
Linear Discriminant Analysis (LDA)	0.675000	0.178536
K-NeighborsClassifier (KNN)	0.600000	0.145774
Decision Tree Classifier (CART)	0.525000	0.165831
GaussianNB (NB)	0.612500	0.152582
SVM	0.625000	0.136931

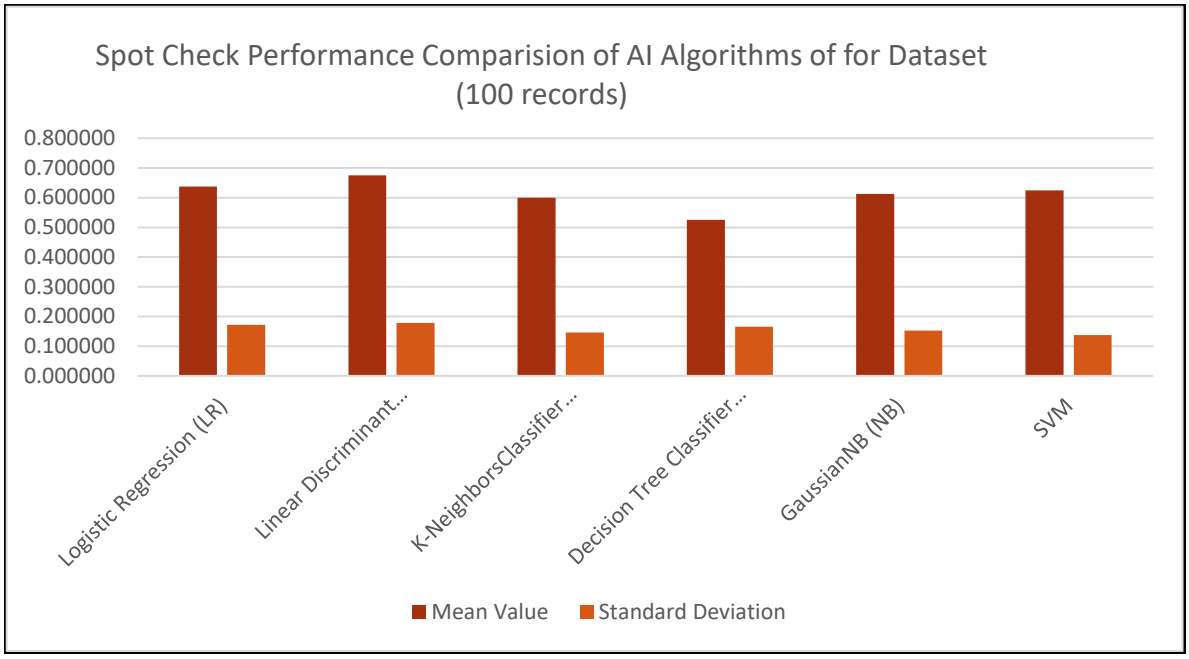


Figure 9.8: Mean Accuracy Scores of Spot Check of AI Algorithms (100 records)

Table- 9.5 represents the experiment results conducted on a dataset of cybercrime with 100 records. We observe performance improvements in the results with Linear Discriminant Analysis (LDA) proving to be best performing algorithm with a mean accuracy score of **67.5%** and lower standard deviation of 17.8%, which demonstrates better performance and shows that dataset size can be an important parameter of performance metrics for algorithm comparison.

The results of experiment conducted with dataset of 100 records and represented in table-9.5 are displaced as histographic chart in figure-9.8, which shows mean accuracy scores comparison of algorithms and the standard deviations which shows LDA as best performing algorithm.

Figure-9.9 shows a histographic comparision of results of descriptive statistics of both implementations on datasets (50 vs 100) and a discussion on the results is provided henceforth. The description of this experimental setup has been provided in section 5.7 and its implementation has been discussed in sections 7.7 and 7.8.

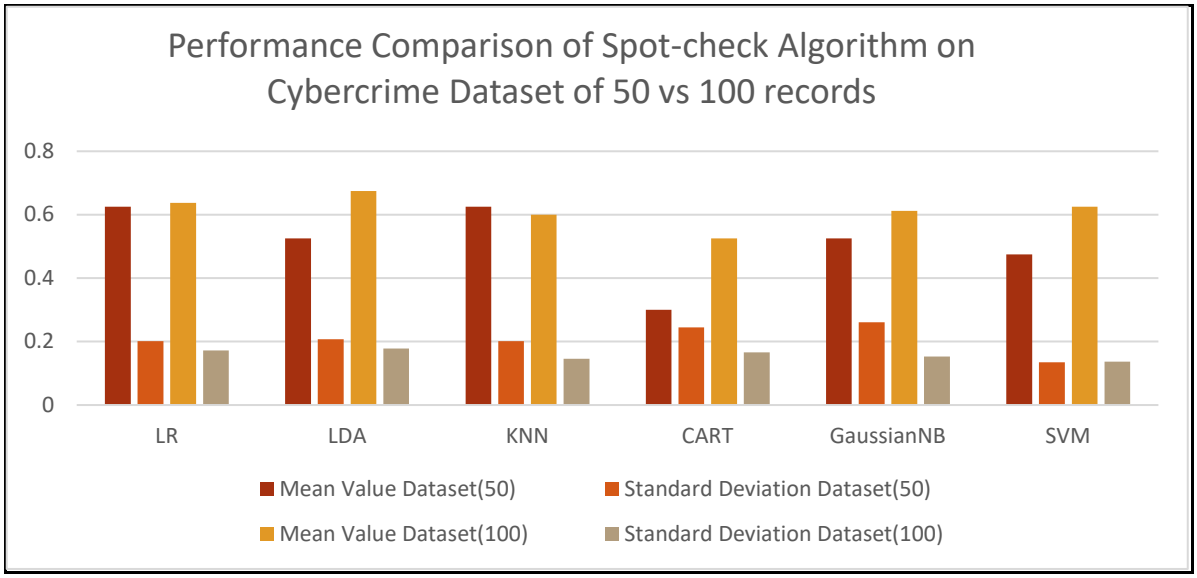


Figure 9.9: Histographic comparison of performance of spot-check algorithm using 50 vs 100 records dataset.

Analysis and Discussion

We observe that in case of dataset with 50 records, the best performing algorithm models were LR and KNN. This has to be mentioned that this dataset had three classes (city) and a split may had only few instances of a specific class considering the small dataset size. We observe that best performing algorithm models are LR and KNN in 50 records experimental setup with overall CV Score of **62.50%** (Table-9.3). On the other hand, with a dataset of 100 records, Linear Discriminant Analysis (LDA) has highest accuracy mean value of **67.50%** as also shown in table-9.4. Figure-9.8 shows histographic representation of output of 100 records, along with standard deviation. CART (Decision Tree) had the lowest accuracy among the models with a score of 52.50%.

Standard deviation provides a measure of how much the accuracy scores vary across different runs or folds in cross-validation. Lower standard deviations (e.g., SVM) suggest more consistent performance. LDA appears to outperform the other models in terms of accuracy, while Decision Tree (CART) has the lowest accuracy. SVM and LR also show relatively competitive accuracy scores. It is essential to interpret these results in the context of the specific problem being solved and the characteristics of the dataset. Additionally, further analysis, such as examining precision, recall, and f1-score, could provide a more comprehensive understanding of model performance, especially if there is class imbalance.

It is also noted that standard deviation of results of 100 records experiment is lower than 50 records. This reflects well on experiment (Dataset-100) as compared to (Dataset-50) and also shows better performance. Therefore number of records in a dataset can improve the performance of the model and show positive correlation. Although a stage arrives when model might get overwhelmed by a huge dataset and therefore may not provide best performance as discussed by Brownlee (2016) and Johnson and Ananthakumaran (2021) in their observations. These scores can be regarded as a decent performance when it comes to predictive analysis and upcoming implementations of new models as described under 'Research Experiment' in section-5.7. We strive to achieve better performance systematically by incorporating more advanced AI implementation and by further developing customised models.

Output of this experiment produced results with varying attributes on both datasets. This experimental setup compared performance data produced by analysis of the two datasets (50 records vs 100), also presented in table-9.3 and table-9.4. This experiment demonstrated that the overall best performing algorithm on the given dataset (100 records) was Linear Discriminant Analysis (LDA) has highest accuracy mean value of **67.50 %** as shown in figure-9.8. Figure 9.9 represents histographic comparison of performance of spot-check algorithm using 50 and 100 records. This comparison indicates that dataset size shows a positive correlation with performance of algorithm models.

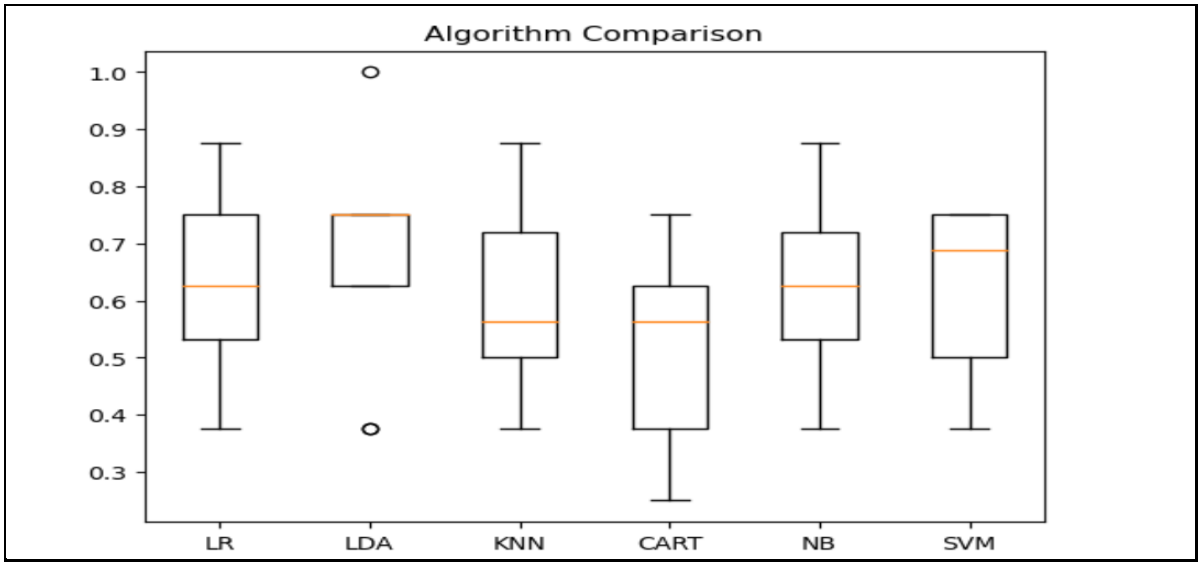


Figure 9.10: Box and Whisker plot of Spot Check AI algorithm results

The actual implantation code of experiment using Python language and its output is shown in section-B.3 (figure-B.3 in APPENDIX-B), while the visual output of spot-check validation process is represented as 'box and whisker plot' in figure-9.10.

The Box and Whisker Plot demonstrates the range between 25% and 75% of standard deviation as box and the mean as a line in between, while the outliers are displayed as circles as shown in figure-9.10. While every AI algorithm model perform differently on different type of problems and type of dataset, it is very difficult to predict which model will perform best until the test is done. It is also a manual process and involves manual testing of models using a test harness. This experiment shows a good performance (67.5%) by the LDA. Next experiments were designed to automate the process of AI model testing using AutoML and TPOT in order to further improve performance of predictive analysis of data (cybercrime dataset), automate the algorithm selection process and perform predicative analysis on cybercrime datasets. The ontology of this experiment (TPOT-AutoML) is explained in sections 8.6.

Comparing prediction parameters of 50 vs 100 dataset in experiment-1

Next step involves comparing other parameters of the best performing models in both (50 vs 100) experimental conditions. Figure-9.11 shows the prediction parameters with dataset-50.

```
In [18]:
1 print(dataset.shape)

(50, 5)

In [19]:
1 # Make predictions on validation dataset
2 model = LogisticRegression(solver='liblinear', multi_class='ovr')
3 model.fit(X_train, Y_train)
4 predictions = model.predict(X_validation)

In [20]:
1 # Evaluate predictions
2 print(accuracy_score(Y_validation, predictions))
3 print(confusion_matrix(Y_validation, predictions))
4 print(classification_report(Y_validation, predictions))

0.6
[[0 0 3]
 [0 1 1]
 [0 0 5]]
              precision    recall  f1-score   support

    Capetown       0.00      0.00      0.00         3
      Durban       1.00      0.50      0.67         2
johannesburg       0.56      1.00      0.71         5

    accuracy                           0.60        10
   macro avg       0.52      0.50      0.46        10
weighted avg       0.48      0.60      0.49        10
```

Figure 9.11: Performance Parameter Comparison of LR (Best performing Algorithm) on Dataset-50 experiment

While running prediction validation separately on the best performing model (LR) for Dataset-50 experiment, we see a precision value=0, recall value =0 and f1-score also as '0' and a low support (support=3) for class value 'Capetown'. Figure-9.12 shows performance and prediction parameters in dataset-100.

Precision, recall, f1-score, and support are commonly used metrics to evaluate the performance of a classification algorithm, including AI prediction algorithms. These metrics provide insights into different aspects of the model's predictive ability.

Precision is the ratio of true positive predictions to the total number of positive predictions made by the model. Precision focuses on the accuracy of positive predictions. A high precision indicates that when the model predicts a positive outcome, it is likely to be correct.

Recall is the ratio of true positive predictions to the total number of actual positive instances in the dataset. Recall measures the model's ability to capture all the positive instances in the dataset. A high recall indicates that the model can identify a large proportion of the actual positive instances.

The f1-score is the harmonic mean of precision and recall. It provides a balance between precision and recall. The f1-score is particularly useful when there is an uneven class distribution. It is a single metric that considers both false positives and false negatives. A higher f1-score indicates a better balance between precision and recall.

```
[13]: # Make predictions on validation dataset
      model = LogisticRegression(solver='liblinear', multi_class='ovr')
      model.fit(X_train, Y_train)
      predictions = model.predict(X_validation)

[14]: # Evaluate predictions
      print(accuracy_score(Y_validation, predictions))
      print(confusion_matrix(Y_validation, predictions))
      print(classification_report(Y_validation, predictions))

0.65
[[5 1 3]
 [2 1 1]
 [0 0 7]]
              precision    recall  f1-score   support

    Capetown       0.71      0.56      0.63         9
      Durban       0.50      0.25      0.33         4
johannesburg       0.64      1.00      0.78         7

    accuracy                           0.65        20
   macro avg       0.62      0.60      0.58        20
weighted avg       0.64      0.65      0.62        20
```

Figure 9.12: Performance Parameter Comparison of LDA (Best performing Algorithm) on Dataset-100 experiment

Support is the number of actual occurrences of the class in the specified dataset. Support provides context by indicating how often the class occurs in the dataset. It helps in understanding the significance of precision, recall, and f1-score in the context of the dataset.

That is a major possibility in small datasets that sample size of train and test data may not contain enough representative values as seen in the case of 'Capetown' class. This impacts the overall performance of algorithm and shows that prediction accuracy is low as in the case of Dataset-50 experiment.

On comparing the performance parameters between two experiments conducted with dataset-50 and dataset-100, we see a significant difference in performance parameters and therefore comparison of these parameters in shown in figure-9.13. Here we can correlate why the performance differs between two experimental setups (50 vs 100). If a bigger dataset is taken, it may provide better performance. However, this comparison is sufficient to prove the hypothesis of correlation between dataset size and prediction accuracy.

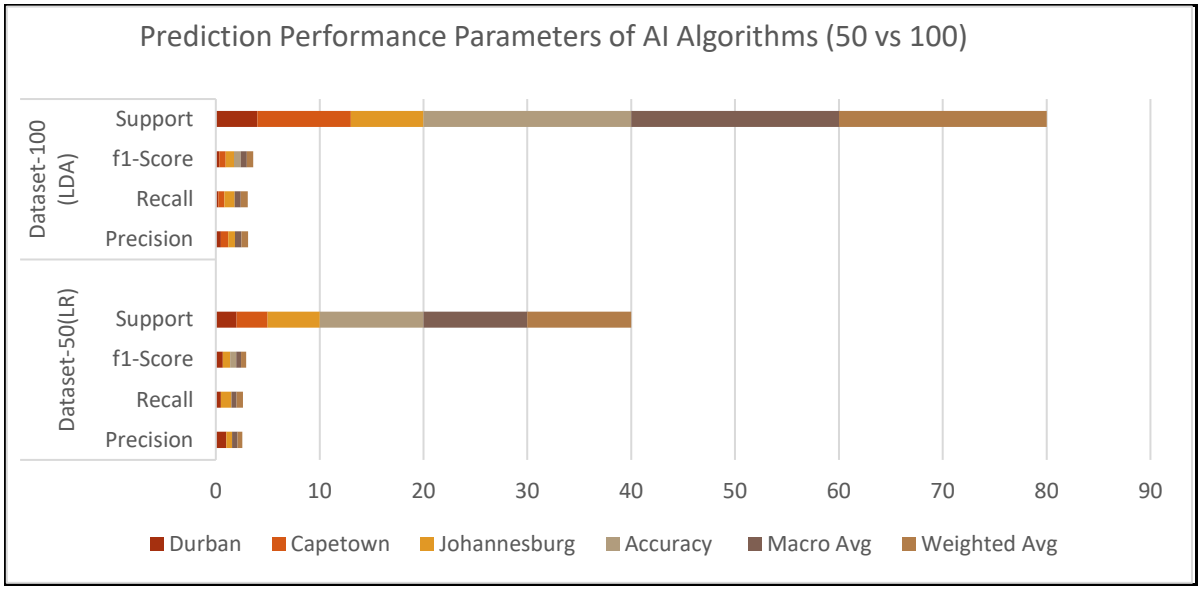


Figure 9.13: Histographic Representation of Prediction Performance Parameters of AI Algorithms (Class\City-wise)

Accuracy and Dataset Size

In order to establish the correlation between the size of dataset and the accuracy of models, an analysis of a dataset with 50 records was done and compared with a dataset of 100 records and results compared. A positive correlation is identified in the performance of models with the size of dataset. Code Snippet-9.1 shows the actual implementation and output of the experiment

consisting of a dataset of 50 records. The implementation for a dataset with 100 records experimental setup, is shown in section-B.3 (figure-B.3, APPENDIX-B). Figure-9.13 clearly illustrates the relationship between the performance parameters, dataset size and accuracy scores. A dataset with bigger size produces better accuracy results.

Next experiments were designed to automate the process of AI model selection and testing using AutoML and TPOT in order to further reduce the effort and perform automated predictive analysis of data (cybercrime dataset) and predict the incidents as explained in sections 8.6 in detail.

9.3.2 Experiment-2: Observations and Discussion

This experiment was illustrated in section-5.7.2 and implementation explained in detail in section-8.6. It showcased the 'Automation' and 'Optimization' of the DIF^2 intelligent framework through the application of AutoML and TPOT, yielding several notable benefits outlined in this discussion and also covered in section-8.6. By incorporating TPOT and AutoML, the research gains access to state-of-the-art techniques, including pruned decision trees, Gradient Boosting, and customized Deep Learning Algorithms. Additionally, TPOT provides an array of advanced encoding and feature preprocessing methods.

In any Machine Learning system, the presence of hyperparameters is inevitable, and the primary objective of AutoML is to autonomously configure these hyperparameters to enhance the performance of models. This automated configuration significantly reduces the level of human effort required for implementing machine learning, highlighting a crucial aspect of AutoML. With TPOT, the research takes advantage of cutting-edge techniques, such as pruned decision trees, Gradient Boosting, and tailored Deep Learning Algorithms. Furthermore, TPOT offers a diverse set of advanced encoding and feature preprocessing methods, contributing to the overall effectiveness and versatility of the ‘automated and optimized’ DIF^2 intelligent framework.

It has been observed that the utilization of automation frameworks like TPOT and AutoML offers significant benefits by reducing human effort in automating the selection of high-performing algorithms and optimisation of hyperparameters. These framework integrations demonstrate the potential of automation techniques that can provide solutions to perform diverse predictive analysis on a wide range of problems and save time. These experiments prove effective in decision-making, aligning with one of the key objectives of this study.

However, it's crucial to acknowledge the practical complexities and resource demands associated with the implementation of AutoML and TPOT. The inherent complexity of AutoML and TPOT becomes evident during the implementation process. While larger datasets can lead to improved accuracy, excessively large datasets pose challenges and create substantial overhead in terms of time and resources. This can be especially problematic in the context of the DIF^2 framework, where optimization aims to enhance the overall performance.

The actual output of various pipelines is discussed in section-8.6 and presented in figure-8.9, showing a creditable efficacy score of around **64.66%** achieved using the best-selected model, KNeighbors Classifier (KNN). However, there is only a marginal improvement in the output score from generation-1 to generation-5. This modest improvement observed in experimental implementation is attributed to the relatively small dataset, emphasizing the need for a more substantial training dataset to realize the full potential of the optimization process.

Table-9.5 shows the output CV (Cross Validation) results of the five generation pipelines using KNN model as best pipeline.

Table 9.5: Output of Experiment-2 showing CV results of the five generation pipelines for KNN model using TPOT and AutoML

Generations (Iterations) using KNN Model	CV Score
1	0.6400
2	0.6466
3	0.6466
4	0.6466
5	0.6467

Figure 9.14 shows the results of the pipelines optimisation process of KNN model using AutoML and TPOT automation implemented in DIF2 framework. Although being resource intensive implementation, TPOT and AutoML offers great possibilities for automation of DIF^2 intelligent framework in future and also supports optimisation pipeline architecture.

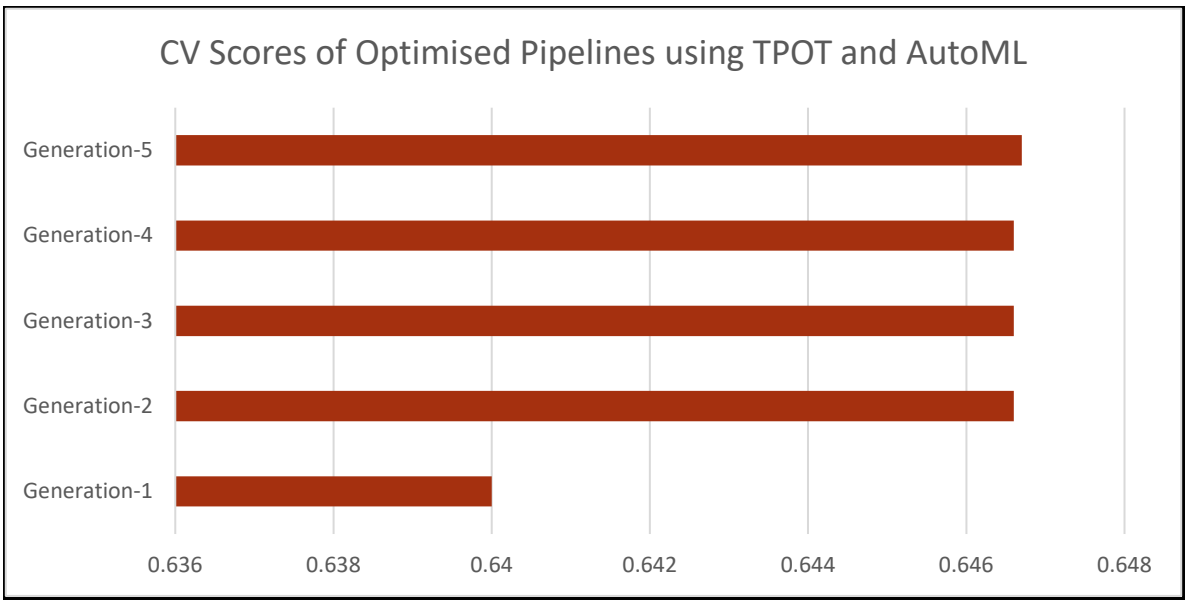


Figure 9.14: Optimised Pipelines CV Scores for KNN Model using AutoML and TPOT

Evaluation of Results and Criticism

Although a significant improvement in performance is observed from generation-1 to generation-2. Thereafter, there is little improvement seen in the results from generation-3 to generation-5. This behaviour can be attributed to the dataset size and the classification problem that the experiment was performed on. Also, to the fact that dataset on which the experiment was conducted was not rescaled or normalised. KNN algorithm perform better on rescaled and normalised datasets.

These results can be further interpreted by understanding the nature of optimisation algorithms (TPOT and AutoML) used in the experiment. Genetic algorithms, such as those used by TPOT, rely on the concept of genetic diversity to explore different regions of the search space. If the initial population lacks diversity, the algorithm may converge prematurely to a suboptimal solution, limiting the improvement in subsequent generations. This limitation can be remedied by having a bigger population (bigger dataset) with more diversity of attributes. The performance of the algorithm can also be improved by increasing the number of generations (iterations) as this provides algorithm to perform more thorough explorations. But this has a negative impact of performance of the framework (and the prototype) as they operate on limited resources under test conditions. Due to this resource limitation, the experiment was conducted on a small dataset of 50 records as bigger datasets gave problems during execution.

Another drawback identified is the resource-intensive nature of the implementation, contradicting the main objectives of designing the framework and the prototype, which should be fast, lightweight, and agile. The framework's intended purpose is to offer a quick and efficient means of evaluating functionality and feasibility. Addressing this drawback involves exploring optimization techniques and alternative approaches, as well as considering user feedback to strike a balance between resource efficiency and performance in the DIF^2 framework.

9.3.3 Experiment-3: Observations and Discussions

Experiment-3 involved demonstration and analysis of a newly developed and customised AI algorithm based on Artificial Neural Network (ANN) using Stochastic Gradient Descent (SGD). Main reason to choose ANN and GSD is that these models present a multitude of benefits compared to alternative AI techniques, rendering them a preferred and potent option across a diverse range of applications. Among their advantages is the ability to discern intricate patterns in data-centric tasks such as data classification. ANNs exhibit effectiveness in both linear and non-linear predictive modelling. Noteworthy for their flexibility and adaptability, they prove suitable for a myriad of tasks with minimal adjustments to their architectural design. Additionally, ANNs demonstrate the capacity to autonomously extract pertinent features from raw data, alleviating the need for laborious manual feature engineering processes. ANNs also have shown good performance with smaller datasets as ones used in the experiments in this research.

The model produced ‘mean accuracy’ score of **94%** for a dataset of records-50 and a score of **97%** with dataset of 100 records, as demonstrated in program output shown in section-B.12 (figure-B.12, APPENDIX-B). This experiment involved parameters settings such as number of folds (n_folds) = 5; learning rate = 30%; number of epoch (n_epoch) = 500 and number of hidden layers (n_hidden) = 5.

A higher percentage of learning rate is normally avoided as high learning rate might cause the optimization algorithm to skip over the minimum of the cost function. High learning rates can cause optimization algorithm to converge to suboptimal or poor local minima instead of the global minimum.

In order to demonstrate the functioning of the algorithm and to show how algorithm performs where we see ‘error rate’ drops gradually with ‘each epoch’, a configuration test setting of

parameters of number of folds (n_folds) = 2 and number of epoch (n_epoch) = 10, was tested and the results are shown in table-9.6. This setting provides an output that can be displayed conveniently and provides the demonstration of the functionality of customised algorithm.

Table 9.6: Error Rate Comparison of Customised Algorithm Developed based on ANN and SGD for Performance Analysis

>epoch=0	lrate=0.300	error=36.532
>epoch=1	lrate=0.300	error=29.686
>epoch=2	lrate=0.300	error=26.065
>epoch=3	lrate=0.300	error=25.762
>epoch=4	lrate=0.300	error=25.530
>epoch=5	lrate=0.300	error=25.268
>epoch=6	lrate=0.300	error=24.966
>epoch=7	lrate=0.300	error=24.617
>epoch=8	lrate=0.300	error=24.213
>epoch=9	lrate=0.300	error=23.747

We observe that the error rate reduces with each epoch and each 'fold' as iteration or 'epoch learning = (number of epoch) x (number of folds)'. The algorithm executes twice as n_folds = 2, and mean error scores are displaced as shown in table-9.6. This demonstration provides a glance into the execution of the algorithm and how it functions. This reduction of error happens due to backpropagation mechanism of customised ANN model which readjusts the weights of each node, at each layer and SGD optimisation algorithm in every iteration or epoch execution.

We decided to repeat the experiment with different configurations and test the algorithm behaviour under different hyperparameter settings and explore best performance outputs. To achieve this, three configurations were tested initially, keeping a fixed learning rate (l_rate) = 30%. These hyperparameters were chosen randomly for testing but keeping in mind the effect that these hypothetical scenarios might have predictive performance.

Table 9.7: Hyperparameter Configuration Settings for Customised Model Based on ANN and SGD analysed in Experiement-3 (l_rate=30%)

Configuration	**Hyperparameters used**	**Mean Accuracy Score**
Config-1	n_folds = 2 l_rate = 0.3 n_epoch = 10 n_hidden = 5	65%
Config-2	n_folds = 3 l_rate = 0.3 n_epoch = 500 n_hidden = 5	97%
Config-3	n_folds = 5 l_rate = 0.3 n_epoch = 100 n_hidden = 5	98%

Table-9.7 shows the 'Mean Accuracy Scores' of the customised algorithm using three different configurations of hyperparameter settings, while keeping the learning rate (l_rate) parameter fixed at 30%. We observe a consistent improvement in accuracy scores in this experiment. We see that configuration-3 produced best results with 98% overall mean score (Table-9.7). This is also proving the fact that model performance improves as it learns more on more iterations and different datasets exposures.

We also tried more experimental variations in hypermeter configurations to analyse the behaviour of the model during experimental testing and different configurations as well as the results are shown in table-9.8.

Table 9.8: Hyperparameter optimisation scores for customised algorithm based on ANN and SGD

Config.	n_folds	l_rate(%)	n_epoch	n_hidden	Mean Score
1.	2	20	10	2	48
2.	2	20	10	3	50
3.	3	30	30	3	94
4.	5	40	100	5	98
5.	5	50	500	5	97

The best mean score was observed for experimental setting in config.4 (n_folds=5, n_rate=40%, n_epoch=100, n_hidden=5) during hypermarameter optimisation experiment conducted on newly developed customised algorithm model based on ANN and SGD ensemble. We also noticed that lowest score was observed in config.1 setting (n_folds=2, l_rate=20%, n_epoch=10, n_hidden=2). Further comparing the results in table-9.7 and 9.8, we observe that a higher values for 'number of epoch' (n_epoch=500) and 'learning rate' (l_rate=50%) parameters do not result into better performance as both configuration produced mean score of 97% which was lower than the best score (config.4). This experiment was carried out on a dataset of 100 records.

Although previous configuration testing as shown in table-9.7, kept the learning rate constant at 30% and different 'n_fold' variations (e.g. 2, 3, 5) were tested against diverse 'n_epoch' values, a consistent increase in performance was observed. This encouraged us to perform more diversified hyperparameter optimisation testing of the model. Therefore, we also tried different variations of l_rate parameters as shown in table-9.8. However, in second experimental observation (refer table-9.8), where different learning rate (l_rate), n_folds and n_epoch configurations were tested, it was observed that increase in learning rate, number of epoch and

number of folds show increase in 'mean scores' for some experiments initially, thus showing a positive correlation. But after a time, increase in l_rate, n_folds or n_epoch did not result in improved scores (refer config. 5). This analysis confirmed the hypothesis that a higher learning rate (l_rate=50%) can cause the optimization algorithm to skip over the minimum of the cost function. High learning rates can also cause optimization algorithm (SGD) to converge to suboptimal or poor local minima instead of the global minimum. This hypothesis was also confirmed by Brownlee (2016) and proves that 'higher learning rate' may not result into best mean score as seen in the case of config. 5 in table-9.8.

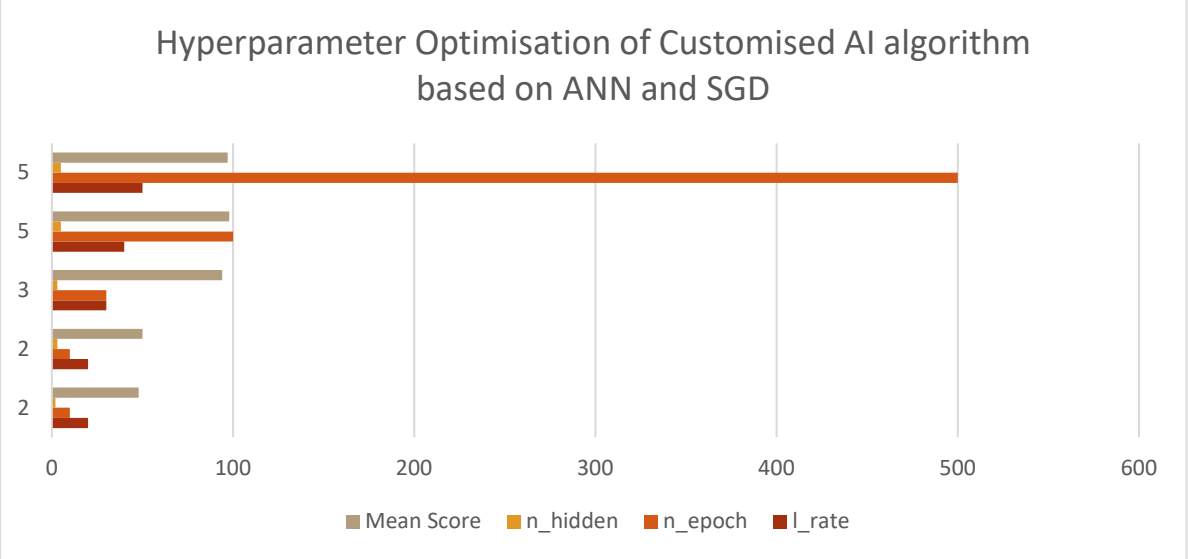


Figure 9.15: Mean scores of hyperparameter optimisation experiment using customised algorithm (ANN_SGD) implemented in DIF^2 framework

Therefore, an optimal balance has to be achieved between these hyperparameter configurations. The performance comparison of hyperparameters and mean scores of the model represented in table-9.8, are shown histographically in figure-9.15 to create a better understanding of model behaviour.

9.3.4 Experiment-4: Observations and Discussion

Experiment-4 involved evaluation of 'Customised Image Classification' algorithm based on Enhanced CNN Model. This algorithm was created and eventually enhanced as described in section-8.8.5. This experiment demonstrated the ability of the model to classify contraband images (e.g. containing child pornography) from normal images. The experiment used a dataset of 25000 images consisting of cats and dogs, where cats represented 'normal' images, while dogs represented 'contraband' images. In order to train the dataset, two separate sub-directories were created, one with cat images and another with dog images. For demonstration purpose we

took a sample dataset of 25000 images consisting of normal images represented as ‘cat’ images and contraband images (representing child pornography) as ‘dog’ images.

A publicly available images dataset from Kaggle (https://www.kaggle.com/c/dogs-vs-cats/data) was used in the experiment, which consisted of around 25000 images combined. A 75-25 % split between train and test dataset was done. The ‘train’ dataset consisted of 18697 images while ‘test’ dataset had 6303 images. The images were further segregated and arranged into subdirectories/subfolders named as cats and dogs for model training.

Prediction of Normal vs Contraband Images

When a new unseen image is presented to the ‘trained’ model, it can predict the image’s class by using classifier as ‘0’ which represents ‘normal’ image or ‘1’ as ‘contraband’ image. The output of the model is evaluated using two parameters namely “Cross Entropy Loss” and “Classification Accuracy”.

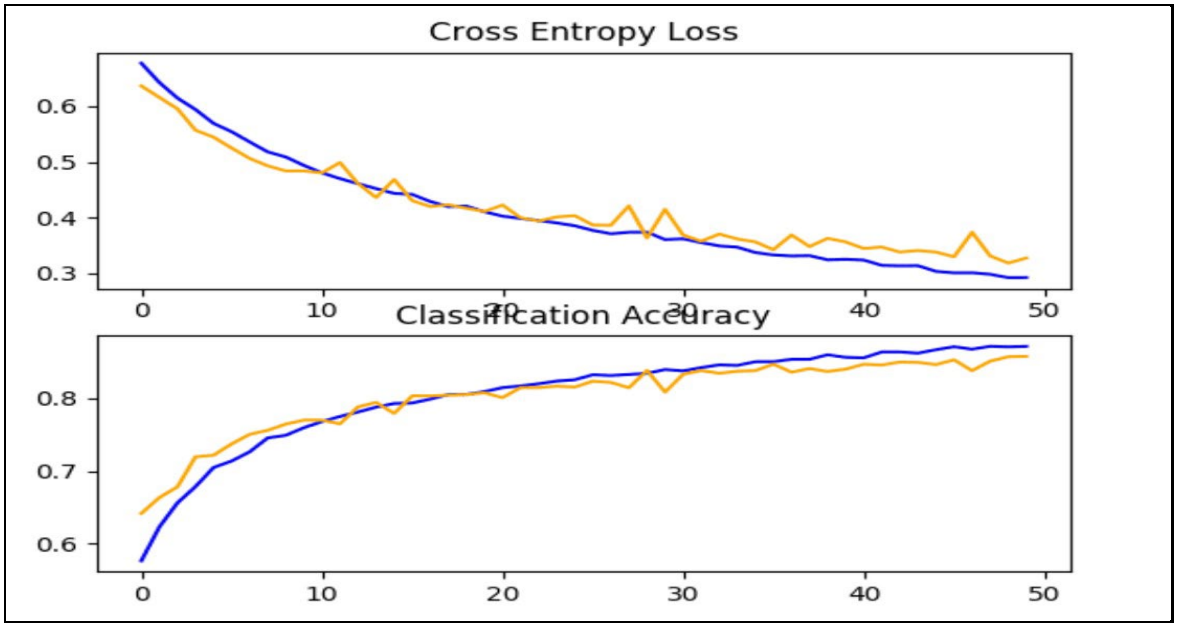


Figure 9.16: Cross Entropy Loss and Classification accuracy Results for Customised Baseline CNN Model

Output of the “Cross Entropy Loss” and “Classification Accuracy” of the customised model developed and optimised further is evaluated in this section. Executing the algorithm involves initially fitting the model, followed by assessing the model's performance on the hold-out test dataset.

It's important to note that the results usually differ due to the stochastic nature of the algorithm or evaluation procedure, as well as variations in numerical precision. To obtain a more robust understanding, the program is executed multiple times and the average outcome compared. The baseline model achieved around **80%**, while the baseline model with 'simple data augmentation' and optimisation demonstrated an enhanced performance of about **85%** as shown in figure-9.16. In this instance, an improvement in performance of approximately 5% was observed.

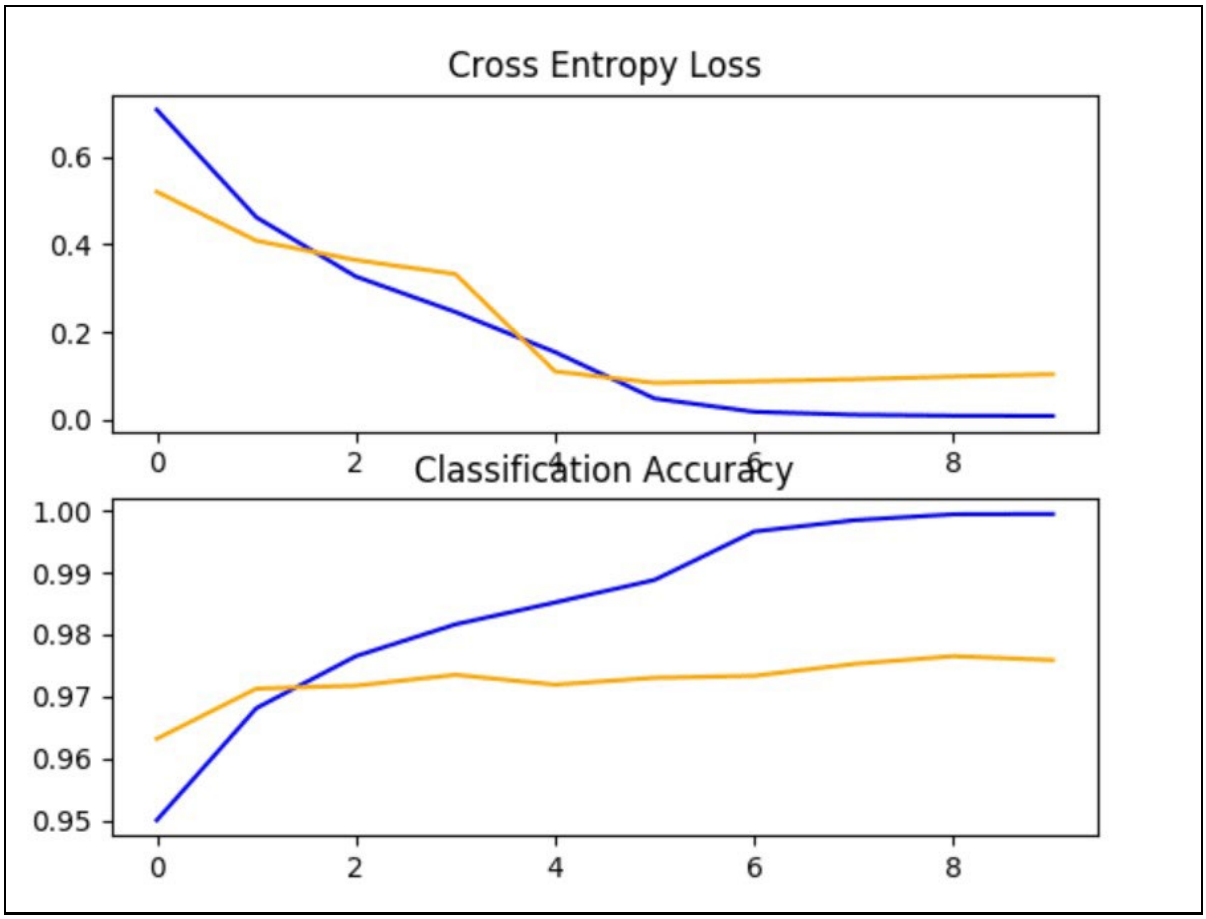


Figure 9.17: Line plot showing Entropy Loss and Accuracy for '3 block VGG Enhanced CNN' Model for Image Classification.

While the baseline model produced a score of 80% initially, with 'simple data augmentation' and optimisation in demonstrated an enhanced performance of about **85%**. While, the 'Enhanced CNN model' with '3 block VGG algorithm', produced accuracy results of **97%,** which is significantly higher compared to the best score of baseline model (85%) even after

‘simple data augmentation’ and optimisation. The output in the form line plots of ‘cross entropy loss’ and accuracy curve is presented in figure-9.16 for baseline model and figure-9.17 for ‘3 block VGG Enhanced CNN’ model respectively.

Therefore, it was observed during the evaluation of the accuracy results of ‘Baseline CNN’ model for image classification, which produced best performance of 85% even after optimisation and ‘simple data augmentation’, whereas the ‘3 block VGG Enhanced CNN Model’ produced impressive results of **97%** performance accuracy using test (cat and dog images) dataset.

These results are impressive by any means. Therefore, this experiment provides good feasibility analysis of integrating image classification features using the customised algorithm based on ‘3 block VGG Enhanced CNN Model’ into the DIF^2 framework and subsequently in the prototype that will assist cybercrime investigators to identify contraband images from the dataset.

This AI implementation can be easily integrated into newly developed prototype which is based on I-DEEP protocol as well as other DF investigation and evidence collection tools to assist first responders/investigators to accurately classify and flag images as contraband (child pornography) or normal images.

9.4 Comparing AI Algorithm Performance Parameters using Datasets- 50 vs 100 vs 300

Three new experiments (experiment-5, 6 and 7) were designed to test the algorithm performances on a bigger dataset of 300 records and also compare the performances of models using datasets with 50, 100 and 300 records. When the AI models were tested using a larger dataset (dataset with 300 records) extracted from the cybercrime database, there were noticeable improvements observed in the performance results of prediction of cybercrime incidents and evaluation of the severity status of cybercrime landscape. Experiments-5, 6 and 7 have been conducted to test the ‘dataset size’ parameter and evaluate the performance output of predictive analysis of AI Models using dataset of 300 records and also performance comparison using different dataset sizes (50 vs 100 vs 300).

9.4.1 Experiment-5: Comparing Performance Parameters of Baseline Model using Datasets-50 vs 100 vs 300

This research experiment (experiment-5) was designed to test the performance parameters using dataset of 300 records and previous experiments were also repeated to compare performances of models using datasets with 50, 100 and 300 records. The performance tests were conducted on the baseline algorithm model developed in order to establish a baseline score for evaluation of the model. This model was tested again on dataset with variable dataset size of 50, 100, 300 records to compare the performance.

Another parameter tested in this experiment was different 'train and test' splits percentages of datasets. Table-9.9 shows the comparison of results of the experiment conducted on three different sized datasets (50 vs 100 vs 300). Another parameter tested in this experiment was the dataset split into training and test datasets. This experiment was implemented by creating a test harness to implement the model.

The 'train-test' split of dataset serves as a straightforward resampling technique for assessing machine learning algorithms. Consequently, it assists as a favourable initial step in creating a test harness. It requires functions for partitioning a dataset into training and testing sets and for evaluating the accuracy of a set of predictions.

Table 9.9: Accuracy scores for Zero Rule Algorithm Baseline Model

Dataset 'Train-Test' Split	**Accuracy Scores % (Dataset- 50 Records)**	**Accuracy Scores % (Dataset- 100 Records)**	**Accuracy Scores % (Dataset- 300 Records)**
Split 50-50%	47.1	48.0	49.3
Split 60-40%	42.0	45.3	49.1
Split 70-30%	38.2	40.3	48.8
Split 80-20%	33.2	35.1	43.3
Split 90-10%	27.1	30.0	50.0

The optimal split percentage is to be determined in order to achieve better results but that does not guarantee that the model is learning from the data patterns. AI models' parameter tuning is an essential step towards achieving optimal performance. A too narrow split may result in random output than accurate predictions, while too wide split may result into in poor accuracy due to lack of proper test data which can allow model to learn from errors.

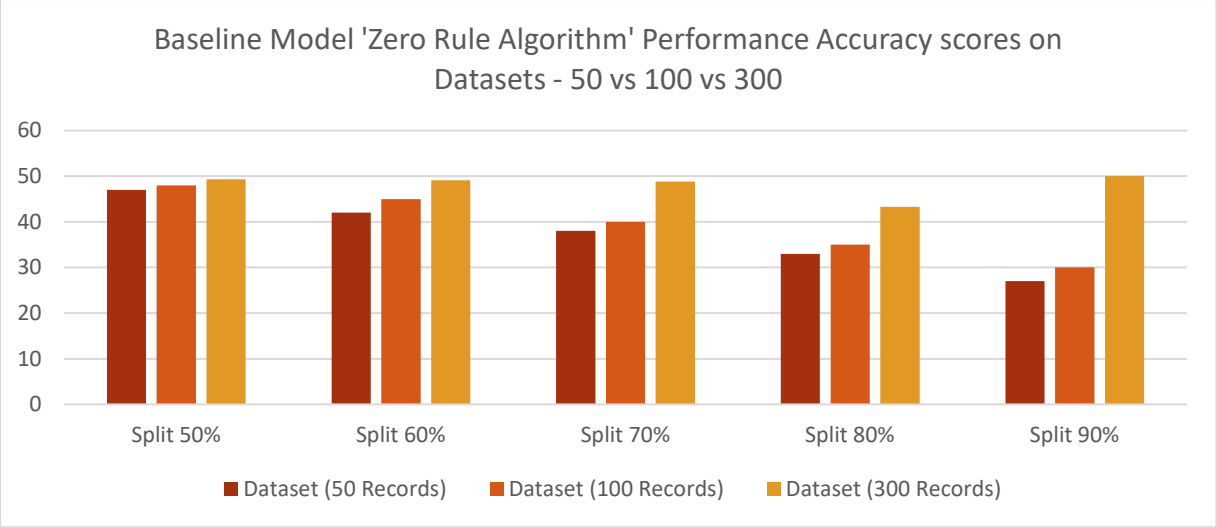


Figure 9.18: Baseline Model 'Zero Rule Algorithm' Performance Accuracy Scores

Figure-9.18 demonstrates the results via a histogram chart for Baseline model using 'Zero Rule Algorithm'. These scores provide evidence that the efficacy shows correlation with the dataset size, although the variation is not very distinct in the baseline model considering the fact that the dataset size does not remarkably differ and also the baseline model can also provide random results without using much machine learning.

On analysis of the results obtained on different split percentages, it was observed that 50-50% split produced best prediction accuracy scores. This explains the nature of the model (baseline model) that it is simply making predictions randomly and needs further training or feature enhancements. This experiment helped to establish a baseline score for other model evaluations in further experiments and also helps to learn about model behaviour. The advantage of creating the baseline model is that, it presents a formal structure to create test harness that is used to create train and test data splits and also provides a template to add more complexity to the development of forthcoming models. Another significant observation was that the dataset size also affects the performance of the model as we compared the results with three different sizes of datasets (50 vs 100 vs 300). An overall performance improvement was observed when using a dataset of 300 records. We clearly observe that dataset size impacts the performance accuracy scores as in case of 60-40% 'Train-Test' split, dataset with 50 records (dataset-50) produces score of **42**. While it produces **45.3** and **49.1** scores with datasets of 100 and 300 records respectively. A split % of train and test dataset that is too narrow (50-50%) or a split too wide (90-10%) often produces either random predictions as in the case of 50-50% split or poor accuracy if the dataset is small (50 or 100 records). On the other hand, AI models do tend to

get saturated if dataset is huge and system configuration parameters also play a role in running the models as discussed by Brownlee (2016) in his experiment.

9.4.2 Experiment 6: Comparing Performance Parameters of Spot-Check Algorithms using k-fold Method using Datasets-50 vs 100 vs 300

This experimental implementation (Experiment-6) comprised of creating a test harness to test existing AI models for predictive modelling for classification and regression problems. This involved steps like creating a test harness, loading the datasets, loading the models, spot-checking the models using k-fold validation process, and performance comparison of various AI algorithms. The existing models are spot-checked using split validation technique. This setup tested the performances using three datasets (dataset of 50 vs 100 vs 300 records) and the results were compared.

The Results provided in Table-9.10 show mean values of Cross Validation (CV) scores and standard deviation for spot-checked algorithm using 50 records. This indicates the performance of the models. These results provide a more reliable estimate of a model's performance compared to a single train-test split, offering insights into its generalization ability and stability across different subsets of the data.

Table 9.10: Spot-check Algorithm Validation Test Results using a dataset of 50 records

Model Name	Mean Value	Standard Deviation
Logistic Regression (LR)	0.625000	0.201556
Linear Discriminant Analysis (LDA)	0.525000	0.207666
K-NeighborsClassifier (KNN)	0.625000	0.201556
Decision Tree Classifier (CART)	0.300000	0.244949
GaussianNB (NB)	0.525000	0.261008
SVM	0.475000	0.134629

Results of the Spot-check algorithm validation process on a dataset with 100 records is provided in table-9.11, which consists of mean value scores (CV results) and standard deviation. While table-9.12 shows Spot-check algorithm scores of best performing models in experiment conducted with dataset of 300 records.

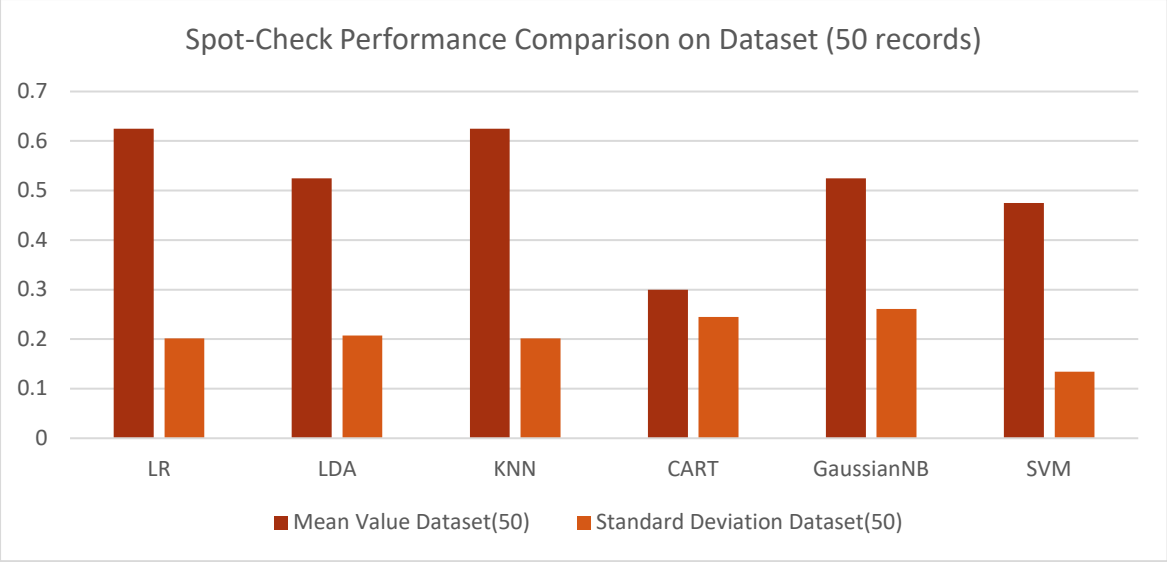


Figure 9.19: Spot Check Comparison of Algorithms Scores on Dataset (50 records)

The Mean value scores and standard deviations of spot checking of different AI algorithms on a dataset of 50 records are represented in figure-9.19. These parameters provide us with an indication of which algorithms are more suitable for predictive analysis of a given dataset. We observe that K-NeighborsClassifier (KNN) and Logistic Regression (LR) both, provide an equal performance in terms of mean value of predictive accuracy scores of 62.5% as also represented it the table-9.10 and also shown in code snippet-9.1 output. Both algorithms show a standard deviation of 20% which again represents a high variation. Table-9.11 represents the experiment results conducted on a dataset of cybercrime with 100 records. We observe performance improvements in the results with Linear Discriminant Analysis (LDA) proving to be the best performing algorithm with a mean accuracy score of **67.5%** and lower standard deviation of 17.8%, which demonstrates better performance and shows that dataset size can be an important parameter of performance metrics for algorithm comparison.

Table 9.11: Spot-check Algorithm Validation Test Results using a dataset of 100 records

Model Name	**Mean Value**	**Standard Deviation**
Logistic Regression (LR)	0.637500	0.171847
Linear Discriminant Analysis (LDA)	0.675000	0.178536
K-NeighborsClassifier (KNN)	0.600000	0.145774
Decision Tree Classifier (CART)	0.525000	0.165831
GaussianNB (NB)	0.612500	0.152582
SVM	0.625000	0.136931

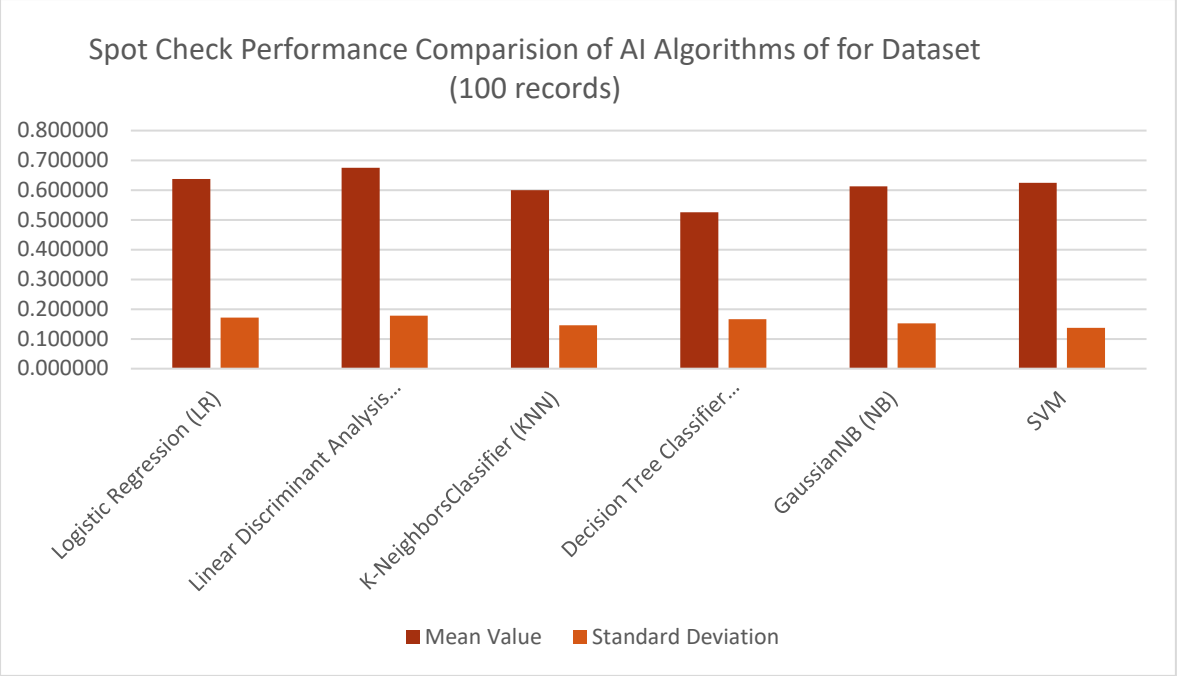


Figure 9.20: Mean Accuracy Scores of Spot Check of AI Algorithms (100 records)

The results of experiment conducted with dataset of 100 records and represented in table-9.11 are shown as histographic chart in figure 9.20. The chart shows mean accuracy scores comparison of algorithms and the standard deviations, which shows LDA as best performing algorithm with a mean accuracy score of **67.5%**.

We extended this experiment further with 300 records to compare the performances of participating algorithms and the results are shown in table-9.12. In this experiment, which is depicted in code snippet-9.2, the spot-check comparison of best performing algorithm is performaed again with a dataset of 300 records. This time we observe quite significant improvement in accuracy results in all models, while Decision Tree Classifier (CART) providing best performance with mean value scores of **94.5%** and a standard deviation of 3%. This experiment showed high efficiency of the CART algorithm on classification problems like in cybercrime incidents ‘critical’ and ‘non-critical’ classification.

```
url = "https://raw.githubusercontent.com/deep0025/CyberCrimeData-SA/main/CybercrimeDataset_300.csv"
names = ['Phishing', 'ZeroDayAttack', 'RansomwareAttack', 'ChildPorn', 'City']
dataset = read_csv(url, names=names)
# Split-out validation dataset
array = dataset.values
X = array[:,0:4]
y = array[:,4]
X_train, X_validation, Y_train, Y_validation = train_test_split(X, y, test_size=0.20, random_state=1, shuffle=True)
# Spot Check Algorithms
models = []
models.append(('LR', LogisticRegression(solver='liblinear', multi_class='ovr')))
models.append(('LDA', LinearDiscriminantAnalysis()))
models.append(('KNN', KNeighborsClassifier()))
models.append(('CART', DecisionTreeClassifier()))
models.append(('NB', GaussianNB()))
models.append(('SVM', SVC(gamma='auto')))
# evaluate each model in turn
results = []
names = []
for name, model in models:
    kfold = StratifiedKFold(n_splits=10, random_state=1, shuffle=True)
    cv_results = cross_val_score(model, X_train, Y_train, cv=kfold, scoring='accuracy')
    results.append(cv_results)
    names.append(name)
    print('%s: %f (%f)' % (name, cv_results.mean(), cv_results.std()))
# Compare Algorithms
pyplot.boxplot(results, labels=names)
pyplot.title('Algorithm Comparison')
pyplot.show()
```

```
LR: 0.654167 (0.074652)
LDA: 0.695833 (0.052869)
KNN: 0.641667 (0.079495)
CART: 0.945833 (0.032543)
NB: 0.691667 (0.077280)
SVM: 0.804167 (0.067315)
```

Code Snippet 9.2: Performance Analysis of Spot-check AI algorithm using dataset-300 (Experiment-6 Output Results)

Therefore, we can conclude from experiment-6 results that the AI algorithms' performance using a cybercrime dataset with 300 records showed a much improved performance overall in all algorithms in 'Spot-check' comparison test. The best performance was demonstrated by CART algorithm showing **94.5%** accuarcy in predictive analysis in cross validation results as also dispalyed in experiment-6 output, shown in code snippet-9.2 (program output) and also results displayed in table-9.12. Figure-9.23 represents the Performance Results of Spot Check of AI Algorithms with dataset comprising of 300 records.

Table 9.12: Performance Results for Spot-check Algorithm using dataset of 300 records

Model Name	**Mean Value**	**Standard Deviation**
Logistic Regression (LR)	0.654167	0.074652
Linear Discriminant Analysis (LDA)	0.695833	0.052869
K-NeighborsClassifier (KNN)	0.641667	0.079495
Decision Tree Classifier (CART)	0.945833	0.032543
GaussianNB (NB)	0.691667	0.077280
SVM	0.804167	0.067315

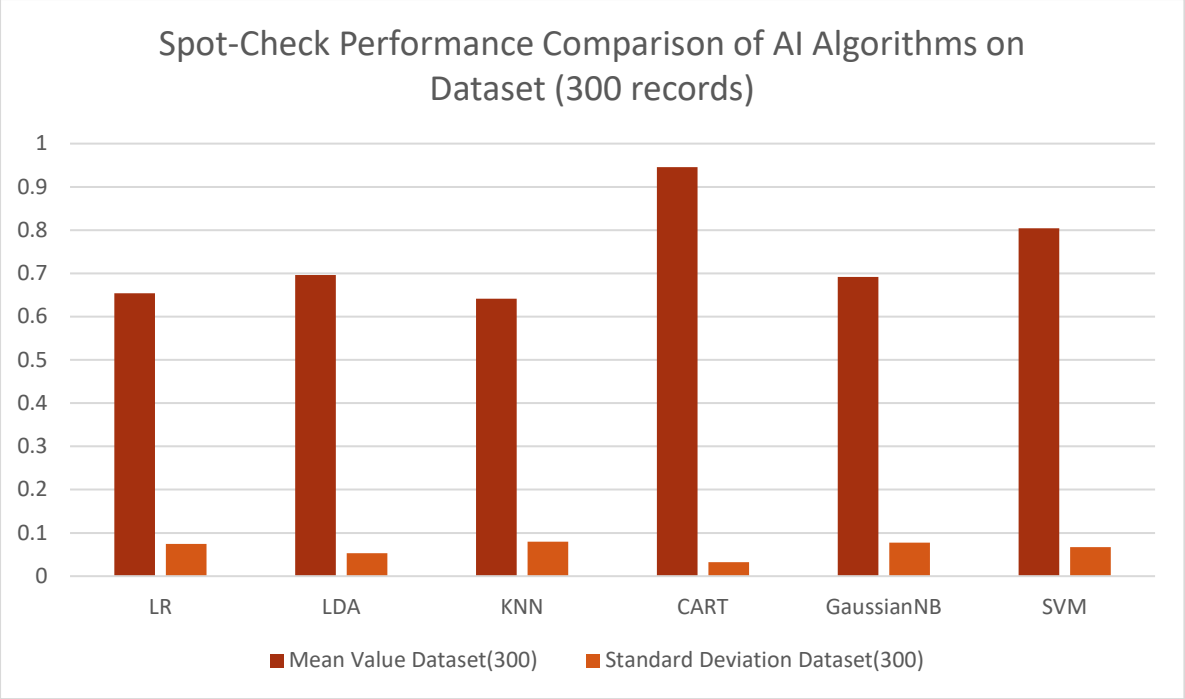


Figure 9.21: Performance Results of Spot Check of AI Algorithms (300 records)

The performance results of experiment-6 with dataset of 300 records are represented as histographic chart in figure-9.21, which CART as the best performing algorithm.

Figure-9.22 shows a histographic comparision of results of performance of spot-check algorithms of three datasets with variable records (50 vs 100 vs 300) and a discussion on the results is provided henceforth. The description of this experimental setup has been provided in section-5.7 and its implementation has been discussed in section-7.7 in detail. Therefore, the discussion is limited to performance results only in forthcoming sections.

Analysis and Discussion

We observe that in the case of dataset with 50 records, the best performing algorithm models were LR and KNN. This has to be mentioned that this dataset had three classes (city) and a split may had only few instances of a perticular class considering the small dataset size. We observed that best performing algorithm models were LR and KNN in 50 records experimental setup with overall 'Cross Validation (CV) Score' of **62.50%** and standard deviation of 20%. CART and NB both performed poorly in this experiment as shown in table-9.10.

On the other hand, with a dataset of 100 records, Linear Discriminant Analysis (LDA) had the highest accuracy mean value scores (CV Scores). Table-9.11 represents the experiment results conducted on a dataset of cybercrime with 100 records. We observed performance

improvements in the results with Linear Discriminant Analysis (LDA) proving to be best performing algorithm with a mean accuracy score of **67.5%** and lower standard deviation of 17.8%, which demonstrated better performance and showed that dataset size can be an important parameter of performance metrics for algorithm comparison. SVM and LR also showed relatively competitive accuracy scores.

It is essential to interpret these results in the context of the specific problem being solved and the characteristics of the dataset. Additionally, further analysis, such as examining precision, recall, and f1-score, could provide a more comprehensive understanding of model performance, especially if there is class imbalance.

On extending this experiment further with 300 records to compare the performances of participating algorithms and the results are shown in table-9.12, we see significant gain in performance of predictive analysis of all algorithms. In this experiment, we observed quite significant improvement in accuracy with Decision Tree Classifier (CART) providing mean value scores of **94.5%** and a standard deviation of only 3%, which shows high efficiency of the algorithm on classification problems. Figure-9.21 represents the scores in histographic chart comparisons of algorithms using a dataset of 300 records.

Standard deviation provides a measure of how much the accuracy scores vary across different runs or folds in cross-validation. Lower standard deviations suggest more consistent performance. LDA also produced a low standard deviation of 5%, but accuracy was not good (69.5%). While Decision Tree (CART) showed the lowest standard deviation and highest accuracy.

It was also noted that standard deviation of results of 100 records experiment is lower than 50 records. This reflects well on experiment (Dataset-100) as compared to (Dataset-50) and also shows better performance. Therefore number of records in a dataset can improve the performance of the model and show positive correlation. On performing this experiment on dataset of 300 records, the standard deviation is further reduced to only 3% and performance accuracy rose to **94.5%**.

This also proves that the performance has a positive coorelation with the dataset size. Although a stage arrives when model might get overwhelmed by a huge dataset and therefore may not provide best performance as discussed by Brownlee (2016) and Johnson and Ananthakumaran (2021) in their observations.

These scores show consisitent performance improvements when it comes to predictive analysis and implementations carried out in various stages of experiment as described under 'Research Experiment' in section-5.7.

Our objective was to strive to achieve better performance systematically by more advanced AI implementation and by further developing customised models, which was met. Output of this experiment produced results with varying attributes on different datasets sizes (50 vs 100 vs 300).

This experimental setup compared performance data produced by analysis of the different dataset sizes, also presented in tables-9.10, 9.11 and 9.12. The experiment conducted with 50 records demonstrated that K-NeighborsClassifier (KNN) and Logistic Regression (LR) both provided an equal performance in terms of mean value of predictive accuracy scores of **62.5%** as presented in the table 9.10 and also shown in code snippet 9.1 output.

Both algorithms show a standard deviation of 20% which again represents a high variation. When this experiment was conducted with 100 records, it demonstrated that the best performing algorithm on the given dataset was Linear Discriminant Analysis (LDA) has highest accuracy mean value of **67.50 %.**

For dataset of 300 records, it was CART (Decision Tree Classifier) with accuracy of **94.5%** and standard deviation of only 3%, which demonstrated best performance. Figure-9.21 shows performance results of dataset of 300 records along with standard deviation, which is highest of all models achieved in all previous experiments in 'spot check comparison' of algorithms. It also shows the lowest standard deviation of only 3%.

Therefore, dataset size can be critical in making effective predictions on the cybercrime dataset where dataset of 300 records showed maximum performance using CART algorithm. Low standard deviation shows that the model is consistent and the accuracy scores obtained across different runs or folds in cross-validation have a low variance.

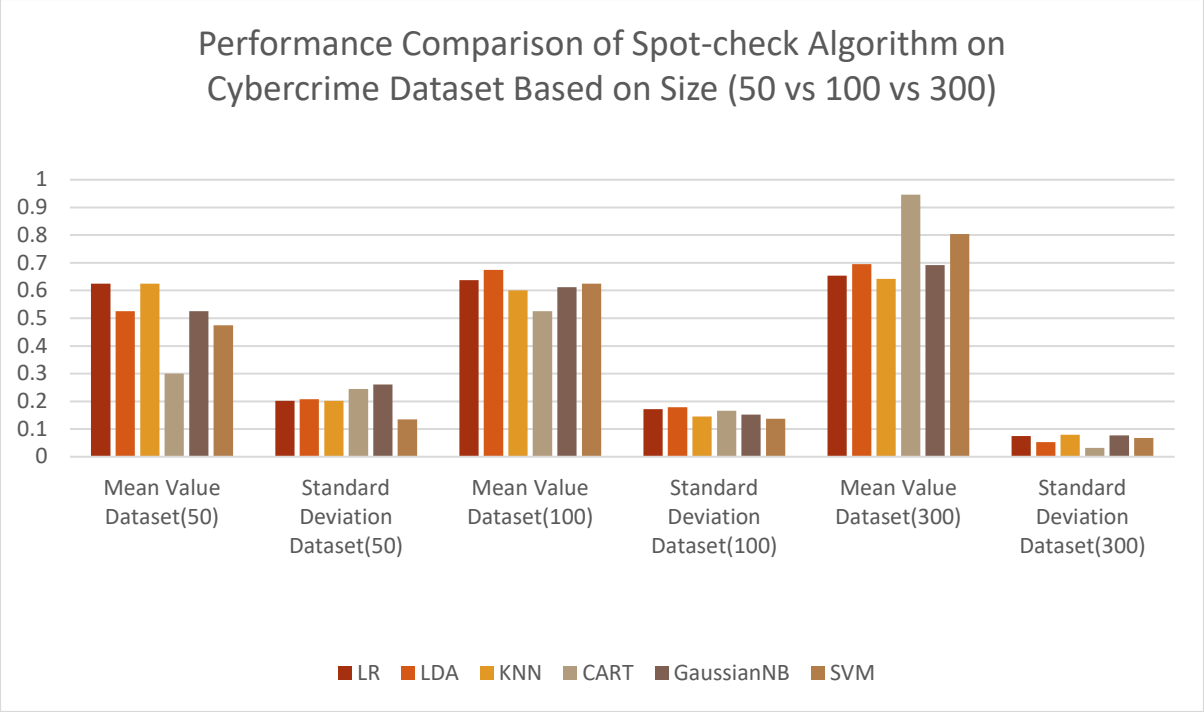


Figure 9.22: Histographic comparison of performance of spot-check algorithm (Dataset- 50 vs 100 vs 300 records)

When running this experiment on a dataset of 300 records, the results show further improvements as displayed in table 9.13. It is observed that Decision Tree Classifier (CART) is the best performing algorithm with a high accuracy of 94.5% and a low standard deviation of 3% shows high consistency of the model. Figure-9.22 shows histographic comparisons of the performance of spot-check algorithm three dataset configurations (50, 100, 300).

There is a clear visible positive correlation observed between dataset size and model performance where the performance increased with dataset size. Figure-9.23 represents the spot check performance results of the algorithms as 'box and whiskers plot'. While CART is clearly showing an outstanding performance, the LR and KNN algorithms show poor performance with the dataset-300 experiment (experiment-6). We also observe that NB (Gaussian NB) and SVM (Support Vector Machine) algorithm show lots of outliers in the 'box and whisker plot' representations. Each algorithm shows different predictive accuracy depending on characteristics of datasets and the problem at hand (regression or classification). Our problem was a multi-classification problem where cybercrime incidents were recorded across three South African cities.

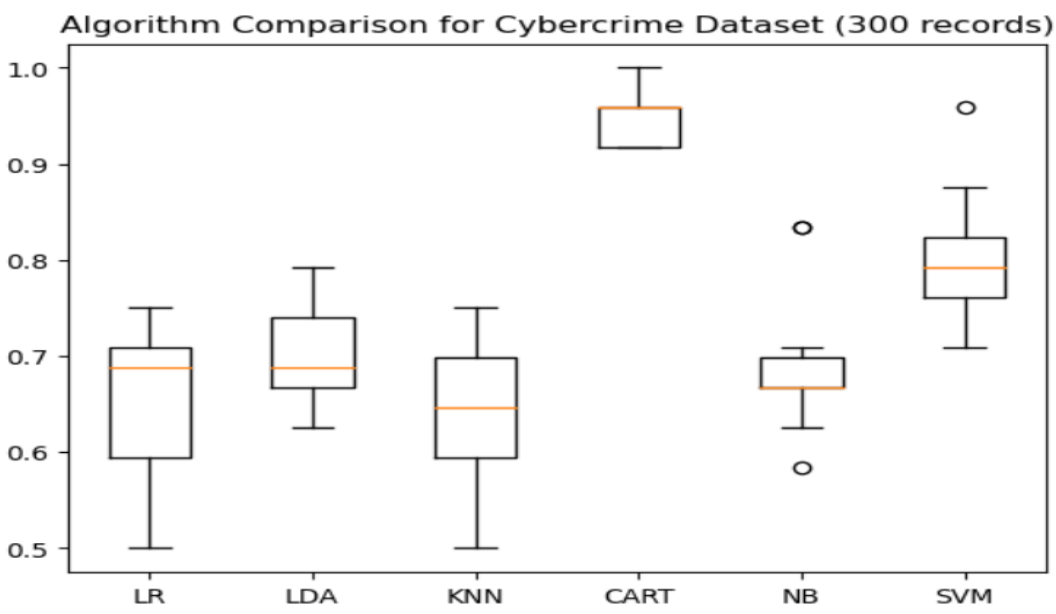


Figure 9.23: Box and Whisker Plot of Spot Check Performance of Algorithms on Dataset-300

Comparing Prediction Parameters of Models using Dataset 50 vs 100 vs 300

Next step in the experiment involves comparing other parameters of the best performing models with experimental conditions using three datasets (50 vs 100 vs 300). Code snippets 9.3, 9.4 and 9.5, show model implementation and comparison of 'prediction parameters' in python language and program output, that demonstrates model fitting and prediction. Model fitting uses two datasets X_train and Y_train. The 'Test and Train split' is 80% and 20%. The algorithms encoded in code snippets demonstrate the process of training the best performing model on the variable datasets (50, 100 and 300) and then using trained model to make prediction on validation dataset. 'X_train' consists of the feature values, while 'Y_train' contains the target values of the training set.

The model uses learned patterns from the training data to predict target values for the validation data. The experiment evaluates the performance of the predictions made by the trained model using three different metrics- 'accuracy, confusion matrix and classification report'. Accuracy is the ratio of correctly predicted instances to the total instances in the dataset. It provides a measure of how often the prediction is correct. Confusion matrix provides a summary of prediction results on a classification problem like ours. The number of correct and incorrect predictions are summarised with count values and listed for each class. It shows the count of true positive, true negative, false positive and false negative predictions. The classification

report is used to generate a detailed report showing the ‘precision, recall, f1-score, and support’ metrics for each class which is ‘City’ in this case.

```
In [18]:
print(dataset.shape)

(50, 5)

In [19]:
# Make predictions on validation dataset
model = LogisticRegression(solver='liblinear', multi_class='ovr')
model.fit(X_train, Y_train)
predictions = model.predict(X_validation)

In [20]:
# Evaluate predictions
print(accuracy_score(Y_validation, predictions))
print(confusion_matrix(Y_validation, predictions))
print(classification_report(Y_validation, predictions))

0.6
[[0 0 3]
 [0 1 1]
 [0 0 5]]
              precision    recall  f1-score   support

    Capetown       0.00      0.00      0.00         3
      Durban       1.00      0.50      0.67         2
johannesburg       0.56      1.00      0.71         5

    accuracy                           0.60        10
   macro avg       0.52      0.50      0.46        10
weighted avg       0.48      0.60      0.49        10
```

Code Snippet 9.3: Performance Parameter Comparison of LR (Best performing Algorithm) on Dataset-50 experiment

While running prediction validation separately on the best performing model (LR) for Dataset-50 experiment, we see a precision value=0, recall value =0 and f1-score also as ‘0’ and a low support (support=3) for class value ‘Capetown’.

Precision, recall, f1-score, and support are commonly used metrics to evaluate the performance of an AI algorithm on mainly classification problems, including other AI prediction algorithms. These metrics provide insights into different aspects of the model's predictive ability.

Precision is the ratio of true positive predictions to the total number of positive predictions made by the model. Precision focuses on the accuracy of positive predictions. A high precision indicates that when the model predicts a positive outcome, it is likely to be correct.

Recall is the ratio of true positive predictions to the total number of actual positive instances in the dataset. Recall measures the model's ability to capture all the positive instances in the dataset. A high recall indicates that the model can identify a large proportion of the actual positive instances.

The f1-score is the harmonic mean of precision and recall. It provides a balance between precision and recall. The f1-score is particularly useful when there is an uneven class distribution. It is a single metric that considers both false positives and false negatives. A higher f1-score indicates a better balance between precision and recall.

Support is the number of actual occurrences of the class in the specified dataset. Support provides context by indicating how often the class occurs in the dataset. It helps in understanding the significance of precision, recall, and f1-score in the context of the dataset.

There is a major possibility in small datasets that sample size of train and test data may not contain enough representative values as seen in the case of 'Capetown' class. This impacts the overall performance of algorithm and shows that prediction accuracy is low as in the case of Dataset-50 experiment.

```
[13]:
# Make predictions on validation dataset
model = LogisticRegression(solver='liblinear', multi_class='ovr')
model.fit(X_train, Y_train)
predictions = model.predict(X_validation)

[14]:
# Evaluate predictions
print(accuracy_score(Y_validation, predictions))
print(confusion_matrix(Y_validation, predictions))
print(classification_report(Y_validation, predictions))

0.65
[[5 1 3]
 [2 1 1]
 [0 0 7]]
              precision    recall  f1-score   support

    Capetown       0.71      0.56      0.63         9
      Durban       0.50      0.25      0.33         4
johannesburg       0.64      1.00      0.78         7

    accuracy                           0.65        20
   macro avg       0.62      0.60      0.58        20
weighted avg       0.64      0.65      0.62        20
```

Code Snippet 9.4: Performance Parameters Comparison of LDA (Best performing Algorithm) on Dataset-100 experiment

On running the experiment with a dataset of 100 records, we observe that the accuracy score improves significantly and an overall score of 65% is observed. A Precision value of 71% indicates that model has predicted a positive outcome that is likely to be correct. Code Snippet 9.4 shows the results and output of the experiment.

Code Snippet 9.5: Performance Comparison of Best Performing Algorithm (CART) on Dataset-300 experiment.

When the experiment is conducted with a dataset of 300 records, a further increase in accuracy score of 85% is observed. The best performing AI model is CART (Decision Tree Classifier). There is a significant increase in all parameters such as precision (75%), Recall (90%) and Support of 20% is observed in case of 'Capetown' class for example. A weighted average of 86% for precision, 85% for recall and a f-1 score of 85% is observed, which provide an insight into the balance between precision and recall values and identifies a number of actual occurrences of a class in a dataset. The results of prediction are shown in code snippet 9.5. An overall accuracy score of 85% clearly indicates that a bigger dataset size can increase the prediction accuracy of the model.

On comparing the performance of models and 'prediction parameters' using different dataset sizes (50 vs 100 vs 300), we see a significant difference in results and therefore compared these parameters in figure-9.24. Here we can correlate why the performance differs when using different dataset sizes (50 vs 100 vs 300). If a dataset size is increased further, it is hypnotised that it will provide more accuracy and better performance. But due to time and resource constraints that is not possible. However, this comparison is sufficient to prove the hypothesis of correlation between dataset size and prediction accuracy. It is observed that all the parameters such as: Accuracy Score, Precision, Recall, f1-Score, and Support, show significant improvements and therefore show better efficacy. These parameters are indicators of high positive rate of prediction, a correlation between false positives and false negative predictions and also show the actual occurrences of a class in the dataset.

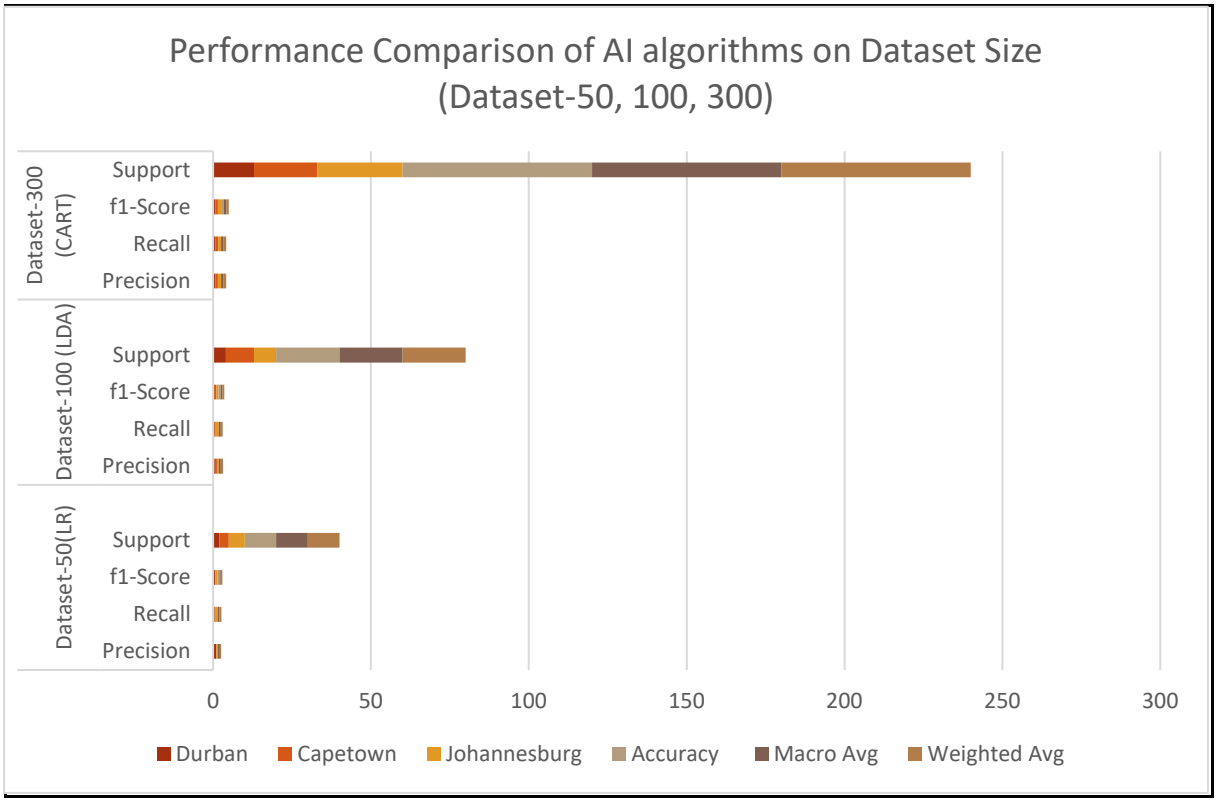


Figure 9.24: Histographic Representation of Prediction Performance Parameters of Algorithms (LR, LDA, CART) on datasets (50 vs 100 vs 300)

Accuracy and Dataset Size

In order to establish the correlation between the size of dataset and the accuracy of models, an analysis of a dataset with 50 records, 100 records and 300 records was done and results were compared. A positive correlation was identified in the performance of models with the size of dataset. Code snippets – 9.3. 9.4 and 9.5 represent the actual implementation and performance parameters (Accuracy Score, Precision, Recall, f1-Score, and Support) in terms of 'program output' and results of the experiment. Figure-9.24 illustrates the relationship between the performance parameters, dataset size and performance accuracy of experiments conducted on dataset size of 50, 100 and 300 records. A dataset with bigger size produced better accuracy results and displayed overall better prediction parameters using best performing algorithm.

9.4.3 Experiment 7: Comparing Performance Parameters of ANN_SGD Algorithm using Dataset-300

When The ANN_SGD Algorithm was tested using dataset of 300 records, there was only a slight improvement observed in the performance results as shown in the code snippet as well as program output in experiment-7. When we compared this performance with datasets-50 and

dataset-100 records, as covered in experiment-3 and discussed in section-9.3.3, which produced 'mean accuracy' score of **94%** for a dataset of records-50 and a score of **97%** with dataset-100, as demonstrated in program output shown in figure-B.12 (APPENDIX-B). A score of **98.33%** was achieved during the experimental testing of dataset with 300 records.

This experimental comparison kept parameters settings consistent to previous experiments (experiment-3) i.e – number of folds (n_folds) = 5; learning rate = 30%; number of epoch (n_epoch) = 500 and number of hidden layers (n_hidden) = 5. This was done in order to specifically focus on dataset size parameter and observe the performance difference.

This can be explained by the fact that 'Custom_ANN_SGD' algorithm is designed specifically to perform efficiently with smaller datasets as the 'case based' scenario we have with first responders, who probably do not have or need access to huge datasets while they use the prototype. Therefore, they will use smaller datasets generated during investigations.

DK_ModelTrain300 Last Checkpoint: 05/12/2024 (autosaved)

View Insert Cell Kernel Widgets Help Not Trusted

Run Code

```
168
169 # Backpropagation Algorithm With Stochastic Gradient Descent
170 def back_propagation(train, test, l_rate, n_epoch, n_hidden):
171     n_inputs = len(train[0]) - 1
172     n_outputs = len(set([row[-1] for row in train]))
173     network = initialize_network(n_inputs, n_hidden, n_outputs)
174     train_network(network, train, l_rate, n_epoch, n_outputs)
175     predictions = list()
176     for row in test:
177         prediction = predict(network, row)
178         predictions.append(prediction)
179     return(predictions)
180
181 # Test Backprop on Cybercrime DataSet
182 seed(1)
183 # load and prepare data
184 filename = 'CybercrimeDataset_SeverityScore_Class300.csv'
185 dataset = load_csv(filename)
186 for i in range(len(dataset[0])-1):
187     str_column_to_float(dataset, i)
188 # convert class column to integers
189 str_column_to_int(dataset, len(dataset[0])-1)
190 # normalize input variables
191 minmax = dataset_minmax(dataset)
192 normalize_dataset(dataset, minmax)
193 # evaluate algorithm
194 n_folds = 5
195 l_rate = 0.3
196 n_epoch =500
197 n_hidden = 5
198 scores = evaluate_algorithm(dataset, back_propagation, n_folds, l_rate, n_epoch, n_hidden)
199 print('Scores: %s' % scores)
200 print('Mean Accuracy: %.3f%%' % (sum(scores)/float(len(scores))))
```

```
Scores: [100.0, 96.66666666666667, 98.33333333333333, 98.33333333333333, 98.33333333333333]
Mean Accuracy: 98.333%
```

Code Snippet 9.6: ANN-SGD Algorithm Performance Scores on Dataset-300 (Experiment-7)

When we compared this performance with earlier experiments conducted with datasets-50 and dataset-100 records, as covered in experiment-3 discussed in section-9.3.3, with similar parameters settings (number of folds (n_folds) = 5; learning rate = 30%; number of epoch (n_epoch) = 500; number of hidden layers (n_hidden) = 5), there was a slight improvement observed in the prediction performance.

Experiment-7 was further extended to analyse model's performance with a bigger dataset (300 records) and different hyperparameter configurations were tested. The output results of the experiment conducted with different hyperparameter configurations that involved multiple iterations of the experiment and the mean scores produced are shown in table-9.13.

Table 9.13: Hyperparameter optimisation scores of algorithm ANN_SGD using Dataset-300

Configuration	n_folds	l_rate	n_epoch	n_hidden	Mean Score
1.	2	20	10	2	48.21
2.	2	20	10	3	50.13
3.	3	30	30	3	94.12
4.	5	40	100	5	98.33
5.	5	50	500	5	97.22

Configuration testing was done using different variations of the parameters as shown in table-9.13. We initially observed a slight increase in performance. This encouraged us to perform more diversified hyperparameter optimisation testing of the model. Therefore, we also tried different variations of 'l_rate' parameters as shown in Table-9.13. We saw that config-4 produced best results (98.33% overall mean score).

The best mean score was observed for experimental setting in config.4 (n_folds=5, n_rate=40%, n_epoch=100, n_hidden=5) during hypermarameter optimisation experiment conducted on newly developed customised algorithm model based on ANN and SGD ensemble and using dataset of 300 records. We also notice that lowest score was observed in config.1 setting (n_folds=2, l_rate=20%, n_epoch=10, n_hidden=2). This shows that low number of iterations and lower number of hidden layers do not provide enough learning and error correction chances using backpropagation method used in model. Further comparing the results in table-13, we observe that a higher value for 'number of epochs' (n_epoch=500) and 'learning rate' (l_rate=50%) parameters do not result into better performance as both configurations produced mean score of 97% which was lower than the best score (config.4).

On the contrary, in later iterations, it was observed that increase in no of epoch after a time does not result in improved scores. This analysis confirmed again that a higher learning rate (l_rate=50%) and number of epoch (n_epoch=500) may not result into best mean scores as seen in config. 5 in table-9.13. Code Snippet-9.6 shows the experiment implementation and program output. Results represented in table-9.13 are also represented histographically in

figure-9.25 to provide a comprehensive relationship between different hyperparameters and overall performance by the model.

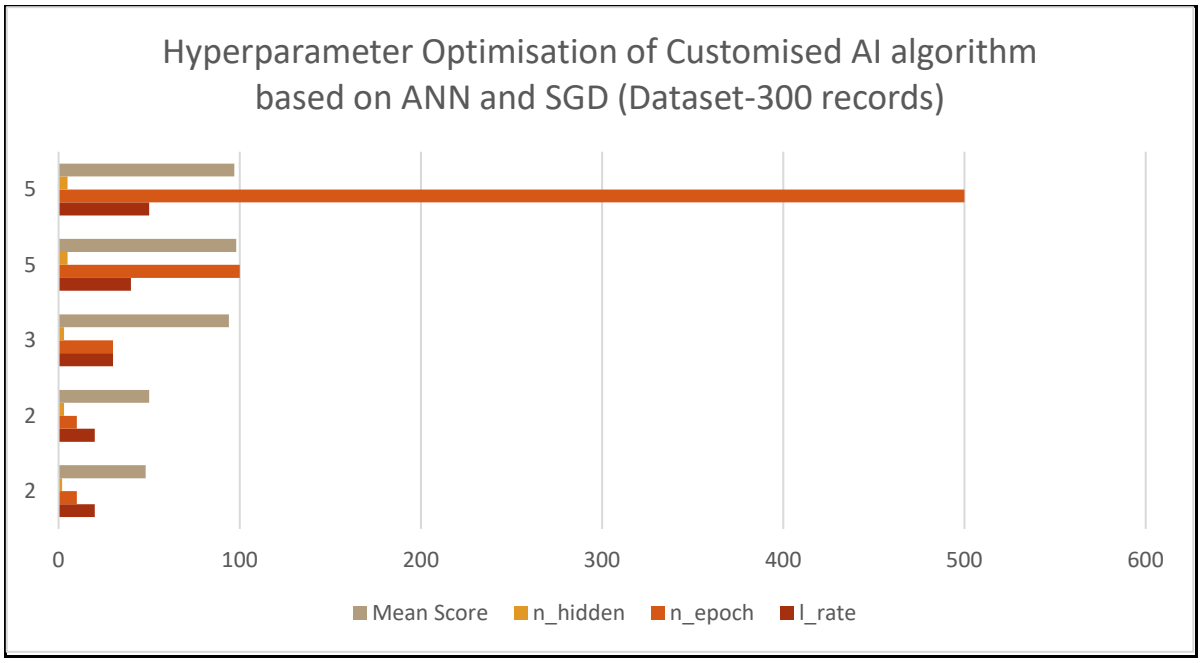


Figure 9.25: Mean scores of hyperparameter optimisation experiment on Custom_ANN_SGD algorithm implemented in DIF^2 framework using dataset-300.

Earlier this experiment was carried out on a dataset of 100 records. On running the same experiment on 300 records a slight improvement of 1.33% was observed (mean accuracy score of 98.33%) as shown in shown in Code Snippet-9.6 (program output).

This behaviour can be explained by the fact that ANN based algorithms are generally known to be very efficient on predictive analysis involving a smaller dataset size also. Unless there is a very big dataset size difference, the performance improvement is not expected to be very substantial. This is not the case with datasets (dataset 50 vs 100 vs 300) compared in experiments conducted (experiment-3 and experiment-7).

9.5 Overall Analysis

While we see that performance improved from the baseline model (experiment-1, stage-1) to other implementations like AI algorithms spot-check method (experiment-1, stage-2), to the implementation of AutoML-TPOT model (experiment-2). We observed major performance gain in implementation of a customised 'Custom_ANN_SGD' ensemble model (experiment-3) in predictive analysis of both regression and classification problems. Most models demonstrated a major gain in prediction performance with dataset size increase (50 to 100 to

300 record datasets) used for extended experiments (experiments-5, 6, and 7), and hyperparameter optimisation, except the Custom_ANN_SGD Model, which was a customised model created for both predictive analysis on regression problems (i.e. prediction of cybercrime incidents and 'severity scores' for three major South African cities) and classification problems (prediction of 'critical' or 'non-critical' status of cybercrime incidents used for triaging). This model (Custom_ANN_SGD) showed a small performance improvement when comparing between dataset 100 to 300 records). This behaviour can be attributed to the fact that the model was custom created to perform better on smaller datasets as in the 'case-based scenario' of its implementation in AI based intelligent framework (DIF^2) using the prototype designed for first responders.

The 'Customised CNN' Model used for image analysis (Experiment-4) also showed improvements in performance in parameters 'cross entropy loss' and 'classification accuracy from **80%** in the baseline model to **85%** in optimised model. Further with '3 block VGG' enhancements, the 'Enhanced CNN 3 VGG' model was able to achieve **97%** performance. This experiment used a dataset of 25000 images consisting of cats and dogs, where cats represented 'normal' images, while dogs represented 'contraband' images. In order to train the dataset, two separate sub-directories were created, one with cats' images and another with dog images.

Therefore, we conclude that the hyperparameters optimisation can improve the algorithm performance and dataset size plays an important role in algorithm performance as observed in all the experiments conducted with an exception of ANN_SGD algorithm as demonstrated in experiment-7. This is due to the nature of ANN_SGD which shows high performance on smaller dataset sizes also. This makes this algorithm a highly suitable for predictive analysis in 'classification' and 'regression problems' as well.

Finally, the experiments provided evidence towards demonstration of 'intelligent framework' that supports optimisation and extension using a systematic, 'iterative and incremental' approach to integrate functional capabilities and also contributed in demonstrating the 'automation and optimisation' of framework using more advanced (AutoML-TPOT) based pipelines can help in designing more sophisticated models. The integration approach uses a 'gearbox model' approach implemented in I-DEEP and DIF^2 framework, where the best performing models can be integrated, selected and adopted by first responders using the prototype.

9.6 Summary

This chapter discussed the results and evaluated the performance output of four stage experimental setup (experiments 1 to 4) initially, which was further extended for performance comparison of AI models using datasets with 50, 100 and 300 records (experiments-5 to 7). The experiments were conducted to evaluate the performance of all newly created artifacts (algorithm models) and their implementations into DIF^2 Framework. The results demonstrated that use AI and the newly developed framework can improve prediction analysis of cybercrime incidents, severity analysis, classification of cybercrime status (critical or non-critical) and can help in triaging, thus overall assisting DF first responders and investigators in quick decision making. This evaluation also provided the valuable observations on different AI models demonstrated applicability of AI automation and optimisation using pipelines and use of AutoML and TPOT automated frameworks. This experimental evaluation also provided a motivation for AI integration into I-DEEP protocol and DIF^2 framework and also demonstrated the feasibility of implementation of triaging and predictive analysis of cybercrime incidents for better decision making. It also demonstrated a stepwise development of DIF^2 framework and its evaluation towards achieving better prediction modelling using cutting edge technology, automation frameworks and pipelines and testing of various artifacts that were developed in the due course of this research. The evaluation found best results achieved by the newly developed customised algorithm model based on ANN and SGD algorithms. Although other models (baseline model, Manual Spot-check and Selection Algorithm and AutoML based automation pipelines also showed their own utility as base model or served as a template to develop more complex models. AutoML and TPOT based 'Automation Framework' showed promising benefits towards automating the data analysis and prediction with bigger datasets and more diverse problems. 'Enhanced 3 VGG CNN' based algorithm also showed high efficacy in image analysis. Finally, it was newly created customised algorithms based on ANN and CNN models that produced better results overall.

CHAPTER 10: CONCLUSION

This chapter provides an overview of all the novel developments and contributions of this research to the body of knowledge in the field of Digital Forensics (DF) and the artifacts developed and evaluated in this study. Since the main objective of this study was to design a novel AI based Digital Forensic Protocol for the first responders, this was achieved in the form of conceptual design and further implementation of the I-DEEP protocol using a working prototype. Another main goal of the study was to develop an 'intelligent digital forensic framework' that demonstrated the technical feasibility of AI technology integration into the framework. In order to achieve 'triaging and prediction', that could facilitate the first responders and digital forensic investigators to make quick decisions, predictive analysis of cybercrime incidents and severity classification of the cybercrime data gathered during investigations, can be an effective strategy. The novel protocol (I-DEEP) developed in this research is agile and easily implementable, which was demonstrated by designing a working prototype based on the newly conceptualized protocol model. To integrate AI technology into the protocol an 'intelligent framework' named as Digital Intelligent Forensic Framework (DIF^2) was also developed. This model is based on a novel 'gearbox' model approach implemented in DIF^2 framework, that allows selection of best performing AI models and technologies. Various algorithm models were developed in a systematic manner with added complexity and enhanced functionality to demonstrate the feasibility of DIF^2 framework. The performance of the models was evaluated and results discussed in chapter-9. This study followed a 'iterative and incremental' research design and research process model specifically designed for this research that is based on DSR Methodology and the novel process model of this research was mapped to the DSRM approach and its outcomes.

10.1 Research contribution

This research intended to contribute towards the body of knowledge in digital forensics by designing and analysing a novel AI based digital forensic protocol model named as I-DEEP, which is a new artifact designed and developed for first responders to assist them in digital investigation and evidence analysis, using prediction analytics, triaging and data visualisation. Research also demonstrated the development and demonstration of various new artifacts in the form of customised algorithm using cutting edge AI techniques to build an AI based framework called Digital Intelligent Forensic Framework (DIF^2) and a working prototype. This framework provided a platform and a roadmap to demonstrate feasibility of AI integration into DF analysis

tools, which would result into productivity enhancement and decision-making for first-responders/cyber-crime investigators. The upcoming sections provide an overview of these artifacts and the contribution they make to the body of knowledge. Research has also produced an improved process model for DSR methodology using an 'iterative and incremental' agile approach by extending existing frameworks and model to bring new adaptations or technological improvements needed to provide guidance and theoretical underpinnings to this research.

10.2 New Artifacts Developed in this Research

This research contributed to body of knowledge by designing and developing new artifacts that enhances the efficacy of DF investigations and providing better productivity to first responders and cybercrime investigators by implementing AI technology. This will allow them do perform predictive analysis and triaging to achieve better decision making. During the research progression, critical gaps were identified during analysis of digital forensic frameworks and process models and developed improved process models and new artifacts. Other artifacts developed were a novel 'intelligent digital evidence extraction protocol' (I-DEEP) and an AI based 'intelligent forensics' framework named as 'Digital Intelligent Forensic Framework' (DIF^2). An overview of these artifacts and their contribution to the body of knowledge in digital forensics field is presented in subsequent sections in this chapter.

10.2.1 Intelligent Digital Evidence Extraction Protocol (I-DEEP)

The novel protocol model designed considers various phases of DF investigation and evidence gathering in detail whilst it tends to simplify the DF investigation process and focusses on a more 'agile' and implementable model. This newly developed protocol (I-DEEP) provides a framework for DF first responders and DF investigators by providing more agility and easy implementation. This pragmatic approach saves time by reducing the complexity and making the model more implementable. It also forms a theoretical framework to develop a working prototype that can be used by DF investigators to collect evidence and cybercrime related data during investigation and maintain 'chain of custody' information for future reference. The design process is covered in detail in section-5.6 in chapter-5. Chapter-6 further covered the prototype design and implementation of prototype based on I-DEEP protocol and also provided integration of AI techniques to make the model 'intelligent' and the resulting intelligent framework is referred as Digital Intelligent Forensic Framework (DIF^2).

10.2.2 Digital Intelligent Forensic Framework (DIF^2)

The newly developed framework named as DIF^2, provides pathway for integration of AI technology into the I-DEEP protocol. The I-DEEP protocol and DIF^2 framework, both are aligned to the key components of the digital forensic established standards and frameworks. It was established in the research that while existing tools implement some of these components or technology to some extent, they lack the advanced interoperable intelligence as discussed in chapter-2 (Literature Review). The proposed system formulation aims to address this gap. The framework combines classical programming, artificial intelligence (AI), and machine learning (ML) techniques, leveraging universally accepted standards and framework worldwide and in South Africa. The newly developed 'intelligent framework' (DIF^2) uses cutting edge technologies and provides integration of investigation process and AI and machine learning (ML) techniques to provide triaging and predictive analysis that relies on contextual data to make informed decisions in different scenarios. The integration of AI and ML at each step and operational ontology of DIF^2 are explained in chapters (7) and (8). The DIF2 framework is modelled on a unique 'gearbox model' approach which allows selection of best performing AI algorithms for predictive analysis and triaging. Newly created and customised AI algorithms like 'Custom_ANN_SGD' for predictive analytics and 'Enhanced_CNN_VGG' algorithm for image classification of contraband images, were created and implemented, which demonstrated remarkable performance. These customised algorithms were integrated into DIF^2 framework to improve performance of predictive modelling. DIF^2 also demonstrated framework automation and optimisation using 'AutoML-TPOT' technologies integration. This allowed for auto-selection of best performing algorithms, automatic hyperparameter optimisation and pipeline creation with minimum human intervention. As AI and ML systems operate based on their knowledge and learned patterns, the approach is referred to as 'intelligent systems' and intelligent forensics. Retraining of models and additional training datasets were utilized to thoroughly train the AI algorithms, following validation with test datasets. This framework provided a stepwise integration of advanced features of AI technology by developing customised algorithm that demonstrated technical feasibility and excellent performance for predictive analysis.

10.2.3 Prototype to implement and demonstrate the I-DEEP and DIF^2 frameworks

A prototype was designed and develop to implement I-DEEP protocol model. The main objective of this activity was to demonstrate that the new protocol model (I-DEEP) developed

is ‘agile’ and easily implementable in practical scenario. Although the prototype does not extensively implement all the theoretical aspects as the primary objective is to provide demonstration of the technical feasibility of the newly designed protocol model. This further allowed for a demonstration of implementation of ‘intelligent framework’ into the protocol model using newly designed AI framework (DIF^2) to be integrated to achieve optimisation and perform predictive analysis of cybercrime data and images. The prototype development maps to the third phase ‘Design and Development phase’ of the Peffers et al. (2007) of Design Science Research (DSR) Methodological Model. Figures 6.1 to 6.11 shows UI design of the application prototype, while section-6.2 explains the key features of the prototype, that assists DF investigators to collect evidential data and ‘chain of custody’ information.

10.2.4 Extended Model of Nucleus of Digital Forensic Research Process

This study found considerable gaps in digital forensic investigation process models described in section-2.3 that demanded an extension or revamp of these models. An improved model of Nucleus of Digital Forensic Research Process was re-modelled in this research, as represented in figure-2.7. The new artifact (Extended Nucleus of Digital Forensic Research Process) highlights that Computer Emergency Response Teams (CERT) forms the core of ‘Defence and Security’ and ‘Business and Industry’ domains. This extended model also emphasise that Digital Forensic (DF) frameworks, standards and tools play an indispensable role in cybercrime investigations, law enforcements and legal proceedings. Therefore, these new constructs are added as they also form key focus areas of DF research.

10.2.5 Extended Linear Process Model: Extended from (DFRWS (2001)

Digital Forensic Process consists of various phases as described in DFRWS (2001) Linear Process Model figure-2.2, representing a Linear Process Model, which comprises of stages from identification to preservation of evidence and finally decision stage. These phases were discussed in detail in section 1.2.8. After critical analysis of the model, some gaps were identified and new features proposed in the form of an ‘Extended Linear Model’ represented in figure-2.8. The main aspect introduced in this extended model is the ‘Optimisation Phase’, which constitute data reduction, predictive analysis, and visual representation of data using scatter charts, matrix plots and whisker plots. The additions in ‘Decision’ phase, like ‘Prediction and Triaging’ and other improvements like ‘optimisation’ and ‘visual representation’ are introduced. These new additions also provide the theoretical underpinnings for design and development of a new digital evidence extraction protocol (I-DEEP) and also

highlight the ‘gaps’ existing in present models. This ‘extended’ model fills the requirement to add cutting edge technological interventions like AI and data visualisation to achieve better efficiency in decision making and therefore, is a contribution to the body of knowledge.

10.4.6 Extended Output Mapping Model for investigation Tools used for incident response

Section-4.7 represents a newly created and extended version of ‘Output Mapping Model’ for DF and IR Tools classification in this research, as shown in figure-4.3. This new model provides a comparative analysis of most popular tools used for forensic incident response which is based on output mapping process. Mapping and classification are done based on output of these tools, which is mapped to DFRWS (2001) digital forensic investigation process model stages. This model provides interventions to identify and fill gaps in existing model. New model provides additional mapping based on AI based predictive analysis capabilities. Tools flagged with (O) are open source/free tools whereas ones flagged with (P) are proprietary. The output mapping clearly indicates that investigation tools used in incident response and the corresponding outputs do not map to all the DF phases and new technologies like AI are not supported by popular tools. Apparently phase 1 and phase 6 require manual interventions. This new model incorporates AI based prediction analysis (*Phase 5) to enhance efficacy mechanism and decision making. These new integrations ‘Prediction Phase’ (using AI) are not fulfilled by many existing tools and therefore shows the gap in present digital forensic tools landscape. It is evident that most popular tools do not support AI based prediction analysis and data visualisation functionality, that can utilise big data, produce intelligent output for greater insight into data and support decision making by first responders/cybercrime investigators. This new ‘extended model’ made a theoretical underpinning and a strong motivation to design and develop new interventions to fill this gap, which is the main objective of this research.

10.3 Justification

This research has contributed towards understanding and exploring the challenges (technical, methodological and legal), which are encountered in the DF research field and has contributed to body of knowledge by presenting new artifacts. After a comprehensive review and analysis of existing models, the research has proposed new models as well as produced new artifacts to fill the gaps explored during literature review (CHAPTER-2). The research delved into creation

of a new protocol (I-DEEP) that is more agile and demonstrated that it is practically implementable by creating a working prototype based on the I-DEEP protocol.

Research further explored the possibilities to implement AI technology and therefore developed new and customised AI based algorithms for predictive Modelling and achieved good efficacy in results. Development of a new intelligent framework (DIF^2) for AI implementation and automation was also demonstrated. New customised AI algorithms were developed based on Artificial Neural Network (ANN) and Convolutional Neural Network (CNN) algorithm models. These AI based models were optimised and enhanced to further improve accuracy in prediction modelling. ANN based customised model demonstrated a **98.33%** accuracy in predictive modelling, while CNN based 'Enhanced 3 VGG CNN' model demonstrated image identification and classification capabilities and achieved high efficacy and accuracy scores of **97%**.

10.4 Limitation of Research

The research was limited in the aspect that all the theoretical aspects of the I-DEEP protocol could not be implemented as the main objective was to demonstrate that the newly developed protocol is agile and implementable in principle.

Implementation of newly developed AI framework (DIF^2) was also limited by resource availability limitations, like system resources available for experimental demonstration were limited. Bigger datasets could not be analysed due to these limitations. Implementation of TPOT and AutoML were also affected by these factors as 'model pipelines' developed place huge demand on the resources. These experiments were implemented to primarily demonstrate that implementation of AI framework can achieve better optimisation of results and perform predictive analysis, along with objectives to demonstrate automation of hyperparameter optimisation with minimum human intervention. Main objective of the research was to develop AI based protocol and intelligent framework that has the potential to assist first responders and DF investigators in 'decision making' and 'triaging'. These two were important objectives achieved in this research, however with some limitations, as mentioned in this section. Another limitation was to test the framework on a 'real dataset' in public domain due to its non-availability. This limitation was due to high confidentially maintained by law enforcement in sharing the data in public domain as discussed by other researchers (Van Niekerk, 2017).

Although, not specifically stated as a 'key objective' of this research, the prototype developed could have been integrated with a database of digital forensic tools, providing an 'expert system' functionality to first responders/cybercrime investigators as a 'decision-making tool' for providing expert advice on 'tools selection' and 'function-list details'. Therefore it could be used for better expertise and decision making. This was a motivation of providing extensive details regarding tools used for digital forensic investigation and Incident Response processes. But the timeframe restrictions prohibited this idea and the scope would have been too big, this integration was left for future research.

10.5 Recommendations

This research demonstrated design and development of a novel DF investigation protocol (I-DEEP) that is more 'agile' and implementable using a prototype. This research further demonstrated design and development of an 'intelligent framework' (DIF^2) using an iterative and incremental' approach and stage-wise integration of newly created and customised AI algorithms to achieve better performance and accuracy in predictive modelling and image classification, as demonstrated in experiment-1 to 7). Experiment-1 focused on evaluation of baseline model and evaluation of performances of popular AI algorithms on 'cloud-based' cybercrime dataset. This study further demonstrated 'automation' using implementation of TPOT and AutoML techniques, in order to achieve 'Automation' and 'Optimisation' of AI models without human intervention as demonstrated in experiment-2. In next step, the research demonstrated the development of customised algorithms based on ANN and SGD ensemble model that was demonstrated in experiment-3. Experiment-4 successfully demonstrated classification capability for contraband images. Other extended experiments (5 to 7) provided more analysis of model performance by comparing their performance using variable datasets (50 vs 100 vs 300 records). These stage-wise enhancements, constitutes a progressive 'intelligent framework' development that can provide more complex AI integration and performance capabilities. This implementation used a unique 'gearbox model' which is analogous to selection of most suitable option (like a gear) for optimum performance and functionality using prototype. These newly developed artifacts not only align with the established standards of the digital forensic investigation process but also enhances their capabilities further. While existing DF tools can perform basic analysis of evidential data, they lack the advanced interoperable intelligence discussed earlier in the research and do not support wide AI integration possibilities. The proposed algorithms and framework enhancements offer potential system formulations aim to address this gap as demonstrated in chapters 6 to 8. The

framework combines classical programming (Python), artificial intelligence (AI), and machine learning (ML) techniques, leveraging on a cybercrime dataset from test data generated using newly developed prototype for training and testing. As AI systems operate based on their knowledge and learned patterns, the approach is referred to as intelligent systems and intelligent forensics. Re-training using different datasets is necessary for improved results, and additional training datasets can be utilized to thoroughly train the AI algorithms, following validation with test datasets. The DIF^2 framework integration into prototype as well as other tools can empower investigators to make informed decisions in different scenarios. The integration of AI at each step, description of operational ontology of the 'Automation Process' of DIF^2 framework using TPOT and AutoML are explained in the chapter-8 in detail, which provides a roadmap for implementing 'Automation' and 'Optimisation' of training the AI models with minimum human intervention and also achieving best performance using hyperparameter optimisation. This implementation helped in making predictive analysis performance better, which demonstrated that automation can be effective to reduce human intervention and results into better decision-making capabilities for first responders and DF investigators.

This research evaluated the newly developed prototype, discussed the experimental -results and presents the performance output of four staged experimental setup (experiments 1 to 4) which was conducted to evaluate the performance of prototype. These experiments were extended with bigger dataset to evaluate the performance of AI models. All newly created artifacts (algorithm models) and their implementations into DIF^2 Framework successfully demonstrated the capabilities and benefits of AI integration and also supports the hypothesis that decision making can be improved using 'triaging' and predictive analytics using AI and ML. This evaluation also provided the valuable observations on different AI models behaviour during predictive analysis in classification problems (City wise prediction of cybercrime incidents and prediction of critical status) and regression problems (prediction od cybercrime incidents, severity and triage scores), which can be used by investigators to perform triaging, decision making and better resource planning. The DIF^2 framework demonstrated applicability of AI automation and optimisation using pipelines and use of AutoML and TPOT automated frameworks. These experimental evaluations provided a strong motivation for AI integration into I-DEEP protocol and DIF^2 framework and also demonstrated the feasibility of implementation of 'triaging' and predictive analysis of cybercrime incidents for better decision making. The evaluation found best results achieved by the newly developed customised

algorithm model based on ANN and SGD algorithms. Although other models (baseline model, Spot-check and Selection algorithm and AutoML based automation pipelines also showed their own utility. While baseline model served as a template to develop more complex models and 'AutoML and TPOT' based 'Automation Framework' showed promising benefits towards automating the data analysis and prediction with diverse datasets and more diverse problems. Enhanced CNN based algorithm also showed high efficacy in image analysis. It was newly created customised algorithms based on ANN and CNN that produced better results overall. Therefore, these technologies can be used by first responders for triaging and better decision-making and can be recommended to be integrated into other DF investigation tools also for added functionality and better efficacy.

The DIF^2 framework which was demonstrated as a prototype's 'intelligent forensics' enhancement, used variable dataset sizes and 'features' extracted from the cloud. This framework can be further extended to be implemented on 'Realtime Cloud Based' Computing platforms to analyse 'Realtime Data' through automated steps. Realtime data or 'Cloud based data' is continuously generated through sources such as sensors, IoT devices, social media streams, transaction logs etc. This data requires 'feature engineering' which involves data transformations, data normalisation, data rescaling, or encoding to improve AI model performance. Model selection and these steps of performance optimisation can be further automated using AutoML and pipeline optimisation techniques. This research used Machine Learning models like ANN (Artificial Neural Networks) and CNN (Convolutional Neural Networks) to achieve desired objectives of data prediction and image analysis. Recurrent Neural Networks (RNN) and other 'Deep Learning' models can also be effective in performing predictive analysis on 'real-time' and 'cloud-based' data. Usually models are trained on historical data. In order to achieve 'real-time analysis', model may be continuously trained on new data to adapt to changing patterns overtime. Once the model is trained, it is deployed to make predictions or classifications in real-time as new data streams in. This involves feeding the incoming data into the model continuously and obtaining predictions or insights almost instantaneously which is known as 'Realtime inference'. This can be achieved by created RESTful APIs to extract 'live data' and use of algorithms ensemble like ones demonstrated in this research. Integration of an expert system that can suggest best tools to first responders or automatically or callup FaaS (Forensics-as-a-Service) based most suitable investigation tools/plugins can be of great assistance to them.

10.6 Future Scope

This system or prototype based on I-DEEP protocol and DIF^2 'intelligent framework' provides enough potential for future improvements. Integration of this system with FaaS or Forensic-as-a-Service (and similar futuristic technologies) can be one of them. The prototype developed in this research can also be integrated with other generative-AI and NLP (Natural Language Processing) platforms like Chat-GPT, to provide functionality as an expert system to assist digital forensic investigations in decision making. Other possibilities include integrating this system to Digital Evidence Management Framework (DEMF) to ensure 'digital chain of custody' and a more comprehensive 'global cloud-based repository' for cybercrime database, that can be analysed by different agencies and obtain valuable insights using predictive analytics. Recurrent Neural Networks (RNN) and other 'Deep Learning' models can also be effective in performing predictive analysis on 'real-time' and 'cloud-based' data. RESTful APIs to extract 'live data' and use of algorithms ensemble using AutoML-TPOT can made more effective to achieve automated hyperparameter tuning and data optimisation without human intervention.

10.7 Summary

This research showcased the design and implementation of an innovative DF investigation protocol named I-DEEP, emphasizing agility in the investigation and practicality of the protocol through a prototype development. Additionally, it introduced an 'intelligent framework' known as DIF^2 framework, incrementally developed and implemented following 'iterative and incremental' approach and a unique 'gearbox model' analogy to support best performing models, to assess the predictive accuracy of AI algorithms as shown in various experiments (1 to 7). The experiments were designed to demonstrate automation through the deployment of TPOT and AutoML, enabling AI model selection and optimization without human intervention. Research demonstrated the creation of custom algorithms based on ANN and SGD ensemble models. These staged enhancements constituted a progressive development of the intelligent framework using agile approach, enhancing AI integration and performance capabilities while aligning with established digital forensic investigation standards. While traditional DF tools offer basic analysis of evidential data, they lack the advanced interoperable intelligence discussed in this research and do not support extensive AI integration. The proposed algorithms and framework enhancements aimed to bridge this gap, as demonstrated in this research. Combining classical (Python) programming with AI and ML techniques, the

framework leveraged a cybercrime dataset of variable size and features generated by the prototype for training and testing. The research demonstrated that integration of the DIF^2 framework into the prototype empowers investigators to make informed decisions across various scenarios using predictive analytics. The research elaborated on the integration of more advanced AI technologies at each step, detailing the operational ontology of the automation process using TPOT and AutoML. This comprehensive implementation provides a roadmap for implementing automation and optimization in training AI models with minimal human intervention, enhancing performance through hyperparameter optimization. Furthermore, this implementation improves performance through predictive analysis, demonstrating the effectiveness of automation in reducing human intervention and enhancing decision-making capabilities for first responders and DF investigators. This chapter also delved into research justification, limitations of this research, recommending new research potential and future scope in the domain of digital forensics for future researchers.

REFERENCES

27 BEST Penetration Testing (Pentest) Tools in 2023. (n.d.). Retrieved March 19, 2023, from https://www.guru99.com/top-5-penetration-testing-tools.html

Adams, R., Hobbs, V., & Mann, G. (2013). The Advanced Data Acquisition Model (Adam): A Process Model for Digital Forensic Practice. *Journal of Digital Forensics, Security and Law*. https://doi.org/10.15394/jdfsl.2013.1154

Association of Certified Fraud Examiners. (n.d.). Retrieved March 27, 2023, from https://www.acfesa.co.za/CFE

Al-Hammadi, Y., & Aickelin, U. (2008). Detecting bots based on keylogging activities. In 2008 *Third International Conference on Availability, Reliability and Security* (pp. 896-902). IEEE.

Al-Hammadi, Y., & Aickelin, U. (2010). Detecting botnets through log correlation. *arXiv preprint arXiv:1001.2665.*

Al-Khawaja, A., & Sadkhan, S. B. (2021, August). Intelligence and electronic warfare: challenges and future trends. In *2021 7th International Conference on Contemporary Information Technology and Mathematics (ICCITM)* (pp. 118-123). IEEE.

Alawida, M., Abu Shawar, B., Abiodun, O. I., Mehmood, A., Omolara, A. E., & Al Hwaitat, A. K. (2024). Unveiling the dark side of chatgpt: Exploring cyberattacks and enhancing user awareness. *Information*, *15*(1), 27.

Baggili, I., Marrington, A., Baabdallah, A., & Al-Safi, D. (2013). *Research Trends in Digital Forensic Science: An Empirical Analysis of*. https://doi.org/10.1007/978-3-642-39891-9_9

Baryamureeba, V., & Tushabe, F. (2004). The enhanced digital investigation process model. *Digital Investigation*.

Bauchner, E. (2014). Computer investigation. In *Solving crimes with science, forensics*. Mason Crest Publishers.

Beebe, N. (2009). Digital forensic research: The good, the bad and the unaddressed.

In Advances in Digital Forensics V: Fifth IFIP WG 11.9 *International Conference on Digital Forensics, Orlando, Florida, USA, January 26-28, 2009, Revised Selected Papers 5* (pp. 17-36). Springer Berlin Heidelberg.

BREZINSKI, D., & Killalea, T. (2002). RFC 3227: Guidelines for Evidence Collection and Archiving. Network Working Group. February.

Birk, D., & Wegener, C. (2011, May). Technical issues of forensic investigations in cloud computing environments. In *2011 Sixth IEEE international workshop on systematic approaches to digital forensic engineering* (pp. 1-10). IEEE.

Britz, M. (2009). *Computer forensics and cyber crime : an introduction* (2nd ed.). Pearson Prentice Hall.

Brown, C. S. (2015). Investigating and prosecuting cyber crime: Forensic dependencies and barriers to justice. *International Journal of Cyber Criminology*, *9*(1), 55.

Brownlee, J. (2016). *Machine Learning Mastery With Python* (v1.21).

Brownlee, J. (2021). *Deep Learning for Computer Vision Archives - MachineLearningMastery.com*. https://machinelearningmastery.com/category/deep-learning-for-computer-vision/

Bulbul, H. I., Yavuzcan, H. G., & Ozel, M. (2013). Digital Forensics: An Analytical Crime Scene Procedure Model (ACSPM). *Forensic Science International*, *233*(1–3). https://doi.org/http://dx.doi.org/10.1016/j.forsciint.2013.09.007

Burden, K., & Palmer, C. (2003). Internet crime: Cyber crime - A new breed of criminal? In *Computer Law and Security Report* (Vol. 19, Issue 3, pp. 222–227). Elsevier Ltd. https://doi.org/10.1016/S0267-3649(03)00306-6

Carrier, B. D., & Spafford, E. H. (2004). *An Event-Based Digital Forensic Investigation Framework* *. http://www.digital-evidence.org/papers/dfrws_event.pdf

Carrier, B. D., & Spafford, E. H. (2006). Categories of digital investigation analysis techniques based on the computer history model. *Digital Investigation*, *3*, 121–130. https://doi.org/10.1016/J.DIIN.2006.06.011

Carrier, B., & Spafford, E. H. (2003). Getting Physical with the Digital Investigation Process.

International Journal of Digital Evidence Fall, *2*(2). https://pdfs.semanticscholar.org/915b/524318e2f0689b586ba7ae89ea39e9b22ce3.pdf

Casey, E., Ferraro, M., & Nguyen, L. (2009). Investigation Delayed Is Justice Denied: Proposals for Expediting Forensic Examinations of Digital Evidence. *Journal of Forensic Sciences*, *54*(6), 1353–1364. https://doi.org/10.1111/j.1556-4029.2009.01150.x

Chaturvedi, A., Awasthi, A., & Shanker, S. (2020). Cyber forensic-A literature review. *Trinity Journal of Management, IT & Media*, *10*(1), 24-29.

Ciardhuáin, S. Ó. (2004). An extended model of cybercrime investigations. *International Journal of Digital Evidence*, *3*(1), 1-22.

Collie, J. (2018). Digital forensic evidence—Flaws in the criminal justice system. *Forensic Science International*, *289*, 154–155. https://doi.org/10.1016/j.forsciint.2018.05.014

Creswell, J. W. (2021). *A concise introduction to mixed methods research*. SAGE publications.

Cross, F. B. (2008). *The theory and practice of statutory interpretation*. Stanford University Press.

Dayal, M., Panwar, D. S., D'souza, D. P., Sharma, G., Upreti, K., Gupta, S., & Duggal, R. (2023). 5G Networks and Their Applications: Ushering in a New Era. In *Applying Drone Technologies and Robotics for Agricultural Sustainability* (pp. 259-268). IGI Global.

Deloitte. (2023). *2023 Global Future of Cyber Survey*. https://www.deloitte.com/global/en/services/risk-advisory/content/future-of-cyber.html

DFRWS. (2001). A Road Map for Digital Forensic Research -Final Approved For Public Release Executive Summary. *Report From the First Digital Forensic Research Workshop (DFRWS)*. http://dfrws.org/sites/default/files/session-files/a_road_map_for_digital_forensic_research.pdf

Diesner, J., Frantz, T. L., & Carley, K. M. (2005). Communication networks from the Enron email corpus “It's always about the people. Enron is no different”. Computational & Mathematical Organization Theory, 11, 201-228.

Du, X., Le-Khac, N.-A., & Scanlon, M. (2017). *Evaluation of Digital Forensic Process Models with Respect to Digital Forensics as a Service*. https://arxiv.org/ftp/arxiv/papers/1708/1708.01730.pdf

Dumont, D. (2010, November). Cyber security concerns of Supervisory Control and Data Acquisition (SCADA) systems. In *2010 IEEE International Conference on Technologies for Homeland Security (HST)* (pp. 473-475). IEEE.

Elhalabi, M. J., Manickam, S., Melhim, L. B., Anbar, M., & Alhalabi, H. (2014). A review of peer-to-peer botnet detection techniques. *Journal of Computer Science,* 10(1), 169.

European Information Society Group (EURIM, 2010). Separating Myth from Reality and Snake-Oil from Practicality; Partnership Policing for the Information Society, European Information Society Group (EURIM): London, UK, 2010. Available online: http://www.eurim.org.uk/activities/e-crime/partpolicing.php (accessed on 10 January 2022).

Feurer, M., Klein, A., Jost, K. E., Springenberg, T., Blum, M., & Hutter, F. (2015). Efficient and Robust Automated Machine Learning. *Advances in Neural Information Processing Systems*, *28*. http://automl.org

Garfinkel, S. L. (2010). Digital forensics research: The next 10 years. *Digital Investigation*. https://doi.org/10.1016/j.diin.2010.05.009

Geldenhuys, K. (2023). Cybercrime as a Service: A growing threat in the cyber world. *Servamus Community-based Safety and Security Magazine*, *116*(11), 26-28.

GERO, J. S., & KANNENGIESSER, U. (2006). A FUNCTION-BEHAVIOUR-STRUCTURE ONTOLOGY OF PROCESSES. In *Design Computing and Cognition '06* (pp. 407–422). Springer Netherlands. https://doi.org/10.1007/978-1-4020-5131-9_21

Gogolin, G., & Jones, J. (2010). Law enforcement's ability to deal with digital crime and the implications for business. *Information Security Journal: A Global Perspective, 19(3),* 109-117.

Graham-Harrison. (2015). *Could Isis's 'cyber caliphate' unleash a deadly attack on key targets? | World news | The Guardian*.

Grispos, G., Storer, T., & Glisson, W. B. (2012). Calm before the storm: The challenges of cloud computing in digital forensics. *International Journal of Digital Crime and Forensics (IJDCF)*, 4(2), 28-48.

Grobler, M., & van Vuuren, J. J. (2012). Collaboration as proactive measure against cyber warfare in South Africa. *African Security Review*, *21*(2), 61–73. https://doi.org/10.1080/10246029.2012.654803

Gunawardhana, Muditha. (2021). Role of Digital Forensic in solving cyber crimes. 10.13140/RG.2.2.18493.95205.

Gupta, M., Akiri, C., Aryal, K., Parker, E., & Praharaj, L. (2023). From chatgpt to threatgpt: Impact of generative ai in cybersecurity and privacy. *IEEE Access*.

Healy, Paul, M., and Krishna G. Palepu. (2003). "The Fall of Enron ." *Journal of Economic Perspectives*, 17 (2): 3–26**.**DOI: 10.1257/089533003765888403

Herrerias, J., & Gomez, R. (2007, April). A log correlation model to support the evidence search process in a forensic investigation. In *Second International Workshop on Systematic Approaches to Digital Forensic Engineering (SADFE'07)* (pp. 31-42). IEEE.

Hevner, A. R., March, S. T., Park, J., & Ram, S. (2004). Design science in information systems research. *MIS quarterly*, 75-105.

Hitchcock, B., Le-Khac, N.-A., & Scanlon, M. (2016). Tiered forensic methodology model for Digital Field Triage by non-digital evidence specialists. *Digital Investigation*, *16*, S75–S85. https://doi.org/10.1016/j.diin.2016.01.010

Ieong, R. S. (2006). FORZA–Digital forensics investigation framework that incorporate legal issues. *digital investigation,* 3, 29-36.

International Organization for Standardization. (2012). https://www.iso.org/home.html (Accessed on -18/07/2024)

Institute of Justice, N. (2008). *Electronic Crime Scene Investigation: A Guide for First Responders, Second Edition www.ojp.usdoj.gov/nij*. https://www.ncjrs.gov/pdffiles1/nij/219941.pdf

Interpol. (2023). *AFRICAN CYBERTHREAT ASSESSMENT REPORT 2023*.

https://www.interpol.int/content/download/16759/file/AfricanCyberthreatAssessment_ENGLISH.pdf

Irons, A., & Lallie, H. S. (2014). Digital forensics to intelligent forensics. Future Internet, 6(3), 584-596.

Jacob, S. (2023). The Rapid Increase of Ransomware Attacks Over the 21st Century and Mitigation Strategies to Prevent Them from Arising.

Johansen, G. (2017). Digital forensics and incident response. Packt Publishing Ltd.

Johnson, C., Davies, R. & Reddy, M (2022). Using digital forensics in higher education to detect academic misconduct. Int J Educ Integr 18, 12. https://doi.org/10.1007/s40979-022-00104-1

Johnson, S. A., & Ananthakumaran, S. (2021). Smart Digital Forensic Framework for Crime Analysis and Prediction using AutoML. *International Journal of Advanced Computer Science and Applications*, *12*(3).

Jusas, V., Birvinskas, D., & Gahramanov, E. (2017). Methods and Tools of Digital Triage in Forensic Context: Survey and Future Directions. *Symmetry*, *9*(4), 49. https://doi.org/10.3390/sym9040049

Kohn, M. D., Eloff, M. M., & Eloff, J. H. P. (2013). Integrated digital forensic process model. *Computers & Security*, *38*(0), 103–115. https://doi.org/http://dx.doi.org/10.1016/j.cose.2013.05.001

Kruse, W. G., & Heiser, J. G. (2001). *Computer forensics : incident response essentials*. Addison-Wesley. https://www.ieee-security.org/Cipher/BookReviews/2002/KruseHeiser.html

Kuechler, B., & Vaishnavi, V. (2008). On theory development in design science research: anatomy of a research project. *European Journal of Information Systems*, *17*(5), 489–504. https://doi.org/10.1057/ejis.2008.40

Lai, K. Y. (2011). Profiling internet pirates. *HKU Theses Online (HKUTO)*.

Lai, P., Chow, K. P., Fan, X. X., & Chan, V. (2013). An empirical study profiling internet

pirates. In Advances in Digital Forensics IX: 9th IFIP WG 11.9 *International Conference on Digital Forensics, Orlando, FL, USA, January 28-30, 2013, Revised Selected Papers 9* (pp. 257-272). Springer Berlin Heidelberg.

Lallie, H. S., & Pimlott, L. (2012). Applying the ACPO principles in public cloud forensic investigations. *Journal of Digital Forensics, Security and Law,* 7(1), 5.

Lee, C. A., Woods, K., Kirschenbaum, M., & Chassanoff, A. (2013). From Bitstreams to Heritage: Putting Digital Forensics into Practice.

Marcella, A. J. (Ed.). (2021). *Cyber Forensics: Examining Emerging and Hybrid Technologies*. CRC Press.

Marsili, M. (2019). The War on Cyberterrorism. *Democracy and Security*, *15*(2), 172–199. https://doi.org/10.1080/17419166.2018.1496826

Mijwil, M. M. (2018). *Artificial Neural Networks Advantages and Disadvantages Solving Traveling Salesman Problem (TSP) View project Artificial Neural Networks Advantages and Disadvantages*. https://www.researchgate.net/publication/323665827

Militano, L., Araniti, G., Condoluci, M., Farris, I., & Iera, A. (2015). Device-to-device communications for 5G internet of things. *EAI Endorsed Transactions on Internet of Things*, *1*(1), e4-e4.

Mithas, S. (2016). *Digital intelligence: What every smart manager must have for success in an information age*. Penguin UK.

Mocas, S. (2004). Building theoretical underpinnings for digital forensics research. *Digital Investigation*, *1*(1), 61–68. https://doi.org/http://dx.doi.org/10.1016/j.diin.2003.12.004

Mohammad, R. M. (2018, October). A neural network based digital forensics classification. In *2018 IEEE/ACS 15th International Conference on Computer Systems and Applications (AICCSA)* (pp. 1-7). IEEE.

Moore, R. (2010). *Cybercrime*. Routledge. https://doi.org/10.4324/9781315721767

Moore, R. (2014). *Cybercrime*. Routledge. https://doi.org/10.4324/9781315721767

Myburgh, D. C. (2016). *Developing a framework for the search and seizure of digital*

evidence by forensic investigators in South Africa (Doctoral dissertation, North-West University (South Africa), Potchefstroom Campus).

Nair, V. U. (2023). *Smart car hacking: a threat to the rising automotive industry* (Doctoral dissertation, Dublin Business School).

Navneet, K. (2018). INTRODUCTION OF CYBER CRIME AND ITS TYPE. *International Research Journal of Computer Science*, *V*, 435–439.

Nieman, A. (2009). Cyberforensics: bridging the law/technology divide. *Journal of Information, Law and Technology*, *2009*(1).

NIST. (2023). http://dx.doi.org/10.6028/NIST.SP.800-61r2

Nobles, C., Burrell, D. N., Burton, S. L., & Waller, T. (2023). Driving into cybersecurity trouble with autonomous vehicles. In *Handbook of Research on Cybersecurity Risk in Contemporary Business Systems* (pp. 255-273). IGI Global.

Nortjé, J., & Myburgh, D. (2019). The Search and Seizure of Digital Evidence by Forensic Investigators in South Africa. *PER: Potchefstroomse Elektroniese Regsblad*, *22*(1), 1–42. https://doi.org/10.17159/1727-3781/2019/v22i0a4886

Olson, R. S., & Moore, J. H. (2016, December). TPOT: A tree-based pipeline optimization tool for automating machine learning. In *Workshop on automatic machine learning* (pp. 66-74). PMLR.

Ono, J. P., Castelo, S., Lopez, R., Bertini, E., Freire, J., & Silva, C. (2021). PipelineProfiler: A Visual Analytics Tool for the Exploration of AutoML Pipelines. *IEEE Transactions on Visualization and Computer Graphics*, *27*(2), 390–400. https://doi.org/10.1109/TVCG.2020.3030361

Paul Joseph, D., & Norman, J. (2019). An analysis of digital forensics in cyber security. In *First International Conference on Artificial Intelligence and Cognitive Computing: AICC 2018* (pp. 701-708). Springer Singapore.

Peffers, K., Tuunanen, T., Rothenberger, M. A., & Chatterjee, S. (2007). A Design Science Research Methodology for Information Systems Research. *Published in Journal of Management Information Systems*, *24*(3), 45–78.

http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.535.7773&rep=rep1&type=pdf

Purao, S., Henfridsson, O., Rossi, M., & Sein, M. (2013). Ensemble artifacts: From viewing to designing in action design research. *Systems, Signs & Actions: An International Journal on Information Technology, Action, Communication and Workpractices*, *7*(1), 73-81.

Queirós, A., Faria, D., & Almeida, F. (2017). Strengths and limitations of qualitative and quantitative research methods. *European journal of education studies*.

Raghavan, S. (2013). Digital forensic research: current state of the art. *CSI Transactions on ICT*, *1*(1), 91–114. https://doi.org/10.1007/s40012-012-0008-7

Reich, P., Reich, P. C., Weinstein, S., Wild, C., & Cabanlong, A. S. (2010). Cyber Warfare: A Review of Theories, Law, Policies, Actual Incidents – and the Dilemma of Anonymity. *European Journal of Law and Technology*, *1*(2).

Reilly, D., Wren, C., & Berry, T. (2011). Cloud computing: Pros and cons for computer forensic investigations. *International Journal Multimedia and Image Processing (IJMIP),* 1(1), 26-34.

Robinson, M., Jones, K., & Janicke, H. (2015). Cyber warfare: Issues and challenges. *Computers & Security*, *49*, 70–94. https://doi.org/10.1016/J.COSE.2014.11.007

Ribaux, O., Girod, A., Walsh, S. J., Margot, P., Mizrahi, S., & Clivaz, V. (2003). Forensic intelligence and crime analysis. Law, Probability and Risk, 2(1), 47-60.

Ribaux, O., Walsh, S. J., & Margot, P. (2006). The contribution of forensic science to crime analysis and investigation: forensic intelligence. Forensic science international, 156(2-3), 171-181.

Ribaux, O., Baylon, A., Roux, C., Delémont, O., Lock, E., Zingg, C., & Margot, P. (2010). Intelligence-led crime scene processing. Part I: Forensic intelligence. Forensic science international, 195(1-3), 10-16.

Rogers, M. K., Goldman, J., Mislan, R., Wedge, T., & Debrota, S. (2006). Computer Forensics Field Triage Process Model. *Journal of Digital Forensics, Security and Law*,

1(2). http://ojs.jdfsl.org/index.php/jdfsl/article/viewFile/222/174

Rogers, M., Seigfried, K., & Tidke, K. (2006). Computer Criminal Behavior: A Psychological Analysis. *DIGITAL FORENSIC RESEARCH CONFERENCE*. https://doi.org/10.1016/j.diin.2006.06.002

Saunders, M. N., & Tosey, P. C. (2013). The layers of research design. *Rapport*, (Winter), 58-59.

Sadiku, M. N. O., Tembely, M., & Musa, S. M. (2017). Digital Forensics. *International Journal of Advanced Research in Computer Science and Software Engineering*, *7*(4), 274–276. https://doi.org/10.23956/IJARCSSE/V7I4/01404

Sathiyanarayanan, M. (2017). Improving visual investigation analysis of digital communication data within e-discovery.

Schultz, C. B. (2016). *CYBERCRIME: AN ANALYSIS OF CURRENT LEGISLATION IN SOUTH AFRICA* [University of Pretoria]. https://repository.up.ac.za/bitstream/handle/2263/60091/Schultz_Cybercrime_2017.pdf?sequence=1

Schopp, M., & Hillmann, P. (2020). Agile Approach for IT Forensics Management. *arXiv preprint arXiv:2007.04125*.

Selamat, S. R., Yusof, R., & Sahib, S. (2008). Mapping process of digital forensic investigation framework. International Journal of Computer Science and Network Security, 8(10), 163-169.

Shakarian, P., Shakarian, J., & Ruef, A. (2013). Introduction to Cyber-Warfare. In *Introduction to Cyber-Warfare*. Elsevier Inc. https://doi.org/10.1016/C2012-0-06618-5

Simonyan, K., & Zisserman, A. (2015). Very deep convolutional networks for large-scale image recognition. *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*.

Singh, A., Saini, K., Nagar, V., Aseri, V., Sankhla, M. S., Pandit, P. P., & Chopade, R. L. (2022). Artificial intelligence in edge devices. In *Advances in Computers* (Vol. 127, pp. 437-484). Elsevier.

Singh, J., & Rahman, N. A. (2023, October). Cybercrime-As-A-Service (Malware). In *2023 International Conference on Evolutionary Algorithms and Soft Computing Techniques (EASCT)* (pp. 1-5). IEEE.

Snail, S. (2009). Cyber Crime in South Africa – Hacking, cracking, and other unlawful online activities. *Journal of Information, Law and Technology*.

Snyder, H. (2019). Literature review as a research methodology: An overview and guidelines. *Journal of business research*, *104*, 333-339.

Sood, A. K., & Enbody, R. J. (2013). Crimeware-as-a-service—a survey of commoditized crimeware in the underground market. *International journal of critical infrastructure protection*, *6*(1), 28-38.

Stanhope, D. J., & Dickson, M. (2012). Welcome to the Era of digital intelligence. Forrester Research.

State v. Ndiki (2008). Retrieved March 19, 2023, from https://www.saflii.org/za/cases/ZAGPJHC/2009/63.pdf

Swales, L. (2018). An analysis of the regulatory environment governing hearsay electronic evidence in South Africa: suggestions for reform–part two. *Potchefstroom Electronic Law Journal/Potchefstroomse Elektroniese Regsblad*, *21*(1).

Symantec. (2018). *ISTR Internet Security Threat Report*. *23*. http://images.mktgassets.symantec.com/Web/Symantec/%7B3a70beb8-c55d-4516-98ed-1d0818a42661%7D_ISTR23_Main-FINAL-APR10.pdf?aid=elq_

Taylor, M., Haggerty, J., Gresty, D., & Lamb, D. (2011). Forensic investigation of cloud computing systems. *Network Security*, 2011(3), 4-10.

Teichmann, F., Boticiu, S. R., & Sergi, B. S. (2023). The evolution of ransomware attacks in light of recent cyber threats. How can geopolitical conflicts influence the cyber climate?. *International Cybersecurity Law Review*, *4*(3), 259-280.

torproject (2023). https://www.torproject.org/ (accessed on 31 December 2023)

(US Department of Justice, 2022). Retrieved March 19, 2022 from:

https://www.justice.gov/sites/default/files/jmd/legacy/2014/08/02/fy09-fbi.pdf

Vacca, J. R. (2010). *Managing information security*. Elsevier.

Vadza, Kejal. (2011). Cyber Crime & its Categories. *Indian Journal of Applied Research*. 3. 130-133. 10.15373/2249555X/MAY2013/39.

Valjarević, A., Venter, H., & Petrović, R. (2016, November). ISO/IEC 27043: 2015—Role and application. In *2016 24th Telecommunications Forum (TELFOR)* (pp. 1-4). IEEE.

Van Dijk, A., Hoogewoning, F., & Punch, M. (2015). What matters in policing. *Change, values and leadership in turbulent times. Bristol, UK: Policy Press, University of Bristol*.

Van Niekerk, B. (2017). An analysis of cyber-incidents in South Africa. *The African Journal of Information and Communication (AJIC)*, 20, 113-132. https://doi.org/10.23962/10539/23573

Van Wyk, B. (2012). Research design and methods Part I. *University of Western Cape*.

Veerasamy, N. (2009). *Towards a Conceptual Framework for Cyber-terrorism*. 4th International Conference on Information Warfare and Security.

Walliman, N. (2021). *Research methods: The basics*. Routledge.

Wen, Z., Liu, H., Shi, J., Li, Q., He, B., & Chen, J. (2020). ThunderGBM: Fast GBDTs and random forests on GPUs. *Journal of Machine Learning Research*, *21*(108), 1-5.

Wu, H., & Zheng, G. (2020). Electronic evidence in the blockchain era: New rules on authenticity and integrity. *Computer law & security review*, *36*, 105401.

Wu, T., Breitinger, F., & O'Shaughnessy, S. (2020). Digital forensic tools: Recent advances and enhancing the status quo. *Forensic Science International: Digital Investigation*, *34*, 300999.

Yaacoub, J.-P. A., Noura, H. N., Salman, O., & Chehab, A. (2021). Digital Forensics vs. Anti-Digital Forensics: Techniques, Limitations and Recommendations. *Cryptography and Security*, 1–40. https://arxiv.org/abs/2103.17028v1

Yuki, J. Q., Sakib, M. M. Q., Zamal, Z., Habibullah, K. M., & Das, A. K. (2019, July). Predicting crime using time and location data. In *Proceedings of the 7th International Conference on Computer and Communications Management* (pp. 124-128).

Zhou, D., Song, Y., Zha, H., & Zhang, Y. (2005, December). Towards discovering organizational structure from email corpus. In *Fourth International Conference on Machine Learning and Applications (ICMLA'05)* (pp. 6-pp). IEEE.

APPENDIX-A

Tools used for Incident Response by First Responders

A.1: Penetration Testing and Diagnostic Tools

SuperScan

SuperScan is a free Windows-based penetration testing tool that is only available in closed-source format. It includes networking utilities like traceroute, ping, HTTP HEAD and whois, as well as a range of noteworthy features. These features include exceptional scanning speed, the ability to support virtually unlimited IPs, improved host detection using numerous ICMP methods, and also support for “TCP SYN” scanning.

SuperScan also enables users to create uncomplicated HTML reports, conduct source port scanning, and perform thorough banner grabbing. The tool has an extensive built-in database describing a large number of ports and the ability to randomize the order of IP and port scanning. Additionally, it is equipped with an extensive capability to enumerate Windows hosts.

IBM Internet Scanner

This is a penetration testing tool that forms the cornerstone of robust network security for businesses. By pinpointing vulnerabilities that exist in the network, it can minimize business risk. The tool offers various features, including the ability to automate scans and detect vulnerabilities, making it one of the best pen-testing tools available. It provides comprehensive vulnerability management, identifying security holes and weaknesses in the network, which helps to minimize the risk. The Internet Scanner is also capable of detecting over 1,300 types of various network devices, making it a powerful and versatile tool for businesses.

Scapy

Scapy is an interactive and powerful pen-testing tool capable of performing several network-related tasks, including scanning, probing, and attacks. It uses a combination of techniques that are difficult to replicate using other tools, providing users with several notable features. These include the ability to send invalid frames and inject 802.11 frames for performing specific tasks, as well as the ability to build packets precisely to user requirements. Scapy also reduces

the number of lines of code that pen-tester is required to write to execute a specific task, making it a highly efficient tool for network-related activities.

Vega

Vega is an open-source web security scanner and pen-testing platform created to test the security of web applications. Its main features include automated, manual, and hybrid security testing, making it useful for identifying vulnerabilities like blind SQL injection, cross-site scripting, stored cross-site scripting and shell injection. Vega is also capable of automatically logging into websites with user credentials and is compatible with OS X, Windows operating systems and Linux. The detection modules of Vega are written in JavaScript, therefore they are portable, thus providing high flexibility and customization options.

Ettercap

Ettercap is a powerful network analysis tool that can be used for a variety of tasks, including network monitoring, host analysis, and protocol dissection. It is widely used by ethical hackers and security professionals for conducting security assessments and penetration testing.

One of the key features of Ettercap is its ability to conduct ARP poisoning, which allows it to intercept and analyze network traffic between two hosts on a switched LAN. This makes it a useful tool for identifying vulnerabilities in network infrastructure and detecting potential security threats.

Ettercap also includes features for injecting characters into a live connection between a server and client, which can be useful for testing the security of web applications and other network-based software. It can also sniff SSL encrypted data, even when using a proxy connection, making it a valuable tool for analyzing and securing network traffic.

Ettercap's API enables the development of custom plugins, allowing users to enhance its functionality and tailor it to their specific needs. This makes it a flexible and versatile tool that can be adapted to a wide range of network analysis tasks.

Aircrack

This tool is widely recognized for its dependability and efficiency, particularly in cracking weak Wi-Fi connections, which makes it valuable for penetration testing purposes. Its capacity

to crack wireless encryption keys for WEP, WPA, and WPA2 simplifies the procedure of cracking these protocols.

Its key features include support for various cards/drivers, compatibility with multiple operating systems and platforms, and a new WEP attack called PTW. It also includes support for various attacks like WEP Fragmentation attack, dictionary attack, and an improved tracking speed.

Angry IP Scanner

Angry IP Scanner is a useful tool for network administrators and ethical hackers to quickly scan networks and identify active hosts and open ports. The software is lightweight and can be run on multiple platforms, making it a versatile tool for network scanning.

One of the key strengths of Angry IP Scanner is its ability to scan both local and external networks, making it useful for identifying potential security vulnerabilities in a network. Its open-source nature also allows for customizations and the addition of new features.

Another notable feature of Angry IP Scanner is its ability to export scan results in various formats, including CSV, TXT, XML, and HTML. This enables administrators and analysts to share and analyze scan data easily.

The command-line interface of the tool also makes it suitable for use in scripts and automated tasks.

NetStumbler

NetStumbler is a hacking and vulnerability testing tool which is compatible with any Windows OS to easily track wireless networks.

Its main features include checking the settings for the network, locating weak-coverage areas in a WLAN, finding the source of wireless interferences in WLANs, finding rogue access points, and assist in finding right direction for antenna aiming for long-distance WLAN connectivity. It can be also used by amateurs to hunt for open networks commonly known as war driving.

SQLMap

SQLMap is an open-source, cross-platform tool used for identifying and exploiting SQL Injection vulnerabilities. It is designed to work with various database engines such as, Oracle, IBM DB2, MS SQL Server, MS Access, MySQL, Sybase, SAP MaxDB, PostgreSQL, Firebird and SQLite.

SQLMap is a free ethical hacking tool that supports most SQL injection techniques like, time-based blind, Boolean-based blind, error-based, UNION query, out-of-bound queries and stacked queries. It is a powerful tool that automates the process of finding and exploiting SQL Injection vulnerabilities, making it a popular choice for security researchers and penetration testers.

A.2: DoS (Denial of Service) or DDoS (Distributed-Denial-of-Service) Attack Tools

Denial of Service also commonly referred to as “DoS” attack is a form of attack that obstructs authorized users from accessing a resource, such as a network, emails, or a website. A Distributed Denial of Service (DDoS) attack happens when multiple compromised machines target the same victim. In such cases, the computer network is overwhelmed with data packets, making it challenging for the targeted system to manage genuine traffic. There are several tools available that can be used to launch DDoS attack against a target server. This list contains both free and commercial tools commonly used for DDoS attacks.

DDoS-Attack

“DDoS-Attack” is a tool that enables launching of distributed denial of service attacks, while also helping to locate and identify DDoS activities by tracking event logs from multiple sources.

Its main features include real-time responses, the ability to filter specific time periods, IPs or other factors, and identification of command and control server communication. It can also assist in detecting malicious communication between the command and control server.

LOIC (Low Orbit ION Cannon)

This is a free, open-source tool programmed in C# to execute DDoS attacks by sending requests to servers through HTTP, TCP, and UDP protocols. It is a useful tool for testing network

performance, and its functionality can aid in identifying DDoS applications that may target a network of computers during DF investigations.

The tool offers several features, including the ability to launch online DDoS attacks against any website controlled by the user. The IP address of the attacker is not concealed, even if a proxy server is not being used. Additionally, LOIC can be employed to perform stress tests to ensure system stability.

HOIC (High Orbit ION cannon)

This is a free tool used to execute Denial-of-Service attacks on multiple URLs using HTTP. The tool offers the flexibility for the user by providing choice for the number of threads to be used in the attack, and provides low, medium, and high levels of attack control.

One of the main features of HOIC is the capability to attack up to 256 DDoS-protected websites simultaneously. It also includes a performance counter to measure the output of the attack. Furthermore, HOIC can be ported to Linux or Mac OS, enhancing its compatibility and versatility.

DDoSIM (D-DoS Simulator)

This is a simulator program built in C++, designed to perform “distributed denial-of-service” attacks (DDoS attacks) on a targeted server. It is compatible with the Linux operating system and is useful in determining a server's ability to withstand DDoS attacks that are application-specific in nature.

DDoSIM offers several key features, such as establishing full TCP connections with the server that is a target, providing multiple configurations for network attacks, and the ability to flood TCP connections on random network ports. These features enable the user to customize the DDoS attack and simulate real-life scenarios, allowing them to test the server's ability to withstand such attacks.

Tor’s Hammer

This is an online DDoS tool used to launch application-layer DDoS attacks on web servers and web applications. It works by simulating browser-based internet requests needed for loading

web pages. Additionally, the tool supports the use of Markdown for creating rich text markup, allowing users to format their content easily.

The key features of Tor's Hammer include the automatic conversion of URLs into links, the ability to create a large number of network connections, thereby utilizing web server resources. It also makes it easy to link other artifacts in a project and hold HTTP POST requests and connections for 1000 to 30000 seconds.

By using this tool, users can test their web servers and web applications' ability to withstand DDoS attacks and identify potential vulnerabilities. Overall, Tor's Hammer is a useful tool for network security testing and can assist in maintaining the security and stability of web-based applications.

RUDY

R-U-Dead-Yet, or RUDY for short, is a freely available DDoS attack tool that can be used to launch DDoS attacks against web applications. One of the key features of RUDY is its ability to target cloud based applications by depleting the number of available sessions on a web server that is under attack. It also has an automatic detection feature that can detect form fields for data submission on the target website, making it easier to initiate attacks.

Additionally, RUDY allows users to conduct “HTTP DDoS” attacks using long-form field submissions, which can lead to resource depletion on the target server. The tool also features an interactive console menu that makes it easy to navigate and customize the attack settings to suit the user's needs.

While RUDY is relatively simple compared to other DDoS attack tools, it is still a potent weapon in the hands of attackers and can cause significant damage to web applications. As such, it is important for website owners and security professionals to take necessary precautions and employ countermeasures to protect against such attacks.

GoldenEye

GoldenEye is a Python-based DDoS tool that uses HTTP requests to initiate attacks on a server. It makes use of Cache-control options and a KeepAlive message to prevent socket connection termination, and utilizes all HTTP/S sockets available on the target server. The tool also randomizes GET and POST commands for mixed traffic. Other features of GoldenEye include

a user-friendly interface, custom user agent creation, and its effectiveness in performing DDoS attacks.

DAVOSET

DAVOSET is a free command-line tool designed to carry out DDoS attacks by exploiting website features. It can launch distributed denial-of-service attacks and supports the use of cookies. DAVOSET also allows for attacks using XML external entities, making it effective against applications that parse XML input.

HULK (HTTP Unbearable Load King)

HULK is a free DDoS attack tool that can specifically attack targets web servers by generating high volumes of traffic. It is capable of bypassing cache servers and helps in generating unique network traffic. The tool can be used for research purposes with ease.

PyLoris

PyLoris is a software tool used for executing online Distributed Denial of Service (DDoS) attacks to test network vulnerabilities. It allows users to manage DDoS attacks and control concurrent connections that are poorly managed. This tool is compatible with Mac OS, Windows, and Linux operating systems, making it accessible to a wider audience.

PyLoris offers some latest options with a maximum limit of 50 threads. Every thread can support 10 connections each. It also features an easy-to-use GUI, enabling users to launch attacks using “HTTP request” headers. Additionally, PyLoris has the latest codebase, ensuring that it is up to date and can be run using Python scripts. These features make it a powerful and versatile tool for network security testing.

A.3: Tools for Damage Control and Protection

Trend Micro House Call

This is a powerful security solution for safeguarding computer systems against various threats. It provides protection against fake financial, banking, and shopping apps, as well as ransomware. With its advanced scanning capabilities, Trend Micro can detect phishing URLs and dangerous websites.

In addition to its robust security features, Trend Micro also offers privacy protection, enabling you to keep your personal information safe. It also provides online backup storage, allowing you to save copies of important documents and files. With its website filtering capabilities, you can easily control which websites other users can access. Lastly, Trend Micro is highly effective at detecting and preventing key-loggers.

Trend Micro Apex One

Trend Micro Apex One is an excellent software that provides top-tier protection for computer systems against a variety of threats, such as fake banking apps, financial apps, shopping apps, and ransomware. The software offers highly effective protection against phishing URLs and malicious websites.

Users can control website access and protect their privacy while also securely storing copies of files and documents on backup storage provided online. This application is an ideal solution for safeguarding your computer against viruses, spywares, and other malwares.

AdGuard Anti-spyware

It's a highly effective antispyware solution which provides exceptional security and is free. Its advanced technology notifies users when it detects any malicious website, thereby protecting them from spyware. With AdGuard, you can enjoy safer and faster browsing as it blocks malicious websites and advertisements. It also enables you to restrict access to inappropriate content for other users. The software is an excellent tool for blocking ads in browsers and apps.

BullGuard Internet Security Tool

BullGuard is a powerful security tool that uses advanced machine learning capabilities to provide comprehensive protection. Its multi-layered approach can identify and block potentially harmful behaviours of malwares before they can cause damage to systems. The software also prevents any applications containing malware from accessing your system.

One of the notable features of BullGuard is its custom-built browser protection that ensures a safer browsing experience. It helps to reduce resource usage on the system and improve performance. The software offers ultimate identity protection and can be used on Android and macOS devices.

Norton Power Eraser Anti-Spyware

Norton Power Eraser is a powerful application that provides exceptional protection against spywares and online threats. The software can quickly scan your programs for malware, spyware, ransomware, and other harmful software that may slow down your system.

With Norton Power Eraser, you can enjoy enhanced protection against digital threats and a greater online privacy is ensured. The software allows user to backup and restore contact information and can help users find their lost device. Additionally, Norton Power Eraser is an excellent tool for safeguarding personal data from malicious websites.

Panda Free Anti-virus (Panda Dome Free)

This free tool provides real-time protection against viruses and spyware, making it one of the best in its category for free anti-virus and anti-spyware protection. It also offers protection against malware execution from infected USB disks. With this tool, you can enjoy uninterrupted multimedia content viewing, and it is one of the best applications for recovering infected systems. It can prevent malware infections by blocking execution from USB drives automatically. The software can also perform battery optimisation and is easy to install. It supports Windows, Android, Mac, and iOS platforms.

360 Total Security Antivirus

360 Total Security is a highly effective application for cleaning your system or device of file-viruses, as well as detecting spyware and providing real-time protection against external threats or malware. The software uses cloud technology to detect viruses and ransomware variants in real-time. Among its many features, 360 Total Security provides protection against viruses and malware, secures online shopping, protects your privacy, and cleans junk and plugins. With this tool, you can also achieve network performance optimization, schedule a scan, and check for Wi-Fi security. It is one of the best tools available for protecting your computer systems from malware attacks, phishing attacks, and other security threats.

SUPER Anti Spyware

SUPER Anti Spyware is a powerful application designed to track and remove spyware and malware. It is an effective anti-spyware program that can block both known and emerging threats. This lightweight software won't slow down your PC and is easy to use. With its robust

features, SUPER Anti-Spyware can protect systems against trojans, ransomware, adware, and more, all while keeping your system running smoothly. The software is particularly good for real-time protection thanks to its updated database.

Outbyte PC Repair

Outbyte PC Repair is a powerful PC optimization tool designed for Windows systems. With this tool, you can quickly assess your computer's performance and detect any performance issues that might be slowing down your PC.

The software's main features include the removal of unused temporary or cached files, the ability to prioritize specific apps for CPU processor time, and the option to disable Windows telemetry features to safeguard your data's privacy. Additionally, Outbyte PC Repair can help you to resolve hundreds of common PC issues, ensuring that your system runs smoothly. Supported exclusively on Windows platforms, Outbyte PC Repair is an effective solution for optimizing your PC's performance and maintaining its health.

Malwarebytes

Malwarebytes stands out as the top-notch malware removal tool that offers impeccable protection to devices against adware, ransomware, malware, and vicious websites. With the ability to detect and block approximately 8 million threats in a single day, Malwarebytes excels in detecting threats and effectively cleaning up infected devices.

Apart from its excellent malware removal capabilities, this software has several other features that make it a preferred choice among users. For instance, it can alert administrators when the networks of an organisation, servers, or any website is found infected. Additionally, it also allows to identify all endpoints on the network, making it easier to manage and control your networked devices. The software also provides a centralized management facility, which is particularly useful for enterprise-level users.

Another notable feature of this software is its ability to offer a safer browsing experience, thanks to its free malware protection capabilities. Additionally, Malwarebytes can perform a privacy audit of applications installed, ensuring that your sensitive data remains protected at all times.

Malwarebytes is available in multiple languages, including English, Italian, German, French, and many more, making it accessible to users across the globe. Lastly, the software is highly effective in detecting and removing adware, making it an excellent choice for those who are tired of intrusive pop-up ads and unwanted software.

HitmanPro

HitmanPro is a powerful malware removal tool that is designed to detect and eliminate malware from your system. One of the unique features of this software is that it can be used without the need for installation, which means users can run it from a USB drive or CD/DVD.

The application is capable of scanning for bad behavior and can effectively identify and remove malware from the system. HitmanPro can bring damaged resources to safe conditions and provide with the best malware protection available in the market.

The software also features a deep scan capability that can be performed before the operating system boots up, ensuring that all malware is detected and removed even before it can cause any harm. Additionally, HitmanPro boasts a “quick scan” feature to check only those parts of the system that are infected, which can save a lot of time.

This anti-malware software is equipped to remove various types of malicious software, including rootkits, viruses, spyware, and adware. With HitmanPro, users can be assured that their system is protected from all types of malware, and they can use devices with confidence.

Malware Hunter

Malware Hunter is a top-notch anti-malware tool that is available for free and can effectively detect and remove malware from the system, protecting it from potential danger. With its automatic updates, the real-time protection feature of this software keeps computer up-to-date and secure at all times.

The software allows user to scan your system quickly, ensuring that any malware present is detected and removed promptly. Malware Hunter provides real-time protection against most known malwares, ensuring that the system is always protected from the latest threats.

One of the key features of Malware Hunter is its interface which is very user-friendly, which allows for easy navigation for users to operate the software effectively. The intuitive interface makes it accessible to users of all levels of technical expertise.

In addition to its malware removal capabilities, Malware Hunter also protects user privacy, ensuring that sensitive data remains secure. The software is designed to protect from all types of threats. Overall, Malware Hunter is an effective anti-malware tool that provides real-time protection against malware threats, and its intuitive interface makes it an ideal choice for users who want a user-friendly malware protection solution.

Malwarefox

This is a robust computer protection tool which provides powerful protection against all major malwares. The software offers effective rootkit protection and also repair damaged files, ensuring that the system remains secure and free from threats. Additionally, Malwarefox offers real-time threat prevention, providing continuous protection against the latest threats.

One of the notable features of Malwarefox is its browser cleanup facility, which helps to remove unwanted and malicious extensions, ensuring a safe and secure browsing experience. The software offers 24x7 day protection, which means the system is always protected, regardless of the time of day. Malwarefox is easy to install, and the process is hassle-free, ensuring that user can get started with the software quickly.

Another benefit of Malwarefox is its lightweight nature, which means that it does not occupy much space on the PC, ensuring that the system remains fast and efficient. Overall, Malwarefox is an excellent choice for users who want an easy-to-use, lightweight anti-malware tool that offers robust protection against malware threats.

iolo System Mechanic

It is a powerful software that utilizes “behaviour monitoring” techniques to detect and remove any malware the system. The software is specifically designed to search for malicious programs and eliminate them, ensuring that the system remains secure and free from threats.

One of the software's key features is its malware analysis using cloud-based services, which allows it to detect the new threats and offer real-time protection. With constant updates of the

latest threat data, iolo System Mechanic ensures that the system is continuously protected from the latest malware threats.

iolo System Mechanic features an interface that is user-friendly, making it easy for users to operate the software effectively. The software is an excellent choice for online privacy protection, ensuring that sensitive data remains secure.

Another notable feature of iolo System Mechanic is its built-in feature to erase PC hard drive for data privacy, ensuring that user data is completely wiped and cannot be recovered by anyone else. This feature is particularly useful if users plan to sell or dispose of their computer.

Overall, iolo System Mechanic is a robust software that offers excellent malware detection and removal capabilities, and its user-friendly interface and advanced privacy features make it an ideal choice for users who want a comprehensive system maintenance and protection solution.

CloudFlare

CloudFlare is a content delivery network (CDN) that provides a range of security features to protect websites and online applications against various threats. Its DDoS protection is particularly noteworthy, as it uses multiple techniques, including rate limiting and IP reputation analysis, to mitigate DDoS attacks.

The “Web Application Firewall” or WAF provided by CloudFlare is also an essential feature that helps protect websites against a range of application-layer attacks, including SQL injection, cross-site scripting (XSS), and cross-site request forgery (CSRF). The WAF uses the “collective intelligence” of the entire CloudFlare network to identify and block malicious traffic, providing effective protection against evolving threats.

CloudFlare also offers secure registration of domains to help prevent domain hijacking, which is a common tactic used by cybercriminals to gain access to sensitive information. Additionally, rate limiting is another critical feature that helps block visitors with abnormal request rates and showing suspicious activity, which can help prevent brute force attacks and other forms of online abuse.

CloudFlare has another unique feature called Orbit that helps prevent and resolve security issues for IoT (Internet of Things) devices. Orbit provides a secure communication channel

between IoT devices and the CloudFlare network, enabling devices to benefit from the security features of CloudFlare, such as DDoS protection and the WAF.

Outbyte PC Repair

Outbyte PC Repair is a PC optimizer tool designed for Windows that provides a quick performance overview of your computer. It helps users identify and resolve performance issues that may be affecting their PCs.

The main features of Outbyte PC Repair include the ability to identify and remove unused temporary or cached files, prioritizing specific apps for CPU processor time, and controlling the privacy of your data by disabling Windows telemetry features. Additionally, it helps to solve hundreds of the most common PC issues. Outbyte PC Repair is supported on the Windows platform.

Restoro

Restoro is a powerful software tool that effectively cleans the registry and recovers valuable storage space, thereby preventing freezing and crashing of computer systems. By securing and repairing computer file systems, this application can optimize systems with ease.

Among its main features, Restoro is capable of fixing error messages, repairing damages caused by viruses, providing convenient PC repair anytime, repairing and rebuilding Windows OS, restoration of DLL files, detecting websites that are suspected to be dangerous, freeing up disk space from system, and identification of apps that pause threats before they can cause harm to computer system.

Iolo’s SystemMechanic

Iolo's SystemMechanic is a computer optimization software that aims to improve computer performance by identifying and fixing various issues. One of its notable features is its ability to generate personalized reports that outline problem details and recommended solutions.

SystemMechanic offers several features to improve computer speed and performance. It can automatically optimize the RAM, CPU, and hard drive to improve speed. It can also identify and remove unwanted start-up programs that slow down the system.

The software is capable of finding and repairing over 30,000 different issues that can impact computer performance. It can also perform hidden Internet settings optimization to improve page load and download speeds. Additionally, it can safely wipe browsers history and patch security vulnerabilities in Windows OS that could potentially harm the computer.

Another useful feature of SystemMechanic is its ability to remove over 50 different types of junk and temporary files, freeing up valuable disk space. This can help improve computer performance and reduce clutter.

TotalAV Antivirus

TotalAV is a software application that provides antivirus and malware removal features to protect household devices from various cyber threats. The software offers real-time protection against malware, spyware, adware, and ransomware. It's designed to be user-friendly and easy to use, providing comprehensive security for both desktop and mobile devices.

TotalAV's main features include a remote firewall that allows users to access their devices remotely, “real-time protection” from deadly viruses, a password vault that is highly secure, and the ability to remove viruses, Trojans, and malware. The software's important feature known as Ad-Block Pro also helps ensure safe browsing by blocking ads and pop-ups and providing a VPN for added security.

TotalAV is available on multiple platforms, including Windows, iOS, and Android, making it a versatile option for households with different devices. However, it's important to note that while the software offers a free version, its functionality is limited, and users may need to upgrade to the premium version for full protection. As with any antivirus software, it's also essential to keep TotalAV updated and use other security measures, such as avoiding suspicious websites and using strong passwords.

Advanced System Protector

Advanced System Protector is a program designed to safeguard any computer from malware. Its primary function is to scan the system for any malware infections and eliminate them. In addition, it provides several features to enhance computer security. For instance, it can help secure user data by detecting and removing browser history and cookies stored in the browser, thus protecting privacy of the user.

It can also isolate files that seems suspicious in order to prevent any further infection from spreading. Another advantage of using Advanced System Protector is that it operates without consuming much system resources, ensuring that it does not slow down computer while scanning it.

IObit Malware Fighter

IObit is a user-friendly software that can identify malware, while also optimizing, cleaning, and speeding up the system. With this application, users can protect their data and online privacy.

Some of its key features include real-time threat blocking, browser and privacy protection, stopping malicious processes running in RAM, improving system security and refreshing web browsing with IObit Malware Fighter, resolving drive errors and repairing Windows, and easily removing leftover software with just one click.

Avast

Avast is a highly regarded malware removal tool that provides effective protection against viruses and other malicious software. Its advanced capabilities can successfully block all potential threats on a computer, and it can help identify browser vulnerabilities. Among its key features, this free tool is capable of detecting and preventing the installation of malicious software at an early stage.

Avast provides protection against links on the web that are infected by malware and conducts regular vulnerability scans to detect potential threats. It also verifies the security of networks with Wi-Fi routers to ensure that they are safe from unauthorized access. Avast is compatible with Mac, Android, and iOS operating systems.

SpyHunter

SpyHunter is a versatile tool designed to scan and eliminate a wide range of malicious software, including spyware, viruses, ransomware, trojans, and worms. This application offers users the ability to schedule scans on a daily, monthly, or weekly basis. Customizable scans can be performed according to user preferences, and the tool can analyse memory, cookies, and registry. SpyHunter provides 24/7 customer support and also prevents malicious objects from

modifying users' internet connection. With these features, SpyHunter ensures comprehensive protection against various forms of malware.

GridinSoft Anti-Malware

GridinSoft is a software designed to protect computers from malware. It is capable of scanning systems for malicious threats, ensuring that they remain clean and secure. This anti-malware tool is particularly useful for Windows 10 users, and comes with a range of features that make it stand out. For instance, user can perform unlimited scans for potential threats, and the software includes a start-up guard that can help speed up the computer start-up process. Additionally, GridinSoft can safeguard browsers against phishing websites, and prevent annoying ads from popping up. The software also has the ability to clear automatic tracking cookies, providing user with a more private browsing experience.

Bytefence

Bytefence is a software designed to safeguard computers from malicious software. Considered among the top-notch malware removal tools, it offers real-time protection, ensuring users are shielded from unwanted programs. The program is equipped to detect and eliminate harmful trojans, spyware, and worms, providing round-the-clock protection. Additionally, users can effortlessly scan their PCs by simply by a click.

Clamav

This software designed to detect and identify trojans, viruses, malwares, and other harmful programs and is available as open-source. It offers a command-line utility, allowing users to scan files as per their requirement. The malware scanner is equipped with many signature languages to handle multiple file formats and, ensuring comprehensive protection. Its user-friendly interface makes it easy to navigate, and users can quickly scan files to detect potential threats. Additionally, the software permits users to scan emails, providing enhanced security. The program is designed to facilitate easy updating of the database, ensuring optimal protection against evolving threats.

AVG's FreeAntivirus

With its user-friendly interface and no cost to users, this application is an Anti-Spyware program that offers a range of features. AVG's FreeAntivirus can safeguard your PC from many

viruses, malwares, and spywares. It can easily detect and provide one-click solutions to various threats.

The program's main features include the ability to perform automatic scans on-demand basis, which can be scheduled daily, weekly, or based on specific criteria. Additionally, the software automatically checks for malware before downloading to your device. It can also wipe clean the contents of your hard drive and lock your device for added data and privacy protection. The app provides SMS spamming protection and offers security updates in real-time.

One of its notable features is the ability to prevent unsafe email attachments, links, and downloads. It also serves as an effective tool for scanning web content, emails, and SMS for any hidden malware.

FreeAntivirus from Avast

Avast's Free Antivirus is capable of safeguarding various types of Android devices against prevalent viruses and malware. Additionally, it offers password management services to enhance web security. The software is equipped to counteract a majority of potential threats to computer systems.

Some key features of the application include straightforward vulnerability and threat scanning, automatic detection of malicious software prior to installation, identification of malware-infected web links, and excellent anti-spyware, anti-malware, and anti-virus detection capabilities, making it one of the best in its class.

Advanced SystemCare Free Cleaner

Advanced SystemCare Free is a user-friendly software designed to help optimize and protect computer system from various malware threats by cleaning it up. It not only improves system performance but also ensures online privacy by removing browsing history and frees up valuable disk space by deleting unnecessary files.

Some of the key features of Advanced SystemCare Free include the ability to manage startup items to boost performance, increase system security, remove web browsing history, repair

errors in Windows, resolve drive errors, fix Windows issues, remove leftover software files, and resolve drive errors with just a single click.

Total Security's Bitdefender

Total Security's Bitdefender is a highly effective anti-virus tool that offers both on-demand and on-install scanning options to protect your system from a range of threats. Additionally, this application safeguards your personal information and ensures online privacy.

The software boasts several noteworthy features, including the ability to locate an Android device remotely if it is lost or stolen, verify mail account breaches, and have minimal impact on battery life of the device. It can respond quickly to threats without slowing down PC performance and also provides round-the-clock security updates. Users can conveniently browse while enjoying VPN protection. Additionally, it is a reliable tool for secure online banking.

APPENDIX-B

B.1: Visual Representation of Dataset of Cybercrime Incidents

```
from pandas import read_csv
from pandas.plotting import scatter_matrix
from matplotlib import pyplot
# Load dataset
url = "https://raw.githubusercontent.com/deep0025/CyberCrimeData-SA/main/cybercrime-sample-dataset.csv"
names = ['Phishing', 'ZeroDayAttack', 'RansomwareAttack', 'ChildPorn', 'City']
dataset = read_csv(url, names=names)
# box and whisker plots
dataset.plot(kind='box', subplots=True, layout=(2,2), sharex=False, sharey=False)
pyplot.show()
# histograms
dataset.hist()
pyplot.show()
# scatter plot matrix
scatter_matrix(dataset)
pyplot.show()
```

Figure B.1: Python Implementation Code for Analysis of cybercrime incidents (dataset)

B.2: Complete Implementation Code and Output for Baseline 'Zero Rule Algorithm' in Python for Classification of 'Critical' and 'Non-Critical' Status of Cybercrime Incidents

```
#Testing Baseline zero rule algorithm for prediction of classification
from pandas import read_csv
from random import seed
from random import randrange
from csv import reader

# Load a CSV file
def load_csv(filename):
    dataset = list()
    with open(filename, 'r') as file:
        csv_reader = reader(file)
        for row in csv_reader:
            if not row:
                continue
            dataset.append(row)
    return dataset

# Convert string column to float
def str_column_to_float(dataset, column):
    for row in dataset:
        row[column] = float(row[column].strip())

# Split a dataset into a train and test set
def train_test_split(dataset, split):
    train = list()
    train_size = split * len(dataset)
    dataset_copy = list(dataset)
    while len(train) < train_size:
        index = randrange(len(dataset_copy))
        train.append(dataset_copy.pop(index))
    return train, dataset_copy

# Calculate accuracy percentage
def accuracy_metric(actual, predicted):
    correct = 0
```

Figure B.2: Complete implementation code for Baseline 'Zero Rule Algorithm' in Python

```
    for i in range(len(actual)):
        if actual[i] == predicted[i]:
            correct += 1
    return correct / float(len(actual)) * 100.0

# Evaluate an algorithm using a train/test split
def evaluate_algorithm(dataset, algorithm, split, *args):
    train, test = train_test_split(dataset, split)
    test_set = list()
    for row in test:
        row_copy = list(row)
        row_copy[-1] = None
        test_set.append(row_copy)
    predicted = algorithm(train, test_set, *args)
    actual = [row[-1] for row in test]
    accuracy = accuracy_metric(actual, predicted)
    return accuracy

# zero rule algorithm for classification
def zero_rule_algorithm_classification(train, test):
    output_values = [row[-1] for row in train]
    prediction = max(set(output_values), key=output_values.count)
    predicted = [prediction for i in range(len(test))]
    return predicted

# Test the train/test harness
seed(1)
#load and prepare data
filename = 'CybercrimeDataset_SeverityScore_Class.csv'
dataset = load_csv(filename)
for i in range(len(dataset[0])):
    str_column_to_float(dataset, i)

# evaluate algorithm
split = 0.6
accuracy = evaluate_algorithm(dataset, zero_rule_algorithm_classification, split)
print('Accuracy: %.3f%%' % (accuracy))
```

```
Accuracy: 45.000%
```

Figure B.2 Contd.: Complete Implementation Code and Output for Baseline ‘Zero Rule Algorithm’ in Python.

B.3: Implementation code in Python for spot checking AI algorithms along with program Output using datset of 100 records

```
# Spot Check Algorithms
models = []
models.append(('LR', LogisticRegression(solver='liblinear', multi_class='ovr')))
models.append(('LDA', LinearDiscriminantAnalysis()))
models.append(('KNN', KNeighborsClassifier()))
models.append(('CART', DecisionTreeClassifier()))
models.append(('NB', GaussianNB()))
models.append(('SVM', SVC(gamma='auto')))
# evaluate each model in turn
results = []
names = []
for name, model in models:
    kfold = StratifiedKFold(n_splits=10, random_state=1, shuffle=True)
    cv_results = cross_val_score(model, X_train, Y_train, cv=kfold, scoring='accuracy')
    results.append(cv_results)
    names.append(name)
    print('%s: %f (%f)' % (name, cv_results.mean(), cv_results.std()))
```

```
LR: 0.637500 (0.171847)
LDA: 0.675000 (0.178536)
KNN: 0.600000 (0.145774)
CART: 0.525000 (0.165831)
NB: 0.612500 (0.152582)
SVM: 0.625000 (0.136931)
```

Figure B.3: Implementation code in Python for spot checking AI algorithms with cybercrime dataset of 100 records along with program Output

B.4: AI algorithm comparison represented as Box and Whisker Plot

```
# Compare Algorithms
pyplot.boxplot(results, labels=names)
pyplot.title('Algorithm Comparison')
pyplot.show()
```

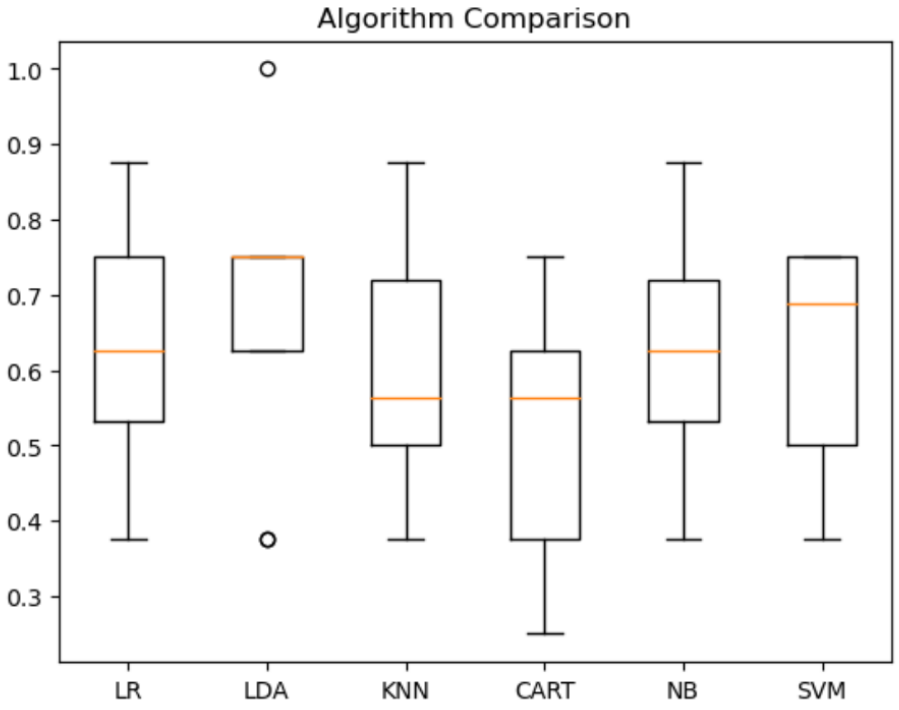


Figure B.4: Box and whisker plot showing algorithm comparison based on performance

B.5: Complete Implementation Code in Python for AI Algorithm Comparison and Evaluation

Complete Python Program Code for Comparing AI Algorithms using a dataset of 100 records, along with split-out validation of dataset, spot-check algorithm and evaluation of each model and finally compare the results in given in figure B.5.

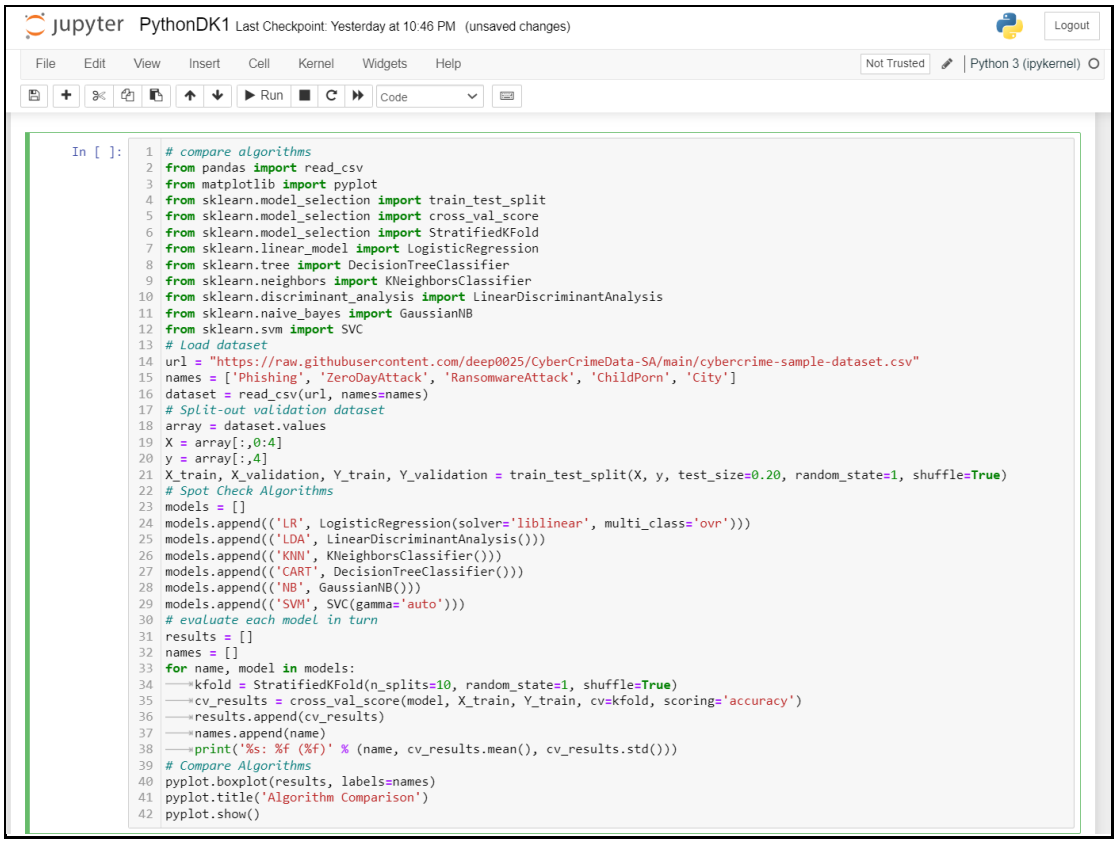


Figure B.5: Complete Python Program Code for Comparing AI Algorithms

B.6: Performance Evaluation of SVC Algorithm Cybercrime Dataset using Data Validation Techniques (Code and Output)

Figure B.6: Accuracy Score, Confusion Matrix and performance parameters analysis of SVC Algorithm using validated dataset

B.7: Output of AI model Comparison of Spot-check and cross validation

```
LR: 0.637500 (0.171847)
LDA: 0.675000 (0.178536)
KNN: 0.600000 (0.145774)
CART: 0.537500 (0.148429)
NB: 0.612500 (0.152582)
SVM: 0.625000 (0.136931)
```

Figure B.7: Output of Accuracy Scores and Standard Deviation of Comparison of AI models

B.8: Output of Visual comparison of Spot-check and Cross Validation of AI algorithm

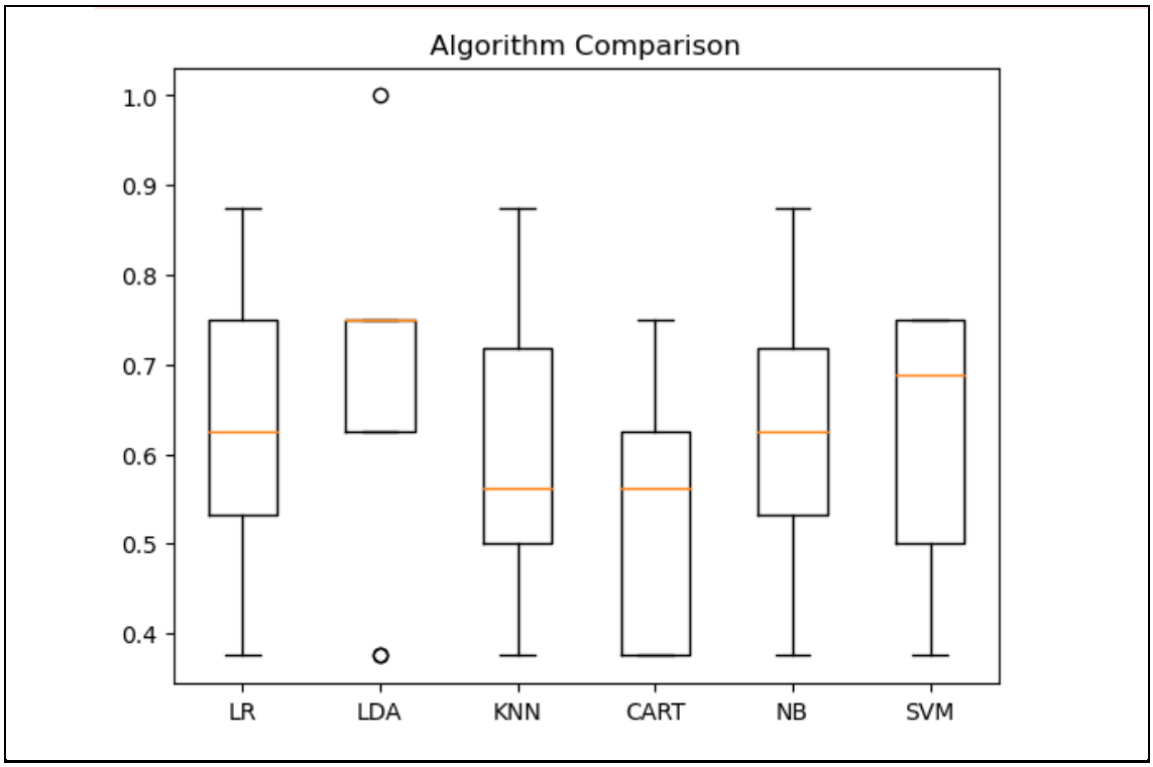


Figure B.8: Visual Comparison of AI algorithm using Box and Whisker Plot

B.9 Python code implementation for 'train and test split' resampling of validation dataset

```
# Split-out validation dataset
array = dataset.values
X = array[:,0:4]
y = array[:,4]
X_train, X_validation, Y_train, Y_validation = train_test_split(X, y, test_size=0.20, random_state=1)

```

Figure B.9: Code implementation for 'train and test split' resampling of validation dataset

B.10: Python implementation code for Pipeline Optimisation using TPOT and AutoML

```
from pandas import read_csv
from sklearn.preprocessing import LabelEncoder
from sklearn.model_selection import RepeatedStratifiedKFold
from tpot import TPOTClassifier
# load dataset
url = 'https://raw.githubusercontent.com/deep0025/CyberCrimeData-SA/main/cybercrime-sample-dataset
dataframe = read_csv(url, header=None)
# split into input and output elements
data = dataframe.values
X, y = data[:, :-1], data[:, -1]
# minimally prepare dataset
X = X.astype('float32')
y = LabelEncoder().fit_transform(y.astype('str'))
# define model evaluation
cv = RepeatedStratifiedKFold(n_splits=10, n_repeats=3, random_state=1)
# define search
model = TPOTClassifier(generations=5, population_size=50, cv=cv, scoring='accuracy', verbosity=2,
# perform the search
model.fit(X, y)
# export the best model
model.export('tpot_cybercrime_data100_best_model.py')
```

Figure B.10: Implementation code for Pipeline Optimisation using TPOT and AutoML

B.11: Program Output of Pipeline Optimisation Progress using TPOT and AutoML

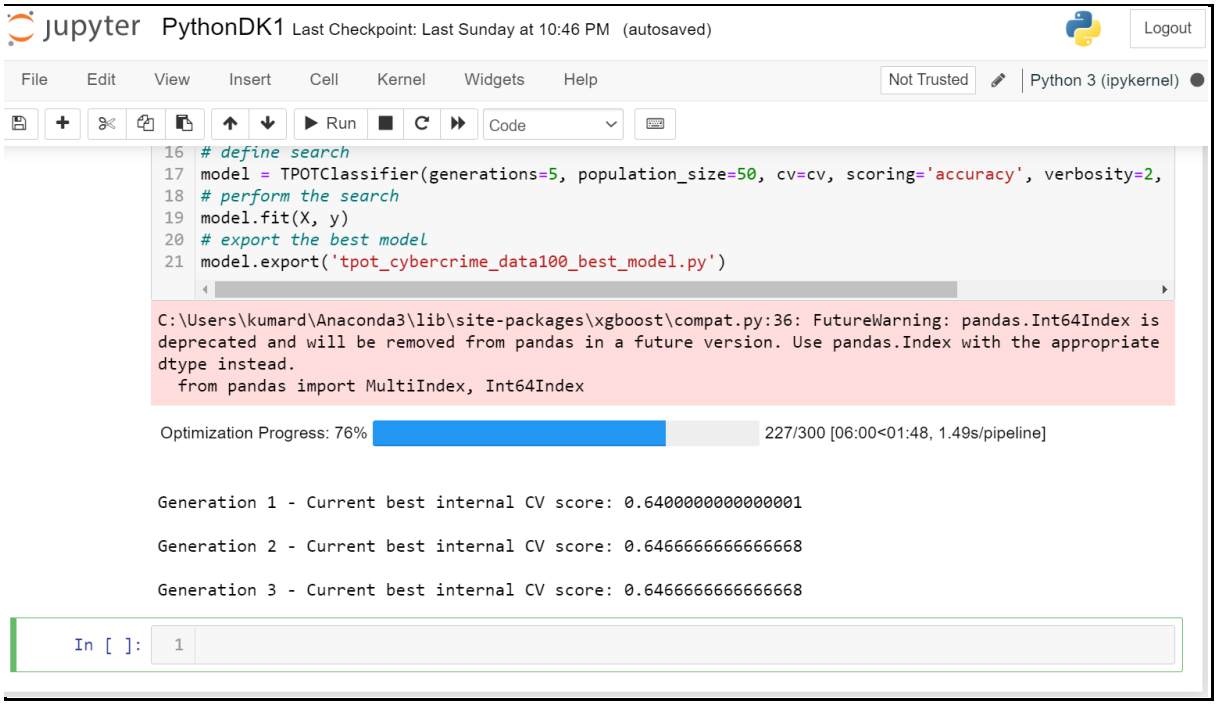


Figure B.11: Output of program showing 'Pipeline Optimisation' progress using TPOT and AutoML

B.12: Python Implementation Code and Output for Customised Algorithm Created using ANN and Stochastic Gradient Descent for Prediction Analysis of Cybercrime Dataset

```
# ANN Backprop Algorithm for the Cybercrime Dataset Predictive Modelling
from random import seed
from random import randrange
from random import random
from csv import reader
from math import exp

# Load a CSV file
def load_csv(filename):
    dataset = list()
    with open(filename, 'r') as file:
        csv_reader = reader(file)
        for row in csv_reader:
            if not row:
                continue
            dataset.append(row)
    return dataset

# Convert string column to float
def str_column_to_float(dataset, column):
    for row in dataset:
        row[column] = float(row[column].strip())

# Convert string column to integer
def str_column_to_int(dataset, column):
    class_values = [row[column] for row in dataset]
    unique = set(class_values)
    lookup = dict()
    for i, value in enumerate(unique):
        lookup[value] = i
    for row in dataset:
        row[column] = lookup[row[column]]
    return lookup

# Find the min and max values for each column
def dataset_minmax(dataset):
    return [[min(column), max(column)] for column in zip(*dataset)]

# Rescale dataset columns to the range 0-1
def normalize_dataset(dataset, minmax):
    for row in dataset:
        for i in range(len(row)-1):
            row[i] = (row[i] - minmax[i][0]) / (minmax[i][1] - minmax[i][0])

# Split a dataset into k folds
def cross_validation_split(dataset, n_folds):
    dataset_split = list()
    dataset_copy = list(dataset)
    fold_size = int(len(dataset) / n_folds)
    for _ in range(n_folds):
        fold = list()
        while len(fold) < fold_size:
            index = randrange(len(dataset_copy))
            fold.append(dataset_copy.pop(index))
        dataset_split.append(fold)
    return dataset_split

# Calculate accuracy percentage
def accuracy_metric(actual, predicted):
    correct = 0
    for i in range(len(actual)):
        if actual[i] == predicted[i]:
            correct += 1
    return correct / float(len(actual)) * 100.0

# Evaluate an algorithm using a cross validation split
def evaluate_algorithm(dataset, algorithm, n_folds, *args):
    folds = cross_validation_split(dataset, n_folds)
    scores = list()
    for fold in folds:
        train_set = list(folds)
        train_set.remove(fold)
        train_set = sum(train_set, [])
        test_set = list()
        for row in fold:
            row_copy = list(row)
            test_set.append(row_copy)
            row_copy[-1] = None
        predicted = algorithm(train_set, test_set, *args)
        actual = [row[-1] for row in fold]
```

Figure B.12: Python Implementation Code and Output for Customised Algorithm Created using ANN and Stochastic Gradient Descent for Prediction Analysis of Cybercrime Dataset

```
        accuracy = accuracy_metric(actual, predicted)
        scores.append(accuracy)
    return scores

# Calculate neuron activation for an input
def activate(weights, inputs):
    activation = weights[-1]
    for i in range(len(weights)-1):
        activation += weights[i] * inputs[i]
    return activation

# Transfer neuron activation
def transfer(activation):
    return 1.0 / (1.0 + exp(-activation))

# Forward propagate input to a network output
def forward_propagate(network, row):
    inputs = row
    for layer in network:
        new_inputs = []
        for neuron in layer:
            activation = activate(neuron['weights'], inputs)
            neuron['output'] = transfer(activation)
            new_inputs.append(neuron['output'])
        inputs = new_inputs
    return inputs

# Calculate the derivative of an neuron output
def transfer_derivative(output):
    return output * (1.0 - output)

# Backpropagate error and store in neurons
def backward_propagate_error(network, expected):
    for i in reversed(range(len(network))):
        layer = network[i]
        errors = list()
        if i != len(network)-1:
            for j in range(len(layer)):
                error = 0.0
                for neuron in network[i + 1]:
                    error += (neuron['weights'][j] * neuron['delta'])
                errors.append(error)
        else:
            for j in range(len(layer)):
                neuron = layer[j]
                errors.append(expected[j] - neuron['output'])
        for j in range(len(layer)):
            neuron = layer[j]
            neuron['delta'] = errors[j] * transfer_derivative(neuron['output'])

# Update network weights with error
def update_weights(network, row, l_rate):
    for i in range(len(network)):
        inputs = row[:-1]
        if i != 0:
            inputs = [neuron['output'] for neuron in network[i - 1]]
        for neuron in network[i]:
            for j in range(len(inputs)):
                neuron['weights'][j] += l_rate * neuron['delta'] * inputs[j]
            neuron['weights'][-1] += l_rate * neuron['delta']

# Train a network for a fixed number of epochs
def train_network(network, train, l_rate, n_epoch, n_outputs):
    for _ in range(n_epoch):
        for row in train:
            forward_propagate(network, row)
            expected = [0 for i in range(n_outputs)]
            expected[row[-1]] = 1
            backward_propagate_error(network, expected)
            update_weights(network, row, l_rate)

# Initialize a network
def initialize_network(n_inputs, n_hidden, n_outputs):
    network = list()
    hidden_layer = [{'weights':[random() for i in range(n_inputs + 1)]} for i in range(n_hidden)]
    network.append(hidden_layer)
    output_layer = [{'weights':[random() for i in range(n_hidden + 1)]} for i in range(n_outputs)]
    network.append(output_layer)
    return network
```

Figure B.12 Contd.: Python Implementation code for Customised Algorithm Created using ANN and Stochastic Gradient Descent

```
# Make a prediction with a network
def predict(network, row):
    outputs = forward_propagate(network, row)
    return outputs.index(max(outputs))

# Backpropagation Algorithm With Stochastic Gradient Descent
def back_propagation(train, test, l_rate, n_epoch, n_hidden):
    n_inputs = len(train[0]) - 1
    n_outputs = len(set([row[-1] for row in train]))
    network = initialize_network(n_inputs, n_hidden, n_outputs)
    train_network(network, train, l_rate, n_epoch, n_outputs)
    predictions = list()
    for row in test:
        prediction = predict(network, row)
        predictions.append(prediction)
    return(predictions)

# Test Backprop on Cybercrime DataSet
seed(1)
# load and prepare data
filename = 'CybercrimeDataset_SeverityScore_Class.csv'
dataset = load_csv(filename)
for i in range(len(dataset[0])-1):
    str_column_to_float(dataset, i)
# convert class column to integers
str_column_to_int(dataset, len(dataset[0])-1)
# normalize input variables
minmax = dataset_minmax(dataset)
normalize_dataset(dataset, minmax)
# evaluate algorithm
n_folds = 5
l_rate = 0.3
n_epoch = 500
n_hidden = 5
scores = evaluate_algorithm(dataset, back_propagation, n_folds, l_rate, n_epoch, n_hidden)
print('Scores: %s' % scores)
print('Mean Accuracy: %.3f%%' % (sum(scores)/float(len(scores))))
```

```
Scores: [95.0, 95.0, 100.0, 100.0, 95.0]
Mean Accuracy: 97.000%
```

Figure B.12 Contd.: Python Implementation code for Customised Algorithm Created using ANN and Stochastic Gradient Descent for Prediction Analysis of Cybercrime Dataset (100 records)

B.13: Customised Baseline Convolutional Neural Network Model for Image Classification

```
# baseline model with data augmentation for the dogs vs cats dataset
import sys
from matplotlib import pyplot
from keras.utils import to_categorical
from keras.models import Sequential
from keras.layers import Conv2D
from keras.layers import MaxPooling2D
from keras.layers import Dense
from keras.layers import Flatten
from keras.optimizers import SGD
from keras.preprocessing.image import ImageDataGenerator

# define cnn model
def define_model():
 model = Sequential()
 model.add(Conv2D(32, (3, 3), activation='relu', kernel_initializer='he_uniform', padding='same', input_shape=(200, 200, 3))
 model.add(MaxPooling2D((2, 2)))
 model.add(Conv2D(64, (3, 3), activation='relu', kernel_initializer='he_uniform', padding='same'))
 model.add(MaxPooling2D((2, 2)))
 model.add(Conv2D(128, (3, 3), activation='relu', kernel_initializer='he_uniform', padding='same'))
 model.add(MaxPooling2D((2, 2)))
 model.add(Flatten())
 model.add(Dense(128, activation='relu', kernel_initializer='he_uniform'))
 model.add(Dense(1, activation='sigmoid'))
 # compile model
 opt = SGD(lr=0.001, momentum=0.9)
 model.compile(optimizer=opt, loss='binary_crossentropy', metrics=['accuracy'])
 return model

# plot diagnostic learning curves
def summarize_diagnostics(history):
 # plot loss
 pyplot.subplot(211)
 pyplot.title('Cross Entropy Loss')
 pyplot.plot(history.history['loss'], color='blue', label='train')
 pyplot.plot(history.history['val_loss'], color='orange', label='test')
 # plot accuracy
 pyplot.subplot(212)
 pyplot.title('Classification Accuracy')
 pyplot.plot(history.history['accuracy'], color='blue', label='train')
 pyplot.plot(history.history['val_accuracy'], color='orange', label='test')
 # save plot to file
 filename = sys.argv[0].split('/')[-1]
 pyplot.savefig(filename + '_plot.png')
 pyplot.close()
```

```

# run the test harness for evaluating a model
def run_test_harness():
 # define model
 model = define_model()
 # create data generators
 train_datagen = ImageDataGenerator(rescale=1.0/255.0,
 width_shift_range=0.1, height_shift_range=0.1, horizontal_flip=True)
 test_datagen = ImageDataGenerator(rescale=1.0/255.0)
 # prepare iterators
 train_it = train_datagen.flow_from_directory('dataset_dogs_vs_cats/train/',
 class_mode='binary', batch_size=64, target_size=(200, 200))
 test_it = test_datagen.flow_from_directory('dataset_dogs_vs_cats/test/',
 class_mode='binary', batch_size=64, target_size=(200, 200))
 # fit model
 history = model.fit_generator(train_it, steps_per_epoch=len(train_it),
 validation_data=test_it, validation_steps=len(test_it), epochs=50, verbose=0)
 # evaluate model
 _, acc = model.evaluate_generator(test_it, steps=len(test_it), verbose=0)
 print('> %.3f' % (acc * 100.0))
 # learning curves
 summarize_diagnostics(history)

# entry point, run the test harness
run_test_harness()
```

Figure B.13: Customised Convolutional Neural Network Model for Image Classification

B.14: Enhanced Model of CNN Algorithm using VGG for Image Classification

```
# save the final model to file
from keras.applications.vgg16 import VGG16
from keras.models import Model
from keras.layers import Dense
from keras.layers import Flatten
from keras.optimizers import SGD
from keras.preprocessing.image import ImageDataGenerator

# define cnn model
def define_model():
 # load model
 model = VGG16(include_top=False, input_shape=(224, 224, 3))
 # mark loaded layers as not trainable
 for layer in model.layers:
 layer.trainable = False
 # add new classifier layers
 flat1 = Flatten()(model.layers[-1].output)
 class1 = Dense(128, activation='relu', kernel_initializer='he_uniform')(flat1)
 output = Dense(1, activation='sigmoid')(class1)
 # define new model
 model = Model(inputs=model.inputs, outputs=output)
 # compile model
 opt = SGD(lr=0.001, momentum=0.9)
 model.compile(optimizer=opt, loss='binary_crossentropy', metrics=['accuracy'])
 return model

# run the test harness for evaluating a model
def run_test_harness():
# define model
 model = define_model()
 # create data generator
 datagen = ImageDataGenerator(featurewise_center=True)
 # specify imagenet mean values for centering
 datagen.mean = [123.68, 116.779, 103.939]
 # prepare iterator
 train_it = datagen.flow_from_directory('finalize_dogs_vs_cats/',
 class_mode='binary', batch_size=64, target_size=(224, 224))
 # fit model
 model.fit_generator(train_it, steps_per_epoch=len(train_it), epochs=10, verbose=0)
 # save model
 model.save('final_model.h5')

# entry point, run the test harness
run_test_harness()
```

Figure B.14: Enhanced Model of CNN with 3 VGG Algorithm for Image Classification

B.15: Python Implementation Customised Algorithm created using ANN and Stochastic Gradient Descent for Prediction Analysis of Cybercrime Dataset using 300 records

```
168
169 # Backpropagation Algorithm With Stochastic Gradient Descent
170 def back_propagation(train, test, l_rate, n_epoch, n_hidden):
171     n_inputs = len(train[0]) - 1
172     n_outputs = len(set([row[-1] for row in train]))
173     network = initialize_network(n_inputs, n_hidden, n_outputs)
174     train_network(network, train, l_rate, n_epoch, n_outputs)
175     predictions = list()
176     for row in test:
177         prediction = predict(network, row)
178         predictions.append(prediction)
179     return(predictions)
180
181 # Test Backprop on Cybercrime DataSet
182 seed(1)
183 # load and prepare data
184 filename = 'CybercrimeDataset_SeverityScore_Class300.csv'
185 dataset = load_csv(filename)
186 for i in range(len(dataset[0])-1):
187     str_column_to_float(dataset, i)
188 # convert class column to integers
189 str_column_to_int(dataset, len(dataset[0])-1)
190 # normalize input variables
191 minmax = dataset_minmax(dataset)
192 normalize_dataset(dataset, minmax)
193 # evaluate algorithm
194 n_folds = 5
195 l_rate = 0.3
196 n_epoch =500
197 n_hidden = 5
198 scores = evaluate_algorithm(dataset, back_propagation, n_folds, l_rate, n_epoch, n_hidden)
199 print('Scores: %s' % scores)
200 print('Mean Accuracy: %.3f%%' % (sum(scores)/float(len(scores))))
```

```
Scores: [100.0, 96.66666666666667, 98.33333333333333, 98.33333333333333, 98.33333333333333]
Mean Accuracy: 98.333%
```

Figure B.15: Python Implementation code for Customised Algorithm Created using ANN and Stochastic Gradient Descent for Prediction Analysis of Cybercrime Dataset (300 records)

APPENDIX-C: ETHICAL CLEARANCE PROTOCOL

Mr Deepak Kumar (212561392)
School Of Man Info Tech &Gov
Westville

Dear Mr Deepak Kumar,

Original application number: 00016755
Project title: The Design and development of an AI based Digital Forensic Protocol for First Responders

Exemption from Ethics Review

In response to your application received on , your school has indicated that the protocol has been granted **EXEMPTION FROM ETHICS REVIEW.**

Any alteration/s to the exempted research protocol, e.g., Title of the Project, Location of the Study, Research Approach and Methods must be reviewed and approved through an amendment/modification prior to its implementation. The original exemption number must be cited.

For any changes that could result in potential risk, an ethics application including the proposed amendments must be submitted to the relevant UKZN Research Ethics Committee. The original exemption number must be cited.

In case you have further queries, please quote the above reference number.

PLEASE NOTE:
Research data should be securely stored in the discipline/department for a period of 5 years.

I take this opportunity of wishing you everything of the best with your study.

[redacted]

Dr Bongani Reginald Qwabe
Academic Leader Research
School Of Man Info Tech &Gov