PhD Thesis

Alleviating the Digital Forensic Backlog: A Methodology for Automated Digital Evidence Processing

Xiaoyu Du

A thesis submitted in fulfilment of the degree of

PhD in Computer Science

Supervisor: Dr. Mark Scanlon

Head of School: Assoc. Prof. Chris Bleakley

UCD School of Computer Science

College of Science

University College Dublin

September 2020

Table of Contents

List of Figures . . . . . **vii**

List of Tables . . . . . **ix**

Abstract . . . . . **xiii**

1 Introduction **1**
1.1 Overview 1
1.1.1 Challenges of Digital Forensics 2
1.1.2 Approaches and Technologies for the Problem 2
1.2 Research Questions 4
1.3 Contribution of this Work 4
1.4 Limitations of this Work 5
1.5 Layout of this Thesis 5
1.6 Brief Overview of the Approach 6

2 Literature Review **7**
2.1 Introduction 7
2.2 Digital Forensics 8
2.2.1 Source of Digital Evidence 9
2.2.2 Mobile Forensics 10
2.2.3 Cloud Forensics 11
2.2.4 IoT Forensics 12
2.2.5 Digital Forensic Artefacts Example 13
2.2.6 Digital Evidence Backlogs 16
2.3 Digital Forensic Test Images 17
2.3.1 Current Available Disk Images 18
2.3.2 Automated Disk Image Generation Approach 18
2.4 Digital Forensic Process Model 19
2.4.1 Digital Forensic Framework in Initial Phase 21
2.4.2 Refined Digital Forensic Process Models 22
2.4.3 Recent Digital Forensic Models for Handling Modern Advancements 23
2.5 Triage Process Model 25
2.5.1 Digital Device Triage Tools 26
2.6 Cloud-based Digital Forensic Framework 27
2.6.1 DFaaS Framework 27

2.6.2 HANSKEN: DFaaS System Used by NFI . . . 28
2.6.3 Benefits and Advantages of DFaaS . . . 29
2.7 Data Deduplication and Data Reduction . . . 30
2.7.1 Data Deduplication Technology . . . 30
2.7.2 Data Deduplication in Digital Forensics . . . 31
2.7.3 Data Reduction Approaches . . . 32
2.8 Automated Digital Forensic Analysis . . . 33
2.8.1 Challenges of Automation in Digital Forensics . . . 33
2.8.2 Metadata and Timeline Analysis . . . 34
2.8.3 Plaso/Log2timeline . . . 35
2.9 Machine Learning and Digital Forensics . . . 35
2.9.1 Background of Machine Learning . . . 35
2.9.2 Machine Learning in Digital Forensics . . . 36
2.9.3 Background of Deep Learning . . . 38
2.9.4 Deep Learning Applications in Digital Forensics . . . 39
2.9.5 Current Challenges and Future Directions . . . 40
2.10 Correlation Analysis . . . 41
2.10.1 Cyber-investigation Analysis Standard Expression (CASE) . . . 42
2.11 File Artefact Prioritisation . . . 42
2.12 Summary . . . 43
2.12.1 Gaps in the State of the Art . . . 43

3 Methodology . . . 45
3.1 Introduction . . . 45
3.1.1 A Centralised Digital Evidence Processing System . . . 45
3.1.2 An Automated File Artefact Analysis Approach . . . 46
3.1.3 Design of Experimentation and Evaluation . . . 47
3.2 Tools for Test Disk Images Generation . . . 48
3.2.1 TraceGen: Overview . . . 48
3.2.2 TraceGen: Existing Automation Options . . . 49
3.2.3 EviPlant: Image Generation Using “Evidence Packages” . . . 51
3.3 Deduplicated Data Acquisition of the System . . . 52
3.3.1 Deduplicated Acquisition Process Pipeline . . . 53
3.3.2 Client Responsibilities . . . 54
3.3.3 Sever Responsibilities . . . 54
3.3.4 Summary . . . 55
3.4 Data Storage of the System . . . 55
3.4.1 Categorising Files Collected from the Disk . . . 56
3.4.2 Unallocated Space and Slack Space on the Disk . . . 56
3.4.3 Metadata and Acquisition Records . . . 57
3.4.4 Acquisition Logs for Performance Analysis . . . 59
3.5 Forensically Sound Image Reconstruction from Deduplicated Acquisition . . . 60
3.6 Data Extraction and Preparation . . . 62
3.6.1 Data Extraction: Tools and Techniques . . . 62

3.6.2 File Data Reduction/Selection . . . 62
3.6.3 File Timeline Generation and Feature Extraction . . . 63
3.6.4 Feature Extraction from File system Metadata . . . 64
3.6.5 Train/Test Data and Evaluation . . . 65
3.7 Relevancy File Artefacts Prioritisation . . . 65
3.7.1 An Overview of the Workflow . . . 67
3.7.2 Relevancy Score Determination . . . 68
3.7.3 Adding Pre-trained Models . . . 69
3.7.4 Relevancy Score Generation . . . 70
3.8 Summary . . . 71

4 Test Disk Image Generation . . . 73
4.1 Overview . . . 73
4.1.1 Choice of Technologies . . . 73
4.2 File Data for Disk Image Creation . . . 74
4.3 EviPlant: Creation, Manipulation and Distribution Disk Image . . . 75
4.3.1 Overview . . . 75
4.3.2 Evidence Packages . . . 75
4.3.3 Testing of Diffing and Injection . . . 76
4.3.4 Summary . . . 77
4.4 TraceGen: Automated User Action Emulation . . . 78
4.4.1 Overview . . . 78
4.4.2 Virtual Machine Configuration . . . 78
4.4.3 File Provenance and Interactions . . . 79
4.4.4 User Action Emulation . . . 80
4.4.5 Continuous Usage Trace Generation . . . 80
4.4.6 Summary . . . 84

5 Deduplicated Digital Evidence Processing . . . 86
5.1 Overview . . . 86
5.2 Prototype Setup and Test Data . . . 87
5.3 Results: Average Acquisition Speed . . . 87
5.4 Results: Storage Space Saved . . . 91
5.5 Results: Image Reconstruction . . . 91
5.6 Tests on an Improved Approach for Data Acquisition . . . 92
5.6.1 Results . . . 93
5.7 Summary . . . 94
5.7.1 Benefits of this Approach . . . 95

6 File Artefact Analysis and Relevancy Prioritisation . . . 96
6.1 Overview . . . 96
6.1.1 Data Processing and Experimentation . . . 96
6.2 Metadata and Timeline from Disk Image . . . 97
6.2.1 Timeline Generation and Analysis . . . 98

6.2.2 An Example Disk Image Timeline . . . 99
6.2.3 File Timeline Generation . . . 101
6.2.4 An Example of File Timeline . . . 102
6.2.5 Feature Extraction from File Timeline . . . 102
6.3 Metadata-based File Artefact Classification . . . 103
6.3.1 Example Scenario . . . 104
6.3.2 Datasets . . . 104
6.3.3 Evaluation Matrix . . . 105
6.3.4 Precision-Recall Curves with Average Precision Scores . . . 105
6.3.5 Result Comparison on Two Datasets . . . 108
6.4 Image File Artefact Classification . . . 109
6.4.1 Experimental Data . . . 109
6.4.2 Experimental Setup and Results . . . 110
6.5 File Artefact Relevancy Prioritisation . . . 112
6.5.1 Experimental Data . . . 112
6.5.2 Example Scenarios . . . 113
6.5.3 Case Investigation and Relevancy Prioritisation . . . 113
6.6 Summary . . . 115
6.6.1 Comparison with the Existing Methodology . . . 115
6.6.2 Benefits of this Approach . . . 116
6.6.3 Limitations of this Approach . . . 116

7 Conclusion and Future Work . . . 118
7.1 Summary of the Work . . . 118
7.1.1 Deduplicated Digital Evidence Acquisition System . . . 118
7.1.2 Automated File Artefacts Relevancy Analysis . . . 119
7.2 Conclusion . . . 119
7.3 Future Work . . . 120
7.3.1 Multiple Devices Investigation Scenario . . . 120
7.3.2 Exploration of Advanced Machine Learning Techniques . . . 121

Statement of Original Authorship

"I hereby certify that the submitted work is my own work, was completed while registered as a candidate for the degree stated on the Title Page, and I have not obtained a degree elsewhere on the basis of the research presented in this submitted work."

Acknowledgements

I would like to express my sincere gratitude to my supervisor Dr. Mark Scanlon for the continuous support of my Ph.D study and research, for his patience, motivation, and encouragement. His guidance helped me in all the time of research and writing of this thesis. My sincere thanks also go to Sixun Ouyang and Jinghui Lu for their help in reviewing my stage transfer report and many interesting discussions in machine learning techniques. Many thanks to Dr. Nhien-An Le-Khac, Dr. Asanka Sayakkara and Felix Anda for many discussions and presentations on digital forensics research. I would also like to thank my viva examiners: Prof. Nicole Beebe and Dr. Liliana Pasquale for their valuable advice on my thesis. Thanks also to the viva chair Dr. David Lillis.

List of Figures

1.1 An Overview of the Research Outlined as Part of this Thesis 6

2.1 Autopsy: Ingest Modules 14
2.2 Autopsy: Data Extraction, Views and Results 15
2.3 Different Types of Digital Forensics 20
2.4 Digital Forensic Framework in Initial Phase 22
2.5 Digital Forensics Frameworks Focusing on a Specific Use Cases 24
2.6 Recent Digital Forensic Models for Handling Modern Advancements 25
2.7 Digital Forensic as a Service 29
2.8 Machine Learning Pipeline 36
2.9 Why Deep Learning? (Slide by Andrew Ng) 38

3.1 Overall System Design 46
3.2 Methodology Overview 48
3.3 TraceGen: Overview of the Approach 49
3.4 EviPlant: Evidence Package Extraction and Injection 51
3.5 File Signature Search Process Overview [1] 53
3.6 Deduplicated Acquisition 53
3.7 Categories of Files on a suspect Device 57
3.8 File Metadata for Deduplication and Image Reconstruction 58
3.9 The Acquired Disk Image 59
3.10 Forensically Sound Disk Image Reconstruction from Deduplicated System . . 61
3.11 The Developed Toolkit for Data Extraction and Processing 62
3.12 Data Processing: File Timeline Generation and Feature Extraction 64
3.13 Relevant File Prioritisation Approach 67
3.14 Work-flow of Relevant File Prioritisation in an Investigation 68
3.15 The Input for Relevancy Score 69
3.16 Relevancy Score Generation 71

4.1 Sample Pictures Used to Populate Test Image 74
4.2 Diffing Tool Storage Savings for Evidence Packages 77
4.3 User Action of Files on the Disk 79
4.4 User Action Emulation: Launch Chrome 80
4.5 The Method to Create Files on the Disk 81
4.6 The Method to set the System Time to a Given Date and Time 82
4.7 Sample Input CSV File 82
4.8 Event Counts of Each Day in September 84

4.9 Event Counts Increment of Each Day in October 85

5.1 Deduplicated Evidence Acquisition Process 87
5.2 Evidence Acquisition Speed of Each Image 89
5.3 System Speed and Efficient Speed Comparison of Each Image 90
5.4 Duplication Ratios and their Impact on Speed 90
5.5 Storage Saving as Number of Acquisitions Increase 91
5.6 Reconstruction Speed of Each Image 92
5.7 An Improved Approach for Deduplicated Data Acquisition 93

6.1 Disk Images Generation and Data Processing for Analysis 98
6.2 Method for File Timeline Generation 101
6.3 An Example File Artefact Timeline 102
6.4 Precision-Recall Curves - Decision Tree 106
6.5 Precision-Recall Curves - Gaussian Naïve Bayes 106
6.6 Precision-Recall Curves - k-NN 107
6.7 Precision-Recall Curves - SVM 107
6.8 Precision-Recall Curves - Logistic Regression 108
6.9 Training Data for Image File Artefact Classification 109
6.10 Training Model on `ResNet-50`: Loss/Train 110
6.11 Training Model on `ResNet-50`: Loss/Test 111
6.12 Training Model on `ResNet-50`: Accuracy/Test 111
6.13 Relevancy Score Generation in the Experiment 114

List of Tables

3.1 Data Selection by File Type/File Extension 63
3.2 Valuable Feature Extracted 65

4.1 Volume of Filesystem Modifications Before and After Story Execution 83
4.2 Event Counts from Different Sources 83

5.1 Test Disk Images 88
5.2 Disk Image Compression Test 94

6.1 Device Related Event 100
6.2 File Artefacts Event - Common 100
6.3 File Artefacts Event - Specific 101
6.4 Example Features from a File Timeline 103
6.5 Datasets Used in the Experiment 104
6.6 Evaluation Matrix of Different Classification Algorithms and Datasets 109
6.7 "Benign" File Information 112
6.8 "Illegal" File Information 112
6.9 Recall of Each Model 115

List of Abbreviations

AI	Artificial Intelligence
API	Application Programming Interfaces
AP Score	Average Precision Score
CNNs	Convolutional Neural Networks
CSAM	Child Sexual Abuse Material
CSEM	Child Sexual Exploitation Material
CSP	Cloud Service Provider
CV	Computer Vision
DFaaS	Digital Forensic as a Service
FAT	File Allocation Table
FFNN	Feedforward Neural Networks
GDPR	General Data Protection Regulation
GUI	Graphical User Interface
IaaS	Infrastructure as a Service
IoT	Internet of Things
LSTM	Long Short-term Memory
MFT	Master File Table
NFI	Netherlands Forensic Institution
NIST	National Institute of Standards and Technology
NLP	Natural Language Processing
NSRL	National Software Reference Library
NTFS	New Technology File System
SaaS	Software as a Service
SCSI	Scientific Crime Scene Investigation
SFTP	SSH File Transfer Protocol
SHA1	Secure Hash Algorithm 1
SVM	Support Vector Machine

OS	Operating System
PaaS	Platform as a Service
PCA	Principal Component Analysis
PE	Preinstallation Environment
RNNs	Recurrent Neural Networks
UFED	Universal Forensic Extraction Device
VM	Virtual Machine

List of Publications

- **Journal Papers**

  - Du, X., Hargreaves, C., Sheppard, J., and Scanlon, M., TraceGen: User Activity Emulation for Digital Forensic Test Image Generation, Forensic Science International: Digital Investigation, ISSN 2666-2825, September 2020. [2]

  - Scanlon, M., Du, X., and Lillis, D. (2017). EviPlant: An Efficient Digital Forensic Challenge Creation, Manipulation and Distribution Solution. Digital Investigation, 20, S29-S36, Elsevier, `https://doi.org/10.1016/j.diin.2017.01.010`. [3]

- **Conference Papers**

  - Du, X., Hargreaves, C., Sheppard, J., Anda, F., Sayakkara, A., Le-Khac, N-A., and Scanlon, M., SoK: Exploring the State of the Art and the Future Potential of Artificial Intelligence in Digital Forensic Investigation, 13th International Workshop on Digital Forensics (WSDF), held at the 15th International Conference on Availability, Reliability and Security (ARES), Virtual Event, August 2020. [4]

  - Du, X., Le, Q., and Scanlon, M., Automated Artefact Relevancy Determination from Artefact Metadata and Associated Timeline Events, The 6th IEEE International Conference on Cyber Security and Protection of Digital Services (Cyber Security), Dublin, Ireland, June 2020. [5]

  - Du, X., and Scanlon, M. Methodology for the Automated Metadata-Based Classification of Incriminating Digital Forensic Artefacts, The 12th International Workshop on Digital Forensics (WSDF), held at the 14th International Conference on Availability, Reliability and Security (ARES), Canterbury, UK, August 2019. [6]

  - Du, X., Ledwith, P., and Scanlon, M. (2018). Deduplicated Disk Image Evidence Acquisition and Forensically-Sound Reconstruction. The 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications (IEEE TrustCom-18). [7]

  - Du, X., and Scanlon, M. (2017). Evaluation of Digital Forensic Process Models with Respect to Digital Forensics as a Service, 16th European Conference on Cyber Warfare and Security (ECCWS), Dublin, Ireland. [8]

- **Poster**

  - Du, X., and Scanlon, M. (2017). Expediting the Digital Forensic Process through a Deduplicated Framework, 16th European Conference on Cyber Warfare and Security (ECCWS), Dublin, Ireland.

Abstract

The ever-increasing volume of data in digital forensic investigation is one of the most discussed challenges in the field. Severe, case-hindering digital evidence backlogs have become commonplace in law enforcement agencies throughout the world. The objective of the research outlined as part of this thesis is to help alleviate the backlog through automated digital evidence processing. This is achieved by reducing or eliminating, redundant digital evidence data handling through leveraging data deduplication and automated analysis techniques. This helps avoid the repeated re-acquisition, re-storage, and re-analysis of common evidence during investigations. This thesis describes a deduplicated evidence processing framework designed with a Digital Forensic as a Service Framework (DFaaS) paradigm in mind.

In the proposed system, prior to the acquisition, artefacts are hashed and compared with a centralised database of previously analysed files to identify common files. Moreover, this process facilitates known pertinent artefacts to be detected at the earliest stage possible in the investigation, i.e., during the acquisition step. The proposed methodology includes a novel, forensically-sound entire disk image reconstruction technique from a deduplicated evidence acquisition system. That is to say, reconstructed disk hashes match the source device without having to acquire all artefacts directly from it. This enables remote disk acquisitions to be possible faster than the network throughput. Known, i.e., previously encountered, pertinent artefacts identified during the acquisition stage are then used for training machine learning models to create a relevancy score for the unknown, i.e., previously unencountered, file artefacts. The proposed technique generates a relevancy score for file similarity using each artefact's file system metadata and associated timeline events. The file artefacts are subsequently ordered by these relevancy scores to focus the investigator towards the analysis of artefacts most likely to be relevant to the case first.

Chapter 1: **Introduction**

1.1 Overview

Digital devices are omnipresent and have become an important part of people's daily lives. Therefore, digital forensics is not only applied to technological crime such as computer intrusion, phishing scams, and so on. Actually, digital evidence is increasingly important during the processing of many civil and criminal cases, e.g., child abuse, drug activity, financial crimes, murder, missing persons, death threats, etc. As a result, the number of cases requiring digital forensic investigation is ever-increasing.

In the age of big data, massive amounts of data are produced through people's everyday activities. The availability of large-scale potential evidence presents new challenges in digital forensic investigations. The capacity of storage devices (hard drives, solid-state drives, memory cards, external storage devices, etc.) is ever-expanding. Besides, the rising prevalence of Internet of Things (IoT) devices in homes and buildings increase the opportunities to recover digital traces that are relevant to an investigation [9].

The more data is available, the harder it is to identify fraudulent activity and the malicious users behind those activities [10]. Severe digital evidence backlogs are faced by law enforcement around the world [11, 12, 13]. Hence, reducing the growing backlog is urgent; as Casey states "investigation delayed is justice denied" [14]. Data deduplication is a technique for eliminating duplicate copies of repeating data; it can be applied to the digital forensic process to reduce the amount of data that must be collected and analysed.

To alleviate the digital evidence backlogs, approaches such as data reduction, data deduplication, devices triage, file artefact ranking and automated association analysis can be applied for collecting, preserving and analysing information during an investigation. Previous analysis results can be preserved into a database, which stores the hash value of the illegal files that have been classified by forensic experts. Illegal files could be child abuse material (pictures, videos), documents demonstrating financial fraud, and so on. Leveraging previously analysed data is also important to avoid duplication of effort in re-analysis the same content or for training machine learning models. This research contributes to an improvement over the current solutions and proposes approaches to improve the efficiency of digital evidence processing.

1.1.1 Challenges of Digital Forensics

The presence of a mixed set of technologies is one factor increasing the complexity of the problem [15]. Future challenges could include investigation within the smart infrastructure and smart cities [16, 17]. Digital forensics requires to battle efficiency of process big data and the ability to processing heterogeneous data.

Digital forensic evidence acquisition speed is traditionally limited by two main factors: the read speed of the storage device being investigated and the storage device for preserving the collected data. Big forensic data is a widely discussed topic in the field [18, 19] and it brings both challenges and opportunities. Expediting the acquisition combined with automated correlation analysis can solve the data volume issue. This is further complicated by the variety of data sources.

For example, the acquisition of data from cloud services is complex; the data is often scattered among different physical or virtual locations and the deployments are various, e.g., Infrastructure as a Service (IaaS), Software as a Service (SaaS), Platform as a Service (PaaS), etc. IoT devices further complicate matters, as these are typically connected to cloud servers through a network and much of the forensically interesting data, e.g., event logs, user activities, etc. are stored in volatile memory [9]. Powering down these devices can result in inadvertent evidence destruction.

Technological advancement brings both challenges and opportunities to digital forensics. In digital forensic research surrounding cloud computing, it could either focused on acquiring and analysing digital evidence from cloud services or applying cloud computing resources to processing digital evidence.

1.1.2 Approaches and Technologies for the Problem

Streamlining the digital forensics process model can have a significant effect on digital evidence backlogs. For example, processing less data (data reduction/device triage) combined with improving the hardware (more computing power) can result in. Data reduction can eliminate non-pertinent file artefacts through filters created at the earliest stage in the investigation, such as to remove image file size less than 5KB. The analysis phase requires automated correlation and prioritisation of file artefacts to reduce the processing time.

- **Digital Forensic Process Model**

  Digital forensic investigative approaches usually are discussed in the context of digital forensic process models. Novel solutions should be integrated to the existing processes. For example, Simou et al. [20] designed a concept model of cloud forensic-enabled services (CFeS) considering cloud forensics are challenging; in their research, cloud forensic constraints at the stages of digital evidence processing, such as accountability,

tractability, are defined. CFeS allows investigators to collect data for analysis while better protecting the system.

For big data forensics, a coordinated process model should be proposed. Digital forensic tools need a new software and hardware framework for large scale forensic processing [21]. One practical solution to these problems is supporting more hardware resources. A solution is proposed to improve the performance by using GPUs for the most computationally intensive tasks of the indexing procedure [22]. Cloud computing, a model for enabling convenient, on-demand remote access to a shared pool of configurable computing resources, can be a solution.

- **Data Reduction**

  Data reduction is required as the data volume and the number of files encountered during each investigation is ever-increasing, which results in slower digital evidence processing. Data reduction approaches can be implemented by verifying the metadata such as file size, file type, file location, etc., to reduce the number of files requiring expert analysis. Experiment results have proved the effectiveness of forensic corpus data reduction techniques to eliminate non-pertinent files for faster analysis [23]. In addition, selective imaging chooses likely more relevant file artefacts from disk and aids in the decision making for device triage [24]. The higher prioritised devices by triage should, of course, have whole disk images acquired, because of the importance of forensic soundness of evidence acquisition [25, 26, 27].

  Hash values of file data can also be applied for file identification. Centralised repositories for preserving previously analysed files can aid the analysis. Leveraging previously processed digital forensic cases and their component artefact relevancy classifications can facilitate an opportunity for training automated AI (artificial intelligence) based evidence processing systems. These can significantly aid investigators in the discovery and prioritisation of both previously encountered and un-encountered evidence.

- **Automated Association/Prioritisation**

  Automated digital forensic tools are necessary, considering the time and cost-efficiency. Automated Child Sexual Abuse Material (CSAM) analysis tool can limit investigative exposure to this content, to avoid investigators working on it at risk for developing secondary traumatic stress [28].

  Manual analysis of large datasets takes too long to arrive at meaningful results. In 2005, Beebe et al. [29] outlined data mining techniques that could be applied to aid digital forensic investigations, as terabyte-sized data sets were already challenging analysts and investigators. In 2019, Karresand et al. [30] pointed out the problem of the large and increasing amount of data to be processed in digital forensics and proposed an approach for prioritising relevant areas in storage media. This approach allows an analysis of the probable position of user data at first without the need for a working file system.

  Machine learning techniques have been applied to assists digital forensics; for instance, in the triage of seized digital devices [31]. An approach using trained support vector

machine (SVM) models for ranking digital forensic string search hits was proposed [32]. Machine Learning is often seen as the solution to many big data problems. Automated evidence processing (leveraging AI-based techniques) shows great promise in expediting the digital forensic analysis process while increasing case processing capacities [4].

1.2 Research Questions

The central focus of this research is to design and implement methodologies to alleviate the digital evidence backlogs through centralised evidence storage, processing and analysis. This research focuses on stored data and file system analysis, which is the most common source of pertinent evidence. The proposed approach aims to reduce the workload of the investigators by eliminating repetitive work and automating digital evidence processing.

The research questions are:

- How can investigation efficiency be improved through centralised evidence storage, processing and analysis?
- How can current research on digital forensic objectives (triage, correlation and prioritisation) be improved through technologies such as data reduction, data mining and machine learning?
- How can redundant analysis on the same digital artefacts be reliably eliminated by employing data deduplication techniques?
- What impact on the level of automation of digital forensic investigation through combining event-based analysis with filesystem metadata is possible?

1.3 Contribution of this Work

The proposed system enables deduplicated acquisition and forensically sound reconstruction. Subsequently, the prior artefact is leveraged for training supervised machine learning models to prioritise file artefacts by their likely relevancy. The premise of the approach is for file artefact relevancy prioritisation to be facilitated through taking advantage of the data stored on a centralised, deduplicated digital evidence system from the previous case analysis. The novelty and contributions of this research are listed as follows:

- A framework for automated test disk image generation is proposed. A prototype is developed, which externally controls a virtual machine running in VirtualBox. The

developed tool is used for the test disk image generation required by this research. The implemented user actions include file creation, access, modification, double click to run an application, click links in an opened web-page, etc.

- A deduplicated digital evidence processing system is proposed. The system allows deduplicated remote digital evidence acquisition, and detection of illegal files at the acquisition stage. Known benign file artefacts are filtered out and redundant re-acquisition and re-analysis of previously encountered content are eliminated. Device triage can be determined by the number of known illegal files detected.

- A novel technique for forensically-sound, complete disk image reconstruction from a deduplicated digital evidence process system is achieved and validated.

- An approach for automated file artefacts prioritisation is proposed and evaluated on several emulated investigative scenarios. This approach uses the detected known file artefacts to train machine learning models for determining the likely relevancy of the unknown file artefacts on the disk. File artefact timelines and metadata are used as input to the model.

1.4 Limitations of this Work

This research considers digital evidence acquisition, analysis and data processing. Experimentation and tests are focused on the most popular operating system; namely Windows. However, this approach can be applied to diverse devices.

This research applies data deduplication and machine learning techniques for reducing or entirely eliminating repeated processing and analysis of common benign or pertinent digital evidence. However, the time saved from deduplication varies in each investigation determined by the data on the seized device. The experimentation conducted evaluated the time saving across several scenarios.

1.5 Layout of this Thesis

- Chapter 2 presents a comprehensive literature review of the background and related work of this research.

- Chapter 3 outlines the methodologies for test image generation, deduplicated digital evidence acquisition and automated file artefacts prioritisation.

- Chapter 4 presents the implementation for test disk image generation and results.

- Chapter 5 presents the implementation and evaluation of a deduplicated centralised digital evidence processing system. Test results demonstrate the validity of this approach.
- Chapter 6 presents the application of a machine learning-based approach for digital file artefacts analysis. The approach assumes the digital evidence is collected by the designed deduplicated system. Text can detect illegal files and benign files at the acquisition stage. The classification of file artefacts is used as a training data set and applied to train models for determining the relevancy of unknown, file artefacts.
- Chapter 7 presents the conclusion and future work.

1.6 Brief Overview of the Approach

This research discusses the challenges and state-of-the-art in digital forensics. The heterogeneity of data encountered in digital forensics requires evidence discovery from a large variety of devices and data formats. The ever increasing data volumes encountered during investigation significantly slows down the analysis phase of an investigation (e.g., filtering, indexing, etc.). Leveraging cloud computing resources and techniques can greatly improve the efficiency of digital evidence processing.

This research proposes a system integrated with a DFaaS framework, leveraging data deduplication and automated file artefact relevancy determination. The focus of this research is to address the issue through a more efficient process model; reducing unnecessary data processing (both during acquisition and analysis) and automated evidence analysis to focus the investigator towards the data most likely to be pertinent evidence at the earliest stage possible in the investigation.

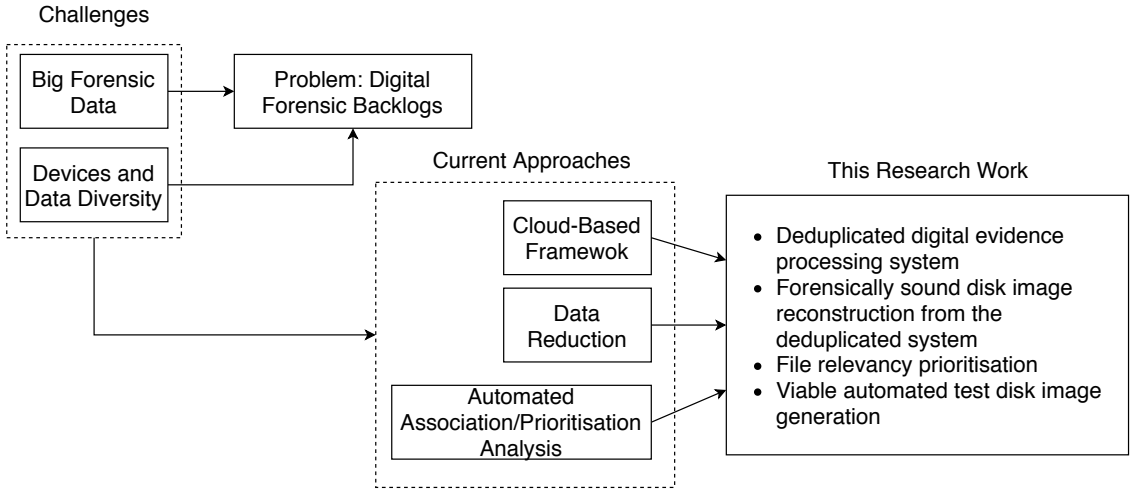


Figure 1.1: An Overview of the Research Outlined as Part of this Thesis

Chapter 2: **Literature Review**

2.1 Introduction

Computer forensics developed as an independent field in the late 1990s and early 2000s when computer-based crime started growing with the increasing popularity of computers [33]. Modern society increasingly depends on communication networks, mobile appliances, IoT solutions, cyber-physical system technologies, and cloud-based services [15]. In this chapter, the state of the art of the field is highlighted including current techniques and methodologies, digital forensics problems and challenges.

Section 2.2 outlines the concept of digital forensics, digital evidence, the current challenges of digital forensics and the situation of digital evidence backlogs faced by law enforcement. Digital evidence originates from various abstraction levels. Autopsy analysis of disk image is presented for showing the data extracted from the disk image. The challenges resulting in digital evidence backlogs are also presented.

Purchasing storage media from the second-hand market was an original source of research data, however, this can no longer be used after the GDRP was released. The lack of shared digital forensic corpora is a problem for scientific validation of digital evidence [34]. As a result, testing disk images are typically only generated manually. In Section 2.3, existing test disk image creation approaches are discussed.

Digital forensic science is very much still in its infancy but is becoming increasingly invaluable to investigators. A popular area for research is seeking a standard methodology to make the digital forensic process accurate, robust, and efficient. In Section 2.4, digital forensic process models are presented. Process models define the methodology for investigation in different scenarios. They specify phases of digital evidence processing and normally feature phases including acquisition, analysis and presentation.

In this age of big data, large volume digital forensic cases produce their own problems. Section 2.7, data deduplication and the existing applications of data deduplication and data reduction in digital forensics are presented. Applying cloud computing techniques to digital forensics can contribute to solving the big forensic data problem. In Section 2.6, Hansken, a cloud-based solution for digital forensics (Digital Forensic as a Service (DFaaS)) system used by the Netherlands Forensic Institution (NFI) is presented. Section 2.8 presents recent research on automatic digital forensic analysis. Automated digital forensic analysis techniques are diverse due to the variety in both types of cases and devices. Machine learning techniques

have been applied to develop applications for addressing problems in security and digital forensics. In Section 2.9, the background of machine learning and developed applications in digital forensics are presented.

2.2 Digital Forensics

Cybersecurity is more focused on the design of secure systems, the prevention and detection of attacks, whereas, digital forensics deals with the investigation after the incident. That is to say when cybersecurity fails to protect systems from damage; it requires digital forensic practitioners to investigate how an incident happened. Many civil or criminal cases, e.g., child abuse, financial fraud, etc., require digital forensic investigation. With the ever-increasing integration of technology in day-to-day life, the ownership of multiple devices, such as smartphones or laptops, is commonplace. For a forensic investigator, the increase in the amount of data that any single person can accumulate across several devices can be a daunting prospect. The increase means longer acquisition times, larger storage requirements and lengthy analysis. All of this time threatens to further extend the current forensic backlog which in some jurisdictions is already unacceptably long. This can leave some cases without vital evidence, case-progressing for years[12].

Digital evidence could be any data restored within digital devices, including files, file fragments, digital audio, digital video, cell phones, digital fax machines, etc. [35]. As the prevalence of smart devices, digital evidence could also from smart vehicle [36], smart city [17]. Digital evidence can be useful in a wide range of criminal investigations, such as network-based attacks, child abuse, drug dealing, etc. Digital forensic investigations are various depending on the device type (e.g., computer, smartphone, IoT) and data format (e.g., pictures, documents, database, network packets). Several process models have been proposed to deal with different case scenarios. The most common steps in the digital investigation are acquisition, examination, analysis and presentation. The subsequent effort on process model evolution is presented in Section 2.4.

Cloud computing can be thought of from two perspectives; one is to regard it as a target to be investigated and the other is to utilise it for powerful investigative analysis. For example, Lanterna et al. [37] present the work on the analysis of deduplicated file systems; Scanlon [38] has a discussion on applying deduplication to the digital forensic framework. As Lee et al. [21] outline the majority of studies have been focused on the former, which considers cloud computing services as forensically pertinent targets.

It is also important to improve the automation of forensic processes [39]. Existing tools designed for automatic digital forensic investigation are mainly focused on the extraction of data from storage media, which often cannot directly answer the question asked by detectives during an investigation. As the case types are various, forensic analysis requires some degree

of critical thinking and implementation of the scientific method [40]. Investigators have to discover the story manually, which is often arduous. Lack of IT professionals conducting the digital forensic investigation in law enforcement is another problem encountered with manual analysis [41].

Machine learning techniques offer a data-driven approach, which results in a more dynamic solution compared with hardcoded scripts. Researchers in many fields look to machine learning to improve the effectiveness and efficiency of their solutions; the same is true in the digital forensics and cybersecurity domains. For example, machine learning has been employed in malware classification [42], establishing forensic analysis priorities [43], automated categorisation of digital media [31], etc.

2.2.1 Source of Digital Evidence

Digital evidence is electronic information stored or transmitted in binary form [44]. Data extracted from seized devices can be used as digital evidence. The role of the seized devices could be various, determined by the case under investigation. For example, Li et al. [45] refined the roles of IoT investigation into IoT as a target, IoT as a tool, and IoT as a witness. It is also true of other types of devices during an investigation. For example, in a cyber attack investigation, the suspect's devices could be a witness or tool, or the victim's devices could be the targets.

With the increasing amount of data, the types of evidence encountered has also increased [46]. From a variety of devices (e.g., PC, laptops, smartphone, tablet, etc.), data can be extracted as digital evidence from three abstract levels:

- Physical level: the block slack and unallocated space could contain information of deleted files;
- File system level: contains information such as files' indices, files' physical locations and allocated blocks;
- Operating system level: contains system and application files, log files, and user-created files.

Pertinent file artefacts on operating system level include; 1) system and application files, which offer information such as version and updating situation to investigators; 2) log files (e.g., registry, cache, history), which record account information and all the user's actions; 3) user-created files usually are documents, multimedia files, content or metadata can help reconstruct what the device was used for, e.g., images and videos are common digital evidence for child abuse cases.

Cloud forensics and IoT forensics are often mixed with network forensics, as data is frequently transmitted by a network to assure their functionality. Cloud forensic investigation could

require digital evidence acquisition from client, server and network. IoT forensics could include evidence collected from cloud servers and services, network and physical devices.

The demand for cloud computing is ever increasing [47]. There is no single central location for files, database artefacts, system artefacts and logs; thereby creating great challenges to identify, lock (to ensure integrity), and retrieve them [48]. The evolution of cloud computing forensics is in its infancy [49]. Currently, there is no standard method or toolset for conducting cloud investigations, or for evaluating and certifying proposed tools.

Watson et al. [50] states the high-level challenges associated with performing digital forensics on IoT devices. Due to most existing forensic tools having been developed for computer or mobile phone forensics, the onboard data storage in IoT devices is often is not accessible to these tools. Furthermore, the cumulative dataset for a device may exist in multiple locations. Even if the data can be acquired, due to the different file systems and file formats, it may not be readable or accessible with existing tools.

Big data is a blend of structured as well as unstructured data. Big data is characterised by the five Vs, which are variety, velocity, volume, veracity and value [51]. The age of big data opens new opportunities in various fields, it also introduces new challenges in digital forensics investigations [10].

Continuous best practice evolution leads to difficulties in digital forensic investigation. The challenges of digital forensics include the increasing popularity of digital devices and the heterogeneity of the hardware and software platforms being used [15]. Furthermore, Montasari et al. [16] outline challenges in digital forensics such as anti-forensic techniques, video and rich media, whole drive encryption, wireless, virtualisation, live response, distributed evidence, borderless cybercrime and dark web tools, combined with a lack of standardised tools and methods, usability and visualisation.

2.2.2 Mobile Forensics

NIST defines mobile forensics as "the science of recovering digital evidence from a mobile device under forensically sound conditions using accepted methods" [44]. The proliferation of mobile devices (i.e., smartphones and tablets) on the consumer market has caused a growing demand for forensic examination of the devices. In addition, law enforcement is much more likely to encounter a suspect with a mobile device than before in either criminal or civil cases.

Over time, commercial tools were developed, which allowed analysts to recover phone content with minimal interference and examine it separately. Mobile forensic toolkits are available in the market including Oxygen forensics, Cellebrite's Universal Forensic Extraction Device (UFED), and XRY Forensic Examiner's Kit [52]. These tools help to extract certain informative data. The basic information obtained from mobile phones, despite the device differences [53, 54, 52]:

- Phone Storage Data
- Messages: SMS, MMS, E-Mails
- Social Media Data
- Call Records: Missed/Outgoing/Incoming Calls
- Multimedia Data: Photos, Videos and Audio files
- Communication Network Data

As Cahyani et al. [55] outlines, existing mobile forensic research can be classified into (1) examining the capabilities of acquisition methods, (2) undertaking detailed forensic procedures, and (3) conducting in-depth forensic analysis of mobile apps or mobile operating systems.

As an example, Sathe et al. [56] presented their experimentation on the data extracted from Samsung Galaxy Grand Duos GT I9082 mobile device by using AFLogical OSE, Andriller and Wondershare Dr. Fone for Android tool. The results shown below demonstrate the difference in the data extraction from each tool:

- *AFLogical OSE*: messages, call logs;
- *Andriller Tool*: messages, call logs, web browser, wifi password, accounts;
- *Wondershare Dr. Fone for Android*: messages, call logs, contacts, images, audio, video, documents, WhatsApp message and attachments. (both deleted and undeleted data)

Al et al.[57] present a model for processing data Android mobiles; experimentation presents the application of commercial tools for data extraction and analysis from *Samsung Android 4.2.2*. However, in-depth analysis of all seized devices during an investigation could take a quite long time; triage on mobile devices can address this issue. In 2011, Marturana et al.[58] proposed an automated approach for mobile devices triage, which takes advantage of Data Mining and Machine Learning theories.

2.2.3 Cloud Forensics

Cloud computing has gained popularity because it offers various benefits including convenience, large capacity, scalability, and on-demand accessibility. However, attacks being discovered and exploited in various cloud-related crimes have led to a need for digital forensics in the cloud environment [59].

Precise definitions and understanding of terms related to new technologies can be various at times. To better understand cloud forensics, such as its definition, scope, challenges,

opportunities as well as missing capabilities, a survey among digital forensic experts and practitioners was conducted by Ruan et al.[60]. According to the results, the respondents believe that cloud forensics is not Internet forensics or classical computer forensics, nor a brand new area. It is rather a mixture of traditional forensic techniques and their applications in a cloud computing environment. Pichan et al. grouped the investigation into three areas, (i) Client forensics (ii) Cloud forensics, and (iii) Network forensics [48].

As cloud-based file synchronisation services have become very popular, which offers a remote backup of data paired with the automation of data across multiple devices. In 2014, a methodology was outlined [61] enabling the recovery of remote digital evidence from decentralised file synchronisation network, as a result, extends the digital evidence acquisition window. In 2018, Teing et al. [62] presented the types and locations of CloudMe residual artefacts on desktop and mobile client devices running Windows 8.1, Ubuntu 14.04.1 LTS, Mac OS X Mavericks 10.9.5, iOS 7.1.2, and Android KitKat 4.4.4.

Experimentation of acquiring forensic evidence from IaaS cloud computing has been conducted to test the current tools, such as EnCase, FTK, etc. The experiments assume that the cloud consumer is the victim of the crime and the plan-tiff in the investigation [63]. It is more challenging to conduct an investigation of suspect's cloud-stored data.

For mitigating the dependency on Cloud Service Provider (CSP), acquisition of data from the cloud could also be conducted through monitoring the communication between client and cloud server. Alex et al.[64] proposed a cloud evidence collection model. The model builds a forensic monitoring plane (FMP) between the client and cloud service provider, which monitoring tool forwards the request to the server and the response to the client. And then FMP sends the collected data to the forensic server.

Event reconstruction in the cloud is challenging due to its multi-tenancy and the huge scale of events generated per unit time [65]. A variant of the popular log aggregation algorithm Leader-Follower (LF) algorithm called LF_{V1} and LF_{V2} was proposed by Raju et al. [65, 66] for cloud log event reconstruction.

2.2.4 IoT Forensics

From connected cars to traffic lights, home security systems, connected toys and smart speakers, the IoT market has ballooned in recent years. According to Cisco[1], the number is predicted to reach 31 billion connected devices by 2020 and 75 billion devices by 2025. Accordingly, the importance of IoT forensics will increase. IoT forensics is challenging as it is a mixer of devices level forensics, network forensics and cloud forensics.

Due to the heterogeneity, customised forensic tools are required to acquire and analyse data from certain devices, rather than the typical commercial forensic tools such as EnCase and

[1]https://www.cisco.com/c/en/us/solutions/internet-of-things/future-of-iot.html

FTK [45]. Furthermore, IoT forensics requires to consider data provenance and the interaction between IoT and cloud servers, network forensics tools or methods are generally applied [45]. Cloud forensics also plays a key role in IoT forensics, especially since the data generated from IoTware and IoT networks are already being, or will increasingly be stored, on cloud locations [59].

In 2017, Kebande et al. [67] proposed a post-event response mechanism to these malicious attacks in cloud-based IoT infrastructures. For example, smart, connected products and IoT in many organisations can result in increased efficiency and improved cash flow. The proposed Cloud-Centric Framework enables to isolate big data as forensic evidence from IoT infrastructures [67].

Decentralised methods are applied for IoT device investigation. To address the problems and limitations of digital forensics in the IoT environment, Ryu et al. [68] proposed a framework using blockchain technology in 2019. This framework stores IoT device activity in the blockchain as transactions to assure the integrity and security of data.

2.2.5 Digital Forensic Artefacts Example

This section presents an example of disk image forensic analysis using Autopsy, an open-source digital forensic tool.

Autopsy[2] is a GUI (Graphical User Interface) based program that allows analysing hard drives and smartphones efficiently. The SleuthKit is a collection of command-line tools and a C library that allows to analyse disk images and recover files from them. It is used behind the scenes in Autopsy and many other open-source and commercial forensics tools. *pytsk* is a Python binding for the SleuthKit and is used for file system metadata extraction in this research work.

Figure 2.1 illustrates the information and analysis of a hard drive investigation by Autopsy. The data source is a Windows 7 disk image in raw format. After selecting the disk image, the available models to process the extracted data are shown.

Autopsy supports the standard features common in digital forensics, such as:

- Recent activity
- Hash calculation and lookup
- Indexed keyword search
- Registry analysis
- Web artefacts

[2]https://www.sleuthkit.org/

- Email
- Carving

To combat big forensic data, autopsy stores all the analysed data in the past (data of previously analysed case) to avoid reanalysis of digital evidence. Local and centralised repositories are employed to store the historical data record. As a result, unique features are offered such as:

- Multi-user collaborative cases
- Analysis-driven acquisition
- Triage
- Timeline
- Communications

Autopsy frequently adds new third-party modules and encourages developers to write new modules. For example, *plaso* (log2timeline), the state of the art on timestamp extraction, is on the ingest modules list. *plaso* is also used by this research work and more discussion on *palso* is presented in Section 2.8.2 (timeline analysis).

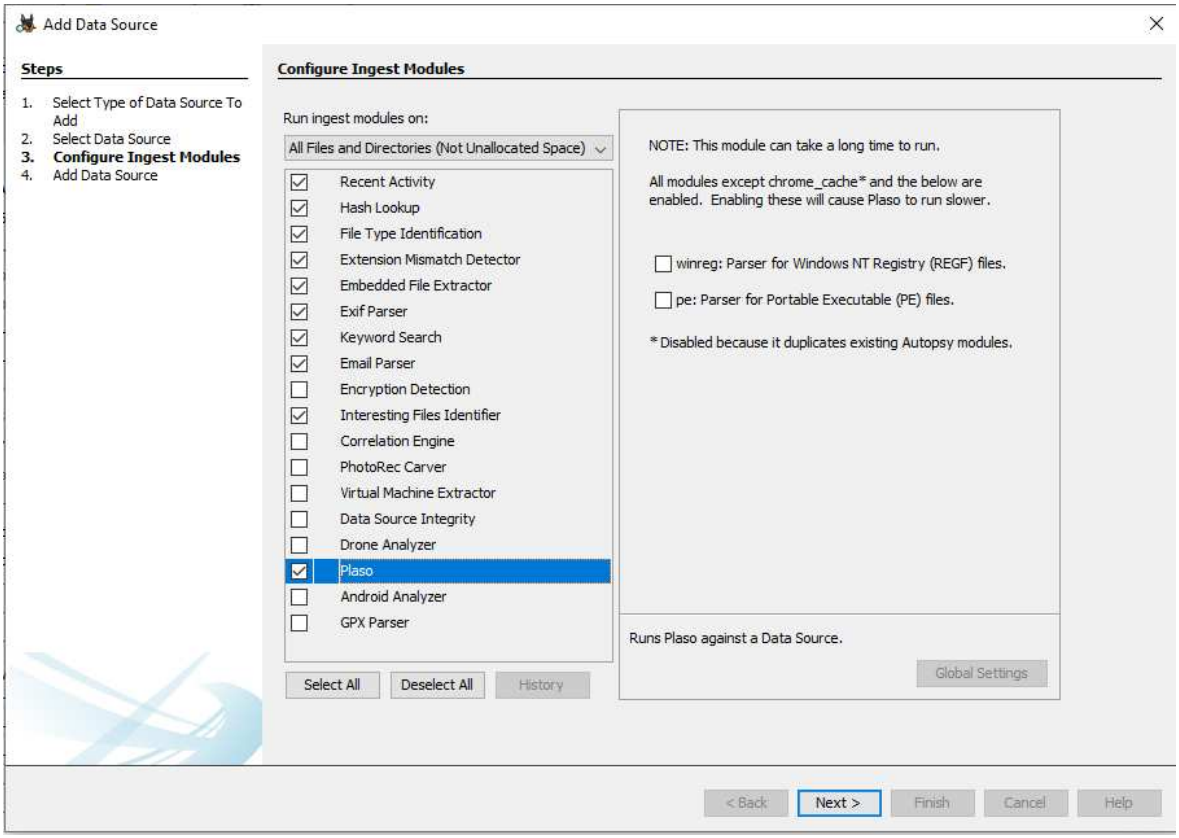


Figure 2.1: Autopsy: Ingest Modules

After adding the data source and the selected ingest modules finish their analysis, the results acquired are presented to the investigator, as shown in Figure 2.2.

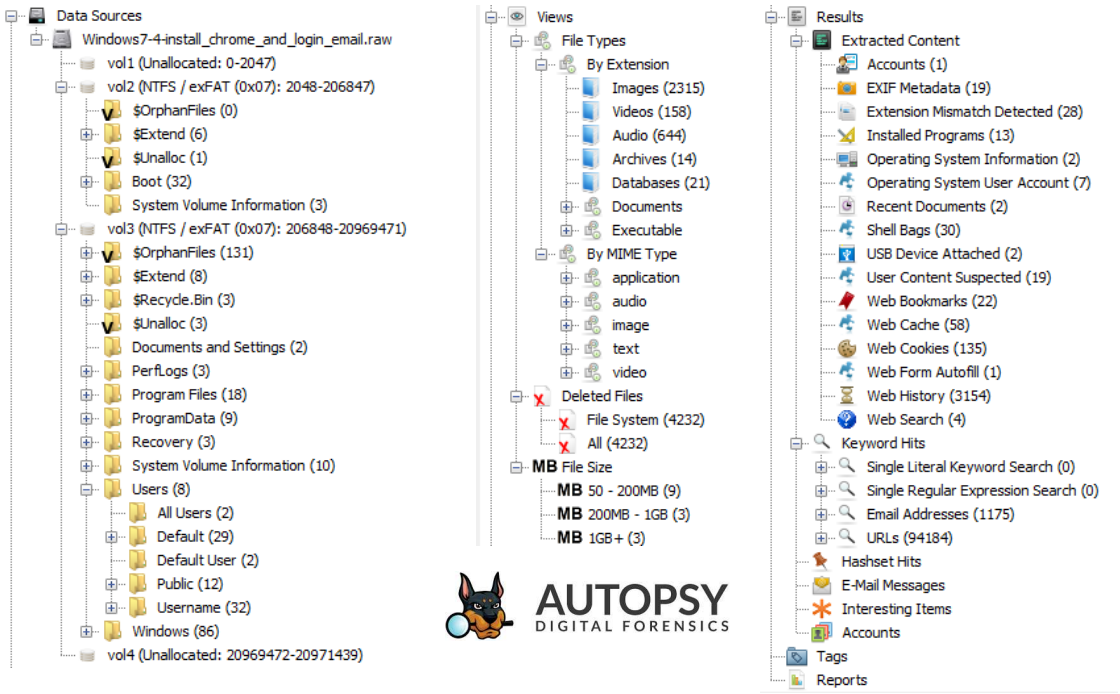


Figure 2.2: Autopsy: Data Extraction, Views and Results

The data extracted from the disk image is shown in *Data Sources* and *Views*, and the automated analysis results from the selected ingest models are in *Results*:

- *Data sources* show the files in a tree structure same as the file system. There are a number of volumes (four volumes in this disk image, the first and fourth are unallocated.).
- *Views* presents files in folders by their metadata. Files are into different categories by the extension and MIME type. Deleted files are also shown. Files are also classified according to the size.
- *Results* presents the extracted content, keyword hits, hash-set hits, etc.

In the results, there is a folder called *interesting items*. Files are shown in this folder if their hash matches the illegal files preserved in the local repository. If the central repository is applied, then the extracted data will be checked with Autopsy's preservation of the previous results, such as hashes, emails, USB device IDs, wifi SSID, ICCID, domains, etc. The tags from past encounters can be employed for deprioritising the non-pertinent benign artefacts. The topic of centralised digital evidence processing and research is discussed in Section 2.6 and Section 2.7.

A List of Tools for Different Examination Targets

As there are the many different forensic objects, a number of commercial tools and developed scripts are typically applied during an investigation. A list of common digital forensic tools designed for different types of examined targets is outlined below [69]:

- **Memory Forensics**: Volatility, WindowsSCOPE, RAM Capturer, Magnet RAM Capture, and Memoryze;
- **Mobile Device Forensics**: CelleBrite UFED, XRY, and Oxygen Forensic Suite;
- **Network Forensics**: Wireshark, Network Miner, Xplico, and E-Detective;
- **Computer Forensics**: EnCASE, FTK, Sleuth Kit Autopsy, TCT, and DEFT.

These GUI based forensic tools present extracted data in a consumable fashion and allow the searching of the artefacts by keyword or time span. However, the analysis from each tool is different and usually cannot be automatically integrated together.

2.2.6 Digital Evidence Backlogs

According to the report by *The Irish Times* in January 2020, the backlog in An Garda Siochana is 2.5 years and the need for new technologies for online child abuse investigations is severely needed . A practitioner survey conducted by Sanchez [70] in 2019 shows participants encountered limitations in their workload (20.55%), time (17.81%), and resources (10.96%); the lack of investigators, time, and advanced hardware and tools, to process and analyse data continuously increases backlogs.

All over the world, each digital forensics case can encounter efficiency problems due to the amount of data needed to be processed in the analysis stage, the amount of data needed to be stored from each acquisition, the speed at which the acquisition can be completed, and the automated of finding blacklisted files. These problems result in the backlog of cases caused by the time taken to analyse the data collected from each case. Digital evidence backlogs have been outlined as a challenge worldwide [14, 71, 72]. The average backlog in digital forensic laboratories around the world was from 6 months to 1 year in 2009 [14]. In the UK, the most severe example saw one case being delayed by more than 21 months in 2015. The reasons for backlogs are various; there are already solutions being proposed by researchers to tackle this problem.

In the not-so-distant past, most cases involving digital forensic investigation involved criminals using computers, networks or other IT infrastructure as a tool for conducting their crimes.

[2]https://www.irishtimes.com/news/crime-and-law/garda-needs-new-technology-for-online-child-abuse-investigations-1.4138583

At that time, the set of devices requiring analysis usually consisted of a single computer and the cases involving digital investigation were infrequent. Society has become increasingly reliant on a variety of digital devices. As a result, there is a massively increased need for expert digital forensic analysis across a range of cases, and a multitude of devices requiring analysis per case has become commonplace.

The factors ultimately leading to the mounting digital forensic backlog commonly encountered in law enforcement [12] include: (i) the increasing number of cases involving digital investigation; (ii) the number of digital devices requiring analysis is also increasing; (iii) the storage volume of each device is growing; (iv) the diversity of digital devices and the various form of storage formats, file systems, e.g., Internet-of-Things devices, wearables, cloud storage, etc. Each of these introduces additional complexity to the digital forensic process.

2.3 Digital Forensic Test Images

The generation and maintenance of sufficiently detailed and documented test datasets is one of the main challenges of testing in digital forensics [73]. Grajeda et al. [74] provides an overview of publicly available datasets for researchers in digital forensics, and outline the reasons why some researchers prefer not to share their datasets. The case for standardised corpora is made by Garfinkel et al. [75], with the primary motivations being reproducibility and education. Hard disk image from many *Digital Corpora* is no longer available with respect to the General Data Protection Regulation (GDPR)[3] legislation. As a result, a common alternative approach is for educators and tool validators to spend significant time creating customised data sets.

Al Fahdi et al. [76] note that the public availability of forensic cases is "very limited". In their work, they make use of two publicly-available cases. The first is "Hunter XP", which provided as a training case for the EnCase digital investigation product. The other was a simulated hacking case that was artificially generated by National Institute of Standards and Technology (NIST) as part of the CFReDS project[4]. They also gained access to two further cases privately, which required non-disclosure agreements to be signed. This emphasises the level of difficulty associated with obtaining realistic cases for analysis and distribution for educational purposes.

As noted by Woods et al. [77], the small number of available corpora means that solutions to standard datasets are frequently available online, potentially undermining the effectiveness of assessments and proficiency testing. In these scenarios, it is desirable that new, unseen challenges be posed to ensure the integrity of the process.

[3] https://gdpr-info.eu/

[4] `http://www.cfreds.nist.gov`

2.3.1 Current Available Disk Images

The problem of providing realistic data for digital forensics education has resulted in a number of techniques being employed by the educator. Moch et al. [78] outline three existing approaches to the creation or acquisition of digital forensic datasets or viable disk images:

- Perhaps the most widespread method is the manual creation of disk images. Here, an instructor creates a disk image that contains specific evidence for students to find. This has the advantage that the precise evidence is known to the instructor and can be used for evaluation purposes. Additionally, there is no requirement to wait for an interesting activity to occur in a natural setting, as the instructor is free to perform/emulate any actions that are desired. However, creating these images is a very time-consuming task, particularly given the requirement to ideally provide realistic wear and depth.
- A *honeypot* involves connecting a computer to a network with the express intention of it being attacked and compromised [79]. By recording the activities of attackers, interesting disk images can be created. However, the majority of attacks are automated, and the quantity of images that feature manual attacks for students to study is low. Due to the low quantity of interesting examples available, analysis results can frequently be found online. From a suitability standpoint, the required analysis of these honeypot generated challenges is often at too high a difficulty level for many learners[77].
- A fruitful source of realistic data is *second-hand hard disks*. This approach results in valuable data on naturally occurring phenomena on disks, as the disks have typically been in use by a real user over a longer period of time than an instructor can dedicate to the manual creation of an image [80]. Pre-used hard disks are the source of the Real Data Corpus, assembled over a number of years by Garfinkel [81, 82]. This forms part of a 30TB collection of research corpora, which also includes items such as network packet traces, known malware and a million document corpus gathered from the `*.gov` TLD. One drawback of this approach is that it does not include materials relating to real crimes that could be used for training purposes [83]. Additionally, the use of data belonging to real users raises a number of legal data protection issues. As the data is generated by real users, privacy law (which greatly varies by jurisdiction) must be taken into account, particularly when redistributing images. Images may also contain copyrighted materials (including the operating system and software) or illegal files.

2.3.2 Automated Disk Image Generation Approach

In 2009, Moch et al. [78] discusses the development of *Forensig2*, which is subsequently further evaluated in [84] in 2011. *Forensig2* allows automated artefact generation of both the hardware and software level, i.e., it can programmatically configure the virtual machine

in *QEMU*[5], including the CPU, network, disks, etc. It can also carry out actions within the VM, including configuring partitions, copying files locally and remotely, and also various other operating system level actions (on Linux), such as installing software (using the command line). This work introduces many of the concepts to be taken forward in any automated disk image generator, i.e., logging of actions performed to use as ground truth, the need for extensibility, and the concept of reproducible randomness. However, it is difficult to see how the approach could realistically synthesise all of the actions on a Windows system caused by for example a user opening a file, which includes numerous registry artefacts, link file creation, jumplist entries, potentially associated browser artefacts, etc. This limited ability to generate artefacts for a GUI based OS (Operating System) is acknowledged by Moch et al. [84].

Yannikos et al. [85] present 'model-based generation of disk images', which focuses on creating a formal model of the scenario to be built. Practically, the actions that were achieved were: creating file systems, creating and deleting files, writing raw data to a disk, downloading a file from the Internet, and disk image import/export. This would be very effective for file system level interactions, but to generate a realistic disk image that could be used to teach forensic investigation techniques at all levels of abstraction, operating system-level artefacts would also be needed and far higher-level operations would need to be emulated.

2.4 Digital Forensic Process Model

The first digital forensic process model proposed contains four steps: Acquisition, Identification, Evaluation and Admission. Since then, numerous process models have been proposed to explain the steps of identifying, acquiring, analysing, storage, and reporting on the evidence obtained from various digital devices. In recent years, an increasing number of more sophisticated process models have been proposed. These models attempt to speed up the entire investigative process or solve various problems commonly encountered in the forensic investigation.

The diversity of devices and sources of digital evidence results in a corresponding diversity in digital forensic process models [8]. There is no single, universal process model suitable for all types of investigation. Reducing the volume of data for arduous, manual analysis will speed up the entire investigative workflow and can significantly aid in alleviating the digital forensic backlogs all too common in law enforcement agencies throughout the world [38].

Even though digital forensics is a relatively new research area, it has already made significant progress. The progress is not only from a technology perspective, such as tools to collect and analyze digital evidence but also with the improvement of methodology. In digital forensics, a process model is a methodology to conduct an investigation; a framework with a number of

[5]QEMU is a generic and open source machine emulator and virtualiser. https://www.qemu.org/

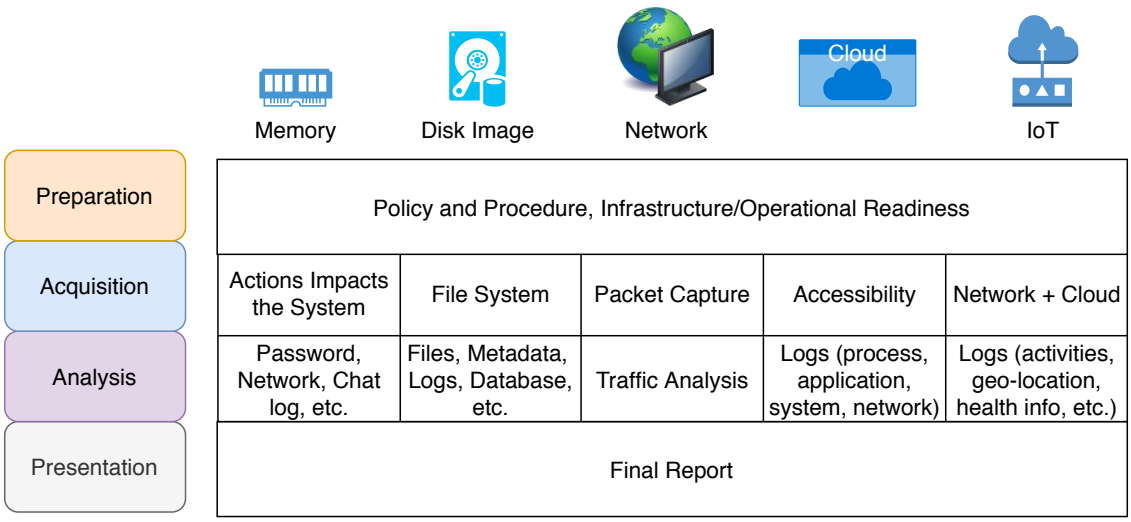


Figure 2.3: Different Types of Digital Forensics

phases to guide an investigation. Generally, process models were proposed on the experience of previous work. Due to the variety of cases, e.g., cyberattacks conducted by IT specialists, civil cases in a corporation, or criminal cases, different investigators tend to follow different methods in their investigative process, there is no standard workflow in digital forensic investigation.

A standard methodology in digital forensics investigation consists of a definition of the sequence of actions necessary in the investigation. A framework, if it is too simplistic or has fewer phases, might not provide much guidance to the investigation process. A framework with more phases and each phase with sub-steps, with more limitation of its usage scenario, may prove more useful. Even though it is almost impossible to design a perfect process model that is able to deal with any investigation, an ideal framework should be general, which means that it could be applied to as many cases as possible. Furthermore, considering that techniques evolve so fast, a well-defined framework should also with the capability to adopt new techniques in the process of investigation.

Numerous process models have been proposed in the current literature. Generally, each framework attempts to provide a methodology for a specific area and these process models have broadly similar approaches. The earlier research is more concentrated on defining the whole process of digital forensic investigation, significant process models had been listed and discussed in the paper [86]. More recently, process model research centres around solving more specific issues in particular cases or focus on a single step (evidence collection, preservation or examination, analysis). The triage model [41, 87] is effective for cases that are time-sensitive. By employing digital forensics triage, investigators could discover pertinent evidence and the police could get leads about the criminal sooner instead of having to wait for the whole report which could take weeks months or longer.

2.4.1 Digital Forensic Framework in Initial Phase

At the turn of the century, it was still the early days of research on digital forensics and digital forensic process models. Initially, one of the most urgent issues in digital forensics was to define a process model to make the entire investigative process consistent and standardised. A number of general digital forensic processing models have been defined. Most of these frameworks define a group of necessary steps in a whole investigation process, and the models were refined over time. The later models improve upon the former ones by including some additional steps or defining sub-steps of the process models - making each step more precisely defined.

The traditional framework had been refined and formed a number of novel frameworks. Some inheritance relation among the existing frameworks listed below (see Figure 2.4):

- DFRWS model (Palmer et al. 2001) → SRDFIM (Agarwal et al. 2011)
- DFRWS model (Palmer et al. 2001)
  → An Abstract Digital Forensics Model (Reith et al. 2002)
- IDIP (Carrier et al. 2003) and DCSA (Rogers 2006)
  → Triage Peocess Model CFFTPM (Rogers et al. 2006)
- Integrated Digital Investigation Process (IDIP) (Carrier and Spafford 2004)
  → Enhanced Integrated Digital Investigation Process(EIDIP)
  (Baryamureeba and Tushabe 2004) [88]
- Integrated Digital Forensic Process Model (Kohn et al. 2013)
  → DFaaS Process Model (van Baar et al. 2014)

The focus of these models is on what phases should be included in the investigation, the sequence of each phase and the definition of key concepts. The fundamental questions were answered by a range of research on digital forensics process models by Palmer et al. [89], Lee et al. [90], Reith et al. [91], Baryamureeba et al. [88], Beebeet et al. [92].

Lee et al. [90] proposed a Scientific Crime Scene Investigation (SCSI) model for digital forensic investigation in 2001. Ciardhuáin [93] criticises SCSI model is not a systematic digital forensic process model, because it only focuses on physical crime scene investigation and lack of describing on digital criminal scene investigation. Kohn et al. [86] explained that the physical crime scene investigation process can be adapted to digital crime scene investigation. An Event-based Digital Forensic Investigation Framework separates the concepts of the physical crime scene and the digital crime scene, collecting digital devices from the physical crime scene and then obtaining digital evidence from the digital devices' storage [94] in 2004. An Enhanced Integrated Digital Investigation Process (EIDIP) model was proposed by Baryamureeba and Tushabe in 2004 [88]. EIDIP model is based on IDIP, by introducing a traceback phase to address the problem of having to reconstruct twice in IDIP.

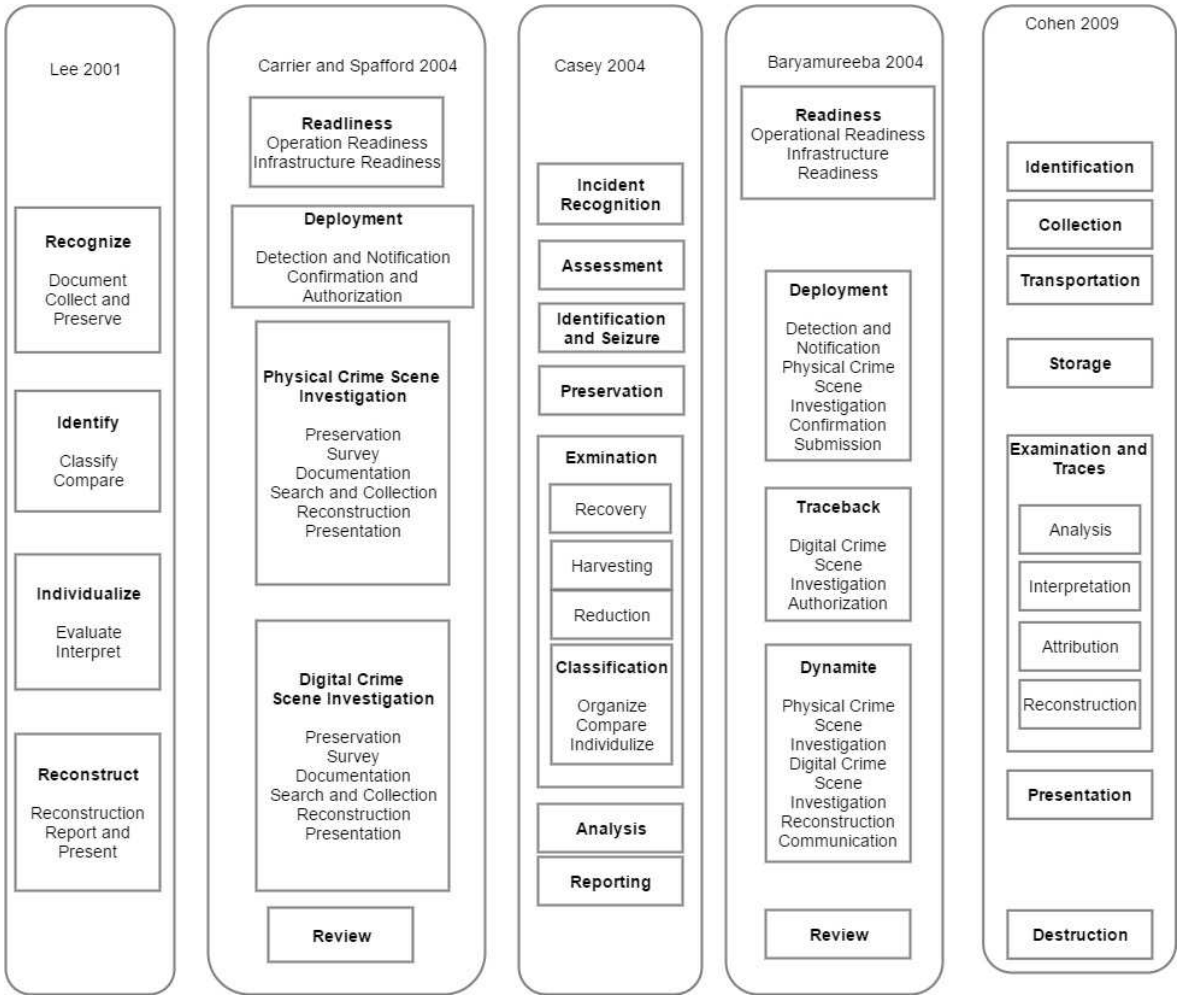


Figure 2.4: Digital Forensic Framework in Initial Phase[a]

[a]Du, X., and Scanlon, M. (2017). Evaluation of Digital Forensic Process Models with Respect to Digital Forensics as a Service, 16th European Conference on Cyber Warfare and Security (ECCWS), Dublin, Ireland.

2.4.2 Refined Digital Forensic Process Models

Merely following a general process model is often not specific enough to handle the broad range of cases typically encountered by law enforcement. The criminal could be an IT specialist and conduct advanced cybercrimes, CCTV cameras' storage may need to be analysed, or data leakage in a corporation, etc. All of these different situations could require bespoke methodologies.

After the general process procedure was clearly defined, researchers started working on specific issues that are more detailed. For example, 1) refining a process model by making an improvement at a specific step of the investigation; 2) dealing only with a specific category of cases, such as network forensics, mobile devices forensics, etc.; 3) Triage models [87, 41] outline specific processes for time-sensitive cases, such as child abductions, missing person cases, etc.

The phases and subphases of each process model are shown in Figure 2.5 below:

- Extended Model of Cybercrime Investigation - In 2004, a number of process models had already been defined. However, each did not include a significant aspect of cybercrime investigation itself. An extended model of cybercrime investigation was proposed [93]. In general, it is waterfall fashion and activities are conducted in sequence. It still allows iteration in some part of the investigation, for example, the iterative process of "examination - hypothesis - presentation - proof/defence".

- Digital Forensic Triage Process Model - In some special cases, such as kidnaps and hostage rescue, acquiring clues from digital devices immediately is crucial, or some other cases such as robbery, crucial information is required as soon as possible to increase the likelihood of catching the criminal before they have escaped to another country. Often traditional models are insufficient for this use case - potentially taking weeks or months to get results. Tiered models are designed to expedite situations like this. Considering traditional models are designed to guide the entire investigation, a triage process model was proposed to deal with time-sensitive cases [87]. This model focuses on the crucial first few hours of an investigation.

- Digital Forensic Model Based on Malaysian Investigation Process - This model is notable in that it is focused on the data acquisition process, including more detailed handling on live data acquisition and static data acquisition in cybercrime investigation (Perumal 2009) [95].

- The Systematic Digital Forensics Investigation Model - This model is focused on computer frauds and cyber-crimes, which is helpful in evidence dynamics and reconstruction (Agarwal et al. 2011) [35].

- Integrated Digital Forensic Process Model - This model is the most recent proposed process model which including a relative generally digital forensic investigation (Kohn et al. 2013) [86].

2.4.3 Recent Digital Forensic Models for Handling Modern Advancements

Some new technologies result in new problems hindering digital forensics investigation. Cloud computing makes evidence collection more difficult; Internet of Things adds a variety of new device and storage forms; more digital devices connected into the Internet result in an ever-increasing volume of data. In recent years, research on process models is more focused on integrating other technologies, such as data mining, to support the original models, or propose novel process models to solve the issues caused by these new technologies.

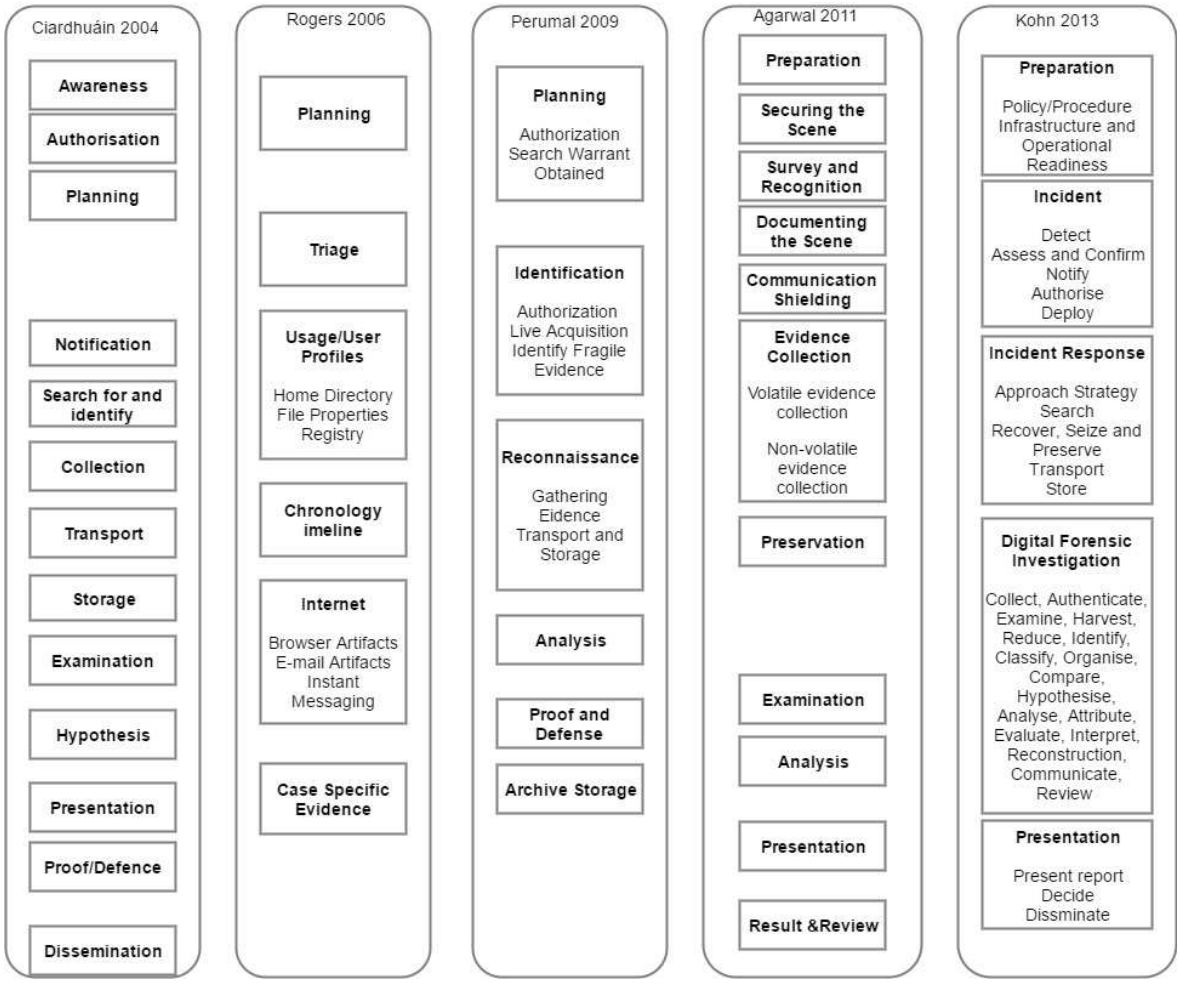


Figure 2.5: Digital Forensics Frameworks Focusing on a Specific Use Cases[a]

[a]Du, X., and Scanlon, M. (2017). Evaluation of Digital Forensic Process Models with Respect to Digital Forensics as a Service, 16th European Conference on Cyber Warfare and Security (ECCWS), Dublin, Ireland.

Some recent models, as outlined in Figure 2.6, include:

- An integrated conceptual digital forensic framework for cloud computing by Martini et al. [96].
- Data reduction and data mining framework by Quick et al. [97].
- IoT Based Digital Forensic Model by Perumal et al. [95].

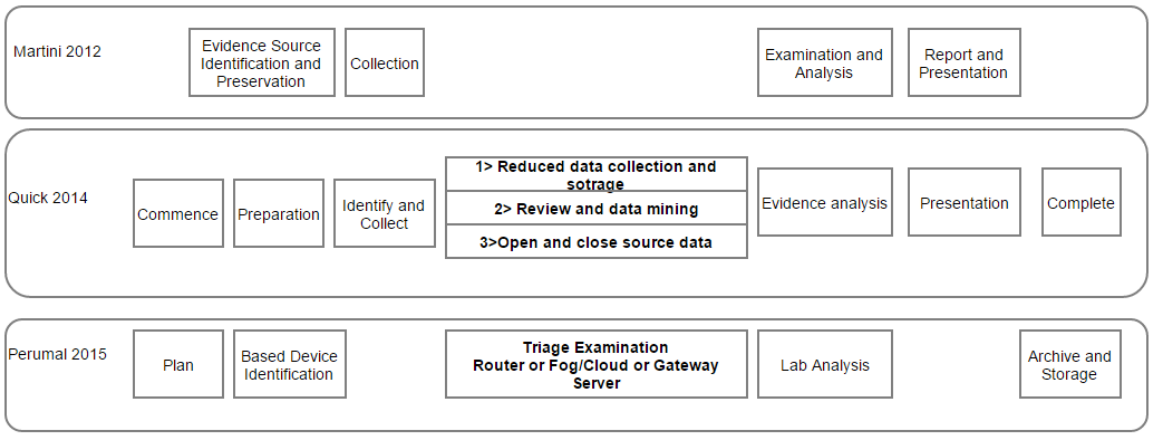


Figure 2.6: Recent Digital Forensic Models for Handling Modern Advancements[a]

[a]Du, X., and Scanlon, M. (2017). Evaluation of Digital Forensic Process Models with Respect to Digital Forensics as a Service, 16th European Conference on Cyber Warfare and Security (ECCWS), Dublin, Ireland.

2.5 Triage Process Model

With the proliferation of digital evidence, the data volumes encountered in investigations is a significant challenge faced by law enforcement agencies. An efficient way is processing the significant relevant evidence first. The triage process model has been proposed to tackle the most critical information first to help acquire the most valuable trace of time-sensitive cases [87]. It allows to timely identification, analysis, and interpretation of digital evidence. Currently, the prioritisation of device acquisition and processing at a crime scene is determined by the investigative officer. As more Artificial intelligence (AI) based techniques are developed, on-scene preliminary inspections could quickly focus the analysis towards the devices most likely to contain case-progressing information first.

Triage methods and tools can help address the issue that the proliferation of digital-based evidence. Hitchcock et al. [41] presented their work on the training of front-line personnel in the field triage process. By employing digital forensics triage, investigators could discover pertinent evidence and the police could get traces or clues about the criminal sooner instead of having to wait for the whole report which could take several months or even years. An enhanced triage process model outlines that training digital forensic first responders to work at the crime scene can solve the problem of the shortage of digital forensic specialists in law enforcement [41].

Triage is essential, especially for mobile devices, because it carries more and more intimate information than ever. Mislan et al. [98] states the requirements for on-scene triage tools, includes user-friendly design, warning before taking any action, accuracy, accessible data format, etc.

2.5.1 Digital Device Triage Tools

For time-sensitive cases, pertinent information acquired from digital forensics has its greatest value at the earliest stage of the investigation. Triage is a process whereby devices and artefacts are ranked in terms of importance or priority [87]. Much work has been done in the area of digital forensic triage in an effort to improve the overall process [70]. A digital forensic triage process model was proposed to use during the investigation by Rogers [87]. The importance of files varies in different types of the case; CSEM, drug activity, financial crimes, etc. The approach for triage usually stems from practical experience.

The triage process usually happens after a quick, preliminary analysis of devices at the crime scene, then a more in-depth analysis is performed in the digital forensic laboratory to identify more relevant evidence. When multiple devices are involved in an investigation, triage can reduce the workload. Prioritisation of devices to be examined is defined as a sub-phase in the "Behavioural Digital Forensics Model", proposed in 2018 [99].

A variety of triage tools has been designed and implemented. Tools for devices triage could using the information about user profiling [43], or synchronisation records between devices [100]; some other research focus on triage on mobile devices [58, 58, 101].

Grillo et al. [43] propose a methodology and a tool to support fast computer user profiling building a wider view for investigators to prioritise seized hard drives. Data extracted from devices for user profiling are applications installed, configurations of the machine, percentage statistic of the type of files found on the disk and so on. Device users are categorised as: occasional user, chat-internet user, office worker user, experienced user, hacker user.

Traditional triage approaches require access to each seized device to perform a general examination to decide whether the device is high or low priority. Hargreave et al. [100] proposed an approach that exploits synchronisation features to determine if a device potentially contains useful information. This can help avoid the requirement for access and acquisition of all encountered devices. The authors present a summary of artefacts that can be used to extract synchronisation information, such as Browsers, Communication Apps, Social Networking, Media/Video, Note-taking apps, Photos and Cloud Storage. The developed tool, *SyncTriage*, creates an output form displaying information about the discovered devices, such as name, make, model, and operating system, etc. Besides, a "universal timeline" is created combining events from all extrapolated devices for device correlation at a particular time.

With the increasing significance of mobile device forensics, Marturana et al. [58] proposed an approach for device prioritisation leveraging data mining and machine learning theory. This work presents the result of a study concerning mobile phone classification in a real child abuse investigation case. The features used consisted of the phone model (GSM or smartphone), phone contacts, calls made, text messages sent/received/read, number of video/audio/photo files, URL, email and memos. The experimentation tested the performance on the feature value represented as numeric (a number) and category (the number is low, medium or high).

In subsequent work, Marturana et al. [31] expanded the triage approach to detect the device's relative importance using features from 1) the timeline of events, 2) the crime's specific features, and 3) the suspect's private sphere (habits, skills and interests). The experimentation in this work was conducted on a copyright infringement and a CSEM exchange case. The dataset applied consisted of 23 cell phones for the CSEM case with 13 digital media files and 45 copyright infringement-related features. A result of 99% correctly classified samples on both cases was achieved.

McClelland et al. [101] focused on improving the classification accuracy of the previously proposed device classification approach. This approach improves the classification performance through feature manipulation techniques such as feature weighting and feature reduction. The experimental data is from digital devices already examined by the Italian Postal and Communication Police [58, 31], and the M57-Patents corpus[6] – based on storage drives purchased on the secondary market.

2.6 Cloud-based Digital Forensic Framework

In the last decade, cloud computing has emerged as a disruptive technological concept, and most leading enterprises such as IBM, Amazon, Google, and Microsoft have set up their own cloud-based services. In the field of digital forensic investigation, moving to a cloud-based evidence processing model would be extremely beneficial and preliminary attempts have been made in its implementation. Moving towards a DFaaS model would not only expedite the investigative process but can also result in significant cost savings – freeing up digital forensic experts and law enforcement personnel to progress their caseload.

Even though cloud computing has become prevalent across many industries, there is limited literature on its use and advantages from a DFaaS perspective: Lee et al. [21]; van Baar et al. [102]; Wen et al. [103]. In this section, the current research on DFaaS will be discussed.

2.6.1 DFaaS Framework

Different cloud computing service models - IaaS, SaaS, PaaS - offer various service for its user. DFaaS can be described as an approach applying cloud service model to forensic investigation to build a centralised evidence process system.

Cloud-based digital forensics is still a relatively new approach while it could have huge potential to improve the efficiency of digital forensic investigations. The concept "Forensic Cloud", a work environment for investigators without special forensic tools knowledge, has

[6] http://digitalcorpora.org/corpora/scenarios/m57-patents-scenario

been proposed by Lee and Un [104] in 2011. And in 2012, these authors have implemented a cloud-based service for index search [21].

The first utilisation is the computing power provided by distributed computing, which can better handle the increasing magnitude of data. Lee et al. [21] show the efficiency of the cloud system working on the indexed search. Wen et al. [103] outline an implementation of a cloud-based system to combat the magnitude of data encountered by digital forensics by leveraging parallel computing. This work highlights the applicability of cloud computing in digital forensics and the improvement that DFaaS could make. One use case of DFaaS is to offer indexed search as a service [21]. Concerning the large volume of data needing to be analysed, distributed computing systems could do the same work in parallel. Such a cloud server can offer highly intensive computing process and a large quantity of storage to deal with the slow processing on big data volume. In their paper, Lee et al. [21] outline a case study that indexed search as a service.

In 2013, Wen et al. [103] designed a cloud-based framework, which deals with a large volume of forensic data, sharing interoperable forensic software, and providing tools for forensic investigators to create and customise forensics data processing workflows. After a series of tests, the experimental results show that the proposed workflow management solution can save up to 87% of analysis time in the tested scenarios. In this framework, the main purpose of making use of cloud systems to deal with the large volume of evidence data through distributed parallelisation.

DFaaS offers services to the user, which not only makes the evidence process easier but also improves the overall case efficiency. A centralised digital forensic service offers significant benefits: 1) facilitates remote evidence acquisition, 2) reduces IT skill requirements for investigators, 3) improves collaboration efficiency between detectives and investigators, 4) enables easier cross-device or cross-case examination.

Conducting big dataset evidence collection and processing requires extremely costly local infrastructures, a cloud-based distributed evidence collection and analysing system can be cost-effective and easily scalable [105]. A Hadoop Distributed File System (HDFS) and cloud-based conceptual model to support reliable forensics investigation on big data [105]. In this framework, a deduplication layer is used to remove redundant data from the incoming data stream.

2.6.2 HANSKEN: DFaaS System Used by NFI

A DFaaS system provides a new method to improve the investigation process, and at the same time, it can be combined with the traditional process model. One implementation with complete service and user interface is Xiraf (its successor is named HANSKEN), has been built and used by Netherlands Forensics Institute (NFI) [102, 106]. The Xiraf system is implemented based on a model proposed by Kohn et al. in 2013 [86]. It follows the process

of the forensic investigation but improves the efficiency to process more data volume.

In 2014, van Baar et al. [102] outlined HANSKEN. It focused on a comparison between the DFaaS framework with the traditional models and list the problems in traditional methodology while outlining how their DFaaS implementation has addressed some of these issues. This work proves the viability and impact cloud-based digital forensic solutions can have on the entire process.

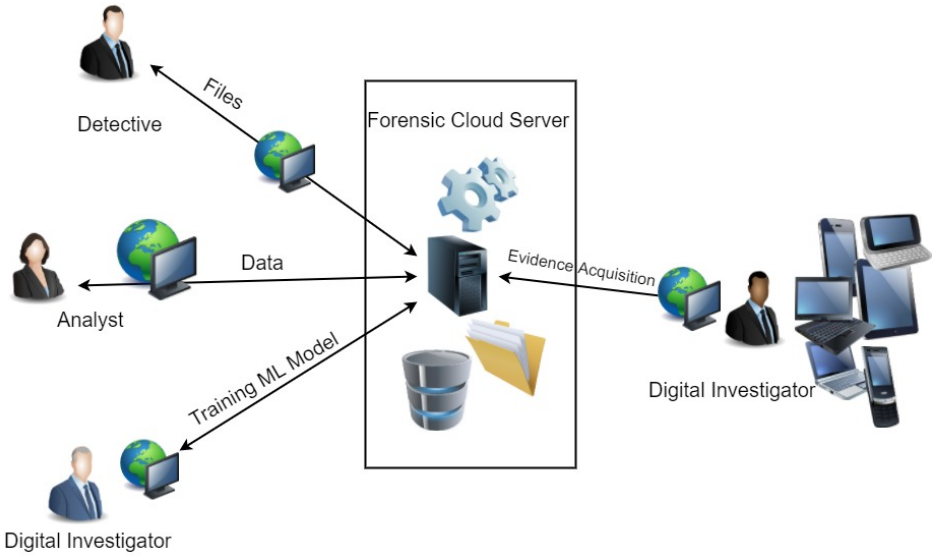


Figure 2.7: Digital Forensic as a Service[a]

[a]Du, X., and Scanlon, M. (2017). Expediting the Digital Forensic Process through a Deduplicated Framework, 16th European Conference on Cyber Warfare and Security (ECCWS), Dublin, Ireland.

2.6.3 Benefits and Advantages of DFaaS

In the big forensic data age, DFaaS systems are proposed to improve the efficiency of the investigative process at the framework level. The growing volume of data results in an increased time needed for each step of a typical digital forensic investigation. Leveraging cloud computing with its significant computing resources would be one obvious solution to this issue. A centralised data storage server could expedite the process of evidence collection and analysis [38]. In addition, a cloud-based digital forensics environment could enable case detectives to directly connect and perform preliminary analysis themselves in a controlled environment without waiting for expert analysis. In this triage model, DFaaS facilitates the investigators preserve and analyse digital evidence on scene by connecting to the server remotely. The management of forensics environment would still ultimately be handled by digital forensic specialists. A broadly applicable framework that can deal with numerous existing situations encountered in digital forensics, while being extensible to handle new technologies has always been desirable. DFaaS enables this to be possible. DFaaS not only benefits from the processing power cloud computing provides but can also influence the future development of digital forensic science – opening up new possibilities for collaborative

investigation. The evidence from cases could be stored into the cloud-based system, making more intelligent forensic processing possible [97]. New tools and techniques developed in the field of structured and unstructured data science could also be easily integrated to further expedite the process.

2.7 Data Deduplication and Data Reduction

During an investigation, the acquired file artefacts usually are categorised by file types, images, documents, etc. Data reduction is usually as a step of a typical digital forensic process model. For example, in the most recent proposed digital forensic framework Integrated Digital Forensic Process Model (IDFPM) proposed by Kohn et.al. [86].

The hash of each file artefact is calculated for the detection of known files. National Software Reference Library **NSRL**[7] is a reference library available for filtering out the boring files (operating system files, application files, etc.) which consist of nothing valuable information for the case through comparing the digital fingerprint (hash value).

2.7.1 Data Deduplication Technology

As the growth of data continues more and more techniques must be created to allow for the storage of such high volumes of data. Many large cloud storage providers have already incorporated such techniques, one of which is data deduplication. Data deduplication is an intelligent compression technique used to optimise data storage where many duplicates can occur. This is a quite useful technique when it comes to large server storage with a high chance of duplicates due to its reduction of redundant data. There are two different types of data deduplication considered, block-level deduplication and file-level deduplication. Before storing a file, a database of files is referenced to see if this file or file block has been seen before if so the metadata is stored along with the pointer to the unique file or file block copy and the file itself is not stored again.

File-level deduplication or a single instance datastore is the costliest type of deduplication space-wise. This compares entire files, uploading those that differ in any way. Alternatively, block-level or sub-file deduplication only stores the repeated blocks within files once. The reduction of redundant data by using block-level is greater than that of file-level as only parts of each file may be uploaded rather than the entirety of each file. Each type has its advantages and disadvantages within its context of use.

The ability to store larger volumes of data in smaller physical disk space allows people to store

[7] https://www.nist.gov/software-quality-group/national-software-reference-library-nsrl

larger amounts of data much more conveniently. This trend is not expected to slow down any time soon and according to discussion on future storage devices by Kryder and Kim [107], it may continue to accelerate faster than expected. Data deduplication techniques have been widely used by the world's largest IT corporations in a number of different products and systems mainly for the purpose of improving storage utilisation, thus reducing network data transfer.

2.7.2 Data Deduplication in Digital Forensics

Data deduplication techniques can reduce the amount of data to be analysed. Data deduplication is an intelligent compression technique used to optimise data storage where many duplicates can occur. Applying data deduplication technique to digital forensic can eliminate the repeat work during the investigation, such as processing of the operating system files, application files, or event the illegal files analysed in the past case.

The phases of the generic digital forensics process model that includes identification, collection, analysis and examination, and opinion/report [108]. Approaches to improve the efficiency of the digital forensic process was constantly discussed. One reason leading to the digital evidence processing very slow is the repeat and manual work waste lots of time of the investigators [38]. A digital evidence processing system which eliminates the repeat and reduces the manual work during the digital forensic investigation.

Data deduplication can be applied by comparing the digital fingerprint of each file to a known file database, such as NSRL. Applying data deduplication to digital forensic acquisition has been proposed since 2009 [109].

The difference between block-level and file level, in the context of a DFaaS model, is the amount of client-server interaction. File-level deduplication checks a database for every file, whereas block-level hashes each block and then check them against a database. Depending on the file system, this could be every 512 bytes.

An improved data reduction method for digital forensic processing was proposed by Neuner et al. [110] to promote the efficiency and effectiveness of an investigation. By employing data deduplication techniques, a known file database preserving the analysis result can be used for white/blacklisting saving the time of repeating analysis of file artefacts. However, a forensically sound copy of disk drive is required to prove the digital evidence is valid and acceptable to the court.

Forensic Soundness

Data deduplication and reduction have to consider forensic soundness, and this processing can only happen after a sound disk image acquisition. Casey stated that digital evidence must be preserved and examined in a forensically sound manner in order to be useful in an

investigation [40]. In the context of disk imaging, digital forensic professionals qualify the term by stating that to be forensically sound, the disk image must be a bit-for-bit copy of the original (i.e., an exact copy) [25]. Physical level bit-for-bit copy of disk enables to collect data out of the file system, i.e. deleted files, block slack, etc. Hash digestion of the disk is used to check the integrity of the acquired data. It is also used to determine if digital evidence has been modified or tampered with by comparing it with an original copy. The integrity of digital evidence is important so as to ensure it is "legally acceptable". Furthermore, a forensically sound disk image also allows the reproducing of the analysis process. For sound image acquisition, data deduplication should be conducted after a whole disk image is collected.

Watkins et al. [109] explain the core concept through a project called Teleporter. The idea behind Teleporter is to reduce the amount of data sent to remote storage by checking it against data that has been found before. The reason this is done is because of the amount of data that is common across most machines, e.g., operating system files. Rsync and LBFS were two projects that influenced Teleporter. Rsync breaks files up into blocks and checks the blocks against the files that already exist on the server-side based on their hash. LBFS is similar in its use of hash but it enables low latency to files and applications over low bandwidth connections. The tests showed that 50% - 70% fewer data had to be sent to the server. An analytically sound concept was proposed by Watkins. Watkins et al. [109] presents an approach acquiring an analytically sound copy during a deduplicated digital evidence transmission.

Proven Effectiveness

Neuner et al. [110] expand the effect of the data deduplication in the digital forensic process. Analysis results can be preserved, so that repeated analysis work is eliminated through the creation of a white/blacklist. However, the forensically sound requirement was not dealt with in this work.

Within the field of digital forensics, there is a significant amount of concern placed on having to store the large amounts of data collected with each acquisition[12, 111, 39]. The increase in data directly affects the time taken to both acquire and to analyse the data. If this problem continues to be left unaddressed it may lead to serious problems in the future. This leads to a requirement for techniques to reduce this amount of data within every acquisition. Based on the testing result from Neuner et al., the storage requirement in realistic scenarios can be decreased by 78%.

2.7.3 Data Reduction Approaches

Data reduction is a process of "finding useful features to represent the data depending on the goal of the task [112]. Data reduction is necessary, as some type of file artefacts potentially

contains more information related to the investigation, such as Internet history, user-created, emails, documents, pictures, audio and video. For a specific case type, the usefulness is different. For instance, in a child abuse investigation, all videos and images are typically extracted during the examination and then are analysed to determine those relevant to the case. In an intrusion investigation, all host interactions produced are analysed to determine which are relevant to the incident.

The selection of those file artefacts most related to the investigation is data reduction. An approach was proposed for digital forensic data reduction by selective imaging (DRbSI) by Quick et al. [24] for big forensic data reduction. DRbSI enables to reduce [24]: 1)the number of file artefacts: data subset reduction; 2) the size of file artefacts: enable to review thumbnail video files at a far greater speed than when reviewing the original video files.

2.8 Automated Digital Forensic Analysis

Automation in digital forensics focused on improving investigative and provides efficiency benefits of saving both cost and human effort. The automation of data acquisition and data extraction is commonly used. This subsection presents the state of the art on automated analysis approaches.

2.8.1 Challenges of Automation in Digital Forensics

In the past, digital forensic tools were mostly designed to extract data from the acquired image, carving the deleted files, searching files through extension, name, and so on. These tools assist the investigator in identifying useful information from raw data. In recent years, automatic digital forensic investigation is often discussed as a technique for law enforcement to achieve more with existing resources.

Up to now, the influence of automatic tools to forensic investigations is limited, as most evidence processing still requires expert human input. Applying automation to digital forensic investigation brings up challenges. The technological issue mainly comes from the diverse new device types and software platforms being used [15]. From the perspective of the political and social implications, automation could deteriorate the quality of the investigation and the knowledge of forensics experts [71]. Current tools attempt to convert binary data to human consumable information, then conclusions are drawn manually.

Digital forensic researchers are trying to build tools to analyse digital evidence automatically, instead of manually repeating the same operations on each device. Automation can be an appropriate way to combat the backlog, but still faces challenges from technical implemen-

tation and being accepted by a court of law [71]. Casey [113] states that in terms of digital investigation too little knowledge is a dangerous thing.

2.8.2 Metadata and Timeline Analysis

In the analysis phase of a digital investigation, standard questions asked by the investigator include when, what, why, how? File system metadata records the most recent file actions, i.e., creation, access, and modification dates. Digital investigation looks to acquire pertinent information available on the system, from metadata and from timeline analysis to identify items of significant forensic value [114].

File system metadata including file size, file path, file name, etc., are usually used for filtering and indexing files in the examination stage of the investigation. File type filtering can allow investigators to conduct data reduction. Directory metadata is used to find out the association between files, e.g., temporal association, spatial association, etc. [115].

Raghavan [33] outlined that digital evidence encompasses: 1) User data, which directly created or modified or accessed by the user; 2) Metadata associated with user data, which provides the context of how, when, who and in what form the user data was created or modified or accessed; 3) Activity logs, which records of user activity by a system or application or both; 4) System logs pertain to variations in system behaviour from the normal based on one or more actions conducted by the users. Event reconstruction is an essential step for investigators to understand the evidence.

OS and application log files record the user's actions on a device. Data extracted from these log files enable the generation of a timeline. Timeline visualisation can prove helpful for digital forensic investigation [116]. However, due to the typically large number of digital events extracted from a disk image, visualisation can often prove counterproductive in identifying pertinent events. As a result of each user action potentially generating several digital events on an abstracted level, the number of timeline events is often too large for manual analysis. Millions of low-level events are difficult to contextualise by investigators attempting to figure out the story on the device. Hargreaves et al. [117] outlined an approach for automatically generating higher-level events, which greatly reduces their number and makes it significantly easier to be understood.

Neuner et al. [110] discusses how techniques such as paralleled or automated approaches, file whitelisting, etc., could be used to reduce back-end storage requirements. In their proposed framework, data has been reduced through file hashing at first and then by using (i) similarity hashing, (ii) whitelisting (using a reference data set (RDS) from NSRL) and (iii) cross-device deduplication to get a reduced copy for analysis. The result of the evaluation states the total reduction can be as much as 78% less than the full copy.

2.8.3 Plaso/Log2timeline

The length of log files varies, as determined by the operating system and applications. As previously discussed, digital evidence analysis is getting more complex due to the diversity of devices. Forensic tools preserve evidence in different formats. An open-source framework for parsing log file artefacts allows structured development of tools for further analysis.

A combined timeline contains digital events from several sources. *log2timeline (plaso)* [118] is a framework facilitating the generation of a "super timeline" including digital events from the file system, OS registry, logs, as well as application software logs. This contains information on both the device access level and the file system level. *log2timeline* has been widely discussed in the field and forms the basis for significant further research.

Timeline2GUI was developed to analyse `*.csv` log files created by *log2timeline* [119]. It is an easy-to-use timeline analysis tool. Log entries which could be relevant are highlighted and thus makes skimming the timeline easier. For example, "Green indicates that a file may have been opened or created; Yellow shows that the event is related to web activity; Blue indicates that an external device (e.g., USB stick) has been mounted/interacted with the system; Red means some sort of execution was done on the system like an application was started".

An abstraction based approach for timeline reconstruction was proposed in 2020, which is based on the timeline data provided by *log2timeline* [120]. The generated timeline is broken into four levels of abstraction in order to reduce the complexity of the timeline, omitting unwanted details.

2.9 Machine Learning and Digital Forensics

2.9.1 Background of Machine Learning

Machine Learning has been widely applied to digital forensic investigation for data discovery [121, 122], device triage [58, 31], network forensics [123], etc. Flach [124] outlined the ingredients of Machine Learning are: **tasks**, the problems that can be solved with Machine Learning; **models**, the output of Machine Learning; **features**, the workhorses of Machine Learning. When to apply Machine Learning application, there are basically three steps: 1) define the task; 2) feature construction; 3) evaluation and optimisation.

The first step in applying Machine Learning to a problem is to define the task, which is an abstract representation of the problem. For a prediction problem, it can be defined to be either a classification/clustering or regression problem, depending on the type of target labels. Take age estimation as an example. If age is considered categorical, it can be defined

as a classification task; while it could be a regression task if the age is numeric.

Features are "the workhorses of Machine Learning", and feature construction is crucial for the success of Machine Learning applications [124]. There are different kinds of features: categorical, ordinal and quantitative. For text analysis, the raw data is a sequence of symbols that cannot be fed directly to algorithms, *bag-of-word* representation is applied. For image data, patch or contiguous patches can be extracted. During the experiment, features are transformed, selected for reducing over-fitting, improving performance or reducing training time. The *No-Free-Lunch* theorem implies that there is no ultimate feature learner, it is various depending on the data distribution and learning algorithm [125].

Models are the output of Machine Learning [124]. Model evaluation enables its refinement, and the process is iterated until the performance is sufficient. A confusion matrix is able to show the accuracy of a classification task, where the classification performance of each class can be found. The F1 score is an average accuracy of each class, which shows the average performance of the model. Precision and recall are usually used in the evaluation matrix.

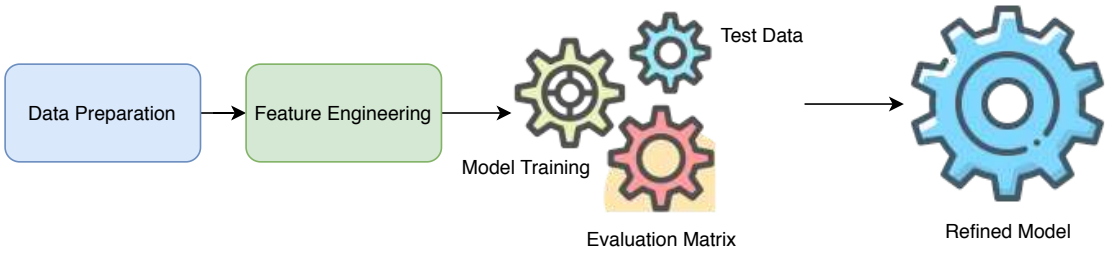


Figure 2.8: Machine Learning Pipeline

Various tools have been developed for Machine Learning for model training and evaluation. For example, *Weka* is for domain specialists rather than Machine Learning experts [126]. It contains built-in tools for standard Machine Learning tasks and facilitates the building of Machine Learning pipelines, training of classifiers, and running evaluations without writing code. In recent years, the python *scikit-learn/sklearn* library is increasingly used both in industry and academic settings. Digital forensic research applying *sklearn* in their experiments include feature extraction for network intrusion detection [127], and metadata-based classification of incriminating digital forensic artefacts [6].

2.9.2 Machine Learning in Digital Forensics

Machine learning is a set of algorithms learned from parsed data and then make intelligent decisions. Machine learning based techniques have been widely applied to diverse fields. Deep learning as a subset of machine learning really shines when dealing with complex problems such as image classification, natural language processing, and speech recognition. Even though deep learning is gaining much popularity due to it is supremacy in terms of accuracy when trained with a huge amount of data in recent years, traditional machine learning

algorithms are preferable with small data size.

Intrusion detection systems employ signature-based methods or data mining-based methods which is typically expensive to produce. Unsupervised Anomaly Detection was designed to process unlabelled data by Eskin et al. [128] in 2002.

In 2007, Khan et al. [129] proposed an approach employing Bayesian Networks trained by file system footprint event to detect the possible execution of certain applications in the event of computer misuse. This approach offers automatic classification, while the challenge is obtaining the accurate footprint of the application execution. In forensic and security, several intelligent malware detection systems can be found. Intelligent Malware Detection System (IMDS) [130] is one example. It applies data mining techniques generating association rules used for malware detection. To handle large volumes of data efficiently, Khan et al. [131] proposed an artificial neural networks based approach generating a post-event timeline of a seized hard disk. Monitoring the file system manipulations, capturing file system snapshots, and then using the captured data to train a neural network to recognise execution patterns of the application programs.

In 2009, an approach for computer user classification was presented in order to quickly classify the seized computer [43]. The proposed method classifies the user through the user's habits, computer skills, interests, etc., which are determined by installed applications, operating system settings, etc. The user profiling could be one of the following 5 categories: occasional user, chat-internet user, office worker user, experienced user, and hacker user. This approach prioritised the seized hard drive. Forensic examiners can focus on only related hard drive images and reduce analysing time.

Machine learning algorithms are easier to be applied, in recent years, with high-level open-source libraries such as *sklearn*. There are many forensic processes potentially solvable by using machine learning methods.

A machine learning solution for device triage was proposed by Marturana et al. ([58] in 2011, and [31] in 2013). The lack of a sufficiently large, shared dataset is a challenge for developing AI triage models. As the triage task consists of a quick, simple examination and analysis to help investigators to reduce the noise and identify relevant information quickly, the development of an emulated, realistic dataset is a substantial task.

Beebe et al. [32] proposed an approach for ranking text search hit to assists faster indexing the aimed objects. The M57 Patents dataset was used in this work, which is a synthetic case constructed by researchers. A range of queries was conducted and the search returned 2,640,681 search hits located in 46,884 allocated files and unallocated clusters. Of these, only 4.24% (112,020 hits) were relevant to investigative objectives. The experimentation indicates by using the ranking model, achieves 81.02% accuracy on the 40% randomly selected test sample by the allocated model, and 85.97% accuracy by the unallocated model.

2.9.3 Background of Deep Learning

The key differentiator of Deep Learning from Machine Learning is that the features are not designed by human engineers. Instead, they are learned from data using a general-purpose learning procedure [132]. Machine Learning tasks require input that is computationally convenient to process. However, it is often difficult to engineer features of real-world data such as images, video, and sensor data. Representation (feature) learning techniques employed by artificial neural networks allows a system to automatically discover the representations needed for feature detection or classification from raw data [132]. In addition, deep learning typically achieves better performance as you feed in more data, as shown in Figure 2.9.

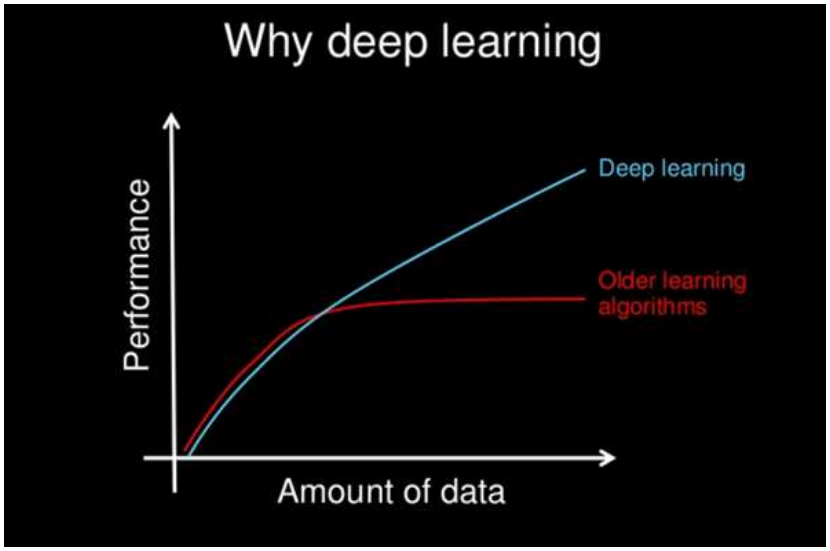


Figure 2.9: Why Deep Learning? (Slide by Andrew Ng)

A deep learning model can be described in two stages, i.e., optimisation and inference. The optimisation process, known as training, is used to update the weights connecting the layers of neurons defined in the model. The process of weight update is achieved by a back-propagation algorithm[132]. Before training a deep learning model, a loss objective is defined to measure the difference/error between the predicted outputs and the targets. The model updates its weights with the objective of minimising the loss function through many iterations. To make it closer to the objective, the mathematics under the hood is gradient descent algorithms for minimising the loss [133]. After completing the optimisation, then the model is applied for inference, namely, making predictions on data that are unseen during training. One key metric exhibiting a deep learning model's performance in inference is the generalisation ability. That says if the model generalises well, it performs on the unseen (test) data as well as the training data.

CNN is one of the variants of neural networks used heavily in the field of computer vision (CV). Recurrent Neural Networks (RNNs) are a very important variant of neural networks heavily used in Natural Language Processing. Long short-term memory (LSTMs) is a special kind of RNN, capable of learning long-term dependencies that make RNNs performant at "remembering" things that have happened in the past and finding patterns across time to make its subsequent guesses make sense.

More specific applications of deep learning include content filtering [134], e-commerce recommendations [135], and search result relevancy scoring [136]. Other applications include camera sensor model identification, image forgery detection, facial detection and recognition, text clustering, etc. There have also been models developed for digital forensic investigation including classification of malware, network intrusion detection, file fragment type, watermarking, steganalysis, pattern recognition, timeline analysis, etc.

2.9.4 Deep Learning Applications in Digital Forensics

One problem faced during file carving is to determine the ownership of carved information when the storage media is used by more than one user. An automated solution to the multi-user carved data ascription was proposed by Garfinkel et al.[137]. The result shows the accuracy of carved data owner classification is from 65.66% to 99.83%. The data used to verify this system is disk images from the Real Data Corpus [75], a collection of more than 2,000 disk images made from hard drives that were purchased on the secondary market. The features used by the proposed automated ascription system are 1) file system metadata (MAC timestamp), 2) file placement (i.e., sector, fragment) information, 3) embedded file metadata (JPEG camera model, Word file save time, etc.).

As the search space for fragments belonging to a particular file is so large, distinguishing the file type of a fragment can shorten the search time. One approach proposed for file fragment classification used NLP (Natural Language Processing) [138]. In this research, a supervised learning approach is taken based on the use of SVM combined with the *bag-of-words* model. File fragments are represented as "bags of bytes" with feature vectors consisting of unigram and bigram counts as well as other statistical measurements (including entropy).

During the investigation of cases such as copyright infringement or ownership attribution, pictures camera model identification is a problem need to be solved [139]. Tuama et al. [140] outlined a novel approach for camera model identification, which is based on CNNs.

Marra [141] applied convolutional neural networks (CNNs) to camera sensor model identification. As described in[141], during the learning process of an ANN, the back-propagation updates the weight, Stochastic Gradient Descent is applied for the weights updating strategy in the model by the authors.

Chen et al. [121] proposed a novel scheme based on fragment-to-grayscale image conversion and deep learning to extract hidden features and therefore improve the accuracy of classification. This CNN model was trained and tested on the public *GovDocs* dataset. The average classification accuracy achieved was 70.9%. Vulinovic et al. [122] stated even though the results of [121] were promising, their approach adds nonexistent correlation to some byte pairs, as bytes 1 and 17 for example now become closely related. Vulinovicet al. [122] applied a CNN model using 1D convolution on the original byte block. Both feedforward neural networks (FFNN) and CNNs are tested. Feedforward neural networks achieved better results

using selected bigrams as input the highest macro-average F1 score is 0.8138.

2.9.5 Current Challenges and Future Directions

There are challenges in applying machine learning approach to digital forensics:

- Improving the accuracy: training models and measuring the accuracy is a challenge because of the lack of large, clean, labelled datasets in some areas or existing datasets not being publicly available. The lack of a sufficiently large, shared dataset is a challenge for developing AI triage models. As the triage task consists of a quick, simple examination and analysis to help investigators to reduce the noise and identify relevant information quickly, the development of an emulated, realistic dataset is a substantial task.
- Explainable AI: to avoid bias, AI process reporting that a system containing criminal activity needs to be able to produce a very clear explanation of why that is the case;
- Security and privacy: sharing models appropriately to avoid attacks such as 'model inversion' and 'membership inference'; Adversarial attacks are one of the challenges of AI model development. It has been suggested that the existence of adversarial attacks may be an inherent weakness of deep learning models [142]. The adversary can manipulate the input resulting in the model producing the incorrect output. Adversarial attacks could also be used as a counter forensics technique. As a result, any pre-trained model could lose its effectiveness during an investigation. To this end, anti-counter-forensics for adversarial attacks remains an open question.
- Validation challenge: for example learning from on-going case processing to expedite evidence discovery in future cases, the result the technique may produce may change on a daily basis;

Despite these challenges, there are many opportunities to enhance AI applications and to apply AI to additional areas of digital forensics. These include inference of behaviour from data obtained from novel sources including smart homes, IoT sensors, vehicle forensics, and combinations thereof. Indeed AI techniques could potentially assist any time there is a need to correlate data from multiple sources, either from multiple suspects, devices or cases. Non-AI based efforts such as a standard form of representations, e.g., CASE [143] will be critical for such efforts.

There will also be significant opportunities in the future for the investigation of AI-based systems themselves. Determining the cause of a decision made by a self-driving car, a smart building, or a SCADA system, will be a new area for digital forensics, although the concept is discussed by Schneider et al. [144]. The investigation of these systems will require significant effort on behalf of the investigator in terms of understanding the models, their training data,

and the state of the model's inputs when the decision was made. This will also require a reasonable level of explainable AI. Of course, the investigation of digital forensic AI systems themselves will be far from exempt from this scrutiny.

2.10 Correlation Analysis

With the big forensic data challenge, the heterogeneity of the data is one important factor. Correlation analysis aims to address the issue of integrating analysis results from multiple sources and formats of evidence data. Analysis tools are usually specialised [145], traditional techniques use one or more forensic tools to process the evidence from each source. For example, *RegRipper*, *plaso (log2timeline)* for log extraction, *Wireshark* for network packet attributes, etc. As a result, cross-correlation analysis on heterogeneous data is a challenge of big data forensics [146, 147]. The diverse device types and data formats further increase the complexity.

The terms Forensic Feature Extraction (FFE) and Cross Drive Analysis (CDA) are proposed by Garfinkel in 2006 [148]. Information extracted from multiple devices, such as credit card numbers, email messages, etc., can be used for discovering the connection between them.

Case et al. [149] state most tools are created for a single specific task resulting in wasted time. Case et al. create the framework Forensics Automated Correlation Engine (FACE) for digital evidence discovery and event correlation.

As the number of files increases, file correlation becomes increasingly important. Metadata plays an important role in digital forensics [150]. Information such as file owner, file size, file date and file type can be used for correlation. The origin of downloaded files can be found using metadata associations [151]. In 2013, Raghavan et al. [152] developed an analysis system *AssocGEN* to determine the association between user files artefacts, logs and disposal not network packets. It can also classify and determine correlations between artefacts.

To speed up the detection and interpretation of incidents from cloud sources process, Raju et al. [153] proposed an approach for cloud-originated event correlation. A normalisation step is required as cloud service logs come in different formats. There are two stages for correlation: 1) consider the events from the perspective of single artefact and perform correlation (homogeneous correlation) and 2) collect the events from multiple artefacts and then perform correlation (heterogeneous correlation).

In 2017, a framework was proposed to get a more valuable reconstruction of events or actions in order to reach case conclusions [154]. The proposed approach applying semantic web technologies to digital investigation processes and tools can improve the automation of parts of the analysis with respect to evidence discovery and correlation.

2.10.1 Cyber-investigation Analysis Standard Expression (CASE)

CASE as a standard digital forensic format was proposed by a number of the EU's leading digital forensic experts at a meeting hosted at the headquarters of Europol's European Cybercrime Centre (EC3) in The Hague in 2017. It is one solution for the correlation analysis problem introduced in the previous subsection, which allows the exchange of information between digital forensic tools. It provides a common language to support automated normalisation, combination and validation of varied information sources to facilitate analysis and exploration of the standard investigative questions (who? when? how long? where?)[8].

A list of industry, including *Magnet forensic, Oxygen, I2 – IBM, and Cellebrite* are currently looking into implementing the standard. Potentially, the use of CASE will provide structure to enhance intelligent analysis (e.g., pattern recognition, machine learning, visualisation) as well as enhancing tool testing and validation.

If it hoped that in the future, leveraging CASE will expedite digital evidence processing speeds. With the CASE standard, automation in digital forensic can also be improved. Besides, digital forensic analysis can be improved as the extracted data for is completed and standardised.

2.11 File Artefact Prioritisation

The ever-increasing number of files encountered on each target device results in a prolonged analysis phase of an investigation. In 2019, a large practitioner survey has been conducted by Sanchez et al. and proved a substantial amount of manual analysis of child abuse material and the lack of an automated tool to assist [70]. Search hit prioritisation can also provide a more efficient analysis process.

Some current tools apply file clustering/classification algorithms to automatically categorise file artefacts. However, the pre-trained model could result in missing relevant artefacts. Prioritisation approaches rank the file artefacts by predicted importance/relevancy.

Gupta [155] proposed a framework to extract evidence and rank artefacts on the basis of their relevancy. A document fraud case test was conducted, applying file extension, file size, file name, file creation time, file access time, file modification time and file depth as learning features. Experimental results demonstrated the viability of the proposed approach.

The larger the number of file artefacts encountered during an investigation, the more pro-

[8] https://caseontology.org/index.html

longed the examination process becomes. Image file examination is important for several cases types. In addition, keyword searching on file artefacts often results in a large number of results being returned. Search hit relevancy ranking algorithms were proposed by Beebe et al. [32] for reducing the analytical burden of text string searching.

2.12 Summary

This chapter presented the current state of the art of digital forensics. The big forensic data challenge results in digital evidence backlog problem. Research work has been done on various levels of abstraction to improve the efficiency of digital evidence processing. Applying machine learning and cloud-based techniques have been proven to improve the efficiency and efficacy of digital evidence processing.

Machine learning is a technique facilitating computers to learn to perform complex tasks on data. One type of machine learning algorithm is supervised learning. Whereby sample labelling is required to train the model. The output of digital evidence deduplication can label the file artefacts as benign or illegal. This has the potential to be used as the input for machine learning models.

Digital forensic process modelling is a topic frequently discussed. The constant evolution of process models has occurred over the last two decades. During this evolution, these process models integrated new components to solve new problems or improve efficiency. Any new tool for digital forensics should also consider how it can fit the general process. In the other words, it should be integratable into the traditional investigation process.

2.12.1 Gaps in the State of the Art

Data deduplication employs hashing techniques to filter out known files. This results in, avoiding the redundant reanalysis of files. Research work has proposed reducing the repeated acquisition process. Experimentation has proven the improvement of acquisition and analysis speeds and storage savings. However, forensic soundness in deduplicated digital evidence acquisition system remains a challenge.

In digital forensics, current machine learning classification techniques are pre-trained solely on illegal files. These models can merely help identify known illegal files or files similar to those illegal files. There is an open question on how bespoke, per-case classification can be used for evidence analysis prioritisation. For example, in both civil and criminal cases, many file artefacts might be not illegal but can nonetheless be relevant to an investigation. There are currently no tools to aid investigators in the identification of likely pertinent previously

unencountered data - both ‘illegal’ or ‘not illegal; but pertinent’.

Chapter 3: **Methodology**

3.1 Introduction

This research introduces a centralised digital forensic system; it allows deduplicated digital evidence acquisition, forensically sound disk image reconstruction, and facilitating machine learning models for file artefact relevancy prioritisation. Due to the requirement of sufficient test disk images for development and evaluation of the approach, tools for test disk image generation are also introduced.

Centralised evidence processing system was considered because gathering the evidence centrally allows preliminary analysis by the case detective without the need for manual, expert analysis. This allows case progressing leads to be identified at the earliest stage possible. In addition, having centrally stored evidence, and evidence processing facilitates the training of ML models and the ability to learn from previous cases.

3.1.1 A Centralised Digital Evidence Processing System

A centralised system for digital evidence processing can benefit from increased hardware resources and collaboration that a single forensic laboratory or police station may have. As discussed in Chapter 2, HANSKEN, evidence as used by the NFI[1], is a centralised system for digital evidence processing and preserves pertinent analysis decisions. Invigilators can apply the developed tools on server-side, and exchange information directly with detectives.

The system implemented as part of this research integrates data deduplication and machine learning techniques. Digital evidence from multiple devices/cases can be simultaneously processed by the system. The acquisition process extracts data from the seized device and transmits it to the server. The collected data from each device are stored uniquely in the central evidence store and the metadata is stored in the database. In fact, should duplicate artefacts be encountered during simultaneous acquisitions, they would only need to be uploaded once (assuming they aren't being processed at exactly the same instant). The timeline generation can also be conducted in parallel. The overall design of the system is shown in Figure 3.1.

[1] https://www.forensicinstitute.nl/

- **Deduplicated Digital Evidence Acquisition**: A centralised database is leveraged for preserving the analysis results of file artefacts encountered. To eliminate repeated acquisition of previously encountered files, metadata and hash values of files are sent to the DFaaS system first. As a result, only newly encountered files are collected by the DFaaS system.

- **Forensically Disk Image Reconstruction**: Forensic soundness is paramount considering the acceptance criteria for digital evidence in court. Even though the effectiveness of deduplicated acquisition has been proven, as discussed in Chapter 2, the forensic soundness of this approach is still an unsolved problem. Therefore, the acquisition procedure of the proposed system includes an additional component for forensically sound disk image reconstruction.

- **Leveraging Known File Artefacts**: At the acquisition stage, the hash value of a file is used for data deduplication. This process facilitates the recognition of known illegal/benign files. These known files then can be used for training model training for classification. The trained models can determine which of the previously encountered files are likely to be more relevant to the investigation.

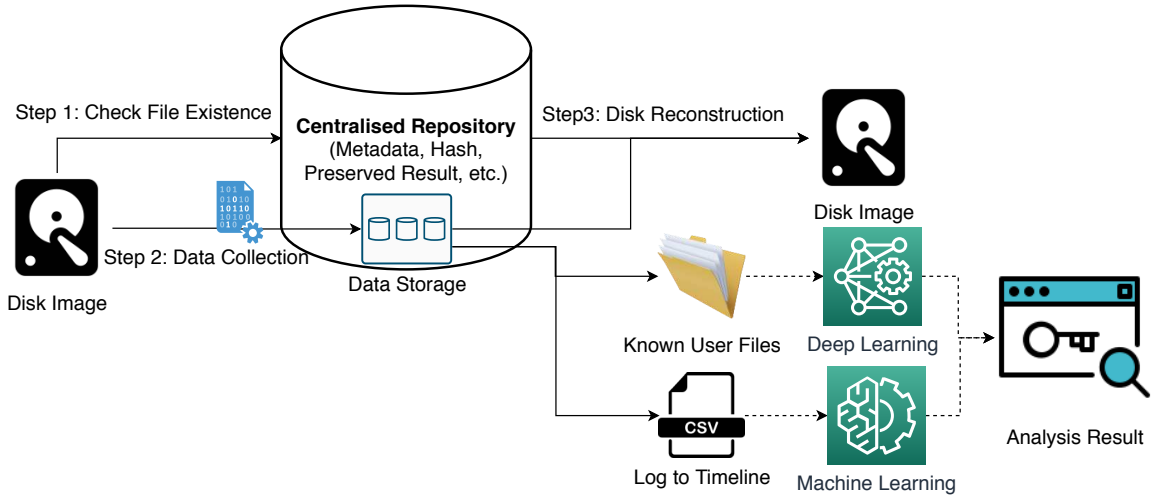


Figure 3.1: Overall System Design

3.1.2 An Automated File Artefact Analysis Approach

Automation tools can assist in processing digital evidence more efficiently, especially when the data volume and the number of files are significantly large. Deduplicated acquisition can recognise known illegal/benign files, and these known files can be used for training machine learning models.

- **Known File Artefact Detection**: One way for file analysis is to compare the hash value of file with the known file database. Hash functions such as SHA1/SHA256 are

commonly used in digital forensics for identifying files. The system checks file existence on the DFaaS system by hash. This approach facilitates the detection of illegal files at the earliest stage of the investigation.

- **Relevancy Determination of Unknown File Artefacts**: As known pertinent files are used for training machine learning models. new files similar to the detected illegal files can be flagged/prioritised as more likely to be interesting to the investigation. each file's metadata and associated timeline events, as well as the file data itself, can be used as input for model training. As a result, new files can be classified into those relevant or not relevant to the investigation.

3.1.3 Design of Experimentation and Evaluation

Figure 3.2 presents the workflow of data generation, acquisition for the analysis used in this research. Experimentation is conducted on disk images generated in virtual machines. The generated disk images are used for testing the effectiveness of both the deduplicated acquisition system and the proposed file relevancy determination approach.

- **Test Image Generation:** TraceGen[2] is a tool for automated wear-and-tear generation on a disk image for analysis. Generated disk images contain common user actions on a computer and files are created, accessed, shared, etc. There are also illegal actions/files emulated on the machine.
- **Deduplicated System** is tested by the repeated acquisition of the generated disk images. Disk images generated are with different duplication ratios, for testing the effectiveness improvement gained through this approach.
- **Relevancy Determination** are tested on disk images with emulated criminal *stories*. The performance can be evaluated for the resulting detection of illegal files.

The result and evaluation conducted as part of this research include:

1. File system analysis on the generated disk images created by the developed approaches;
2. Forensic soundness verification of the reconstruction disk image to assure the acquisition integrity;
3. Repeated acquisition of disk images to evaluate acquisition speed

[2]Du, X., Hargreaves, C., Sheppard, J., and Scanlon, M., TraceGen: User Activity Emulation for Digital Forensic Test Image Generation, Forensic Science International: Digital Investigation, ISSN 2666-2825, September 2020.

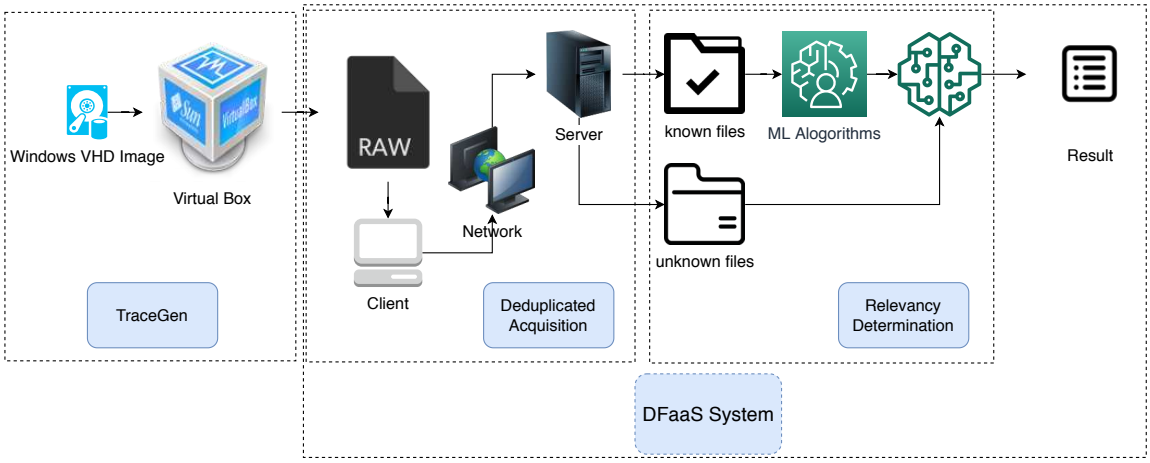


Figure 3.2: Methodology Overview

4. Evaluation of the performance of file relevancy ranking by the personal recall of illegal files covered at the top n percentages;

5. The performance of file prioritisation based on the selection of features used;

6. The performance of file prioritisation on realistic disk images (with different amounts of known files, various case scenarios, etc.).

3.2 Tools for Test Disk Images Generation

In this subsection, the developed toolkit to generate data for the experimentation and evaluation are introduced; *TraceGen* and *EviPlant*. These tools can benefit disk image generation, manipulation and distribution.

EviPlant can be used to distribute "stories" and merge into base images reducing the need for large storage of hard drive images for training, tool testing and validation and enables more realistic scenarios for analysis and processing in a remote education scenario. This is achieved using the diffing engine to create mergeable evidence packages. Tracegen can be used for executing both "stories" and generating realistic wear-and-tear.

3.2.1 TraceGen: Overview

The overall aim of the tool is to provide an automated approach to generating disk images for digital forensic research and tool testing and validation, proficiency testing and education.

An overview of the system design is shown in Figure 3.3. The input is a list of user actions.

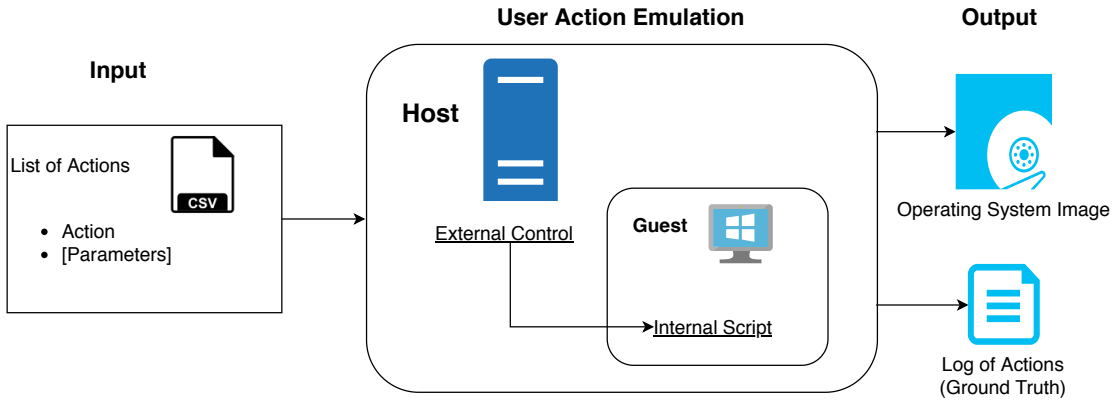


Figure 3.3: TraceGen: Overview of the Approach[a]

[a]Du, X., Hargreaves, C., Sheppard, J., and Scanlon, M., TraceGen: User Activity Emulation for Digital Forensic Test Image Generation, Forensic Science International: Digital Investigation, ISSN 2666-2825, September 2020.

The script executes on the host and controls the guest machine; from the boot, performing internal and external actions, to shut down. In the end, a forensic image with user traces in the guest OS is generated alongside the log recording the actions executed, providing a ready-made ground truth.

At present, internal scripts are executed inside the VM to automate internal user actions. There will be inevitable traces left on the disk of the VM that would not be present with human-only generated data. However, building disk images inside virtualisation platforms, which is common practise, is at some level already inconsistent from a real system (for example virtual hard disk identifiers, virtual USB controllers, etc.) Certainly, in educational assignments/proficiency testing, this can be covered using the phrase "During your analysis, please ignore any virtualisation artefacts that are part of the data generation process". Students/test-takers could also be instructed to ignore results of the artefact automation process, if these artefacts can be identified, minimised, and segregated. An entirely GUI based, external approach would eliminate this issue.

3.2.2 TraceGen: Existing Automation Options

The types of actions that can be performed on a target VM are therefore split into three categories:

- **Machine Control Actions** - these are performed outside of the VM, e.g., powering on the VM, performing unsafe shutdown the VM (i.e., "pull the plug"), adjusting the BIOS time, etc.
- **External User Actions** - these are performed from outside the VM, e.g., copying

or moving files, performing a Google search, shutting down the VM in a controlled manner.

- **Internal User Actions** - these are performed inside of the VM, e.g., copy or move files, perform a Google search, shut down the VM in a controlled manner.

When implementing emulated user actions, there are a number of options available. At a high-level, these include:

- **Application Programming Interfaces (APIs)** – For example, *pywinauto*[3] facilitates the automation of the Windows GUI using the Win32 API. It allows Pythonic interaction with GUI components, e.g., to save a file in Notepad:

```
app.UntitledNotepad.menu_select      ("File->SaveAs")
```

- **Simple Mouse and Keyboard Control** – This allows injection of keyboard input, mouse movement and clicks at specific coordinates.

- **Graphical User Interface (GUI) Interaction** – this technique is more advanced than 'blind clicks' above, and can also visually process any given application and programmatically execute specific mouse and keyboard actions. Examples of GUI based automation tools include *Sikuli* [156] and *PyAutoGUI*. Sikuli enables screenshots of GUI control elements to be taken (such as a toolbar button or icon), which can be included in a sequence of actions in order to script complex interactions with any application. In a similar vein, PyAutoGUI[4] is a cross-platform GUI automation tool designed for programmatically controlling the mouse and keyboard. One benefit to this approach over Macros is that the script is resilient against any specific control element not appearing in precisely the expected coordinates on the screen.

- **Browser Automation** – There are also application-specific automation tools that could be used, for example *Selenium*[5] that automate most major web browsers.

One limitation is this tool leaves extra traces by running internal scripts. However, this approach is to attempt to programmatically automate the actions of a user and allowing the system to generate realistic artefacts, rather than trying to artificially create them.

[3] `https://pywinauto.readthedocs.io/en/latest/`

[4] `https://github.com/asweigart/pyautogui`

[5] `https://www.seleniumhq.org`

3.2.3 EviPlant: Image Generation Using "Evidence Packages"

EviPlant[6] is a tool created for easier creation, manipulation and distribution of disk images for digital forensic testing [3]. There are two components of EviPlant: the *diffing* tool and injection tool. The fundamental building blocks for EviPlant are evidence packages.

Evidence packages contain all the digital artefacts and associated metadata created during the emulation of the crime or wear-and-tear actions. The artefacts contained within evidence packages fall into two categories, X and Y which combine to capture all the necessary data modified from the base image after performing a specific task or set of tasks.

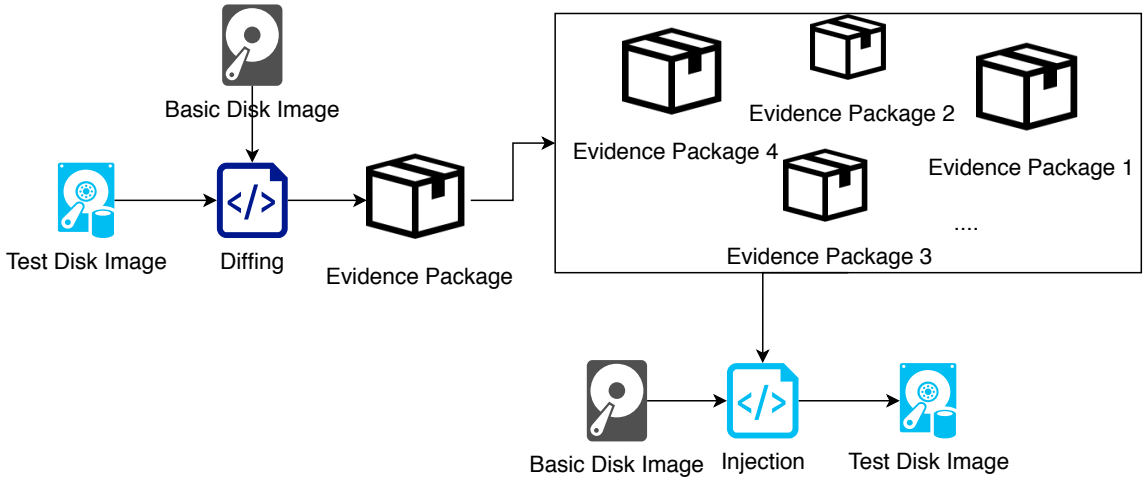


Figure 3.4: EviPlant: Evidence Package Extraction and Injection

- **The Diffing Tool** compares two images and extracts all artefacts that prove different. The resulting evidence package contains the artefact(s), the associated `*.csv` metadata (i.e., file name, logical position, etc.) of each extracted artefacts in the evidence package. In addition, this tool enables high-frequency package creation, real-time monitoring, and reconstruction of the device state at any point necessary.
- **An Evidence Package** consists of the metadata and artefacts extracted from disk images by the diffing tool. It is easier to distribute evidence packages as they are smaller than entire disk images. The preservation of different disk images subsequently requires the storing of one basic disk image and many associated evidence packages.
- **The Injection Tool** put the artefacts into the target image based on the metadata stored in the database. Test disk images then can be created as needed.

[6]Scanlon, M., Du, X., and Lillis, D. (2017). EviPlant: An Efficient Digital Forensic Challenge Creation, Manipulation and Distribution Solution. Digital Investigation, 20, S29-S36, Elsevier. `https://doi.org/10.1016/j.diin.2017.01.010`

When many different images have to be created, it can be completed by editing the evidence packages, instead of manual operations on each image. The benefits of EviPlant include:

- Saving Storage Space: If many different disk images have to be stored, it likely requires very large storage space. Preserving only the *diffed* files and metadata between acquisitions saves a significant amount of space.
- Distribution: Sharing disk images over a network is slow. The smaller size of the evidence packages makes it easier to be distributed and share.

3.3 Deduplicated Data Acquisition of the System

Disk devices are the most common devices in digital forensic investigation and typically contain a wealth of information. Disk data acquisition requires a verifiable copy of the storage device. Disk imaging is a process of making a bit-by-bit copy of the entire contents of a hard drive. Hashing the output from this technique is used to assure forensic soundness.

Overall acquisitions conducted in a forensic laboratory, there are many repeated artefacts; from operating system and applications to commonly shared files, e.g., music files. These are acquired again and again; wasting valuable investigative time. Eliminating previously encountered files is crucial for achieving a more efficient investigation process overall.

Block-level hashing wasn't used as block-level hashing increases the time for the hashing data on the disk. In addition, further analysis (timeline event-based file relevancy determination) is file-based. A file's hash value is used for recognising previously classified benign/illegal file, as shown in Figure 3.5 [1]. The deduplicated acquisition system proposed as part of this research is designed to move this process to the earliest stage possible. It eliminates repeated acquisition of common data and more importantly, it enables the detection of known illegal files during the acquisition stage.

The system proposed as part of this research acquires data residing on the disk device. Only file data that is new to the DFaaS system is collected. The physical location of each file fragment is recorded during the acquisition. For collecting complete disk images, *the free space, unallocated space, and file slack* are also acquired.

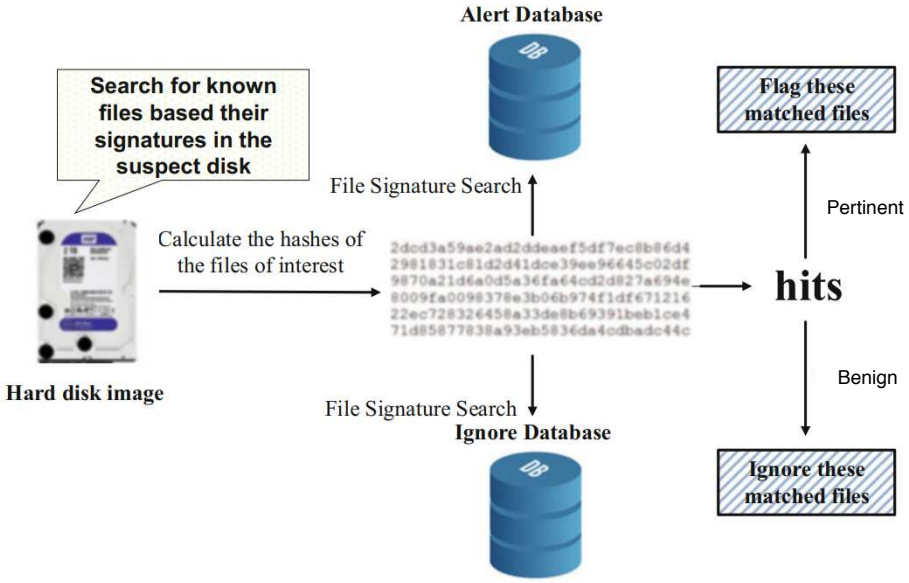


Figure 3.5: File Signature Search Process Overview [1]

3.3.1 Deduplicated Acquisition Process Pipeline

Figure 3.6 shows the process of evidence data transmitted from the client to the server. The data acquired from the target devices include the files, slack space and unallocated space. The steps include:

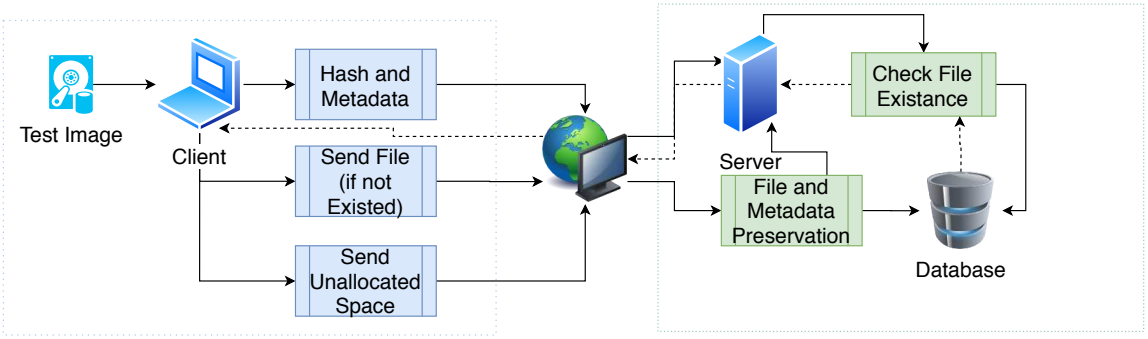


Figure 3.6: Deduplicated Acquisition

1. The client extracts metadata from disk image including the file hashes;
2. The client sends metadata to the server;
3. The server checks for the prior existence of the received files hash values;
4. The server sends *Yes* or *No* existence of the file to the client;
5. The client sends the file (file data and file cluster slack) if required;
6. The server checks the integrity of the file received;
7. The server sends a request to the Client again if the hash value does not match;

8. The client sends non-file data (i.e., free space and unallocated space) on the device to the server.

This approach calculates files' hash value before acquisition enables illegal files to be detected at an earlier stage of the investigation. Hashing files for checking data integrity and recognising known files are implemented in this approach, there is no need for other tools to do the same work. When applying other tools for further processing, investigators could choose to skip the file hashing step. In addition, future commercial tools developed under the standard CASE, information exchange (import and export data) between forensic tools is possible. Hash values generated by this approach could be imported to other commercial tools when needed.

3.3.2 Client Responsibilities

The client extracts the file system metadata, file data and hashes the file data. For disk image reconstruction purposes, data from file cluster slack, free space and unallocated space are also collected.

- **File Data and Metadata Extraction**: The file system analysis library `pytsk` is used to extract file system metadata and file data on a disk drive. The physical location of each file fragment is also included in a file's metadata.
- **File Hashing**: Before data transmission, the file data is hashed using SHA1. Its metadata and hash are sent to the server first. The hash value is used to check for the existence of a file on the server. If it does not exist, then a file request will be sent to the client. More importantly, if it is an illegal file, it can be flagged during this process.
- **File Slack Data Extraction**: `pytsk` provides the start block and length of each file fragment. The summation of the allocated block size minus the file size provides the slack size. The Linux `dd` command copies a storage device bit-by-bit from an input or source file to an output or destination file and is applied for copying this data.
- **Extracting Other Data on the Disk**: The blocks numbers of free space and unallocated space can be acquired through discounting the blocks allocated to files already acquired from the total blocks of the partition. The `dd` command is also applied for copying this data.

3.3.3 Sever Responsibilities

The server side of the proposed system checks if each file requires upload, verifies the transmission of evidence is verified and preserves the collected data. Digital evidence analysis is also conducted on the server and will be discussed in later sections.

- **File Existence Check**: A database preserves the hash value of previously encountered files. Each time a file is encountered, its file signature is searched for in the database to determine if the uploaded acquisition is required.
- **File Integrity Check**: In case of data loss or corruption, every uploaded artefact is verified server-side. If the hash does not match, then the server will send a request to re-upload that specific file again.
- **File and Metadata Preservation**: The files are saved using the same logical structure of files on the target disk. Each time a new file is acquired, the existence of its parent directory is checked and created if it does not exist.

3.3.4 Summary

- The proposed system collects unique data encountered across all gathered evidence, saving bandwidth and storage space and facilitates basic cross-case intelligence, i.e, detecting previously classified pertinent data during acquisition.
- This acquisition process acquires more than just a copy of evidence artefacts, it has also completed part of the preliminary analysis (generating the hash digests and storing associated metadata into the database for future examination).
- This approach facilitates the detection of illegal files at the earlier stage of digital forensic investigation. It can also potentially shorten the time for remote whole disk image acquisition, i.e., in scenarios where the network speed is the bottleneck of the acquisition, eliminating data to be transmitted is desirable.

3.4 Data Storage of the System

This subsection introduces the data storage aspect of the proposed system. It is specifically designed to enable deduplicated digital evidence acquisition and forensically sound disk image reconstruction. It has been presented how the client and server operate in unison for data collection.

As discussed, even though a data deduplication technique is applied, the cumulative data preserved on the DFaaS system contains a complete copy of the target disk. There are different abstract levels of data residing on the disk; block level, file level, file slack, and unallocated space. Besides the files recovered, data at the physical level is also preserved for complete disk image reconstruction. The collected metadata and file artefacts are saved server-side ready for reconstruction when needed.

Server-side, a file is saved in the same logical path/directory as it was on the original target disk. The categories of files are presented below in Section 3.4.1. Non-file data on the device are saved in another directory; with the physical location (block number) being used as its file name. A detailed description of this is presented in Section 3.4.2. A database is used to preserve the metadata, as well as the logical and physical locations of a file. The metadata preserved are presented in Section 3.4.3. Section 3.4.4 discusses the log of each acquisition - stored into an auditable file.

3.4.1 Categorising Files Collected from the Disk

As shown in Figure 3.7, from a file analysis perspective, files are categorised into three classes as benign files, log files and user files. User files are multimedia or documents; these could be known as benign/illegal and unknown benign/illegal. Log files, e.g., browser history, should typically be unknown files as the operations are logged different on different devices.

- **Benign Files** operating system files, installed applications files, as well as files that were classified as such in a previous investigation. To reduce the time of processing, these files are filtered out through data deduplication at the acquisition stage;
- **User Files** are files such as images, audio, video, archives, documents, and will require content analysis to ascertain if they are illegal or benign. This analysis result should be preserved, and subsequent occurrences of this file will be recognised as known files. These user files are analysed at the next phase:
  - **Known benign/illegal file artefacts** can be used for training machine learning models for prediction of new case investigation;
  - **Unknown file artefacts** can have their relevancy determined from these trained models.
- **Log Files** are various, from operating systems and applications. These files often contain the answers to the most common questions of an investigation (what, when, where, why, how, who). Digital events from log files are also capable of supporting the analysis of user files. The associated timestamps extracted represent digital events on file artefacts, which are used as features to represent files as input data to train machine learning models.

3.4.2 Unallocated Space and Slack Space on the Disk

When the client extracts non-file data from the seized disk, the data is also extracted and sent to the server. The block numbers, which represents the physical location of the data, are used in their server-side name.

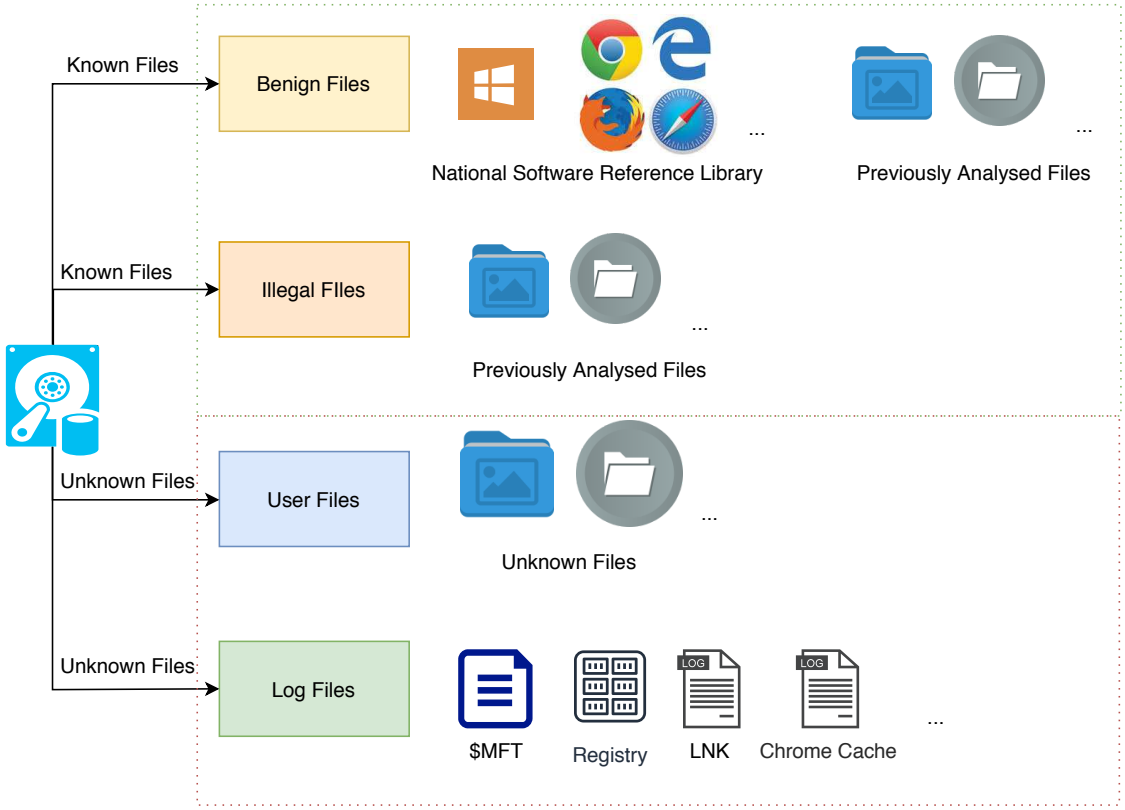


Figure 3.7: Categories of Files on a suspect Device

- Unallocated Space: A partition or logical drive must be formatted before it can be available for data storage. It is possible that less than the entire partition or physical drive is used, and some space is left unformatted. Under normal conditions, it cannot be allocated to files. However, it could be used for data hiding;
- Unallocated Space: Another kind of unallocated disk space is the unused space on a formatted partition; the blocks that are not allocated to a file. However, these blocks could contain data from the previous file allocations. Deleted data can be recovered from these blocks.
- File Level Slack: File systems allocate disk space to a file in fix-length clusters; there could be space leftover in a block of the end of the file. The unused bytes in the last data unit for a file are called slack space and can contain file fragments from previously deleted content.

3.4.3 Metadata and Acquisition Records

Metadata storage has three uses: 1) it supports deduplicated evidence acquisition; 2) metadata examination is important in the process of digital forensic investigation; 3) it collates all the metadata of each acquisition together, enabling the future intelligent examination.

File metadata such as *DedupedPath*, *file_fragments* are collected for disk image reconstruction. Other filesystem metadata, such as MACB (Modification, Access, Change and Birth) timestamps for digital forensic analysis, is generated independently. Figure 3.8 shows an example of the preserved metadata for each individual file.


```
{
    "_id" : ObjectId("5f0a599186816f6e56c1d777"),
    "sha1" : "6b97a2de356963676f616ff54734f103141b7517",
    "file_fragments" : [
        [
            8789,
            8853,
            0
        ],
        [
            8788,
            8789,
            0
        ],
        [
            8274,
            8275,
            1
        ]
    ],
    "cmd" : "SERVER file request",
    "DedupedPath" : "/mnt/md0/dedup_code_win/Received_Files/2020-07-
12.01.26.50.527891/NTFSSPACESPACExFATSPACE(0x07)1///$MFT",
    "files_partition_offset" : 1048576,
    "filename" : "$MFT",
    "mongo_id" : "5f0a598e86816f6e56c1d74a",
    "acquisition_id" : "/home/xiaoyu/test_images_raw/Windows7_10GB.raw_20200712013004",
    "partition_dir" : "NTFS  exFAT (0x07)1",
    "file_requested" : "//$MFT",
    "inode" : 0
}
```


Figure 3.8: File Metadata for Deduplication and Image Reconstruction

- *sha1* is the hash value of the file;
- *file_partition_offset* helps to find out the physical location of a file for disk image reconstruction;
- *file_fragments* is the blocks allocated to the file. The number of fragments can also be found;
- *DedupedPath* is the logical directory of a file on the server. When one file is used to reconstruct for a disk image, its location can be found here;
- *mongo_id* is generated by MongoDB;
- *acquisition_id* represents which disk image the file is from;
- *partition_dir* is the partition name;

- *file_requested* is the file name;
- *inode* is a "file ID" in NTFS, which is the record number representing the file in the Master File Table.

The devices acquisition records are also saved in the database. Figure 3.9 presents an example of a device acquisition record.


```
{
    "_id" : ObjectId("5f0a6aea86816f6e56c31951"),
    "partition_size" : NumberLong(10630463488),
    "image_id" : "/home/xiaoyu/test_images_raw/Windows7_10GB.raw_20200712013004",
    "acquisition_time" : 1594513805.91503,
    "image_hash" : "7908120e26bed8b3f9dfe5fce3f90ca6ec4ead7f",
    "image_size" : NumberLong(10737377280),
    "dedup_root_path" : "/mnt/md0/dedup_code_win/Received_Files/2020-07-
12.01.26.50.527891",
    "partition_sha1" : "da39a3ee5e6b4b0d3255bfef95601890afd80709",
    "image_name" : "/home/xiaoyu/test_images_raw/Windows7_10GB.raw",
    "partition_name" : "NTFS  exFAT (0x07)"
}
```


Figure 3.9: The Acquired Disk Image

- *partition_size* represents the size of the collected partition;
- *image_id* is generated by the disk name and timestamp of acquisition time;
- *acquisition_time* is the time taken for disk acquisition;
- *image_size* is the size of the disk image;
- *dedup_root_path* is the root path of the saved files;
- *partition_SHA1* refers to the sha1 hash of the partition data;
- *image_name* refers to the name of the disk;
- *partition_name* refers to the name of the partition.

3.4.4 Acquisition Logs for Performance Analysis

The server-side operations generate auditable log files for analysing the performance and verifying the accuracy of the proposed system. The acquisition log (stored in a *.csv) records the data verification, collection time, transmission speed, duplication ratio, etc. The full list of entries and their description is outlined below:

- Log Time Stamp: the start time, i.e., when the server first time receives the client's request;
- Image Processed: `caseid_devicename`;
- Reading Time(s): the time taken for metadata extraction and file hash calculation;
- Files on Disk: the number of files on disk;
- File Size on Disk: a sum of all files' sizes;
- Files Sent to Server: the number of newly encountered files;
- Total Bytes Sent to Server: the size of all collected files and data;
- Actual System Throughput (MB/s): transmitted data/time taken;
- "Effective" System Throughput (MB/s): reconstructable data/time taken, reconstructable data is the complete data on the disk;
- Duplication Rate: (disk size - transmitted size)/disk size.

3.5 Forensically Sound Image Reconstruction from Deduplicated Acquisition

Even though previous work as outlined in Section 2.7 have proven the efficiency of a deduplicated acquisition system for digital forensics, the forensic soundness of duplicated digital evidence is an unsolved problem. In comparison with the current state of the art and alternative approaches, the approach outlined as part of this work progresses one step further, i.e., forensically-sound complete disk image reconstruction.

In the proposed system, data imaging is not achieved through an entire bit-for-bit copy, but it does afford the same forensically-sound disk image to functionality the investigator. The acquisition result can be verified by comparing the hash of the reconstructed disk image to the original one. For the purpose of disk image reconstruction, there are three constituents of binary data necessary. The file data, the block-level slack space, and the unallocated space on the disk. This reconstructed image can subsequently be verified against the original drive by comparing their hash values.

During acquisition, the previously acquired data is skipped and is not redundantly acquired by the system. The data subsequently used for disk image reconstruction uses the previously collected data. Retrieving a file is conducted using its hash, and then reading for the required file from the *dedup_path*, which is saved the directory of the file on the server.

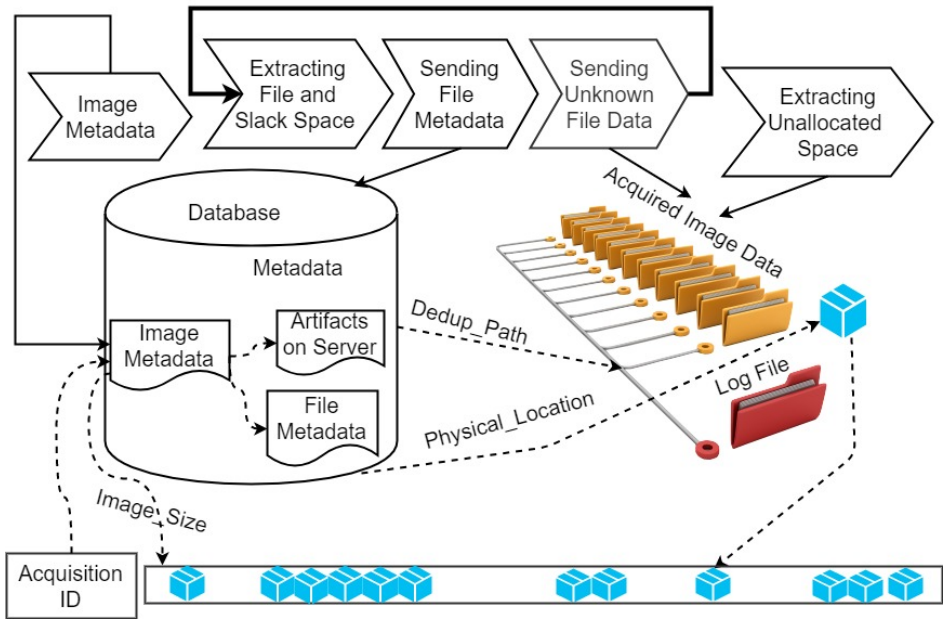


Figure 3.10: Forensically Sound Disk Image Reconstruction from Deduplicated System[a]

[a]Du, X., Ledwith, P., & Scanlon, M. (2018). Deduplicated Disk Image Evidence Acquisition and Forensically-Sound Reconstruction. The 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications (IEEE TrustCom-18).

Figure 3.10 shows the process for image reconstruction. The forensic artefacts from each acquisition and the metadata stored in the database are necessary for recreating a forensically-sound image. To recreate an image, the system first needs a specific *acquisition id* and based on this *id*, the information such as image size and data storage locations can be retrieved. A blank staging image is first created and subsequently, each of the artefacts is placed at the same specific physical block offset as in the original disk. Finally, a hash is generated and compared with the original device's to verify successful reconstruction. The reconstruction log is generated; including the reconstruction time, result, the hash of the complete disk image, etc.

Evidence reassembly is only required if a traditional digital forensic tool is needed to be used for analysis and this tool requires access to a full disk image. The growing paradigm shift towards cloud-based evidence processing can completely negate the need for full disk image reconstruction. However, as a bridging technology, full disk image reconstruction is needed to ensure court admissibility and compatibility with existing workflows.

3.6 Data Extraction and Preparation

3.6.1 Data Extraction: Tools and Techniques

The system implemented is tested on disk images with Windows 7/8/10 and NTFS file system. The Windows disk image for analysis is run on VirtualBox. File system-level data extraction uses `pytsk`, timestamps extraction from logs uses *plaso*.

The digital forensic analysis approach employed in this research is for user files relevancy determination. File data can be used to determine its relevancy, in combination with its associated metadata and timeline events. The different input for model training offers multiple options for relevancy score generation. This will be determined by the investigative scenarios. More detailed discussion on this topic is attached in Section 3.7.4.

Figure 3.11 shows how the data for training machine learning models are generated from a disk image. The input is disk image in CSV format. The output is a file in *csv* format. The tools for metadata and digital events extraction operate directly on the raw disk image. There are three parts at this stage: 1) metadata extraction using `pytsk`; 2) digital events extraction using `plaso`; 3) merging each file's metadata and digital events. In addition, feature engineering and feature selection are required before being used for training models.

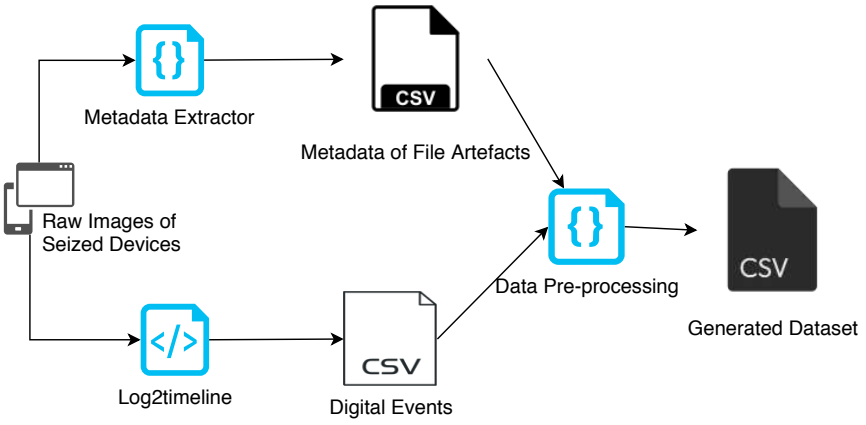


Figure 3.11: The Developed Toolkit for Data Extraction and Processing

3.6.2 File Data Reduction/Selection

Data reduction allows forensic experts to focus on analysis files potentially more useful to the investigation. For different types of cases, investigation focus is also different. Multimedia are more interesting in a child abuse case investigation. Documents are more likely relevant to a financial fraud investigation. These files can be selected for automated classification by using file header information. This would first identify all files of a specific category, e.g.,

image files, and then these would be used to perform automated classification based on their associated metadata and user usage patterns.

As discussed, machine learning models are trained for user files relevancy classification. User-generated files in an investigation commonly are multimedia files and documents. It could also be computer code/scripts in a hacking investigation.

Therefore, a data reduction process is required before analysis for selecting only the objective file types. Table 3.1 shows file type and file extensions that can assist the file selection for investigation.

File Type	Common File Extensions
Pictures	jpg, bmp, png, gif, psd, jpeg, tif, tiff
Videos	wmv, avi, mov, mpg, mp4, flv, 3gp, mts, vob
Audios	mp3, wav, amr, ogg, m4a, wma
Documents	doc, docx, dot, wri, rtf, odt, odg, ods, ots, pdf, xls, xlsx, xlsm, ppt, pptx, myob

Table 3.1: Data Selection by File Type/File Extension

File extensions can be faked and edited. There are tools to detect extension forgery and correctly identify the file type. Tools can be integrated into this system, but this is beyond the scope of this work.

3.6.3 File Timeline Generation and Feature Extraction

Researchers in this field have developed and proven file clustering/classification models using metadata or file data to assist digital forensic analysis. This research adds file specific timeline events into consideration for file classification to determine files that are likely to the investigation.

A **file timeline** is composed of the digital events associated with a specific file. These digital events are from the generated "Super Timeline" (a disk image timeline). Figure 3.12 presents the file timeline generation process. The file name list contains the selected types of user files. Each file name is used as the keyword to search from the disk image timeline for file timeline generation. Subsequently, features can be extracted from the file's timeline to be used for the representation of the file.

Feature Extraction: Features from the file timeline express the digital events associated with the file. There can be digital event sources, the number of each event source associated with the file, etc. The sources of digital events include `FILE, LOG, LNK, WEBHIST, REG`, etc. As several feature options are possible, feature selection is required before to inputting the data to machine learning models.

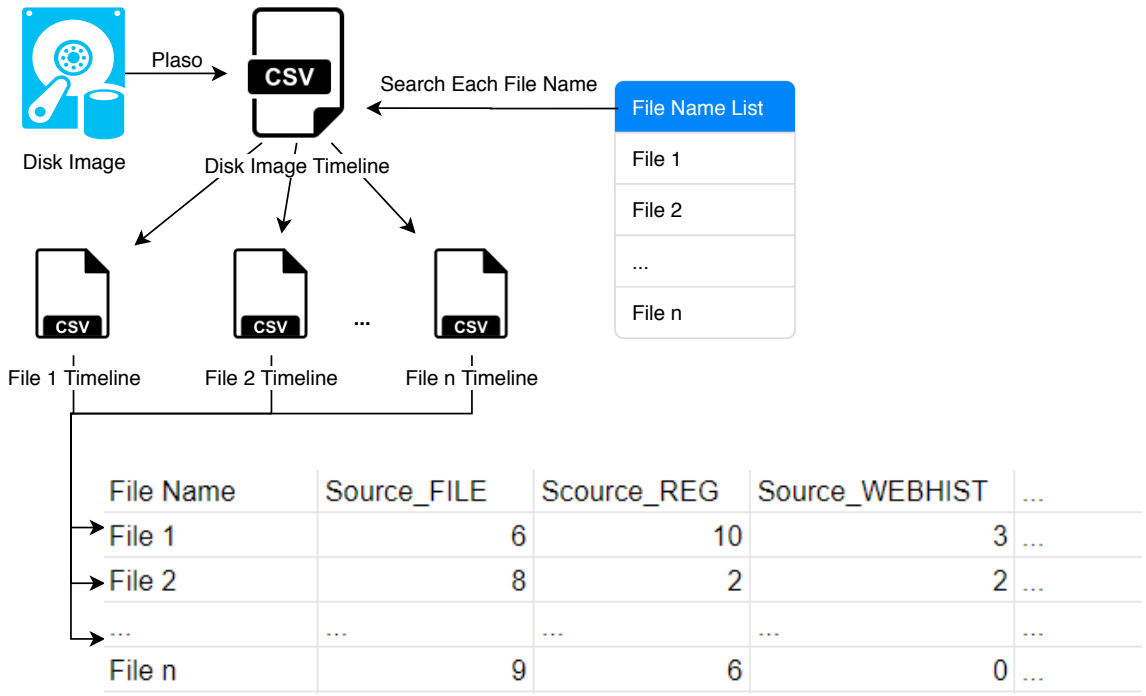


Figure 3.12: Data Processing: File Timeline Generation and Feature Extraction

An abundance of features can be easily obtained from file artefact timelines. However, the number of features should fit the dataset to achieve optimal performance. Feature selection helps to avoid missing useful features while also identifying the most significant features available. Which features and how many should be included? Generally, larger datasets can handle more features. So when the dataset is small, fewer features can be beneficial to maintain a usable performance.

3.6.4 Feature Extraction from File system Metadata

Features can also be extracted from file system metadata. The most popular events (and their corresponding source, type, etc.) can be used as features, but bespoke metadata, such as filename and timestamp, are likely not suitable to use directly for training classification models. Table 3.2 lists the useful metadata and the corresponding features that can be extracted, manipulated, and transformed.

In training machine learning models, not every feature available proves valuable. For example, randomly generated numerical features, artefact hash values, or `inode` values are not helpful to a prediction task. Feature manipulation is usually needed for a specific task or purpose for each machine learning model. Through feature transformation, the information input for training the model can be added, changed or removed as desired for each task. In fact, the resultant model is a way of constructing a new feature that solves the task at hand [124].

Metadata	Features Extracted
Timestamp	Day of the week, time of day, the number of years/months/days/hours of the file created/last access/modified, etc.
Filename	Length of the filename, character types in the filename, language, etc.
File Type	Type of file, for example, image, document, executable, etc. This is based on file header information as opposed to file extensions.
Owner	Username of the file creator.
File Size	Categorising the size in KB.
File Path	Depth of the directory; depth is defined as the number of parent directories in the hierarchical file system.
Digital Events	The number of associated events occurred on the file artefacts in total; the number of the events occurred on the file artefacts on/in a specific day/week/month; the most frequent type/source of events happened; and so on.

Table 3.2: Valuable Feature Extracted

Feature selection techniques should be applied to determine what features are best applied to the model. Identifying the most influencing features can be used to improve the performance of a machine learning model. However, a balance must be struck – having too few features in a model could lead to over-fitting.

3.6.5 Train/Test Data and Evaluation

Testing disk images used contained emulated "illegal actions", and "illegal files". Some other files in the disk are normal/benign files. In the experimentation, assuming a subset of these files are assumed by the deduplicated system as "known illegal files". The proposed machine learning approach is then applied to test if other illegal files can be ranked at the top. The result is evaluated by the recall of reviewing the top n percentages of the ranking result.

When applying this approach to an investigation, both the training and prediction process is conducted during the investigation. The training data are known files classified as known as illegal/benign. The features applied can be selected during the investigation. Instead, the unknown files are ranked by the relevancy score predicted by the model.

3.7 Relevancy File Artefacts Prioritisation

Traditional triage or data reduction approaches build filters based on investigative experience. For example, looking for a document or an image in a financial crime, e.g., scanned documents,

can cause an issue with the volume of results returned if merely filtering by file type. Specific keyword searching might only retrieve a very limited or empty result. Files can be selected or removed by metadata (file size, file type, etc.). However, in some cases, e.g., a child abuse material investigation process, the number of files can remain very large. File prioritisation is required to provide a more efficient analysis when the number of files is huge.

To reduce the manual analysis effort required during the investigation, automated detection of suspicious file artefacts (i.e., file artefacts that are pertinent to the investigation) is proposed. Previous approaches applied hash value for automated filtering of illegal and benign files. The proposed approach takes advantages of the detected illegal file artefacts and uses metadata file to automatically identify files more likely to be pertinent to the investigation.

This approach predicts the relevancy of unknown files by the machine learning model trained on the known files. It can be used in different investigate scenarios, with the difference being the selection of features applied: 1) files are from the same device: file path, file source, file access, etc. 2) files are from the same case but multiple devices: file source, file access, file shown on different devices, etc.

- **Relevant Files**: Relevant files are files interesting to investigators. Files with similar content, metadata or interactive user behaviour to the detected illegal files are likely relevant files. These files are ranked by the relevancy score provided by the machine learning model.
- **Relevancy Determination**: It is not necessary to use all possible features to represent a file. In different situations, the selection of features determines what files are more relevant. Different investigate scenarios are tested in this research.
- **Relevancy Score**: This approach generates a relevancy score for each file encountered on a disk image. A detailed description is provided in Section 3.7.4.

Machine Learning Model Training Process - As described, the machine learning model is trained by known files, features extracted from each file, the file's metadata, and the file timelines. For example, the proposed investigation tool can select files by their size range, MACB time period, etc. The proposed approach uses timeline events containing more information the file system alone identifies out similar files. Similar files are those more with comparable investigative value (pertinent or not).

Relevancy Score Generation - As shown in Figure 3.13, files for analysis are firstly checked with the known database. Then the known relevant and benign files detected are used for model training. This is different from many other applications in digital forensics whereby models are trained for general usage, i.e., network intrusion detection, malware detection, etc. In this research, the user actions (i.e., timeline events) on all encountered files are used as input. Similar files are then identified by the associated user behavioural actions. Therefore, the model is trained specifically for each case, i.e., potentially for each image under investigation.

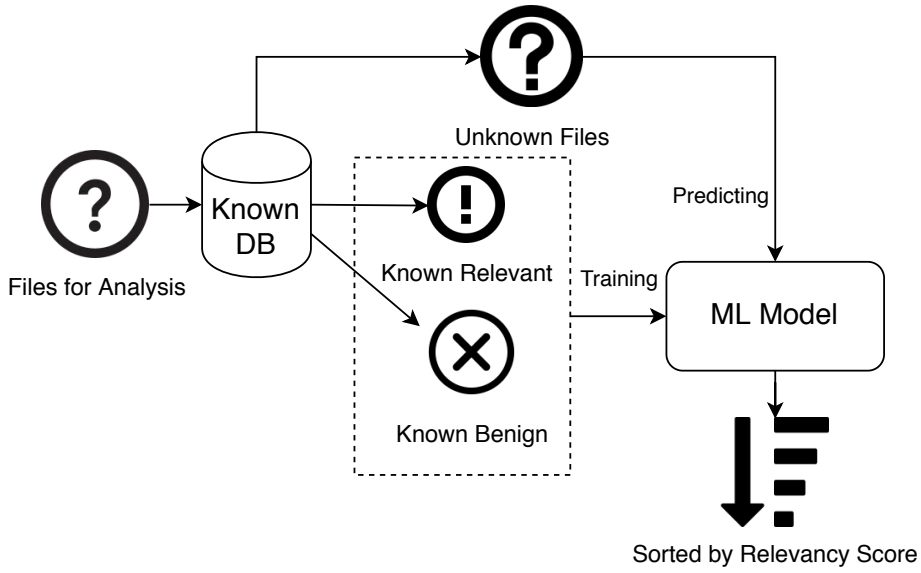


Figure 3.13: Relevant File Prioritisation Approach[a]

[a]Du, X., Le, Q., and Scanlon, M., Automated Artefact Relevancy Determination from Artefact Metadata and Associated Timeline Events, The 6th IEEE International Conference on Cyber Security and Protection of Digital Services (Cyber Security), Dublin, Ireland, June 2020.

3.7.1 An Overview of the Workflow

This research assumes files with similar usage pattern are more probably in the same class (illegal or benign). For example, files A.doc and B.doc are created by `Microsoft Word` on the same day (02/09/2020), files C.pdf and D.jpg is downloaded from Google drive on 24/10/2019. If A has been detected by the hash database as illegal, B is more relevant to the investigation; If D is detected as an illegal file, C should be ranked at a higher priority than A and B.

Figure 3.14 illustrates the designed workflow as could be applied in an investigation. From a raw format image copy to the prediction result, the process is completely automated. The processing is broken down in four steps:

1. **Dataset Generation** - As has been discussed in Section 3.6, the developed toolkit is used to generate the training dataset each artefact's format for machine learning modelling.

2. **Filtering** - This works through comparing the hash with the known database, which will split the file artefacts into two categories; known file artefacts and unknown file artefacts.

3. **Training the Model** - Training a machine learning model using known file artefacts.

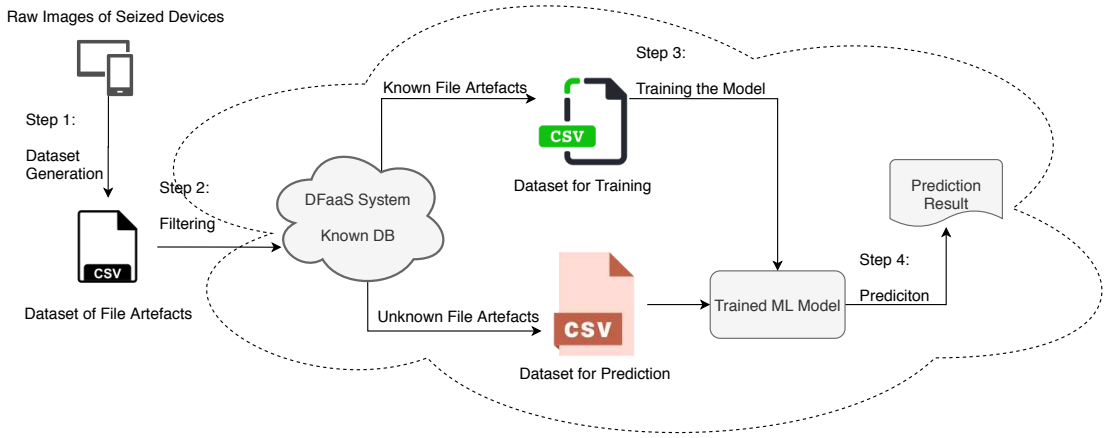


Figure 3.14: Workflow of Relevant File Prioritisation in an Investigation[a]

[a]Du, X. and Scanlon, M. Methodology for the Automated Metadata-Based Classification of Incriminating Digital Forensic Artefacts, The 12th International Workshop on Digital Forensics (WSDF), held at the 14th International Conference on Availability, Reliability and Security (ARES), Canterbury, UK, August 2019.

4. **Prediction** - The categorisation of the previously unencountered files are predicted by the trained model (i.e. if it is relevant or suspicious to the investigation).

Through the above process, from the raw image input, an initial automatic analysis result can be performed automatically. The output is a prediction of each artefact as suspicious or not to assist the investigator in identifying the file artefacts likely relevant to the investigation.

3.7.2 Relevancy Score Determination

There are different options for the input data to determine the relevancy score. They can be applied to different analysis scenarios and investigative objectives. When the files requiring analysis are different types, only the metadata and timeline events are selected as the input, the possible features extracted from pictures and text are different. In the other scenario, the analysis files to be analysed are the same type, the file data can be part of the input features for training the model. There are different approaches to combine these two sources of input, as shown in Figure 3.15.

- **RS1 - Metadata and/or Timeline Event**: These input features can detect files with similar usage patterns, as metadata and timeline events record the user's behavioural actions on files. It is for a single type of file relevancy determination, e.g., in child abuse material investigation, the analysis would focus on image files.
- **RS2 - File Data**: The most common application is the use of file data as input. It has

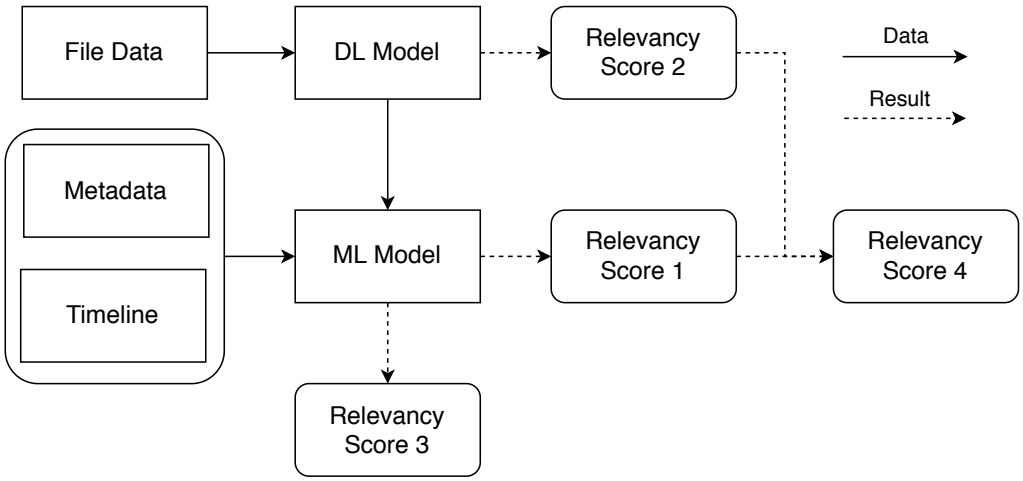


Figure 3.15: The Input for Relevancy Score

been successful in training models for recognising objects recognition in pictures (e.g., cat, dog, etc.). Therefore, it is easy to think to use to train a model to recognise gun or drug. The application in digital forensics can fit the investigation problem better.

The analysis is performed on user-created files such as documents, images, videos, audios, etc. It takes advantages of NLP and computer vision (CV) techniques to analysis documents and pictures. For example, the pre-trained model **ResNet-50**[7] model can be applied on the collected dataset. The output can be used directly or as an input (together with metadata and timeline event) to the machine learning model.

- **RS3 - Metadata and/or Timeline Events plus Output of Deep Learning Model**: The file content can be used as input to identify similar based on both file data and metadata. Firstly, train an image classification model. The output of this model can be used for the input of the next. Classifications can be converted to relevancy scores and subsequently used as the input for the next stage of machine learning model training. The file data, in this case, has less influence in determining the final relevancy score compared to these approaches.

- **RS4 - Relevancy Score 1 plus Relevancy Score 3**: The two scores can be directly summed to be used as the files' final relevancy scores. File data, in this case, has more influence in determining the final relevancy score comparing to RS3.

3.7.3 Adding Pre-trained Models

As described, file data (document or pictures) can be automatically processed by machine learning models. The application of deep learning has improved the performance in computer vision. Training deep learning models is expensive. However, there are a variety of pre-trained models available. It is called "Transfer Learning", which enables the use of pre-trained models

[7] $https://pytorch.org/hub/pytorch_vision_resnet/$

from other projects. Applying a pre-trained model, or developing models to work together with pre-trained models can be used for feature enhancement.

ResNet-50 is a convolutional neural network that is 50 layers deep. The project ships with a pre-trained version of the network trained on more than a million images from the ImageNet dataset. The pre-trained network can classify images into 1000 different object categories, e.g., keyboard, mouse, pencil, and animals, etc.

The prediction result by the pre-trained models could be used in the approach in two ways: one is to be used for feature expansion, i.e., to add labels to multimedia content, the other is use the result to the relevancy score generation function.

3.7.4 Relevancy Score Generation

Known files detected during the acquisition phase are used as training data for (supervised learning with binary classification). Subsequently, the unknown files are classified by the model to relevant/not relevant. In general, classification algorithms are used with discrete data, and the output variable must be a discrete value. Classification algorithms are used to predict/classify the discrete values such as male or female, true or false, spam or not spam, etc. A relevancy score to rank artefacts requires a regression model.

In the context of most investigative scenarios, in order to guarantee not missing out on any important information, a relevancy score based ranking for investigation is more helpful. However, regression algorithms are used to predict continuous values such as price, salary, age, etc.

In linear models, the target value is modelled as a linear combination of the features. The linear classification functionality of *scikit-learn*[8] offers functions for coefficients. Coefficient of the features is the decision function for optimal modelling. As shown in Figure 3.16, this research uses the coefficient for relevancy score calculation.

- **Classification/Regression Algorithms**

  There is a range of algorithms that can be applied both for both classification and regression problem. They are evaluated for relevancy score generation in this research and include, SVM, Decision Tree and RandomForest.

- **Relevancy Score Generation**

  Coefficients can be obtained from linear modelling, i.e., Linear SVM, Logistic Regression, etc. Random Forest modelling also affords flexible determination of the significance of each feature. The sample script below shows how the coefficients of the model are acquired:

[8]https://scikit-learn.org/

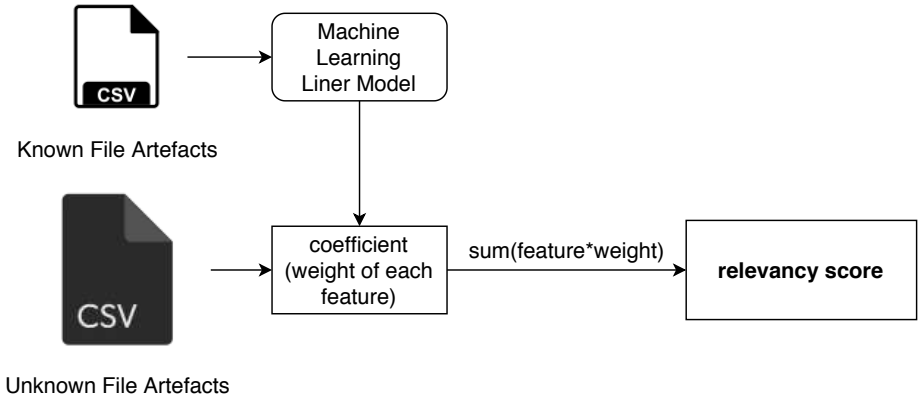


Figure 3.16: Relevancy Score Generation

```
from sklearn import svm
svm = svm.SVC(kernel='linear')
svm.fit(features, labels)
svm.coef_
```

To use relevancy determined combined with output from the pre-trained image classification model, the predicted result can be used to determine the relevancy score.

$$combined\_RS = 0.8 * event\_RS + 0.2 * content\_RS \tag{3.1}$$

This approach is mainly for detection of files with similar actions to illegal files, so only 20% of the score is determined by the file content. The weightings can be adjusted as necessary when it is applied in the real world investigation.

3.8 Summary

This Chapter presented a methodology for automated digital evidence processing approach designed to compliment on a deduplicated digital evidence acquisition and storage system. This work applied the existing data deduplication approach in digital evidence processing but improved the operation by ensuring a forensically sound reconstruction component. The file data hashing as it is acquired facilitates the detection of known illegal files at the earliest phase of the investigation possible.

The proposed analysis approach for ranking files by the IR relevancy to assist investigators can help shorten the analysis process. Files acquired by the system are categorised as known benign, known illegal, and unknown. Machine learning biased modelling classifies files into

relevant or benign. A relevancy score is generated by the trained classification model. Both the data used for training is classified by expert human analysis. It's an iterative model that is frequently updated during the analysis phase using both human expert classifications, and human expert acceptance/denial of the relevancy predicted.

A large number of disk images are required for testing the methodology to evaluate its viability performance and stability. TraceGen solves the problem of arduous, manual generation of disk images with realistic user traces and wear-and-tear. It emulates user actions and file operations executed on suspect devices. EviPlant employs evidence packages in order to make the generation of many different disk images more efficient. In addition, it saves storage space on disk and makes network transmission of disk images easier.

Chapter 4: **Test Disk Image Generation**

4.1 Overview

Digital forensic test images are commonly used across a variety of digital forensic use cases including education and training, tool testing and validation, proficiency testing, malware analysis, and research and development. Using real digital evidence for these purposes is often not viable or permissible, especially when factoring in the ethical and in some cases legal considerations of working with individuals' personal data. Furthermore, when using real data it is not usually known precisely what actions were performed when i.e. what exactly was the 'ground truth'. The creation of synthetic digital forensic test images typically involves an arduous, time-consuming process of manually performing a list of actions, or following a script 'story' to generate artefacts in a subsequently imaged disk. Besides the manual effort and time needed in executing the relevant actions in the scripted scenario, there is often little room to build a realistic volume of non-pertinent wear-and-tear or 'background noise' on the suspect device, meaning the resulting disk images are inherently limited and to a certain extent overly simplistic.

This chapter introduces the generation of test disk images containing realistic features including regular wear-and-tear, background noise, and the actual digital traces to be discovered during the investigation. The disk images used in this research are Windows 7/8/10 disk images. Emulated wear and tear actions (surfing the Internet, installing software, downloading files from browsers, etc.) were conducted in a virtual environment.

4.1.1 Choice of Technologies

The virtualisation technology used for this research was VirtualBox, owing to its cross-platform compatibility. This could also be said for VMware, but as VirtualBox is freely available, this approach could be used by any organisation regardless of any budget limitations. Qemu is another option, but the setup process is slightly more complex and the intention was to produce images automatically that could be easily later supplemented with later human-generated actions.

The operating system chosen as the guest is Microsoft Windows as this still represents the majority of end-user computers[1], and therefore the majority of systems seen in digital forensic

[1] 87.36% at the time of writing (https://netmarketshare.com)

laboratories.

4.2 File Data for Disk Image Creation

The `Govdocs1`[2] dataset from `digitalcorpora.org` was used as a source of benign files in this research. It consists of almost 1 million freely-redistributable files of various formats. Types of files within the "seized device" for analysis are:

- Documents: txt, doc, pdf
- Pictures: png, jpg
- Container: zip
- Python scripts: py

For picture content, images from Google Images, Wikipedia and a data set for child age estimation were used. Pictures are downloaded from the results of keyword searching from Google (as shown in Figure 4.1). These files are used for testing the deep learning classification model.

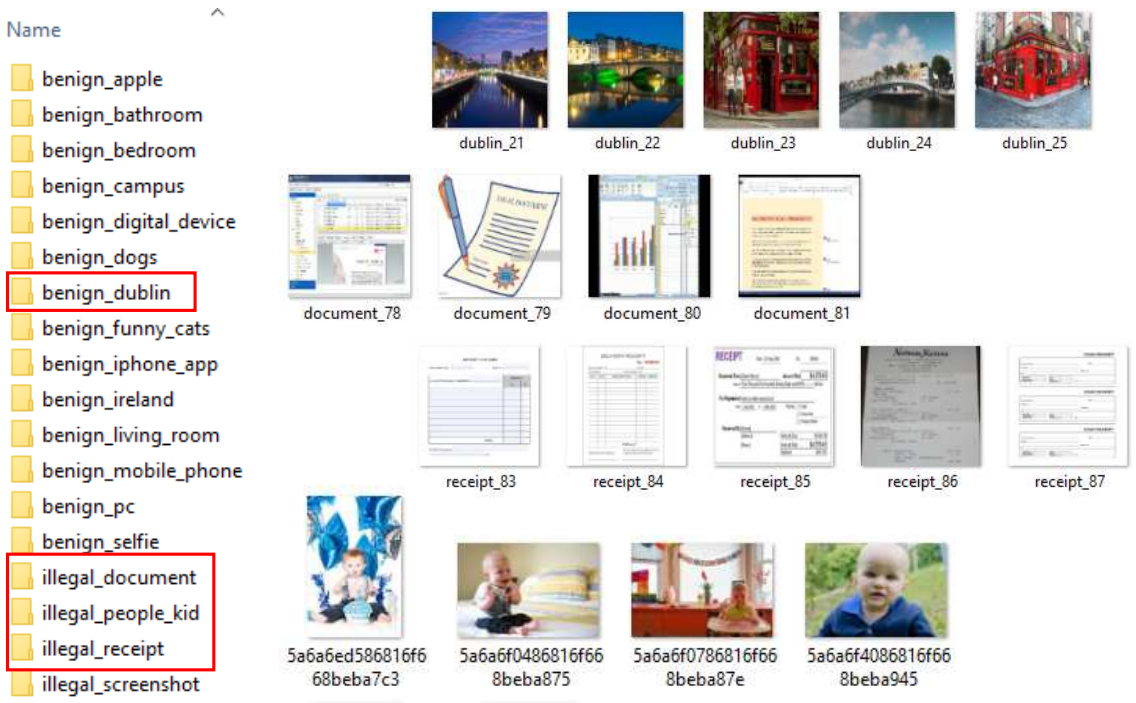


Figure 4.1: Sample Pictures Used to Populate Test Image

[2]https://digitalcorpora.org/corpora/files

4.3 EviPlant: Creation, Manipulation and Distribution Disk Image

4.3.1 Overview

Typically, the creation of digital forensic challenges requires an overly arduous effort on the part of the user/educator to ensure their viability. In addition, the storage and distribution step also requires significant time. EviPlant is a system designed for the efficient creation, manipulation, storage and distribution of challenges for digital forensics education and training.

The system relies on the initial distribution of base disk images, i.e., images containing solely base operating system installations. Firstly, the base system is booted up, and the desired activity is manually conducted. Subsequently, a "diffing" of resultant image and the base image is performed ("diffing" in this context stemming from the `diff` tool in *nix systems [157]). This diffing process extracts all modified artefacts and associated metadata and stores them in an "evidence package". Evidence packages can be created for different personae, different wear-and-tear patterns, different emulated crimes, various levels of complexity, etc.

4.3.2 Evidence Packages

The fundamental building blocks for EviPlant are evidence packages. The current approach typically requires the maintenance of large collections of complete disk images. Using EviPlant, the approach changes to creating and curating a variety of evidence packages. The creation of these evidence packages relies on the comparison or diffing of two disk images.

The diffing tool is provided with two inputs; the base image used to emulate the specific user activity and the subsequently modified image containing all traces of the "crime", wear-and-tear, or persona emulated. The tool scans through the modified image and extracts all modified and newly created artefacts (and their associated metadata) into an evidence package, i.e., eliminating all artefacts from the modified image also present on the base image.

In this manner, evidence packages are capable of being created to capture small events, e.g., a boot cycle of the operating system, or large events, e.g., a complex usage pattern to build a complete emulated user persona over an extended period of time.

Evidence packages contain all the digital artefacts (files, file fragments, slack space, etc.) and associated metadata created during the emulation of the crime. The artefacts contained within evidence packages fall into two categories, which combine to capture all the necessary data modified from the base image after performing a specific task or set of tasks:

1. **"Black Box" Artefacts** – These are simple artefacts encapsulating a modification from the base image. In order to use these evidence packages in a classroom deployment scenario, nothing need necessarily be understood of their construction in order to inject them into the base images. The answer set for these challenges is created by the educator (effectively the script they followed while performing the actions to be discovered).

2. **"Reverse Engineered" Artefacts** – The artefacts and metadata contained within this package are understood in their entirety – effectively reverse-engineered. As a result, their manipulation is possible, e.g., SQLite databases for Internet browsing history, VoIP application call logs, etc. In the scenario of using multiple evidence packages for a single challenge, packages with overlapping artefacts must be manipulable to ensure that the traces from each package are integrated into the final disk image.

Evidence Package for Storage Savings and Easier Distribution

Figure 4.2 shows the storage-saving by using this approach. A complete disk image of Windows 7 with user traces is 9.99 GB. An evidence package is generated (size is 2.06 GB) preserving the modified files from the base image. When using zip for the evidence package, it can be compressed to 203 MB. This evidence package is easier to store and distribute. Images with complete user traces can be re-created by using a base Windows 7 image and the evidence package. When many disk images are stored using this approach, a substantial reduction in storage space can be achieved.

4.3.3 Testing of Diffing and Injection

To demonstrate the viability of the proposed system, the two main components of the system (namely the diffing tool and the injection tool) were developed in Python using the `pytsk`[3] library for disk image processing. `pytsk` is a python wrapper for The Sleuth Kit[4]. For testing purposes, a Windows 10 virtual machine was created and used as the base image. For each test, the base image was cloned, booted and user activity was emulated on the machine.

In terms of the testing of the diffing tool, a variety of usage patterns were tested ranging from a single boot cycle of the virtual machine to an extended session involving internet browsing, application installation, file downloading, multiple boot cycles, etc. In each scenario, the diffing tool discovered all of the modified artefacts relating to recorded usage, including a number of operating system files that were modified during the regular usage of the virtual machine (e.g., `$MFT`, `pagefile.sys`, etc.). These artefacts and associated metadata were output into an evidence package.

To assess the injection methodology, the base operating system was booted and the injection

[3]https://github.com/py4n6/pytsk

[4]http://www.sleuthkit.org/

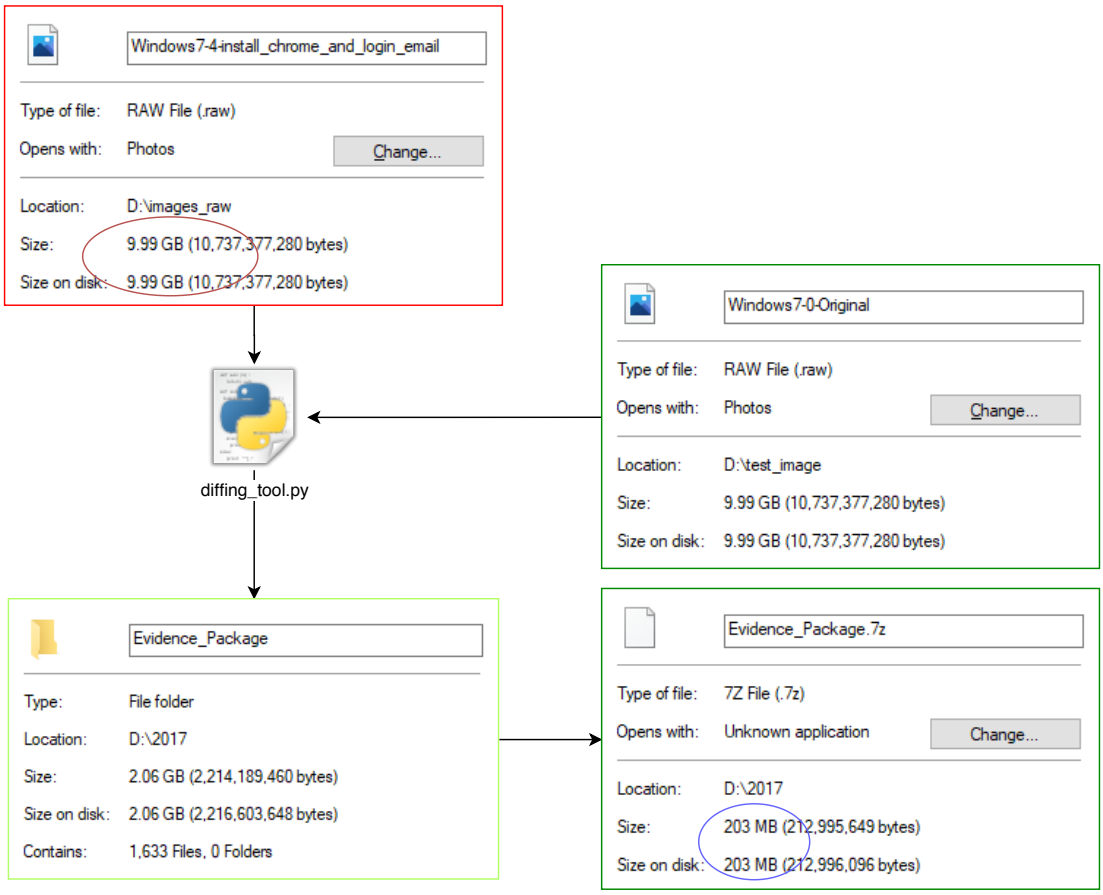


Figure 4.2: Diffing Tool Storage Savings for Evidence Packages

tool was run locally on the live machine to perform a logical injection of the artefacts. The injection tool took each individual artefact and added it to the virtual disk. In the eventuality of conflicting artefacts, the version from the base image was overwritten by that from the evidence package. To demonstrate the viability of the results generated disk images and to confirm the injection of the necessary artefacts, these images were subsequently analysed using EnCase and, unsurprisingly, the pertinent planted evidence was identifiable and recoverable.

4.3.4 Summary

EviPlant provides solutions to distribution of digital forensic images through the use of evidence packages and makes the creation of complete digital forensic challenges easier. More digital forensic challenges are capable of being stored in the same disk capacity than if entire disk images were used.

Manually creating background noise in a disk image to increase its benign activity can include simple file browsing and artefact generation over multiple artefact types, e.g., injecting web history, file system artefacts, event logs, registry files. Maintaining these artefacts to be evidentially consistent with each other is a difficult manual tack. The EviPlant approach,

while potentially increasing the reusability of manual effort, does not address the issue of reducing the manual effort in creating the evidence packages to begin with.

4.4 TraceGen: Automated User Action Emulation

4.4.1 Overview

TraceGen's focus is to address the aforementioned requirement for manual user activity emulation. TraceGen is an automated system focused on the emulation of user actions to create realistic and comprehensive artefacts in an auditable and reproducible manner. The framework consists of a series of actions contained within scripts that are executed both externally and internally to a target virtual machine. These actions use existing automation APIs to emulate real user behaviour on a Windows system to generate realistic and comprehensive artefacts. These actions can be quickly scripted together to form complex stories or to emulate wear-and-tear on the test image over a long period of time.

4.4.2 Virtual Machine Configuration

The automation is performed using a series of Python scripts to perform user actions. Python was chosen for its advantages for rapid prototype development and the large range of libraries available for emulating user activity. The script executes on the host and controls the guest machine; from the boot, performing internal and external actions, to shut down. Internal scripts are executed inside the VM to automate internal user actions.

In addition to the internal and external scripts and the stories, at present, the VM must also be configured in a specific manner that is conducive to simulated user actions. The following changes were made to the VM to allow reliable user action emulation: 1), install *VirtualBox Extension Pack* on Host OS; 2), install *Guest Additions* on Guest OS; 3), install Python on Guest OS; 4), install related Python libraries and any other dependencies, e.g., SendKey; 5), install app needed in the action emulation on Guest OS, e.g., Google Chrome. There are also several other more subtle changes that are needed. For example, for now, it is necessary to auto-login the user, rather than providing a password. Also, as the system may be running for several days, the power settings of the guest and the host were modified to avoid sleeping or powering off displays as this could interfere with the user emulation process.

4.4.3 File Provenance and Interactions

The generated disk should contain user files with usage pattern. Applications are installed on the guest OS, e.g., Google Chrome, WinSCP[5], Sublime Text[6], etc.

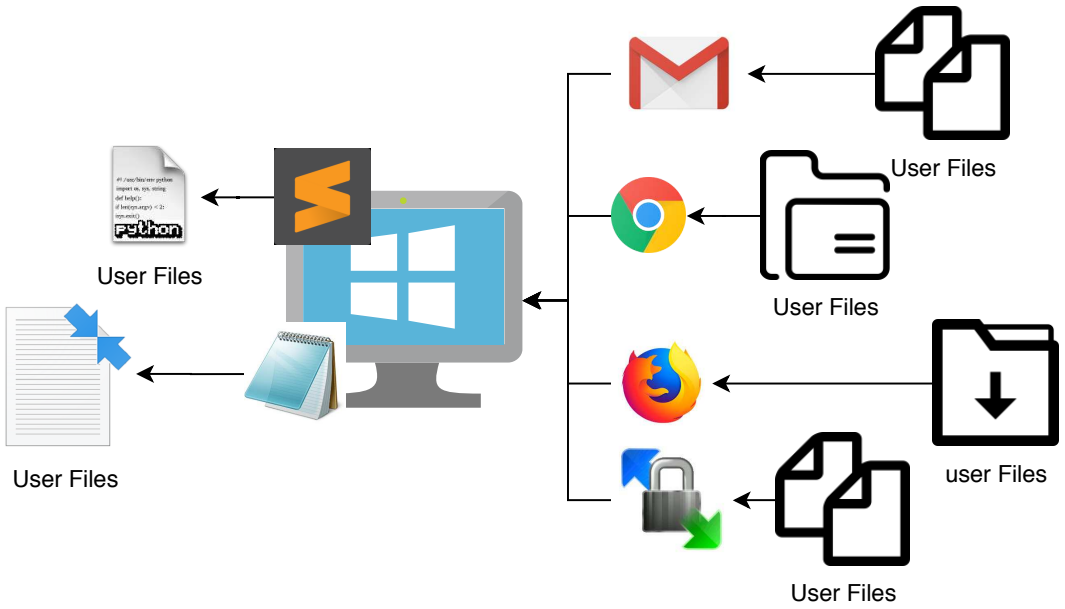


Figure 4.3: User Action of Files on the Disk

There are several developed dynamic actions that can combine together to perform more complex, compound interactions to files on a disk:

- Creation
  - Downloaded from the web
  - Downloaded from WinSCP
  - Created by Notepad
- Access
  - Opened with SublimeText
  - Notepad
  - web browser
- Modification
  - edited by SublimeText
  - edited by Notepad

[5]The main function of WinSCP is secure file transfer between a local and a remote computer
[6]Sublime Text is programming focused text editor

- Other Operations
  - Executed by python
  - Zip or unzip

4.4.4 User Action Emulation

pywinauto facilitates the automation of the Windows GUI using the Win32 API. It allows Pythonic interaction with GUI components. Example usage of the *pywinauto* module to launch Chrome by opening the Start Menu and typing 'Chrome' followed by the enter key is shown in Figure 4.4.

```
def launch_chrome():
    pywinauto.keyboard.send_keys('{VK_LWIN}')
    time.sleep(0.5)pywinauto.keyboard.send_keys('chrome')
    time.sleep(0.5)pywinauto.keyboard.send_keys('{ENTER}')
    time.sleep(0.5)
```

Figure 4.4: User Action Emulation: Launch Chrome

Figure 4.5 presents the function for transferring data into the virtual machine, through downloading files from a server. It can be used for file creation on the disk.

Facilitating easy new module development is paramount for the success of this framework. Currently, modules have been developed to:

- generates a new file with notepad in a specific location on the disk with customised content;
- access, edit, delete a specific file;
- copy files from one directory to another;
- use Google to search for a particular keyword, and then viewing a subset of the results over a defined period of time;
- send and read emails from Gmail accounts (Python module for using the standard SMTP and IMAP email protocols).

4.4.5 Continuous Usage Trace Generation

There are two modes by which a story can be executed. The first is the *live simulation* mode. In this mode, the controlling script checks the time for the next action to be performed

```
def download_server_file(hostname, username, password):
    import pysftp
    conpts = pysftp.CnOpts()
    conpts.hostkeys = None
    local_dir = utils.random_path_l()
    remote_dir = utils.random_path_r()
    print (local_dir)
    print (remote_dir)
    with pysftp.Connection(hostname, username=username, password=password, cnopts=conpts)
          as sftp:
        dir_items = sftp.listdir(remote_dir)
        os.path.exists(local_dir) or os.makedirs(local_dir)

        remote_dir += dir_items[random.randint(0, len(dir_items))-1]
        print(remote_dir)
        local_dir += dir_items[random.randint(0, len(dir_items))-1]

        os.path.exists(local_dir) or os.makedirs(local_dir)

        dir_items = sftp.listdir_attr(remote_dir)
        for item in dir_items:
            print (item.filename)
            remote_path = remote_dir + '/' + item.filename
            local_path = os.path.join(local_dir, item.filename)
            print(remote_path)
            print(local_path)
            sftp.get(remote_path, local_path)

    return {"timestamp": utils.datetime2str(datetime.now()), "operation":
        "download_server_file", "notes": "hostname=" + hostname+"
        download_file="+local_dir}
```

Figure 4.5: The Method to Create Files on the Disk

against the host (assumed to be correct) and the actions are performed when the scheduled time arrives. This means that the actions are carried out in real-time but in an automated way. This mode provides the advantage that as all actions are performed in real-time resulting in all timestamps, whether from the virtualised system or remotely fetched content, will be consistent. The disadvantage is that to generate several months of activity, it would take several months. Therefore, the second mode is a *compressed-time simulation*. In this mode, for each action, the clock of the virtual machine is adjusted to the specified time, then the action carried out. This has the disadvantage that there will be timestamp artefacts that are not consistent, but it does allow a large amount of activity to be synthesised in a very short period of time.

The function in Figure 4.6 shows how the developed method facilitates the setting of the system date and time for action emulation.

Evaluation of the Compressed-time Simulation Mode

Setting the system time before booting the machine enables the emulation of usage over a much longer period, i.e., weeks, months or years of activity emulated in just hours or days. An input CSV file defines the actions to be executed on the machine. A random method

```
def set_f_bios_time(vm_name, datetime_to_set):
    from datetime import datetime
    datetime_now = datetime.now()
    datetime_to_set =datetime.fromisoformat(datetime_to_set)
    time_delta =  datetime_now - datetime_to_set
    command = r'vboxmanage modifyvm%s --biossystemtimeoffset -%d'% (vm_name,
        time_delta.total_seconds()*1000)
    f = Popen(command, stdout=PIPE, shell=True).stdout
    data = [eachLine.strip() for eachLine in f]
    return data
```

Figure 4.6: The Method to set the System Time to a Given Date and Time

can be used to make sure repeated operations would occur at different times. The same *component* or *compound* actions can result in different artefacts being generated due to the specified arguments, e.g., which file to open.

Two sample experiments are outlined below:

1. Setting the system time to a point in the past and repeat simple actions, i.e., 1) set the system time; 2) boot the VM; 3) create a new file; 4) shut down the VM.
2. Setting the system time to a point in the future, repeat more complex actions at each time: 1) set the system time; 2) boot the VM; 3) create a new file; 4) copy `*.png` file from one folder to another; 5) use browser 6) run *WinSCP* and log in; 7) shut down the VM.

The scheduled user actions are saved into an operation list in a CSV file, as shown in Figure 4.7. For running each action, external control, internal action and associated parameters are required.

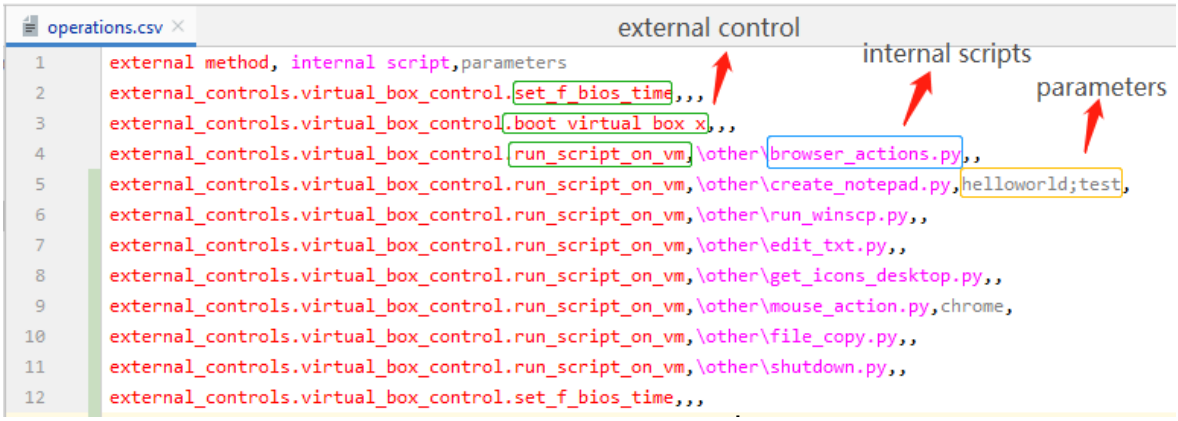


Figure 4.7: Sample Input CSV File

Analysis of the Trace Generation by a Comparison of Timeline Events

Aforementioned two sample experiments running by each time before booting the VM, the system time is set to a new date. The operations were repeated 20 times, each time the date

was set to a different day in September 2019, assuming the computer was not used every day. Two "Super Timeline"[7] of the disk were generated for the disk at before and after running TraceGen on the VM. A comparison of the two timelines can find out changes in the device.

One interesting analysis is on the diverse value generated for each property. The diverse value for the date relates to how many instances days this machine was used, i.e., as long as the machine is being used, correlating timestamps are generated. The file name count represents how many different files are retrieved from in the machine. Inode count represents the file number on file system level. Table 4.1 shows the value counts for each of these different properties. Before the story was executed, in the generated timeline, the number of timestamp dates is 189. After the story's execution, it is 204. The increment on the count of filenames is 11,099 representing the number of new were generated during the story's emulation. Table 4.2 shows the increment of events from each source. These results demonstrate the effectiveness of this tool for the generation of background noise user traces over time.

Event Property	Before Story	After Story	Increment
date	189	204	15
file name	90,941	102,040	11,099
inode	31,350	36,139	4,789

Table 4.1: Volume of Filesystem Modifications Before and After Story Execution

Event Source	Before Story	After Story	Increment
EVT	183,628	246,653	63,025
REG	81,181	125,598	44,417
WEBHIST	27,869	30,689	2,820
Log	2,753	3,160	407
PE	1,019	1,032	13
LNK	621	852	231
OLECF	348	348	0
Total	297,419	408,332	110,913

Table 4.2: Event Counts from Different Sources

Two sample continuous running tests are outlined. The first is to set the system date and time in the past and the second is to set it to a point in the future. These were dates in September 2019 and October 2019 respectively. Figures 4.8 and 4.9 shows the number of event changes detected for each day. It can be seen that the day configured in the story results in corresponding changes on the timestamp. Also, as can be seen, the number of timestamps increased after the story's execution emulation. This is a result of later actions updating the same specific files modified by previously executed actions.

[7]A super timeline contains timestamps from the filesystem and various log files on a device.

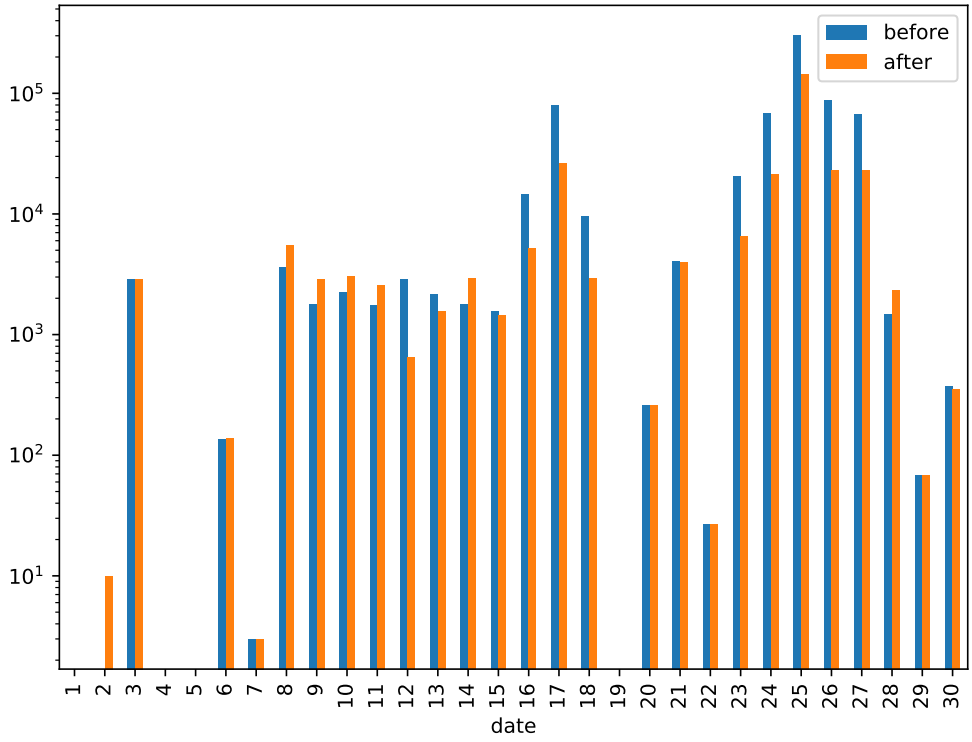


Figure 4.8: Event Counts of Each Day in September[a]

[a]Du, X., Hargreaves, C., Sheppard, J., and Scanlon, M., TraceGen: User Activity Emulation for Digital Forensic Test Image Generation, Forensic Science International: Digital Investigation, ISSN 2666-2825, September 2020.

4.4.6 Summary

TraceGen enables the creation of a substantial amount of background noise, wear-and-tear, and other non-pertinent actions on the device. In addition, complex stories can be quickly and easily performed on the virtual machine. These stories can be queued or looped to create substantially large traces as required. The experimentation and analysis presented in this section verified that the TraceGen framework can operate in a viable manner. This framework enables 1) a large amount usage traces generated in a compressed time period, 2) usage traces generated at a specific date in time, and 3) the generated actions can be across a diverse set of event categories.

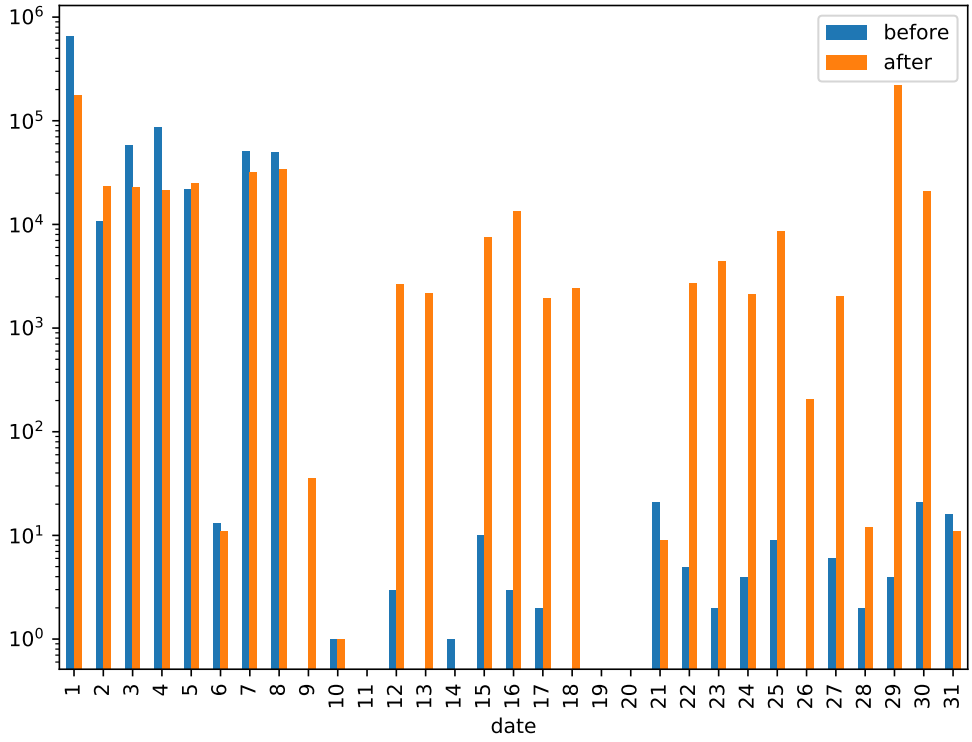


Figure 4.9: Event Counts Increment of Each Day in October[a]

[a]Du, X., Hargreaves, C., Sheppard, J., and Scanlon, M., TraceGen: User Activity Emulation for Digital Forensic Test Image Generation, Forensic Science International: Digital Investigation, ISSN 2666-2825, September 2020.

Chapter 5: **Deduplicated Digital Evidence Processing**

The goal of implementing the proposed deduplicated method of acquisition is focused on the reduction of data to be transported, stored and analysed. The system is split into two processes that work together to both reduce the time taken to get from the acquisition stage to making the data available as evidence while reducing the volume of storage required for each acquisition. Eliminating the need to transfer duplicated files from every device encountered both speeds up each acquisition and removes the need to analyse the files that have been marked as a duplicate. The process is designed to be simple to execute facilitating a minimal amount of training required to being to perform acquisitions removing the need for specialists on-site to collect the data.

5.1 Overview

To begin the data reduction, each file contained within the suspect device must undergo preprocessing. As the system begins to discover each file, a line of communication is opened between the client and the server. As each file is discovered a hash is calculated; which is then transferred to the server. This is checked against a database of all files previously discovered and is only uploaded if it has not been acquired before. Whether a file is a duplicate or not, its associated metadata is uploaded, which can be used in both the file's analysis or in the reconstruction of the entire disk, as outlined in Figure 5.1.

This system employs *MongoDB*[1] to store metadata of forensic images and the files are stored on disk. *pymongo* is a Python library for interfacing with MongoDB and is employed by this system. The data preserved (image metadata, file metadata and file artefacts on the server) has been introduced in Section 3.4.

[1]MongoDB is a free and open-source cross-platform document-oriented database program.

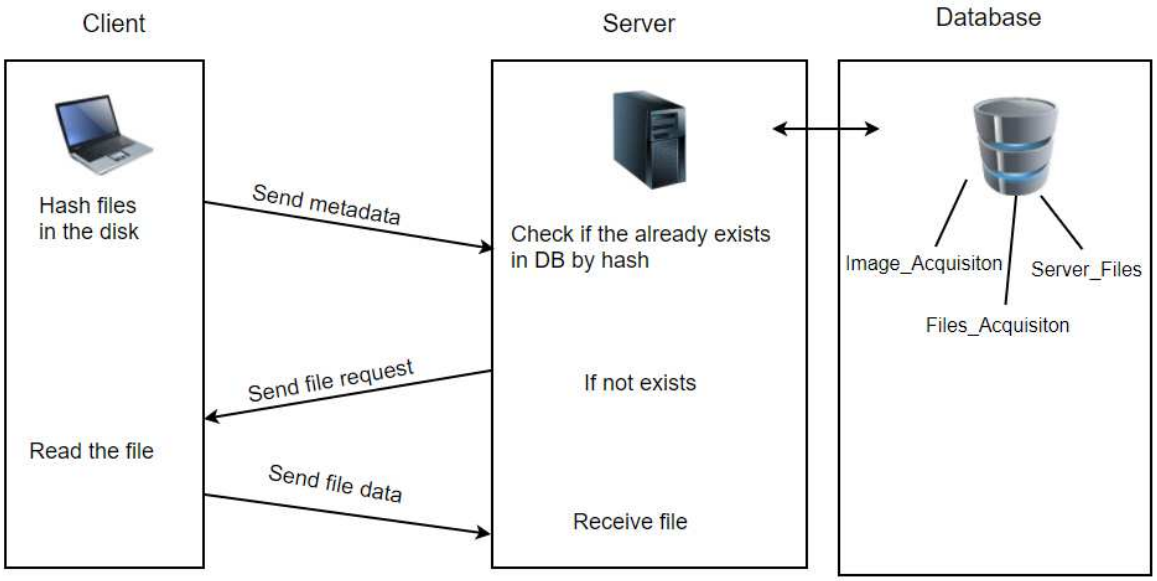


Figure 5.1: Deduplicated Evidence Acquisition Process[a]

[a]Du, X., Ledwith, P., & Scanlon, M. (2018). Deduplicated Disk Image Evidence Acquisition and Forensically-Sound Reconstruction. The 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications (IEEE TrustCom-18).

5.2 Prototype Setup and Test Data

The prototype system was built on a server with Ubuntu 16.04.2 LTS operating system, Linux 4.4.0-121-generic kernel, and x86-64 architecture.

For testing the performance of evidence acquisition and reconstruction, disk images were created with various file systems. Test data specifics are shown in Table 5.1. The images were created through *dd* copying data from a USB drive. Various duplication ratios were created to test the performance of duplicated data.

5.3 Results: Average Acquisition Speed

Through iterative acquisitions of the images, measurements of the acquisition speed and time were recorded, alongside the acquisition date and time, and the encountered duplication ratio. Figure 5.2 shows the average acquisition speed for each image. Each of the created images was acquired several times. Analysis of the results identified three different factors influencing the speed:

[2]Du, X., Ledwith, P. and Scanlon, M. (2018). Deduplicated Disk Image Evidence Acquisition and Forensically-Sound Reconstruction. The 17th IEEE International Conference On Trust, Security And

No.	Image Name	Image Size	No. of Files	File System	Note
01	A01.dmg A02.dmg A03.dmg	144 MB	477 491 513	FAT	Disk image all with pictures
02	D201.raw	2 GB	36	NTFS	Disk image with pictures, videos, audios and documents
03	D801.raw	8 GB	234	NTFS	Created through `dd` from 8 GB USB key with pictures
04	D1601.raw D1602.raw D1603.raw	16 GB	244,025 24,412 24,413	NTFS	Created through `dd` from 16 GB USB key with pictures, three separate images with specified number of files
05	Windows_PE.img	2 GB	1,645	NTFS	Created by install Windows PE on USB key and then `dd`
06	Windows7-1.raw	10 GB	48,401	NTFS	Created by `qemu-img` converting `vhd` to raw
07	Windows7-2.raw	10 GB	49,295	NTFS	Incorporates operations (surf Internet, create new documents) on virtual machine
08	Windows7-4.raw	10 GB	50,382	NTFS	Incorporates operations (install software, create new documents) on virtual machine
09	Windows7-3.raw	10 GB	58,547	NTFS	Incorporates operations (surf Internet, create new documents) on virtual machine
10	Windows8.raw	10 GB	81,163	NTFS	Original Windows 8 image

Table 5.1: Test Disk Images

1. The Image - The overall size of the image, the number of files on the image, and the ratio of small files compared with large files;

2. Duplication Ratio - This determines how much data has to be transmitted to the server;

3. Execution Environment - Network bandwidth, client computer processing speed, storage hardware performance, etc.

Figure 5.3shows a comparison of two speeds, one is the actual hardware read-speed, the other is the *effective speed*, i.e., the throughput of the system factoring in the speed enhancements

Privacy In Computing And Communications (IEEE TrustCom-18).

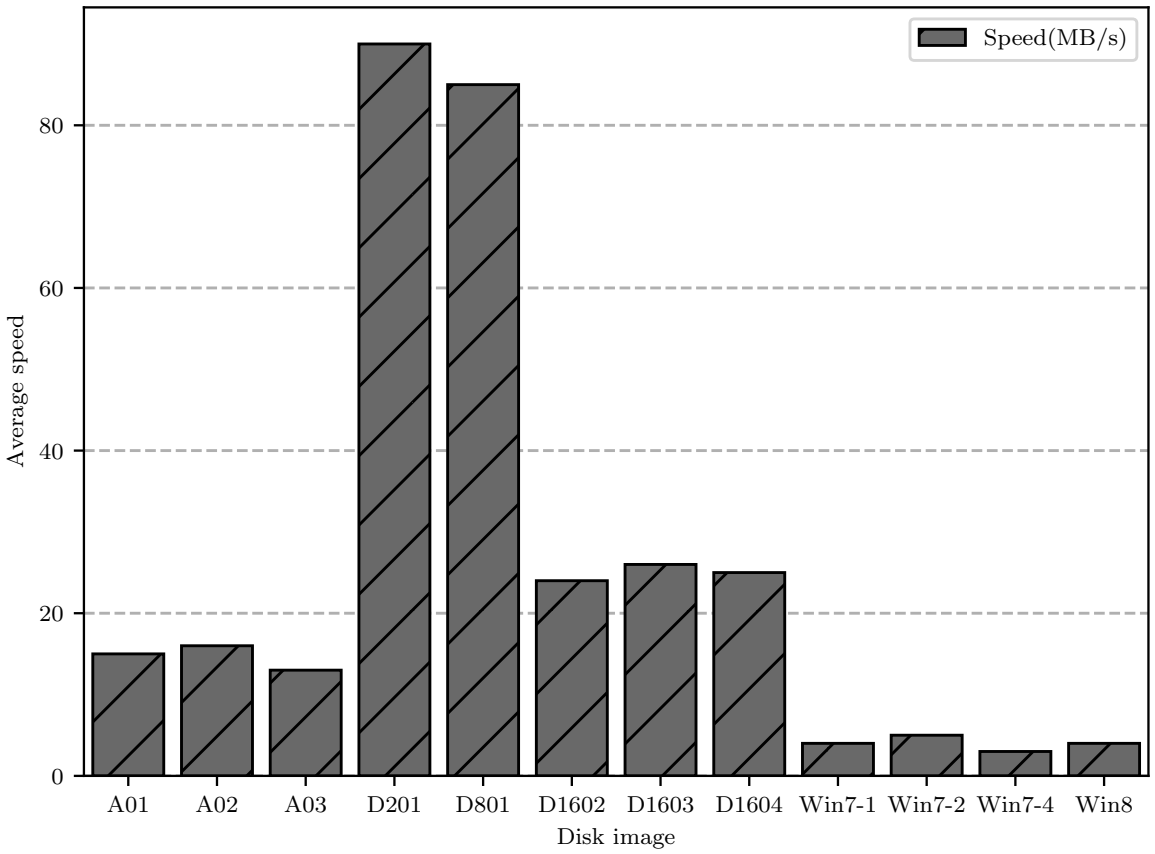


Figure 5.2: Evidence Acquisition Speed of Each Image [2]

achieved through deduplication. This effective speed is consistently faster than the actual speed. The formulas below demonstrate precisely what constitutes these two speeds (the reduced size is the original data less the duplicated data which need not be re-uploaded).

$$actual\_speed = reduced\_size\ /time\_taken \tag{5.1}$$

$$effective\_speed = image\_size\ /time\_taken \tag{5.2}$$

Data deduplication improves system speed significantly. Figure 5.4 illustrates that the higher the duplication ratio, the faster the effective acquisition speed. Test results show the speed can be ten times the disk-read speed when the duplication ratio is approximately 90%.

When interpreting the above acquisition speeds, it is important to note that some preprocessing of the data has already taken place in addition to the acquisition. For example, the file

[2]Du, X., Ledwith, P., & Scanlon, M. (2018). Deduplicated Disk Image Evidence Acquisition and Forensically-Sound Reconstruction. The 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications (IEEE TrustCom-18).

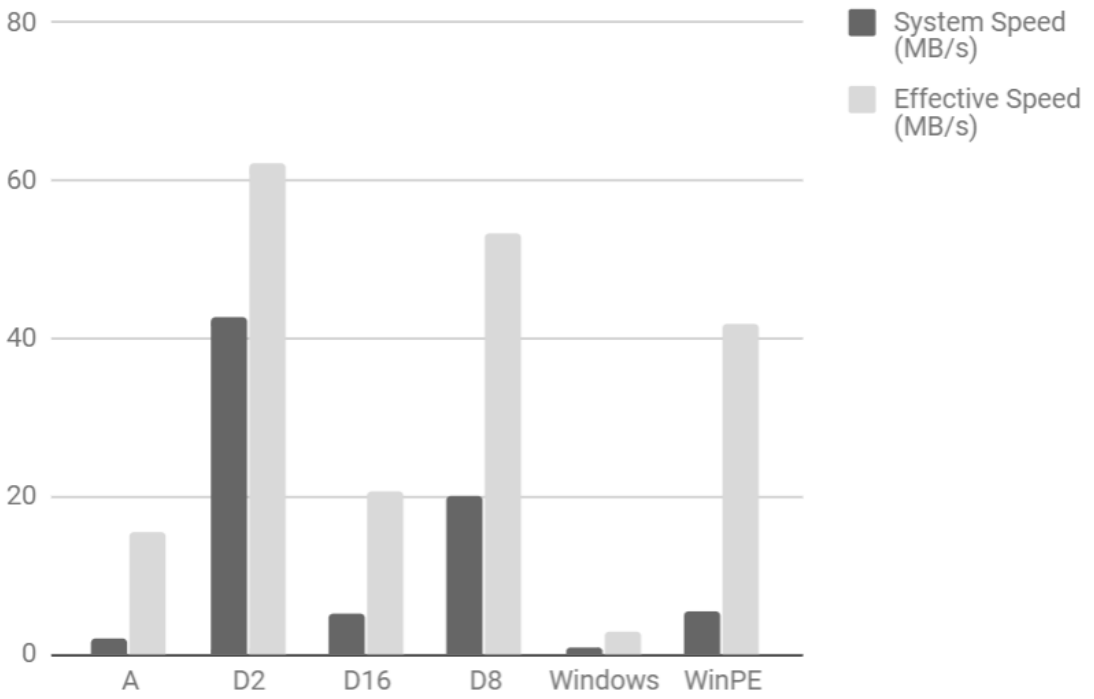


Figure 5.3: System Speed and Efficient Speed Comparison of Each Image

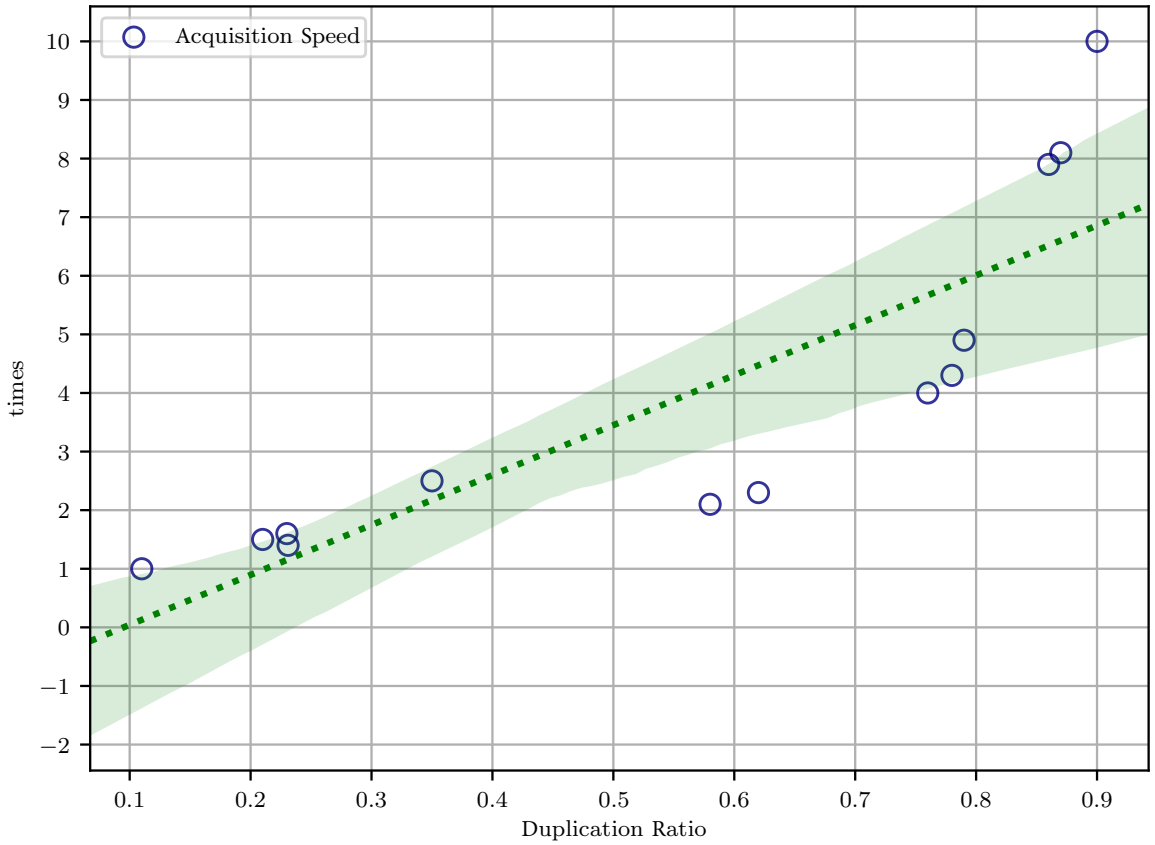


Figure 5.4: Duplication Ratios and their Impact on Speed

system metadata for each artefact has already been extracted and recorded in the database including its path, filesize, hash, block locations, etc. It is also possible to highlight known

illegal files *during* the acquisition phase of the investigation, flagging pertinent information to the investigator at the earliest stage possible.

5.4 Results: Storage Space Saved

Data deduplication not only speeds up data transmission but also saves system storage space requirements. The system is designed for a big volume of data storage. As the volume of data collected grows, the more duplicates are encountered, the more storage is saved. In this test, the first acquisition of all the images, the system acquired 20% extra storage space, because the file slack and unallocated space data on the disk are acquired by the system. When 1TB of digital evidence had been acquired, the deduplicated system saves over 32% storage space. As an increased number of artefacts are acquired, more storage savings will be possible. Figure 5.5 shows the storage-saving as the number of acquisition increase.

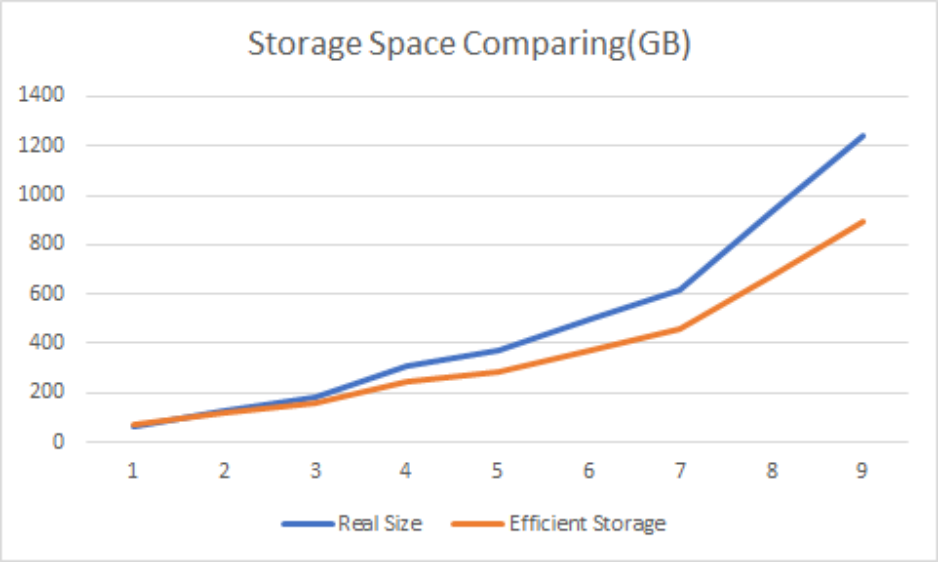


Figure 5.5: Storage Saving as Number of Acquisitions Increase[a]

[a]Du, X., Ledwith, P., and Scanlon, M. (2018). Deduplicated Disk Image Evidence Acquisition and Forensically-Sound Reconstruction. The 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications (IEEE TrustCom-18).

5.5 Results: Image Reconstruction

Hundreds of validated disk image reconstructions from the deduplicated data store have been successfully performed. Figure 5.6 shows the average speed of each disk image reconstruction.

The speed varies due to the aforementioned influencing factors. The fastest individual reconstruction attempt during testing was over 150 MB/s. The average of all the reconstruction speed is 43.78 MB/s, which can be improved upon in the future through the employment of RAID storage enabling faster disk I/O. Disk image reconstruction may only be necessary if incriminating evidence is discovered. The Windows Preinstallation Environment (PE) image used is faster than Windows image as the number of small files is significantly less. D_Image was faster than each of the others with average reconstruction speed 113.67 MB/s due to it containing a higher proportion of large files.

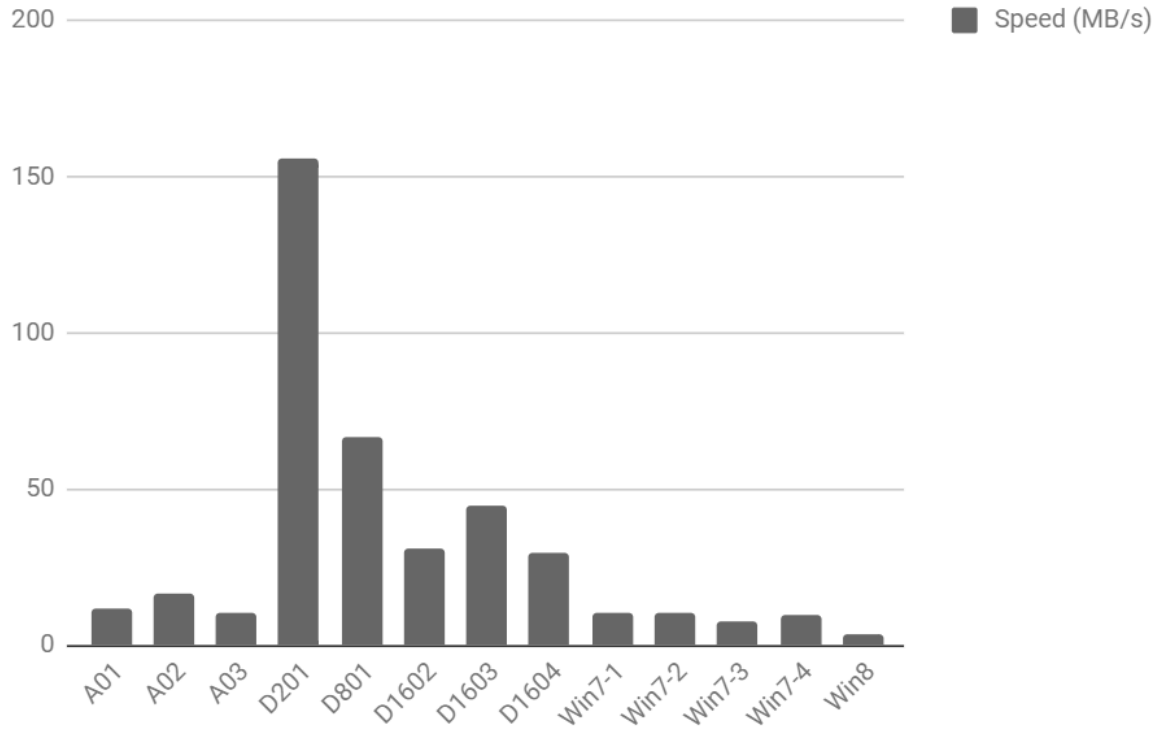


Figure 5.6: Reconstruction Speed of Each Image[a]

[a]Du, X., Ledwith, P., and Scanlon, M. (2018). Deduplicated Disk Image Evidence Acquisition and Forensically-Sound Reconstruction. The 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications (IEEE TrustCom-18).

5.6 Tests on an Improved Approach for Data Acquisition

The acquisition speed of the system varies due to the number of files, between 1 MB/s to 10 MB/s. A large number of small files (less than 100 KB) significantly slows down the data acquisition speed, because it increases the interactions between the client and server.

A local client-side data store can be implemented to check the existence of duplicates resulting in minimising the network traffic. The client acquires the complete known hash list. *numpy.setdiff1d*[3] is used for detection of known/unknown files through comparing the two hash value lists (hash values from the central database and hash value of files on the disk). In addition, the extracted unknown data (files, file slack space and unallocated space) can be compressed and sent to the server at once. This can reduce the communication overhead between the client and the server. This results in the amount of data to be can dramatically reduce.

Figure 5.7 shows the workflow of the improved deduplicated data acquisition approach. The steps of the approach are as follow:

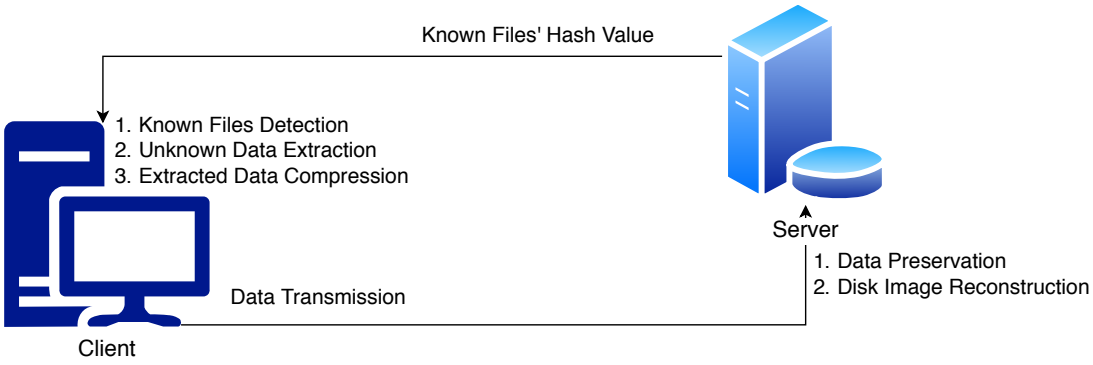


Figure 5.7: An Improved Approach for Deduplicated Data Acquisition

1. The client connect to MongoDB on the server and acquires the known files' hash values;
2. The client extracts the target disk metadata and calculates the files' hash values;
3. The client compares the disk hashes and the list of hashes from the server; known good and known illegal files can be detected at this stage;
4. The client extracts all unknown data and compresses the data;
5. The client sends the compressed data to the server;
6. The server preserves metadata and compressed file;
7. A verifiable forensically-sound disk image can be reconstructed whenever it is needed.

5.6.1 Results

The disk image Windows10.img is used for this test. Table 5.2 shows the size of data and the number of files. Originally, the Windows 10 disk contains 50,430 files, the size is 9.5 GB.

[3]Find the set difference of two arrays. Return the unique values in ar1 that are not in ar2.

The deduplicated system extracts 86,806 artefacts from the disk. The number of artefacts increased because file slack space is preserved as an artefact. The size of the deduplicated file folder is 7.6 GB. *7z*[4] is applied to compress the deduplicated files; reducing the size of the data to 1.92 GB. The compressed file then is sent to the server.

Data	Size	Number of Files
Original Disk (Windows 10)	9.5 GB	106,970
Deduplicated Files	7.26 GB	86,806
After Compression	1.92 GB	86,806

Table 5.2: Disk Image Compression Test

Experiments on the test image were conducted and the results for each step are outlined below.

1. **Metadata Extraction and File Hashing**

   The file system metadata extraction is implemented by *pytsk*; the file hashing uses python *hashlib*. This process on the test disk image took 3 minutes 41 seconds on average across multiple tests.

2. **Known/Unknown Files Detection**

   In this test, the number of known files from the server is 148,405; the number of files on the disk is 106,970. It takes 1 minute 52 seconds to detect the unknown files list, resulting in 36,931 files being identified as unknown by the system.

3. **Compressed File Transmission to the Server**

   pysftp[5] is applied for file transfer to the server. The time taken is 16 minutes 42 seconds to transfer the test disk image (9.5 GB original, 1.92 GB after compression). This speed is limited by the network used. The network of the client machine used for this test is 28.6 MB/s for download and 2 MB/s for upload.

Data compression allows further reduction of the amount of data. The test was conducted for remote digital evidence acquisition; this approach can also work for on-scene data collect.

5.7 Summary

This research explores a novel approach to collect digital evidence from a variety of devices and evaluates the storage requirements and speed through several forensically-sound evidence

[4]7-Zip is a file archiver with a high compression ratio.

[5]pysftp provides a simple interface to SFTP.

acquisitions. The factors that influence the performance of the proposed system were also evaluated. From the analysis of the results, the summary is as follows: i) Acquiring data from suspect devices is complete and verifiably accurate; ii) Forensically-sound complete disk image reconstruction was achieved for all test data; iii) As a desirable product of the deduplicated acquisition process, evidence preprocessing has also taken place including metadata extraction and artefact hashing; iv) The performance is better for disk images containing a high proportion of large files; for complete operating system images, the speed achieved shows great promise for the technique, but still needs to be improved to be viable. In a remote acquisition scenario, i.e., acquiring a forensic image over the Internet, the acquisition speed is reasonable when compared with typical broadband upload speeds.

5.7.1 Benefits of this Approach

The benefits of transitioning to a cloud-based deduplicated digital forensic process model include: i) always up-to-Date software resources; ii) pooled hardware resources; iii) resource management; iv) flexible location and time. Additionally, a significant cost can be saved by law enforcement by centralising the processing of digital forensic evidence.

Data reduction techniques can aid in decreasing the volume of data to be analysed. Data deduplication is a data reduction technique used to optimise data storage and is particularly efficient when common data is encountered. In addition, the benefits of employing the proposed deduplicated digital evidence processing system include:

- **On-the-Fly Incriminating File Detection** - Known illegal artefacts can be detected *during* the acquisition step, rather than after complete acquisition;
- **A Model for Non-expert Acquisition and Analysis** - This system can preliminarily process digital evidence automatically;
- **The Bigger, The Better** - The more acquisitions performed using the proposed system, the higher the expected duplication rate encountered and the faster future acquisitions will become.
- **Intelligent Digital Evidence Analysis** - Stored expert analysis and previous data analysis can be used to train AI activity/event patterns to detect suspicious file artefacts automatically.

Chapter 6: **File Artefact Analysis and Relevancy Prioritisation**

6.1 Overview

Traditional, hash-based white-listing or blacklisting methods are usually the go-to automatic solution for finding known file artefacts during a digital forensic investigation. If nothing pertinent has been detected, the investigators will have to manually perform a keyword search or filter to examine the artefacts. Using hash matching to detect illegal files can only detect the precisely same file artefacts or artefacts with a minor change (i.e., by using approximate matching). Usually, most of the file artefacts on seized devices are not pertinent to the investigation. Manually retrieving suspicious files relevant to the investigation is akin to finding a needle in a haystack.

A methodology for the automatic determination of suspicious file artefacts (i.e., file artefacts that are pertinent to the investigation) is proposed to reduce the manual analysis effort required. The associated metadata and digital events occurred are employed to extracting features of each file artefact. Combining this with a centralised, deduplicated digital evidence processing framework, illegal file artefacts encountered in previous cases are labelled as such in the database. These files on the blacklist can be used to train classifiers for detecting previous unencountered suspicious file artefacts in new cases.

This methodology is designed to work in a human-in-the-loop fashion. In other words, it predicts/recommends that an artefact is likely to be suspicious rather than giving the final analysis result.

6.1.1 Data Processing and Experimentation

Metadata and timeline events extracted from the test disk image prepares data for file artefacts relevancy analysis. In Section 6.2, the details of the metadata and disk image timeline are introduced. In addition, the feature extraction technique used from each file artefact's timeline is also presented.

The approach is designed to prioritise file artefacts based on relevancy; experiments are con-

ducted for both classification and prioritisation. Experimentation presented in this Chapter includes: using a metadata-based model for relevancy classification, using image data as input for training a classification model and using features from timeline events for file relevancy prioritisation.

File Artefact Relevancy Classification

A supervised machine learning approach is employed, which leverages the recorded results of previously processed cases. The process of feature extraction, dataset generation, and training and evaluation are presented. In addition, a toolkit for data extraction from disk images is outlined, which enables this method to be integrated with the conventional investigation process and work in an automated fashion.

Image File Artefact Classification using a Pre-trained Model

Known file artefacts by the system can also be used for file data-based model training. The trained model can be used to integrate with metadata and timeline event-based models. A child abuse material case investigation data is emulated for the experimentation. For the purpose of this experiment, any picture with a child's face is classified as illegal; and all the other images are classified as benign.

File Artefact Relevancy Prioritisation

To demonstrate the viability of relevancy prioritisation, three sample scenarios are used. For each case, a set of features are used considering the different investigation focuses. The features applied to the model are determined by the detected pertinent files and what specific similarities/characteristics are looking for.

6.2 Metadata and Timeline from Disk Image

In the real-world investigation, the ground truth for training comes from expert human classifications which any models are built upon. For experimentation, the analysis is on generated disk images with emulated user files/actions. As shown in Figure 6.1, the process of experimental dataset generation includes:

1. Emulating wear-and-tear of the device on a virtual environment and planting the "illegal" file artefacts on the test virtual machine. The "illegal files" resulting from the emulated pertinent/illegal actions are hashed and preserved into an 'answer' file. This answer file can be subsequently used to validate the accuracy of any subsequent classification; either from a student in an educational scenario or by a digital evidence analysis tool or an automated evidence classification approach.

2. Using the developed tool to extract file system metadata and a "super timeline" from the disk image. This timeline contains the ground truth of the actions performed on the disk image, which can be subsequently used to validate whatever classification tasks that follow.

3. Merging the information about file artefacts from two sources (i.e., metadata and the extracted timeline).

4. Labelling the file artefacts on the dataset based on the hash of file artefacts. Find the "illegal" files by the preserved hash value in the answer file. Any remaining artefacts (i.e., files not identified by the answer file as being illegal) are labelled as "benign".

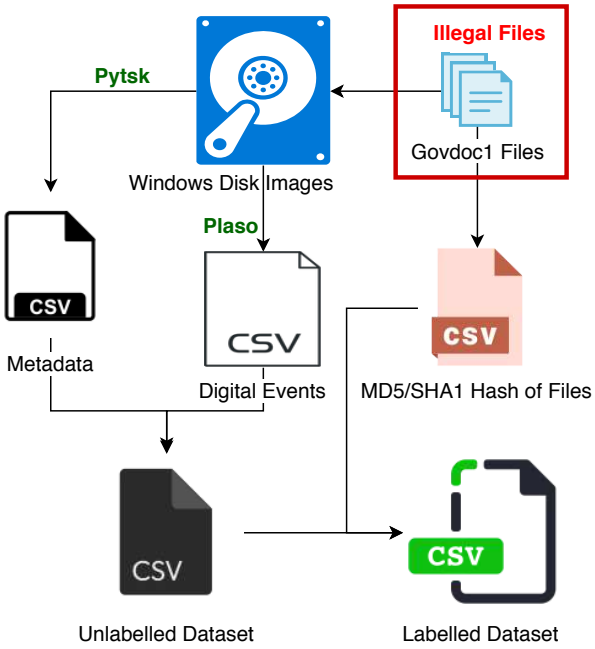


Figure 6.1: Disk Images Generation and Data Processing for Analysis[a]

[a]Du, X. and Scanlon, M. Methodology for the Automated Metadata-Based Classification of Incriminating Digital Forensic Artefacts, The 12th International Workshop on Digital Forensics (WSDF), held at the 14th International Conference on Availability, Reliability and Security (ARES), Canterbury, UK, August 2019.

6.2.1 Timeline Generation and Analysis

Timelines contain more information than merely timestamps. The "super timeline" generated by `log2timeline` consists of what happened, when it happened, on which artefact, and where each digital event was recorded on the system. A complete list of the fields of generated timeline are categorised and listed as follows:

- Fields describing the event:

  - **date** - Date that the event occurred;
  - **time** - Time that the event occurred;
  - **MACB** - Modification, access, creation and birth times;
  - **desc** - A description of the timestamp object;
  - **short** - A shorter version description of the timestamp;
  - **filename** - The file object on which the event occurred;
  - **sourcetype** - Description of the source type, e.g., "Opera Browser History";
  - **source** - A shorted form of the source, e.g., "WEBHIST";
  - **type** -Timestamp type, e.g., "Last Time Executed".

- Fields describing the source:
  - **inode** - The inode or MFT number of the parsed artefacts;
  - **user** - The user owns the parsed artefacts;
  - **host** - The host that the data came from.

- Fields describing the tool used:
  - **version** - The version of the tool;
  - **timezone** - Timezone where applying the tool generating the timeline;
  - **format** - The parsing module.

- Fields describing other information:
  - **notes** - An operational field;
  - **extra** - A reference to a hash that stores all additional fields that might be used.

6.2.2 An Example Disk Image Timeline

This Section presents an overview of timeline generation. The full disk timeline reflects the usage of the seized machine, the number of digital events discovered in total, the number of files, the count of each digital events type, etc. *psteal* is a tool in *Plaso* for comprehensive disk image timeline generation and the command used is:

```
psteal.py --source disk_image_name.dd -t l2tcsv -w timeline_name.csv --partitions
all
```

A test disk image is *Windows 7.raw*, with "illegal" actions emulated in VirtualBox. From the generated timeline, basic information about the created disk image can be retrieved; in this scenario:

- Number of Events: 3,120,364

- Number of Files: 307,971

The timeline consists of all level of digital events. *pandas* is used to further analyse the timeline. To acquire counts of unique values the method `count_values()` is used. On a full disk image level, the source of digital events reveals information about the usage of the device:

Event Type	**Count**	**Event Type**	**Count**
Last Connection Time	3	Last Login Time	2
Scheduled to start	3	Last Shutdown Time	1
Last Password Reset	3	Last Used Time	1
Installation Time	2	-	-

Table 6.1: Device Related Event

Digital events are related to file system metadata information. Table 6.2 shows the information extracted in the sample scenario.

Event Type	**Count**
Content Modification Time	962,293
Metadata Modification Time	551,502
Creation Time; Last Access Time; Metadata Modification Time	343,906
Content Modification Time; Creation Time; Last Access Time; Metadata Modification Time	302,467
Last Access Time	283,980
Creation Time	235,871
Content Modification Time; Creation Time	212,687
Creation Time; Last Access Time	45,898
Content Modification Time; Last Access Time; Metadata Modification Time	35,818
Last Access Time; Metadata Modification Time	32,881

Table 6.2: File Artefacts Event - Common

There are some types of events that can only occur to a specific type of file as shown in Figure 6.3. For example, *Previous Last Time Executed* could occur to an executable file, but not to a document or image file. Another example is a *File Downloaded* event – this can only occur if a file originates from a request to another machine through a network connection. These special events can be used as features pertaining to associated file artefacts, i.e., *true* or *false* as the feature value.

Event Type	Count
Last Visited Time	5,534
Previous Last Time Executed	1,107
File Last Modification Time	585
Start Time	410
Last Time Executed	401
File Downloaded	118
Document Creation Time	86
First Connection Time	85
Document Last Save Time	82
Content Deletion Time	58

Table 6.3: File Artefacts Event - Specific

6.2.3 File Timeline Generation

A file timeline is generated by searching for a filename as the keyword in the entire disk image timeline. The extracted digital events are related to the file and gathered together to form a file timeline.

As the code shown below demonstrates, the extracted digital events associated with a specific file are preserved into a CSV file. *NumPy* (`np`) is the library used to search keyword occurrence in each column of the disk image.

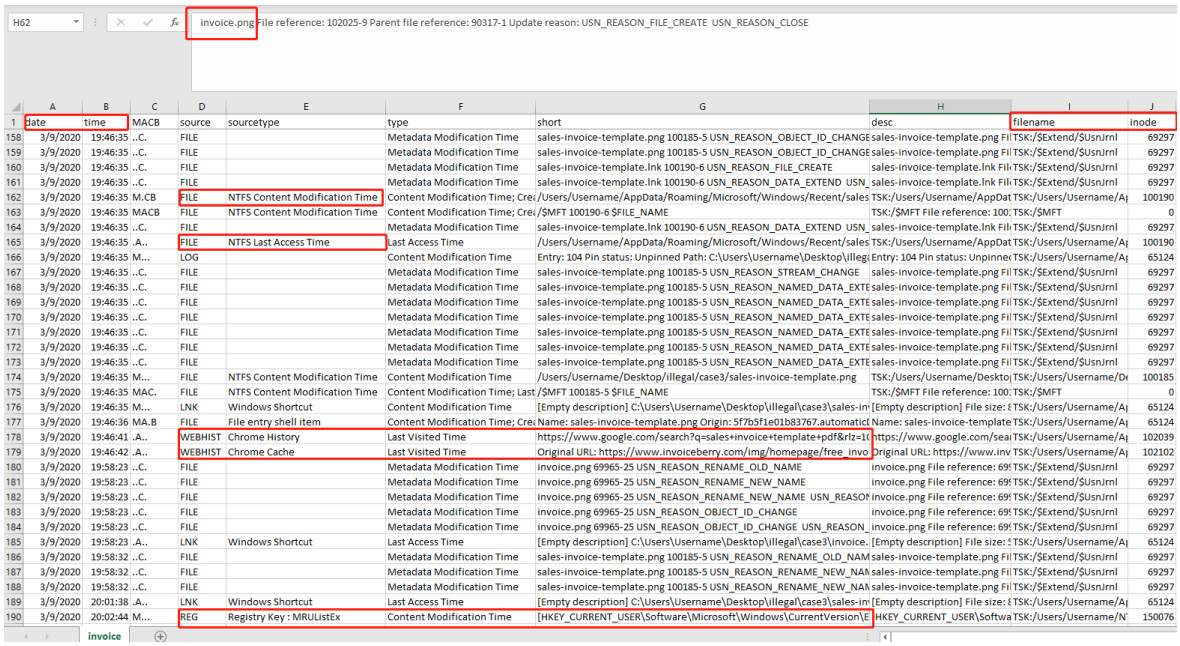
The disk image timeline generation is a necessary step in a conventional approach as well. The file timeline generation time is the extra time taken for the proposed approach; it takes approximately 1 minute for each file. This process can be improved through parallel processing in the future.

```
# given a keyword (filename for example),
# the timeline as input, output is a csv contains records include to the keyword
def keyword_search_and_extraction(csv_file, csv_input, search_keyword):
    head, tail = ntpath.split(csv_input)
    directory = head + os.sep + csv_file + os.sep
    if not os.path.exists(directory):
        os.makedirs(directory)
    csv_output = head + os.sep + csv_file + os.sep + search_keyword  + ".csv"
    data = pd.read_csv(csv_input, header=0, encoding='utf_8', low_memory=False)
    for col in data:
        if data[col].dtype == 'int64':
            del data[col]
    mask = np.column_stack([data[col].astype(str).str.contains(search_keyword, na=False)
    for col in data])
    data_filtered = data.loc[mask.any(axis=1)]
    data_filtered.to_csv(csv_output, encoding='utf-8')
    print('output to: ' + csv_output)
    return csv_output
```

Figure 6.2: Method for File Timeline Generation

6.2.4 An Example of File Timeline

Figure 6.3 shows an example file timeline. The fields in the file timeline include date, time, MACB, source, sourcetype, type, short, desc, file name, and inode. The emulated investigation does not involve multiple devices/suspects, the fields of user, host, etc. are not included. These fields can be useful in the investigation of real cases.



	A	B	C	D	E	F	G	H	I	J
1	date	time	MACB	source	sourcetype	type	short	desc	filename	inode
158	3/9/2020	19:46:35	..C.	FILE		Metadata Modification Time	sales-invoice-template.png 100185-5 USN_REASON_OBJECT_ID_CHANGE	sales-invoice-template.png Fi	TSK:/$Extend/$UsnJrnl	69297
159	3/9/2020	19:46:35	..C.	FILE		Metadata Modification Time	sales-invoice-template.png 100185-5 USN_REASON_OBJECT_ID_CHANGE	sales-invoice-template.png Fi	TSK:/$Extend/$UsnJrnl	69297
160	3/9/2020	19:46:35	..C.	FILE		Metadata Modification Time	sales-invoice-template.lnk 100190-6 USN_REASON_FILE_CREATE	sales-invoice-template.lnk Fil	TSK:/$Extend/$UsnJrnl	69297
161	3/9/2020	19:46:35	..C.	FILE		Metadata Modification Time	sales-invoice-template.lnk 100190-6 USN_REASON_DATA_EXTEND USN_	sales-invoice-template.lnk Fil	TSK:/$Extend/$UsnJrnl	69297
162	3/9/2020	19:46:35	M.CB	FILE	NTFS Content Modification Time	Content Modification Time; Crea	/Users/Username/AppData/Roaming/Microsoft/Windows/Recent/sales	TSK:/Users/Username/AppDat	TSK:/Users/Username/Ap	100190
163	3/9/2020	19:46:35	MACB	FILE	NTFS Content Modification Time	Content Modification Time; Crea	/$MFT 100190-6 $FILE_NAME	TSK:/$MFT File reference: 100	TSK:/$MFT	0
164	3/9/2020	19:46:35	..C.	FILE		Metadata Modification Time	sales-invoice-template.lnk 100190-6 USN_REASON_DATA_EXTEND USN_	sales-invoice-template.lnk Fil	TSK:/$Extend/$UsnJrnl	69297
165	3/9/2020	19:46:35	.A..	FILE	NTFS Last Access Time	Last Access Time	/Users/Username/AppData/Roaming/Microsoft/Windows/Recent/sales	TSK:/Users/Username/AppDat	TSK:/Users/Username/Ap	100190
166	3/9/2020	19:46:35	M...	LOG		Content Modification Time	Entry: 104 Pin status: Unpinned Path: C:\Users\Username\Desktop\illeg	Entry: 104 Pin status: Unpinne	TSK:/Users/Username/Ap	65124
167	3/9/2020	19:46:35	..C.	FILE		Metadata Modification Time	sales-invoice-template.png 100185-5 USN_REASON_STREAM_CHANGE	sales-invoice-template.png Fi	TSK:/$Extend/$UsnJrnl	69297
168	3/9/2020	19:46:35	..C.	FILE		Metadata Modification Time	sales-invoice-template.png 100185-5 USN_REASON_NAMED_DATA_EXTE	sales-invoice-template.png Fi	TSK:/$Extend/$UsnJrnl	69297
169	3/9/2020	19:46:35	..C.	FILE		Metadata Modification Time	sales-invoice-template.png 100185-5 USN_REASON_NAMED_DATA_EXTE	sales-invoice-template.png Fi	TSK:/$Extend/$UsnJrnl	69297
170	3/9/2020	19:46:35	..C.	FILE		Metadata Modification Time	sales-invoice-template.png 100185-5 USN_REASON_NAMED_DATA_EXTE	sales-invoice-template.png Fi	TSK:/$Extend/$UsnJrnl	69297
171	3/9/2020	19:46:35	..C.	FILE		Metadata Modification Time	sales-invoice-template.png 100185-5 USN_REASON_NAMED_DATA_EXTE	sales-invoice-template.png Fi	TSK:/$Extend/$UsnJrnl	69297
172	3/9/2020	19:46:35	..C.	FILE		Metadata Modification Time	sales-invoice-template.png 100185-5 USN_REASON_NAMED_DATA_EXTE	sales-invoice-template.png Fi	TSK:/$Extend/$UsnJrnl	69297
173	3/9/2020	19:46:35	..C.	FILE		Metadata Modification Time	sales-invoice-template.png 100185-5 USN_REASON_NAMED_DATA_EXTE	sales-invoice-template.png Fi	TSK:/$Extend/$UsnJrnl	69297
174	3/9/2020	19:46:35	M...	FILE	NTFS Content Modification Time	Content Modification Time	/Users/Username/Desktop/illegal/case3/sales-invoice-template.png	TSK:/Users/Username/Deskto	TSK:/Users/Username/De	100185
175	3/9/2020	19:46:35	MAC.	FILE	NTFS Content Modification Time	Content Modification Time; Last	/$MFT 100185-5 $FILE_NAME	TSK:/$MFT File reference: 100	TSK:/$MFT	0
176	3/9/2020	19:46:35	M...	LNK	Windows Shortcut	Content Modification Time	[Empty description] C:\Users\Username\Desktop\illegal\case3\sales-in	[Empty description] File size: 8	TSK:/Users/Username/Ap	65124
177	3/9/2020	19:46:36	MA.B	FILE	File entry shell item	Content Modification Time; Cre	Name: sales-invoice-template.png Origin: 5f7b5f1e01b83767.automaticI	Name: sales-invoice-template	TSK:/Users/Username/Ap	65124
178	3/9/2020	19:46:41	.A..	WEBHIST	Chrome History	Last Visited Time	https://www.google.com/search?q=sales+invoice+template+pdf&rlz=1C	https://www.google.com/sea	TSK:/Users/Username/Ap	102039
179	3/9/2020	19:46:42	.A..	WEBHIST	Chrome Cache	Last Visited Time	Original URL: https://www.invoiceberry.com/img/homepage/free_invo	Original URL: https://www.inv	TSK:/Users/Username/Ap	102102
180	3/9/2020	19:58:23	..C.	FILE		Metadata Modification Time	invoice.png 69965-25 USN_REASON_RENAME_OLD_NAME	invoice.png File reference: 69	TSK:/$Extend/$UsnJrnl	69297
181	3/9/2020	19:58:23	..C.	FILE		Metadata Modification Time	invoice.png 69965-25 USN_REASON_RENAME_NEW_NAME	invoice.png File reference: 69	TSK:/$Extend/$UsnJrnl	69297
182	3/9/2020	19:58:23	..C.	FILE		Metadata Modification Time	invoice.png 69965-25 USN_REASON_RENAME_NEW_NAME USN_REASON	invoice.png File reference: 69	TSK:/$Extend/$UsnJrnl	69297
183	3/9/2020	19:58:23	..C.	FILE		Metadata Modification Time	invoice.png 69965-25 USN_REASON_OBJECT_ID_CHANGE	invoice.png File reference: 69	TSK:/$Extend/$UsnJrnl	69297
184	3/9/2020	19:58:23	..C.	FILE		Metadata Modification Time	invoice.png 69965-25 USN_REASON_OBJECT_ID_CHANGE USN_REASON_	invoice.png File reference: 69	TSK:/$Extend/$UsnJrnl	69297
185	3/9/2020	19:58:23	.A..	LNK	Windows Shortcut	Last Access Time	[Empty description] C:\Users\Username\Desktop\illegal\case3\invoice.	[Empty description] File size: 8	TSK:/Users/Username/Ap	65124
186	3/9/2020	19:58:32	..C.	FILE		Metadata Modification Time	sales-invoice-template.png 100185-5 USN_REASON_RENAME_OLD_NAM	sales-invoice-template.png Fi	TSK:/$Extend/$UsnJrnl	69297
187	3/9/2020	19:58:32	..C.	FILE		Metadata Modification Time	sales-invoice-template.png 100185-5 USN_REASON_RENAME_NEW_NA	sales-invoice-template.png Fi	TSK:/$Extend/$UsnJrnl	69297
188	3/9/2020	19:58:32	..C.	FILE		Metadata Modification Time	sales-invoice-template.png 100185-5 USN_REASON_RENAME_NEW_NA	sales-invoice-template.png Fi	TSK:/$Extend/$UsnJrnl	69297
189	3/9/2020	20:01:38	.A..	LNK	Windows Shortcut	Last Access Time	[Empty description] C:\Users\Username\Desktop\illegal\case3\sales-in	[Empty description] File size: 8	TSK:/Users/Username/Ap	65124
190	3/9/2020	20:02:44	M...	REG	Registry Key : MRUListEx	Content Modification Time	[HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\E	HKEY_CURRENT_USER\Softwa	TSK:/Users/Username/N	150076

Figure 6.3: An Example of File Artefact Timeline

File timelines contain all the event information associated with one specific file. The information can be used as features for input to machine learning models. For the example file timeline shown in Figure 6.3, the following can be seen:

- the number of digital events (the higher the number of associated digital events for a file indicates it is more frequently being used by the user);
- the file's creation/last access time (the first/last time the file was used);
- the file timeline contains digital events from Chrome (Chrome could be the provenance of the file);
- One associated event is from the registry;

6.2.5 Feature Extraction from File Timeline

Features can be extracted from the file timeline by a variety of approaches. Table 6.4 shows example features can be used as input for machine learning model training.

Features: Type and Value of Timeline Events

The value of fields MACB, source, sourcetype, type, etc. are categories. A file timeline could include events with a different number of values. Features can be chosen from the type and value which are characteristic of a file and the investigator is looking for files with similar events. For example, this file is 'png' extension and contains events from 'chrome', 'history', 'cache'. These words are used as a bag of words (BoWs) features.

Features: Words Count/Frequency

The "short" and "desc" are text, and can be represented by a BoWs (describing the occurrence of words within it). Either count or frequency of words can be used. Term Frequency-Inverse Document Frequency (TF-IDF) takes another approach, where it is believed that high-frequency occurrences may not able to provide much information gain. In other words, rarer words can contribute more weight to the model.

Feature Name	**Feature Type**	**Feature Name**	**Feature Type**
Event Count	Numeric	**Event Type (Least)**	Categories
Time (Most)	Categories (Hour)	**Event Type (Most)**	Categories
Time (Least)	Categories (Hour)	**Date (Least)**	Categories
Event Source	Categories	**Date (Most)**	Categories
WEBHIST	Numeric	**Word Frequency (Most)**	Categories
FILE	Numeric	**Word Frequency (Least)**	Categories
LIN	Numeric	**Word Occurrence (email)**	Numeric
REG	Numeric	**Word Occurrence**	Numeric

Table 6.4: Example Features from a File Timeline

6.3 Metadata-based File Artefact Classification

This Section presents an approach that is used to automatically identify suspicious file artefacts through the learning of a machine learning model. Several machine learning classification algorithms including Logistic Regression, k-NN, SVM, Decision Tree, Gaussian Naïve Bayes are evaluated in the experimentation.

6.3.1 Example Scenario

In child abuse material possession/distribution investigation cases, pertinent digital evidence often consists of multimedia content with similar file size, under similar directories, similar creation times, last access times, etc. The proposed methodology aims to detect suspicious files in such investigative scenarios. Hence, file artefacts with a similar number of associated events, same file type, or under the same directory to known illegal file artefacts should likely be predicted as suspicious/relevant.

Based on the purpose of the prediction task, the complete matrix of the features are:

- **Depth of Dir** - Integer representing the depth of the file directory (i.e., the number of parent directories);
- **File Extension** - Categorical data type;
- **Length of Name** - An integer representing the filename's length.
- **Creation Time (y)** - How many years old is the file;
- **Creation Time (m)** - How many months old is the file;
- **Creation Time (d)** - How many days old is the file;
- **Creation Time (h)** - How many hours old is the file;
- **Size** - The file size in KB;
- **Count** - The number of associated file events;
- **Class** - If the file benign or illegal. (the value is 0 for benign files, 1 for illegal files).

6.3.2 Datasets

Table 6.5 shows the generated datasets used in this experiment; mainly differing on the percentage of illegal artefacts. The dataset is split into training and testing data.

Dataset	No. of Artefacts	Benign Artefacts	Illegal Artefacts
Dataset1	42,326	41,339	987
Dataset2	55,296	49,328	5,968

Table 6.5: Datasets Used in the Experiment

6.3.3 Evaluation Matrix

Due to the severe imbalance of the dataset, accuracy is not used to evaluate the performance. Because a model can predict the value of the majority class for all predictions and achieve a high classification accuracy, this model is not useful in the problem domain.

The performance metrics used are precision, recall and F1-score:

- **Precision** is the fraction of relevant instances among the retrieved instances:

$$precision = \frac{TP}{TP + FP} \tag{6.1}$$

- **Recall** is the fraction of relevant instances that have been retrieved:

$$recall = \frac{TP}{TP + FN} \tag{6.2}$$

- **F1-Score** is an overall measure of a model's accuracy that combines precision and recall:

$$F1\text{-}Score = 2 \times recall \times precision \tag{6.3}$$

6.3.4 Precision-Recall Curves with Average Precision Scores

Precision-Recall curves summarise the trade-off between the true positive rate and the positive predictive value for a predictive model using different probability thresholds. For getting an overall understanding of the models' performance, Precision-Recall curve (RP curve) is visualised and average precision score (AP score) is calculated. AP score is calculated using the library sklearn.metrics.average_precision_score [1], it summarises a precision-recall curve as the weighted mean of precisions achieved at each threshold, with the increase in recall from the previous threshold used as the weight:

$$AP\ Score = \sum_{n=1} (R_n - R_{n-1}) P_n \tag{6.4}$$

where P_n and R_n are the precision and recall at the nth threshold.

Comparing the average precision score, the performance of the tested models from best to worst is Decision Tree, Gaussian Naïve Bayes, SVM, k-NN, and Logistic Regression respectively.

[1] https://scikit-learn.org/stable/modules/generated/sklearn.metrics.average_precision_score.html

- **Decision Tree** - Figure 6.4 shows the RP curve of the employed decision tree algorithm on `dataset1`. Observing from the AP score and precision, recall and F1 scores, the Decision Tree classifier performs best for both datasets compared against other models. In addition, the performance is stable on different datasets.

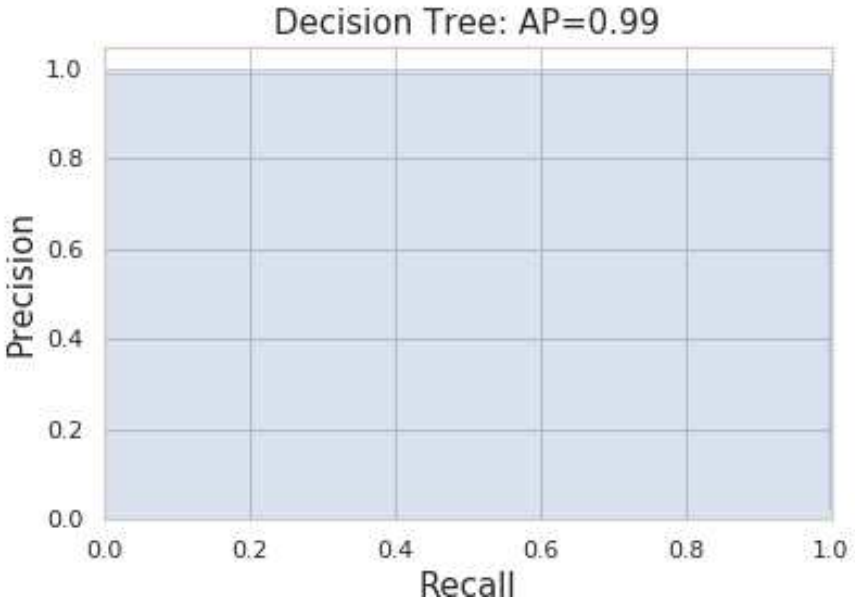


Figure 6.4: Precision-Recall Curves - Decision Tree

- **Gaussian Naïve Bayes** - Figure 6.5 shows the RP curve of the employed Gaussian Naïve Bayes algorithm on `dataset1`. The performance of Gaussian Naïve Bayes on `dataset1` is worse than `dataset2`. Even though the AP score is good, the scores on class 1 are much worse. For this imbalanced dataset, the score for the class with fewer samples is very low.

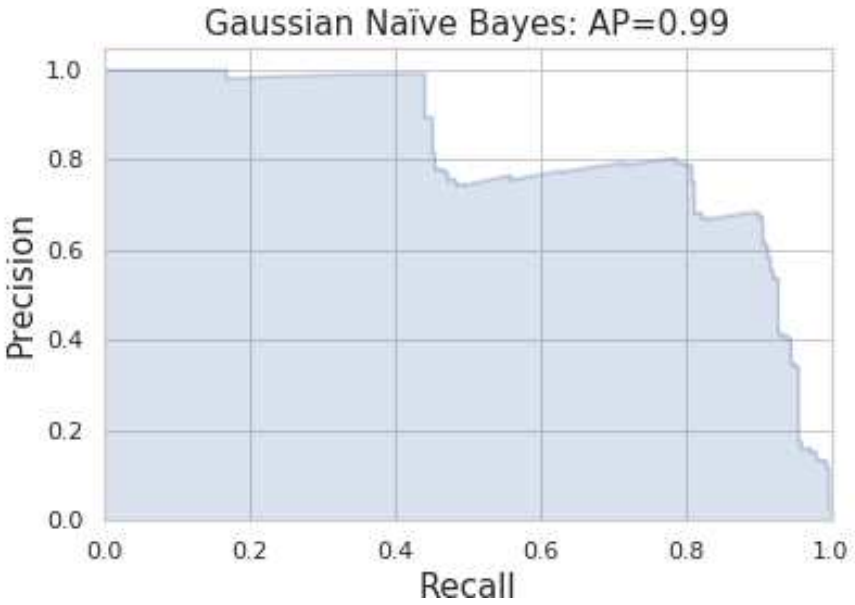


Figure 6.5: Precision-Recall Curves - Gaussian Naïve Bayes

- **k-NN** - Figure 6.6 shows the RP curve of the employed Gaussian Naïve Bayes algorithm on `dataset1`. One important parameter of the k-NN algorithm is the selection of `k`. In this experiment, the model achieved its best accuracy when `k` was set to be 5.

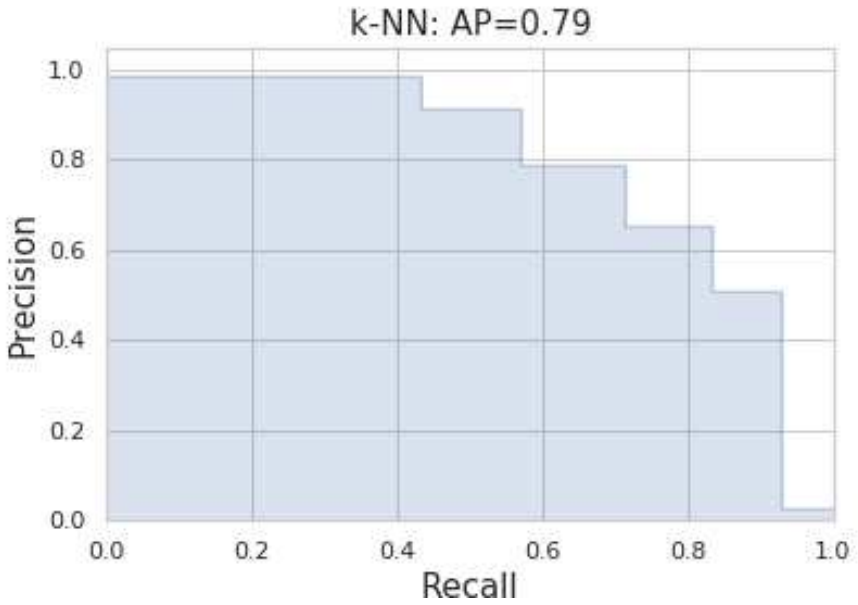


Figure 6.6: Precision-Recall Curves - k-NN

- **SVM** - Figure 6.7 shows the RP curve of the employed SVM algorithm on `dataset1`. Through experimentation, the model realises a poor performance when the given data is not normalised. Hence, the dataset is normalised before it is fed into the model and the parameters are left default for the experiment.

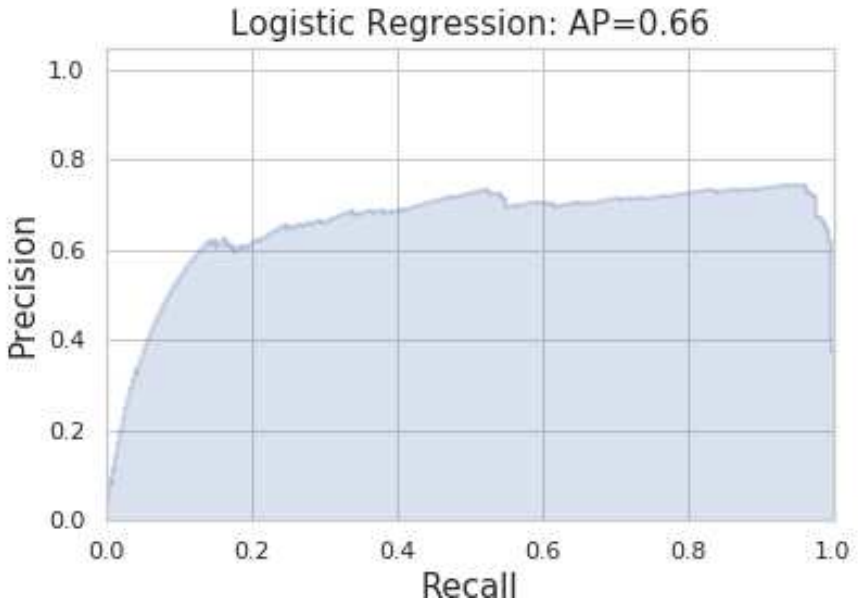


Figure 6.7: Precision-Recall Curves - SVM

- **Logistic Regression** requires quite large sample sizes, which could be the reason it

scores lower comparing with the other models. Figure 6.8 shows the RP curve of the employed Logistic Regression algorithm on `dataset1`.

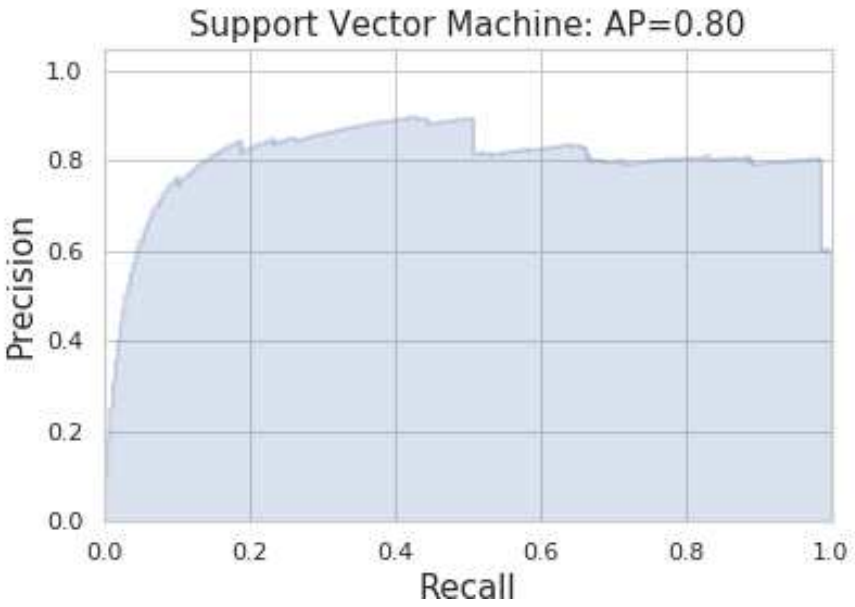


Figure 6.8: Precision-Recall Curves - Logistic Regression

6.3.5 Result Comparison on Two Datasets

As the practical usage of the trained model is for digital forensic investigation, more concern should be put on the classifying of illegal files (class 1). Precisely, the precision, recall and F1 score on class 1, instead of the average value on class 0 and class 1, should be used to represents the performance of the models. The scores shown in Table 6.6 are focused on class 1. This table shows an evaluation matrix from two datasets.

As shown in Table 6.6, the performance with `dataset2` is significantly better than `dataset1`. This is due to the number of "illegal" file samples in `dataset2` being more than `dataset1`. The imbalanced class problem is apparent in these datasets. The datasets were created in this manner due to the assumption that only a subset of the "illegal" files is classified as known illegal/known benign from the centralised database.

The performance of these models indicates that the proposed methodology is valid and justifies further exploration. Among the aforementioned algorithms, Decision Tree achieved the best performance. The percentage distribution of the file artefacts among the different classes can be various in real-world investigations. Different distribution ratios of illegal and benign files in the dataset should be tested for giving further conclusion which model is more suitable for specific scenarios.

Algorithms	datsaset1			datsaset2			Short Summary
	precision	recall	f1-score	precision	recall	f1-score	
Decision Trees	0.99	1.00	0.99	1.00	1.00	1.00	Best models in this experiment
Gaussian Naïve Bayes	0.16	0.97	0.27	0.99	1.00	0.99	Performance influenced by the dataset very much
k-NN	0.79	0.71	0.75	1.00	1.00	1.00	Better performance on `dataset2`
SVMs	0.82	0.52	0.64	1.00	1.00	1.00	Better performance on `dataset2`
Logistic Regression	0.71	0.67	0.69	0.99	1.00	0.99	Better performance on `dataset2`

Table 6.6: Evaluation Matrix of Different Classification Algorithms and Datasets

6.4 Image File Artefact Classification

In this experiment, a popular pre-trained neural network-based model - `ResNet-50` is applied for image classification in the child abuse case investigative scenario. This section reports the process of dataset construction and experimental results.

6.4.1 Experimental Data

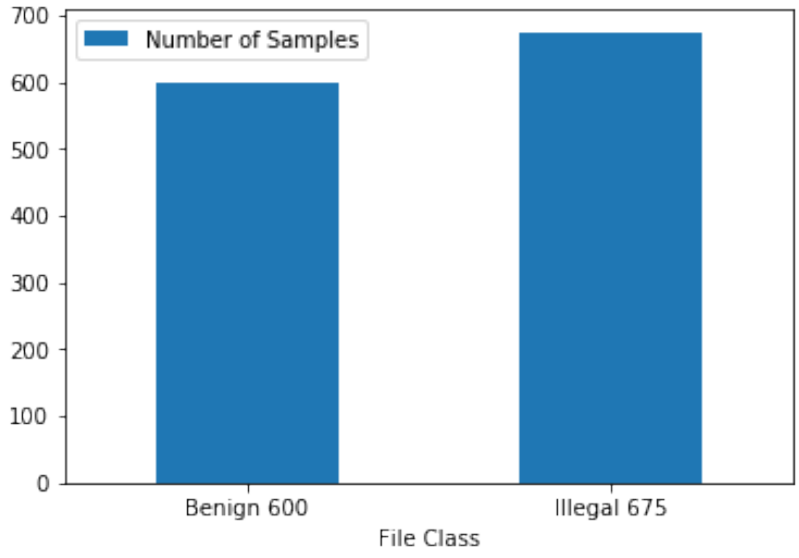


Figure 6.9: Training Data for Image File Artefact Classification

The problem is defined as a binary classification task and thus we construct a data set of images annotated as either "benign" or "illegal". The images labelled as "illegal" refer to that those contain children' faces, while benign ones are pictures gathered from Google Images and Wikipedia by keywords: selfie, screenshot, receipt, etc. In the end, the dataset comprises a training set with 600 benign and 675 illegal samples, as outlined in Figure 6.9, and a test set with 500 benign and 500 illegal samples.

6.4.2 Experimental Setup and Results

In the experiment, the training of `ResNet-50` is set up with *Pytorch*[2]. The pre-trained `ResNet-50` is fine-tuned with 5 epochs using Adam[3] [158] as the optimiser. The training is performed with a constant learning rate of 0.003 and the parameters of the model are optimised based on the objective function of cross-entropy loss. The batch size during training is set to be 32. It is estimated that the whole training process takes 10 minutes to be finished.

Figure 6.10 shows the trend of loss on the training set; Figure 6.11 shows the decrease of loss on the test set during training. The results show the model is trained with reduced loss as the training steps increase; namely the ability to distinguish between the benign and illegal images increases.

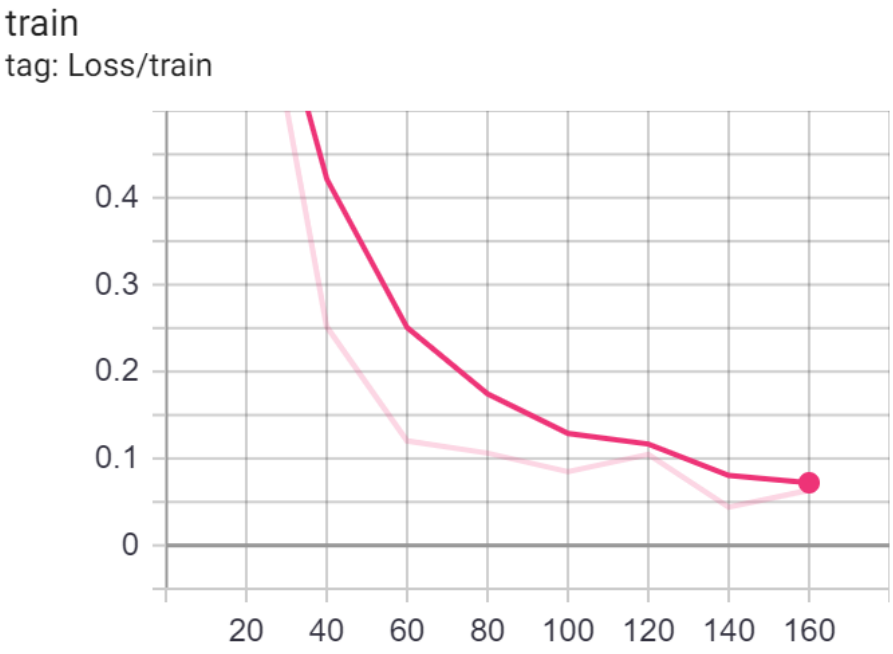


Figure 6.10: Training Model on `ResNet-50`: Loss/Train

[2]PyTorch is an open-source machine learning library based on the Torch library, used for applications such as computer vision and natural language processing, primarily developed by Facebook's AI Research lab.

[3]An algorithm for first-order gradient-based optimisation of stochastic objective functions.

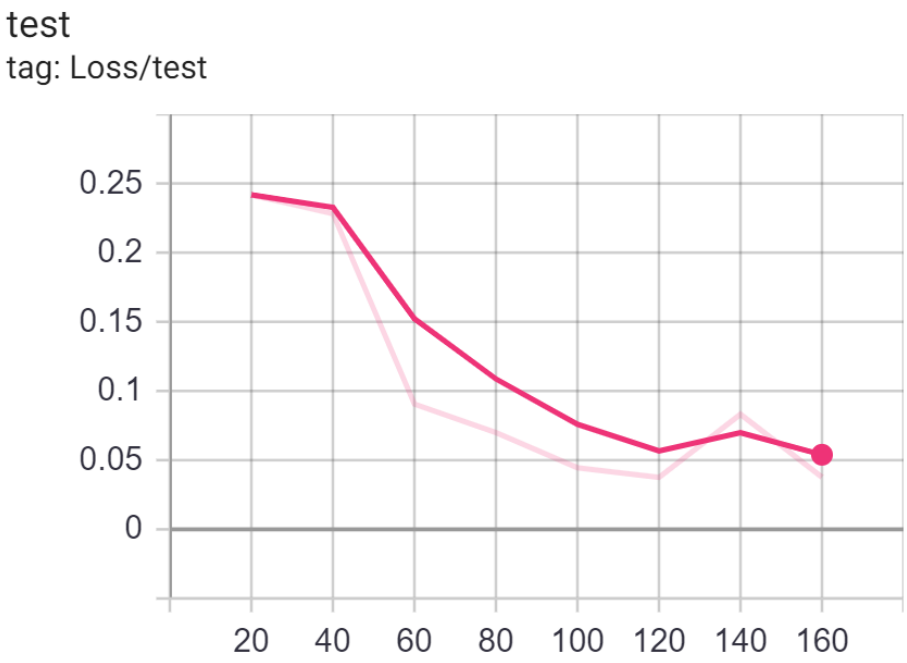


Figure 6.11: Training Model on `ResNet-50`: Loss/Test

Figure 6.12 presents the evaluation of the test set at different steps of training. It shows the best checkpoint of the model achieves a 98% accuracy.

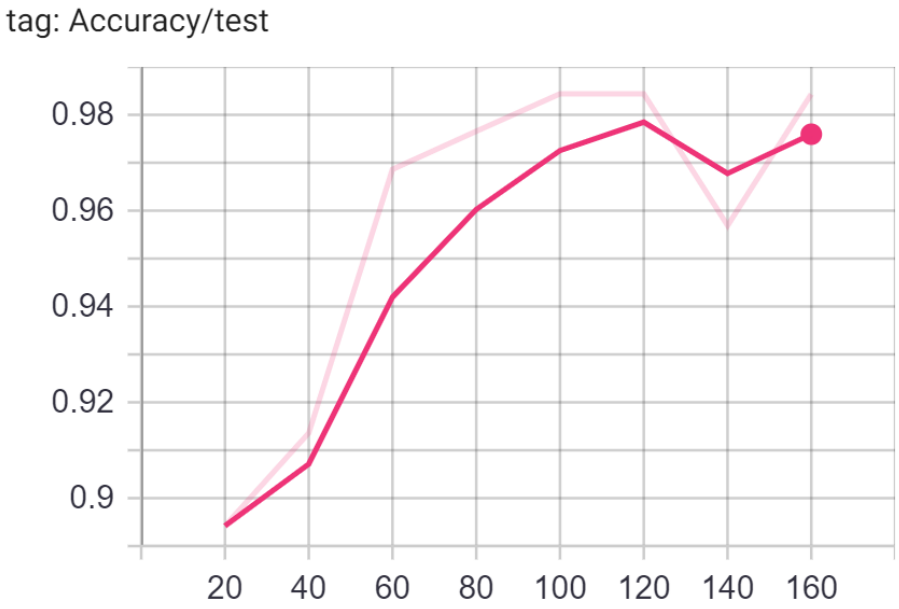


Figure 6.12: Training Model on `ResNet-50`: Accuracy/Test

The model trained in this approach can be integrated with the metadata and timeline event-based machine learning model. The output of the file data-based model can be used as a field as input for metadata and timeline event-based model training.

6.5 File Artefact Relevancy Prioritisation

6.5.1 Experimental Data

As this experimentation requires performance analysis, emulated data is used that has the following benefits: 1) files with illegal content are not needed, the proposed approach used the files' associated digital events to determine if it is suspicious or not; and 2) generated data has a clearer and more detailed ground truth.

Disk images were generated as virtual machines. Firstly, the emulated actions were conducted to generate the files for investigation. This experiment aims to test the recognition of similar files through digital events – therefore file metadata and content do not influence the experiment. Various files, with several file types, are randomly generated and downloaded onto the VM. Files with various user actions are emulated. General information for these files is listed in Table 6.7. These files in the VM are labelled as "benign", mixed with "illegal" file artefacts.

File Type	User Actions	Number
pdf	creation (download from web)	999
txt	creation (notepad)	100
png	creation (download from web)	100
py	creation, access, run by python	63

Table 6.7: "Benign" File Information

The "pertinent" actions included emulated user activities for each of the three sample case scenarios; downloading CSEM (downloading research paper on the topic, picture download and photos sent/received using online chat tools); the execution of a hacking python script for cracking user's password; and creating fake invoices for a financial fraud investigation. The actions defined as pertinent are those surrounding the activities with each scenario. The files related to these actions are labelled as "pertinent".

File Type	User Actions	Number
txt	creation, access, edit	6
py	creation, unzip, access, move, copy	6
jpg	creation, access	13
png	creation (download from web), access	4
gif	creation (download from web), access	1
pdf	creation (download from web)	1

Table 6.8: "Illegal" File Information

6.5.2 Example Scenarios

To demonstrate the viability of our approach, three sample scenarios are used (and will be referenced below):

Possession of CSEM Investigation

The suspect uses a computer to access a chat room online related to child sexual exploitation material (CSEM). Videos and pictures are downloaded to local disk from an installed browser. A computer belonging to a suspect was sized during a CSEM case investigation. Investigators use the known hash database filtering out the known illegal files; then a data reduction tool gets a set of user files that is most common to find pertinent files. These are chat log files, email files and picture files. With these picture files, some are detected as illegal from a known hash database. The investigator puts these file into an SVM model for training. In the end, other unknown files were put into the model, files are sorted by relevancy score for further analysis.

Hacking Case Investigation

A computer was seized during a hacking case investigation. The suspect uses an email account. Keyword searching for "username" and "password" identifies several files. These are text files with content related to the use of password cracking scripts and scripts for hacking wireless networks. Then investigators choose to put these files into a model to look for other relevant files.

Financial Fraud Investigation

The suspect creates a phishing site to con victims into supplying their email address and password and other personal information. The suspect uses their accounts to conduct fraud online. During an investigation of a financial fraud case, investigators are looking to find out potentially fraudulent financial instruments, invoices or other financial records. Searching the keyword "invoice(s)" in `pdf` *and* `doc` *files from the raw disk image results in the discovery of some relevant files. Then investigators use the analysis result to build a model to recognise similar files.*

6.5.3 Case Investigation and Relevancy Prioritisation

This Section presents the results of the experimentation and investigative process conducted on each of the emulated case scenarios.

For each case, a different set of features are used considering the different investigation focuses.

The features applied to the model are determined by the detected pertinent files and what specific similarities/characteristics are sought. The features extracted for building the model for each case are listed below:

1. For the CSEM case scenario, the investigation focuses on images, videos, etc. The detected illegal files found have associated digital events from browsing activity. In addition, several file copying and moving actions for a number of the files were found in file timelines. For training the model to discover more files with a similar usage behaviour, the features used are: 'chrome', 'child', 'png', 'jpg', and 'MFT'.

2. In the hacking case scenario, python scripts for user password cracking and a couple of related text files were found. The python project was unzipped from a compressed file. Based on these details, the features used are: 'hack', 'python', 'py', 'txt', 'zip', 'unzip'.

3. Investigation of the financial fraud scenario found emails that were sent with fake invoices (files in `pdf` format). The user had accessed the files close to time last use of the seized machine. The model building for further exploration uses features: 'pdf', 'invoice', 'email', 'fraud', 'Last Access Time', 'Creation Time'.

The known files are used to train a binary-class SVM model. Since the linear SVM algorithm is applied, the weights can be used as coefficients in a linear discriminant function to calculate the relevancy score.

```
def cal_score(feature_weight, X, y):
    from numpy import dot
    score_list = np.matmul(feature_weight, np.transpose(X)).squeeze()
    score_tup = list(zip(np.transpose(score_list), np.transpose(y)))
    score_tup_s = sorted(score_tup, key=lambda tup: tup[1])
    return score_tup_s
```

Figure 6.13: Relevancy Score Generation in the Experiment

The cases were tested on a dataset with 5.6% of the files labelled as pertinent. Table 6.9 shows that for each model, the recall metric obtained 75% to 89%, when only looking at the top 10% of the resultant ranked result.

However, illegal files that are not similar to the detected known files can not be detected by this approach, which leads to the false-negative error. Not similar in this context refers to both a dissimilarity of the file content itself and the behavioural usage pattern of the file. In the other words, an illegal file could get a low relevancy score and rank at the bottom of the list. Therefore, this approach can not be assured to find out all illegal files, but can be used to guide the investigator's focused towards sources of pertinent information at the earliest stage possible in the investigation.

No. Reviewed	Case 1	Case 2	Case 3
10%	0.75	0.82	0.89
20%	0.75	0.82	0.89
30%	0.79	0.82	0.89
50%	0.79	0.82	0.89
100%	1.0	1.0	1.0

Table 6.9: Recall of Each Model

6.6 Summary

Leveraging previously processed digital forensic cases and their component artefact relevancy classifications can facilitate an opportunity for training automated AI-based evidence processing systems. These can significantly aid investigators in the discovery and prioritisation of evidence. This work presents one approach for file artefact relevancy determination building on the growing trend towards a centralised, DFaaS paradigm. This approach enables the use of previously encountered pertinent files to classify newly discovered files in an investigation. Trained models can aid in the detection of these files during the acquisition stage, i.e., during their upload to a DFaaS system. The technique generates a relevancy score for file similarity using each artefact's filesystem metadata and associated timeline events.

This approach prioritises file artefacts that are similar to previously analysed pertinent files. The automated process is assisted by developed feature extraction tools and machine learning models. The results show the advantages of the approach and indicate the promise of an expedited investigation. As a result, this approach would work best at an early stage in the examination to focus the investigation in promising directions.

6.6.1 Comparison with the Existing Methodology

Differentiating this approach from previous research on using metadata to cluster the file artefacts, this research leverages expert human analysis results from the manual processing of previous investigations. The hypothesis that the features of analysed file artefacts can enable trained machine learning models to determine how relevant newly encountered file artefacts are to a specific type of investigation.

One advantage of the proposed method is performing supervised machine learning tasks on the investigated file artefacts; the automated categorisation can more directly steer the investigator's focus towards pertinent data at the earliest possible stage.

6.6.2 Benefits of this Approach

This approach leverages the suspect device's "super timeline" that consists of all levels of digital events, allowing comprehensive automated analysis on disk images. The approach outlined in this paper has the following potential benefits for digital forensic investigation:

- **Automated analysis**: Automated device image analysis on suspect devices performed immediately after acquisition can make full use of the computation infrastructure available and can help prioritise the expert human investigator's focus during the analysis phase.
- **Data-driven approach**: Many existing tools can only obtain insight specific to a current case. For example, keyword search and filtering tools are limited to the current device under investigation and lose the insights learning for future investigations. A data-driven approach enables the detection of likely pertinent artefacts that are more difficult to be detected by traditional approaches by leveraging what has been processed before. Applying existing knowledge to explore new, previously unencountered data could prove fruitful in expediting the discovery process.
- **Better performance as the known database grows**: The approach takes advantages of centralised evidence processing and their associated database. The performance of this approach can be improved as the centralised dataset of processed cases gets bigger; a juxtaposition to the current digital forensic volume challenge. This is due to the bigger the known hash database gets, the higher the chance of detecting known pertinent file artefacts, the better the predictions can become.

6.6.3 Limitations of this Approach

The objective of this work is to prioritise file artefacts and reduce the time needed for expert human file artefact examination. However, some limitations of the presented approach are observed:

- *Lack of known pertinent samples as input*: Known file artefacts are needed to train the machine learning models. The performance of the approach highly depends on the volume of previously analysed and categorised pertinent files. In real-world cases, multiple options should be implemented for investigators to choose for the different proportion of illegal files. For example, when detected illegal files are less than 1% of the benign, the calculation of similarity directly between the files may perform better than training a classification model.
- *False positive and negative errors are possible*: Important artefacts could be missed solely relying on this approach if some artefacts are misclassified. However, as an evidence prioritisation/triage step, this approach can assist the investigation's focus.

It is not intended as a substitution of existing analysis procedures. In fact, both data reduction discussed in Section 2.7, and triage in Section 2.5, are based on previous investigation experience. The purpose is to unearth meaningful information at the earliest time possible.

Consequently, this approach should be used to assist an investigation as a supplementary technique in conjunction with the existing investigation tools. Manual analysis is still a necessity before and after using this tool, but it is envisioned that this approach can expedite the overall processes.

Chapter 7: **Conclusion and Future Work**

7.1 Summary of the Work

7.1.1 Deduplicated Digital Evidence Acquisition System

This research explores a novel approach to collect digital evidence from a variety of devices and evaluates the reduced storage requirements. Improvements speed through several forensically-sound evidence acquisitions. The factors that influence the performance of the proposed system were also evaluated. From an analysis of the results, the contribution of the proposed system is as follows:

- Acquiring data from suspect devices are complete and verifiably accurate;
- Forensically-sound complete disk image reconstruction was achieved for all test data;
- As a beneficial product of the deduplicated acquisition process, preliminary evidence preprocessing has also taken place including metadata extraction and artefact hashing;
- The performance is better for disk images containing a high proportion of large file. For complete operating system images, the speed achieved shows great promise for the proposed technique. In a remote acquisition scenario, i.e., acquiring a forensic image over the Internet, the effective acquisition throughput speed is faster than the available broadband upload speed.

The importance of device triage has been discussed as it is a commonplace that multiple devices are seized during an investigation. Besides, the timeline generated from various devices contains millions of digital events. The detected pertinent files indicate the forensic value of the device. Devices with more known illegal file artefacts should be a higher priority for expert analysis. The detection of file artefact pertinent to the investigation offers an indication of where to initially focus the investigation.

7.1.2 Automated File Artefacts Relevancy Analysis

This system can also benefit from integrating evidence prioritisation targeting potentially pertinent evidence at the earliest stage of acquisition. An automatic approach to use machine learning models to predict which file artefacts are likely pertinent to an investigation (i.e. which file artefacts are likely more suspicious than others) was presented. It is designed to integrate with a DFaaS framework, rather than a stand-alone experiment on an individual device. The associated toolkit was introduced that was developed for supporting the automation of some the digital forensic process.

An approach that prioritises file artefacts that are similar to detected/analysed suspicious/relevant files was presented. The approach offers an option for faster detection of file artefacts likely to be relevant to the investigation. The automated process is assisted by developed feature extraction tools and machine learning models. The developed tools were tested on the datasets generated. The results show the advantages of the approach and the promising result acquired. As a result, it should be used at an early stage in the examination to focus the investigator. Any bias would be hinged on the bias from the training data. In the real world application of the system, the training data would be evidence classified by forensic analysts; tools developed by this approach does not make the decision that files are illegal or benign.

Example scenarios were outlined and tested, indicating the feasibility and effectiveness of the proposed methodology. The promising experimental results for suspicious artefact detection provides motivation for further research.

7.2 Conclusion

The proposed digital evidence processing system leverages data deduplication to combat the big forensic data challenge. Building upon previous work in the area of data deduplication for digital forensics, stored evidence classification facilitates automated guidance for investigators in the analysis of new, previously unencountered data. Ultimately, a novel forensically sound reconstruction technique facilitates verifying the integrity of the deduplicated data acquisition system in a manner that courts have come to accept. The system eliminates repeated data processing, recognises file artefacts pertinent to the investigation and automated determination of file artefacts are likely to be pertinent to the investigation. The detected pertinent file artefacts can offer an indication for the rest of the investigation and assist the device triage.

Evaluation of the proposed methodologies requires various test disk images. The implemented tool, EviPlant, is used for disk image creation and manipulation. TraceGen is a framework implemented for automated generation of wear-and-tear on disk device; user action is auto-

mated emulated on a computer system running in VirtualBox. Test disk images are generated by TraceGen with emulated investigation scenarios.

Experimentation results presented proves the validity of the proposed methodologies. The f1-score of pertinent file artefact classification achieves 90% on average in the emulated investigation scenarios. The illegal file artefacts on the test disk image, assuming unknown by the system, are ranked at the top 10% for expert analysis. The proposed system can reduce the amount of data and allow the expert to be focused on interesting devices and file artefacts. This system can improve the efficiency of digital evidence processing, which is crucial to alleviate the digital evidence backlogs problem. Digital evidence backlogs can be alleviated by leveraging the proposed system.

7.3 Future Work

For the experiments in this thesis, the illegal and benign ratio for testing is 1:10. In real case investigation, the ratio could be less than 1:100. Experiments should be conducted in the future to find out what the minimum ratio is to allow the proposed ML algorithms work and what effect various ratios of benign to illegal content will have on the overall performance and accuracy of the system.

The file timeline generation process extracts digital events associated with file artefact. The pluralisation of the data processing should be implemented using the MapReduce algorithm to expiate this process.

The emulated scenarios for testing in this research are file artefacts and user actions on a disk drive. A real case could involve more evidentiary sources, e.g., IoT and network traffic. The proposed system and analysed approach should be tested under more complex scenarios with multiple devices.

TraceGen has shown that a better approach to automated disk image generation is possible and that it could have significant benefits in the areas of digital forensic teaching, research, and tool and process validation. At the same time, there is still a long way to go. Generation of test data, such as computer network traffic, the usage of device data synchronisation, etc., is not covered by TraceGen yet. These emulations are necessary for testing of multiple device analysis approaches.

7.3.1 Multiple Devices Investigation Scenario

When multiple computers and IoT devices are found during an investigation, the detected illegal file artefacts can guide the subsequent investigation. For example, analysis of the

network and IoT logs at the time of the file's creation and last access time. In this case, more sophisticated emulation is needed.

- Generating more complex scenarios - The experimental scenario used in this thesis was child abuse material possession and distribution investigation. The purpose of this case type is to find out files with similar size, directory, creation time, etc. More investigative scenarios will be designed and evaluated. For example, training models for determining suspicious files through each origin source (from an email attachment, cloud accounts, USB devices, etc.), third party owner, etc.

- Cross-device analysis - This can be conducted through analysis on a combined timeline from multiple devices. Seized devices and evidence sources from the same case or suspect can be joined together such as combining disk image artefacts with an email account, cloud service account, file transfer services, etc.

  The system detects pertinent file artefact, user actions on this file provide valuable information. For example, the creation and last access to the file, what other activities on the device occurred? User's network activity and IoT devices from combined timeline could also provide information pertinent to the investigation. A tool for extracting digital events of a given time interval should be developed for supporting this analysis.

7.3.2 Exploration of Advanced Machine Learning Techniques

For the investigation of more complex cases, advanced machine learning techniques could improve the performance.

- Feature extraction from file content: Further extension of this approach will integrate the files' content as input features. For example, computer vision techniques can be applied to image and video file analysis; Natural Language Processing techniques can be applied to document file analysis.

- Further exploration is required on the extent of imbalanced classes influencing each model's performance. Form the experiments conducted in this research, it was shown that the dataset could influence the performance of the models. In future work, more diverse datasets will be generated for testing. The generation of these datasets should provide a wide variety of the ratios of the benign/illegal files.

- Exploring appropriate feature selection and feature engineering approach for the defined machine learning tasks. In this thesis, the features are selected by the common knowledge of the digital forensic investigation. One avenue of exploration that could

improve the performance is extracting as many features as possible initially, and subsequently selecting from them by using techniques such as *SelectBest*, *RFE (Recursive Feature Elimination)*, and *PCA (Principal Component Analysis)*.

- Tuning the model so that it can fit the needs of additional investigative scenarios better. For example, to focus on the recall scores, rather than insisting on a higher precision. Because, in any digital forensic investigation, any pertinent inculpatory or exculpatory file artefact must not be inadvertently overlooked.

Bibliography

[1] Xiaodong Lin. Examining NTFS File System. In *Introductory Computer Forensics*, pages 163–197. Springer, 2018.

[2] Xiaoyu Du, Christopher Hargreaves, John Sheppard, and Mark Scanlon. TraceGen: User Activity Emulation for Digital Forensic Test Image Generation. *Forensic Science International: Digital Investigation*, 03 2021.

[3] Mark Scanlon, Xiaoyu Du, and David Lillis. EviPlant: An Efficient Digital Forensic Challenge Creation, Manipulation and Distribution Solution. *Digital Investigation*, 20:S29–S36, 2017.

[4] Xiaoyu Du, Chris Hargreaves, John Sheppard, Felix Anda, Asanka Sayakkara, Nhien-An Le-Khac, and Mark Scanlon. SoK: Exploring the State of the Art and the Future Potential of Artificial Intelligence in Digital Forensic Investigation. In *Proceedings of the 15th International Conference on Availability, Reliability and Security*, pages 1–10, 2020.

[5] Xiaoyu Du, Quan Le, and Mark Scanlon. Automated Artefact Relevancy Determination from Artefact Metadata and Associated Timeline Events. In *The 6th IEEE International Conference on Cyber Security and Protection of Digital Services (Cyber Security)*. IEEE, 06 2020.

[6] Xiaoyu Du and Mark Scanlon. Methodology for the Automated Metadata-Based Classification of Incriminating Digital Forensic Artefacts. In *Proceedings of the 14th International Conference on Availability, Reliability and Security*, page 43. ACM, 2019.

[7] Xiaoyu Du, Paul Ledwith, and Mark Scanlon. Deduplicated disk image evidence acquisition and forensically-sound reconstruction. In *2018 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/12th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE)*, pages 1674–1679. IEEE, 2018.

[8] Xiaoyu Du, Nhien-An Le-Khac, and Mark Scanlon. Evaluation of Digital Forensic Process Models with Respect to Digital Forensics as a Service. In *Proceedings of the 16th European Conference on Cyber Warfare and Security (ECCWS 2017)*, pages 573–581, Dublin, Ireland, 06 2017. ACPI.

[9] Francesco Servida and Eoghan Casey. IoT Forensic Challenges and Opportunities for Digital Traces. *Digital Investigation*, 28:S22–S29, 2019.

[10] Shams Zawoad and Ragib Hasan. Digital Forensics in the Age of Big Data: Challenges, Approaches, and Opportunities. In *2015 IEEE 17th International Conference on High Performance Computing and Communications, 2015 IEEE 7th International Symposium on Cyberspace Safety and Security, and 2015 IEEE 12th International Conference on Embedded Software and Systems*, pages 1320–1325. IEEE, 2015.

[11] Simson L Garfinkel. Digital Forensics Research: The Next 10 Years. *digital investigation*, 7:S64–S73, 2010.

[12] David Lillis, Brett Becker, Tadhg O'Sullivan, and Mark Scanlon. Current Challenges and Future Research Areas for Digital Forensic Investigation. In *The 11th ADFSL Conference on Digital Forensics, Security and Law (CDFSL 2016)*, pages 9–20. ADFSL, 2016.

[13] Oluwasola Mary Adedayo. Big Data and Digital Forensics. In *2016 IEEE International Conference on Cybercrime and Computer Forensic (ICCCF)*, pages 1–7. IEEE, 2016.

[14] Eoghan Casey, Monique Ferraro, and Lam Nguyen. Investigation Delayed Is Justice Denied: Proposals for Expediting Forensic Examinations of Digital Evidence. *Journal of forensic sciences*, 54(6):1353–1364, 2009.

[15] Luca Caviglione, Steffen Wendzel, and Wojciech Mazurczyk. The Future of Digital Forensics: Challenges and the Road Ahead. *IEEE Security & Privacy*, 15(6):12–17, 2017.

[16] Reza Montasari, Richard Hill, Simon Parkinson, Pekka Peltola, Amin Hosseinian-Far, and Alireza Daneshkhah. Digital Forensics: Challenges and Opportunities for Future Studies. *International Journal of Organizational and Collective Intelligence (IJOCI)*, 10(2):37–53, 2020.

[17] Michael M Losavio, KP Chow, Andras Koltay, and Joshua James. The Internet of Things and the Smart City: Legal Challenges with Digital Forensics, Privacy, and Security. *Security and Privacy*, 1(3):e23, 2018.

[18] Alessandro Guarino. Digital Forensics as a Big Data Challenge. In *ISSE 2013 securing electronic business processes*, pages 197–203. Springer, 2013.

[19] Amin Azmoodeh, Ali Dehghantanha, and Kim-Kwang Raymond Choo. Big Data and Internet of Things Security and Forensics: Challenges and Opportunities. In *Handbook of Big Data and IoT Security*, pages 1–4. Springer, 2019.

[20] Stavros Simou, Christos Kalloniatis, Stefanos Gritzalis, and Vasilios Katos. A Framework for Designing Cloud Forensic-enabled Services (CFeS). *Requirements Engineering*, 24(3):403–430, 2019.

[21] Jooyoung Lee and Sungyong Un. Digital Forensics as a Service: A Case Study of Forensic Indexed Search. In *ICT Convergence (ICTC), 2012 International Conference on*, pages 499–503. IEEE, 2012.

[22] Giuseppe Totaro, Massimo Bernaschi, Giancarlo Carbone, Marco Cianfriglia, and Antonio Di Marco. ISODAC: A High Performance Solution for Indexing and Searching Heterogeneous Data. *Journal of Systems and Software*, 118:115–133, 2016.

[23] Paul Joseph and Jasmine Norman. Forensic Corpus Data Reduction Techniques for Faster Analysis by Eliminating Tedious Files. *Information Security Journal: A Global Perspective*, 28(4-5):136–147, 2019.

[24] Darren Quick and Kim-Kwang Raymond Choo. Big Forensic Data Reduction: Digital Forensic Images and Electronic Evidence. *Cluster Computing*, 19(2):723–740, 2016.

[25] Rodney McKemmish. When is Digital Evidence Forensically Sound? In *IFIP international conference on digital forensics*, pages 3–15. Springer, 2008.

[26] Quang Do, Ben Martini, and Kim-Kwang Raymond Choo. A Forensically Sound Adversary Model for Mobile Devices. *PloS one*, 10(9):e0138449, 2015.

[27] Nurul Hidayah Ab Rahman, William Bradley Glisson, Yanjiang Yang, and Kim-Kwang Raymond Choo. Forensic-by-design Framework for Cyber-physical Cloud Systems. *IEEE Cloud Computing*, 3(1):50–59, 2016.

[28] Carolyn M Burns, Jeff Morley, Richard Bradshaw, and José Domene. The Emotional Impact on and Coping Strategies Employed by Police Teams Investigating Internet Child Exploitation. *Traumatology*, 14(2):20–31, 2008.

[29] Nicole Beebe and Jan Clark. Dealing with Terabyte Data Sets in Digital Investigations. In *IFIP International Conference on Digital Forensics*, pages 3–16. Springer, 2005.

[30] Martin Karresand, Stefan Axelsson, and Geir Olav Dyrkolbotn. Using NTFS Cluster Allocation Behavior to Find the Location of User Data. *Digital Investigation*, 29:S51–S60, 2019.

[31] Fabio Marturana and Simone Tacconi. A Machine Learning-based Triage Methodology for Automated Categorization of Digital Media. *Digital Investigation*, 10(2):193–204, 2013.

[32] Nicole Lang Beebe and Lishu Liu. Ranking Algorithms for Digital Forensic String Search Hits. *Digital Investigation*, 11:S124–S132, 2014.

[33] Sriram Raghavan. Digital Forensic Research: Current State of the Art. *CSI Transactions on ICT*, 1(1):91–114, 2013.

[34] Humaira Arshad, Aman Bin Jantan, and Oludare Isaac Abiodun. Digital Forensics: Review of Issues in Scientific Validation of Digital Evidence. *Journal of Information Processing Systems*, 14(2), 2018.

[35] Ankit Agarwal, Megha Gupta, Saurabh Gupta, and Subhash Chandra Gupta. Systematic Digital Forensic Investigation Model. *International Journal of Computer Science and Security (IJCSS)*, 5(1):118–131, 2011.

[36] Nhien-An Le-Khac, Daniel Jacobs, John Nijhoff, Karsten Bertens, and Kim-Kwang Raymond Choo. Smart Vehicle Forensics: Challenges and Case Study. *Future Generation Computer Systems*, 109:500–510, 2020.

[37] Dario Lanterna and Antonio Barili. Forensic Analysis of Deduplicated File Systems. *Digital Investigation*, 20:S99–S106, 2017.

[38] Mark Scanlon. Battling the Digital Forensic Backlog through Data Deduplication. In *2016 Sixth International Conference on Innovative Computing Technology (INTECH)*, pages 10–14. IEEE, 2016.

[39] Nicole Beebe. Digital Forensic Research: The Good, the Bad and the Unaddressed. In *IFIP International Conference on Digital Forensics*, pages 17–36. Springer, 2009.

[40] Eoghan Casey. *Digital Evidence and Computer Crime: Forensic Science, Computers, and the Internet*. Academic press, 2011.

[41] Ben Hitchcock, Nhien-An Le-Khac, and Mark Scanlon. Tiered Forensic Methodology Model for Digital Field Triage by Non-Digital Evidence Specialists. *Digital Investigation*, 16(S1):75–85, 03 2016. Proceedings of the Third Annual DFRWS Europe.

[42] Quan Le, Oisín Boydell, Brian Mac Namee, and Mark Scanlon. Deep Learning at the Shallow End: Malware Classification for Non-domain Experts. *Digital Investigation*, 26:S118–S126, 2018.

[43] Antonio Grillo, Alessandro Lentini, Gianluigi Me, and Matteo Ottoni. Fast User Classifying to Establish Forensic Analysis Priorities. In *IT Security Incident Management and IT Forensics, 2009. IMF'09. Fifth International Conference on*, pages 69–77. IEEE, 2009.

[44] Rick Ayers, Sam Brothers, and Wayne Jansen. Guidelines on Mobile Device Forensics. *NIST Special Publication*, 1(1):85, 2014.

[45] Shancang Li, Kim-Kwang Raymond Choo, Qindong Sun, William J Buchanan, and Jiuxin Cao. IoT Forensics: Amazon Echo as a Use Case. *IEEE Internet of Things Journal*, 6(4):6487–6497, 2019.

[46] E Ramadhani and S Mulyati. Mapping the Use of Expert System as a Form of Cloud-based Digital Forensics Development. *Journal of Physics: Conference Series*, 1567(3):032032, 2020.

[47] Hyunji Chung, Jungheum Park, Sangjin Lee, and Cheulhoon Kang. Digital Forensic Investigation of Cloud Storage Services. *Digital investigation*, 9(2):81–95, 2012.

[48] Ameer Pichan, Mihai Lazarescu, and Sie Teng Soh. Cloud Forensics: Technical Challenges, Solutions and Comparative Analysis. *Digital investigation*, 13:38–57, 2015.

[49] Zubair A Baig, Patryk Szewczyk, Craig Valli, Priya Rabadia, Peter Hannay, Maxim Chernyshev, Mike Johnstone, Paresh Kerai, Ahmed Ibrahim, Krishnun Sansurooah, et al. Future Challenges for Smart Cities: Cyber-security and Digital Forensics. *Digital Investigation*, 22:3–13, 2017.

[50] Steve Watson and Ali Dehghantanha. Digital Forensics: the Missing Piece of the Internet of Things Promise. *Computer Fraud & Security*, 2016(6):5–8, 2016.

[51] Sapna Saxena and Neha Kishore. Big Data–Challenges and Opportunities in Digital Forensic. *BIG*, 2012.

[52] G Maria Jones and S Godfrey Winster. Forensics Analysis on Smart Phones Using Mobile Forensics Tools. *International Journal of Computational Intelligence Research*, 13(8):1859–1869, 2017.

[53] Kim-Kwang Raymond Choo and Ali Dehghantanha. *Contemporary Digital Forensic Investigations of Cloud and Mobile Applications*. Syngress, 2016.

[54] Sengul Dogan and Erhan Akbal. Analysis of Mobile Phones in Digital Forensics. In *2017 40th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, pages 1241–1244. IEEE, 2017.

[55] Niken Dwi Wahyu Cahyani, Nurul Hidayah Ab Rahman, William Bradley Glisson, and Kim-Kwang Raymond Choo. The Role of Mobile Forensics in Terrorism Investigations Involving the Use of Cloud Storage Service and Communication Apps. *Mobile Networks and Applications*, 22(2):240–254, 2017.

[56] Sneha C Sathe and Nilima M Dongre. Data Acquisition Techniques in Mobile Forensics. In *2018 2nd International Conference on Inventive Systems and Control (ICISC)*, pages 280–286. IEEE, 2018.

[57] Aiman Al-Sabaawi and Ernest Foo. A Comparison Study of Android Mobile Forensics for Retrieving Files System. *International Journal of Computer Science and Security (IJCSS)*, 13(4):148, 2019.

[58] Fabio Marturana, Gianluigi Me, Rosamaria Berte, and Simone Tacconi. A Quantitative Approach to Triaging in Mobile Forensics. In *2011IEEE 10th International Conference on Trust, Security and Privacy in Computing and Communications*, pages 582–588. IEEE, 2011.

[59] Edewede Oriwoh, David Jazani, Gregory Epiphaniou, and Paul Sant. Internet of Things Forensics: Challenges and Approaches. In *9th IEEE International Conference on Collaborative computing: networking, Applications and Worksharing*, pages 608–615. IEEE, 2013.

[60] Keyun Ruan, Joe Carthy, Tahar Kechadi, and Ibrahim Baggili. Cloud Forensics Definitions and Critical Criteria for Cloud Forensic Capability: An Overview of Survey Results. *Digital Investigation*, 10(1):34–43, 2013.

[61] Mark Scanlon, Jason Farina, Nhien-An Le Khac, and M-Tahar Kechadi. Leveraging Decentralisation to Extend the Digital Evidence Acquisition Window: Case Study on BitTorrent Sync. *Journal of Digital Forensics, Security and Law: Proc. of Sixth International Conference on Digital Forensics and Cyber Crime (ICDF2C 2014)*, pages 85–99, 09 2014.

[62] Yee-Yang Teing, Ali Dehghantanha, and Kim-Kwang Raymond Choo. CloudMe Forensics: A Case of Big Data Forensic Investigation. *Concurrency and Computation: Practice and Experience*, 30(5):e4277, 2018.

[63] Josiah Dykstra and Alan T Sherman. Acquiring Forensic Evidence from Infrastructure-as-a-service Cloud Computing: Exploring and Evaluating Tools, Trust, and Techniques. *Digital Investigation*, 9:S90–S98, 2012.

[64] M Edington Alex and R Kishore. Forensics Framework for Cloud Computing. *Computers & Electrical Engineering*, 60:193–205, 2017.

[65] BKSP Kumar Raju, Nikhil Bharadwaj Gosala, and G Geethakumari. CLOSER: Applying Aggregation for Effective Event Reconstruction of Cloud Service Logs. In *Proceedings of the 11th International Conference on Ubiquitous Information Management and Communication*, pages 1–8, 2017.

[66] BKSP Kumar Raju and G Geethakumari. Timeline-based Cloud Event Reconstruction Framework for Virtual Machine Artifacts. In *Progress in Intelligent Computing Techniques: Theory, Practice, and Applications*, pages 31–42. Springer, 2018.

[67] V. R. Kebande, N. M. Karie, and H. S. Venter. Cloud-Centric Framework for Isolating Big Data as Forensic Evidence from IoT Infrastructures. In *2017 1st International Conference on Next Generation Computing Applications (NextComp)*, pages 54–60, 2017.

[68] Jung Hyun Ryu, Pradip Kumar Sharma, Jeong Hoon Jo, and Jong Hyuk Park. A Blockchain-based Decentralized Efficient Investigation Framework for IoT Digital Forensics. *The Journal of Supercomputing*, 75(8):4372–4387, 2019.

[69] Da-Yu Kao, Ni-Chen Wu, and Fuching Tsai. A Triage Triangle Strategy for Law Enforcement to Reduce Digital Forensic Backlogs. In *2020 22nd International Conference on Advanced Communication Technology (ICACT)*, pages 1173–1179. IEEE, 2020.

[70] Laura Sanchez, Cinthya Grajeda, Ibrahim Baggili, and Cory Hall. A Practitioner Survey Exploring the Value of Forensic Tools, AI, Filtering, & Safer Presentation for

Investigating Child Sexual Abuse Material (CSAM). *Digital Investigation*, 29:S124–S142, 2019.

[71] Joshua I James and Pavel Gladyshev. Challenges with Automation in Digital Forensic Investigations. *arXiv preprint arXiv:1303.4498*, 2013.

[72] Adrian Shaw and Alan Browne. A Practical and Robust Approach to Coping with Large Volumes of Data Submitted for Digital Forensic Examination. *Digital Investigation*, 10(2):116–128, 2013.

[73] Graeme Horsman. Tool Testing and Reliability Issues in the Field of Digital Forensics. *Digital Investigation*, 28:163–175, 2019.

[74] Cinthya Grajeda, Frank Breitinger, and Ibrahim Baggili. Availability of Datasets for Digital Forensics–and What is Missing. *Digital Investigation*, 22:S94–S105, 2017.

[75] Simson Garfinkel, Paul Farrell, Vassil Roussev, and George Dinolt. Bringing Science to Digital Forensics with Standardized Forensic Corpora. *digital investigation*, 6:S2–S11, 2009.

[76] M Al Fahdi, NL Clarke, F Li, and SM Furnell. A Suspect-oriented Intelligent and Automated Computer Forensic Analysis. *Digital Investigation*, 18:65–76, 2016.

[77] Kam Woods, Christopher Lee, Simson Garfinkel, David Dittrich, Adam Russel, and Kris Kearton. Creating Realistic Corpora for Forensic and Security Education. In *Proceedings of the ADFSL Conference on Digital Forensics, Security and Law*, pages 123–134, 2011.

[78] Christian Moch and Felix C Freiling. The Forensic Image Generator Generator (Forensig2). In *IT Security Incident Management and IT Forensics, 2009. IMF'09. Fifth International Conference on*, pages 78–93. IEEE, 2009.

[79] Niels Provos and Thorsten Holz. *Virtual Honeypots: From Botnet Tracking to Intrusion Detection*. Addison-Wesley Professional, first edition, 2007.

[80] Felix C. Freiling, Thorsten Holz, and Martin Mink. Reconstructing Peoples Lives: A Case Study in Teaching Forensic Computing. In *In Proceedings of the 4th International Conference on IT Incident Management & IT Forensics (IMF 2008*, 2008.

[81] Simson Garfinkel. Forensic Corpora: A Challenge for Forensic Research. *Electronic Evidence Information Center, April*, pages 1–10, 2007.

[82] Simson Garfinkel. Lessons Learned Writing Digital Forensics Tools and Managing a 30TB Digital Evidence Corpus. *Digital Investigation*, 9:S80–S89, 2012.

[83] Ibrahim Baggili and Frank Breitinger. Data Sources for Advancing Cyber Forensics: What the Social World Has to Offer. In *2015 AAAI Spring Symposium Series*, 2015.

[84] Christian Moch and Felix C Freiling. Evaluating the Forensic Image Generator Generator. In *International Conference on Digital Forensics and Cyber Crime*, pages 238–252. Springer, 2011.

[85] York Yannikos, Lukas Graner, Martin Steinebach, and Christian Winter. Data Corpora for Digital Forensics Education and Research. In *IFIP International Conference on Digital Forensics*, pages 309–325. Springer, 2014.

[86] Michael Donovan Kohn, Mariki M Eloff, and Jan HP Eloff. Integrated Digital Forensic Process Model. *Computers & Security*, 38:103–115, 2013.

[87] Marcus K Rogers, James Goldman, Rick Mislan, Timothy Wedge, and Steve Debrota. Computer Forensics Field Triage Process Model. *Journal of Digital Forensics, Security and Law*, 1(2):19–38, 2006.

[88] Venansius Baryamureeba and Florence Tushabe. The Enhanced Digital Investigation Process Model. In *Proceedings of the Fourth Digital Forensic Research Workshop*, pages 1–9, 2004.

[89] Gary Palmer et al. A Road Map for Digital Forensic Research. In *First Digital Forensic Research Workshop, Utica, New York*, pages 27–30, 2001.

[90] Henry C Lee, Timothy Palmbach, and Marilyn T Miller. *Henry Lee's Crime Scene Handbook*. Academic Press, 2001.

[91] Mark Reith, Clint Carr, and Gregg Gunsch. An Examination of Digital Forensic Models. *International Journal of Digital Evidence*, 1(3):1–12, 2002.

[92] Nicole Lang Beebe and Jan Guynes Clark. A Hierarchical, Objectives-based Framework for the Digital Investigations Process. *Digital Investigation*, 2(2):147–167, 2005.

[93] Séamus Ó Ciardhuáin. An Extended Model of Cybercrime Investigations. *International Journal of Digital Evidence*, 3(1):1–22, 2004.

[94] Brian Carrier and Eugene H Spafford. An Event-based Digital Forensic Investigation Framework. In *Digital forensic research workshop*, pages 11–13, 2004.

[95] Sundresan Perumal. Digital Forensic Model Based on Malaysian Investigation Process. *International Journal of Computer Science and Network Security*, 9(8):38–44, 2009.

[96] Ben Martini and Kim-Kwang Raymond Choo. An Integrated Conceptual Digital Forensic Framework for Cloud Computing. *Digital Investigation*, 9(2):71–80, 2012.

[97] Darren Quick and Kim-Kwang Raymond Choo. 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, 2014.

[98] Richard P Mislan, Eoghan Casey, and Gary C Kessler. The Growing Need for On-scene Triage of Mobile Devices. *Digital Investigation*, 6(3-4):112–124, 2010.

[99] Noora Al Mutawa, Joanne Bryce, Virginia NL Franqueira, Andrew Marrington, and Janet C Read. Behavioural Digital Forensics Model: Embedding Behavioural Evidence Analysis into the Investigation of Digital Crimes. *Digital Investigation*, 28:70–82, 2019.

[100] Christopher Hargreaves and Angus Marshall. SyncTriage: Using Synchronisation Artefacts to Optimise Acquisition Order. *Digital Investigation*, 28:S134–S140, 2019.

[101] David McClelland and Fabio Marturana. A Digital Forensics Triage Methodology Based on Feature Manipulation Techniques. In *2014 IEEE International Conference on Communications Workshops (ICC)*, pages 676–681. IEEE, 2014.

[102] RB Van Baar, HMA Van Beek, and EJ Van Eijk. Digital Forensics as a Service: A Game Changer. *Digital Investigation*, 11:S54–S62, 2014.

[103] Yuanfeng Wen, Xiaoxi Man, Khoa Le, and Weidong Shi. Forensics-as-a-service (FaaS): Computer Forensic Workflow Management and Processing Using Cloud. In *The Fifth International Conferences on Pervasive Patterns and Applications*, pages 1–7, 2013.

[104] Jooyoung Lee and Dowon Hong. Pervasive Forensic Analysis Based on Mobile Cloud Computing. In *Multimedia Information Networking and Security (MINES), 2011 Third International Conference on*, pages 572–576. IEEE, 2011.

[105] S. Zawoad and R. Hasan. Digital Forensics in the Age of Big Data: Challenges, Approaches, and Opportunities. In *2015 IEEE 17th International Conference on High Performance Computing and Communications, 2015 IEEE 7th International Symposium on Cyberspace Safety and Security, and 2015 IEEE 12th International Conference on Embedded Software and Systems*, pages 1320–1325, 2015.

[106] HMA Van Beek, EJ van Eijk, RB van Baar, Mattijs Ugen, JNC Bodde, and AJ Siemelink. Digital forensics as a service: Game on. *Digital Investigation*, 15:20–38, 2015.

[107] Mark H Kryder and Chang Soo Kim. After Hard Drives—What Comes Next? *IEEE Transactions on Magnetics*, 45(10):3406–3413, 2009.

[108] Marcus Rogers. Forensic Evidence and Cybercrime. *The Palgrave Handbook of International Cybercrime and Cyberdeviance*, pages 425–445, 2020.

[109] Kathryn Watkins, Mike McWhorte, Jeff Long, and Bill Hill. Teleporter: An Analytically and Forensically Sound Duplicate Transfer System. *digital investigation*, 6:S43–S47, 2009.

[110] Sebastian Neuner, Martin Schmiedecker, and Edgar Weippl. Effectiveness of File-based Deduplication in Digital Forensics. *Security and Communication Networks*, 9(15):2876–2885, 2016.

[111] Sebastian Neuner, Martin Mulazzani, Sebastian Schrittwieser, and Edgar Weippl. Gradually Improving the Forensic Process. In *Availability, Reliability and Security (ARES), 2015 10th International Conference on*, pages 404–410. IEEE, 2015.

[112] Usama M Fayyad, Gregory Piatetsky-Shapiro, Padhraic Smyth, et al. Knowledge Discovery and Data Mining: Towards a Unifying Framework. In *KDD*, volume 96, pages 82–88, 1996.

[113] Eoghan Casey. Cutting Corners: Trading Justice for Cost Savings. *Digital investigation*, 3(4):185–186, 2006.

[114] Florian Buchholz and Eugene Spafford. On the Role of File System Metadata in Digital Forensics. *Digital Investigation*, 1(4):298–309, 2004.

[115] Neil C Rowe and Simson L Garfinkel. Finding Anomalous and Suspicious Files from Directory Metadata on a Large Corpus. In *International Conference on Digital Forensics and Cyber Crime*, pages 115–130. Springer, 2011.

[116] Jens Olsson and Martin Boldt. Computer Forensic Timeline Visualization Tool. *digital investigation*, 6:S78–S87, 2009.

[117] Christopher Hargreaves and Jonathan Patterson. An Automated Timeline Reconstruction Approach for Digital Forensic Investigations. *Digital Investigation*, 9:S69–S79, 2012.

[118] Kristinn Gudhjónsson. Mastering the Super Timeline with Log2timeline. *SANS Institute*, 2010.

[119] Mark Debinski, Frank Breitinger, and Parvathy Mohan. Timeline2GUI: A Log2timeline CSV Parser and Training Scenarios. *Digital Investigation*, 28:34–43, 2019.

[120] Sandeepak Bhandari and Vacius Jusas. An Abstraction Based Approach for Reconstruction of TimeLine in Digital Forensics. *Symmetry*, 12(1):104, 2020.

[121] Qian Chen, Qing Liao, Zoe L Jiang, Junbin Fang, Siuming Yiu, Guikai Xi, Rong Li, Zhengzhong Yi, Xuan Wang, Lucas CK Hui, et al. File Fragment Classification using Grayscale Image Conversion and Deep Learning in Digital Forensics. In *2018 IEEE Security and Privacy Workshops (SPW)*, pages 140–147. IEEE, 2018.

[122] Kristijan Vulinović, Lucija Ivković, Juraj Petrović, Kristian Skračić, and Predrag Pale. Neural Networks for File Fragment Classification. In *2019 42nd International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, pages 1194–1198. IEEE, 2019.

[123] Jiankun Hu Nour Moustafa and Jill Slay. A Holistic Review of Network Anomaly Detection Systems: A Comprehensive Survey. *J. Network and Computer Applications*, 128:33–55, 2019.

[124] Peter Flach. *Machine Learning: the Art and Science of Algorithms that Make Sense of Data*. Cambridge University Press, 2012.

[125] Shai Shalev-Shwartz and Shai Ben-David. *Understanding Machine Learning: From Theory to Algorithms*. Cambridge university press, 2014.

[126] Geoffrey Holmes, Andrew Donkin, and Ian H Witten. Weka: A Machine Learning Workbench. In *Proceedings of ANZIIS'94-Australian New Zealnd Intelligent Information Systems Conference*, pages 357–361. IEEE, 1994.

[127] Joydev Ghosh, Divya Kumar, and Rajesh Tripathi. Features Extraction for Network Intrusion Detection Using Genetic Algorithm (GA). In *Modern Approaches in Machine Learning and Cognitive Science: A Walkthrough*, pages 13–25. Springer, 2020.

[128] Eleazar Eskin, Andrew Arnold, Michael Prerau, Leonid Portnoy, and Sal Stolfo. A Geometric Framework for Unsupervised Anomaly Detection. In *Applications of data mining in computer security*, pages 77–101. Springer, 2002.

[129] Muhammad Naeem Ahmed Khan, Chris R Chatwin, and RC Young. Extracting Evidence from Filesystem Activity using Bayesian Networks. *International journal of Forensic computer science*, 1:50–63, 2007.

[130] Yanfang Ye, Dingding Wang, Tao Li, and Dongyi Ye. IMDS: Intelligent Malware Detection System. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1043–1047. ACM, 2007.

[131] Chris R. Chatwin Muhammad Naeem Khan and Rupert CD Young. A Framework for Post-event Timeline Reconstruction Using Neural Networks. *digital investigation*, 4(3-4):146–157, 2007.

[132] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep Learning. *nature*, 521(7553):436–444, 2015.

[133] Sebastian Ruder. An Overview of Gradient Descent Optimization Algorithms. *arXiv preprint arXiv:1609.04747*, 2016.

[134] Xinxi Wang and Ye Wang. Improving Content-based and Hybrid Music Recommendation Using Deep Learning. In *Proceedings of the 22nd ACM international conference on Multimedia*, pages 627–636, 2014.

[135] Devashish Shankar, Sujay Narumanchi, HA Ananya, Pramod Kompalli, and Krishnendu Chaudhury. Deep Learning Based Large Scale Visual Recommendation and Search for E-commerce. *arXiv preprint arXiv:1703.02344*, 2017.

[136] Dawei Yin, Yuening Hu, Jiliang Tang, Tim Daly, Mianwei Zhou, Hua Ouyang, Jianhui Chen, Changsung Kang, Hongbo Deng, Chikashi Nobata, et al. Ranking Relevance in Yahoo Search. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 323–332, 2016.

[137] Simson L Garfinkel, Aleatha Parker-Wood, Daniel Huynh, and James Migletz. An Automated Solution to the Multiuser Carved Data Ascription Problem. *IEEE Transactions on Information Forensics and Security*, 5(4):868–882, 2010.

[138] Simran Fitzgerald, George Mathews, Colin Morris, and Oles Zhulyn. Using NLP Techniques for File Fragment Classification. *Digital Investigation*, 9:S44–S49, 2012.

[139] Yu Wang, Luca Bondi, Paolo Bestagini, Stefano Tubaro, David J Edward Delp, et al. A Counter-Forensic Method for CNN-Based Camera Model Identification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 28–35, 2017.

[140] Amel Tuama, Frédéric Comby, and Marc Chaumont. Camera Model Identification with the Use of Deep Convolutional Neural Networks. In *Information Forensics and Security (WIFS), 2016 IEEE International Workshop on*, pages 1–6. IEEE, 2016.

[141] Francesco Marra, Giovanni Poggi, Carlo Sansone, and Luisa Verdoliva. A Deep Learning Approach for Iris Sensor Model Identification. *Pattern Recognition Letters*, 113:46–53, 2018.

[142] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards Deep Learning Models Resistant to Adversarial Attacks. *arXiv preprint arXiv:1706.06083*, 2017.

[143] Eoghan Casey, Sean Barnum, Ryan Griffith, Jonathan Snyder, Harm van Beek, and Alex Nelson. The Evolution of Expressing and Exchanging Cyber-investigation Information in a Standardized Form. In *Handling and Exchanging Electronic Evidence Across Europe*, pages 43–58. Springer, 2018.

[144] Johannes Schneider and Frank Breitinger. AI Forensics: Did the Artificial Intelligence System Do It? Why? *arXiv preprint arXiv:2005.13635*, 2020.

[145] Sriram Raghavan and SV Raghavan. A Study of Forensic and Analysis Tools. In *2013 8th International Workshop on Systematic Approaches to Digital Forensics Engineering (SADFE)*, pages 1–5. IEEE, 2013.

[146] Hussam Mohammed, Nathan Clarke, and Fudong Li. An Automated Approach for Digital Forensic Analysis of Heterogeneous Big Data. *Journal of Digital Forensics, Security and Law (JDFSL)*, 2016.

[147] Leyla Bilge and Tudor Dumitraş. Before We Knew It: An Empirical Study of Zero-Day Attacks in the Real World. In *Proceedings of the 2012 ACM Conference on Computer and Communications Security*, CCS '12, page 833–844, New York, NY, USA, 2012. Association for Computing Machinery.

[148] Simson L Garfinkel. Forensic Feature Extraction and Cross-drive Analysis. *digital investigation*, 3:71–81, 2006.

[149] Andrew Case, Andrew Cristina, Lodovico Marziale, Golden G Richard, and Vassil Roussev. FACE: Automated Digital Evidence Discovery and Correlation. *digital investigation*, 5:S65–S75, 2008.

[150] Fahad Alanazi and Andrew Jones. The Value of Metadata in Digital Forensics. In *2015 European Intelligence and Security Informatics Conference*, pages 182–182. IEEE, 2015.

[151] Sriram Raghavan and SV Raghavan. Determining the Origin of Downloaded Files Using Metadata Associations. *Journal of Communications*, 8(12):902–910, 2013.

[152] Sriram Raghavan and SV Raghavan. AssocGEN: Engine for Analyzing Metadata Based Associations in Digital Evidence. In *2013 8th International Workshop on Systematic Approaches to Digital Forensics Engineering (SADFE)*, pages 1–8. IEEE, 2013.

[153] BKSP Kumar Raju and G Geethakumari. Event Correlation in Cloud: A Forensic Perspective. *Computing*, 98(11):1203–1224, 2016.

[154] Flora Amato, Giovanni Cozzolino, Antonino Mazzeo, and Nicola Mazzocca. Correlation of Digital Evidences in Forensic Investigation through Semantic Technologies. In *2017 31st International Conference on advanced information networking and applications workshops (WAINA)*, pages 668–673. IEEE, 2017.

[155] Anuradha Gupta. Privacy Preserving Efficient Digital Forensic Investigation Framework. In *2013 Sixth International Conference on Contemporary Computing (IC3)*, pages 387–392. IEEE, 2013.

[156] Tom Yeh, Tsung-Hsiang Chang, and Robert C. Miller. Sikuli: Using GUI Screenshots for Search and Automation. In *Proceedings of the 22Nd Annual ACM Symposium on User Interface Software and Technology*, UIST '09, pages 183–192, New York, NY, USA, 2009. ACM.

[157] J. W. Hunt and M. D. McIlroy. An Algorithm for Differential File Comparison. *Computer Science*, 1975.

[158] Diederik P Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. CoRR abs/1412.6980, 2014.