2018

Automated Identification of Digital Evidence across Heterogeneous Data Resources

Mohammed, Hussam J

http://hdl.handle.net/10026.1/12839

http://dx.doi.org/10.24382/693
University of Plymouth

AUTOMATED IDENTIFICATION OF DIGITAL EVIDENCE ACROSS HETEROGENEOUS DATA RESOURCES

by

HUSSAM JASIM MOHAMMED

A thesis submitted to the University of Plymouth in partial fulfilment for the degree of

DOCTOR OF PHILOSOPHY

School of Computing, Electronics and Mathematics

November 2018

COPYRIGHT STATEMENT

Acknowledgements

First and foremost, I would like to thank Allah (God) Almighty for giving me the strength, knowledge, ability, and opportunity to undertake this research study and to persevere and complete it satisfactorily. Without his blessings, this achievement would not have been possible.

I would like to express my appreciation and gratitude to my supervisor Prof. Dr Nathan Clarke for his continuous support, interest, patience, and guidance throughout my studies. Thanks must also go to my other supervisor, Dr Fudong Le, who has spent a lot of time proofreading papers and my thesis, in addition to providing helpful experience and guidance throughout my studies.

My acknowledgement would be incomplete without thanking the biggest source of my strength, my family. Thank you for encouraging me in all of my pursuits and inspiring me to follow my dreams. I am especially grateful to my parents (Jasim and Bushra) for their support and never-ending love.

My unreserved love, thanks, and appreciation must go to my wife (Yasameen) and my daughter (Mayar) who have been very patient, understanding, and inspiring to me throughout this endeavour, spending days, nights, and sometimes even holidays without me. I hope the potential success of this research will compensate some of what they have missed. May Allah bless them.

Many thanks to my friends Leith Abed, Yaseen Alheety, Ahmed Suhail, Ayad Al-Adhami, and Omar Alsaiari for their support and for the motivating ideas and thoughts they provided during my PhD journey.

Finally, I would like to acknowledge, with thanks and appreciation, the government of Iraq and the Higher Committee for Education Development in Iraq, for granting me a scholarship and sponsoring my PhD studies.

Author's Declaration

At no time during the registration for the degree of Doctor of Philosophy has the author been registered for any other University award without prior agreement of the Doctoral College Quality Sub-Committee.

Work submitted for this research degree at the University of Plymouth has not formed part of any other degree either at the University of Plymouth or at another establishment.

This study was financed with the aid of a scholarship from the Iraqi Government. Relevant seminars and conferences were attended at which work was often presented and several papers prepared for publication

1. Mohammed, H. J., Clarke, N., & Li, F. (2016). An Automated Approach for Digital Forensic Analysis of Heterogeneous Big Data. Journal of Digital Forensics, Security and Law, 11(2), 137-152.
2. Mohammed, H. J., Clark, N. L., & Li, F. (2018). Automating the harmonisation of heterogeneous data in digital forensics. In 17th European Conference on Cyber Warfare and Security (pp. 299-306). Academic Conferences and Publishing International Limited.
3. Mohammed, H., Clarke, N., & Li, F. (2018). Evidence identification in heterogeneous data using clustering. In Proceedings of the 13th International Conference on Availability, Reliability and Security (p. 35). ACM.

Word count of main body of thesis: 37,100 words

Signed…………………………………………

Date…………………………………………

Abstract

Automated Identification of Digital Evidence across Heterogeneous Data Resources

Hussam J. Mohammed

Digital forensics has become an increasingly important tool in the fight against cyber and computer-assisted crime. However, with an increasing range of technologies at people's disposal, investigators find themselves having to process and analyse many systems with large volumes of data (e.g., PCs, laptops, tablets, and smartphones) within a single case. Unfortunately, current digital forensic tools operate in an isolated manner, investigating systems and applications individually. The heterogeneity and volume of evidence place time constraints and a significant burden on investigators. Examples of heterogeneity include applications such as messaging (e.g., iMessenger, Viber, Snapchat, and WhatsApp), web browsers (e.g., Firefox and Google Chrome), and file systems (e.g., NTFS, FAT, and HFS). Being able to analyse and investigate evidence from across devices and applications in a universal and harmonized fashion would enable investigators to query all data at once. In addition, successfully prioritizing evidence and reducing the volume of data to be analysed reduces the time taken and cognitive load on the investigator.

This thesis focuses on the examination and analysis phases of the digital investigation process. It explores the feasibility of dealing with big and heterogeneous data sources in order to correlate the evidence from across these evidential sources in an automated way. Therefore, a novel approach was developed to solve the heterogeneity issues of big data using three developed algorithms. The three algorithms include the harmonising, clustering, and automated identification of evidence (AIE) algorithms.

The harmonisation algorithm seeks to provide an automated framework to merge similar datasets by characterising similar metadata categories and then harmonising them in a single dataset. This algorithm overcomes heterogeneity issues and makes the examination and analysis easier by analysing and investigating the evidential artefacts across devices and applications based on the categories to query data at once. Based on the merged datasets, the clustering algorithm is used to identify the evidential files and isolate the non-related files based on their metadata. Afterwards, the AIE algorithm tries to identify the cluster holding the largest number of evidential artefacts through searching based on two methods: criminal profiling activities and some information from the criminals themselves. Then, the related clusters are identified through timeline analysis and a search of associated artefacts of the files within the first cluster.

A series of experiments using real-life forensic datasets were conducted to evaluate the algorithms across five different categories of datasets (i.e., messaging, graphical files, file system, internet history, and emails), each containing data from different applications across different devices. The results of the characterisation and harmonisation process show that the algorithm can merge all fields successfully, with the exception of some binary-based data found within the messaging datasets (contained within Viber and SMS). The error occurred because of a lack of information for the characterisation process to make a useful determination. However, on further analysis, it was found that the error had a minimal impact on subsequent merged data. The results of the clustering process and AIE algorithm showed the two algorithms can collaborate and identify more than 92% of evidential files.

Table of Contents

List of Figures

List of Tables

List of Algorithms

1 Introduction

1.1 Introduction

Digital forensics has become commonplace and has gained importance as a result of the increasing prevalence of technology over the last few years. The rapid development in technology, such as the volume of data and cloud computing environments, has relevance regarding criminal activity. The efficient organizations share analysis approaches of huge volumes of data to gain information to support their work and serve their customers. These data are generated from transaction records of online purchases, video, audio, images, emails, logs, posts, search queries, health records, social networking interactions, science data, sensors, mobile phones etc. (Sagiroglu and Sinanc, 2013).

Cloud computing and big databases are increasingly used by governments, companies, and other users for storing huge amounts of information. In addition, increasing interest in the use of cloud computing services presents both opportunities for criminal exploitation and challenges for forensic investigation owing to a lack of support (Cheng et al., 2013). Therefore, the rapid development of technology has brought various challenges to digital forensics. This development, including the variety of devices, operating systems (OS), files, and applications, increases the complexity, diversity, and correlation issues within forensic analysis (Garfinkel, 2006). Conducting a forensic analysis of a case containing multi-resources and applications can be difficult owing to the heterogeneity of the evidence across those devices. In general, investigators normally take each device and examine it individually using an existing forensic tool to understand the nature and relationship of artefacts. Unfortunately, these tools were designed to work on a single forensic image with specific data types (e.g., a workstation or a smartphone) (Cahyani et al., 2017). Consequently, the forensic tools are currently struggling to deal with individual cybercrime cases that have a larger size of evidence (e.g., between 200 GB and 2 TB of data) (Casey, 2011). However, the volume of data that needs to be analysed can range from several terabytes to a few petabytes.

With the significant increase in computing, individuals have increasingly come to own several devices (e.g., PCs, laptops, tablets, and smartphones) with each using different applications across various platforms (Bennett, 2012). Additionally, companies producing electronic devices have to choose an operating system, either open-source or commercial, for their core technology (Almunawar, 2018). Consequently, the file structure is formatted according to the OS and results in a variety of files across various Oss, such as NTFS, FAT, HFS, and Ext4 (Tanenbaum, 2009). Several applications can also run on one platform and achieve similar purposes, such as web browsers (e.g., Google Chrome, Mozilla Firefox, and Apple's Safari) and messaging (e.g., SMS, Viber, and WhatsApp). However, being able to examine and analyse data from across many systems and applications at once based on a data category is currently impossible.

Data categories, including files, databases, documents, pictures, media files, web browsers, etc., hold valuable information that can be used to answer some of the key questions of a forensic investigation. Examples of these questions concern who did something to a file, when they did it, and where it was carried out. Although a wide range of forensic tools and techniques exist both commercially and via open source (including Encase, AccessData FTK, and Autopsy), they only extract and analyse metadata for certain types of systems and applications (Ayers, 2009).

Recently, several researchers have tried to use metadata within the digital forensics domain to reconstruct past events. Metadata describes the attributes of any files or applications in most digital resources (Guptill, 1999); it provides rich information about files that can lead to files being processed using metadata instead of the files themselves (Raghavan, 2014). Digital forensic cases can include several categories of similar metadata within a single forensic image or across multiple resources, resulting in repeating the forensic process many times and increasing the investigator's workload. However, the automated correlation between the evidential artefacts from various sources is currently impossible.

1.2 Research Aims and Objectives

The aim of this research is to develop a novel framework to deal with heterogeneous data resources for forensic examinations and analyses. The novel framework will be developed to provide a robust, modular, and automated system for digital forensic analysis of heterogeneous data. Additionally, the automated system will assist investigators to reduce the complexity of undertaking data examination and analysis process. To achieve this, the following research objectives are established:

- Develop a current state-of-the-art understanding of heterogeneous data frameworks and digital forensics, including the challenges and available solutions.

- Investigate the examination and analysis techniques that are provided by various fields, such as within the huge volume of data domain and in the digital forensics area.
- Seek to evaluate the extent to which current forensic techniques can be applied.
- Establish the current state of the art in terms of the key issues of heterogeneous data within the forensic field including critical criteria in order to establish the requirements for a novel model.
- Develop a novel approach to solve the heterogeneity issue related to the big forensic data and identify the evidence in an automated way.
- Test the developed approach with data collected from a public domain and real-world cases.
- Validate the functionality and the accuracy of the developed approach based on its results.

1.3 Thesis Summary

The thesis is organised into eight chapters. Chapter One introduces the problem of digital forensics in the heterogeneous data environment and its relation to technology, as well as how the growth of data with heterogeneity affects digital forensic investigations.

Chapter Two presents a comprehensive view of digital forensic science in terms of the fundamental concept, the main processes, and challenges. It discusses the main methodologies of digital forensics that are used for conducting digital investigations.

In addition, this chapter investigates the common challenges faced by investigators with the rapid development of cybercrime.

Chapter Three provides a literature review of the existing research in volume and heterogeneity of data within the forensic field and discusses a number of open problems in the chosen domain.

Chapter Four demonstrates an automated approach for analysing and merging datasets. This approach seeks to provide a fusion of similar metadata categories across multiple and heterogeneous resources. A series of experiments using real-life forensic datasets was conducted.

Chapter Five presents a clustering approach to identify evidential files and isolate non-related files based on their metadata. A series of experiments using real-life forensic cases was conducted to evaluate the approach. The experimental results were analysed and discussed in detail to show the impact of various factors on the output.

Chapter Six shows a developed algorithm to identify evidential files in an automated way based on the data within clusters. A series of experiments based the four cases was achieved to validate the proposed algorithm.

Chapter Seven is the final chapter. It presents the conclusions arising from the research and highlights the key contributions, achievements, and limitations. It also contains a discussion on potential areas for future research.

2 Digital Forensics

2.1 Introduction

During the past thirty years, new types of crimes have become popular with the digital revolution. Traditional methods of forensic science could not deal with these crimes; therefore, digital forensics was introduced. Digital forensics is a field of forensic science that depends on technologies and tools to collect and analyse data that is seized from digital devices. According to Garfinkel (2010), digital forensics is roughly 40 years old; the golden age was from 1999 until 2007. Over that time, digital forensics became attractive because there were few types of OSs, few database systems, and a single type of machine to be investigated. Nowadays, cybercrime cases, however, consist of heterogeneous machines with large volumes of data. Digital forensics can be used to review the past by recovering deleted files and restoring deleted emails and messages. Consequently, a range of definitions has been presented, such as the following definition by the Digital Forensics Research Workshop (DFRW):

> *"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" (Palmer, 2001).*

Digital evidence comprises significant information stored in digital devices that supports and proves how a cybercrime occurred. It also proves the relationship between the hacker and crime (Casey, 2011).

An effective digital forensic investigation to identify evidence can be characterised by the following (Marziale, 2009):

- Reliability: the digital evidence is precise and free from manipulation. In addition, it is seized properly without any additions or changes to the original media.
- Comprehensiveness: the analysis phase analyses all potential artefacts in a particular case as much as possible by using sophisticated techniques and tools to view artefacts under investigation and find specific evidence.
- Efficiency: investigators make the best use of constrained resources, which include processing power, time, and data storage. As most cases are complicated, effective investigation requires a mechanism to analyse artefacts through several layers.
- Coherence: the digital evidence, which is a result of the analysis phase, can be used to prove the relationship between the crime and its victim or hacker in order to be accepted in court.

2.2 Digital Forensics Process

The digital forensics process should be flexible for all systems rather than being limited to a specific one. Many models of the forensic investigation process have been developed since 1984; each model has its own methodology that tries to deal with incidents by executing certain steps. The computer forensic investigation process is the earliest method of forensic investigation and was proposed by Pollitt (1995). It is composed of four main phases: acquisition, identification, evaluation, and admission. In 2001, the DFRW was held with the purpose of exploring a model that could be used by both academics and practitioners to find evidence from digital systems during forensic investigations (Ademu et al., 2011). This model is the most robust and popular approach. It consists of seven stages: identification, preservation, collection, examination, analysis, presentation, and decision.

Carrier and Spafford (2004) proposed an Event-based Digital Forensic Investigation Framework that can be applied to digital crimes based on the cause and effect of incidents in order to document physical crime scenes. This framework consisted of five phases: readiness (which includes operations and infrastructural readiness), deployment (which includes detection and notification, as well as confirmation and authorization of the investigation), physical crime scene investigation (physical collection of evidence), digital crime scene investigation (examining the digital data for evidence), and presentation (presenting the results of the investigation). Perumal (2009) suggested a digital forensic model, based on Malaysian cyber law, covering both technical and legal aspects of the forensic investigation. This model has seven major stages: planning, identification, reconnaissance, analysis, result, proof and defence, and diffusion of information.

Agarwal et al. (2011) proposed the Systematic Digital Forensic Investigation Model (SRDFIM), which can be used by investigators and organizations to help them follow appropriate procedures in a methodical way during the investigation. It is made up of eleven steps: preparation, securing the scene, survey, recognition, documenting the scene, communication shielding, evidence collection, preservation, examination, analysis, presentation, and result and review. As demonstrated above, most investigators follow six main steps during an investigation. These steps are shown in Figure 2-1.

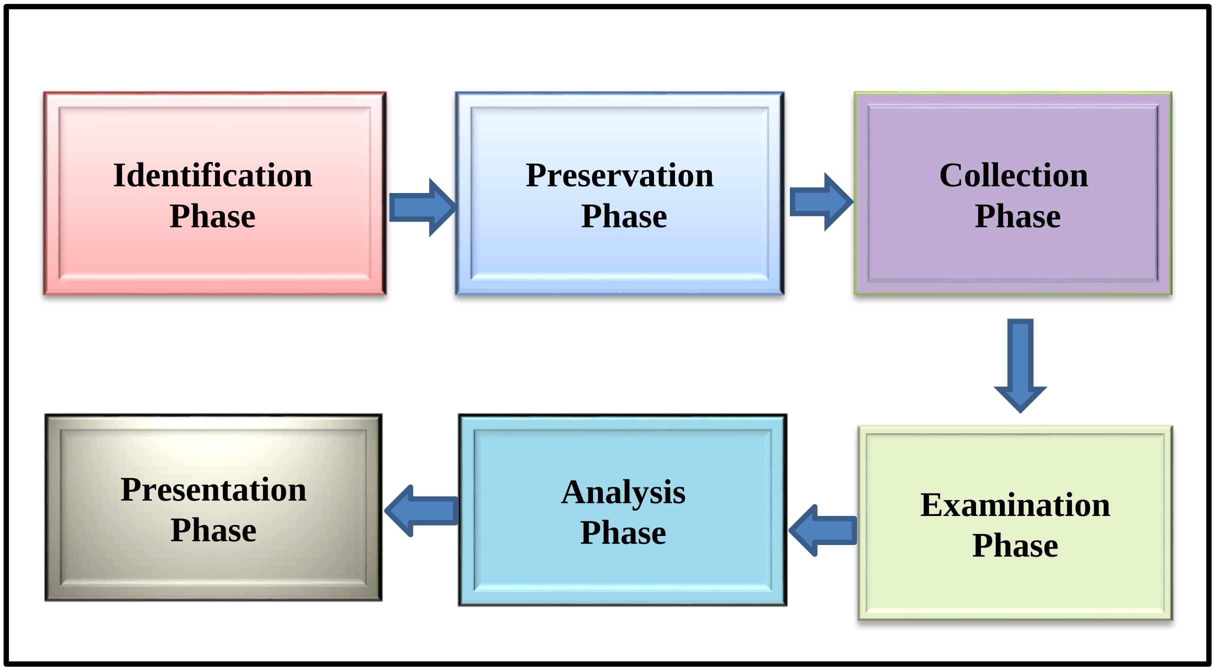


Figure 2-1:The Digital Forensics Process (Ademu et al., 2011)

- **Identification Phase**

According to Kent et al. (2006), identification is the first step in the investigation process; it includes searching, identifying, and documenting potential sources of data that might contain evidence. These data sources can be computers, servers, database systems, networks, or storage devices. The identification phase is mission critical, as all other subsequent steps of the investigation depend on it.

- **Preservation Phase**

Digital preservation is used to ensure that digital objects remain unmodified or are changed as little as possible. Investigators should use tools and techniques to recover and preserve data from digital devices in a forensically sound manner. A method used to preserve and verify the integrity of data is the hash function (e.g., MD5 and SHA1) (Martini and Choo, 2012).

- **Collection Phase**

The collection phase refers to the physical collection of data. This collection is conducted by using forensic tools to copy all information from a suspect's device to a trusted device to produce a forensic image. There are two ways of acquiring data: dead (physical) or live (logical) acquisition. A dead acquisition is to obtain data from the suspect's system without the assistance of its OS. A live acquisition is to copy information from the suspect's system while the OS is still running (Carrier, 2005).

- **Examination Phase**

The aim of this phase is to find evidence by filtering and reducing the amount of data via various forensic tools. This data may exist in different forms, such as files, images, videos, databases or hidden folders in the system. Well-established tools, such as the AccessData's FTK and Encase, are essential to the investigation as investigators rely on them to find evidence in a timely fashion during this phase (Carrier, 2005).

- **Analysis Phase**

The analysis phase is the important step of an investigation where the relevant artefacts of evidence are evaluated. The examination phase outputs contain relevant artefacts that are analysed to draw conclusions, as well as to correlate evidence with the incident (Kent et al., 2006). According to Clarke (2010), digital forensic analysis could be conducted in two ways: dead analysis and live analysis. In the dead analysis, the data of a forensic image is analysed by a trusted forensic system, where the integrity of data is preserved and never changed. Live forensic analysis is used to collect and analyse evidence from a suspect's machine while its OS is still running.

- **Presentation Phase**

The output of the presentation phase is a report that is conducted by the investigator summarising and explaining the conclusion of the investigation. Moreover, the investigator should present the tools and techniques used during the process and how they were used (Benredjem, 2007).

2.3 Challenges of Digital Forensics

In the previous section, although most models have similar steps, there is no standard model of digital forensic investigation to deal with digital crime. Several researchers have attempted to find a substantial model; however, they have mainly focused on technical aspects without focusing on legal matters, or vice versa. In addition, the number of cyber crimes increases every year and has become more complicated. Therefore, many challenges have risen with the domain, from the volume of data to the heterogeneity of environments.

2.3.1 Limitation of Digital Forensic Tools

Digital forensic tools are significant in the digital world as they can be used to improve the security of stored data. Richard and Roussev (2006) stated that investigators have used various tools to aid them in preserving and analysing digital evidence, which provide an appropriate environment and are user-friendly for conducting investigations. These tools could be either commercial or open-source. However, because of the great increase in storage capacity and the diversity of data sources, such as network and cloud computing, these digital forensic tools are insufficient because they were designed to be used with one case over a single workstation (Ayers, 2009). Table 2-1 provides comparisons between some of the commercial and open-source forensic tools (Yadav, 2011; Al Fahdi, 2016).

Table 2-1: Comparison of Digital Forensic Tools

	Tools' Features	EnCase	FTK	Sleuth Kit	Liveview
1	User interface	For professional forensics	Easy to use	Easy to use	Easy to use
2	Language interface	Traditional Chinese & English	Simple Chinese & English	English	English
3	Support for image file creation	Yes	Yes	No	Yes
4	Hash value calculating	MD5	MD5	MD5 and SHA-1	MD5 and SHA-1
5	Cost	Expensive (Commercial)	Expensive (Commercial)	Open-source software	Open-source software
6	Support	Graphical disk interface information	Digital evidence classification	Digital evidence classification	Acquired internet history, screen capture, memory
7	Metadata extraction	Only file system metadata	Only file system metadata	Only file system metadata	Memory attributes
8	File system examination	Yes	Yes	Yes	No
9	Memory dumps examination	Yes	Yes	No	Yes
10	Log examination	No	No	No	No
11	Packet capture examination	No	No	No	No
12	Text indexing and search	Yes	Yes	Yes	Yes
13	Identify correlations	No	No	No	No
14	Multiple sources of DE	Can group artefacts using FS metadata, one at a time	Can group artefacts using FS metadata, one at a time	Can group artefacts using FS metadata, one at a time	No

There are various limitations that are related to digital forensic tools, including:

- Processing speed: most of the digital forensic tools are slow when analysing an average volume of evidential data, taking many hours or days to process them. Some instances require a high processing speed because they might be a real risk to public safety (Ayers, 2009).
- Forensic data abstraction: according to Garfinkel (2010), only five types of data abstraction are widely used in a forensic perspective and many attempts to develop a new format and abstraction have failed.
    - Disk image: an image of the whole disk is copied and achieved as raw.
    - Packet capture files: a format is used to capture network traffic.
    - Files: used to recognize documents and images.
    - File signatures as outputs of MD5 and SHA1 hashes.
    - Extracted named entities: classified as ASCII text files or Unicode files such as emails, names, phone numbers, etc.
- Software errors: one major concern of using forensic tools is software errors that might lead to various problems, such as unexplained crashes or loss of work during data analysis. Occasionally, unexplained crashes could be caused by the difficulty to parse data because the input data might be either incompletely or inadequately validated. In addition, some programming languages, such as C and C++, are unsafe coupled with programming errors (Ayers, 2009).
- Planning of analysis phase: most of the forensic tools lack support for plans of how the investigation is performed. The analyst will be responsible for documenting the results of the analysis by using basic tools, such as a pen

and a notebook. As long as the analysis process carried out manually by the analyst, many mistakes could be made (Ayers, 2009).

2.3.2 Increasing Volumes of Data

A common issue with digital forensics investigations is the volumes of data that needs to be processed. This data is constantly increased because of the continuing development of storage technology, including the increasing storage capacity of customer devices and cloud computing services (Quick and Choo, 2014). It should be noted that the data volume being referred here is much smaller than "big data" and it typically refers to datasets in the terabyte range. However, the rapid growth of storage capacity coupled with the increasing number of cyber crimes result in various challenges facing forensic investigation. Alink et al. (2006) stated the challenges of forensic investigation are often in the feature extraction and analysis phases. The storage of modern computer systems is approximately hundreds of gigabytes; however, digital investigation cases could consist of multiple systems where the amount of data for an individual case might reach terabytes.

According to Perry et al. (2009), cloud computing has become one of the most important transformative developments in computing history. A current trend of users is towards using cloud services because they provide massive storage. However, cloud storage services may be used by criminals for illegal purposes; in addition, current forensic tools do not support digital forensic investigation within cloud computing (Quick et al., 2013).

2.3.3 Access of Evidential Sources

The access to the source of evidence is an important step to gain information in order for it to be analysed by investigators. In the past, it was easy to achieve this goal. All the relevant sources could be physically collected at the crime scene. However, the situation has now changed because of big data, especially for cloud environments (Biggs and Vidalis, 2009).

For instance, if several malicious hackers launched a certain attack by scripts residing in or attacking programs in Amazon Elastic Cloud, all the attacking procedures are implemented according to the scripts saved in that Amazon Elastic Cloud account. The first problem concerns where the forensic investigators will find the physical devices, such as hard disks, CDs, etc. As all the attacking behaviours took place in the cloud environment, the evidential data may exist on the Amazon servers or the suspects' account. To access the attacker's account, the confidential login should be identified in advance (Birk and Wegener, 2011). Otherwise, it is impossible to pass the authentication mechanism by Amazon servers. Maybe investigators can collect the evidence in Amazon's data centre in some special cases, such as CIA, FBI, or NSA law enforcement agencies and so on. However, in most circumstances, getting evidential artefacts from service providers is challenging (Budu and Boateng, 2015; Choo et al., 2017).

2.4 Conclusion

The number of criminal activities carried out on digital devices constantly increases each year. Therefore, the science of digital forensics is important to face those issues. As a result, researchers and specialists in digital forensics have invented several tools and methodologies to cope with these criminal activities. Digital forensic tools, such as FTK and Encase, provide an appropriate environment and are user-friendly for conducting investigations, but they have failed to keep pace with the development of technology in recent years. With regard to the methodologies, there is no standard methodology to conduct investigations because each model has steps for treating a particular case. Major challenges of digital forensic tools are the multitude of sources, large volumes of data, and heterogeneous datasets where the current tools are struggling to keep pace in achieving modern forensic investigations. These tools are known as first-generation tools and include some limitations related to processing and analysing huge amounts of data. These issues and others have become even more complicated when investigators deal with cybercrimes over a big and heterogeneous data environment.

3 Digital Forensic Challenges

3.1 Introduction

This chapter presents a comprehensive review of the current state of the art in heterogeneous data within a forensic domain. One major challenge of digital investigation is the increasing volume of data within digital forensic cases. A limited number of researchers have undertaken studies in recent years in relation to heterogeneous data and its challenges within digital forensics. As a result, this chapter focuses on the analysis of two significant issues: the volume of data and its heterogeneity. These issues concerning the analysis of data volume have inspired varied efforts to find solutions. These include artificial intelligence, data clustering, and data reduction. From the perspective of data heterogeneity, there are various potential solutions, such as data integration and data correlation. However, researchers have not yet found a way to overcome all the problems involved. This chapter, therefore, presents an exhaustive review of heterogeneous data forensics and suggests a direction for future developments.

This section presents the methodology for undertaking a comprehensive literature review related to the heterogeneity of huge volumes of forensic data. This covers many aspects, including data acquisition and analysis, artificial intelligence, data clustering, data reduction, database forensics, heterogeneous data and resources, and data correlation. The methodology of the literature review was to search for related publications across a range of different academic databases, such as Springer, IEEE, ScienceDirect, and Google Scholar, by using various keywords (e.g., big data analysis, digital forensic data volume, forensic data mining, digital forensic triage, forensics intelligence, examination of big data, digital forensic challenges, big data acquisition, heterogeneous data, forensics discovery, search in big data correlation, data reduction, and data clustering). The word “forensic” was added to some of these keywords to narrow down the search results.

This chapter was designed in a structured format, starting with a data acquisition and analysis section, followed by artificial intelligent studies, data clustering, and data reduction with hash sets. It then provides a section that discusses database forensics, followed by a section focussing on the heterogeneity of data, and then exploring methods of correlating data in a heterogeneous environment.

3.2 Data Acquisition and Analysis

Recently, digital forensic investigations have faced many challenges and have failed to keep pace with the problems of analysing evidence in large and heterogeneous data. Various solutions and techniques have been suggested for dealing with data analysis, such as artificial intelligence, data mining, data clustering, and data reduction. Artificial intelligence is a process to simulate the human intelligence actions by using machines (Russell and Norvig, 2016). These actions include learning (obtaining information and rules for using the information), and reasoning (utilising rules to identify approximate conclusions). While machine learning is a field of artificial intelligence that can be used by computer systems to learn from data without constant supervision from the human. Machine learning algorithms can be classified into four categories: supervised learning (training an algorithm with labelled data), unsupervised (dividing data without prior knowledge), semi-supervised (it falls between completely labelled training data and without any labelled training data), and reinforcement learning (continuously learning from the environment to determine the ideal behaviour within a specific context) (Burrell, 2016). However, unsupervised algorithms are commonly used among the others within the digital forensic domain as there is no prior knowledge about data within the digital forensic cases.

To acquire huge data properly, Xu et al. (2013) proposed a big data acquisition engine that merges a rule engine and finite state automaton to solve the issue of big data acquisition. They reported that the rule engine was used to maintain big data collection, determine problems, and discover the reason for breakdowns while finite state automation was used to describe the state of big data acquisition. They demonstrated that five steps need to be followed. The first is to create a global Java Expert System Shell (JESS) engine, which is a rule engine and scripting environment written entirely in Java language; it is responsible for rule-based loading and loading/changing a rule base that is defined according to demand. After that, data acquisition is achieved by integrating the JESS rule engine and data automation through two processes: the device interaction module and the acquisition server automation. The device interaction module connects directly to a device during acquisition and each device interaction module corresponds to an acquisition server. The acquisition server is responsible for gathering and transmitting the data collected from the device interaction module for analysis and display. Fourthly, the engine executes match rules and, finally, the export process results, as shown in Figure 3-1. Generally, this combination facilitates the acquisition process within a big data environment and provides a flexible way of verifying the security and correctness of the acquisition. However, the rule engine is pre-defining decision, which can work on a specific type of data. It must be updated with each dataset to make the right decision, thereby causing a heavy burden on the system. In addition, this approach is theoretical, and neither evolution nor implementation was conducted.

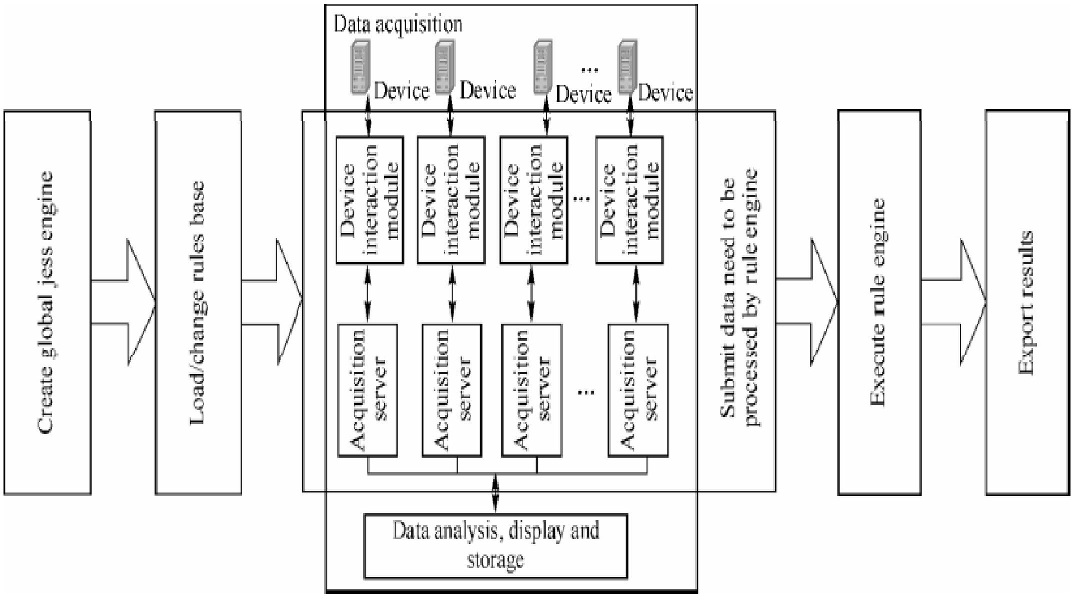

Data acquisition
Device
Device
Device
Device
...
Create global jess engine
Load/change rules base
Device interaction module
Device interaction module
Device interaction module
Device interaction module
...
Acquisition server
Acquisition server
Acquisition server
Acquisition server
...
Submit data need to be processed by rule engine
Execute rule engine
Export results
Data analysis, display and storage

Figure 3-1: Big Data Acquisition Engine (Xu et al., 2013)

In attempting to deal with data analysis, Noel and Peterson (2014) acknowledged the major challenges involved in finding relevant information in digital forensic investigations were as a result of an increasing volume of data. They propose the use of latent Dirichlet allocation (LDA), which is a method of natural language processing. LDA works to minimize practitioners' overhead in two ways. First, it extracts hidden subjects from documents and offers summary details of contents with a minimum of human intervention. Second, it offers the possibility of isolating relevant information and documents based on easy keyword selection in the search. Their work performs three comparison tests between LDA and current regular expression search techniques by using real data corpus (RDC). RDC is a set of disks extracted from the storage devices of 2,432 disks from 25 different countries, where each storage device is between 8 MB to 480 GB in size, including data from USB drives, phones, flash cards, and multi-partition hard drives. The first test evaluated three information retrieval tasks on passport files, legal documents, and power generation documents. The results showed that LDA was capable of retrieving documents of higher relevance. However, LDA took much longer than a regular expression does, approximately eight hours compared to approximately one minute. The second test demonstrated a corpus-trained LDA model for browsing all the important documents by automatically arranging and sub-dividing various document collections. The regular expression search method could be used later to find a specific document from the various sub-divided collections. A final test applied LDA as a "query by document" to analyse overlapping topics and compare it to regular expression search. Accordingly, this study attempted to show that LDA can provide a possible technique to help filter noise, isolate relevant documents, and produce results with a higher relevance. However, all three tests were applied to RDC and users' data in RDC is hugely unstructured and lacks truth data. Moreover, only a selection of keywords that were likely to be contained within target documents was tested. The evaluation of tests was performed using the data of five persons only, which was rather limited. Using a greater number of people would have helped decrease bias in the results.

Further studies were conducted within big data analytics by Chandarana and Vijayalakshmi (2014) and Elgendy and Elragal (2014) investigating and analysing the methods and tools that could be applied to big data to enhance decision making. They claimed big data analytic technologies are highly significant in relation to decision making in many fields, such as quality management, risk management, and fraud detection. A similar study conducted by Chandarana and Vijayalakshmi (2014) provided deep insight into current big data analytics frameworks. They made a comparison between three frameworks (Apache Hadoop, Project Storm, and Apache Drill) and found that the Apache Drill framework was the best for interactive and ad-hoc analysis while Project Storm was appropriate for analysing data streaming and Apache Hadoop was the most appropriate for the workload. Although these three frameworks are suitable for distributed processing of big data, they were not designed for security and forensic purposes. Tannahill and Jamshidi (2014) attempted to provide a bridge between System of Systems (SoS) and big data analytics. SoS is an integrated OS that operates independently to achieve high goals in non-homogeneous systems. Moreover, SoS contributes to generating unmanageable big data in many domains (e.g., cloud computing, healthcare, transportation, and cyber-physical systems). The authors highlighted some available tools in MATLAB that could be used to extract information from unmanageable big data, as well as enable users to draw helpful conclusions. They used tools such as fuzzy interference, neural networks, principal component analysis (PCA), and genetic algorithms to generate a prediction model for solar irradiance.

In another effort to link deep learning applications and big data analytics, Najafabadi et al. (2015) reported that deep learning algorithms were used to extract high-level abstractions in data. They explained that because of the nature of big data, deep learning algorithms could be used for analysis and learning from a massive amount of unsupervised data, which helped to solve specific problems in big data analytics. However, deep learning still has problems in learning from streaming data, and has difficulty in dealing with high dimensional data and distributed and parallel computing.

3.2.1 Artificial Intelligence

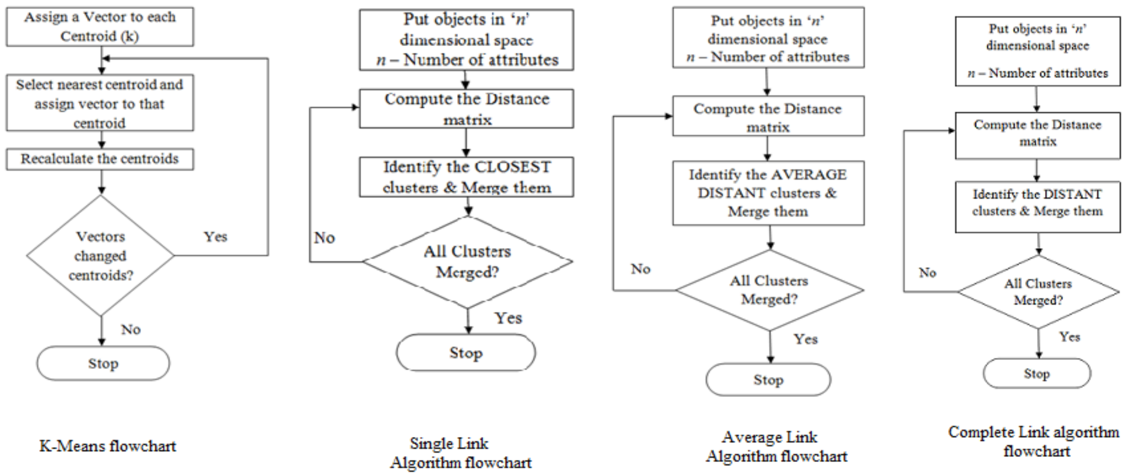
Artificial intelligence (AI) has always played an important role in many fields involving more data than humans can handle. Many studies and experiments have been carried out with the aim of coping with the rapid increase of data. From this respective, Dilek et al. (2015) presented advanced work on the application of artificial intelligence techniques in cybercrime detection systems. Their review displayed research that used AI applications, such as an artificial neural network, an intelligent agent, an artificial immune system, a genetic algorithm, and fuzzy sets. They explained that AI applications assist in the confrontation with cybercrime by providing the flexibility and learning capabilities of intrusion detection and prevention systems (IDPS) software. However, although AI applications offer the opportunity to discover unknown attacks, IDPS suffers from certain limitations, such as being sensitive and prone to giving a high number of false-positive alarms.

Hoelz et al. (2008) proposed the multi-agent digital investigation toolkit (MADIK), which is a collaborative approach towards aiding experts during a computer digital investigation. This approach consists of four agents, each with a dedicated process. HashSetAgent is used to calculate the MD5 hash for the files and to compare them with the knowledge base in order to discount irrelevant files. FilePathAgent is used to keep file paths that are commonly used in the knowledge base. FileSignatureAgent is used to check the integrity of files by examining file headers. TimelineAgent is used to examine the creation date and the modification of files in order to detect events such as software installation, data backups, and web browser usage. The MADIK was tested on real data, consisting of 450,000 files from seven different hard drives belonging to a real investigation case. The total size of all files was approximately 113 GB. The experimental results showed that each agent made a suggested reduction reach of 42%, 30%, 25%, and 5%, respectively. The time spent on the experiment was about five hours, whereas a human examiner would have to spend 25 hours to achieve the same reduction on the same data. As a result, there were only 51 user files related to evidence, consisting mostly of documents and spreadsheets. Further, using the MADIK is not possible with data sizes that are greater than one TB. The MADIK can be used with a small volume of data.

Similarly, Bandgar et al. (2012) proposed a research approach to study spam emails by using the MADIK to reduce the investigation time and to retrieve useful features from spam emails. They then suggested the use of data mining techniques, such as clustering algorithms, to find relationships between email messages. The authors claimed the MADIK's output results were not handled well; in contrast, clustering algorithms lead to data reduction. Therefore, they used a hybrid model to enable the two to complement each other. Bandgar et al.'s (2012) method is theoretical, without the backing up of implementation or an evaluation. Instead, intelligent data analysis (IDA) was developed to handle a number of issues related to data analysis. In addition, IDA is an important field of data mining that uses AI to extract useful information from huge data sets. In this context, Kong et al. (2014) discussed IDA and its challenges in the big data environment from three aspects: the algorithm principle and the scale and type of datasets handled by IDA. However, IDA is still facing many problems, especially with practical applications.

3.2.2 Data Clustering

Recently, data clustering has been studied and used in many areas, especially in data analysis. This literature will focus on algorithms and techniques that give potential solutions for big data analysis. da Cruz Nassif and Hruschka (2011) and Gholap and Maral (2013) proposed a forensic analysis approach for computer systems through the application of clustering algorithms to discover useful information in documents. Both approaches consist of two steps: a pre-processing step, followed by running clustering algorithms. In the pre-processing step, they performed a dimensionality reduction technique called term variance (TV). TV assigns a relevance score to each feature based on its deviation from its mean value. TV can also increase both the effectiveness and efficiency of clustering algorithms. Experiments involving both approaches were conducted by applying the two steps, using the same five clustering algorithms (i.e., k-means, k-medoids, single link, complete link, and average link) as illustrated in Figure 3-2. They were applied to five different datasets seized from computers in real-world investigations. Both of their results showed that the average link and complete link algorithms give the best results in determining relevant or irrelevant documents while L-means and K-medoids algorithms present good results when there is suitable initialisation. Based on their experiment, they claimed that clustering algorithms provide significant assistance to expert examiners by determining the most relevant documents without reading every document in detail. However, the scalability of clustering algorithms may be an issue because they are based on independent data.


Assign a Vector to each Centroid (k)
Select nearest centroid and assign vector to that centroid
Recalculate the centroids
Vectors changed centroids?
Yes
No
Stop
Put objects in 'n' dimensional space n – Number of attributes
Compute the Distance matrix
Identify the CLOSEST clusters & Merge them
All Clusters Merged?
No
Yes
Stop
Put objects in 'n' dimensional space n – Number of attributes
Compute the Distance matrix
Identify the AVERAGE DISTANT clusters & Merge them
All Clusters Merged?
No
Yes
Stop
Put objects in 'n' dimensional space n – Number of attributes
Compute the Distance matrix
Identify the DISTANT clusters & Merge them
All Clusters Merged?
No
Yes
Stop

K-Means flowchart

Single Link Algorithm flowchart

Average Link Algorithm flowchart

Complete Link algorithm flowchart

Figure 3-2: Clustering Algorithms (Gholap and Maral, 2013)

From a similar perspective, Beebe and Liu (2014) carried out an examination by using four of the competing clustering algorithms for clustering digital forensics text string search output. Their study concentrated on realistic data heterogeneity and its size. They evaluated K-Means, Kohonen Self-Organizing Map (SOM), Latent Dirichlet Allocation (LDA) followed by K-Means and LDA, and LDA followed by SOM. Their experiment showed that LDA followed by K-means obtained the best performance, with an average precision rate 67%, and determining more than 6,000 relevant search hits after only 0.5% of search hit results. In addition, the experiment showed that both algorithm K-Means and SOM, when performed individually, gave a poorer performance than when either is combined with LDA. However, the evaluation was carried out on one synthetic case, which was small in size compared to real-world cases.

Rowe and Garfinkel (2012) developed the Dirim tool, which can be used to determine anomalous or suspicious files automatically in a large corpus. This is achieved by analysing the directory metadata of files, such as the filename, extensions, paths, size, times, fragmentation, status flags, and hash codes. In addition, Dirim depends on two complementary ways of automatically finding a comparison of predefined semantic groups and contrast between file clusters. The first method is executed by features count or numeric attributes of files in drives, such as file sizes that are very large or very small. Afterwards, the files are clustered by using the k-means algorithm based on two factors: temporal association (files were created or modified within a threshold) and spatial association (files in the same directory of a file system). Once the corpus is clustered, a supercluster is used to compare a new drive with clusters that already exist. The supercluster is a new approach that is used for comparing both overall drive statistics and clusters of related files to determine anomalous files. A number of superclusters should be larger than the number of clusters for drives, as superclusters include more diversity of data than the data that exists in a single drive. Using the superclusters method, any cluster outside the superclusters on the new drive is considered anomalous. The experiment was conducted on a corpus consisting of 1,467 drive images and 8,673,012 files were found. They clustered 335 Windows drives with 50 clusters as a target number and obtained 63 superclusters. The Dirim approach led to 6,983 files that were suspicious based on their extensions, as well as 3,962 files that were suspicious according to their paths. However, the main challenge of this approach is its inability to find hidden data in a file because the hidden data does not appear within the metadata of that file. It also analyses the data in each drive individually, which leads to the process repeating multiple times.

Pringle and Burgess (2014) explored the integrity of forensic data in a distributed system. They indicated some technical issues within the distributed system that were forensically unsound. For these reasons, they proposed an FCluster framework that aims to provide assurance for the forensics data in the distributed system. The

FCluster consists of four layers with a number of functions: acquisition, ingestion, distribution, and processing. The functions of FCluster (as illustrated in Figure 3-3) are as follows:

- Acquisition authority: provides the cryptographic keys that used to authorise imaging.
- Imaging: creates the directory metadata submission information package (SIPs) and image files.
- FCluster file-system metadata storage: follows the principles of Hadoop middleware by using Multi-Featured File System in User Space (FUSE) on their distributed mechanism.
- SIP Ingestion: determines new evidence SIPs and leads ingestion.
- Load balancer: selects storage which hosts the elementary copy of the data for processing.
- Replicator: to make multiple copies of the data to ensure redundancy and that the data is still valid.
- Data storage server: stores the data.
- Processing: performs the data processing.

However, the speed of FCluster is slow; this can influence the performance of the system. In addition, each system should process its data individually so there is no connection between the data across the network. Therefore, the management of network would be difficult and overhead.

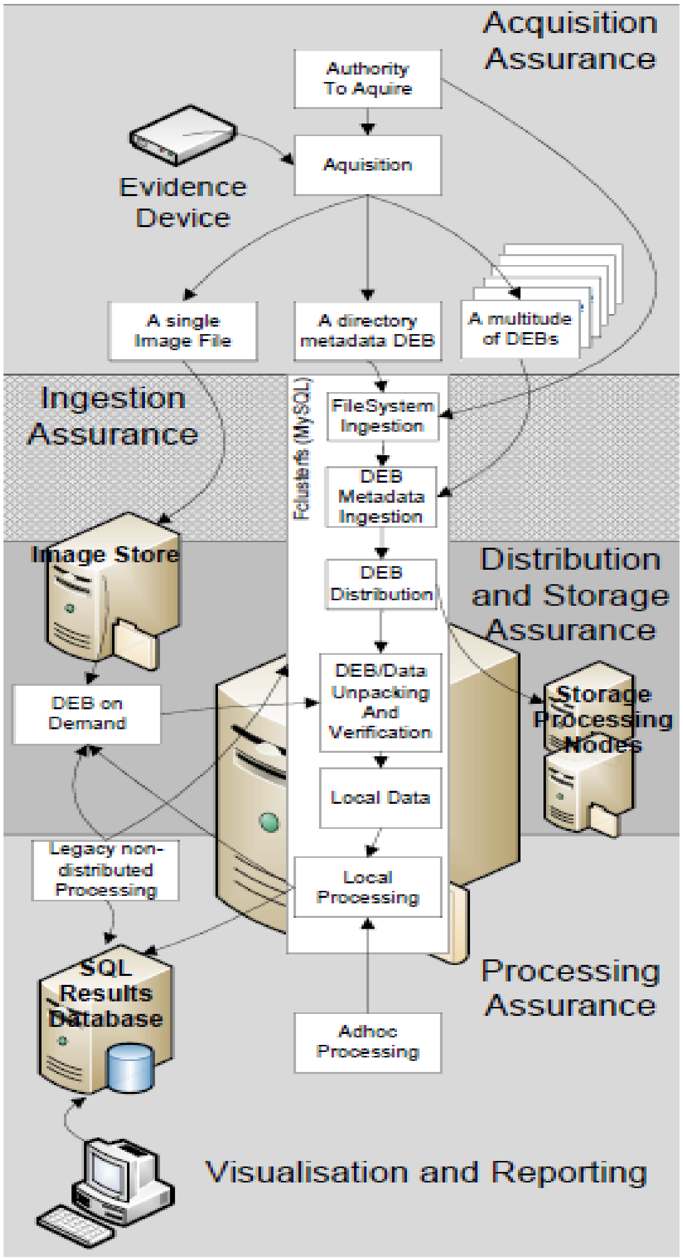


Figure 3-3: FClustering Functions (Pringle and Burgess, 2014)

Yang et al. (2014) proposed a digital forensic approach to form a link between digital media and a criminal profiling system by using a developed fuzzy c-means (FCM) clustering algorithm. The developed FCM algorithm automatically classified data in smartphones to accelerate the discovery of clues for the investigation. They developed FCM because the classical one has various shortcomings, such as high sensitivity to noise and input data. The developed FCM algorithm provides generated fuzzy numbers using results from the FCM clustering algorithm. Their model consists of four steps: collection, examination, analysis, and reporting. After data are collected, they are classified by using a proposed clustering algorithm for examination purposes. Afterwards, the data, which is obtained by clustering, is applied to the system of criminal profiling to extract only related artefacts. The output from examination phase analyses to reconstruct past events and generate a report. However, the approach was conceptual only.

In an attempt to find evidential artefacts in an automated way by using a clustering algorithm, Al Fahdi et al. (2016) proposed an automated approach for identifying the evidence and speeding up the analysis process for computer forensics. Their approach mainly consists of three general steps: metadata extraction, clustering, and automated evidence identification. Real forensic datasets were used to apply the approach and four metadata categories instead of files themselves were chosen and extracted individually (i.e., file system, email, EXIF, and internet history). They then used unsupervised pattern recognition to cluster evidential artefacts to aid the investigators to focus on the evidential files, thereby saving their time and effort. The self-organising map (SOM) was used to automatically group the input data without any supervision. The investigator determined the number of clusters before the process began. Their experiment was conducted using four forensic cases, where each case included a single forensics image. The experiment based on clustering has shown that 93.5% of interesting artefacts were grouped in the top five clusters. However, their approach was only applied to single images with a limited number of metadata categories. In addition, the SOM algorithm cannot handle the missed values of metadata fields such as the timestamps of deleted files. Consequently, the approach ignored a large number of files that might contain evidence.

3.2.3 Data Reduction with Hash-Sets

An enormous quantity of data has to be examined and analysed in forensic investigations, and the amount continues to grow, thus constituting one of biggest issues that confronts forensic investigation. For this reason, many researchers have attempted to use hash sets and data reduction techniques to solve the problem. Roussev and Quates (2012) attempted to use similarity digests as a practical solution for content-based forensic triage. They explained that similarity digests have been widely used in identifying embedded evidence and artefacts, as well as cross-target correlation. Moreover, similarity digests are quicker than other hash set methods because it versus hashes of individual files. Their experiment was applied to the M57 case study, comprising 1.5 TB of raw data, including disk images, RAM snapshots, network captures, and USB flash media. Roussen and Quates (2012) were able to examine and correlate all the components of the M57 triage case in approximately 40 minutes whereas traditional manual correlation and examination methods may have required a day or more to achieve the same result. Ruback et al. (2012) developed a method for determining uninteresting data in a digital investigation by using hash sets within a data-mining application that depend on data being collected from a country or geographical region. This method uses three hash databases for the files' filtration, which are taken from conventionally known hash databases. The experimental method developed by Ruback et al. (2012) showed a reduction of known files by 30.69% in comparison to a conventional hash set, although it had approximately 51.83% hash values in comparison to a conventional hash set.

Similarly, Rowe (2014) compared nine automated methods for eliminating uninteresting files during digital forensic investigations. These methods depend on the file name, size, path, time of creation, and directory. In total, their methods have identified 8.4 million hash values of uninteresting files that could be used for different cases. The experiment was conducted using an international corpus containing 83.8 million files, with the capability of eliminating 54.7% of files that matched with two of nine methods. In addition, false negatives and false positives were 0.1% and 19%, respectively. As a result, the investigators could select one or more methods to reduce data, depending on their investigative objectives. In the same context, Dash and Campus (2014) proposed an approach that uses five methods to eliminate unrelated files for faster processing of large forensic data. They tested the approach with different volumes of data collected from various OSs. Their experiment consisted of two steps. The first consisted of extracting metadata and hash values to eliminate uninteresting files by matching them against the NSRL-RDS database; the second step was to execute the five methods. The results of the experiment showed that an additional 2.37% and 3.4% of unrelated files were eliminated from Windows and Linux, respectively. However, their approach can only be applied to file systems and applications are excluded.

3.3 Database Forensics

Databases contain critical and sensitive information and they can be exposed to many incidents if not protected properly. However, database forensics has received little attention from researchers. Olivier (2009) highlighted some aspects that could involve forensics databases and focused on the difference between databases and file systems in forensic usage. The research compared databases as multidimensional paradigms and file systems as dimensional constructs. This researcher also attempted to gain insights into metadata within database examinations and explored what queries might look like during that examination. A survey by Khanuja and Adane (2011) focused on database security issues and the challenges they present in database forensics. In their view, database forensics is still in the dark ages. However, they indicated there were opportunities to stimulate this area. Hence, Fasan and Olivier (2012) proposed a reconstruction algorithm that gives investigators the capacity to determine, at an early stage of forensic investigation, whether interesting data exists in a database. This algorithm traverses a query log and values blocks and then applies inverse operators of the relational algebra to database reconstruction. They demonstrated conceptual examples to illustrate the application of forensics database reconstruction.

Khanuja and Adane (2012) also proposed a framework that involved an expert system for analysing and reconstructing the activity of any suspicious behaviour in a database. The framework included two stages: the first consisted of making copies of the database and its multiple log files using MySQL programs. This was followed by an attempt to make decisions using inference rules with the assistance of expert knowledge to get interesting and filtered information for analysis. The second stage consisted of reconstructing the activity and preparing a final report from the collected information. However, no implementation or evaluation was carried out with this framework.

Khanuja and Adane (2014) further proposed an automated system within private banks for monitoring ongoing financial transactions by checking database audit logs to determine any suspicious activities. Suspicious transactions were analysed using the Dempster Shafer theory to produce a final report. The Dempster Shafer theory is a general framework suggested in 1967 by Dempster to combine evidence from independent items. Khanuja and Adane's (2014) system was tested on synthetic datasets and acceptable results were obtained.

An increasing volume of databases and the difficulty of processing and managing them with traditional techniques such as SQL have led to new challenges such as distributed, unstructured, semi-structured, and heterogeneous databases. SQL databases use structured query language to retrieve and manipulate data in structured databases that have predefined schema (Birgen et al., 2014). Therefore, a new concept of databases has been identified to solve the issues stated above, which is the NoSQL database. The NoSQL database is a procedure for storing, retrieving, and managing unstructured data. It has a dynamic schema for storing data, such as column-oriented, document-oriented, graph-based, or organised as a key-value store (Birgen et al., 2014). From this respective, Mangle and Sambhare (2013) and Qi (2014) discussed NoSQL (Not Only SQL) database techniques as an alternative to RDBMS for managing big data from a digital forensics perspective. They claimed that NoSQL gives high availability and scalability for distributed systems. In addition, Mangel and Shabhare (2013) made a comparison between relational databases and NoSQL databases in the trend of big data and found that NoSQL databases were better in all aspects. Qi (2014) evaluated the performance of two types of NoSQL, MongoDB and Riak, in terms of big data processing. Qi's experiment was conducted using the Amazon EC2 Cloud and showed that the performance of Riak was better than MongoDB in coping with large datasets, although MongoDB performed better than Riak with reasonably smaller datasets. NoSQL will be used in the next generation of databases because it is non-relational, distributed, open-source, and has high scalability.

3.4 Heterogeneous Data and Resources

The development of information technology and the increasing use of sources that run in different environments have led to difficulties in processing and exchanging data across different platforms. However, several researchers have suggested potential solutions to the problem of the heterogeneity of data and resources. Zhenyou et al. (2011) studied the nature of heterogeneous databases and integration between nodes in distributed heterogeneous databases. They suggested the use of hibernating technology and query optimisation strategy, which have the capability of linking between multi-heterogeneous database systems. Further, Liu et al. (2010) proposed a system framework based on middleware technology for integrating heterogeneous data resources that come from various bioinformatics databases. They explained that middleware is independent software that works with distributed processing, where it is located between different platforms, such as heterogeneous source systems and applications. Liu et al.'s (2010) system used XML to solve the heterogeneity of data structure issues that describe the data from different heterogeneous resources while ontology was used to solve the semantic heterogeneity problem. The key benefit of this system is that it provides a unified application for users. The above finding is consistent with Ge et al.'s (2012) study. They proposed a semantic framework system to integrate heterogeneous data from different sources based on domain ontology. The domain ontology of this framework has two main processes: semantic integration and query. The semantic integration process is executed to integrate heterogeneous data from multiple sources into domain ontology schema to enhance the capability of data understanding. Next, a semantic query process is completed to retrieve the results most relevant to user requirements by inferencing over the ontology. The experiment was conducted in a real environment with simple queries. The result showed the performance of the proposed approach was better than the keywords-based method for retrieving results. However, it was of limited use with complex queries.

Another experiment was made by Liu et al. (2012), wherein they suggested a theoretical approach to the integration of heterogeneous databases based on hybrid ontology. The authors indicated the drawbacks of global ontology and local ontology and proposed employing these drawbacks in a hybrid integration method. In another study, Chang et al. (2013) developed the Universal Heterogeneous Data Integration Standard (UHDIS) with the assistance of a parsing algorithm to integrate real-time data for monitoring purposes. In their system, they first collected heterogeneous data from different sources using (DAMs) acquisition techniques and then transferred them to UHDIS, which has the capacity to reduce redundant information and transmission time. The UHDIS output was uniform data. Next, the parsing algorithm performed parsing within uniform data and mapped integrated data into user tables in a database. The result of this experiment demonstrated that the system's performance was acceptable and efficient. Mezghani et al. (2015) proposed a generic architecture for heterogeneous big data analysis that comes from different wearable devices based on the Knowledge as Service (KaS) approach. This architecture extended the NIST big data model with a semantic method of generating understanding and valuable information by correlating big heterogeneous medical data. This was achieved by using Wearable Healthcare Ontology (WH_Ontology), which aids in aggregating heterogeneous data, supports the data sharing, and extracts knowledge for better decision-making. Their approach was presented with a patient-centric prototype in a diabetes scenario, and it demonstrated the ability to handle data heterogeneity. However, the research aim tended to focus on heterogeneity rather than security and privacy through data aggregation and transmission. In the context of heterogeneity, Zuech et al. (2015) reviewed the available literature on intrusion detection within big heterogeneous data. Their study sought to address the challenges of heterogeneity within big data and suggested some potential solutions, such as data fusion. Zuech et al. (2015) explained that data fusion is a technique of integrating data from different sources that commonly have different structures into a consistent, accurate, and useful representation. The more significant findings to emerge from this study are that big heterogeneous data still present many challenges in the form of cybersecurity threats. Further, data fusion has not been widely used in cyber security analysis. Therefore, they suggested using the data fusion in a multi-system framework to solve the problem of heterogeneity by dealing with data at once instead of repeating the process. However, the techniques suggested can only be applied to a particular type of dataset and is not comprehensive to be applied to all data types.

3.4.1 Data Correlation

Although there has already been some work in the data correlation of digital forensics to discern the relationship between evidence from multiple sources, there is a need for further research in this direction. Garfinkel (2006) proposed a new approach, using forensic feature extraction (FFE) and cross-drive analysis (CDA) to extract, analyse, and correlate data over many disk images. FFE is a diversity of techniques and is used to identify and extract certain features from digital media, such as credit card numbers and email message IDs. The researcher used pseudo-unique identifiers and feature extractors to obtain these features. When this is achieved, CDA plays a significant role in the analysis and correlation of datasets spanning multiple drives. The analysis and correlation are achieved by applying two forms of CDA: first and second orders. The first order of CDA is carried out using the CDA stop list and hot drive identification to automatically select drives that have a large number of features. The second order of CDA uses email address multi-drive correlation and scores the correlation for connecting the dots between features to produce a final report. In addition, this architecture was used to analyse 750 images of devices containing confidential financial records and interesting emails. In comparison, the practical techniques of multi-drive correlation and multi-drive analysis require improvements to their performance in order to work with large datasets. However, the CDA can be appropriate to images with a small size of data. CDA performance is likely reduced by increasing the number and size of images. In addition, the feature extractors lack the ability to extract most features of files within the disk images. Figure 3-4 illustrates the implementation of the cross-drive analysis process.

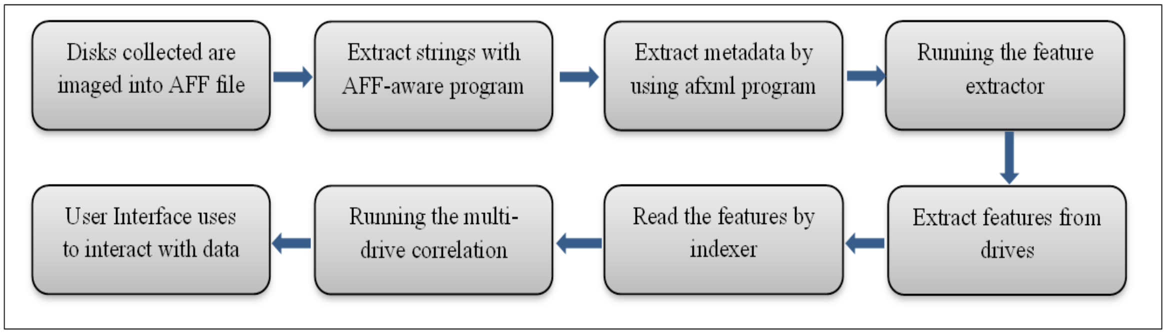

Disks collected are imaged into AFF file
Extract strings with AFF-aware program
Extract metadata by using afxml program
Running the feature extractor
Extract features from drives
Read the features by indexer
Running the multi-drive correlation
User Interface uses to interact with data

Figure 3-4: Implementation of Cross-drive Analysis

Another experiment sought to perform forensic analysis and the correlation of computer systems. Case et al. (2008) presented two contributions to assist the investigator in “connecting the dots”. First, they developed a tool called Ramparser, which is used to perform a deep analysis of Linux memory dump. The investigator used this tool to get detailed output about all processes that occur in the memory. Second, they proposed a framework called FACE (Forensics Automated Correlation Engine), which was used to discover and make correlations between evidence automatically. FACE provides automated parsing over five main objects, namely memory images, network traces, disk images, log files, and user accounting and configuration files. This prototype provides five main data views to display a list of the artefacts in that object: users, groups, processes, files systems, and network capture. The authors tested the FACE framework with a hypothetical scenario and the application was successful. Any integrated tools of computer forensics should be able to function with most OSs. However, this approach can be applied to a limited number of specific resources and has not been tested with multiple resources that contain similar datasets. In addition, it cannot process data that is generated at the application level.

Raghavan et al. (2009) proposed the Forensic Integration Architecture (FIA) to integrate evidence from multiple sources. The FIA consists of four layers: an access layer, an interpretation layer, a meta-information layer, and a visualisation layer. The first layer provides a binary abstraction of all data acquired during the investigation while the second layer has the capability of supporting various OSs, system logs and mobile devices. In addition, it provides an interpreter semantic to extract logical blocks of data from evidence sources, which are passed to the layer above. A meta-information layer provides interface applications to facilitate metadata extraction from files. The fourth layer is responsible for integrating and correlating information from multiple sources. These combined sources can serve as comprehensive evidentiary information to be presented to a detective. There are three sub-layers used to achieve the goal of the last layer: content indexing, cross-referencing and knowledge representation, and a reasoning sub-layer. However, as the FIA architecture was merely conceptualised via a car theft case study, further investigation is required to evaluate its practicality. Additionally, there is no explanation about how the system will work if the resources contain similar evidential categories. A similar but more thorough study was conducted by Raghavan and Raghavan (2013), based on the FIA approach to integrate different sources and unify analysis. They proposed an analysis engine called AssocGEN, which uses metadata to find associations between heterogeneous artefacts. These artefacts belong to files in hard disk images, system and applications logs, and network packet dumps. The AssocGEN approach consists of three basic layers: a digital evidence layer, a digital artefacts traversal and metadata parser layer, and an evidence composition layer. Their experiment was conducted to check the time performance of AssocGEN by a comparison with FTK 3.2 across different datasets. The experiment showed that FTK was slower than their approach. In addition, FTK only searched file system metadata while AssocGEN extracted both application and file system metadata. Further, to be comprehensive, this approach requires automated methods without human involvement to identify related artefacts.

Regarding big data correlation, Nakanishi (2015) proposed an anteroposterior method for correlation, based on the time between the heterogeneous thing, events, or phenomena, using conditional probability. The author emphasised some critical issues in big data analysis, such as heterogeneity, continuity, and visualisation. The experiment was applied to verify the effectiveness of his method with a hypothetical case. However, there was an inconsistency in this experiment that needs further explanation.

Regarding identifying the evidence in an automated way, Al Fahdi (2016) proposed an automated algorithm was called the Automated Evidence Profiler (AEP) to analyse and identify the related artefacts across all clusters of metadata SOMs. The AEP contains two steps. The first concerns identifying the first cluster based on prior work achieved in profiling criminal behaviour. The second is to identify subsequent clusters using the timeline analysis of each file being presented in the first cluster. The experiment was conducted using four forensic cases, where each case included a single forensics image. The AEP algorithm presented acceptable results, showing that it can reduce the investigator’s time taken to analyse the cases and present the relevant evidence in a report. However, the AEP algorithm does not work with all cases because it depends on some prior work completed in profiling criminal behaviour to identify the first cluster. There might be new criminal behaviour cases that are not yet analysed.

3.5 Discussion

It appears from the aforementioned research that numerous problems need to be overcome to achieve an effective approach to digital forensic investigations in heterogeneous data environments. Table 3-1 summarises the existing work in data analysis of data consisting of different types and sizes.

Table 3-1: A Comprehensive Study for Analysing Data and Heterogeneous Information in a Forensic Manner

	Authors	*Year*	*Context*	*Challenge*	*Potential Solution*	*Type*
1	Hoelz et al.	2008	AI	Automated forensic analysis	MADIK approach	Real
2	Bandgar et al.	2012		Automated forensic analysis of emails	Combine MADIK and data mining	Conceptual
3	Kong et al.	2014		Intelligent big data analysis	-	Survey
4	Dilek et al.	2015		Cybercrimes detection	Neural network, genetic algorithm, fuzzy set	Conceptual
5	XU et al.	2013	data acquisition & analysis	Big data acquisition	Acquisition engine	Conceptual
6	Chandarana & Vijayalakshmi	2014		Enhance decision-making	-	-
7	Elgendy & Elragal	2014		Big data analysis	-	-
8	Noel & Peterson	2014		Find interesting information	Latent Dirichlet allocation (LDA)	Real
9	Tannahill & Jamshidi	2014		Big data analysis	System of system	Conceptual
10	Najafabadi et al.	2015		Big data analysis	Deep learning	Conceptual
11	da Cruz Nassif et al.	2011	Data Clustering	Find related information	Clustering algorithms	Real
12	Rowe & Garfinkel	2012		Determining anomalous files	Dirim tool	Real
13	Gholap & Maral	2013		Forensics analysis	Clustering algorithm	Real
14	Beebe and Liu	2014		Text string search	Approach based on	Real

					clustering algorithms	
15	Pringle & Burgess	2014		Forensics data assurance	FClustering framework	Conceptual
16	Yang et al.	2014		Link between digital media & criminal profiling system	Developed c-means algorithm	Conceptual
17	Al Fahdi, M	2016		Find evidential artefacts using clustering	SOM algorithm	Real
18	Garfinkel, S. L	2006	Data Correlation	Extraction, analysis, correlation data	Forensics feature extraction and cross-drive analysis	Real
19	Case et al.	2008		File correlation in computer systems	FACE Framework	Prototype
20	Raghavan et al.	2009		Evidence integration	FIA architecture	Prototype
21	Raghavan, S	2013		Heterogeneous artefacts	AssocGEN engine	Prototype
22	Al Fahdi, M	2016		Identifying the related evidential artefacts	The Automated Evidence Profiler (AEP)	Real
23	Roussev et al.	2012	Data Reduction	Content-based forensics triage	Similarity Digest	Prototype
24	Ruback et al.	2012		Determining uninteresting data	Hashset & data mining applications	Real
25	Dash & Campus	2014		Eliminating unrelated files	Hashset	Real
26	Rowe, N	2014		Eliminating unrelated files	Hashset	Real
27	Olivier & Martin	2009	Database Forensics	Multidimensional construct of databases	-	Survey
28	Khanuja & Adane	2011		Database security	-	Survey
29	Fasan et al.	2012		Determining interested data	Reconstruction algorithm	Prototype
30	Khanuja & Adane	2012		Analysis of suspicious behaviour	Expert system	Prototype
31	Mangle & Sambhare	2013		Managing of big databases	NoSQL technology	Conceptual
32	Khanuja & Adane	2014		Monitoring financial transactions	Automated system	Prototype

33	Qi, M	2014		Managing of big databases	NoSQL technology	Conceptual
34	Liu et al.	2010	Heterogeneous Data	Integration of heterogeneous information	Middleware technology	Prototype
35	Zhenyou et al.	2011		Integration of heterogeneous databases	Hibernate technology	Conceptual
36	Ge et al.	2012		Integration of heterogeneous information	Semantic framework	Real
37	Liu et al.	2012		Integration of heterogeneous databases	Hybrid ontology	Conceptual
38	Chang et al.	2013		Integration of real-time data	UHDIS & parsing algorithm	Prototype
39	Mezghani et al.	2015		Heterogeneous big data analysis	WH_Ontology	Prototype
40	Zuech et al.	2015		Big heterogeneous data	-	Survey

From the perspective of the data volume, the current tools of digital forensics, such as FTK and Encase, have failed to keep pace with the increase. For that reason, Noel and Peterson proposed using the LDA method based on real data corpus (RDC) to find relevant information from a large volume of data. Although they obtained reasonable results, there is lacking in their work because they use specific keywords to obtain target results and RDC is hugely unstructured. In addition, this chapter has examined several technologies, such as AI, data clustering, and data reduction, all of which have the potential capacity to save digital investigators time and effort. Although AI provides flexibility and learning capabilities to forensics software, there have been only a limited number of studies in this area. One of these studies is the multi-agent digital investigation toolkit proposed by Hoelz et al. (2008) for using AI for forensic purposes; however, this toolkit does not meet the forensics requirements of big data. In contrast, data clustering techniques have been widely used to speed up the investigation process by determining relevant information more quickly. So far, however, these techniques have been applied to large volumes of data but not to big data. It appears from the aforementioned investigations that most attention has been paid to data reduction that helps to eliminate uninteresting files. In addition, Roussev and Quates' research has been given great prominence because they handled cases that comprised 1.5 TB of data in minimal time.

Likewise, there are only a few available studies on the use of big heterogeneous data in digital forensics. Further, it may be noted that most of the studies were aimed at integrating heterogeneous databases of insufficient sizes. However, integration technology based on ontology techniques offer promising prospects of a solution. In this context, Ge et al. (2012) proposed a semantic framework for integrating data from different and heterogeneous sources. However, it has not been used in forensic investigations with the heterogeneity of big data. From a data correlation perspective, there has only been limited research on data correlation that offers a potential solution to heterogeneous data issues. Although Raghavan suggested forensic integration architecture (FIA) to integrate and correlate evidence from multiple sources; there is no implementation-based evaluation. Therefore, these issues have yet to be resolved—particularly issues related to big data.

The above research has produced various techniques for big data analysis, as well as potential solutions to the accompanying problems, including data clustering, data reduction, and artificial intelligence. The major challenges with big data analysis are volume, complex interdependence across content, and heterogeneity. However, existing frameworks attempt to cope with a specific issue. As a result, big data analytics regarding forensics needs a comprehensive framework that can handle issues such as volume, variety, and heterogeneity of data. Several solutions have been suggested for dealing with these problems.

3.6 Conclusion

There have been several studies that present comprehensive surveys of existing work in forensics analysis, with different types and sizes of data from various fields. These studies have shown significant increases in data volume and the amount of digital evidence being analysed in digital investigations. Moreover, digital forensic investigation is facing new challenges that may require the abandonment or modification of well-established tenets and processes. These challenges include the diversity, heterogeneity, and large volume of data. Accordingly, several solutions have already been suggested to cope with these issues individually and few researchers have proposed technical solutions to mitigate these challenges. Although AI, data clustering, and data reduction techniques are reasonable solutions to cope with these challenges, there are restricted studies in this regard. However, there is a growing need to optimise these solutions in a comprehensive framework to enable all the issues to be dealt with together. To the best of the author's knowledge, most of the current forensic tools and techniques are invalid or unsuitable for the large volume and heterogeneity of data forensics work. This deficit has motivated the present study.

The chapter has illustrated the analysis techniques of big data and the challenges they pose in forensics. It has also proposed solutions and determined the need for a viable framework.

4 The Harmonisation of Heterogeneous Data

This chapter proposes novel approach to the merging of metadata datasets through a ‘characterisation and harmonisation’ process. The characterisation process analyses the nature of the metadata and the harmonisation process merges the data. A series of experiments using real-life forensic datasets were conducted to evaluate the algorithm across different categories of datasets (i.e. messaging, graphical files, file system, Internet history, and emails), each containing data from different applications across different devices.

4.1 Introduction

The rapid development of technology over the last decade has brought various challenges to digital forensics. This development, including the variety of devices, OSs, files, and applications, clearly increases the complexity, diversity, and correlation issues within forensic analysis (Garfinkel, 2006). A wide range of tools and techniques, such as Encase and AccessData’s Forensic Toolkit, have been developed to investigate and analyse the cybercrimes and threats. Unfortunately, the increasing number of digital crime cases and extremely large datasets (e.g., which are found in big data projects) are difficult to be processed using existing software solutions, including conventional databases, statistical software, and visualisation tools (Shang et al., 2013). The goal of using traditional forensic tools is to collect, preserve and analyse information on a single computing device to find potential evidence. However, the situation becomes complicated within the big data environment (e.g., big databases). Further, data is likely to be split across multiple places (Patrascu and Patriciu, 2013).

In some cases, digital evidence across big heterogeneous sources consists of multiple connected artefacts. In such cases, the artefacts in each source are manually analysed to generate a report that is corroborated in the final step. This leads to an ever-increasing burden on investigators to determine the association between the artefacts. Therefore, it is necessary to develop a cohesive tool that can be used to analyse multiple artefacts across diverse sources to arrive at a consolidated outcome.

Recently, several researchers have tried to use metadata within the digital forensic domain to reconstruct past events. Digital forensic cases can include several categories of similar metadata within a single forensic image or across multiple resources resulting in repeating the forensic process many times and increasing the workload of the investigator. Subsequently, automated correlation between the evidential artefacts from various sources is currently impossible. Therefore, in this chapter, an automated approach for analysing and merging datasets by applying a novel algorithm of characterisation and harmonisation is proposed. This approach seeks to provide a fusion of similar metadata categories across multiple and heterogeneous resources within a single case. Consequently, it leads to overcoming the heterogeneity issues and making the examination and analysis easier.

4.2 Metadata

The metadata concept was first introduced in the 1960s in the library management field (Manso-Callejo et al., 2010). Metadata is data about data or information about information, which provides a short description of the required information for a digital resource identification (Guptill, 1999). It is structured information that makes retrieving and using digital resources easier. Further, Khan (2008) stated that metadata describes the attributes of files and folders, including file size, timestamps, access control, authorship, linkages, and organisation of folders and files in storage media. Therefore, metadata can become a vehicle for integrating the examination and analysis of different sources. Anastasiou and Vázquez (2010) highlighted that metadata became popular and important when the internet was launched in the 1990s, in conjunction with the need for scaling and filtering information. In addition, Tim Berners Lee (1997), the inventor of the web, introduced metadata as "machine understandable information for the Web". Subsequently, in 2004, the Resource Description Framework (RDF), a language that links metadata about resources on the web, including authors, and modification dates of web pages, copyright, and licensing information about web documents (Anastasiou and Vázquez, 2010). In addition, the metadata term can be used in many contexts, such programming languages, where metadata is used to give information about program structure itself, such as classes, methods, and attributes (Guerra and Oliveira, 2013). In addition, it is essential in database technologies and information retrieval systems to understanding and interpreting the contents of these systems. Additionally, there are three main types of metadata in most resources (Press, 2004):

- Descriptive metadata describes a resource for purposes, such as discovery and identification. It can include elements such as title, abstract, author, and keywords.
- Structural metadata indicates how compound objects are put together, such as how pages are ordered to form chapters.
- Administrative metadata provides information to help manage a resource, such as when and how it was created, file type and other technical information, and who can access it.

Metadata plays an essential role as a key step in a development strategy for various systems. Moreover, metadata composition and its properties rely on the underlying features of systems, the features of resources they describe, the users' needs, and other factors. Recently, a number of researchers have tried using metadata within the digital forensic domain to reconstruct past events (Raghavan and Raghavan, 2014).

4.3 Properties, Functions, and Facilities of Metadata

A number of metadata properties, functions, and facilities exist in many digital resources that could be universal or specialised.

4.3.1 Metadata Properties

This section describes some priorities of metadata in various spheres:

- Explicit representation of resource properties: metadata provides a conspicuous representation for most resources properties, such as text documents, images, graphical diagrams, etc. (Kogalovsky, 2013).
- Static and dynamic metadata: static metadata does not change over time, for example, the database schema in databases is not expected to change. In comparison, dynamic metadata is expected to change because the content of data is relatively changed. For example, digital library content is always updated when a new object is added (Marinemetadata.org, 2015).
- Autonomous metadata: metadata could be isolated from the digital resource in which it can be built; for example, a document type definition (DTD) is a set of markup declarations that describe the type of XML document and are stored separately from XML documents on the web (Kogalovsky, 2013).

- Syntactic and semantic metadata: syntactic metadata gives a description of what the information looks like and how it is organised. Whereas semantic metadata describes what the information means (Marinemetadata.org, 2015).
- Content-dependent and content-independent metadata: content-dependent metadata provides a description about the data content of resources; in contrast, content-independent metadata provides information about the creation date and location of the resource; in the other words, it does not provide information about the content (Kogalovsky, 2013).
- Metadata accuracy: the information about resources provided by metadata should be as accurate as possible (Ochoa and Duval, 2009).
- Logical consistency and coherence: Metadata should be consistent with the file or an application it describes; further, metadata information about the same resource should be coherent (Ochoa and Duval, 2009).
- Timeliness: metadata should be changed whenever the resource data it describes changes (Ochoa and Duval, 2009).

4.3.2 Metadata Functions

Metadata has been used in many systems with four major functions. The four functions of metadata are briefly explained below (Guerra and Oliveira, 2013; Marinemetadata.org, 2015):

- Metadata as a means of representation: most systems use metadata to access information entities of resources; therefore, it should be chosen carefully in order to ensure the system does not lose control of any action.

- Metadata as an aid to structuring a system: metadata use is significant for structuring the information entity in systems. Therefore, a framework for choosing metadata should be existing for organising a system.
- Metadata as a basis of the visual display of information: the third function of metadata is to provide assistance by displaying summary information about data entities for the system's users.
- Discovery and retrieval of information resources: this is a significant function of metadata, as it can be used in search criteria. In addition, the semantic search by metadata is an effective way to reduce noise while searching for information.

4.3.3 Facilities for Metadata Representation

A number of facilities can be used to represent metadata elements, such as natural languages, artificial languages, and markup languages. The following section contains descriptions of these facilities (Kogalovsky, 2013; Lee, 2003).

- Natural languages: metadata could be represented by natural languages, such as annotations of publications, research, and different information about resources and their authors.
- Artificial languages: many computer languages can be used to represent metadata. For instance, ontology languages, workflow languages, conceptual modelling languages, and DBMSs for data description.
- Markup languages: XML, HTML, and XHTML are the most popular examples of markup languages. They are designed to describe the metadata of documents as their specification allows for prescribing structured data.

4.4 Metadata and Digital Forensics

A number of metadata types exist and provide some attributes, as shown in the previous section, which is important in any process. These attributes belong to file system metadata, application metadata, email metadata, document metadata, file header, and many more. However, the three most important types are file system metadata, application metadata, and email metadata, as they are included in most devices and digital forensic cases.

4.4.1 File System Metadata

File system metadata provides summary information about a file system and aids in controlling and retrieving that file. The summary information describes the layout and attributes of regular files and directories (Buchholz and Spafford, 2004). These attributes store the file owner, file size, file extension, file permissions, creation timestamp, last access timestamp, last modified timestamp, last metadata change timestamp, etc. (Raghavan, 2014). Table 4-1 shows a comparison of some file system metadata within various file systems (Raghavan, 2014).

Table 4-1: Comparison of file system metadata over different file systems

File system	Stores file owner	POSIX file permissions	Creation timestamp	Last access timestamp	Last modified timestamp	Last metadata change timestamp	Extended attributes
FAT12	No	No	Yes	Yes	No	No	No
FAT16	No	No	Yes	Yes	Yes	No	No
FAT32	No	No	Yes	Yes	Yes	No	No
exFAT	No	No	Yes	Yes	Yes	Unknown	Unknown
HPFS	Yes	No	Yes	Yes	Yes	No	Yes
NTFS	Yes	Yes	Yes	Yes	Yes	Yes	Yes
HFS	No	No	Yes	No	Yes	No	Yes
HFS+	Yes	Yes	Yes	Yes	Yes	Yes	Yes
EXT2	Yes	Yes	No	Yes	Yes	Yes	Yes
EXT3	Yes	Yes	No	Yes	Yes	Yes	Yes
EXT4	Yes	Yes	Yes	Yes	Yes	Yes	Yes

As illustrated in Table 4-1 above, timestamps are available within most file system technology across various computing environments. They provide meta information that could be used to analyse and reconstruct the events that happened on a machine (Chow et al., 2007).

4.4.2 Event Log Metadata

Event log metadata provides significant information to reconstruct the events in most modern IT systems (Vaarandi, 2005). IT systems have the capability to log their events and audit them in a local or remote log server, including many applications, OSs, network devices, and other system components. In the 1980s, the syslog protocol was demonstrated for BSD UNIX, which is supported by many OSs (Lonvick, 2001). Syslog can be implemented with various devices, including routers, switches, laser printers, etc. Vaarandi (2005) stated that the log client creates a message to log an event and send it to a local or remote syslog server. The contents of the syslog message consist of a message string, program name, level, and facility. The program name is used to identify the name of the application or process that sent the message. The level describes the seriousness of the event–for example, warning or emerging–while the facility describes the event category (e.g., mail or print). Figure 4-1 shows a sample of syslog infrastructure.

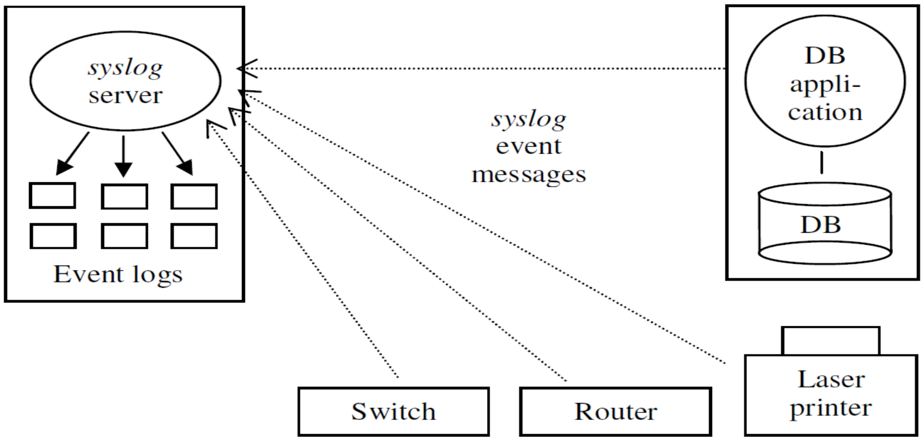


Figure 4-1: Syslog Infrastructure (Vaarandi, 2005)

4.4.3 Email Metadata

Document type definition is introduced as email metadata in Extensible Markup Language (XML), which holds content-feature keywords about an email (Sharma et al., 2008). Researchers have employed email metadata within alternative insight to facilitate dealing with email lists, such as filtration, organisation, sorting (Fisher et al., 2007). Metadata aids in filtering emails based on reading status, organising email by sender, and sorting senders based on their history of interaction with the user.

4.4.4 Metadata Categories within Digital Forensics

Some research considers metadata as an evidentiary basis for the forensic investigation process because it describes either physical or electronic resources (Khan, 2008; Raghavan and Raghavan, 2014). Metadata aids in identifying the associated artefacts that can be used to investigate and verify fraud, abuse, and many other types of cybercrime. Rowe and Garfinkel (2012) proposed a tool for automatically finding suspicious or anomalous files in a large corpus based on their directory metadata. Similarly, Raghavan and Raghavan (2014) proposed a method of identifying the association of evidence artefacts in a digital investigation by using metadata; their method was applied to find the association of metadata from collections of image files and word processing documents. Al Fahdi (2016) attempted to reduce the examination time of investigation by using a self-organising map (SOM) to identify notable artefacts automatically based on files' metadata. Table 4-2 illustrates some of the metadata parameters that can be used by digital forensic tools for investigative proposes.

Table 4-2: Some of input metadata parameters for forensics tools

	Metadata Categories	Features	Data Type
10	File system metadata	File name	String
		File extension (.exe, .txt, .jar etc.)	String
		Creation timestamp	Date & Time
		Last access timestamp	Date & Time
		Last modification timestamp	Date & Time
		Duplication	Boolean
		File size	Numerical
		File path	String
		File Status (Active, Hidden, read-only, encrypted, Deleted etc.)	Boolean

2	Skype metadata	Contact name	String
		Call history, date, and time	Date & Time
		Phone number	Numerical
		Address for each contact	String
		Time zone	Time
		Birthday	Date & Time
		Messages content	String
3	EXIF metadata	Last write date	Date & Time
		Last access date	Date & Time
		Date taken	Date & Time
		Camera make	String
		GPS (longitude, altitude)	Numerical
4	Email	Subject	String
		File name	String
		To, from, cc, bcc,	String
		Submit date and time	Date & Time
		Delivery date and time	Date & Time
		Unread	Boolean
		Unsent	Boolean
		Has attachment	Boolean
		Physical size	Numerical
		Logical size	Numerical
5	Recycle Bin	File name	String
		File type	String
		File path	String
		File size	Numerical
		Time of deletion	Date & Time
		Timestamp (creation, access, modification)	Date & Time
6	Call history	Contact name	String
		Phone type (missed, outgaining, incoming),	Enumeration
		Phone number	Numerical
		Timestamp	Date & Time
7	Messages history	Message to (Contact Name & Phone Number)	String & Numerical
		Message from (contact name and phone number)	String
		Timestamp	Date & Time
		Content	String

8	Browser logs	Timestamp	Date & Time
		The domain (e.g., visitors from .edu, .com, and .gov).	String
		The number of requests for each page on the site.	Numerical
		The host server IP address	String
		Event description	String
		Username associated with the event	String
9	Log file entries (Security, application, setup and system Logs)	Event ID	Numerical
		Log name	String
		User name	String
		Date generated	Date
		Time generated	Time
		Machine (computer name)	String
		Task category	String
		Source (e.g., Outlook, Security-SPP)	String
10	Network packet	Application layer protocol (e.g. HTTP)	String
		Transport layer protocol (TCP, UDP)	String
		Source and destination IP addresses	String
		MAC addresses	Numerical
		Port number	Numerical
		Timestamp	Date & Time
		Packet size	Numerical
		Data	String

4.5 A Novel Forensic System for Merging Multi-Images

The proposed system attempts to bridge the gap between several evidential resources that are included in a single case. It aims to decrease the burden on the investigator by merging similar datasets from multi-resources and producing a single forensic image, thereby dealing with all data at once. Therefore, it seeks to provide an automated framework to merge similar datasets by characterising similar metadata categories and then harmonising them in a single dataset. This approach aims to overcome heterogeneity issues and makes the examination and analysis easier by analysing and investigating the evidential artefacts across devices and applications based on the category to query data at once.

To achieve this, preliminary steps should be undertaken to prepare the datasets before merging them. These steps include resource acquisition, data carving, hashing (pre-processing), and metadata extraction. Therefore, all available suspect resources within a single case should be acquired in a forensically sound manner to produce authentic forensic images that are reliably obtained and admissible. The pre-processing step can recover and extract files from the unallocated file system space (i.e., data carving). It then calculates the hash values of all files for identification, verification, and authentication purposes. Having established that metadata can help with recognising patterns, establishing timelines, and can point to gaps in datasets, it can aid in correlating the evidential artefact in a digital investigation. Therefore, the automated process of metadata extraction undertakes obtaining suitable information (metadata) for the digital forensic process. This information can be extracted or created from any file or application, such as file systems, network packets, databases, and many more. However, a number of metadata categories might contain fields that are not metadata, such as the actual content of a message. Thus, the meta and non-metadata identification process can be used to eliminate these fields. However, simultaneously, it considers an optional step, as it can only be applied to specific categories. Afterwards, the characterisation process identifies and analyses the nature and the types of datasets in order to merge them using the harmonisation process, as illustrated in the next section. The entire system is illustrated in Figure 4-2.

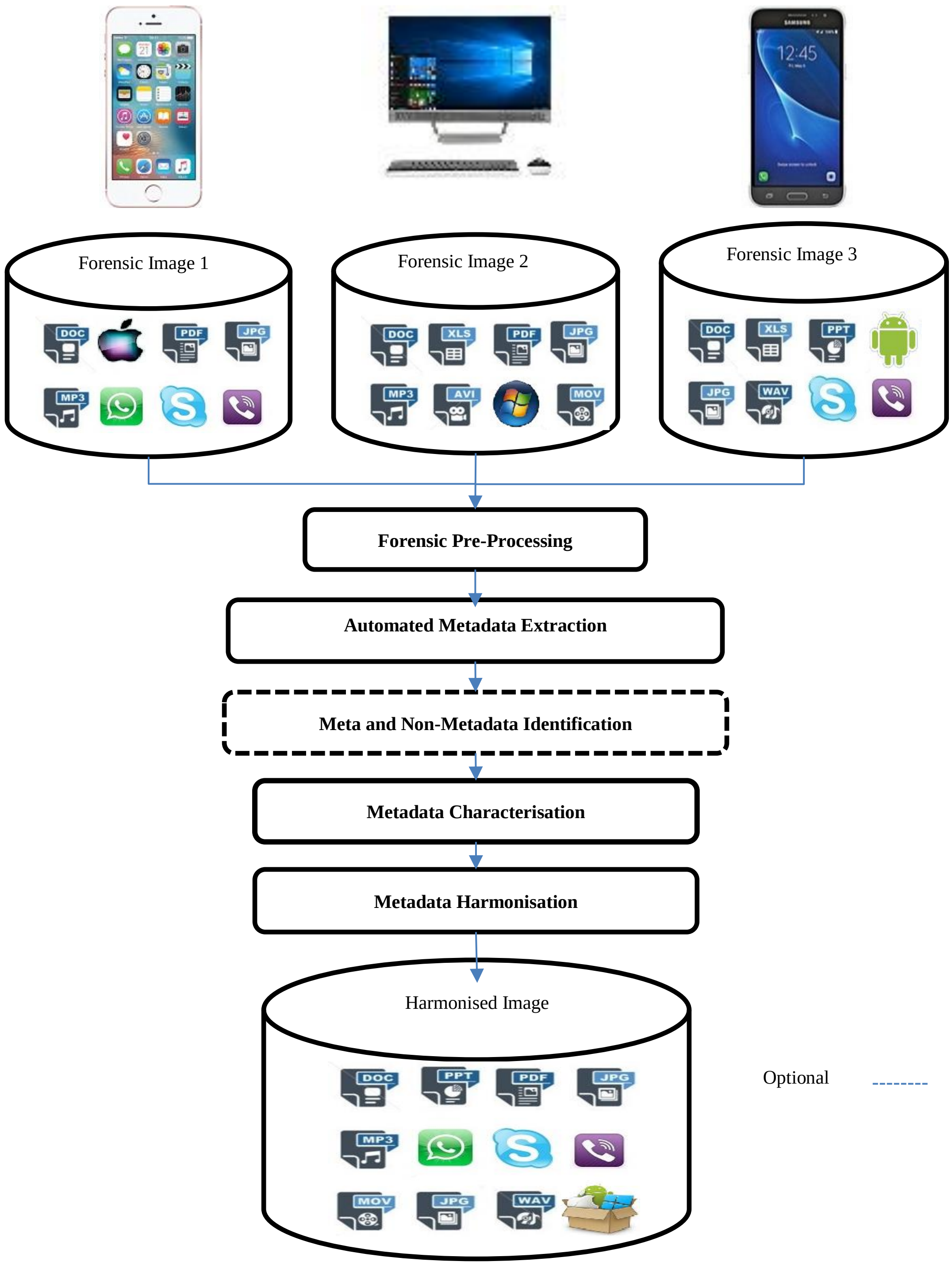


Figure 4-2: Overview of the Proposed Process

4.5.1 An Automated Approach for Metadata Characterisation and Harmonisation

This approach, as illustrated in Figure 4-3, completely depends on the metadata categories where metadata has a particular structure with most datasets related to a single category. Digital forensic cases might include several categories of similar metadata within a single image or across multiple resources. This can lead to repeating the forensic process many times and increasing the encumbrance placed on investigators. Consequently, the automated approach for metadata characterisation and harmonisation splits the problem of merging the datasets into the following aspects:

- **Meta and Non-Metadata Identification**,
- **Characterisation.**
- **Harmonization**.

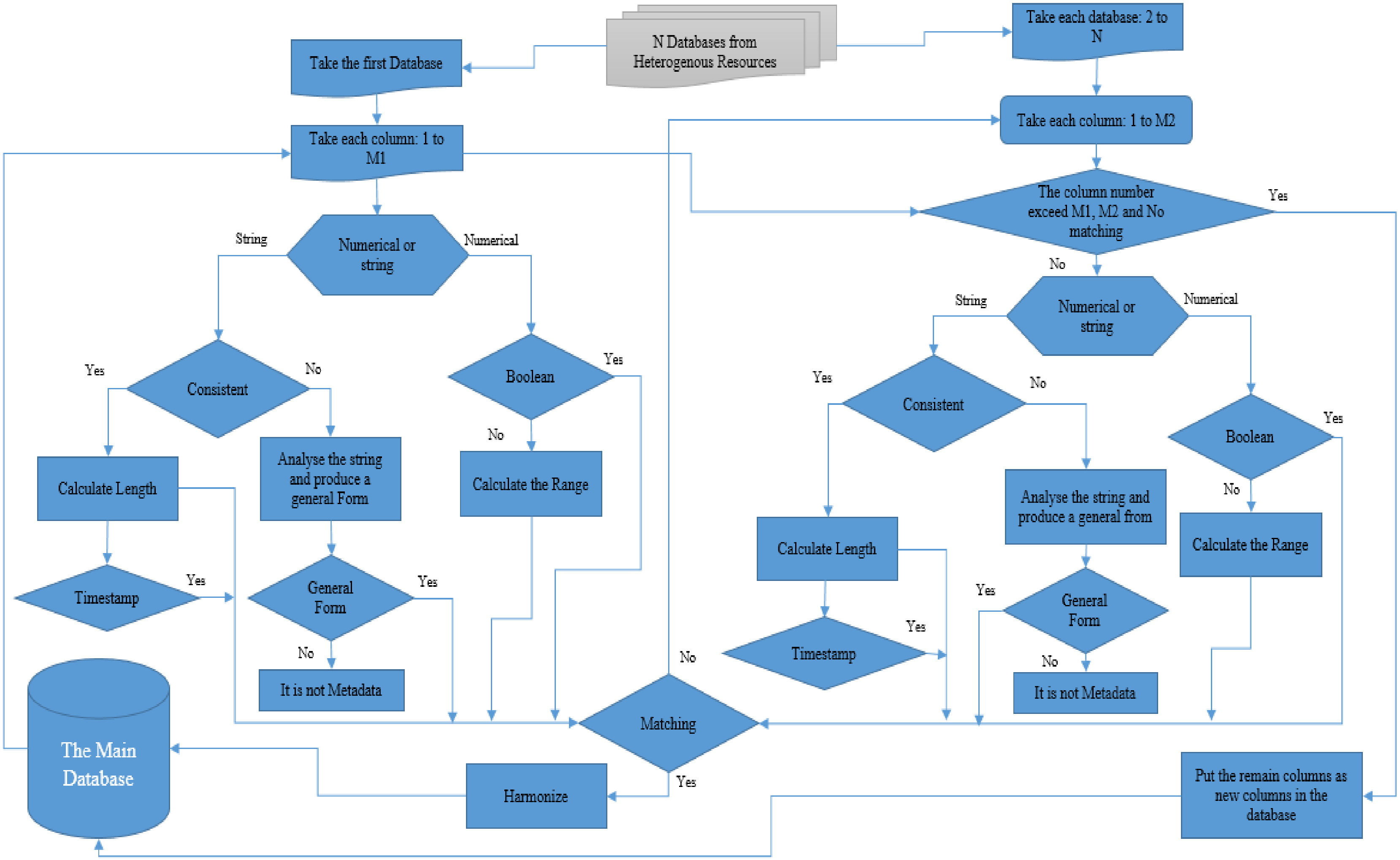


Figure 4-3: Algorithm of Metadata Characterisation and Harmonisation

4.5.1.1 Meta and Non-Metadata Identification

This approach uses the metadata categories as a base to merge datasets. In addition, datasets that contain non-metadata fields should be eliminated. For example, Skype and SMS applications contain fields describing the actual content of messages. Therefore, the variability of the string can be used to identify meta from non-metadata fields because most metadata of the same field has a specific structure and format. In comparison, most non-metadata fields are in the string format. For instance, the dimension of an image is presented as (width x height) (e.g., 300x200, 2000x1500) and this pattern of string can be represented as (NxN), which means (number, letter x, number). Additionally, the file name in most OSs can be represented as (Name.extension), which means (String, Full Stop, Short String). Consequently, the string variability, as in Algorithm 4-1, has the ability to analyse the string to produce a pattern that aids to find the similar metadata fields across multiple categories.

Algorithm 4-1: String Variability Algorithm

```
Input: String Fields.
Output: Decision, Meta or Non-Metadata.
--------------------------------------------------------------
Process
Step 1: Read the first value of the field that has M of values. Go to step 2.
Step 2: Extract the string pattern. Go to step 3.
Step 3: Save it in the record file. Go to step 4.
Step 4: Read the next value. If the counter exceeds M, go to step 5.
                                     Else go to step 2.
Step 5: Read the record file and check if the most patterns are similar.
                                Go to step 6. Else, go to step 7.
Step 6: This field is metadata. Go to step 8.
Step 7: This field is not metadata. Go to step 8.
Step 8: End
```

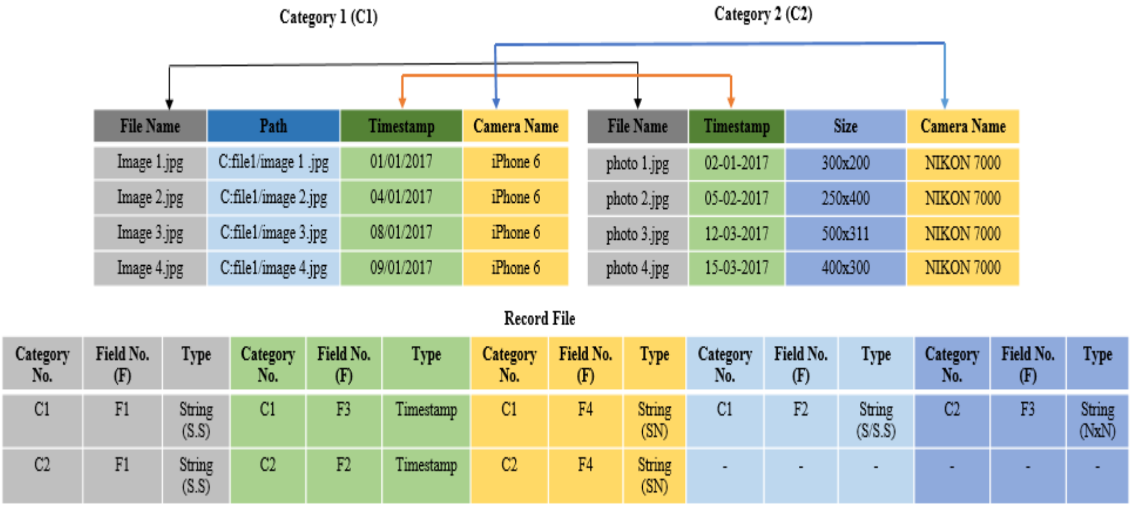
Algorithm 4-2 utilises to pre-process and identify meta from non-metadata for each metadata category.

Algorithm 4-2: Meta and Non-Metadata Identification Algorithm

```
Input: Meta and Non-Metadata fields in a category.
Output: Only Metadata fields.
------------------------------------------------------------
Process
Step 1: Read the first field of the category that has N of fields. Go to step 2.
Step 2: Read all value of the field. Go to step 3.
Step 3: If the value is Numerical, go to step 4.
                        else go to step 5.
Step 4: Read the next field. If the counter exceeds N, go to step 7.
                           Else go to step 2.
Step 5: Check the string variability. If there is a similar pattern across all values,
                               go to step 4.
                           Else, this column is not metadata, go to step 6.
Step 6: Remove this field, go to step 4.
Step 7: Return the metadata database.
Step 8: End
```

4.5.1.2 Characterisation

The solution to the problem of dataset characterisation can be achieved by using a rule-based system with a high level of fundamental conditions and rules. Rule-based systems are a method used to manipulate knowledge to interpret information in a useful manner (Aronson et al, 2005). It is built to simulate a human expert level in a narrow, specialised domain. It also uses a heuristics technique to guide the reasoning, thereby reducing the search area for a solution. Therefore, a small number of fundamental conditions areused such as string, consistency, numerical, Boolean, and timestamp. The characterisation algorithm uses these rules and conditions, which contain all the appropriate knowledge for matching similar categories. Regarding the string condition, the string variability algorithm will be used to produce a specific pattern that aids in checking and matching a similar field of strings across various categories. The consistency condition means all the string values within the field should have a fixed length of string with the same pattern while the numerical condition can be identified by measuring the range of the field within the category to match with another field in the compared category. Additionally, most files have two sizes, the physical and logical size, and there is a slight difference between them. The algorithm can identify the physical and logical size across various categories. The Boolean data type is a field with only two possible values: true or false. The timestamp is considered a fundamental condition because it exists within most files and applications. This algorithm can characterise most of the timestamp formats across various categories. The final output of the characterisation process is a record that contains all similar metadata categories, as shown in Figure 4-4.



Category 1 (C1)

File Name	Path	Timestamp	Camera Name
Image 1.jpg	C:file1/image 1 .jpg	01/01/2017	iPhone 6
Image 2.jpg	C:file1/image 2.jpg	04/01/2017	iPhone 6
Image 3.jpg	C:file1/image 3.jpg	08/01/2017	iPhone 6
Image 4.jpg	C:file1/image 4.jpg	09/01/2017	iPhone 6

Category 2 (C2)

File Name	Timestamp	Size	Camera Name
photo 1.jpg	02-01-2017	300x200	NIKON 7000
photo 2.jpg	05-02-2017	250x400	NIKON 7000
photo 3.jpg	12-03-2017	500x311	NIKON 7000
photo 4.jpg	15-03-2017	400x300	NIKON 7000

Record File

Category No.	Field No. (F)	Type	Category No.	Field No. (F)	Type	Category No.	Field No. (F)	Type	Category No.	Field No. (F)	Type	Category No.	Field No. (F)	Type
C1	F1	String (S.S)	C1	F3	Timestamp	C1	F4	String (SN)	C1	F2	String (S/S.S)	C2	F3	String (NxN)
C2	F1	String (S.S)	C2	F2	Timestamp	C2	F4	String (SN)	-	-	-	-	-	-

Figure 4-4: Characterisation Process

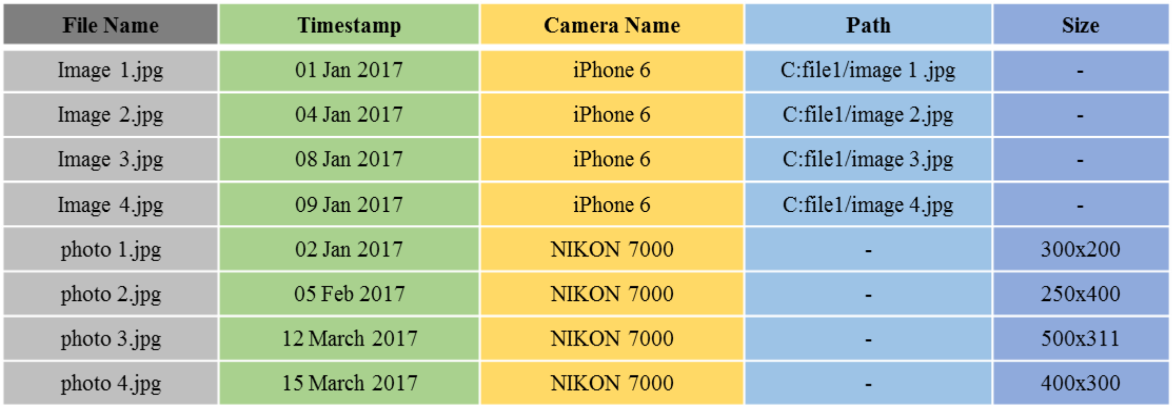
Algorithm 4-3 illustrates whole steps of characterising datasets.

Algorithm 4-3: Metadata Characterisation

```
Input: Several Categories.
Output: Record File.
Process
Step 1: Read the first category where there are N of categories. Go to step 2.
Step 2: Read the first field of the category that has M of field. Go to step 3.
Step 3: Check the field is a string or numerical. If string, go to step 4.
                                                  Else, go to step 6.
Step 4: Check the field if it is consistent or not. If it is consistent, go to step 7
                                                  Else, go to step 5.
Step 5: Check the string variability and extract the string pattern. Go to step 9.
Step 6: Check the field, if it is Boolean or not. If it is Boolean, go to Step 9
                                                  Else go to step 8.
Step 7: Calculate the length of string and check if it is timestamp and go to step 9
Step 8: Calculate the range of it and check whether it is physical size, logical size or
        timestamp, go to step 9.
Step 9: Save the number of category, field and type in the record file. Go to step 10.
Step 10: Read the next field. If the counter exceeds M, go to step 11.
                                                  Else, go to step 3.
Step 11: Read the next category. If the counter exceeds N, go to step 12.
                                                  Else, go to step 2.
Step 12: Return the record file.
Step 13: End.
```

4.5.1.3 Harmonisation

The problem of merging the similar datasets can be solved by applying the harmonisation algorithm, which is used to merge the similar categories based on the characterisation record. It can adjust the differences and inconsistencies among different measurements, methods, procedures, schedules, specifications, or systems to make them uniform or mutually compatible. Many fields within the metadata categories are stored in various forms across heterogeneous systems (i.e., timestamp, phone number, and file size). For example, the timestamp can be stored in several forms such as ('yyyy-MM-dd', 2014-04-19), ('dd/MM/yyyy', 19/04/2014), ('dd.MM.yyyy', 19.04.2014), ('yyyy-MM-dd''T''HH:mmXXX', 2014-04-19T21:41-04:00) or can be formed as a Unix timestamp, which is just number with 10 or 13 digits. Likewise, phone numbers can be represented in different ways (i.e., they can be stored with country codes or area codes). Additionally, the country code can be placed in varchar type (e.g., +91-9654637894). The file size can also be saved in various units of measurement (i.e., it is measured from the lowest to the highest in bits, bytes, kilobytes, megabytes, gigabytes). Consequently, the core of the harmonisation process is to merge similar categories systematically and make them uniform, as illustrated in Figure 4.5.



File Name	Timestamp	Camera Name	Path	Size
Image 1.jpg	01 Jan 2017	iPhone 6	C:file1/image 1 .jpg	-
Image 2.jpg	04 Jan 2017	iPhone 6	C:file1/image 2.jpg	-
Image 3.jpg	08 Jan 2017	iPhone 6	C:file1/image 3.jpg	-
Image 4.jpg	09 Jan 2017	iPhone 6	C:file1/image 4.jpg	-
photo 1.jpg	02 Jan 2017	NIKON 7000	-	300x200
photo 2.jpg	05 Feb 2017	NIKON 7000	-	250x400
photo 3.jpg	12 March 2017	NIKON 7000	-	500x311
photo 4.jpg	15 March 2017	NIKON 7000	-	400x300

Figure 4-5: Harmonisation Process

Algorithm 4-4 utilises to merge the similar metadata categories based on the Record File.

Algorithm 4-4: Metadata Harmonisation

```
Input: Several categories and the Record File.
Output: Main category.
--------------------------------------------------------------------------------

Process
Step 1: Read the first record in the record file that has N of records.
Step 2: If the field requires pre-processing, go to step 4,
          Else, go to step 9.
Step 3: If the field is numerical, go to step 4.
                                  Else go to step 7.
Step 4: If they are phone numbers, process and add them to the uniform field in
           The output category. Else, go to step 6.
Step 5: If they are file size, convert them to byte size and add them to the uniform
          field in the output category. Else, go to step 7.
Step 6: If they are phone numbers, process and add them to the uniform field in
           the output category. Else, go to step 8.
Step 7: If they are timestamps, process and add them to the uniform field in the
output
          category
Step 8: Add the fields in the sequence in the uniform field in the output category.
Step 9: Read the next record in the record file. If the counter exceeds N, go to
step 10.
                                                 Else, go to step 2.
Step 10: Return the output category.
Step 11: End.
```

4.6 Experimental Methodology

The purpose of the experiment is to evaluate and validate the characterisation and harmonisation. The following aims are defined:

- To differentiate between metadata and non-metadata
- To identify the metadata categories that are equivalent
- To merge similar categories

The experiment was repeated three times to evaluate the algorithms' performance. In this context, two standards should be considered (repeatability and reproducibility). Repeatability and reproducibility mean the outputs of the algorithm must be repeatable and reproducible to obtain the same results when using the same method on the same datasets in same or different laboratories.

4.6.1 Datasets

To investigate the conceptual designs of the system, there is an essential need for access to real investigative data. This is key in validating whether the novel approach is capable of merging similar datasets from several resources and applications. However, the limitations of available datasets already exist, especially with a heterogeneous domain. These limitations are a result of the difficulties in accessing real forensics data in academic communities. This requires long-term cooperation with security institutions. Additionally, even if datasets are available, they might not contain all the attributes that may be required for evaluating the proposed system. Consequently, four forensics cases (three private and one public) from multiple resources, such as smartphones, computers, and external hard drives, were used.

The reasons for using public cases were because of the limited number of real forensic cases and validating the reliability and effectiveness of the approach.

For this research, four cases were previously analysed manually during normal forensic analysis. Notably, it is not suitable to use new cases without previous knowledge to evaluate a hypothesis for the research. It produces the difficulty of assessing the results to determine whether they were accurate and mapped to the expected findings. The details of these cases are provided in the following:

1. Public case (data leakage case)

This case was generated by The National Institute of Standards and Technology (NIST) for training purposes on how to deal with heterogeneous evidence resources (NIST, 2015). It consists of three evidential resources (a personal computer and three USB removable storage devices), which were acquired in a forensically sound manner in two forms: Encase images and DD images. The evidence across these evidential resources is diverse between emails, user accounts, internet browsers, and various documents. Table 4-3 explains the evidence types.

Table 4-3: First Case Details

Id	Type & size	#Files	#Files after carving	#Files after KFF	#Evidence				
					File List	Email	IE browser	Chrome browser	Total
1	Personal Computer.E01 / 20 GB	139565	219800	143180	61	18	22	24	152
2	Removable Media.E01 / 4 GB	55	1085	0/1085	11	-	-	-	11
3	CD.E01 /700 MB	3	867	0/867	15	-	-	-	15

The methodologies of the data leakage case that were achieved by the suspect were local computer usage, network transmission, and storage devices. The local computer was used to create the crime by installing some applications to alter and leak some confidential data out the company. The network transmission was achieved via email and cloud storage services to send and upload secret data. Storage devices were used to leak the important data that were difficult to leak in other ways.

2. Private cases

Three private cases were obtained from the Republic of Iraq related to three crimes in the province of Anbar. The first case consisted of two evidence sources: a smartphone and a USB memory stick belonging to the same person. The smartphone was a Samsung mobile with Android OS containing evidential data from different applications. In the scenario for this case, the terrorist used the phone to do many actions. He sent and received SMS orders to execute a terrorist action. Viber and Facebook messenger were also used to send and receive images related to people and locations to execute missions. In the same context, the phone's camera was used to take pictures of places that could be attacked by car bombs or something else. Additionally, the internet browser was used to search for how to make films for executed terrorist actions. The USB memory stick was used to save video and audio that was recorded. Table 4-4 illustrates the summary of two evidence sources that exist in this case.

Table 4-4: Second Case Details

Id	Type / Size	#Files before carving	#Files after carving	#Files after KFF	#Evidence						
					File list	EXIF	Internet	Viber	SMS	FB	Total
1	Smart phone.DD / 30 GB	239574	262721	260914	86	150	23	78	12	240	568
2	USB memory stick.E01 / 2 GB	323	324	324	4	-	-	-	-	-	4

The second case comprises two external hard drives belonging to a group of terrorists. These drives were used as archives to store the terrorist actions that were achieved by terrorists. This case contains several videos that show how they attack villages and record their actions. Likewise, the recorded audio comprises varies things, such as recorded calls, recorded talks, and radical songs while the documents include various details, such as their plans to kill people, attack army colonies, and many more. Table 4-5 shows the details of these evidential sources.

Table 4-5: Third case details

Id	Type / Size	#Files	#Files after carving	#Files after KFF	Evidence Type		
					File List	EXIF	Total
1	External Hard Drive1.E01 / 42.8 GB	4501	22648	22018	503	42	545
2	External Hard Drive 2.E01 / 40.8 GB	7310	10325	9922	768	325	1093

The third case consists of two evidential resources: a desktop computer and a USB flash drive. It is about kidnappings that were executed by a group in order to gain money from hostages' families. The desktop computer and USB flash drive contained evidential data, such as videos and pictures of their actions, as well as a number of recorded conversations between kidnappers and a hostage's family for negotiation purposes. Table 4-6 illustrates the relevant data of this case.

Table 4-6: Fourth Case Details

Id	Type / Size	#Files before carving	#Files after carving	#Files after KFF	Evidence Type		
					File List	EXIF	Total
1	Desktop Computer.E01 / 37.7 GB	242510	318369	182870	178	203	381
2	USB Flash Drive.E01 / 13 GB	66882	92531	66062	54	43	97

4.6.2 Experimental Setup

During the metadata extraction phase, various metadata was generated and extracted from these resources, such as file systems and applications, as illustrated in Table 4.7. The metadata of these images was exported into individual comma separated value (CSV) files. Several CSV files contain missing metadata features within the same category because they were extracted from heterogeneous resources. For instance, the EXIF metadata, which was extracted from smartphone datasets, has complete metadata features, such as filename, timestamp, camera manufacturer and model, size of the image file, size of the image (width x height), IOS, latitude, longitude, and GPS timestamp. The EXIF metadata within computer datasets, however, contained missing features, such as IOS, latitude, longitude, and GPS timestamp.

Similarly, internet browsing metadata is different across forensic images based on platforms and applications. In computer images, two browsers (Firefox and Chrome) have features such as URL, visit count, visit timestamp, referrer URL, title, and profile. Whereas, smartphone browsers only have URL, visit count, and visit timestamp. The smartphone images contain SMS and Viber applications and both serve to send and receive messages. Many features between SMS and Viber are similar such as account number, sending timestamp, delivery timestamp, message body, status, seen, and recipient number. They also contain binary-based data such as opened, deleted, seen, etc. Regarding the file system, heterogeneous OSs are included across these images, but most of these OSs hold common features, such as file name, timestamp, size, etc. Likewise, the emails of two images include mutual features in addition to the email body, which is presented as a non-metadata characteristic.

Table 4-7: Overview of Experimental Datasets

Id	Type	Evidence Type				
		File List	Messaging	Pictures	Internet	Emails
1	PC.	NTFS	-	EXIF	IE, Chrome	Outlook
	Memory stick 1	FAT	-	-	-	-
	CD	CDFS	-	-	-	-
2	Hard drive	NTFS	-	EXIF	-	-
	Hard drive	NTFS	-	EXIF	-	-
3	Smartphone	Ext4	SMS, Viber	EXIF	Internet browser	-
	Memory stick	NTFS	-	-	-	-
4	PC.	NTFS	-	EXIF	-	-
	Memory stick	FAT	-	EXIF	-	-

As presented in Table 4-7, case one contains four metadata categories (a total of seven disparate datasets) but only two categories require merging (i.e., file systems, internet browsers). The second case includes two metadata categories (a total of four disparate datasets) where the similar categories (File List and EXIF) should be harmonised while the third case comprises of four metadata categories (a total of six disparate datasets) with two of these categories (File List and messaging) needing to be merged. The fourth case contains two metadata categories (a total of four disparate datasets) where the similar categories (File List and EXIF) should be merged.

4.6.3 Results

All the metadata categories within each case were provided to the system in a single instance. As illustrated in Table 4-7, there are four categories across the four cases (email, Viber, and SMS) containing non-metadata fields. Therefore, the meta from non-metadata identification based on email, Viber, SMS, and Skype categories was achieved successfully. All non-metadata fields were also automatically eliminated.

To identify the categories, the characterisation process was used to generate a record file. This record contains the categories that are similar as being represented in the previous section. To make it clear, the algorithm takes a dataset and checks it with all datasets in sequence. Then, it counts the number of identical fields (I) within the compared datasets against different fields (D). There is a threshold used to decide whether the two datasets are similar or not. This threshold has been modified five times to obtain the ultimate threshold, as shown in Table 5-8. The experiment results prove that when the threshold of I is greater than or equal to D, the best results can be obtained. Consequently, the algorithm creates a record that contains similar files.

Table 4-8: Experimental Results

Case ID	Threshold	Threshold									
		I < D		I <= D		I == D		I >= D		I > D	
		#	%	#	%	#	%	#	%	#	%
1	True Positive	0	0	2	28.5	2	28.5	7	100	5	71.5
	False Positive	7	100	5	71.5	5	71.5	0	0	2	28.5
2	True Positive	0	0	0	0	0	0	4	100	2	50
	False Positive	4	100	4	100	4	100	0	0	2	50
3	True Positive	0	0	1	14.3	2	28.5	7	100	6	85.7
	False Positive	7	100	6	85.7	5	71.5	0	0	1	14.3
4	True Positive	0	0	0	0	0	0	4	100	2	50
	False Positive	4	100	4	100	4	100	0	0	2	50

Table 4-8 shows the impact on the performance of the characterisation algorithm applied on the four cases across different thresholds. The worst results have been obtained when using the threshold of I less than D, where the algorithm matched the files that are completely different across all four cases. The threshold of I less than or equal to D showed poor results. The proportion of proper merging was small with 25.5 % for case one, 0 % for case two, 14.3 % for case three, and 0 % for case four. Using the equality threshold, the results were enhanced a little with only two out of six datasets matching within case three while, in other cases the results were unchanged. This is still unacceptable. The threshold of I greater than D showed a good proportion of matching compared with aforementioned thresholds with 71.5 % for case one, 50 % for case two, 85.7 for case three, and 50 % for case four. Ultimately, the threshold of I greater than or equal to D gave the best results with 100% of the true positive across all four cases. Noticeably, this threshold might be changeable according to the nature of the study cases and their metadata categories.

To merge the similar categories, the harmonisation algorithm took the record file and the datasets of each case. The algorithm merged and produced new datasets representing the five main categories. The main five categories were messaging, EXIF, emails, file list, and internet browsing metadata. In addition, the algorithm's performance and accuracy completely depend on the record generated by the characterisation algorithm. Accordingly, it merges and harmonises similar categories together in one file. Although the results of this algorithm are encouraging, there are some errors being detected owing to the only binary-based data that exists within the messaging category. Case three showed that the most challenging category was Viber-SMS, where two fields of binary data within each category were wrongly merged. These were the "seen" field merged with the "deleted" field and the "read" field merged with the "hidden field". However, binary data represents with only two values (0 or 1) and does not contain valuable information compared to other fields of SMS and Viber categories.

4.7 Discussion

From the aforementioned results, the novel approach of harmonising datasets was capable of identifying and merging similar categories. This can lead to overcoming the heterogeneity of data and to not repeat the digital forensics process many times on the same categories. In addition, the harmonising approach completely depends on the characterisation process, which uses the rule-based system with the possibility of scaling by adding new rules and conditions. However, the implication of adding new rules and conditions should be investigated to check the compatibility of the approach.

One feature of the characterisation algorithm is its ability to generate a pattern for some string and numerical fields within categories that consist of a specific structure and format. This pattern easily identifies a field to match other fields from other datasets. However, some fields contain data without format or structure, which the algorithm recognises as non-metadata fields (i.e., email subject).

To create a record for all similar datasets, the characterisation process revealed that the number of identical fields greater than or equal to the number of nonidentical fields between compared datasets gave the best results. Based on the results, this threshold considers all possibilities from matching categories of all four cases while other thresholds showed the characterisation process failed to identify matching datasets. This means most datasets within each case contained a small number of similar fields, such as binary data, file size, and file name. Therefore, by using the thresholds of I less than D, I less than or equal, and I equal to D, the algorithm considered a small number of identical fields across different datasets and incorrectly matched them.

In contrast, the characterisation algorithm showed its ability to recognise the categories containing many similar fields. These fields include binary-based data which mostly exist within the messaging category (i.e., Sent, Opened, Seen, and Read). Noticeably, the number of identical fields becomes distinguishable compared to other categories. However, the datasets, which contain a number of similar fields, cause confusion for the harmonisation algorithm to precisely merge similar fields. In this case, the algorithm depends on the sequence of similar fields in the merge. Thereby, it might harmonise the wrong fields.

4.8 Conclusion

The evidentiary nature of digital forensics has changed over the years and cases increasingly contain multiple devices and applications. Existing digital forensic tools are struggling to keep pace in achieving modern forensic investigations, such as examining and analysing many systems and applications at once. Therefore, this chapter proposed and demonstrated an automated approach for metadata characterisation and harmonisation to overcome the heterogeneity issues. In the experimental study, the live forensic data was used to evaluate the novel process. The results have shown that the characterisation and harmonisation process can be appropriated to merge and create a common standard across different formats for a similar metadata category. Although the harmonisation algorithm has not been able to merge all binary data fields, the binary data provides minimal valuable information within the investigation.

5 Clustering Approach

This chapter proposes a clustering approach based fuzzy c-means (FCM) and k-means, and k-medoids algorithms to identify the evidential files and isolate the non-related files based on their metadata. A series of experiments using real-life forensic cases was conducted to evaluate the proposed approach. This chapter aims to prioritise large proportions of evidence and reduce the volume of benign files to be analysed—thereby reducing the time taken and cognitive load on the investigator.

5.1 Introduction

The amount of digital forensic data has significantly increased in recent years (Quick et al., 2016). However, the proportion of evidence within this data is relatively small. Several methods have been used to find evidential artefacts in an automated way, such as unsupervised machine learning algorithms (e.g., clustering algorithms) (Harichandran et al., 2016). Clustering algorithms group data into clusters containing objects sharing common characteristics (Xu and Wunsch, 2005). The algorithms divide the data without any prior knowledge about the data. This exists in most digital forensic cases containing data that are not labelled. Therefore, there is a need for intelligence to reduce the volume of data to an acceptable level—where ideal performance would be defined as identifying all artefacts of interest and leaving behind all benign files. This can lead to grouping only suspicious data and thereby minimising the burden on the investigators. However, it is difficult to apply clustering algorithms on files themselves. Therefore, metadata categories can be used instead (Gupill, 1999). Data categories, including files, databases, documents, pictures, media files, and web browsers hold valuable information that can be used to answer important questions in a forensic investigation. Examples of the questions include, who did what to a file, when they did it, and where it was carried out.

5.2 Clustering Theory

Data clustering is a powerful technique in data examination and analysis. It is also a standard process to analyse multivariable datasets. It is used to group similar objects in one cluster and dissimilar ones in other clusters. There are two main methods used to obtain these clusters: the partitioning and hierarchical methods (Cristogor et al, 2002). In the partitioning category, the aim is to split the data into a fixed number of non-overlapping subsets or clusters using k-means and k-medoids (Äyrämö and Kärkkäinen, 2006). While the hierarchical category can be further subdivided where data is divided into a set of nested clusters as a tree (i.e., single link and complete link). However, this chapter will only focus on partition algorithms, which were widely applied on digital forensic data.

5.2.1 K-Means Algorithm

K-means is one common algorithm of unsupervised machine learning approaches. It is used to classify unlabelled data through a certain number of groups (a predefined number of clusters) (Wagstaff et al., 2001). These predefined clusters are used to generate centres to categorise the unlabelled data thereon. As much as possible, these centres are chosen randomly by the algorithm far away from each other because a good choice of centres results in valuable results. Afterwards, next is to take each point and calculate the distance between the point and the centres. The closest distance is considered and then the point is assigned to that centre. The centres should be recalculated after completing the assignment of all points. The process of changing centres' locations should continue until there is no change happening to their locations. Finally, the K-means algorithm aims to minimise the squared error function based on the following equation (Hartigan and Wong, 1979):

$$J(V) = \sum_{i=1}^{c}\sum_{j=1}^{c_i}(\|x_i - \ v_i\|)^2$$

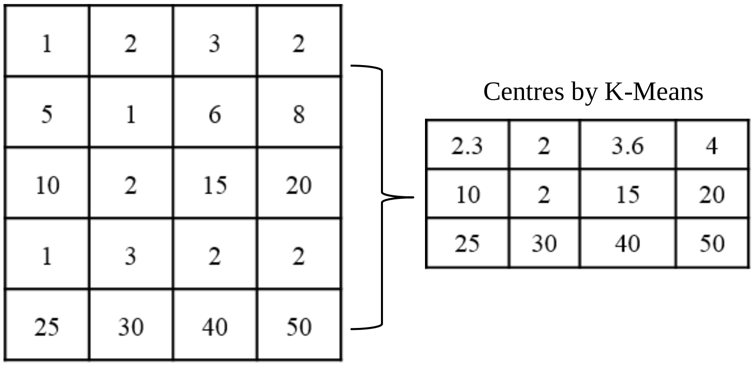
where,

'||xi - vj||' is the Euclidean distance between xi and vj.

'ci' is the number of data points in i^{th} cluster, and

'c' is the number of cluster centres.

Figure 5-1 shows an example of a dataset with five objects were to generate three centres by the k-means algorithm. The centres of three clusters were calculated by finding the mean of all features related to a similar cluster.



1	2	3	2
5	1	6	8
10	2	15	20
1	3	2	2
25	30	40	50

Centres by K-Means

2.3	2	3.6	4
10	2	15	20
25	30	40	50

Figure 5-1: K-Means Centres Generation

The details of k-means method illustrated in Algorithm 5-1 (Wagstaff et al, 2001).

Algorithm 5-1: K-Means

```
Input: Dataset X.
Output: Number of Clusters (c).
Process
Step 1: Let X = {x1, x2, x3,....,xn} is a dataset.
Step 2: Randomly select ‘V’ centres based on the number of clusters where V = {v1,v2,…….,vc} is a set of centres.
Step 3: Calculate the distance between each vector within the dataset and cluster centres.
Step 4: Assign the vector to the cluster centre whose distance from the cluster centre is minimum of all the centres.
Step 5: Recalculate the new centres using:
        $v_i = \left({}^{1}/_{c_i}\right) \sum_{j=1}^{c_i} x_i$
        where, ‘ci’ represents the number of vectors in i^th cluster.
Step 6: Recalculate the distance between each vector and new obtained cluster centres.
Step 7: If no data point was reassigned then stop, otherwise repeat from step 3).
Step 8: End
```

5.2.2 K-Medoids Algorithm

The k-medoids algorithm is a clustering approach using to partition the dataset into a fixed number of clusters. It is relatively similar to the k-means algorithm but it is used to find medoids (which means the centre point of a cluster) in a group (Park and Jun, 2009). These centres represent the minimal summation of objects' dissimilarities within the dataset. The details of the k-medoids method is illustrated in Algorithm 5-2.

Algorithm 5-2: K-Medoids

```
Input: Dataset X.
Output: Number of Clusters (c).
Process
Step 1: Let X = {x1, x2, x3,....,$x_i$} is a dataset.
Step 2: Randomly select ‘M’ medoids based on the number of clusters where M = {m1,m2,.......,$m_c$}.
Step 3: Calculate the distance between each vector within the dataset and medoids to find the closest medoids by applying:
    $(x_n, m_n) = \sum_{i=1}^{n} |x_i - m_i|$
Step 4: Assign the vector to the medoids with a minimum distance.
 Step 5: For each medoid m and each vector x associated to m apply step 6 and 7.
Step 6: Swap m and x to compute the total distance by the equation in step 3.
Step 7: Select x as medoids that contain the lowest distance.
Step 8: If there is no change in the assignments repeat steps 3, 4, and 5 alternately.
Step 8: End
```

5.2.3 Fuzzy C-Means

Fuzzy c-means (FCM) is a partitioning technique of data clustering wherein each vector within the dataset belongs to a cluster. These points contain a membership grade that is used to specify the degree of vector to a cluster. The main benefit of this algorithm is to measure gradual memberships of the vectors within datasets as [0,1] to assign them to the clusters. It works to minimise the object function of the following equation (Bezdek et al., 1984):

$$J(V) = \sum_{i=1}^{c}\sum_{j=1}^{c_i} u_{ij}^{m} \, (\|x_i - \; c_j\|)^2$$

where

m is a real number which greater than 1,

x_i is the vector within the dataset with *i* vectors,

c_j is the number of cluster with *j* clusters,

u_{ij} is the degree of membership of x_i in the cluster *j*.

Algorithm 5-3: Fuzzy C-Means

Input: Dataset X.

Output: Number of Clusters (c).

Process

Step 1: Let X = {x1, x2, x3,….,x$_n$} is a dataset.

Step 2: Calculate the centres vectors by:

$$c_j = \frac{\sum_{i=1}^{n} u_{ij}^{m} \, . \, x_i}{\sum_{i=1}^{n} u_{ij}^{m}}$$

Where n= number of features in the vectors,

$$u_{ij}^{m} = \frac{1}{\sum_{k=1}^{c}(\frac{\|x_i - \; c_j\,\|}{\|x_i - \; c_k\,\|})^{\frac{2}{m-1}}}$$

Step 3: Calculate $J(V)$ between each vector within the dataset and centres to find the closest centres:

$$J(V) = \sum_{i=1}^{c}\sum_{j=1}^{c_i} u_{ij}^{m} \, (\|x_i - \; c_j\|)^2$$

Step 4: Assign the vector to the centres that has the minimum value of J(v).

Step 5: End

5.3 Experimental Setup

Clustering is the most powerful method for analysing the data which can divide a dataset into a number of distinguished groups (Harichandran et al, 2016). However, clustering algorithms generally have no internal way to handle textual data and missing values. Instead, a common solution is to represent each string feature by a numerical value and fill-in the missing values in a pre-processing step. Consequently, the traditional way for numerating leads to the two main problems: huge dimensionality and sparse distribution. While the filled-in values are inherently less reliable than the observed data. To overcome these issues, Figure 5-2 illustrates an approach to cluster both the string and numerical values with recodes that contain missing values.

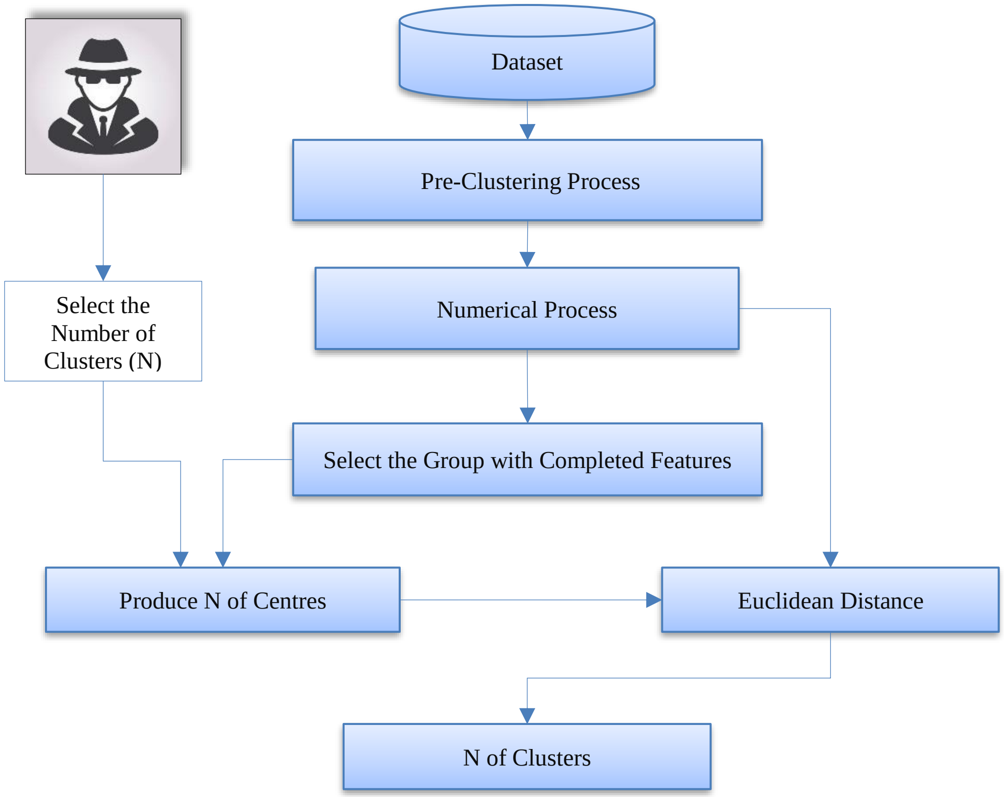


Figure 5-2: Clustering Process

1. The Pre-Clustering Process is to split up the dataset vectors into groups that are filled-in similar features. This leads to identify the group containing vectors with completed features.
2. Numerical process: it is necessary to convert string values to numeric values in order to use clustering techniques within forensic investigations. This algorithm uses a developed method to numerate the string values, isolate the non-defined features, and avoid the problems of traditional numerical methods. Firstly, it neglects the predefined char such as "space",":", and ".". It will then predict a weight for both the string characters and numeric characters; it clears that numeric characters have the ASCII values between 48 to 57. For instance, if a string value such as "300x200" contains mixed characters, the percentage of the string characters is (1/7) * 100 = 14.28%, while the percentage of numeric characters is (6/7) * 100 = 85.72%. Therefore, the algorithm will consider the given example as a numeric value by neglecting the string values and becomes 300200. In contrast, a string value such as "apple iPhone 6" contains mixed characters, the percentage of string characters is around (11/12) *100 = 91.67%, and the percentage of numeric characters is (1/12) *100 = 8.33%. In this case, the algorithm will consider this as a string and apply the numerical process to predict a numerical value of the textual value. The algorithm will create a database which contains unique strings and dedicate them unique numbers. For instance, the first string will be given number one, where the rest will be checked with the database to find the distance between the unique strings and the new one. To achieve that,

the following steps illustrate how the algorithm can calculate the distance between two strings:

- The extra spaces from the strings will be removed.
- Spaces will be added to the end of the string which contains fewer characters to make the length of two strings is equal
- The circular shift operation will be applied to one of these values to obtain all string probabilities as a tuple and produce several strings to match them with another string. The circular shift is a special kind of cyclic permutation, which, in turn, is a special kind of permutation. Formally, a circular shift is a permutation X of n characters in the tuple such that:

$$X(i) = X(n - i)$$

where n is the length of string, i =0, …., n-1.

- These probabilities will be matched with the source string to discover the distance between them. In addition, the algorithm will calculate the difference between the characters in the same position (i.e., If s[j] equals t[j], the difference is 1. If s[j] does not equal t[j], the difference is 0. The following equation calculates scores between the source string and all the probabilities of the target string. It then takes the maximum score:

$$Score(i) = \sum_{j=1}^{j=n} \frac{(tj - sj)}{n}$$

where i represents the probabilities of target string, while n represents n the length of string.

- If the maximum score is greater than 0.7, the target string will be given a numerical value as following:

  $$Sn = Nq + (1 - MaxScore)$$

  where Sn is the numerical value of the target string, and Nq is the numerical value of the source string. For clarity, 0.7 is a threshold to identify the similarity between two strings as it has been changed several times to obtain the ideal value which is 0.7.

- If the maximum score is less than 0.7, the algorithm will check the next string in unique database and so on. If there is no matching, the target string will consider as a unique string and will be given a numerical value as follows:

  $$Sn = Ln + 1$$

  where Sn is the numerical value of target string, and Ln is the last number in unique database.

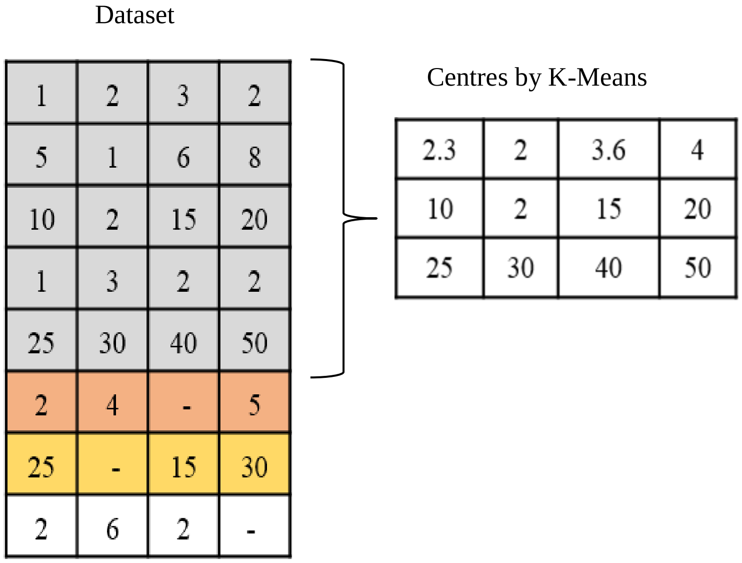
3. Centres generation: the filled-in group with completed features will be selected to generate centres by using one of the current methods such as k-means, k-medoids, and fuzzy c-means (FCM) clustering. The investigator will select the number of centres before the process begins, where these clustering algorithms are only used to predict the centres of the clusters.
4. Euclidean distance (ED) (Danielsson, 1980): ED is matrices of the squared distances between points. The centres will be used to find the other vectors using ED. Each pre-cluster group contains specific features that will only be calculated with same features of centres. Afterwards, the shortest distance

between a vector and a centre, the vector will be assigned to this particular cluster. ED can be calculated by using following equation:

$$d = \sum_{i=1}^{n} (x_i - \ y_i)^2$$

where d is the distance between two vectors, n is the length of the vector, xi is the first vector, and yi is the second vector.

For instance, the sixth vector within the dataset in Figure 5-3 includes a missing value. In this case, the algorithm will calculate the distance between this vector and the three centres by ignoring the third feature in all centres because it is already missed.



Dataset

1	2	3	2
5	1	6	8
10	2	15	20
1	3	2	2
25	30	40	50
2	4	-	5
25	-	15	30
2	6	2	-

Centres by K-Means

2.3	2	3.6	4
10	2	15	20
25	30	40	50

Figure 5-3: Appling K-Means

Figure 5-4 shows applying the ED between the vector and the centres without the third feature; the shorter distance is 2.25. Therefore, this vector belongs to the first centres.

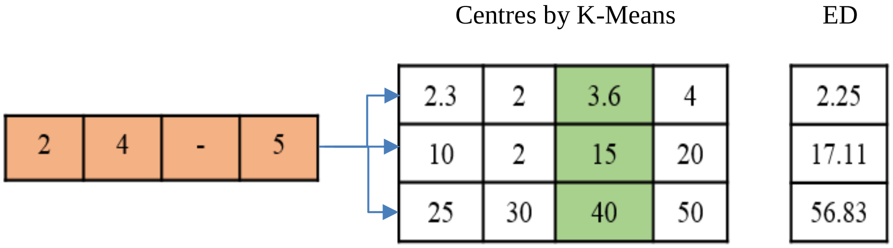


Figure 5-4: Appling Euclidean Distance

5.4 Experimental Results

This experimental hypothesis was to determine that notable artefacts can be grouped in the same clusters with a minimum number of benign data. Therefore, two questions are proposed:

- What influence does clustering algorithms have on the accuracy?
- What influence does the cluster size have on algorithms that are used?

For each category within the four cases, the clustering procedure was run three times to ensure the stability of the developed process. For the experiment, six clustering sizes were selected (15, 25, 35, 50, 75, 100) to obtain a view of clustering performance across all categories using FCM, k-means, and k-medoids algorithms. In addition, it is important to investigate the influence of cluster size on the algorithm itself because the categories with large amounts of data might be clustered in a good way using the large sizes. The following section shows the results were obtained based on three clusters containing a high number of notable artefacts. These results illustrate a proportion of notable versus the benign data with the actual number of artefacts.

Table 5-1 shows cases 1, 2, 3, and 4’s details, which were obtained from the characterisation and harmonisation process as explained in Chapter Five. Based on these results, the clustering process was executed.

Table 5-1: Case Details

Case ID	#Files	Evidence Type						
		File List	Email	EXIF	Messaging	FB	Internet browsers	Total
1	145132	87	18	-	-	-	45	150
2	31940	1270	-	367	-	-	-	1637
3	170389	90	-	62	90	240	23	505
4	248932	232	-	246	-	-	-	478

5.4.1 Case 1 Analysis

Table 5-2 shows the results of the file list category across three algorithms with six configurations of cluster size. Noticeably, the clustering based on file list with 15 and 25 cluster sizes provided successful isolation for the notable artefacts with 100% proportion across FCM while only the 15-cluster size with k-means and k-medoids grouped all notable artefacts within the top three clusters. It is obvious 83.9% of notable files were obtained in only a single cluster out of 15 clusters across all methods with a small proportion of benign files, which was less than 12.7%. In addition, a good proportion of the benign data with at least 85.75% based on FCM, 87.75% based on k-means, and 87.51% based on k-medoids were eliminated within top three clusters. By increasing the number of cluster size (e.g., 35) the proportion of notable files was slightly decreased within the top three clusters compared with their counterparts from 15 and 25 cluster sizes while the proportion of benign files decreased also. However, by increasing the number of cluster size configurations (i.e., 50, 75, and 100), the proportion of benign and notable artefacts that were presented within the top 3 clusters decreased gradually. Indeed, more than 4.6% and 88.62% of notable and benign artefacts were grouped in other clusters. To evaluate the performance of used algorithms with File List, the FCM showed its ability to group the notable artefacts with a good proportion compared to k-means and k-medoids. However, it is also noteworthy that k-means and k-medoids showed better performance than FCM in isolating benign files. Regarding cluster sizes, the results revealed that the small configurations gave better grouping of notable artefacts than the large size configurations, but the large size configurations contained a small proportion of benign files.

Table 5-2: Results of File list (Case 1) (✓: Notable; ×: Benign)

Centres Generation	Cluster ID		Cluster Size											
			15		25		35		50		75		100	
			✓	×	✓	×	✓	×	✓	×	✓	×	✓	×
FCM	1	#	73	18357	73	16969	59	15968	58	15824	58	15761	56	5570
		%	83.9	12.7	83.9	11.7	68	11	66.6	10.9	66.6	10.8	64.4	3.8
	2	#	13	366	13	366	13	366	13	366	13	366	13	366
		%	15	0.25	15	0.25	15	0.25	15	0.25	15	0.25	15	0.25
	3	#	1	1878	1	626	12	342	12	342	12	178	12	11
		%	1.1	1.3	1.1	0.45	13.8	0.23	13.8	0.23	13.8	0.12	13.8	0.007
	Rem.	#	0	124444	0	120784	3	127740	4	128513	4	128740	6	139098
		%	0	85.75	0	87.6	3.2	88.52	4.6	88.62	4.6	88.83	6.8	95.95
K-Means	1	#	73	16973	61	15994	59	15832	58	15782	35	2767	56	5537
		%	83.9	11.7	70.1	11	68	10.9	66.6	10.8	40.2	1.9	66.6	3.8
	2	#	13	366	13	366	13	366	13	366	22	12853	13	366
		%	15	0.25	15	0.25	15	0.25	15	0.25	25.3	8.9	15	0.25
	3	#	1	626	12	345	12	133	12	131	13	366	12	11
		%	1.1	0.43	13.8	0.23	13.8	0.09	13.8	0.09	15	0.25	13.8	0.007
	Rem.	#	0	127080	1	128340	3	128714	4	128766	17	129059	6	139131
		%	0	87.62	1.1	88.52	3.2	88.76	4.6	88.86	19.5	88.95	4.6	95.95
K-Medoids	1	#	73	16995	71	15975	59	15971	58	15780	58	15369	54	2543
		%	83.9	11.7	81.6	11	68	11	66.6	10.9	66.6	10.6	62	1.75
	2	#	13	366	13	366	13	366	13	366	13	366	13	366
		%	15	0.25	15	0.25	15	0.25	15	0.25	15	0.25	15	0.25
	3	#	1	796	1	626	12	342	12	131	12	11	12	11
		%	1.1	0.54	1.1	0.45	13.8	0.23	13.8	0.09	13.8	0.007	13.8	0.007
	Rem.	#	0	126888	2	128078	3	128366	4	128768	4	129299	8	142128
		%	0	87.51	2.3	88.3	3.2	88.52	4.6	88.76	4.6	89.14	9.2	97.9

Regarding the category of internet data, Table 5-3 illustrates the detailed results. The clustering-based FCM and k-medoids show the proportion of notable artefacts in the top three clusters using 15 cluster size is exactly the same at 86.7%, and the proportion of benign files is also relatively similar at about 61%. For the same configuration, the clustering-based k-means showed that the results were getting better in comparison with the results obtained from their counterparts with a proportion reaching 95.4%. In contrast, the density rate of benign files within the top three clusters was comparatively large with more than 40% on average. However, this phenomenon might happen because the small number of files was provided in the clustering procedure.

In comparison, the results from the configuration with large cluster sizes (25 -100) based on FCM presented an accepted proportion of notable artefacts within the top three clusters with more than 70% on average; but the clustering-based k-means and k-medoids showed a challenging proportion where more than a half of the notable files were grouped out of the top three clusters. Noticeably, the proportion of benign artefacts dropped to reach less than 25% with large configurations of cluster size across all methods, indicating the ineffectiveness of these settings and most investigations are required on these configurations.

Table 5-4 illustrated the results of the email category. It demonstrates that all notable and benign artefacts were grouped in one cluster for all algorithms across all cluster sizes. This phenomenon happened because there were only 19 files included in the email category. This small number of files is less than the cluster sizes, which led to grouping them in one cluster.

Table 5-3: Results of Internet Category (Case 1)

Centres Generation	Cluster ID		Cluster Size											
			15		25		35		50		75		100	
			✓	×	✓	×	✓	×	✓	×	✓	×	✓	×
FCM	1	#	19	43	25	63	25	62	28	57	11	34	14	40
		%	42.2	13.9	55.5	20.3	55.5	20	62.2	18.4	24.4	10.9	31.1	12.9
	2	#	14	33	7	69	6	42	7	76	8	7	10	13
		%	31.1	10.6	15.5	22.2	13.3	13.5	15.5	24.5	17.8	2.2	22.2	4.2
	3	#	6	42	6	5	4	3	5	17	8	16	7	19
		%	13.3	13.5	13.3	1.6	8.8	0.9	11.1	5.8	17.8	5.1	15.5	6.1
	Rem.	#	6	192	7	173	10	203	5	160	18	253	14	238
		%	13.4	62	15.5	55.9	22.4	65.6	11.1	51.3	40	81.8	31.2	76.8
K-Means	1	#	33	76	12	17	7	13	4	5	4	5	4	26
		%	73.3	24.5	26.6	5.5	15.5	4.2	8.8	1.6	8.8	1.6	8.8	8.4
	2	#	6	42	8	27	6	6	4	11	4	26	3	0
		%	13.3	13.5	17.7	8.7	13.3	1.9	8.8	3.5	8.8	8.4	6.6	0
	3	#	4	18	7	18	6	42	4	28	3	4	3	2
		%	8.8	5.8	15.5	5.8	13.3	13.6	8.8	9.1	6.6	1.3	6.6	0.65
	Rem.	#	2	174	18	248	26	261	33	266	34	275	35	282
		%	4.6	56.2	40.2	80	57.8	80.3	73.6	85.8	75.8	88.7	78	90.95
K-Medoids	1	#	19	43	15	23	6	7	4	5	4	5	4	26
		%	42.2	13.9	33.3	7.4	13.3	2.2	8.8	1.6	8.8	1.6	8.8	8.4
	2	#	14	35	11	33	6	42	4	8	4	26	3	0
		%	31.1	11.3	24.4	10.6	13.3	13.5	8.8	2.6	8.8	8.4	6.6	0
	3	#	6	42	7	20	4	6	4	26	3	2	3	2
		%	13.3	13.5	15.5	6.4	8.8	1.9	8.8	8.4	6.6	0.65	6.6	0.65
	Rem.	#	6	190	12	234	29	255	33	271	34	277	35	282
		%	13.3	61.3	26.8	75.6	64.6	82.4	73.6	87.4	75.8	89.3	78	90.95

Table 5-4: Results of Email Category (Case 1)

	Cluster ID	Cluster Size

Centres Generation			15		25		35		50		75		100	
			✓	×	✓	×	✓	×	✓	×	✓	×	✓	×
FCM	1	#	18	1	18	1	18	1	18	1	18	1	18	1
		%	100	100	100	100	100	100	100	100	100	100	100	100
	Rem.	#	0	0	0	0	0	0	0	0	0	0	0	0
		%	0	0	0	0	0	0	0	0	0	0	0	0
K-Means	1	#	18	1	18	1	18	1	18	1	18	1	18	1
		%	100	100	100	100	100	100	100	100	100	100	100	100
	Rem.	#	0	0	0	0	0	0	0	0	0	0	0	0
		%	0	0	0	0	0	0	0	0	0	0	0	0
K-Medoids	1	#	18	1	18	1	18	1	18	1	18	1	18	1
		%	100	100	100	100	100	100	100	100	100	100	100	100
	Rem.	#	0	0	0	0	0	0	0	0	0	0	0	0
		%	0	0	0	0	0	0	0	0	0	0	0	0

5.4.2 Case 2 Analysis

The results of file list within Case 2 are presented in Table 5-5. This case shows that 100% of notable artefacts were grouped within the top three clusters across all cluster sizes by using the FCM algorithm. For the same configurations, the proportion of benign artefacts was also small with a range of 13%-17% being presented in the top three clusters. Interestingly, the clustering based 15 and 25 cluster size configurations for FCM and 15, 25, and 35 cluster size configurations for k-means and k-medoids configurations show that all notable files were clustered within a single cluster. The proportion of irrelevant files was relatively small across all configurations of cluster sizes and algorithms. Notably, there is a slight difference in this proportion across the three algorithms that were used, where k-means and k-medoids achieved better results than FCM.

The large configurations of cluster sizes (i.e., 75) revealed that about 13% and 10% of notable based k-means and k-medoids, respectively, were grouped in the remaining clusters. The worst results were given by the large setting of cluster size (i.e., 100) using k-means with more than 34% of notable files scattered in the other 97 clusters while only less than 72% of notable files were collected in top three clusters, which is considered poor compared to FCM results.

Generally, the results of file list within this case appear to work well. Indeed, the large number of files led to improving the performance of the clustering approach. In addition, the timestamps of evidential files, in this case, were convergent. This could contribute to obtaining accurate results.

Table 5-5: Results of File list (Case 2)

Centres Generation	Cluster ID		Cluster Size											
			15		25		35		50		75		100	
			✓	×	✓	×	✓	×	✓	×	✓	×	✓	×
FCM	1	#	1271	5074	1271	3973	1116	3057	770	2684	723	1681	754	1685
		%	100	16.5	100	12.9	**87.9**	10	**60.6**	8.7	**56.9**	5.5	**59.4**	5.5
	2	#					155	916	501	1289	394	1376	517	2288
		%					**12.1**	3	**39.4**	4.2	**31**	5	**40.6**	7.5
	3	#									154	916		
		%									**12.1**	3		
	Rem.	#	0	25596	0	26697	0	26697	0	26697	0	26697	0	26886
		%	0	83.5	0	87.1	0	87	0	87.1	0	86.5	0	87
K-Means	1	#	1271	4350	1271	3980	1271	3963	1225	3778	831	2996	539	1837
		%	100	14.2	100	13	100	12.9	**86.5**	12.3	65.4	9.8	42.4	6
	2	#							46	2	210	58	166	34
		%							**13.5**	0.006	16.5	0.2	13.1	0.1
	3	#									98	8	131	44
		%									7.7	0.03	10.3	0.14
	Rem.	#	0	26320		26690	0	26707	0	26890	131	27608	434	28755
		%	0	85.8		87	0	87.1	0	87.69	10.4	89.97	34.2	93.76
K-Medoids	1	#	1271	4338	1271	3978	1271	3784	1112	3762	816	2984	571	2845
		%	100	14.1	100	13	100	12.3	**87.5**	12.2	64.2	7.7	45	9.3
	2	#							149	16	208	63	221	68
		%							**11.7**	0.05	16.4	0.2	17.4	0.2
	3	#							10	2	114	11	133	49
		%							**0.8**	0.006	9	0.04	10.5	0.15
	Rem.	#	0	26332	0	26692	0	26886	0	26890	132	27612	345	27708
		%	0	85.9	0	87	0	87.7	0	87.74	10.4	92.06	27.1	90.35

The results of the EXIF category are presented in Table 5-6. The clustering-based EXIF data achieved excellent results using FCM as 100% of notable files were grouped in the top two clusters across all setups of cluster size except 35-cluster size, where notables were grouped in the top three clusters. Concerning notable artefacts, the best result in this category was achieved using the 100-cluster size, where more than 97% of notables were obtained in a single cluster. Regarding benign files, they were relatively small within a range between 11.6% and 14.8% grouped within the top three clusters under all setups. Noticeably, regarding the density rate of benign data, the best performance of the approach was achieved by using the setups of 50 and 75 in which over 89% of noise data was scattered across other clusters.

The clustering-based k-means and k-medoids showed that all notable artefacts were only obtained in the top three clusters by using the 15-cluster size. In addition, the proportion of benign files using the same setup was low. In contrast, the remaining setups (i.e., 25, 35, 50, 75, and 100) proved challenging compared with the FCM results, as the proportion of notable artefacts decreased with increased cluster sizes. The results indicate more than 10% and 27% of relevant files were not obtained in the top three clusters using small and large setups, respectively. Nevertheless, the proportion of noise data, which were eliminated out of the top three clusters, was relatively high with more than 94%.

Table 5-6: Results of EXIF Data (Case 2)

Centres Generation	Cluster ID		Cluster Size											
			15		25		35		50		75		100	
			√	×	√	×	√	×	√	×	√	×	√	×
FCM	1	#	239	476	292	354	250	41	292	303	292	328	358	474
		%	65.12	12.71	**79.5**	9.4	**68.2**	1.1	**79.6**	8.1	**79.6**	8.8	**97.5**	12.6
	2	#	128	7	75	124	75	78	75	96	75	69	9	81
		%	34.88	0.19	20.5	3.3	**20.4**	2.1	20.4	2.5	20.4	1.8	2.5	2.1
	3	#					42	339						
		%					**11.4**	9						
	Rem.	#	0	3261	0	3266	0	3286	0	3345	0	3347	0	3189
		%	0	87.1	0	87.3	0	87.8	0	89.4	0	89.4	0	85.3
K-Means	1	#	203	213	128	8	128	8	125	105	122	4	122	4
		%	55.32	5.69	34.9	0.2	34.9	0.2	34.1	2.8	33.2	0.1	33.2	0.1
	2	#	125	165	125	105	125	105	122	8	75	80	75	80
		%	34.06	4.40	34	2.8	34	2.8	33.2	0.2	20.4	2.1	20.4	2.1
	3	#	39	105	75	155	75	80	75	80	69	125	65	105
		%	10.62	2.80	20.4	4.1	20.4	2.1	20.4	2.1	18.8	3.3	17.7	2.8
	Rem.	#	0	3261	39	3476	39	3551	45	3551	101	3535	105	3555
		%	0	87.11	10.7	92.9	10.7	94.9	12.3	94.9	27.6	94.5	28.7	95
K-Medoids	1	#	203	78	130	130	130	130	122	15	122	4	122	4
		%	55.32	2.08	35.4	3.5	35.4	3.5	33.2	0.4	33.2	0.1	33.2	0.1
	2	#	137	200	128	8	122	8	75	88	75	79	75	90
		%	37.33	5.34	34.9	0.2	33.2	0.2	20.4	2.3	20.4	2.1	20.4	2.4
	3	#	27	216	75	168	75	88	70	130	65	108	65	108
		%	7.35	5.70	20.4	4.5	20.4	2.3	19.1	3.5	17.7	2.9	17.7	2.9
	Rem.	#	0	3250	34	3438	40	3518	100	3511	105	3553	105	3542
		%	0	86.88	9.3	91.8	11	94	27.3	93.8	28.7	94.9	28.7	94.6

5.4.3 Case 3 Analysis

This case contains five main categories: file list, EXIF data, Facebook messages, internet data, and messaging. The results of file list are presented in Table 5-7. The results show that k-means and k-medoids gave the best result in grouping all notable artefacts within the top three clusters by using a cluster size of 15. The most notable artefacts were obtained in one cluster with 95.5% based k-means centres using the 15-cluster size. Under the same configuration, the amount of notable data being allocated to the top three clusters for FCM was smaller than its k-means and k-medoids counterpart with 93.3%.

There is a noticeable difference in the proportion of notable and benign data with increasing the setup of cluster size because the files within this category were collected from different devices and, thereby, different file systems. However, these results indicate the performance of the clustering approach was better using small setups in terms of related files. There is stability in obtaining notable artefacts in the first cluster of top three clusters where the proportion of notable artefacts is about 41% across all algorithms with all setups. On the other hand, the performance improved with increasing the configuration of cluster size in terms of isolating the noise data. Indeed, more than 99% of noise data across FCM, k-means, and k-medoids was eliminated using large setups.

Table 5-7: Results of File list (Case 3)

Centres Generation	Cluster ID		Cluster Size											
			15		25		35		50		75		100	
			✓	×	✓	×	✓	×	✓	×	✓	×	✓	×
FCM	1	#	38	3779	37	2151	37	244	37	217	37	207	37	209
		%	42.2	2.2	41.1	1.7	41.1	0.1	41.1	0.1	41.1	0.12	41.1	0.12
	2	#	37	2170	28	3186	36	643	33	572	24	443	29	365
		%	41.1	1.3	31.1	1.9	40	0.4	36.6	0.3	26.6	0.26	32.2	0.21
	3	#	9	2037	12	540	7	513	9	254	6	98	7	90
		%	10	1.2	13.3	0.3	7.7	0.3	10	0.15	6.6	0.05	7.7	0.05
	Rem.	#	6	162313	13	164422	10	168899	11	196256	23	169551	17	169635
		%	6.7	95.3	14.5	96.1	11.1	99.1	12.3	99.45	25.7	99.57	19	99.62
K-Means	1	#	86	25723	37	2182	37	759	41	765	37	381	37	276
		%	95.5	15.1	41.1	1.3	41.1	0.4	45.5	0.4	41.1	0.2	41.1	0.2
	2	#	4	221	33	3715	33	3651	37	752	33	528	33	575
		%	4.5	0.12	36.6	2.2	36.6	2.1	41.1	0.4	36.6	0.3	36.6	0.3
	3	#			14	2047	14	1852	4	174	7	365	7	361
		%			15.6	1.2	15.6	1.1	4.5	0.1	7.7	0.2	7.7	0.2
	Rem.	#	0	144355	6	162355	6	164037	8	168608	13	169025	13	169087
		%	0	84.73	6.7	95.3	6.7	96.4	9	99.1	14.6	99.3	14.6	99.3
K-Medoids	1	#	47	5809	40	3723	37	2180	41	720	37	595	37	216
		%	52.2	3.4	44.4	2.2	41.1	1.4	45.5	0.4	41.1	0.3	41.1	0.1
	2	#	39	19914	37	19591	31	859	37	716	25	448	27	521
		%	43.3	11.7	41.1	11.5	34.4	0.5	41.1	0.4	27.7	0.3	30	0.3
	3	#	4	818	9	2336	14	1942	4	3174	11	224	9	312
		%	4.5	0.5	10	1.4	15.5	1.2	4.5	1.9	12.2	0.1	10	0.2
	Rem.	#	0	143758	4	144629	8	165318	8	165689	17	169032	17	196250
		%	0	84.4	4.4	84.9	9	96.9	9	97.3	19	99.3	18.9	99.4

The results of the EXIF category are presented in Table 5-8. These results revealed that all notable artefacts were founded within a single cluster-based on FCM, k-means, and k-medoids using all configurations of cluster sizes. This phenomenon could happen owing to the small number of evidential pictures. Additionally, these pictures were taken in one location where GPS data was relatively similar.

In contrast, the proportion of benign data being gathered within the single cluster decreased slightly as the cluster size increased across the three algorithms. Noticeably, the performance of clustering-based FCM centres using the 15-cluster size outperformed k-means and k-medoids in terms of separating the benign data, where only 17.8% of benign data was obtained in the first cluster while 22% of benign data was found in the same cluster using the centres of other algorithms.

Within the same case, the results of the clustering-based internet data category were illustrated in Table 5-9. This category reflected challenging results because all artefacts (both notables and benign) were grouped in a single cluster across all clustering setups based on FCM, k-means, and k-medoids. The reason for this issue could be because of the small number of total internet actions (78 actions in total) from one browser. Additionally, some of clustering setups (i.e., 75, 100) were close or larger than the number of artefacts as the clustering process requires a large number of files to work properly.

Table 5-8: Results of EXIF (Case 3)

Centres Generation	Cluster ID		Cluster Size											
			15		25		35		50		75		100	
			√	×	√	×	√	×	√	×	√	×	√	×
FCM	1	#	62	238	62	208	62	204	62	194	62	194	62	194
		%	100	17.6	100	15.4	100	15.1	100	14.4	100	14.4	100	14.4
	Rem.	#	0	1109	0	1139	0	1143	0	1153	0	1153	0	1153
		%	0	82.4	0	84.6	0	84.9	0	85.6	0	85.6	0	85.6
K-Means	1	#	62	296	62	206	62	204	62	194	62	194	62	194
		%	100	22	100	15.3	100	15.1	100	14.4	100	14.4	100	14.4
	Rem.	#	0	1051	0	1141	0	1143	0	1153	0	1153	0	1153
		%	0	78	0	84.7	0	84.9	0	85.6	0	85.6	0	85.6
K-Medoids	1	#	62	296	62	206	62	194	62	194	62	194	62	194
		%	100	22	100	15.3	100	14.4	100	14.4	100	14.4	100	14.4
	Rem.	#	0	1051	0	1141	0	1153	0	1153	0	1153	0	1153
		%	0	78	0	84.7	0	85.6	0	85.6	0	85.6	0	85.6

Table 5-9: Results of Internet (Case 3)

Centres Generation	Cluster ID		Cluster Size											
			15		25		35		50		75		100	
			✓	×	✓	×	✓	×	✓	×	✓	×	✓	×
FCM	1	#	23	55	23	55	23	55	23	55	23	55	23	55
		%	100	100	100	100	100	100	100	100	100	100	100	100
	Rem.	#	0	0	0	0	0	0	0	0	0	0	0	0
		%	0	0	0	0	0	0	0	0	0	0	0	0
K-Means	1	#	23	55	23	55	23	55	23	55	23	55	23	55
		%	100	100	100	100	100	100	100	100	100	100	100	100
	Rem.	#	0	0	0	0	0	0	0	0	0	0	0	0
		%	0	0	0	0	0	0	0	0	0	0	0	0
K-Medoids	1	#	23	55	23	55	23	55	23	55	23	55	23	55
		%	100	100	100	100	100	100	100	100	100	100	100	100
	Rem.	#	0	0	0	0	0	0	0	0	0	0	0	0
		%	0	0	0	0	0	0	0	0	0	0	0	0

The results of the messaging category are illustrated in Table 5-10. This category has the most critical results of this case as the high proportion of benign data were found within the top three clusters using the centres of k-means and k-medoids. For simplicity, the proportion of benign data was more than 72% for k-means and k-medoids data using the small setups of cluster size (i.e.,15, 25, 35). In contrast, the proportion of benign data within the top three clusters across all clustering setups using FCM was smaller outperformed the other algorithms, where the amount of benign data was relatively persistent with 35% on average.

With regard to the proportion of notable artefacts, the clustering-based FCM shows that there is stability in the results across all cluster sizes with a proportion between 77.7-83.3%. In contrast, the results-based k-means and k-medoids illustrate there is a similarity in the findings between them where the small setups of cluster sizes (i.e., 15, 25, and 35) gave better results and outperformed the large setups (i.e., 50, 75, and 100).

The results of the Facebook messenger category are presented in Table 5-11. The clustering-based FCM revealed that a large number of both evidential and noise artefacts were found in top three clusters. The cluster sizes in this category using FCM showed a similarity in the results in terms of notable and benign artefacts between small and large cluster sizes. To clarify, the similarity was between 15 and 100-cluster sizes and between 25 and 75-cluster sizes. In comparison, the clustering-based on k-means and k-medoids were more accurate compared to FCM. A large proportion of notable artefacts was obtained in the top three clusters using small setups. By increasing the cluster size, the proportion of benign data was significantly decreased with a slight decrease in the proportion of notable artefacts.

Table 5-10: Results of Messaging data (Case 3)

Centres Generation	Cluster ID		Cluster Size											
			15		25		35		50		75		100	
			✓	×	✓	×	✓	×	✓	×	✓	×	✓	×
FCM	1	#	47	34	47	34	47	34	47	34	47	34	47	34
		%	**52.2**	13.3	52.2	13.3	52.2	13.3	52.2	13.3	52.2	13.3	52.2	13.3
	2	#	19	30	14	28	14	29	15	31	13	25	14	29
		%	**21.1**	11.7	15.5	10.9	15.5	11.3	16.6	12.1	14.4	9.8	15.5	11.3
	3	#	9	24	10	29	9	24	9	24	9	24	9	24
		%	**10**	9.4	11.1	11.3	10	9.4	9.4	9.4	9.4	9.4	10	9.4
	Rem.	#	15	168	19	165	22	169	19	133	21	173	20	169
		%	16.7	65.6	21.2	64.5	22.3	66	21.2	65.2	24	67.5	22.3	66
K-Means	1	#	47	34	47	34	47	34	21	10	12	2	8	0
		%	**52.2**	13.3	52.2	13.3	52.2	13.3	23.3	3.9	13.3	0.8	8.9	0
	2	#	28	152	20	30	20	30	19	14	9	5	7	1
		%	**31.1**	59.4	22.2	11.7	22.2	11.7	21.1	5.4	10	1.9	7.8	0.4
	3	#	11	13	8	122	8	122	9	6	6	2	5	0
		%	**12.2**	5.1	8.9	47.6	8.9	47.6	10	2.3	6.6	0.8	5.5	0
	Rem.	#	5	56	15	70	15	70	41	192	65	247	70	255
		%	4.5	22.2	16.7	27.4	16.7	27.4	45.6	88.4	70.1	96.5	77.8	99.6
K-Medoids	1	#	47	34	47	34	47	34	21	10	12	2	7	2
		%	**52.2**	13.3	52.2	13.3	52.2	13.3	23.3	3.9	13.3	0.8	7.8	0.8
	2	#	28	152	28	152	20	30	13	7	9	2	5	0
		%	**31.1**	59.4	31.1	59.4	22.2	11.7	14.4	2.7	10	0.8	5.5	0
	3	#	11	13	4	0	8	122	10	8	5	0	5	0
		%	**12.2**	5.1	4.4	0	8.9	47.6	11.1	3.1	5.5	0	5.5	0
	Rem.	#	5	56	11	70	15	70	46	231	64	252	73	254
		%	4.5	22.2	12.3	27.4	16.7	27.4	51.2	90.3	71.2	98.6	81.2	99.2

Table 5-11: Results of Facebook Messenger (Case 3)

Centres Generation	Cluster ID		Cluster Size											
			15		25		35		50		75		100	
			✓	×	✓	×	✓	×	✓	×	✓	×	✓	×
FCM	1	#	223	479	223	479	156	316	223	479	156	316	223	479
		%	**92.9**	27.5	**92.9**	27.5	65.1	18.1	**92.9**	27.5	65.1	18.1	92.9	27.5
	2	#	16	858	16	492	67	163	16	858	67	163	16	858
		%	**6.7**	49.3	**6.7**	28.3	27.9	9.3	**6.7**	49.3	27.9	9.3	6.6	49.3
	3	#	1	88	1	79	16	858	1	74	16	529	1	83
		%	**0.4**	5	**0.4**	4.5	6.6	49.3	**0.4**	4.2	6.6	30.4	0.4	4.7
	Rem.	#	0	315	0	690	1	403	0	329	1	732	0	302
		%	0	18.2	0	39.7	0.4	23.3	0	19	0.4	42.2	0	18.5
K-Means	1	#	106	29	57	6	44	4	39	6	36	2	25	0
		%	44.2	1.6	23.7	2.5	2.5	0.2	16.3	2.5	15	0.1	10.4	0
	2	#	84	48	54	12	43	5	37	2	36	4	25	2
		%	35	2.7	22.5	5	2.5	0.3	15.4	0.1	15	0.2	10.4	0.1
	3	#	33	264	47	16	41	7	37	4	36	4	23	4
		%	13.7	15.2	19.6	6.6	2.4	0.4	15.4	0.2	15	0.2	9.6	0.2
	Rem.	#	17	1399	82	1706	112	1727	127	1728	132	1730	167	1734
		%	7.1	80.5	34.2	85.9	92.6	99.1	52.9	97.2	55	99.5	69.6	99.7
K-Medoids	1	#	150	39	66	13	74	6	44	4	36	4	27	0
		%	62.5	2.2	27.5	0.7	30.8	0.3	2.5	0.2	15	0.2	11.3	0
	2	#	73	302	65	6	54	9	42	8	33	0	24	1
		%	30.4	17.3	27.1	0.3	22.5	0.5	17.5	0.4	13.8	0	10	0.05
	3	#	16	480	31	52	52	24	41	5	33	2	22	0
		%	6.7	27.6	12.9	3	21.6	1.4	17.1	0.3	13.8	0.1	9.2	0
	Rem.	#	1	919	78	1669	60	1701	113	1723	138	1734	167	1739
		%	0.4	52.9	32.5	96	25.1	97.8	62.9	99.1	57.4	99.7	69.5	99.95

5.4.4 Case 4 Analysis

This case includes two main categories: file list and EXIF data. The results of file list are presented in Table 5-12. Although the clustering based on the file list category showed the notable artefacts across FCM, k-means, and k-medoids under all configurations of cluster size were not all grouped in the top three clusters, there was a high proportion of notable files obtained within the first top cluster. The best performance was obtained based on the top first cluster under the setup of a 35-cluster size using FCM where more than 86% of notable versus 0.2% of benign data were concentrated in a single cluster. In addition, under the setups of 35 and 50-cluster sizes using the FCM, more than 92% of benign data with only less than 4% of notable files were clustered out of the top three clusters.

Regarding the k-means and k-medoids, the results revealed there was a similarity in the proportion of evidential artefacts containing within the top three clusters. For clarity, the proportion of notable artefacts was slightly decreased with increasing the configuration of cluster size in the top three clusters where more than 94% of notable files was included using the small setups. In contrary, a significant amount of benign data was eliminated with increasing the configuration of cluster size where more than 97% of noise data was excluded using the large setups.

Table 5-12: Results of File List (Case 4)

Centres Generation	Cluster ID		Cluster Size											
			15		25		35		50		75		100	
			✓	×	✓	×	✓	×	✓	×	✓	×	✓	×
FCM	1	#	207	4325	207	3072	200	507	207	3061	205	2668	207	2965
		%	**89.2**	1.7	**89.2**	1.2	86.2	0.2	89.2	1.2	88.4	1.1	89.2	1.2
	2	#	14	18259	14	16828	16	15495	12	15033	9	16042	12	9302
		%	**6**	7.3	**6**	6.8	6.9	6.2	5.2	6	3.9	6.5	5.2	3.7
	3	#	7	23750	7	23560	5	2062	4	548	7	14668	7	16525
		%	**3**	9.5	**3**	9.5	2.1	0.8	1.7	0.2	3	5.9	3	6.6
	Rem.	#	4	202366	4	205240	11	230636	9	230058	11	215322	6	219908
		%	1.8	81.5	1.8	82.5	4.8	92.8	3.9	92.6	4.8	86.5	2.6	88.5
K-Means	1	#	207	4325	207	4325	207	4325	207	4325	207	2899	203	2431
		%	89.2	1.7	89.2	1.7	89.2	1.7	89.2	1.7	89.2	1.7	87.5	0.9
	2	#	9	14908	8	13710	9	2701	9	1746	9	1654	9	1655
		%	3.9	6	3.4	5.5	3.9	1.1	3.9	0.7	3.9	0.7	3.9	0.7
	3	#	7	2026	7	1580	4	1279	4	1118	4	426	4	635
		%	3	0.8	3	0.6	1.7	0.5	1.7	0.4	1.7	0.2	1.7	0.3
	Rem.	#	9	227441	10	229085	12	240395	12	241511	12	243721	16	243979
		%	3.9	91.5	4.4	92.2	5.2	96.7	5.2	97.2	5.2	97.4	6.9	98.1
K-Medoids	1	#	207	4325	207	4325	207	4260	207	4260	205	2743	207	2899
		%	89.2	1.7	89.2	1.7	89.2	1.7	89.2	1.7	88.4	1.1	89.2	1.2
	2	#	9	14984	9	8504	9	5345	9	1764	9	1752	6	380
		%	3.9	6	3.9	3.4	3.9	2.1	3.9	0.7	3.9	0.7	2.6	0.2
	3	#	7	1590	7	2014	4	1177	4	1117	4	1049	4	206
		%	3	0.6	3	0.8	1.7	0.5	1.7	0.4	1.7	0.4	1.7	0.1
	Rem.	#	9	227801	9	233857	12	237918	12	241559	14	243156	15	245215
		%	3.9	91.7	3.9	94.1	5.2	95.7	5.2	97.2	6	97.8	6.5	98.5

The results of EXIF data are illustrated in Table 5-13. The clustering based on the EXIF data alone proved successful, where all notable artefacts were obtained within the top three clusters using the centres of all algorithms and small setups of cluster sizes. The performance of the clustering approach based on the FCM centres was better than the clustering based on k-means and k-medoids in grouping the evidential files within the top three clusters with a small number of benign files. For clarity, the clustering based on FCM centres using small setups of cluster sizes (i.e., 15, 25, and 35) showed that 100% of notable files with less than 11% of benign data were clustered within the top three clusters. In addition, more than 99% of evidential artefacts with a tiny proportion of noise data was obtained in a single cluster across the first three setups.

The larger setup of cluster size (i.e., 100) based on FCM provided the same results for the small setup (i.e., 15) within k-means and k-medoids, in terms of notable and benign data. However, small setups of cluster sizes using k-means and k-medoids were more accurate than the large setups in which the most notable files were found in a single cluster with a tiny amount of noise files. Moreover, the larger setup failed to group the notable files in the top three clusters, where less than a half of the notable artefacts were obtained within the other clusters.

Table 5-13: Results of EXIF (Case 4)

Centres Generation	Cluster ID		Cluster Size											
			15		25		35		50		75		100	
			✓	×	✓	×	✓	×	✓	×	✓	×	✓	×
FCM	1	#	244	101	244	101	244	101	144	45	142	49	244	655
		%	**99.2**	1.8	**99.2**	1.8	**99.2**	1.8	58.5	0.8	57.7	0.9	**99.2**	11.6
	2	#	1	272	1	271	1	123	100	4	51	0	2	545
		%	**0.4**	4.8	**0.4**	4.8	**0.4**	2.2	40.7	0.1	20.7	0	**0.8**	9.7
	3	#	1	274	1	160	1	271	1	2	50	0		
		%	**0.4**	4.9	**0.4**	2.8	**0.4**	4.8	0.4	0.05	20.3	0		
	Rem.	#	0	4993	0	5108	0	5145	1	5589	3	5591	0	4440
		%	0	88.5	0	90.6	0	91.2	0.4	99.05	1.3	99.1	0	78.7
K-Means	1	#	244	107	244	49	142	49	142	49	67	0	45	0
		%	**99.2**	1.9	**99.2**	0.9	**57.7**	0.9	57.7	0.9	27.2	0	18.3	0
	2	#	2	1093	1	271	102	0	102	0	48	8	40	37
		%	**0.8**	19.4	**0.4**	4.8	**41.5**	0	41.5	0	19.5	0.1	16.3	0.7
	3	#			1	274	2	545	1	271	48	37	33	0
		%			**0.4**	4.9	**0.8**	9.7	0.4	4.8	19.5	0.7	13.4	0
	Rem.	#	0	4440	0	5046	0	5046	1	5320	83	5595	128	5603
		%	0	78.7	0	89.4	0	89.4	0.4	49.3	33.8	99.2	52	99.3
K-Medoids	1	#	244	107	244	49	244	49	244	49	102	0	43	0
		%	**99.2**	1.9	**99.2**	0.9	**99.2**	0.9	**99.2**	0.9	41.5	0	17.5	0
	2	#	2	1093	2	545	2	545	1	271	94	40	40	37
		%	**0.8**	19.4	**0.8**	9.7	**0.8**	9.7	**0.4**	4.8	38.2	0.7	16.3	0.7
	3	#							1	274	48	9	34	0
		%							**0.4**	4.9	19.5	0.2	13.8	0
	Rem.	#	0	4440	0	5046	0	5046	0	5046	2	5591	129	5603
		%	0	78.7	0	89.4	0	89.4	0	89.4	0.8	99.1	52.4	99.3

5.5 Discussion

From the aforementioned results, the proposed approach of clustering has the ability to group the evidential artefacts within the top three clusters. Therefore, the approach can correlate the related artefacts in the same category. Indeed, each case contains more than one evidential source with various categories. These categories were classified into file system and applications. Within each case, there are similar categories, such as file list, messaging, and internet data. The process of merging the similar categories has been successfully achieved without any effect on the clustering process.

The clustering based on the file list showed the best results across the four cases with a high proportion of notables being grouped with the top three clusters using FCM, k-means, and k-medoids with a relatively small amount of benign data being included. This was because of a large number of files contained in these categories, as the clustering works well with large volumes of data. However, as illustrated in Figure 5-5, the results-based FCM centres within Case 1, Case 2 and Case 4 were relatively similar where most notable artefacts were grouped within rank three clusters using all setups of cluster sizes. In contrast, as illustrated in Figures 5-6 and 5-7, the clustering-based k-means and k-medoids presented better results compared with the FCM in terms of small setups (i.e., 15, 25, 35). For clarity, more than 94% of notable files was obtained within the top three clusters for all cases. However, the large setups of cluster sizes based on FCM (i.e., 50, 75, 100) presented good results that outperformed the k-means and k-medoids, except in Case 3. The Case 3-based FCM proved the most challenging results in terms of grouping the evidential files within the top three clusters where more than 20% of notable artefacts were clustered out of top three clusters. Nevertheless, the smallest proportion of benign data was found in Case 3 with less than 1% across all the setups based on the FCM.

In Case 2, the clustering FCM illustrated there was no influence apparently in the results in terms of notable and benign data when changing the cluster size using the FCM algorithm. Moreover, the proportion of benign data was relatively constant and small. Similarly, the clustering-based k-means and k-medoids using the setups of 15, 25, 35, and 50- cluster sizes revealed that 100% of notable files was obtained in the top three clusters with less than 12% of benign data.

Generally, whenever the size of the cluster configuration increases, the proportion of notable and benign data decreases. This means the small cluster configuration comparatively contained a large number of both notable and benign data while large cluster configuration comparatively contained less proportion of both notable and noise data.

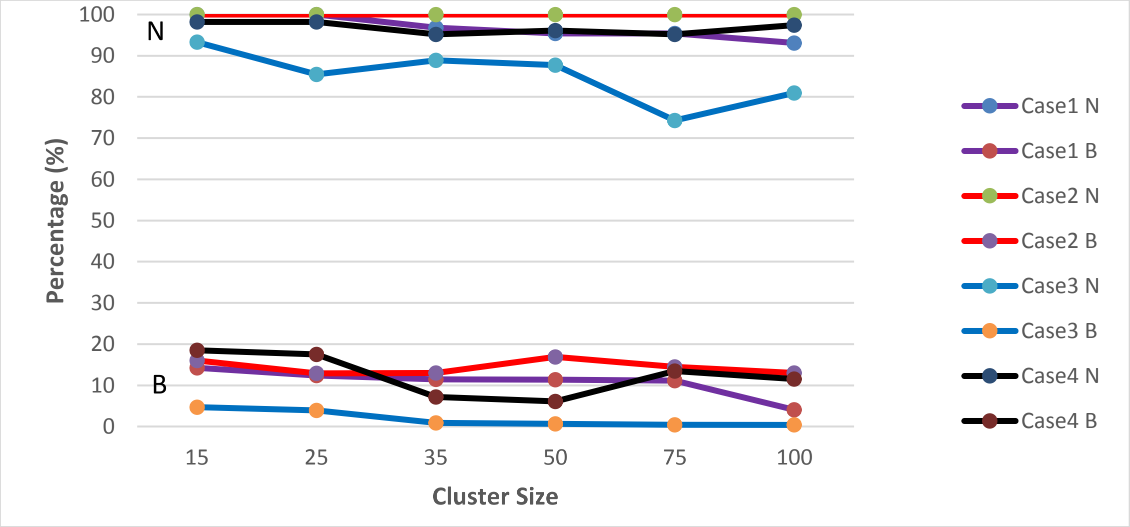


Figure 5-5: File List results within top three clusters across the four cases using FCM centres (N: Notable, B: Benign)

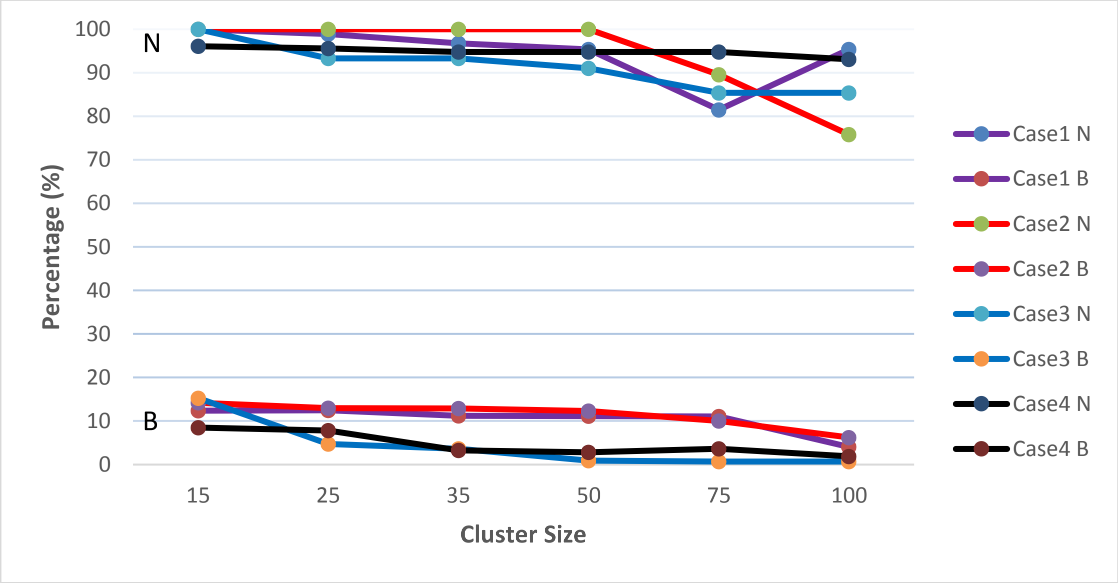


Figure 5-6: File List results within top three clusters across the four cases using K-Means centres (N: Notable, B: Benign)

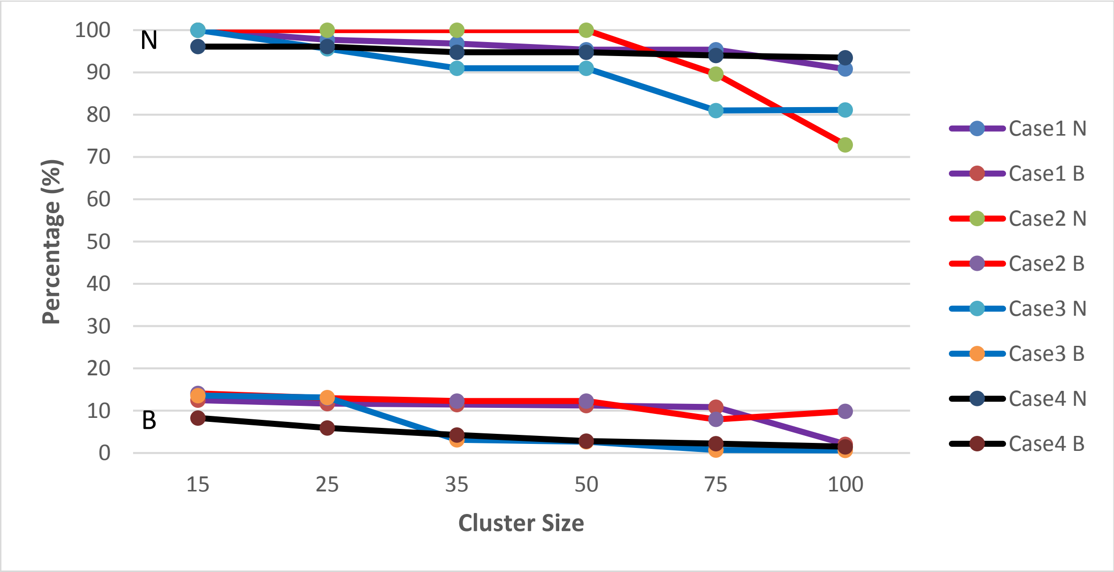


Figure 5-7:File List results within top three clusters across the four cases using K-Medoids centres (N: Notable, B: Benign)

Concerning the clustering-based application categories, it was revealed that the performance of grouping the evidential artefacts with a minimum proportion of benign artefacts were less efficient compared with clustering based on the file list. This could be because of the small number of files within the applications' categories. However, the clustering-based EXIF category presented the best results among applications categories in terms of grouping the notable artefacts in Cases 2 and 3 using FCM and k-means. However, it is notable that the proportion of benign files using k-means clusters was less than the proportion of benign files using FCM. The results of the email category within Case 1 and the internet category within Case 3 showed the worst results because all notable and benign files were grouped in one cluster. This was as a result of the small number of files that provided for the cluster procedure (e.g., only 19 files in the email category). The messaging and Facebook messenger categories only appeared in Case 3. The results of messaging show the large setups of cluster sizes can offer better performance compared to small setups. This is because the number of evidential artefacts is relatively small thereby causing a burden on the clustering algorithms to identify them within only three-clusters. Therefore, the large setups can be more flexible to only isolate the evidential files. In contrast, the category of Facebook messenger contains a large number of artefacts where the performance of clustering algorithms using the small setups of cluster sizes was noticeably better in determining the evidential artefacts in three clusters.

5.6 Conclusion

This chapter examined the possibility of using clustering algorithms in digital forensic analysis. The proposed approach of clustering works on the merged datasets, which come from various resources within a single case. The experimental results proved that the evidence can be correlated within a dataset and the evidential artefacts can be grouped in the rank-three cluster. The results of identifying notable artefacts within rank-three cluster revealed that clustering based file systems was more accurate than clustering-based applications. The results also illustrated that there is a slight difference among FCM, k-means, and k-medoids algorithms but the FCM showed stability in the results across various configurations of cluster size.

6 Automated Identification of Evidential artefacts

This chapter proposes an automated approach to identifying the evidential artefacts obtained by the clustering approach. This approach has two steps: identify the first cluster based on information that was obtained during the preliminary investigation of case; and identify the sub-clusters using the timeline analysis and association-matching of the artefacts within the first cluster. A series of experiments based on the fourth cases was achieved to validate the proposed approach.

6.1 Introduction

The previous chapter presented encouraging results of grouping the evidential artefacts in a small number of clusters. In practice, however, these clusters need to be identified. Therefore, there is an essential need to develop an algorithm to obtain the evidence in an automated way. The algorithm should initially identify the first cluster, which contains a large number of notable artefacts. These artefacts will then be used to identify the sub clusters across all clusters of the file list and applications categories within a case.

To identify the first cluster, various methods can be used to determine the cluster containing related files. These include timeline analysis, as well as information obtained from the suspects themselves, such as names, nicknames, and emails, type of crimes, and many more. This information makes the algorithm concentrate on certain artefacts in a certain category, thereby identifying the cluster containing a large number of these artefacts.

Benefiting from the artefacts' features offered by the first cluster timestamps, search of associated files—would enable the approach in identifying the next cluster that should be analysed. The approach introduces an intelligent process based on the timeline analysis and association search in finding the related files of the first cluster. As a result, the approach can select the cluster with a large proportion of interest.

6.2 Problem Identification

The digital forensic process transforms the suspected media into data, data into information, and information into evidential artefacts. An adequate process of digital forensics should rely on sequential steps to identify evidence. Reliable and valid steps in the forensic process to identify evidence in digital investigations are becoming essential for law enforcement agencies worldwide. These steps must be algorithmically robust and provide assurance and quality with standardisation to ensure all probative information is recovered (Bulbul et al, 2013). They must also be legally defensible ensuring nothing in the original evidence was altered and that no data was added to or deleted from the original. Having established how to acquire data, extracting metadata categories, merging the similar categories, and obtaining clusters with similar artefacts across different categories in chapters 5 and 6, there is a need to find these clusters to complete the evidence analysis phase.

However, it is worth exploring current approaches in profiling crimes to obtain a comprehensive view in the current state of the art. Crime profiling is a method used to identify the characteristics of criminals by using previously gathered information from committed offences or offenders (Horsman et al, 2011). By profiling each case, various unique features can be produced, thereby helping to examine and find evidence within similar cases. The digital forensic field has brought new aspects for forensic science in a method of identifying the evidence, but various fundamentals are based on the same goals of traditional investigation, such as auditability and replicability of findings (Palmer, 2001). Therefore, the findings of digital investigation are to determine what has occurred, where it occurred, when it occurred, how it occurred, why it might have occurred, and hopefully who is responsible. The investigators might use criminal profiling to answer these questions, thereby reducing the number of possibilities of determining the evidential artefacts. In addition, as more digital cases become profiled, more evidential artefacts will easily be identified.

Law enforcement agencies tried to collect the databases containing detailed information about major criminal acts. These databases can be used to produce criminal behaviours that can be used to support the investigation in finding the evidence. In this context, Baumgartner et al. (2008) proposed a new Bayesian network (BN) approach for criminal profiling using the database of cleared homicides. This was achieved by selecting the most important relationships between the related features of criminal profiling. These features were used to predict the unknown offender based on a number of features from the crime scene, as presented in Table 6-1. The BN was used to gather the characteristics of an unknown criminal from the evidence of the crime scene. These characteristics can help investigators decrease the number of suspects in unsolved cases. Their experiment showed over 80% of criminal characteristics was correctly predicted based on new single-victim homicides. However, this approach requires constantly updating these features because various incidents occur every day with characteristics that might not be included in available collected databases.

Table 6-1: Description of selected features of offender and crime scene (Baumgartner et al., 2008)

Feature	Description
Y4	Prior record of property damage
Y5	Prior record of disorderly conduct
Y6	Previous imprisonment or youth detention
Y9	History of sex crime
Y10	Record of armed services
E11	Victim sustained stabbing wounds
E12	Blunt instrument used on victim
E13	Offender used own body as weapon (e.g. strangulation)
E14	Victim was shot
E15	Victim sustained wounds to head (excluding face and neck)
E16	Victim sustained wounds to face (ears forward)
E31	Victim was sexually assaulted
E33	Arson to crime scene or body
E34	Body was found in water

The Home Office categorised the digital crimes based on criminal activities into two types: cyber-dependent crimes and cyber-enabled crime (McGuire and Dowling, 2013). Cyber-dependent crimes refers to crimes that can only be committed using electronic devices (i.e., computers, mobiles, and networks), such as the spreading of viruses and malicious applications, hacking of computers, and network resources. Cyber-enabled crimes refer to traditional crimes that can be committed assisted by the electronic devices, such as sexual offending against children (i.e., creation and/or distribution of sexual imagery). Noticeably, mapping digital crimes to their categories helps investigators narrow down the investigation to save and effort the time by isolating the files that are not related and focusing on the related files.

In the same context, the US Department of Justice reported that certain types of electronic crimes can occur based on particular types of files (Holder et al., 2009). For instance, phishing scams are attempts to obtain personal information, such as bank account numbers, passwords, and credit card numbers, and using calling and messaging applications. While the child abuse cases contain image-based files, the illegal downloading might include browsing history, multimedia, and documents.

Based on a report by the US Department of Justice (Holder et al., 2009), Al Fahdi (2016) proposed the Automated Evidence Profiler (AEP) approach to identify the evidential artefacts, as illustrated in Figure 6-1. This approach used prior work in criminal profiling behaviour, which was reported by the US Department of Justice, as explained above. The approach tried to identify the cluster containing a large number of artefacts with a particular type of file related to the criminal activity in file system clusters. To find other artefacts within other clusters, the approach used the timeline analysis of files within a first cluster to find the related files in other clusters. However, the AEP approach does not work with all cases because it depends on some prior work completed in profiling criminal behaviour to identify the first cluster. There might be new criminal behaviour cases that are not yet analysed. In addition, only using the profiled criminal behaviour to identify the first cluster might lead to the wrong clusters, as there may be many clusters containing similar types of files, thereby obtaining benign files instead of evidential files.

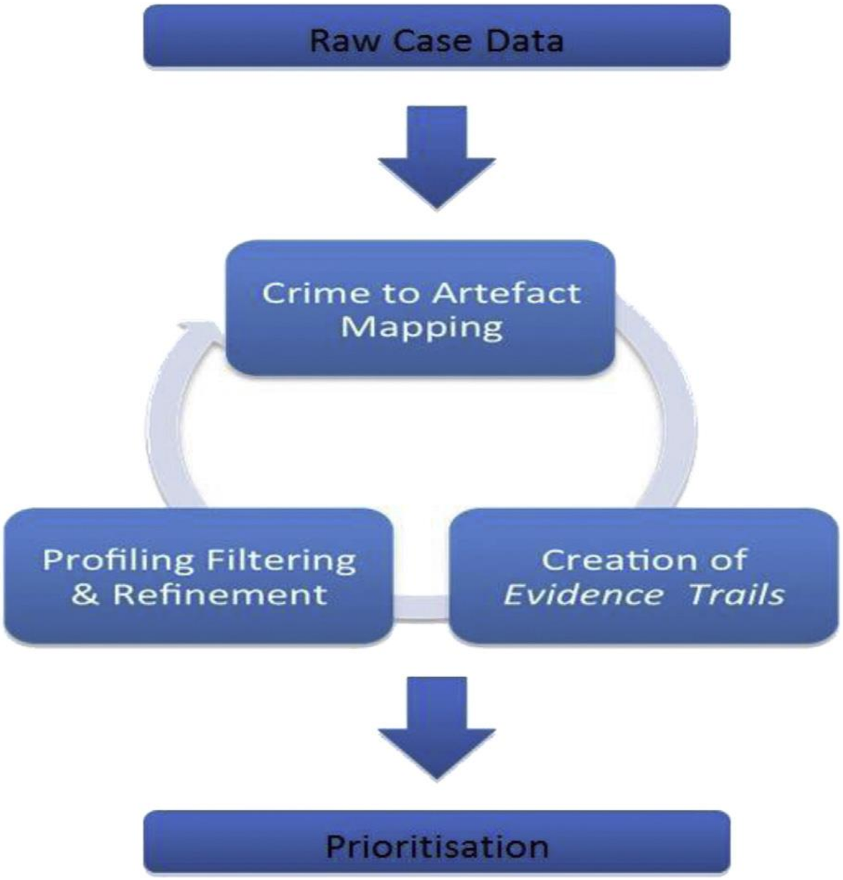

Raw Case Data
Crime to Artefact Mapping
Profiling Filtering & Refinement
Creation of *Evidence Trails*
Prioritisation

Figure 6-1: Automated Evidence Profiler Process (Al Fahdi, 2016)

6.3 Automated Identification of Evidence (AIE)

Whilst the clustering process can be appreciated as a solution to effectively group the evidential artefacts within a small number of clusters, it does not have the capability to only identify the clusters containing the evidential artefacts automatically. Therefore, there is an essential need to use other domains, such as a criminal profiling of previous cases and preliminary information of suspected cases as features to map them in identifying the clusters including the evidence. This mapping requires an “intelligent system” to develop a holistic evidence locator and collector. In this chapter, an intelligent approach was developed to identify the related artefacts in an automated way. This approach tries to answer the two main research questions:

- How is the first cluster located for analysis?
- How are sub-clusters across a given case identified?

The proposed approach tries to answer these questions and maximise the number of evidential artefacts by minimising the number of benign files, as illustrated in Figure 6-2. This approach combines various domains to obtain the evidence, such as prior work in criminal profiling based on the crime type to distinguish the file types within the case. Additionally, using various keyword lists related to suspected case can lead to getting the evidential artefacts and, thereby, identifying the clusters containing the evidence.

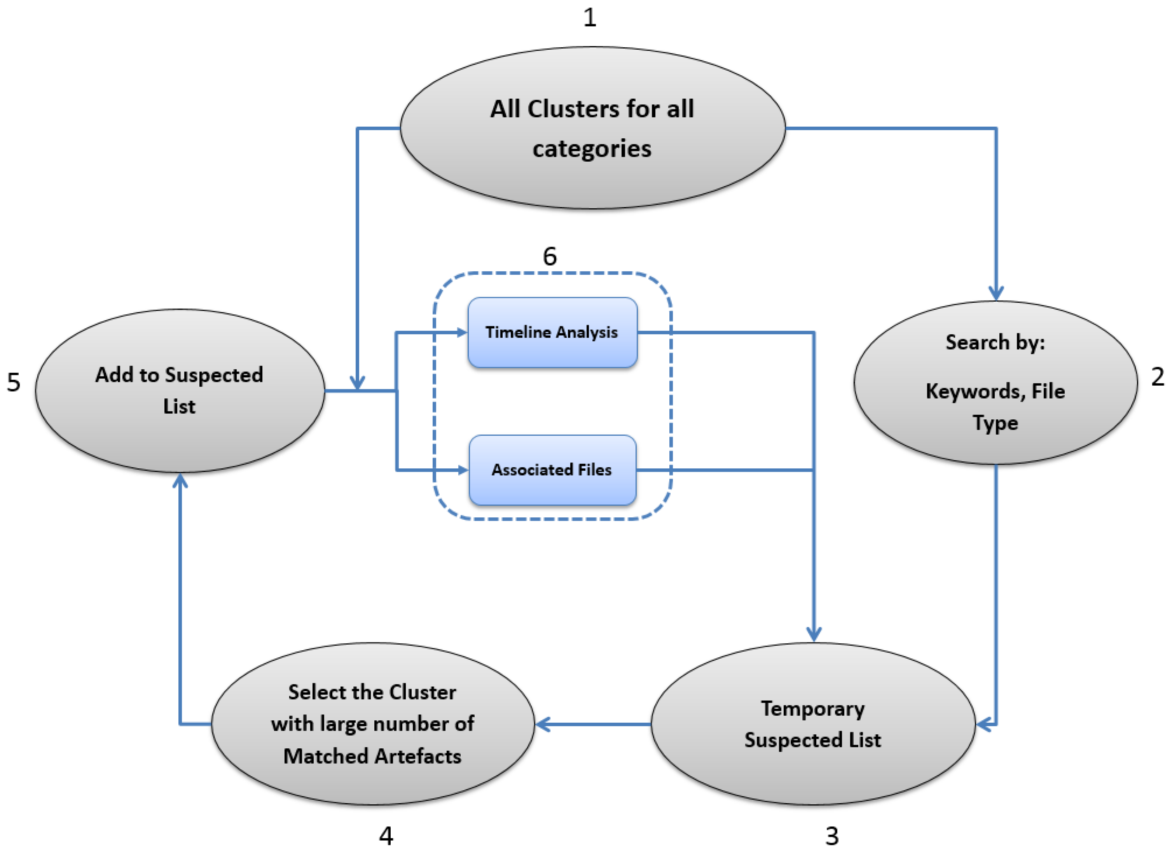


Figure 6-2: Evidence Identification Process

For clarity, the solution to the first question is based on prior work in behaviour profiling, as presented in the previous section with using some basic information about the suspected person. This information can be obtained from the suspects themselves or preliminary reports and documents of the case, such as suspected names, their nicknames, or locations. Using various methods in identifying the first clusters can be beneficial to avoid the limitations of only having criminal profile activity. Additionally, if one of the methods fails to obtain evidential artefacts, there are alternative methods. For instance, if there is a lack of information of criminal activity, the keyword list can be used to maximise the number of evidential files. Algorithm 6-1 shows the steps on how to identify the first cluster.

Algorithm 6-1: First Cluster Identification

```
Input: Clusters, keywords list, criminal profile.
Output: First cluster to analyse.

Process
Step 1: Read the first cluster.
Step 2: If the counter exceeds the end go to step 7.
                    Else, go to steps 3, and 4.
Step 3: Search in the cluster about a file type that is matched to the criminal activity. If there is a match go to step 5.
                    Else go to step 6.
Step 4: Search in the cluster about the keywords list, if there is a match go to step 5.
                    Else go to step 6.
Step 5: Save the finding in temporarily buffer. Go to step 6
Step 6: Read the next cluster. Go to step 2.
Step 7: Find the cluster with a large number of files.
Step 8: End.
```

To find the solution for the second question and to find the evidential artefacts within incorporated results of many clusters from various metadata categories, the proposed approach uses two methods that can be applied on the artefacts within the first cluster to identify the related files in other clusters: timeline analysis algorithm and association algorithm.

The timeline analysis is a process used to link potential files of interest within digital crimes by creating a chronological record of events. It can also aid digital forensic investigators to obtain an overview of the sequence of activities that indicate the crime and then gain a full story of the event subject to the investigation. As a result, high-level events can direct low-level events within the whole case. The process timeline analysis can lead to other files or information that were not suspected.

Once the first cluster is identified correctly, the timeline analysis works to identify the second cluster for analysis. Therefore, the timestamp of all artefacts within the first cluster will be subjected to the timeline analysis process. The timeline analysis process is executed on the entire artefacts in the clusters across all metadata categories. For instance, an image for a suspect was previewed at 13:05:00. After one minute, the Photoshop application was executed to modify this image. Subsequently, an internet browser was used to open an Outlook account to send the modified image to someone. In this scenario, files and applications were used to achieve an activity, as presented in Figure 6-3.

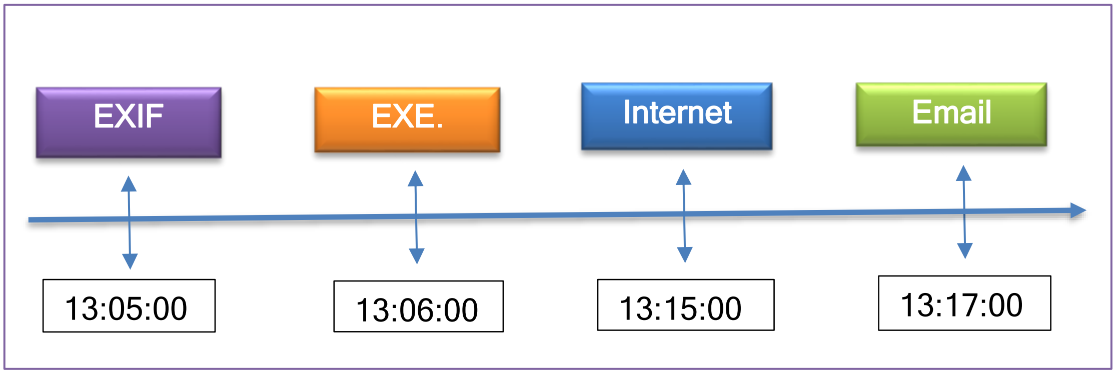


Figure 6-3: Timeline Analysis

Algorithm 6-2 illustrates the process of timeline analysis that will be applied on the first cluster to identify the files of interest, thereby leading to the next cluster to analyse.

Algorithm 6-2: Timeline Analysis

```
Input: First cluster, other clusters, time window.
Output: Suspected files.
------------------------------------------------------------
Process
Step 1: Read the timestamp of the first file within the first cluster.
Step 2: If the counter exceeds the end, go to step 6.
            Else, go to step 3.
Step 3: Search in other all clusters about the matched timestamps before and after the period of the time window. If there is a match, go to step 4.
            Else go to step 5.
Step 4: Save the finding in temporarily buffer, go to step 5.
Step 5: Read the timestamp of the next file within the first cluster. Go to step 2.
Step 6: Arrange the findings based on the large proportion of clusters that appear.
Step 7: End.
```

The second method of identifying the related clusters is by the matched artefacts based on their metadata features, having established that the first cluster mostly contains the large number of interest files. In addition, this cluster often comes from the file system category, which includes various files and artefacts belonging to several applications. For instance, a particular image contains many features, such as file name, file size, file type, and many more. This image might be sent via email and a messaging application. As a result, the features of this image can appear within various metadata categories, such as the file system, EXIF, and messaging categories. Consequently, these features can lead to association among these incorporated categories, as illustrated in Figure 6-4.

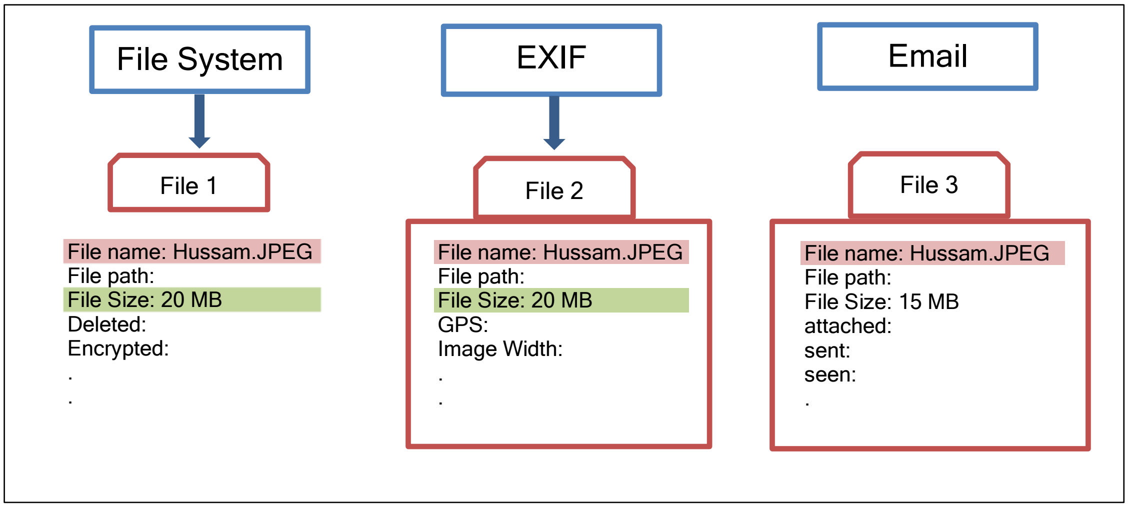


Figure 6-4: Metadata Association

Algorithm 6-3 illustrates the process of finding the association files based on their metadata that will be applied on the first cluster to identify the files of interest thereby leading to the next cluster for analysis.

Algorithm 6-3: Associated Files

```
Input: First cluster, other clusters.
Output: Suspected files.
---------------------------------------------------------------
Process
Step 1: Read the features of the first file in the first cluster.
Step 2: If the counter exceeds the end, go to step 6.
                    Else, go to step 3.
Step 3: Search in other all clusters about the matched features. If there is a
match, go to step 4.
                    Else go to step 5.
Step 4: Save the finding in temporarily buffer. Go to step 5.
Step 5: Read the features of the next file within the first cluster. Go to step 2.
Step 6: Arrange the findings based on the large proportion clusters that appear.
Step 7: End.
```

6.4 Experimental Methodology

This experiment seeks to validate the Automated Approach for Evidence Identification in finding the clusters containing the file of interest. In addition, there are various aspects that can influence the performance of the algorithm:

- The influence of algorithm type: how the clustering algorithms can affect the results.
- The influence of cluster size: what the cluster size does to algorithms in identifying the evidence.
- The influence of time window: the length of the time window is required to identify the related artefacts to evidential files.
- The influence of iteration numbers: the number of iterations is required to obtain the clusters containing notable artefacts.

The proposed approach was carried on the output of the clustering approach, as presented in the Chapter 5. The analysis of results is carried out on the four cases based on the above aspects.

6.4.1 Case 1 Analysis

The results of Case 1 for five iterations based on FCM, k-means, and k-medoid were presented in tables 6-2, 6-3, and 6-4 respectively. It is obvious from the results there is a difference in performance based on the algorithm type. The results showed the proportion of notable artefacts based on the clusters of k-means and k-medoids were slightly better than clusters of FCM. In the other words, over 77% of notable files was obtained based the mentioned clusters. This phenomenon happened because the clusters within k-means and k-medoids include fewer benign files than the clusters of FCM.

There are two controllable factors that can affect the performance of the algorithms: cluster size and the time window. Both factors are related to each other where the large setups of cluster sizes with a long time window demonstrated the best results in this case while the small setups with a short time window presented a high proportion of benign data compared with the long time window. However, the long time window with large setups offered a tiny proportion of benign data with an acceptable proportion of notable files.

Concerning the number of iterations, the best performance within this case was obtained by using five iterations. Noticeably, the wrong clusters were identified when increasing the number of iterations. Thereby, it would lead to analysing a large number of benign files instead of notable files. It is worth it to highlight that each iteration can identify a cluster either within the same category of the first cluster or different one. For instance, the AIE algorithm showed its ability to identify five clusters across all categories that were included in this case.

Table 6-2: Results of Case 1 based on FCM (✓: Notable; ×: Benign)

Time window (minute)	1		3		5	
Cluster size	✓	×	✓	×	✓	×
15	59.8	26.3	70.6	13.9	70.1	13.2
25	65	24	72.8	14	76.8	12.9
35	40.6	23.2	50.3	13.8	52.5	11.7
50	46.7	20.4	65.3	15.1	69.3	12.5
75	46.7	20.4	65.3	15.1	69.3	12.5
100	60	10.5	66.9	6.3	66.9	6.3

Table 6-3: Results of Case 1 based on K-Means

Time window (minute)	1		3		5	
Cluster size	✓	×	✓	×	✓	×
15	65.7	22.9	76.1	13.5	77.4	12.2
25	62.3	19.6	65.8	13.2	66.1	11.5
35	60.4	18.4	63.1	12.4	65.5	10.9
50	55.4	16.6	60.4	15.6	64.7	13.1
75	46.6	16.2	51.7	14.1	51.7	14.1
100	60.1	8.2	66.3	6.6	66.3	6.6

Table 6-4: Results of Case 1 based on K-Medoids

Time window (minute)	1		3		5	
Cluster size	✓	×	✓	×	✓	×
15	65.7	20.1	76.1	13.2	77.4	11.2
25	60.3	18.1	63.8	14.3	73.6	11.5
35	60.1	18.5	62.1	12.6	64.2	10.7
50	55.4	20.5	60.4	17.4	64.7	12.9
75	56.7	16	59.4	15.2	59.4	15.2
100	59.8	5.6	60.4	2.1	66.3	2.1

6.4.2 Case 2 Analysis

The results of Case 2 for two iterations based on the clusters of FCM, k-means, and k-medoid were presented in Tables 6-5, 6-6, and 6-7, respectively. This case proved successful results where more than 90% of notable files was obtained based on the clusters of three algorithms (FCM, k-means, and k-medoids). Generally, the influence of different clustering algorithms on the performance of AIE was examined and investigated empirically where the type of clustering algorithm within this case had no influence on the performance of the AIE algorithm. However, the performance of the AIE algorithm can be influenced by the cluster size and time window.

The best performance was obtained using a one-minute time window on 25-cluster size for FCM and 15-cluster size for k-means and k-medoids. The small duration of the time window with small setups of cluster size presented a good performance. The proportion of notable artefacts was decreased with increasing the time window. The notable files were also decreased by using the clusters of large setups. On the other hand, the proportion of benign files using the clusters of k-means and k-medoids with small a time window was relatively small compared to the FCM clusters.

The above analysis was based on two iterations of the AIE algorithm because the evidential artefacts within this case were only obtained in a small number of clusters. The further investigation showed that a high proportion of benign data was obtained with increasing the number of iterations more than two. For clarity, the first cluster was identified belonging to the file list category while the second cluster belongs to the EXIF category. This means the AIE algorithm has the ability to identify the evidence across various categories.

Table 6-5: Results of Case 2 based on FCM

Time window (minute)	1		3		5	
Cluster size	✓	×	✓	×	✓	×
15	77.6	18.7	77.6	20.3	77.6	25.1
25	92.3	17.37	92.3	17.3	77.6	22.6
35	83.4	9.7	77.6	9.3	77.6	9.3
50	64.9	9.4	77.6	12.4	77.6	12.4
75	62	6.3	62	6.3	68.3	9.6
100	67.9	6.7	77.6	12.4	77.6	12.4

Table 6-6: Results of Case 2 based on K-Means

Time window (minute)	1		3		5	
Cluster size	✓	×	✓	×	✓	×
15	90.1	14.2	77.6	23.1	77.6	29.4
25	85.5	12.5	82.2	13.9	77.6	19.7
35	85.5	12.4	82.2	13.9	77.6	19.7
50	82.5	8.7	82.5	8.7	77.6	18.1
75	63.6	9.5	58.2	7.8	56.7	9.4
100	40.4	5.8	48.4	6.2	45.3	7.2

Table 6-7: Results of Case 2 based on K-Medoids

Time window (minute)	1		3		5	
Cluster size	✓	×	✓	×	✓	×
15	90.1	13.8	77.6	19.8	77.6	29.4
25	85.6	12.9	85.4	12.9	77.6	25.1
35	83.4	12.2	84.1	11.9	77.6	25.8
50	75.4	11.8	72.5	12.1	67.9	22.2
75	63.6	9.1	62.5	9.5	57.3	10.8
100	48.3	5.8	42.4	7.3	34.8	9.7

6.4.3 Case 3 Analysis

The results of Case 3 based on the clusters of FCM, k-means, and k-medoids are presented in tables 6-8, 6-9, and 6-10, respectively. This case was considered a challenge because the performance of the AIE algorithm was low compared to the aforementioned cases. However, the AIE algorithm demonstrated its ability to identify over 69% of evidential files using the FCM clusters while a low performance was obtained based on the clusters of k-means, and k-medoids. This phenomenon might have occurred because the evidential files were small and scattered across five different categories with a large number of benign files.

It can be seen from the listed results on tables 6-8, 6-9, and 6-10 that the time window has little impact on the performance in identifying the notable files. It is noticeable that the longer length of the time window can lead to more files being selected during the timeline analysis. Thereby, a high proportion of both notable and benign files is expected. Consequently, the process of associating artefacts might have a potential influence on the performance to match the related files. In addition, the setup of cluster size showed a high impact on the algorithm's performance. The best performance was obtained using the largest cluster-size across all clustering methods. In contrast, the small setups of cluster sizes revealed poor results where over three quarters of the results were not identified.

Based on the results of this case, the best-obtained performance of the AIE algorithm was by applying five iterations. Notably, with increasing the iterations' number, the wrong clusters were selected, thereby leading to a large amount of noise data, which was subject to analysis. However, 60% of notable files was obtained within four iterations-based on 75-cluster size while the five iterations led to identifying wrong clusters. Thereby, the proportion of benign files became 3.1%, meaning, the AIE algorithm demonstrated its ability to find four to five clusters from the five categories in this case.

Table 6-8: Results of Case 3 based on FCM

Time window (minute)	1		3		5	
Cluster size	✓	✗	✓	✗	✓	✗
15	35.4	22.1	33.7	28.4	30.3	31.2
25	34.7	20.4	28.7	26.2	28.7	30.4
35	65.7	1.6	67.2	1.2	67.2	1.2
50	50.2	1.3	56	1.1	56	1.1
75	60.2	3.1	55.9	8.2	50.7	10.6
100	69.2	2.9	57.2	7.6	52.1	9.6

Table 6-9: Results of Case 3 based on K-Means

Time window (minute)	1		3		5	
Cluster size	✓	✗	✓	✗	✓	✗
15	17.2	24.1	17.2	25.6	17.2	29.2
25	33.4	21.2	29.6	22.2	27.1	29.1
35	26.7	11.6	22.3	12.6	23.3	16.7
50	59.6	2.4	58.4	4.1	58.4	4.1
75	61.1	0.7	63.4	0.9	63.4	1
100	62.7	0.6	63	0.5	63	0.5

Table 6-10: Results of Case 3 based on K-Medoids

Time window (minute)	1		3		5	
Cluster size	✓	×	✓	×	✓	×
15	20.1	23.7	19.4	24.9	18.7	26.5
25	19.8	23.1	19.8	25.1	19.8	26.4
35	23.9	12.6	20.3	13.7	20.3	14.6
50	60.4	1.1	63.1	1	66.7	0.9
75	58.1	1.3	59.9	1.1	59.9	1.1
100	63.4	0.6	64.2	0.5	64.2	0.5

6.4.4 Case 4 Analysis

The results of Case 4 based on the clusters of FCM, k-means, and k-medoids are presented in Tables 6-11, 6-12, and 6-13, respectively. It is apparent from the results that the performance of the AIE algorithm based on the FCM's clusters was better compared to its counterparts where more than 92% of evidential files was identified using the 35-cluster size. On the other hand, the proportion of notable files based on the clusters of k-means and k-medoids was less than 58% and 65%, respectively. This may have happened because the number of benign data within the first clusters was relatively high; consequently, this can lead to the clusters containing non-notable files.

The results highlighted that the setup of cluster size can influence the performance of the AIE algorithm in specific setups. For instance, 35-cluster size presented the highest performance among all the setups of FCM clusters while the 75-cluster size of k-means and the 100-cluster size of k-medoids demonstrated the best results. This occurred because there was a high proportion of notable artefacts with a small proportion of noise files included in the first cluster, thereby leading to the right sub-clusters. The time window had a small impact on the performance of the algorithm where the results were relatively similar among the three timeframes. However, the performance of the algorithm was noted to become low by increasing the length of the time window using the 75-cluster size of k-means. In contrast, the process of associating artefacts showed its ability to identify the sub-cluster containing the notable artefacts. The above analysis was conducted with only two iterations because the increase of iterations number gains high proportion of benign data instead of notable data. However, the AIE algorithm could only identify two clusters containing evidential artefacts. Those clusters were not located in a single category where the first one belongs to the file list while the second cluster belongs to the EXIF category.

Table 6-11: Results of Case 4 based on FCM

Time window (minute)	**1**		**3**		**5**	
Cluster size	✓	✗	✓	✗	✓	✗
15	43.3	3.2	43.3	3.7	43.3	4.1
25	43.3	2.4	43.3	2.9	43.3	3
35	92.9	0.2	92.9	0.2	92.2	0.2
50	57.3	1.1	43.3	3	43.3	3.3
75	42.9	2.1	53.5	2.2	42.9	2.5
100	43.3	2.5	45.8	2.7	43.3	2.9

Table 6-12: Results of Case 4 based on K-Means

Time window (minute)	1		3		5	
Cluster size	✓	×	✓	×	✓	×
15	43.3	3.2	43.3	3.7	43.3	4.1
25	43.3	3.2	43.3	3.7	43.3	4.1
35	43.3	3.2	43.3	3.6	43.3	3.9
50	43.3	3.2	43.3	3.8	43.3	3.9
75	57.3	1.2	45.2	1.8	45.2	1.8
100	43.3	2.3	43.3	2.5	43.3	2.9

Table 6-13: Results of Case 4 based on K-Medoids

Time window (minute)	1		3		5	
Cluster size	✓	×	✓	×	✓	×
15	43.3	3.2	43.3	3.7	43.3	4.1
25	43.3	3.2	43.3	3.7	43.3	4.1
35	43.3	2.8	43.3	3.4	43.3	3.5
50	43.3	3.1	43.3	3.6	43.3	3.7
75	64.2	1.1	64.2	1.1	64.2	1.1
100	43.3	2	43.3	2.3	43.3	2.7

6.5 Discussion

From the aforementioned results, the performance of the AIE algorithm is generally encouraging. It has proved its ability to identify more than 92% of notable artefacts with a small amount of benign data. The intelligent approaches of evidence identification in an automated way can help solve issues regarding the volume of data, the time taken to investigate, and the likelihood of human investigative error.

With the aim of gaining a proper analysis of interesting files during the investigation, it is important to identify the first cluster containing a large number of evidential artefacts. The algorithm has succeeded in determining the cluster including a large proportion of evidence across the four cases. Once the first cluster is specified, the process sub-cluster identification can be performed.

The results of sub-clusters were analysed based on four factors: the influence of the algorithm type, the influence of the cluster size, the influence of the time window, and the influence of the iteration number. Regarding the algorithm type, the performance of AIE using the clusters of FCM showed the best results across the four cases. This might have occurred because the AIE algorithm tries to identify the clusters with a large number of notable files where the clusters of FCM contained a high proportion of evidential artefacts compared to other clustering algorithms. In comparison, the results from the AIE algorithm based on the clusters of k-means and k-medoids also presented a good identification of evidence. In addition, a small proportion of benign data was obtained besides the evidential files.

Figure 6-5 illustrates the impact on the performance of the AIE algorithm based on the setup of cluster size and time window using the FCM clusters. It is noticeable from the figures that the cluster size has a high influence on the algorithm's performance. The performance of the first two cases was very good using the clusters of small setups. In contrast, the large setups of cluster size within the third case presented the best results. While the results of the fourth case revealed that only the 35-cluster size had a high impact on the performance, the others showed its inability to identify the evidential files. However, the variability in results across the four cases based the cluster size indicates the AIE algorithm can create better results using the small setups of cluster sizes with the cases containing a small number of artefacts. For the cases containing a large number of files, the large setups of cluster sizes would be suitable to obtain the algorithm's best performance.

The factor of time window can also have a high impact on the performance of the algorithm if the actions of a forensic case took place simultaneously. It is obvious the time window with a long duration can lead to more files for the analysis in terms of notable and benign data. In this situation, the results of the first case illustrated that the longer duration of the time window demonstrated more notable files than the shorter duration. In contrast, the second case showed that short durations with small setups of cluster sizes showed a better performance than long durations. The results of the third and fourth cases illustrated that time window has little impact on the algorithm's accuracy. With those cases, the process of associating artefacts showed the ability to identify the sub-clusters in assisting with timeline analysis.

Concerning how many iterations should be performed to gain the proper clusters, the results revealed all the above factors can be counted to determine the number of iterations. In addition, the number of metadata categories and the size of each category can determine the iteration number. For instance, using five iterations, the cases with larger numbers of artefacts appeared to operate better in larger configurations of cluster size while the cases with a small number of artefacts appeared to work better in small configurations using only two iterations.

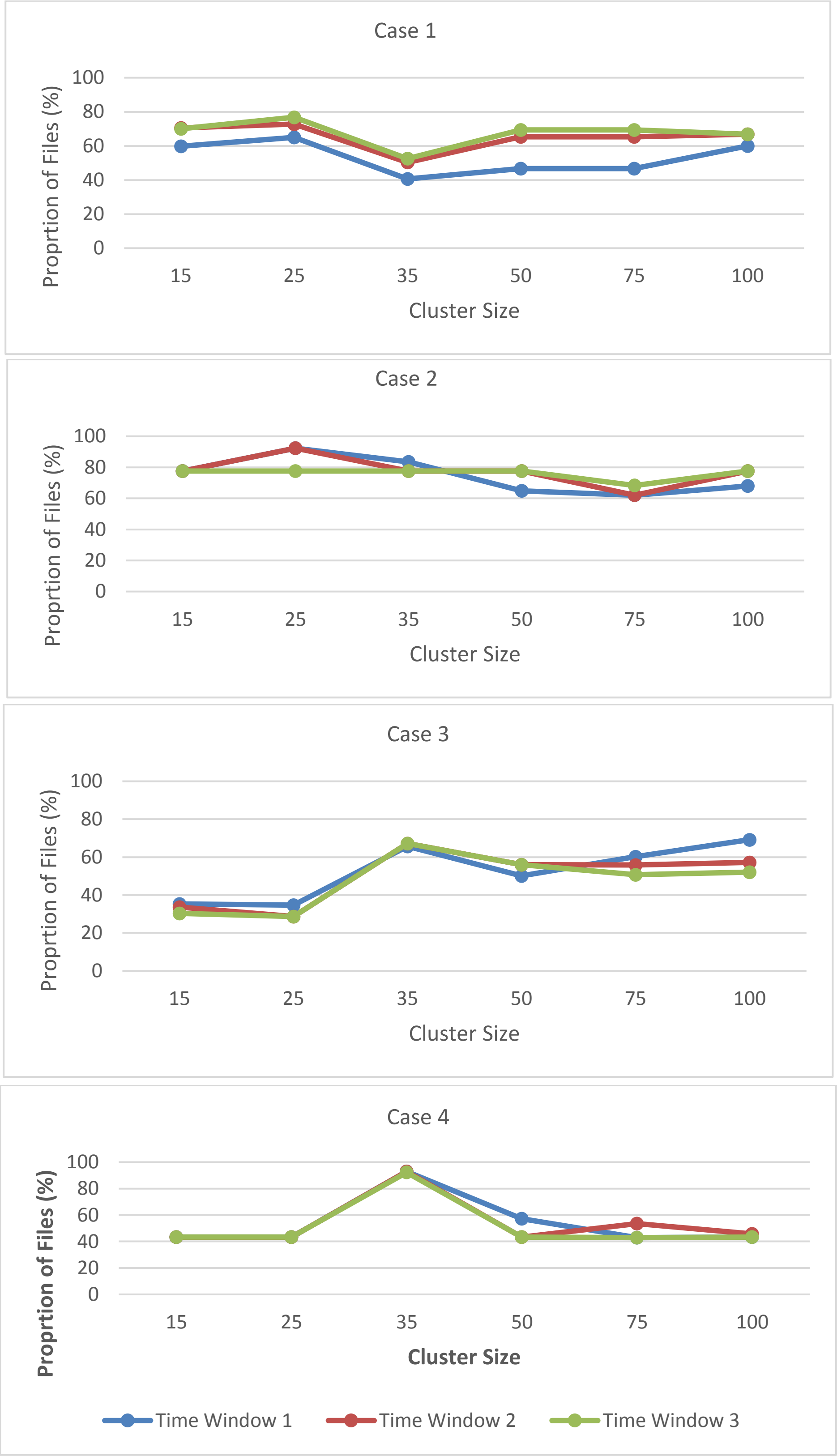


Figure 6-5: Impact of the cluster size and time window on AIE’s performance

6.6 Conclusion

This chapter has presented a developed algorithm to identify evidence in an automated way. The algorithm is performed on the output of a clustering approach to identify the first cluster and then sub-clusters. The experimental results revealed the developed algorithm can correlate the evidential artefacts from various clusters where those clusters come from different categories of data within a single case. In addition, the algorithm can be used as a triage tool to refine and evaluate the view of investigators in solving various computer-based crimes. Therefore, it can help to reduce the burden on the investigators in correlating the files in big and heterogeneous cases, thereby saving their time and effort.

7 Conclusion and Future Work

This chapter concludes the thesis by highlighting the contributions and achievements of research, the limitations and obstacles encountered, and outlining the potential areas that can be investigated in future research work. The research aimed to define, design, and develop an automated approach that can analyse and correlate evidence from big and heterogeneous resources in an efficient and timely manner.

This aim was achieved by examining the current state of the art to define the gap that needs to be addressed and by carefully studying the possible and most suitable approaches to dealing with the problem. Empirical experiments were conducted using various cases of real-life forensic data to validate the defined concept and the result was evaluated.

7.1 Contributions and Achievements of the Research

The research has achieved all the aims and the objectives stated in Chapter 1. The following points are the main achievements of this research:

- Investigating the domain of big and heterogeneous resources within digital forensic investigation from various aspects, such as volume of digital forensic data, the heterogeneity of evidence, and the required time to identify evidence.
- Demonstrating comprehensive literature of existing research in the domain of big and heterogeneous data to explore the aspects of the research problem that the literature has not addressed. These aspects were how to deal with similar datasets within a single case, how to apply clustering

methods on datasets containing string records, what clustering methods can work on forensics data, and how to identify the evidence in an automated way.

- Developing a novel algorithm for the merging of datasets through a ‘characterisation and harmonisation’ process. This algorithm provides a fusion of similar metadata categories across multiple and heterogeneous resources within a single case. Consequently, it leads to overcoming heterogeneity issues and making the examination and analysis easier.
- Developing a clustering approach using c-means (FCM), k-means, and k-medoids algorithms to identify the evidential files and isolate the non-related files based on their metadata.
- Developing an automated algorithm to identify the evidential artefacts based on a combined process of the clusters, timeline analysis, and association artefacts. This combination is used to provide a robust and refined artefact identification process.
- Conducting a series of experiments using both real life and public cases aiming to evaluate the effectiveness and the performance of the above-developed algorithms and approaches.

Several papers related to the research have been presented and published in refereed journals and conferences. As a result, the research is considered having made positive contributions to the field of digital forensic investigation and specifically in the domain of big and heterogeneous resources.

7.2 Limitations of Research

Although the research’s objectives have been achieved, a number of issues related to this research must be considered. Limitations of the research are follows:

- The experimental dataset was limited in terms of the number of resources being included in each case. Ideally, more resources would have provided a more reliable measure of performance that could be achieved in practice.
- The characterisation process demonstrated good results, but it failed to identify the nature of binary data in order to merge the right files.
- The clustering approach proved its ability to group the evidential artefacts within three clusters, but it has no ability to isolate the evidential files within a single cluster only.
- The AIE algorithm only depends on the association artefacts and timeline analysis to identify the sub-clusters, but it may not be an ideal process to apply on the cases with deleted executable files.

7.3 Opportunities for Future Work

The research contribution has improved the concerns of the heterogeneity of big data within digital forensics. Nevertheless, a number of further investigations relevant, particularly with the presented study scope, exist for future work. These suggestions are described below.

- Developing the harmonisation process to be more accurate by using an intelligent procedure by analysing the nature of binary data to merge similar fields. In addition, further evaluation is also required upon a wide range of technologies and applications to make the characterisation and harmonisations algorithms more generalised in practice.

- Using alternative algorithms of unsupervised machine learning in the clustering approach to generate centres of clusters thereby determining the only the evidential artefacts.
- Developing the AIE algorithm in terms of identifying the evidential artefacts using more various features of criminals, such as a criminal behaviour, as well as combining the AI techniques and AIE algorithm in determining additional artefacts, such as deleted files.

7.4 The Future of Heterogeneous Data in Digital Forensics

The continuing development of the storage technology, including increasing the storage capacity for customer devices and cloud computing services can clearly increase the challenges of digital forensic investigation. These challenges include the complexity, diversity, and correlation issues within forensic analysis. Despite that various digital forensic tools have been used in digital forensic investigations, their functionalities are not sufficiently enough to solve these above issues. In addition, these tools are struggling in dealing with various applications, such as WhatsApp, Skype, and many others. A few of forensic tools support multiple forensic images with a limited ability to correlate artefacts across file system and application data. Therefore, the examiner should use various forensic applications manually to examine and analyse the case. As a result, this research has suggested novel approaches to overcome the issue of heterogeneous data in which the examiner can identify the evidential artefacts in an automated way, thereby saving time and effort.

A further opportunity for future research relates to outcomes of triage processes on real-world devices and data to determine the most applicable methodology to deploy, which also provides for future needs, which could also include a review of the acceptance of triaged evidence in a legal environment and whether the various processes are potentially missing exculpatory evidence.

References

Ademu, I. O., Imafidon, C. O. & Preston, D. S. 2011. A new approach of digital forensic model for digital forensic investigation. *IJACSA) International Journal of Advanced Computer Science and Applications,* 2**,** 12.

Al Fahdi M 2016. *Automated Digital Forensics & Computer Crime Profiling.* Ph.D. thesis, Plymouth University.

Agarwal, A., Gupta, M., Gupta, S. & Gupta, S. 2011. Systematic digital forensic investigation model. *International Journal of Computer Science and Security (IJCSS),* 5**,** 118-131.

Alink, W., Bhoedjang, R. A. F., Boncz, P. A. & De Vries, A. P. 2006. XIRAF – XML-based indexing and querying for digital forensics. *Digital Investigation,* 3, Supplement**,** 50-58.

Allemang, D. & Hendler, J. 2011. *Semantic web for the working ontologist: effective modeling in RDFS and OWL*, Elsevier.

Almunawar, M. N., Anshari, M. & Susanto, H. 2018. Adopting Open Source Software in Smartphone Manufacturers' Open Innovation Strategy. Encyclopedia of Information Science and Technology, Fourth Edition. IGI Global, pp 7369-7381.

Anastasiou, D. & Vázquez, L. M. 2010. Localisation standards and metadata. *Metadata and Semantic Research.* Springer.

Aronson, J. E., Liang, T.-P. & Turban, E. (2005) Decision support systems and intelligent systems. Pearson Prentice-Hall.

Ayers, D. 2009. A second generation computer forensic analysis system. *digital investigation,* 6**,** S34-S42.

Äyrämö, S., & Kärkkäinen, T. (2006). Introduction to partitioning-based clustering methods with a robust example. Reports of the Department of Mathematical Information Technology. Series C, Software engineering and computational intelligence 1/2006.

Bandgar, S. B., Sale, M. & Meshram, B. 2012. , Artificial Intelligence Applied to digital Email for forensic Application. *International Journal of Managment, IT and Engineering,* 2, 114-122.

Baumgartner, K., Ferrari, S., & Palermo, G. 2008. Constructing Bayesian networks for criminal profiling from limited data. Knowledge-Based Systems, 21(7), 563-572.

Beebe, N. L. & Liu, L. 2014. Clustering digital forensic string search output. *Digital Investigation,* 11, 314-322.

Bennett, D. 2012. The challenges facing computer forensics investigators in obtaining information from mobile devices for use in criminal investigations. Information Security Journal: A Global Perspective, 21 (3). pp 159-168.

Benredjem, D. 2007. *Contributions to cyber-forensics: processes and e-mail analysis.* Concordia University.

Bezdek, J. C., Ehrlich, R., & Full, W. 1984. FCM: The fuzzy c-means clustering algorithm. Computers & Geosciences, 10(2-3), 191-203.

Biggs, Stephen; VIDALIS, Stilianos. Cloud computing: The impact on digital forensic investigations. In: Internet Technology and Secured Transactions, 2009. ICITST 2009. International Conference for. IEEE, 2009. p. 1-6.

Birgen, C., Preisig, H., & Morud, J. (2014). SQL vs. NoSQL. Norwegian University of Science and Technology, Scholar article.

Birk, Dominik; WEGENER, Christoph. Technical issues of forensic investigations in cloud computing environments. In: Systematic Approaches to Digital Forensic Engineering (SADFE), 2011 IEEE Sixth International Workshop on. IEEE, 2011. p. 1-10.

Buchholz, F. & Spafford, E. 2004. On the role of file system metadata in digital forensics. *Digital Investigation,* 1, 298-309.

Budu, J., & Boateng, R. (2015). Mobile Service Capabilities: Evidence from a Ghanaian Mobile Service Provider. International Journal of E-Services and Mobile Applications (IJESMA), 7(3), 1-17.

Bulbul, H. I., Yavuzcan, H. G., & Ozel, M. 2013. Digital forensics: an analytical crime scene procedure model (ACSPM). Forensic science international, 233(1-3), 244-256.

Burrell, J. (2016). How the machine 'thinks': Understanding opacity in machine learning algorithms. Big Data & Society, 3(1), 2053951715622512.

Cahyani, N. D. W., Martini, B., Choo, K. K. R., & Al-Azhar, A. K. B. P. 2017. Forensic data acquisition from cloud-of-things devices: windows Smartphones as a case study. Concurrency and Computation: Practice and Experience, 29(14).

Carrier, B. 2005. *File system forensic analysis*, Addison-Wesley Reading.

Carrier, B. & Spafford, E. H. An event-based digital forensic investigation framework. Digital forensic research workshop, 2004. 11-13.

Case, A., Cristina, A., Marziale, L., Richard, G. G. & Roussev, V. 2008. FACE: Automated digital evidence discovery and correlation. *digital investigation,* 5**,** S65-S75.

Casey, E. 2011. *Digital evidence and computer crime: Forensic science, computers, and the internet*, Academic press.

Chandarana, P. & Vijayalakshmi, M. Big Data analytics frameworks. Circuits, Systems, Communication and Information Technology Applications (CSCITA), 2014 International Conference on, 2014. IEEE, 430-434.

Chang, F., Zhu, L., Liu, J., Yuan, J. & Deng, X. A universal heterogeneous data integration standard and parse algorithm in real-time database. Proceedings of the 2012 International Conference on Information Technology and Software Engineering, 2013. Springer, 709-720.

Choo, K. K. R., Esposito, C., & Castiglione, A. (2017). Evidence and forensics in the cloud: challenges and future research directions. IEEE Cloud Computing, 4(3), 14-19.

Chow, K.-P., Law, F. Y., Kwan, M. Y. & Lai, P. K. The rules of time on NTFS file system. Systematic Approaches to Digital Forensic Engineering, 2007. SADFE 2007. Second International Workshop on, 2007. IEEE, 71-85.

Clarke, N. 2010. Computer Forensics. *New york: IT Governance Ltd*.

Cristofor, E. D., & Simovici, D. A. (2002). Information-theoretical methods in clustering. University of Massachusetts Boston.

Da Cruz Nassif, L. F. & Hruschka, E. R. Document clustering for forensic computing: An approach for improving computer inspection. Machine Learning and Applications and Workshops (ICMLA), 2011 10th International Conference on, 2011. IEEE, 265-268.

Danielsson, P. E. (1980). Euclidean distance mapping. Computer Graphics and image processing, 14(3), 227-248.

Dash, P. & Campus, C. 2014. Fast Processing of Large (Big) Forensics Data. *Indian Academy of Sciences Summer Research Fellowship Program* Institute of Development and Reaearch in Banking Technology.

Dilek, S., Çakır, H. & Aydın, M. 2015. Applications of Artificial Intelligence Techniques to Combating Cyber Crimes: A Review. *arXiv preprint arXiv:1502.03552*.

Elgendy, N. & Elragal, A. 2014. Big data analytics: a literature review paper. *Advances in Data Mining. Applications and Theoretical Aspects.* Springer.

Fasan, O. M. & Olivier, M. 2012. Reconstruction in database forensics. *Advances in Digital Forensics VIII.* Springer.

FBI. 2011. *Financial Crimes Report 2010-2011* [Online]. Available: https://www.fbi.gov/stats-services/publications/financial-crimes-report-2010-2011 [Accessed 17 March 2016].

Fensel, D., Bussler, C., Ding, Y., Kartseva, V., Klein, M., Korotkiy, M., Omelayenko, B. & Siebes, R. Semantic web application areas. NLDB Workshop, 2002. sn.

Fisher, D., Brush, A., Hogan, B., Smith, M. & Jacobs, A. 2007. Using social metadata in email triage: Lessons from the field. *Human Interface and the Management of Information. Interacting in Information Environments.* Springer.

Garfinkel, S. L. 2006. Forensic feature extraction and cross-drive analysis. *digital investigation,* 3**,** 71-81.

Garfinkel, S. L. 2010. Digital forensics research: The next 10 years. *digital investigation,* 7**,** S64-S73.

Ge, J., Qiang, B. & Chen, Z. 2012. Design and Application of Heterogeneous Data Semantic Integration System based on Domain Ontology. *International Journal of Advancements in Computing Technology,* 4.

Gholap, P. & Maral, V. 2013. Information Retrieval of K-Means Clustering For Forensic Analysis. *International Journal of Science and Research (IJSR)*.

Guerra, E. M. & Oliveira, E. 2013. Metadata-based frameworks in the context of cloud computing. *Cloud Computing.* Springer.

Guptill, S. C. 1999. Metadata and data catalogues. *Geographical information systems,* 2, 677-692.

Harichandran, V. S., Breitinger, F., Baggili, I., & Marrington, A. (2016). A cyber forensics needs analysis survey: Revisiting the domain's needs a decade later. Computers & Security, 57, 1-13

Hartigan, J. A., & Wong, M. A. (1979). Algorithm AS 136: A k-means clustering algorithm. Journal of the Royal Statistical Society. Series C (Applied Statistics), 28(1), 100-108.

Hoelz, B. W., Ralha, C. G., Geeverghese, R. & Junior, H. C. A cooperative multi-agent approach to computer forensics. Proceedings of the 2008 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology-Volume 02, 2008. IEEE Computer Society, 477-483.

Holder, E., Robinson, L., & Rose, K. (2009). Electronic Crime Scene Investigation: An On-the-Scene Reference for First Responders. US Department of Justice Office of Justice Programs, 810.

Horsman, G., Laing, C., & Vickers, P. (2011). A case based reasoning system for automated forensic examinations.

Kataria, M. & Mittal, M. P. 2014. BIG DATA: A Review. *International Journal of Computer Science and Mobile Computing,* Vol.3, 106-110.

Kent, K., Chevalier, S., Grance, T. & Dang, H. 2006. Guide to integrating forensic techniques into incident response. *NIST Special Publication*, 800-86.

Khan, M. N. A. 2008. *Digital Forensics using Machine Learning Methods.* University of Sussex.

Khanuja, H. K. & Adane, D. Database security threats and challenges in database forensic: A survey. Proceedings of 2011 International Conference on Advancements in Information Technology (AIT 2011), available at http://www. ipcsit. com/vol20/33-ICAIT2011-A4072. pdf, 2011. Citeseer.

Khanuja, H. K. & Adane, D. 2012. A framework for database forensic analysis. *Computer Science & Engineering: An International Journal (CSEIJ),* 2**,** 27-41.

Khanuja, H. K. & Adane, D. S. 2014. Forensic Analysis for Monitoring Database Transactions. *Security in Computing and Communications.* Springer.

Kogalovsky, M. R. 2013. Metadata in computer systems. *Programming and Computer Software,* 39**,** 182-193.

Kong, W., Wu, Q., Li, L. & Qiao, F. Intelligent Data Analysis and its challenges in big data environment. System Science and Engineering (ICSSE), 2014 IEEE International Conference on, 2014. IEEE, 108-113.

Lee, P. W. 2003. Metadata Representation and Management for Context Mediation. *Composite Information Systems Laboratory Working Paper# 2003,* 1.

Liu, Y., Liu, X. & Yang, L. Analysis and design of heterogeneous bioinformatics database integration system based on middleware. Information Management and Engineering (ICIME), 2010 The 2nd IEEE International Conference on, 2010. IEEE, 272-275.

Liu, Z. Y., Yan, F. B. & Zhang, B. H. Heterogeneous databases integration approach based on hybrid ontology. Applied Mechanics and Materials, 2012. Trans Tech Publ, 1948-1952.

Lonvick, C. 2001. The BSD syslog protocol.

Mangle, N. & Sambhare, P. B. 2013. A Review on Big Data Management and NoSQL Databases in Digital Forensics. *International Journal of Science and Research (IJSR)* Volume 4 Issue 5.

Manso-Callejo, M., Wachowicz, M. & Bernabé-Poveda, M. 2010. The design of an automated workflow for metadata generation. *Metadata and Semantic Research.* Springer.

Marinemetadata.Org. 2015. *Metadata Classifications | Marine Metadata Interoperability* [Online]. Available:

https://marinemetadata.org/guides/mdataintro/mdatadefined/howclassified [Accessed 17 Dec. 2015].

Martini, B. & Choo, K.-K. R. 2012. An integrated conceptual digital forensic framework for cloud computing. *Digital Investigation,* 9, 71-80.

Marziale, L. 2009. *Advanced Techniques for Improving the Efficacy of Digital Forensics Investigations.* Doctor of Philosophy, University of New Orleans.

McGuire, M., & Dowling, S. (2013). Cyber Crime: A Review of the Evidence. Research Report 75. Chapter 4: Improving the Cyber Crime Evidence Base. Home Office, available online at:
https://www.gov.uk/government/uploads/system/uploads/attachment_data/file/246756/horr75-chap4. pdf (accessed February 2017).

Mezghani, E., Exposito, E., Drira, K., Da Silveira, M. & Pruski, C. 2015. A Semantic Big Data Platform for Integrating Heterogeneous Wearable Data in Healthcare. *Journal of medical systems,* 39, 1-8.

Najafabadi, M. M., Villanustre, F., Khoshgoftaar, T. M., Seliya, N., Wald, R. & Muharemagic, E. 2015. Deep learning applications and challenges in big data analytics. *Journal of Big Data,* 2, 1-21.

Nakanishi, T. 2015. A Discovery Method of Anteroposterior Correlation for Big Data Era. *Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing.* Springer.

News, B. 2012. Barclays fined for attempts to manipulate Libor rates - BBC News. *BBC News.*

NIST. The CFReDS project. 2015.
https://www.cfreds.nist.gov/data_leakage_case/data-leakage-case.html

Noel, G. E. & Peterson, G. L. 2014. Applicability of Latent Dirichlet Allocation to multi-disk search. *Digital Investigation,* 11, 43-56.

Ochoa, X. & Duval, E. 2009. Automatic evaluation of metadata quality in digital repositories. *International Journal on Digital Libraries,* 10, 67-91.

Olivier, M. S. 2009. On metadata context in database forensics. *Digital Investigation,* 5, 115-123.

Palmer, G. A road map for digital forensic research. First Digital Forensic Research Workshop, Utica, New York, 2001. 27-30.

Patrascu, A. & Patriciu, V.-V. Beyond digital forensics. A cloud computing perspective over incident response and reporting. Applied Computational Intelligence and Informatics (SACI), 2013 IEEE 8th International Symposium on, 2013. IEEE, 455-460.

Palmer, Gary, et al. A road map for digital forensic research. In: First Digital Forensic Research Workshop, Utica, New York. 2001. p. 27-30.

Park, H. S., & Jun, C. H. (2009). A simple and fast algorithm for K-medoids clustering. Expert systems with applications, 36(2), 3336-3341.

Perry, R., Hatcher, E., Mahowald, R. P. & Hendrick, S. D. 2009. Force. com Cloud platform drives huge time to market and cost savings. *IDC-Whitepaper. URL: http://thecloud. appirio. com/rs/appirio/images/IDC_Force. com_ROI_Study. pdf [Zugriff: 29.04. 2010]*.

Perumal, S. 2009. Digital forensic model based on Malaysian investigation process. *International Journal of Computer Science and Network Security,* 9**,** 38-44.

Pollitt, M. M. Principles, practices, and procedures: an approach to standards in computer forensics. Second International Conference on Computer Evidence, 1995. 10-15.

Press, N. 2004. Understanding metadata. *National Information Standards,* 20.

Pringle, N. & Burgess, M. 2014. Information assurance in a distributed forensic cluster. *Digital Investigation,* 11**,** S36-S44.

Qi, M. Digital forensics and NoSQL databases. Fuzzy Systems and Knowledge Discovery (FSKD), 2014 11th International Conference on, 2014. IEEE, 734-739.

Quick, D., & Choo, K. K. R. (2016). Big forensic data reduction: digital forensic images and electronic evidence. Cluster Computing, 19(2), 723-740.

Quick, D. & Choo, K.-K. R. 2014. Data reduction and data mining framework for digital forensic evidence: storage, intelligence, review and archive. *Trends & Issues in Crime and Criminal Justice,* 480**,** 1-11.

Quick, D. & Choo, K.-K. R. 2014. Impacts of increasing volume of digital forensic data: A survey and future research challenges. *Digital Investigation,* 11, 273-294.

Quick, D., Martini, B. & Choo, R. 2013. Cloud storage forensics. *Elsevier Ltd.*, 273-294.

Raghavan, S. 2014. A framework for identifying associations in digital evidence using metadata.

Raghavan, S., Clark, A. & Mohay, G. 2009. FIA: an open forensic integration architecture for composing digital evidence. *Forensics in telecommunications, information and multimedia.* Springer.

Raghavan, S. & Raghavan, S. AssocGEN: engine for analyzing metadata based associations in digital evidence. Systematic Approaches to Digital Forensic Engineering (SADFE), 2013 Eighth International Workshop on, 2013. IEEE, 1-8.

Raghavan, S. & Raghavan, S. 2014. Eliciting file relationships using metadata based associations for digital forensics. *CSI transactions on ICT,* 2, 49-64.

Rajakumari, S. B. 2014. An Evolution of Forensic Data Analysis and Methodologies. *Middle-East Journal of Scientific Research,* 19, 904-907.

Richard III, G. G. & ROUSSEV, V. 2006. Digital forensics tools: the next generation. *Digital crime and forensic science in cyberspace*, 75.

Roussev, V. & Quates, C. 2012. Content triage with similarity digests: the M57 case study. *Digital Investigation,* 9, S60-S68.

Rowe, N. C. 2014. Identifying forensically uninteresting files using a large corpus. *Digital Forensics and Cyber Crime.* Springer.

Rowe, N. C. & Garfinkel, S. L. 2012. Finding anomalous and suspicious files from directory metadata on a large corpus. *Digital Forensics and Cyber Crime.* Springer.

Ruback, M., Hoelz, B. & Ralha, C. 2012. A new approach for creating forensic hashsets. *Advances in Digital Forensics VIII.* Springer.

Russell, S. J., & Norvig, P. (2016). Artificial intelligence: a modern approach. Malaysia; Pearson Education Limited,.

Sagiroglu, S. & Sinanc, D. Big data: A review. Collaboration Technologies and Systems (CTS), 2013 International Conference on, 2013. IEEE, 42-47.

Shang, W., Jiang, Z. M., Hemmati, H., Adams, B., Hassan, A. E. & Martin, P. Assisting developers of big data analytics applications when deploying on hadoop clouds. Proceedings of the 2013 International Conference on Software Engineering, 2013. IEEE Press, 402-411.

Sharma, A., Chaudhary, B. & Gore, M. Metadata Extraction from Semi-structured Email Documents. Computing in the Global Information Technology, 2008. ICCGI'08. The Third International Multi-Conference on, 2008. IEEE, 56-61.

Tannahill, B. K. & Jamshidi, M. 2014. System of Systems and Big Data analytics–Bridging the gap. *Computers & Electrical Engineering,* 40**,** 2-15.

Tanenbaum, A. S. 2009. Modern operating system. Pearson Education, Inc.

UK, R. 2014. *EBay considering PayPal spinoff - report* [Online]. Available: http://uk.reuters.com/article/uk-ebay-divestiture-idUKKBN0GL1LQ20140821 [Accessed 17 March 2016].

Vaarandi, R. 2005. *Tools and Techniques for Event Log Analysis*, Tallinn University of Technology Press.

Wagstaff, K., Cardie, C., Rogers, S., & Schrödl, S. (2001, June). Constrained k-means clustering with background knowledge. In ICML (Vol. 1, pp. 577-584).

Xu, X.-B., Yang, Z.-Q., Xiu, J.-P. & Chen, L. 2013. A big data acquisition engine based on rule engine. *The Journal of China Universities of Posts and Telecommunications,* 20**,** 45-49.

Xu, R., & Wunsch, D. (2005). Survey of clustering algorithms. IEEE Transactions on neural networks, 16(3), 645-678.

Yadav, S. 2011. Analysis of Digital Forensic and Investigation. *International Journal of Computer Science & Information Technology,* 1.

Yang, S., Son, Y. & Chung, M. 2014. A Digital Forensic Model Based on the Generated Fuzzy Number Using FCM Clustering. *Ubiquitous Information Technologies and Applications.* Springer.

Zhenyou, Z., Jingjing, Z., Shu, L. & Zhi, C. Research on the integration and query optimization for the distributed heterogeneous database. Computer Science and Network Technology (ICCSNT), 2011 International Conference on, 2011. IEEE, 1533-1536.

Zuckerman, L. (2014). Montana health record hackers compromise 1.3 million people. Available: https://www.reuters.com/article/us-usa-hacker-montana-idUSKBN0F006I20140625. [Accessed 19 April 2016].

Zuech, R., Khoshgoftaar, T. M. & Wald, R. 2015. Intrusion detection and Big Heterogeneous Data: a Survey. *Journal of Big Data,* 2**,** 1-41.

Appendix A - Publications

1. Mohammed, H. J., Clarke, N., & Li, F. (2016). An Automated Approach for Digital Forensic Analysis of Heterogeneous Big Data. Journal of Digital Forensics, Security and Law, 11(2), 137-152.

DOI: https://commons.erau.edu/jdfsl/vol11/iss2/9/

Abstract: the major challenges with big data examination and analysis are volume, complex interdependence across content, and heterogeneity. The examination and analysis phases are considered essential to a digital forensics process. However, traditional techniques for the forensic investigation use one or more forensic tools to examine and analyse each resource. In addition, when multiple resources are included in one case, there is an inability to cross-correlate findings which often leads to inefficiencies in processing and identifying evidence. Furthermore, most current forensics tools cannot cope with large volumes of data. This paper develops a novel framework for digital forensic analysis of heterogeneous big data. The framework mainly focuses upon the investigations of three core issues: data volume, heterogeneous data and the investigators cognitive load in understanding the relationships between artefacts. The proposed approach focuses upon the use of metadata to solve the data volume problem, semantic web ontologies to solve the heterogeneous data sources and artificial intelligence models to support the automated identification and correlation of artefacts to reduce the burden placed upon the investigator to understand the nature and relationship of the artefacts.

2. Mohammed, H. J., Clark, N. L., & Li, F. (2018). Automating the harmonisation of heterogeneous data in digital forensics. In 17th European Conference on Cyber Warfare and Security (pp. 299-306). Academic Conferences and Publishing International Limited.

DOI: https://core.ac.uk/download/pdf/160743377.pdf

Abstract: Digital forensics has become an increasingly important tool in the fight against cyber and computer assisted crime. However, with an increasing range of technologies at people's disposal, investigators find themselves having to process and analyse many systems (e.g. PC, laptop, tablet, Smartphone) in a single case. Unfortunately, current tools operate within an isolated manner, investigating systems and applications on an individual basis. The heterogeneity of the evidence places time constraints and additional cognitive loads upon the investigator. Examplels of heterogeneity include applications such as messaging (e.g. iMessenger, Viber, Snapchat and Whatsapp), web browsers (e.g. Firefox and Chrome) and file systems (e.g. NTFS, FAT, and HFS). Being able to analyse and investigate evidence from across devices and applications based upon categories would enable investigators to query all data at once. This paper proposes a novel algorithm to the merging of datasets through a 'characterisation and harmonisation' process. The characterisation process analyses the nature of the metadata and the harmonisation process merges the data. A series of experiments using real-life forensic datasets are conducted to evaluate the algorithm across five different categories of datasets (i.e. messaging, graphical files, file system, Internet history, and emails), each containing data from different applications across difference devices (a total of 22

disparate datasets). The results showed that the algorithm is able to merge all fields successfully, with the exception of some binary-based data found within the messaging datasets (contained within Viber and SMS). The error occurred due to a lack of information for the characterisation process to make a useful determination. However, upon the further analysis it was found the error had a minimal impact on subsequent merged data.

3. Mohammed, H., Clarke, N., & Li, F. (2018). Evidence identification in heterogeneous data using clustering. In Proceedings of the 13th International Conference on Availability, Reliability and Security (p. 35). ACM.

DOI: https://dl.acm.org/citation.cfm?id=3233271

Abstract: Digital forensics faces several challenges in examining and analyzing data due to an increasing range of technologies at people's disposal. The investigators find themselves having to process and analyze many systems manually (e.g. PC, laptop, Smartphone) in a single case. Unfortunately, current tools such as FTK and Encase have a limited ability to achieve the automation in finding evidence. As a result, a heavy burden is placed on the investigator to both find and analyze evidential artifacts in a heterogenous environment. This paper proposed a clustering approach based on Fuzzy C-Means (FCM) and K-means algorithms to identify the evidential files and isolate the non-related files based on their metadata. A series of experiments using heterogenous real-life forensic cases are conducted to evaluate the approach. Within each case, various types of metadata categories were created based on file systems and applications. The results showed that the clustering based on file systems gave the best results of grouping the evidential artifacts within only five clusters. The proportion across the five clusters was 100% using small configurations of both FCM and K-means with less than 16% of the non-evidential artifacts across all cases -- representing a reduction in having to analyze 84% of the benign files. In terms of the applications, the proportion of evidence was more than 97%, but the proportion of benign files was also relatively high based upon small configurations. However, with a large configuration, the proportion of benign files became very low less than 10%. Successfully prioritizing large proportions of evidence and reducing the volume of benign files to be analyzed, reduces the time taken and cognitive load upon the investigator.