applied sciences

MDPI

Article

Forensic Investigation of Remnant Data on USB Storage Devices Sold in New Zealand

Zawar Shah [1,*], Arkar Kyaw [2], Hong Phat Truong [3], Imdad Ullah [4] and Andrew Levula [1]

[1] Department of Information Technology, Sydney International School of Technology and Commerce, Sydney, NSW 2000, Australia; andrew.l@sistc.nsw.edu.au
[2] School of Information Technology, Wellington Institute of Technology (WelTec), Wellington 5012, New Zealand; arkar.kyaw@weltec.ac.nz
[3] School of Information Technology, Whitireia Community Polytechnic, Auckland 5022, New Zealand; hong.phat@whitireia.ac.nz
[4] College of Computer Engineering and Sciences, Prince Sattam Bin Abdulaziz University, Al-Kharj 11942, Saudi Arabia; i.ullah@psau.edu.sa
* zawar.s@sistc.nsw.edu.au

Abstract: The digital forensic tools used by law enforcement agencies for forensic investigations are mostly proprietary and commercially expensive; although open-source tools are used, the investigations conducted with such tools are not verified by reputable organisations, and hence, users are reluctant to practice such tools. To address this issue, we experimentally evaluate three open-source forensic tools based on various requirements recommended by the National Institute of Standards and Technology (NIST) framework for forensic investigation. The experimental setup consists of a forensic workstation, write-blocker, and purchased USB hard drives investigated via digital forensic imaging tools, i.e., `DC3DD`, `DCFLDD`, and `Guymager`. We create various test cases, which distribute USB hard drives in different groups and investigate the functional and optional requirements of NIST along with recovering and analysing remnant data. We evaluate these forensic tools by analysing the log information, following, anonymously (to ensure that data were not disclosed or misused during or after the investigations) collecting, examining, and classifying the remnant data restored from the USB hard drives. We observe that the percentage of hardware resources usage and the processing time of each tool are remarkably different, e.g., `Guymager` was the fastest tool and met all the functional requirements in each test case, but it utilised more CPU and memory resources than `DC3DD`, `DCFLDD`. We note that 88.23% of the USB hard drives contained sensitive personal or business information (e.g., personal photos, bank transactions, and contracts). Subsequently, the remnant data analysis shows that consumers in New Zealand are unaware of personal data security and the associated vulnerabilities of data leakages.

Keywords: forensic investigation; computer crime; digital devices; data leakage; data security

Citation: Shah, Z.; Kyaw, A.; Truong, H.P.; Ullah, I.; Levula, A. Forensic Investigation of Remnant Data on USB Storage Devices Sold in New Zealand. *Appl. Sci.* **2022**, *12*, 5928. https://doi.org/10.3390/app12125928

Academic Editors: Konstantinos Rantos, Konstantinos Demertzis and George Drosatos

Received: 21 April 2022
Accepted: 7 June 2022
Published: 10 June 2022

1. Introduction

Nowadays, digital forensic investigation plays an important role in criminal investigations [1,2] and requires many steps to collect the data and analyse the collected data to find evidence. There are many digital forensic tools available on the market (e.g., open-source tools, and commercial tools) to recover, collect, and analyse data; the forensic tools must follow the standards for digital evidence that are mentioned in a country's laws. It was reported [3] that unless forensic tools could generate quality data, the court would not accept the evidence. The forensic investigation and reporting could be any of three ways, i.e., technical, investigative, and evaluative, where each investigative type has a specific purpose and context [4]. Commercial forensic disk imaging tools are usually costly, and not all companies can afford to purchase these tools. These companies usually search for a tool within their budget or use a popular open-source tool. Based on the US case laws and

Federal Rules of Evidence and the Daubert standards [5], the procedures and approaches used to collect evidence must be tested experimentally. However, most researchers [6–8] use many open-source tools which do not meet the forensic requirement standards because these open-source tools have not been evaluated with suitable procedures and methods.

Researchers [6,7,9] use a mix of commercial and free tools (e.g., `EnCase`, `FTK Imager`, and `SANS SIFT Workstation`) in their studies based on a trust relationship with suppliers, and these tools are often compared and evaluated based on the Computer Forensics Tool Testing Program (CFTT) (the CFTT is a computer forensic tool testing framework developed by the NIST of the United States of America (USA) to evaluate the functionalities of any forensic tool [10,11]) framework. The functionalities or requirements for forensic tools mentioned in CFTT include the following: a forensic tool should be able to perform such functions as obtaining a digital source using a suitable interface (e.g., universal serial bus (USB), and firewire), notifying a user about the type of error from the source, generating a different type of log to fully audit results, etc. The previous research [12–14] used various functions required from NIST's CFTT framework to evaluate different forensic imaging tools. However, no researcher exhausted all the functional requirements mentioned in the NIST framework to evaluate different open-source forensic imaging tools. In addition, we note that no researchers had compared the hardware usage and time consumption of each forensic imaging tool.

Universal serial bus (USB) devices are commonly used storage devices due to their robustness, compact size, high storage capacity, low power consumption, ease of transportation and reasonable price [15]. With the development of such personal storage devices, the exchange of data for individuals and commercial entities through portable storage devices has become increasingly popular [6]. The stored data usually include personal and business information; however, these USB devices can be lost, stolen, or sold (typically on online auction websites), which can result in high-security risks for both personal privacy and commercially sensitive information [7,8,16,17]. This happens as a result of the users' carelessness, as they fail to forensically wipe out data before selling their devices online, leading to private or confidential data being exposed [7,17]. Therefore, it is essential to investigate the data found in second-hand USB devices sold (we targeted local markets in New Zealand), in addition, to investigate whether local users are aware of the related privacy or security issues. The authors [18] carried out a similar study on second-hand hard disks in 2011; however, the current study was carried out on second-hand USB external hard drives by testing all functional requirements as recommended by NIST and evaluating the comparison of the hardware usage and time consumption for the imaging process to identify the best open-source forensic imaging tool currently available.

The primary purpose of this research was to evaluate the open-source forensic tools based on the National Institute of Standards and Technology (NIST) Computer Forensics Tool Testing Program (CFTT) framework [10,11] and to identify their effectiveness. In this research, we purchased 17 USB external hard drives on the TradeMe (https://www.trademe.co.nz/, accessed on 7 June 2022) website. Among those, we determined that 15 of the USB hard drives contained sensitive data (e.g., personal info, movies, photos, bank account, and company information) and only two hard drives had securely deleted the data. We categorised the collected data from the USB device's images into various formats e.g., application data, audio, images, messages, text, documents, and video. The findings from this research compared with previous articles discovered similar issues, which is that the remnant data are still available in the second-hand storage devices sold online, and the users are still not aware of the security risk or threats from the data leak.

To carry out this study, we developed an experimental setup to discover remnant data from the purchased USB hard drives. We used three open-source digital forensic imaging tools, i.e., `DC3DD` (http://www.dc3.mil, accessed on 7 June 2022) [19], `DCFLDD` (http://dcfldd.sourceforge.net/, accessed on 7 June 2022) [20], and `Guymager` (http://guymager.sourceforge.net/, accessed on 7 June 2022) [21], to create images from the purchased USB hard drives. We generated four test cases to test the tool functionalities mentioned in the

NIST framework (these functionalities are presented in Appendix A). Each test case has its generic procedures, and suitable commands were applied for each forensic tool based on the guideline from the developers to collect accurate information and also to minimise error during the experiment. The selected tools were tested with the same test cases on different purchased USB external hard drives. We note that the `Guymager` meets most of the tool's functionality requests in all of the tests followed by `DC3DD`. The `DC3DD` imaging tool had the same functionalities as `Guymager`; however, the log file generated by `DC3DD` was less detailed compared to `Guymager`. The `DCFLDD` did not meet many requirements in different test cases, and there was no log file generated by `DCFLDD`. In addition, among the selected tools, the `Guymager` imaging tool used most of the hardware resources (i.e., CPU and memory), followed by `DC3DD` whereas `DCFLDD` consumed the fewest resources.

The following are the contributions of this work:

- To exhaustively test the selected open-source forensic imaging tools for fulfilment of the functional and optional requirements of imaging tools based on NIST's CFTT framework, compare the tool's hardware usage and time consumption for the imaging process.
- To experimentally acquire the best open-source forensic tools based on the various NIST requirements for forensic investigations.
- To understand the level of awareness of users in New Zealand towards the dangers lurking in sensitive data, which can be leaked through the sale of old hard drives.
- To summarise several common methods for users and businesses to limit or eliminate potential dangers before selling old hard drives.

Section 2 presents the background and importance of a suitable forensic tool for the investigation and the criteria to evaluate the suitable forensic tool. Section 3 explains the experimental methodology, which uses the NIST framework to evaluate the selected tools based on the tool functionalities, hardware usage, and time consumption. Section 4 explains the results obtained by carrying out the experiments. Section 5 discusses the findings from the forensic disk imaging tool experiments and compares the selected tools' performance in detail. Section 6 presents the current forensic disk imaging tools that are available on the market or were used by previous researchers to understand the evaluation process from the related articles. Finally, Section 7 concludes this work.

2. Background

The storage medium is an important component of computing systems and is any technology (including devices, e.g., flash drives, hard disks, RAID, and optical disks) that is used to store and retrieve electronic data relating to applications and users' information. The digital content could range from personal and sensitive data (e.g., personal documents, personal information, such as photo, name, address, phone number, or email address) to generic data (e.g., application preferences, pictures, video, music, programs, and operating system). USB devices are commonly used storage devices due to their robustness, compact size, high storage capacity, low power consumption, ease of transportation and reasonable price [15,22]. With the development of personal storage devices such as the thumb drive (a type of USB storage device), the exchange of data for individuals and commercial entities through portable storage devices has become increasingly popular [6], mainly for storing personal and business information. However, USB devices can be lost, stolen, or sold (typically on online auction websites). This can result in high-security risks for both personal privacy and commercially sensitive information [6,7]. In addition, the storage devices are also an important evidence component in criminal investigations (via digital forensic tools) involving corporate lawsuits, such as credit card fraud, intellectual property theft, and private content [23]. The rapid development and continuous change of technology push the method of collecting evidence from digital devices to also catch up with technology [24].

There are two types of forensic imaging tools: hardware and software. The hardware forensic imaging tool is outstanding in performance and less time consuming in creating forensic images, compared to the software forensic imaging tool. These tools are included in

management and control software exclusively for this system. These devices were designed with different types of connectors and cables suitable for a variety of storage devices [25]. On the other hand, the software-based forensic tools are broadly available in three major types, i.e., commercial, free and open-source tools, where most of the forensic tools used in law enforcement are often closed-source and developed specifically for organisations or are expensive commercial forensic tools. These commercial tools are usually fixed, patched, and updated with new features by the manufacturer; however, the customers have to wait for the vendor's updates [26]. On the other hand, free software may be a simplified version of commercial software, or it may be a product with basic features. However, these software are rarely updated and require error correction that often takes a long time to wait, or the software is not developed further due to lack of funds [27]. Additionally, the open-source tool allows anyone to see the software source code; the copyright holders allow anybody to change, develop and create a more distributed version of the software for any purpose [13]; and it could save other resources (e.g., money and updated patches) with the open-source tools compared to using commercial software [28].

2.1. Importance of a Suitable Forensic Tool for Investigation

Digital evidence is a series of information and data used for criminal investigation [24], which requires the extraction of valid data or information in specific computer file formats [29,30]. Table 1 shows different typical types of digital evidence that must meet various criteria before they can be accepted in the court of law [13]. We note that selecting a suitable forensic tool is very important because the court of law will not accept evidence if the forensic tools are not able to generate quality data [3]. In a forensic investigation, forensic imaging tools are used to create images from storage devices to ensure data integrity from original sources. In addition, these tools analyse, recover or reconstruct the data from the images created by imaging tools or directly from the original storage [24]. The US lawmakers have issued several laws to determine the standard for digital evidence and issued instructions on how to collect and process data evidence [5]. Digital evidence needs to meet the requirements in the code of Federal Rules of Evidence 702, which states that evidence is credible if it is collected and analysed based on proven methods and tools [13]. The authors in [31] mention three main mandatory requirements that a digital evidence must meet: (1) The information and data which are identified as evidence must be relevant to the case being investigated, (2) all relevant information must be collected and analysed using approved methods from the court, and (3) all the evidence must be accepted and confirmed by an appropriate authority.

Table 1. Features of digital evidence.

Digital Evidence	Explanation
Progressive Technology	Because of the ever-changing technology, the methods of collecting and analysing digital evidence need to change over time and keep pace with the development of technology.
Diversity	Digital evidence could be saved in different file formats and all types of information (e.g., video, audio, text, and image).
Modifiability and Duplicability	The data or information transferred over the network may be damaged or no longer preserve integrity. In addition, digital evidence can be easily modified, changed, stolen, or copied without leaving a trace.
Hiddenness	For any internet activities (e.g., web browsing, e-commerce, and email) the personal information, network traffic will be allocated across the internet. Thus, any information which is relevant and reliable will be considered digital evidence.
Cross the border of law	Digital crimes could happen in any nations and very hard to prosecute digital crimes due to different laws of each country.

The authors [32] claim that there are three main issues regarding digital evidence to ensure the acceptance of forensic evidence from the court of law, e.g., the investigation process must follow meticulous procedures to ensure admissible evidence for the court [33],

secondly, the investigators must have a suitable depth of knowledge and skills and be certified to use forensic tools [34], and finally, the evidence must be analysed, and its integrity must be preserved [3]. Due to the limited availability of forensic imaging tools, it may lead to unreliable evidence [34], which could ruin the entire investigation process, and criminals may evade fines or appear to be innocent for lack of persuasive evidence, which may cause investigation errors and the waste of time and money [26]. Hence, for the best practice in collecting digital evidence or any study relevant to data integrity, many researchers suggest using a write blocker (either hardware-based or software-based) to ensure data integrity, even if it may reduce the reading speed and increase the time for imaging process [35]. The authors in [36] mentioned the difference in the hash values when creating a forensic image with and without a write blocker. They set up a test environment to experiment with different scenarios to check whether the hash values would be changed from the same storage devices with and without the use of a write blocker. The result showed that when using a hardware-based or software-based write blocker, the MD5 hash value is the same, and the MD5 hash value is only different without using a write blocker. To explain the result, Refs. [37,38] mentioned that imaging tools usually do not change the data integrity of the hard drive; however, the operating system will change the hard drive register records, update the information for the hidden trash folder on the hard drive, and this causes the MD5 hash value to change.

2.2. Criteria for Evaluating Imaging Tools

There are many open-source digital forensic tools available on the market, and hence, it is of utmost importance to find out the most suitable tool to be used during a forensic investigation. Due to the importance of reliable evidence, the National Institute of Standards and Technology (NIST) created the Computer Forensics Tool Testing Program (CFTT) (https://www.nist.gov/itl/ssd/software-quality-group/computer-forensics-tool-testing-program-cftt, accessed on 7 June 2022) framework used to evaluate the function of forensic tools. Based on the NIST evaluation framework, many works are critically evaluated to evaluate and validate digital forensic tools [13,33,39,40], e.g., the necessary time to finish the process to create images, amount of collected raw data and evidential data from the forensic image. The Appendix A summarises a comprehensive list of requirements listed in the NIST CFTT framework. Nonetheless, none of the previous researchers used all the requirements listed in the NIST CFTT framework to evaluate the forensic imaging tools in their studies.

The focus of this study relates to the acquisition of USB external hard drives using a write blocker hardware or write blocker software connected via a computer over a USB 2.0 interface to create a forensic image file and generate the selected hash type for this image. The test is based on the NIST (2005) functional requirement list and identifies the extent to which the tools can fulfil all listed requirements. The test was repeated many times with different USB external hard drives applied to different forensic imaging tools. We checked the number of requirements set by NIST (listed in Appendix A) that the tool can adequately perform.

2.3. Brief Overview of Selected Forensic Imaging Tools

In the following, we present a short overview of the forensic tools selected in this work, i.e., `DC3DD`, `DCFLDD`, `Guymager`. Recall that these open-source tools are widely used for forensic investigations, hence, it is important to experimentally determine the best open-source forensic tools based on various NIST requirements. The `DC3DD` (https://www.kali.org/tools/dc3dd/, accessed on 7 June 2022) was developed by the Department of Defence Cyber Crime Center based on the original GNU DD source code (https://github.com/coreutils/coreutils/blob/master/src/dd.c, accessed on 7 June 2022) with added features for computer forensics, e.g., hashing on the fly with multiple algorithms (MD5, SHA-1, SHA-256, and SHA-512), comprehensive logging information. We found other imaging tools, such as automated image and restore (AIR) (https://www.linuxlinks.com/

automatedimageandrestore/, accessed on 7 June 2022) that is the graphical version of DC3DD. The first release of DC3DD was published in 2008 corresponding to the Coreutils (https://www.gnu.org/software/coreutils/, accessed on 7 June 2022). An important characteristic of the DC3DD is that the patches are automatically updated every time that DD is updated.

DCFLDD (http://dcfldd.sourceforge.net/, accessed on 7 June 2022) was developed based on the GNU source code, with modified and upgraded functional components to meet the requirements to become an approved tool for the investigation process. It has several features over the basic DD, e.g., hashing on-the-fly, status output, flexible disk wipes, image/wipe verification, and can send all its log data and output to commands. It has several features that are far more advanced than the DD tool, including its ability to calculate hash values for input data when the process of creating the image is underway, and to create an image after the imaging process by using hash values, and the tool can also split the forensic images or save the output image into many locations [41]. Similarly, the Guymager (https://guymager.sourceforge.io/, accessed on 7 June 2022) software is fast, free, and the most user-friendly forensic image tool for media acquisition. This software is designed to support the graphical user interface (GUI) to improve user friendliness and help users who do not know how to use the command line. The software was developed based on libewf (https://labs.ece.uw.edu/nsl/students/alomair/LB-Arabic/general/forensic-tools/libewf.html, accessed on 7 June 2022) and libguy tools by using a multi-threaded engine which helps computers with multi-processor and hyper-threading hardware support parallel compression for better performance [42].

Table 2 describes the functionality and differences between the three open-source forensic imaging tools.

Table 2. Overview of functionalities of selected forensic tool.

Function	DC3DD	DCFLDD	Guymager
Support Operating System	Windows and Linux	Windows and Linux	Linux
Input Source	Physical hard drive, logical volume, files, folders	Physical hard drive, logical volume, files, folders	Physical hard drive, logical volume, files, folders
Support Image Format	Raw, dd, img, split	Raw, dd, img, split	Raw, dd, img, E01, Ex01, aff
Partition Format Supports	NTFS, FAT and Linux partitions	NTFS, FAT and Linux partitions	NTFS, FAT and Linux partitions
Hash Values Supports	MD5, SH1, SH256, SH512	MD5, SH1, SH256, SH512	MD5, SH1, SH256
Verify Image Integrity	Yes	No	Yes
Split Image Into Sections	Yes	Yes	Yes
Forensic Clear Storage Device	Yes	Yes	No
Logging	Yes	Yes	Yes

3. Experimental Evaluation

Following, we present the experimental environment and various phases during the experimentation along with deviations in experiments; in addition, we present the remnant data collection and analysis method.

3.1. Experimental Environment

We experimentally evaluate the collected data from the created forensic images and verify the validity of the various forensic disk imaging tools. We note that a similar

research methodology was also used by previous researchers who carried out forensic data investigation [6,8,16,17]. For this research, a third-party person, not involved in the research, purchased 17 USB hard drives from different sellers on Trade Me (https://www.trademe.co.nz/, accessed on 7 June 2022) in New Zealand. The essential aim of this study is to test whether open-source forensic imaging tools meet the imaging tool functions required by NIST (Appendix A). These requirement functions were grouped and tested using different test scenarios, which were repeated on each purchased USB external hard drive. By creating different test cases together with the tool requirements that need to be tested, we used these test cases to evaluate the open-source forensic tools.

We used three open-source digital forensic imaging tools i.e., `DC3DD` version 7.2.646 [19], `DCFLDD` version 1.3.4-1 [20], and `Guymager` version 0.8.8 [21], to create images from the purchased USB devices. In addition, we used SHA-256sum to calculate the hash values for the source devices and create forensic images. All images were firmly securely stored in an encrypted folder, and the purchased USB devices were also stored in a secure place. Different open-source analysis tools used by many researchers including `Autopsy` [8], `WinHex` [6], and `ProDiscover Basic` were used to analyse the acquired images. Data obtained from the USB devices were grouped based on the file format (e.g., PDF, DOCX, or JPG). All gathered data were presented anonymously to ensure confidentiality and to ensure that the data were not disclosed or misused during or after the research. Based on the results, possible risks and factors leading to potential personal or commercial security risks were outlined. Figure 1 shows a broad overview of all the components (e.g., forensic tools, test cases, and tool functions requirements) along with the research contributions, as the final output of the proposed research. It further displays the principles of evaluating the open-source forensic tool and the evaluation process of tools and how the data are collected and analysed from the forensic image.

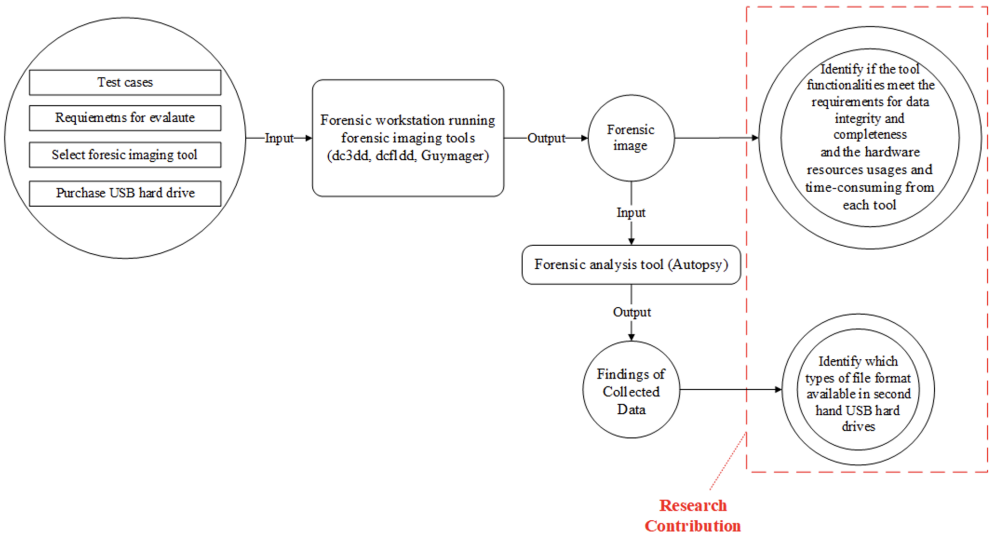

Figure 1. A broad overview of the research methodology for forensic data analysis.

The USB devices were read through the write-blocker device to ensure the integrity of the original data during the imaging process. We divided the USB devices into three categories i.e., small, medium, and large, respectively with storage capacities of 40 GB to 160 GB, 170 GB to 350 GB, and 360 GB to 640 GB, as shown in Table 3; complete information about these USB hard drives is presented in Appendix C (https://github.com/imdadullahunsw/forensics, accessed on 7 June 2022), specifically, hard drive ID, source device, source hashes with write blocker, and source hashes without write blocker. We carried out experimentations and analyses on two different machines, i.e., physical and virtual machines; all the tests for evaluating the selected tools were conducted on the virtual machine (e.g., calculating the hash values of the source devices before and after each test and conducting all test cases on `DC3DD`, `DCFLDD`and `Guymager`), while the process

of analysing the collected data was performed on the actual physical machine (such as analysing the created forensic images by using `Autopsy`). The specifications of the **physical** machine follows: *CPU*: Intel Core I7 6800 K (6 cores, 12 threads 3.50 GHz), *Motherboard*: Gigabyte X99P-SLI-CF, *BIOS Version*: F22, RAM: 32 GB DDR4 (2799.3 MHz), *Graphics Card*: NVIDIA GTX 1060 6 GB, *Operating System Drive*: Samsung SSD 950 PRO M2 256 GB, *Data Analysis Drive*: Samsung SSD EVO 850 SATA 500 GB, *Forensic Image Container Drive*: WD HDD Green WD30EZRX SATA 3 TB, *Operating System*: Windows 10 Enterprise version 1809. The specifications of the **virtual** machine follows: *CPU*: Intel Core I7 6800 K (4 cores, 8 threads 3.50 GHz), *Motherboard*: Gigabyte X99P-SLI-CF, *BIOS Version*: F22, *RAM*: 8 GB DDR4 (2799.3 MHz), *Graphics Card*: NVIDIA GTX 1060 6 GB, *Operating System Drive*: SSD EVO 850 SATA 500 GB, *Forensic Image Output Drive*: WD HDD Green WD30EZRX SATA 3 TB, *Operating System*: DEFT X virtual appliance.

Table 3. Second-hand USB external hard drives used for tool evaluation experiment (dUSB-HD means USB hard drive).

Categories	USB Storage	USB Drive ID
Small (40 to 160 GB)	40 GB	USB-HD-01
	120 GB	USB-HD-06
		USB-HD-07
	160 GB	USB-HD-12
		USB-HD-15
Medium (160 to 350 GB)	250 GB	USB-HD-05
		USB-HD-10
	320 GB	USB-HD-08
		USB-HD-14
Large (360 to 640 GB)	500 GB	USB-HD-03
		USB-HD-04
		USB-HD-11
		USB-HD-13
	640 GB	USB-HD-02
		USB-HD-09
Extra-Large (650 to 100 GB)	750 GB	USB-HD-17
	1000 GB	USB-HD-16
Total		17

3.2. Experimental Phases

This study involves five concrete steps, as shown in Figure 2, described below:

1. Selecting the forensic imaging tools, and tool function requirements based on the study of relevant articles (e.g., conference papers, journals, and reports) and the information gathered from other sources. An overview of the evaluation of forensic imaging tools is provided in this research. The authors [43] suggested the method of linking relevant functional requirements to each suitable test case. This method created an abstract level to identify the necessary functions for researchers, software developers, companies, and other users wishing to evaluate these forensic tools. CFTT specified the mandatory and optional tool functions for the forensic imaging tool, so all the function requirements used in this research were adapted from NIST (2004)

and NIST (2005), as presented in Appendix A. The complete list of test cases, which includes relevant test function requirements, can be read in Section 3.3.2.

2. Involves setting up a testing environment in the laboratory with the acquired USB storage devices, which were forensically imaged using different open-source forensic imaging tools (i.e., `DC3DD`, `DCFLDD`, and `Guymager`). Beforehand, the USBs were connected to the forensic work station, and the USB devices were read through the write-blocker device to ensure the integrity of the original data during the imaging process. Data were collected from the experiments, following, categorised based on the test cases.
3. Performing experiments to evaluate and compare the performance and functionality of forensic imaging tools based on the test cases and tool functions requirements.
4. Retrieving the data by using the forensic images created in Phase 2. All the collected data from forensic images of USB storage devices were analysed and categorised based on the file formats. Finally, the research findings were comprehensively discussed.

In Phase 2, a series of tests, developed in Phase 1, were performed based on the test cases. Four test cases were created to test the tool functionalities in four aspects, i.e., accuracy, completeness, completion speed, and hardware usage. By using the same test case on different USB external hard drives to evaluate these tools, the collected data could present the tool's functionality, hardware usage, and time consumption among different storage capacities. After the data were collected, a comparison table and graph were created showing the performance and tool's functionality of the selected tools. All the forensic images created in Phase 2 were used in Phase 4 to collect all the remnant data in the USB external hard drive via the Autopsy tool (a forensic analysis tool). The tool categorised all recovered files from the forensic images according to their file format.

The procedure to experiment can be listed in these steps as follows:

- Step 1: Setup and configured the forensic work station with the selected forensic tools on the suitable operating system.
- Step 2: Experiment with the selected tools and evaluate their functionalities based on the created test cases.
- Step 3: Collect the log files generated during the experiments and save the created images into another hard drive.
- Step 4: Analyse the created forensic images to collect the remnant data and categorise the data found based on the file format.
- Step 5: Verify the log files generated in Step 3 and specify whether the tool functionalities meet the request of verifying the passing or failing of the tool according to the tool function requirements.

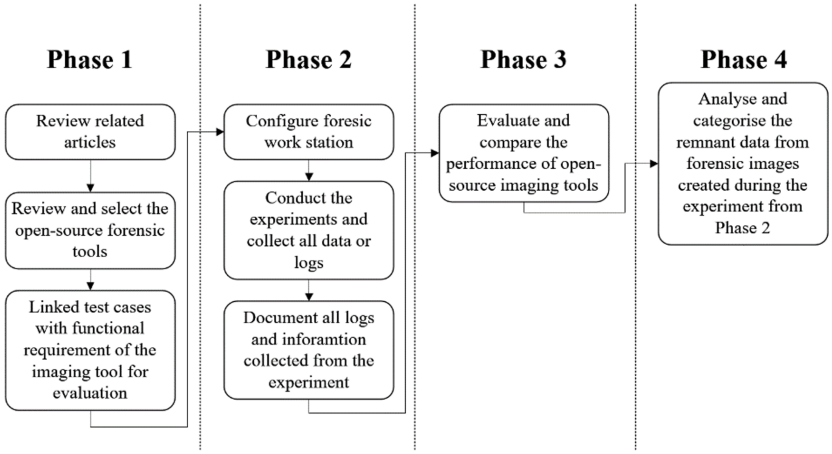

Figure 2. Forensic analysis phases.

3.3. Data Collection and Analysis

The following section describes the method to collect suitable data and the process to analyse the collected data from the experiments.

3.3.1. Research Data Collection Method

NIST is the first organisation to publish a standard method of evaluating and comparing forensic imaging tools, which is recognised by many organisations and industry professionals. In addition, several documents mention the use of preliminary tool test requirements from NIST CFTT framework in their research, e.g., [12–14,44]. This research also adapted tool functions requirements mentioned in NIST; however, due to limited time and space, this study used only 16 out of the 28 functions listed by NIST. The test cases were designed to use suitable tool function requirements from NIST (2005), presented in detail in Appendix A. These test cases focused on evaluating the forensic tool's accuracy, the forensic tool's completeness, speed of completion, and level of hardware usage. This research mainly focuses on USB external hard drive devices connected via `WiebeTech USB 2.0 Writeblocker`. The execution environment is the `DEFT X` virtual appliance (Linux based) for `DC3DD`, `DCFLDD`, and `Guymager`.

3.3.2. Test Cases and Procedures for Each Test Case

To fulfil the requirements for each test case, we configured the testing environment and the purchased USB external hard drives (as described in Table 3) to conduct the forensic evidence experimentation. The test case IDs, their descriptions, and the tested features of digital evidence i.e., mandatory and optional requirements (as presented in Appendix A, listed in the NIST CFTT framework) are presented in Table 4. Each test case has its generic procedures, while each tool has its specific commands and processes (procedures to conduct the evaluation of the tools and collect data are presented in Appendix B i.e., https://github.com/imdadullahunsw/forensics (accessed on 7 June 2022)) e.g., using `DCFLDD`, follow these steps (1). Find the disk on the machine using the following command: '`fdisk -l`', (2). Select the disk to create the image via '`echo ''started at: $(date)'' > timelog.txt; sudo dcfldd if=/dev/sdc hash=sha256 errlog=errorlog.txt hashconv=after hashlog=hashlog.txt bs=512 conv=noerror,sync of=image.raw; echo ''ended at: $(date)'' » timelog.txt`', and (3). Apply '`Dcfldd top`' to find out the *Task ID* and then '`pidstat -h -r -u -v -p` *Task ID* `30 » hardware-usage.txt`' to log the hardware resources usage, such as, CPU/Memory usage and time consumption, etc. The forensic tools were executed to acquire the data from the purchased hard drives based on the requests of the test case. After the image was created, and all the log files were collected, the hash values between the source and the created forensic image were compared. The analysis was conducted based on the collected logs to verify the tool's functionalities and performance.

Table 4. Test case IDs, their descriptions, and the tested features of digital evidence, i.e., mandatory and optional requirements.

Test Case ID	Description	Tested Features of Digital Evidence	
		Requirements Mandatory	Requirements Optional
FIT-ID-01	Create a forensic image from a USB external hard drive via a write blocker (hardware-based or software-based) and calculate hash values SHA-256 for the created image.	DI-RM-01, DI-RM-02, DI-RM-03, DI-RM-04, DI-RM-05, DI-RM-06, DI-RM-07, DI-RM-08	DI-RO-01, DI-RO-02, DI-RO-05, DI-RO-17
FIT-ID-02	Create multi-file images with the selected image file size (2 GB) from a USB external hard drive via a write blocker device and calculate hash values SHA-256 for the created image.		DI-RO-04, DI-RO-05, DI-RO-17

Table 4. *Cont.*

Test Case ID	Description	Tested Features of Digital Evidence	
		Requirements Mandatory	Requirements Optional
FIT-ID-03	Create a forensic image from a USB external hard drive via a write-blocker device or write-blocker software.		DI-RO-03, DI-RO-07, DI-RO-17
FIT-ID-04	Create a forensic image from a USB external hard drive without a write-blocker device or write-blocker software. The output image integrity will not be changed during the imaging process.		DI-RO-05, DI-RO-17, DI-RO-18

DI-RM means Digital Imaging Requirement Mandatory. DI-RO means Digital Imaging Requirement Optiona

3.3.3. Analysis of Collected Data

The gap analysis method was used to analyse the collected data used in the experiment. According to [45], the gap analysis method can classify the differences between the evaluated tools based on the collected results. This method also enables a comparison to be made between collected values and the test cases requirement's criteria [46]. The test cases requirement's criteria can be rating as "Passed" or "Failed" to validate the differences between the selected tools. The usage of numerical values in displaying the hardware usage level and the required time in finishing the requested task helped to identify which tools performed better. Hence, before recovering data from USB external hard drives, we set a few on the forensic analysis tool (i.e., `Autopsy`) to help the software automatically preliminarily classify different types of data (application, audio, image, message, text, and video) from the recovered original data. Following this, we reviewed the remnant data to find out what kind of data and information remained on these hard drives. Finally, we categorised all the remnant data collected from the purchased hard drives based on the data type format (e.g., photos, CV, videos, graduation letter, and driver's license).

3.4. Deviations in Experiments

We carried out the entire set of experimental tests based on the test plan presented in Sections 3.2 and 3.3. However, some deviations appeared during the experiment. Several tool functions requirements were not able to be applied in the experiment due to the limited budget and the difficulty of purchasing the suitable hard drives, which are required to test some specific functions. For example, the function 'DI-RM-07' requires testing on the faulty hard drive or faulty data sectors on a hard drive. However, all the purchased hard drives did not meet the requirements to test this function thoroughly. In addition, due to the time limitation, the clone functions from the selected tools were not tested (e.g., tool function ID: 'DI-RO-08', 'DI-RO-09', and 'DI-RO-10'); a summary of these results can be found in Table 5 Section 4.1.

Table 5. Summary of the success/failure rate of each forensic tool for optional and mandatory requirements under various test cases.

Digital Evidence Features	`DC3DD`	`DCFLDD`	`Guymager`	Test Case ID
Tested	DI-RM-01, DI-RM-02, DI-RM-03, DI-RM-04, DI-RM-05, DI-RM-06, DI-RM-07, DI-RM-08, DI-RO-01, DI-RO-02, DI-RO-05, DI-RO-17			FIT-ID-01
Failure	N/A	DI-RO-05 and DI-RO-17	N/A	
Tested	DI-RO-04, DI-RO-05, DI-RO-17			FIT-ID-02
Failure	N/A	N/A	N/A	
Tested	DI-RO-03, DI-RO-07, DI-RO-17			FIT-ID-03
Failure	DI-RO-07	DI-RO-07	DI-RO-07	
Tested	DI-RO-05, DI-RO-17, DI-RO-18			FIT-ID-04
Failure	N/A	DI-RO-05 and DI-RO-17	N/A	

As mentioned in Section 2.2, during the experiments, we had to change the hash value of the source devices as soon as the devices were connected to the forensic workstation without the write blocker. To be able to identify whether the selected tools changed the integrity of the digital source or not without the use of a write blocker, the hash values of the source device were checked before the experiment. Moreover, after the imaging process was done, the hash values of the source device was rechecked and then compared with the original hash values. The process of analysing the created forensic image required additional information (e.g., forensic case number, examiner name, address, and time zone). Additionally, the selected forensic analysis tool (`Autopsy`), which is an open-source tool, has several functions (data crawling functions) that did not operate as expected, and some of the recovered data were unreadable. NIST's (2005) tool function requirements were developed based on the IDE hard drives to analyse and evaluate the forensic imaging tools. In this research, the digital source devices were the USB external hard drives, and these devices were connected to the forensic station over the USB 2.0 interface, which is the most common interface used by external storage devices [6].

4. Experimental Results

This section presents the experimental results in detail, i.e., we start with the tool functions evaluations, following a detailed discussion over resource consumption for testing all cases and then we present a detailed analysis of collected remnant data.

4.1. Tool Functions Evaluation

Table 5 illustrates the experimental result after testing the selected forensic tools with different USB external hard drives. The test case IDs along with their descriptions and the tested mandatory and optional requirements are presented in Table 4. As mentioned in Section 3.4, there are various limitations of tested forensic tools, e.g., the function 'DI-RM-07' requires testing on a faulty hard drive or faulty data sectors on a hard drive. Hence, we note that in test case 'FIT-ID-01', only `DCFLDD` could not save the information of the imaging process to the log file ('DI-RO-17'), and it failed to check the hash values of the created image for the 'DI-RO-05' optional requirement, as mentioned in Table 5. The forensic imaging tools evaluation results are presented in Appendix D (https://github.com/imdadullahunsw/forensics, accessed on 7 June 2022), where we evaluated the four test cases (presented in Table 4) for various USB hard drives. Specifically, we presented the following: sample log information, time taken to create the image and verify the hash, reference hash vs. tool hash, result by requirement, and hardware usage (i.e., CPU and memory usage).

Similarly, for test case 'FIT-ID-02', all the tools met all the functionalities requirements. In addition, compared with the result with 'FIT-ID-01', the `DCFLDD` successfully saved the information in the log file and calculated the hash value information from the created image. Furthermore, for test case 'FIT-ID-03', all of the selected tools could not change the output location when the first selected output destination was full during the imaging process while testing the 'DI-RO-07' digital forensic functionality. In addition, in test case 'FIT-ID-03', the purchased second-hand USB external hard drives were connected to the forensic workstation via the write blocker. In test cases 'FIT-ID-01' and 'FIT-ID-02', the output destination was able to contain created log files and created a forensic image. However, in this test, the output destination was limited to only 3 GB, which was designed to test whether the selected forensic tool could notify or show the error of the limited output destination before the imaging process. Additionally, this test aims to test whether the tool can change the output destination during the imaging process. Furthermore, we note that, for test case 'FIT-ID-04', similar to 'FIT-ID-01', the `DCFLDD` did not successfully save all the information during the imaging process into the log file (for testing 'DI-RO-17'). In addition, the tool could not calculate hash values of the generated forensic image while testing the 'DI-RO-05' requirement.

4.2. Time Consumption for Imaging Process Evaluation

Figure 3 shows the time consumption for each forensic tool (i.e., `DC3DD`, `DCFLDD`, and `Guymager`) to finish the imaging process for each USB external hard drive under various test cases, i.e., 'FIT-ID-01' through 'FIT-ID-04'. Refer to the Table 5 for the success/failure rates of each forensic tool for optional and mandatory requirements under various test cases.

We note, on average, that `Guymager` (i.e., grey bar in Figure 3a–d) was the fastest tool and would finish the imaging process earlier compared to `DC3DD` and `DCFLDD` imaging tools. For example, for 'FIT-ID-01', `Guymager` took 1042 min to complete imaging process compared to `DCFLDD` and `DC3DD`, which respectively took 1360 min and 1227 min to finish the imaging process. Similarly, for 'FIT-ID-03', these calculations were respectively 80 s, 698 s, and 119 s, as presented in Figure 3c. Note that, in test case FIT-ID-03, the output location for all tools had the same storage capacity (3 GB) as mentioned in Section 4.1, which means the tools should stop the process of creating the forensic image in the same amount of time, even with a different storage capacity of USB hard drives. However, the finishing time for creating images of the selected tools was varied between each tool.

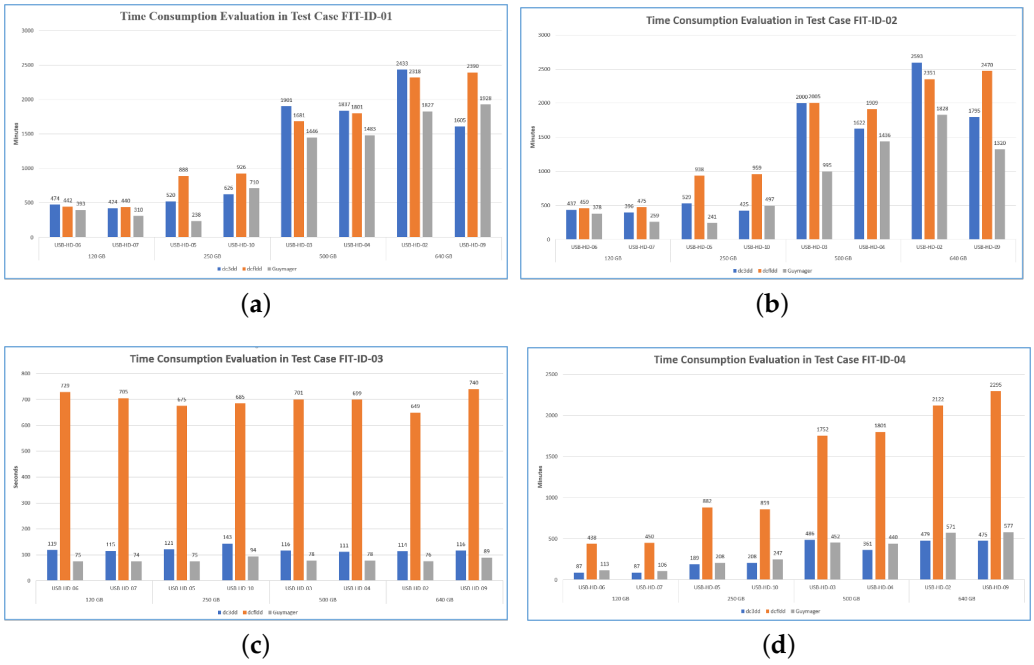

Figure 3. Time consumption for each forensic tools in different test case. (**a**) FIT-ID-01. (**b**) FIT-ID-02. (**c**) FIT-ID-03. (**d**) FIT-ID-04.

4.3. Hardware Resource Usage Evaluation

In the following, we present the CPU and memory usage evaluated from the experimentations with various forensic tools under various test cases for different functional and optional requirements, as presented in Table 5.

4.3.1. CPU Consumption

Test case FIT-ID-01: Figures 4a–d illustrate average CPU usage by selected forensic tools; note that we also evaluated the lowest and highest CPU usage with all the forensic tools for various test cases and categories of USB hard drives. However, due to space restrictions, we did not add those figures. We note that the `DC3DD` consumed the lowest CPU usage compared to `DCFLDD` and `Guymager`, where `DCFLDD` consumed the highest CPU usage. In addition, we note that in this test case, when compressing the image and calculating hash values, `Guymager` used the highest CPU resources while `DCFLDD` used the least. Based on our evaluations, we note that `DC3DD` and `Guymager` used mostly the same CPU resources

with different hard drive storage capacities. Overall, DC3DD used fewer CPU resources than Guymager, and DCFLDD was the tool that used the most CPU resources on average.

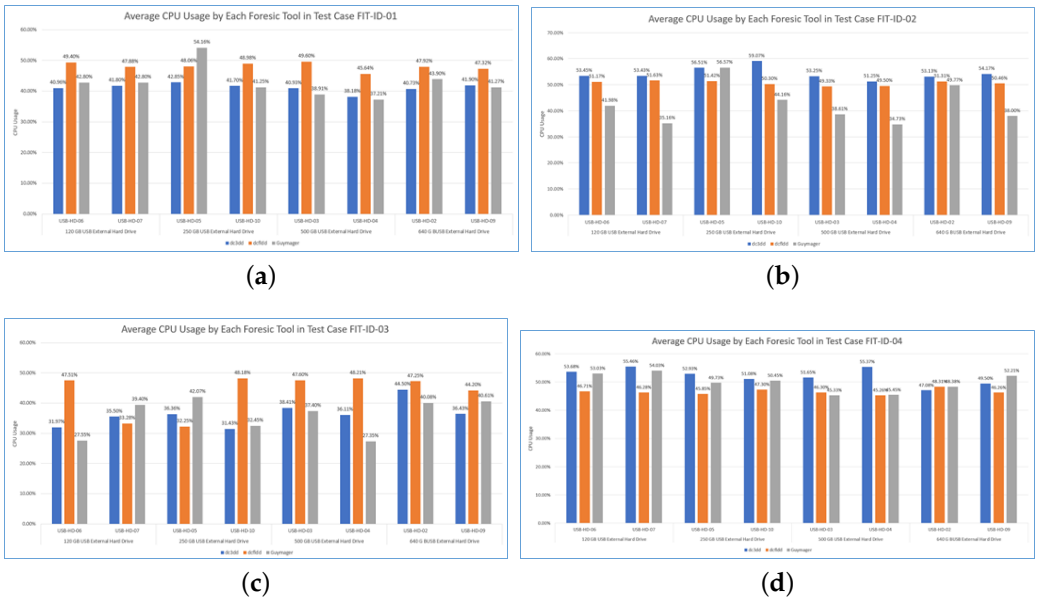

Figure 4. Average CPU consumption for each forensic tools in different test case. (**a**) FIT-ID-01. (**b**) FIT-ID-02. (**c**) FIT-ID-03. (**d**) FIT-ID-04.

Test Case FIT-ID-02: In Figure 4b, indicating the average CPU usage, the DC3DD used more CPU resources than the other tools. For example, for USB-HD-06, the average CPU usage for the DC3DD, DCFLDD, and Guymager respectively observed as 53.45%, 51.17%, and 41.98%. On average, as shown in Figure 4b, the DC3DD used most of the CPU resources compare to DCFLDD and Guymager.

Test Case FIT-ID-03: In Figure 4c, the lowest CPU usage of Guymager and DC3DD changed erratically during the experiment, while DCFLDD had a more stable use of CPU resources. We note that the DCFLDD used the highest CPU resources than Guymager or DCFLDD while testing over different capacity storage of USB devices. On average, as indicated in Figure 4c, DCFLDD utilised the majority of the CPU resources, whereas Guymager utilised the least CPU usage.

Test Case FIT-ID-04: In Figure 4d, the lower CPU usage with Guymager was observed 35%, whereas it was 30.77% with DC3DD and 32% with DCFLDD. Similarly, the highest CPU usages were observed as 90.77%, 76%, and 61% respectively with Guymager, DC3DD, and DCFLDD. On average, as depicted in Figure 4d, the DC3DD utilised most of the CPU resources, followed by Guymager and DCFLDD.

4.3.2. Memory Usage

Figure 5a–d depicts the memory resource used by the forensic tools. **Test Case FIT-ID-02**: We note that the amount of memory used by the tools did not change, even though the storage capacity of USB external hard drives was different, as depicted in Figure 5a–d. we note that there is a huge difference in using memory resources among three forensic tools; for example, Guymager used up to 949.8 MB on average. On the other hand, on average, DCFLDD used only 4.57 MB of memory resources. In conclusion, we note that Guymager used most of the memory resources, whereas DCFLDD utilised the least memory.

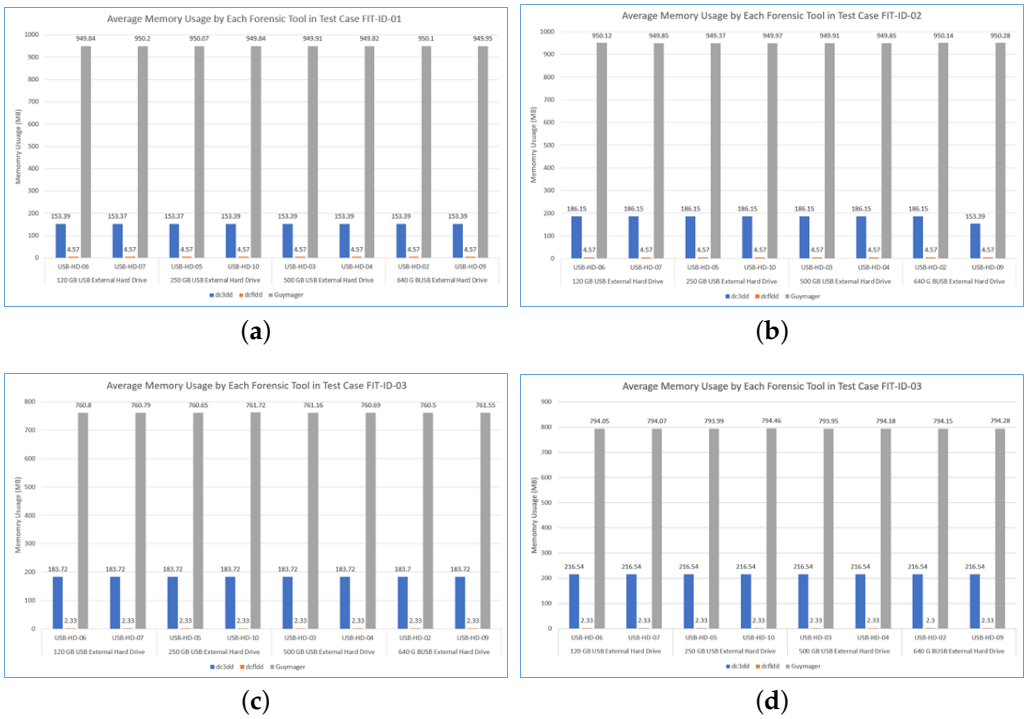

Figure 5. Average Memory consumption for each forensic tools in different test case. (**a**) FIT-ID-01. (**b**) FIT-ID-02. (**c**) FIT-ID-03. (**d**) FIT-ID-04.

Test Case FIT-ID-02: Similar to the test case 'FIT-ID-01', this test case resulted in similar memory usage for various forensic tools, i.e., `Guymager` utilised most of the memory resources compared to `DC3DD` and `DCFLDD`, evaluated for lower, medium, and highest memory usage. On average, we note that `Guymager` used 949 MB of memory resources for USB drives of different capacities. `DCFLDD` utilised the least memory usage. The average memory usage with these experimentations is shown in Figure 5b.

Test Case FIT-ID-03: In these experimentations, the lowest memory usage by `Guymager` was much higher (i.e., 728 MB) than `DC3DD` (183.72MB) and `DCFLDD` (2.33 MB). This trend in memory usage remained the same with the highest memory usage. Overall, on average (Figure 5c), `Guymager` used most of the memory resources, followed by `DC3DD` and `DCFLDD`.

Test Case FIT-ID-04: During these experiments, we note that `Guymager` used most of the memory resources (i.e., the lowest memory usage was 794 MB) compare with `DC3DD` (183 MB) and `DCFLDD` (2.33 MB). As noticed with other experiments under different test cases, this tendency of memory usage remained the same once we calculated the highest and average memory usage with these forensic tools. Overall, the average memory usage with the `Guymager` was highest, followed by the `DC3DD` and `DCFLDD`, as depicted in Figure 5.

4.4. Collected Remnant Data from Purchased USB External Hard Drive

Table 6 presents the purchased hard drives that we used to collect remnant data along with various types of data recovered from respective hard drives. In this research, different types of file format were categorised into six main categories as follows: application (e.g., EXE, JAVA, INI, C, F, and H), audio (e.g., MP3, WAV, and MIDI), image (e.g., PNG, JPG, and ICON), message (e.g., PST, and MBOX), text (e.g., TXT, DOC, DOCX, and XLSX), and video (e.g., WAV, AVI, and MKV). The purchased hard drives were divided into four categories, i.e., small, medium, large, and extra-large; in addition, the drive IDs along with their exact size (in gigabytes) are presented in this table. Complete information about the experimented USB hard drives is presented in Appendix C https://github.com/imdadullahunsw/forensics, accessed on 7 June 2022), specifically, hard drive ID,

source device, source hashes with write blocker, and source hashes without write blocker. After recovering data from the small storage USB hard drives, many data types were found, such as videos (e.g., movies, TV series, music, and videos), photos (e.g., family and wedding photos, car number plate, and contract payment), several personal documents (e.g., CV, and driving license) and organisation information (e.g., advertising documents and business contracts), and some sensitive information (e.g., bank information, and bank transaction records).

Table 6. Second-hand USB external hard drives used for tool evaluation along with various types of data recovered in respective hard drives.

Categories	USB Storage	USB Drive ID	Application	Audio	Image	Message	Text	Video
Small (40 to 160 GB)	40 GB	USB-HD-01	1045	1	24	0	2	87
	120 GB	USB-HD-06	105,561	1874	43,672	4	3208	338
		USB-HD-07	163	2	5	0	3	0
	160 GB	USB-HD-12	595,954	730	29,596	26	42,253	341
		USB-HD-15	7573	5	1542	0	2035	46
Medium (160 to 350 GB)	250 GB	USB-HD-05	205,169	5	21	0	66	0
		USB-HD-10	189,106	4603	11,338	0	542,232	415
	320 GB	USB-HD-08	19	4	3	0	0	0
		USB-HD-14	617,139	106	9229	950	285	25
Large (360 to 640 GB)	500 GB	USB-HD-03	80,893	14,967	103,738	0	160,454	1325
		USB-HD-04	536,671	4766	99,914	48	99,652	855
		USB-HD-11	9	5	1	0	0	0
		USB-HD-13	1,445,302	8215	92,111	4425	56,630	237
	640 GB	USB-HD-02	83,524	7166	82,847	69	3234	321
		USB-HD-09	52	448	7	0	0	57
Extra-Large (650 to 100 GB)	750 GB	USB-HD-17	2619	405	65	0	20	1305
	1000 GB	USB-HD-16	33	280	66	0	1255	2783

As an example, in hard drive USB-HD-01 (complete remnant data analysis with individual USB hard drives can be found in Appendix E: https://github.com/imdadullahunsw/forensics, accessed on 7 June 2022), many photos, videos, databases, and PDF documents were found. Among these files, there is some information from PDF files and photos which can be used to identify the previous owner (e.g., wedding photos, car plate number, and contract payment). Similarly, the previous owner of the hard drive USB-HD-12 did not format the hard drive properly because some remnant data were found in the hard drive after the recovery procedure. There are many sensitive documents recovered from the hard drive (e.g., contracts, bank statements, brochures, and credit card information) and many photos illustrated the production line or guidelines on how to use their internal system. Furthermore, on the hard drive USB-HD-13, many videos, images, and documents were collected after the recovery process. After inspecting these documents, we note that there was much sensitive information, which is relevant to the previous owner (e.g., photos, tax invoices, bank transfers, and credit cards). From these data, it is suggested that the previous owner is a director of an accounting consultant company. Additionally, there is a lot of financial information which is related to the previous owner's company (e.g., bank account number, bank information, and tax code number). The number of files recovered during the recovery process is presented in Table 6.

5. Discussion

This section discusses the findings from the forensic disk imaging tool experiments and compares the selected tools' performance (functionalities, hardware usage, and time consumption) in detail. We also discuss the analysis of collected remnant data and present several solutions/guidelines for completely erasing the data to prevent security or privacy risks.

5.1. Findings of Evaluated Forensic Imaging Tools

This section discusses and compares the performance data collected from the experiments of the three selected forensic tools that are best suited for investigating remnant data present in second-hand USB storage devices.

Test Case FIT-ID-01: For this test case, we note that the DC3DD and DCFLDD tools successfully passed all the functional requirements, as presented in Table 5. However, the DCFLDD tool could not generate a log file after creating a forensic image or calculate the hash value of created image, even though these functions are mentioned in the tool manual. In addition, log file generated by DCFLDD provided more information (e.g., total time to create the image and verify, source device information, and hash values comparison) compared to DC3DD (presented in Appendix D (https://github.com/imdadullahunsw/forensics, accessed on 7 June 2022)). The forensic images created by these tools had the same hash values as the source. In addition, hash values in generated log files were the same as forensic images after being rechecked. Based on experimental data collected for CPU usage, we note that the DCFLDD used more CPU resources (31%) than DC3DD (5%) or DCFLDD (4.8%). In addition, only with hard drive 'USB-HD-05' was the percentage of CPU resources used by DCFLDD during the experiment higher compared to DC3DD. Furthermore, DCFLDD used more CPU resources in peak time (77% CPU) compared to DC3DD (76%) or DCFLDD (60%). The percentage of CPU resources used by DCFLDD did not change with different hard drives in the test. However, DCFLDD and DC3DD used more CPU resources in peak time (i.e., highest CPU usage) compared to downtime (lowest CPU usage). On average, DCFLDD used more CPU resources (around 48%) compared to DC3DD (around 41%) or DCFLDD (around 43%). Furthermore, we note that the amount of CPU usage from selected tools did not change too much with the hard drive of different storage capacities.

From the memory usage viewpoint, the DCFLDD used much higher memory resources (949 MB) than DC3DD (119 MB) or DCFLDD (4.57 MB). In peak time (i.e., highest memory usage), the amount of memory resources used by DCFLDD did not change too much (around 0.6 MB), and DCFLDD did not even change the memory usage, while DC3DD increased the amount of memory resources from 119 MB to 187.23 MB during peak time. On average, the DCFLDD tool used most of the memory resources compared to DC3DD and DCFLDD. In addition, the amount of memory used by DCFLDD was six times higher than DC3DD and 41 times higher than DCFLDD. Recall from Figure 3a, DCFLDD was the first tool to finish the process, the second tool was DC3DD, and the last one was DCFLDD. When overviewing the total amount of CPU usage, there is not a big difference in the use of CPU between DC3DD and DCFLDD. Even though sometimes the CPU resources used by DC3DD were even higher than DCFLDD, DCFLDD was still faster than DC3DD. However, the amount of memory resources usage affected the speed of the imaging process. The amount of memory usage used by DCFLDD was significantly higher compared to DC3DD or DCFLDD, and DC3DD also used more memory resources than DCFLDD (i.e., Figure 5a). Although when running each tool in several hard drives that have the same storage capacity, the time it takes for the tool to create forensic images still varies. This may happen because of the differences of USB external hard drives and how the tools were developed to collect the data.

In conclusion for this test, the DCFLDD and DC3DD met the tool functional requirements. However, the information provided by DCFLDD was more detailed, compared to DC3DD, and its friendly graphical user interface (GUI) was a significant advantage compared to other tools. In terms of hardware resources usage, DCFLDD was the tool that used most of the hardware resources compared to others, though it helped DCFLDD to finish the imaging process faster than DC3DD or DCFLDD. Overall, with all the advantages above, DCFLDD was the best tool in the test case.

Test Case FIT-ID-02: In this test case, all the selected forensic imaging tool's functionalities met the tool functional requirements, as presented in Table 5. These tools successfully created multiple forensic images and still ensured the integrity of the source device. The DCFLDD generated the hash values and recorded them into the log file; however, it failed to provide other important information (e.g., reading speed from hard drive, start-

ing time and ending time, and hard drive information). In addition, due to the creation of each forensic image being limited to only 2 GB, it may help the DCFLDD tool to save the log information and check the hash values for all created forensic images. On the other hand, DC3DD and DCFLDD generated more information in log files than DCFLDD , and DCFLDD provided more information about the source device than DC3DD. However, DC3DD and DCFLDD created detailed log files which contained the hash values of each created forensic image in the SHA-256 format. On the other hand, DCFLDD did not log the hash values for each created forensic image. DCFLDD only logged the overall SHA-256 values and compared it with the source device (Appendix D (https://github.com/imdadullahunsw/forensics, accessed on 7 June 2022)).

As depicted in Figure 4b, for low CPU usage calculation, the DCFLDD used most of the CPU resources (around 40%); however, the CPU usage from DC3DD and DCFLDD was inconsistent compared to DCFLDD. In peak time (i.e., highest CPU usage), there is a significant difference in how DC3DD and DCFLDD used CPU resources. DC3DD used 100% CPU resources with different hard drive storage capacities. DCFLDD used up to 95.53% CPU resources, whereas DCFLDD only used up to 62% CPU resources. On average (i.e., Figure 4b), DC3DD used most of the CPU resources (53% to 60%), whereas DCFLDD used the least (i.e., 34% to 56%). Furthermore, DCFLDD was the tool with a more consistent use of CPU resources than other tools (around 51%) in different USB hard drives. Similarly, as shown in Figure 5b, the DCFLDD used more memory resources (949.72 MB) compared to DC3DD (119.55 MB) and DCFLDD (4.57 MB). During the highest memory consumption calculation, DCFLDD utilised the majority of the memory resources compared to other tools, while this usage did not increase too much between the lowest memory used and highest memory used (only 1.25 MB extra in highest used). On the other hand, DC3DD used more memory resources during peak time compared to the lowest used, apart from DCFLDD, which did not change the memory resources even in the highest used and lowest used. On average (as shown in Figure 5b), DCFLDD used most of the memory resources (950 MB), followed by DC3DD (186 MB) and DCFLDD (4.57 MB).

Base on the experimental results, as shown in Figure 3b, it took DCFLDD more time to finish the imaging process compared to DCFLDD and DCFLDD. The amount of time to finish the process by DCFLDD was the most consistent when using different hard drives. Overall, DCFLDD was the fastest tool, and the slowest one was DCFLDD. As shown in Figure 4b, DC3DD used more CPU resources than DCFLDD or DCFLDD. However, the amount of memory used by DCFLDD (Figure 5b) on average, was significantly higher than DC3DD or DCFLDD, which explains why DCFLDD was the fastest tool in this test. We note that in this test case, all of the selected forensic tools met the expected tool functionalities. However, the information logged by DC3DD and DCFLDD was less detailed compared to DCFLDD (e.g., hard drive info, time log, and hash log comparison). In addition, the amount of CPU usage by DC3DD was higher than other tools on average, even though the amount of CPU resources used by DCFLDD and DC3DD was similar, the memory usage by DCFLDD was higher than DC3DD or DCFLDD, which affected the imaging process speed of DCFLDD and DC3DD. Overall, in this test case, DCFLDD was the best tool out of the selected tools.

Test Case FIT-ID-03: For various functionalities shown in Table 5, all of the selected tools successfully stopped the process when the output location met the limit of the storage capacity. However, none of these tools was successful in changing the output destination during the imaging process (Function 'DI-RO-07'); these tools only notified that the output destination reached the limit storage capacity and stopped the imaging process. In addition, only DCFLDD sent out a pop-up warning that the output destination was smaller than the forensic image before the starting of the imaging process (Appendix D (https://github.com/imdadullahunsw/forensics, accessed on 7 June 2022)). The DC3DD and DCFLDD started the imaging process without any notifications and only showed the error after the process stopped. All the selected tools logged the correct error for limited output destination storage capacity, except for DCFLDD's log files, which contained more information about the source device compared to DC3DD and DCFLDD. Furthermore, DCFLDD did mention whether

the output destination had reached the limit storage capacity in the log file. However, no further information (e.g., reading speed, starting time, and ending time) was logged.

The lowest CPU usage from DCFLDD was inconsistent among the small USB hard drive storage capacity (120 GB and 250 GB hard drive). The amount of CPU resources used by DC3DD was also inconsistent, except for the percentage of CPU usage from DCFLDD, which was consistent with any hard drive storage capacities. In the highest CPU usage calculations, the DCFLDD used more CPU resources (62%) than other tools most of the time, and DC3DD used 62% CPU resources for only one time. On average (as shown in Figure 4c), the DCFLDD utilised most of the CPU resources, whereas the DCFLDD used the least. Similarly, DC3DD used more CPU resources for hard drives with significant storage capacity (500 GB and 650 GB hard drives), while DCFLDD and DCFLDD fluctuated within the percentage of CPU usage for different USB storage capacity. Similarly, we note that the DCFLDD used only a small amount of memory resources (2.3 MB). In contrast, DCFLDD used up to 194 MB of memory resources, and DC3DD used 183 MB memory resources. In this test, each tool processed the imaging process in a short amount of time (Figure 3c), which explains why the number of memory resources used by these tools did not change too much, i.e., Figure 5c, except for DC3DD, which increased the memory usage from 119 MB to 250 MB between the downtime and peak time. On average, we note that the DCFLDD utilised most of the memory resources (950 MB), followed by DC3DD (186 MB) and DCFLDD (2.3 MB).

According to our evaluation of time consumption for various tools, as shown in Figure 3c, all of the selected tools did not stop the process at a consistent time. The amount of time before the task stop was inconsistent, even within the same tool (e.g., d3dd took 121 s in hard drive 'USB-HD-05' and took 116 seconds in hard drive 'USB-HD-09'). Overall, DCFLDD was the first tool which completed the imaging process (75 s) in the test with limited output location, followed by DC3DD (119 s) and DCFLDD (700 s). As shown in Figure 4c, DCFLDD used most of the CPU resources; however (in Figure 5c), DCFLDD used most of the memory resources. The reason why DCFLDD was faster than other tools is because of the amount of memory usage by DCFLDD was higher than DCFLDD or DC3DD. Furthermore, in this test case, apart from the fact that all of the tools did not meet all the expected results, at least DCFLDD notified the user about the error before the process had started. In addition, the information logged by DCFLDD had more details than other tools. This was a significant advantage for DCFLDD when comparing the functionalities of the selected tools in this test case. Overall, DCFLDD was the best tool in this test case because of the detailed log information, friendly GUI, and the necessary time to finish the task.

Test Case FIT-ID-04: In this test case, based on Table 5, DCFLDD did not meet the functional requirements. The tool evaluation results from this test are the same as the test case 'FIT-ID-01'; the DCFLDD was not able to check the created forensic image and log any information into the log file (i.e., for tool function 'DI-RO-05' and 'DI-RO-17'). DCFLDD and DCFLDD generated the log and calculated the hash values after the imaging process as expected, aside from DC3DD, which did not log any information and verify the source device integrity. Additionally, the log information generated by DCFLDD contained more information compared to DC3DD (e.g., source device information, average imaging speed, and total time spent).

For lower CPU usage calculation, we note that the DCFLDD (30%) and DCFLDD (31%) used more CPU resources than DCFLDD (15%). Similarly, the amount of CPU resources used by DCFLDD did not change when using different hard drive storage capacities. In peak time, DCFLDD was the tool which used most of the CPU resources (84%) compared to DC3DD (76%) and DCFLDD (61%). Even though DCFLDD used more CPU resources in peak time than DC3DD, on average (Figure 4d), DC3DD used more CPU resources (52%) than DCFLDD (49.83%), followed by DCFLDD (46%). In addition, for lower memory usage, the DCFLDD used only a small amount of memory resources (2.33 MB) compared to DCFLDD (183.72) and DCFLDD (727 MB). The amount of memory resources used by DC3DD and DCFLDD did not change in peak time, except for DCFLDD , which used more memory resources (from 727 MB to

793 MB). Likewise, DCFLDD , DCFLDD and DC3DD used the same amount of memory resources in different hard drive storage capacities. On average (Figure 5d), DCFLDD was the tool used most of the memory resources (760 MB), followed by DC3DD (183 MB) and DCFLDD (2.33 MB).

During our experimentation, we note that DC3DD first finished the tasks, followed by DCFLDD and DC3DD (Figure 3d). We note that among all the previous experimentations with the three test cases, DCFLDD was always the fastest tool. However, the result from Figure 3d is different from other test cases, where the performance of DC3DD is better than DCFLDD . The only difference between this test case and test case 'FIT-ID-01' is that the source devices were connected directly to the forensic station without a write blocker. The DC3DD used more CPU resources than DCFLDD on average. In addition, DCFLDD used more memory resources than DC3DD. However, in the test cases 'FIT-ID-01' and 'FIT-ID-02', DC3DD used more CPU and memory resources than in the test cases ('FIT-ID-04'). Furthermore, with the write blocker (test cases 'FIT-ID-01' and 'FIT-ID-02') DCFLDD used less CPU and memory resources than in test case 'FIT-ID-04', which explains why in this test case, DCFLDD finished the task slower than DC3DD. Likewise, in the test case 'FIT-ID-04', DCFLDD did not meet the functional requirements, and the generated log file by DC3DD had less information than the generated log file by DCFLDD. However, DC3DD was the fastest tool in this test and used less memory than DCFLDD. Nevertheless, DC3DD used more CPU resources than any other tools in this test case. Overall, even though DC3DD was faster than DCFLDD for a few minutes, in terms of functionaries and user friendliness, DCFLDD was better than DC3DD.

5.2. Discussion over the Tool Performance

According to the above investigations, the tool performance results in the aspect of time consumption and hardware resource usage were impacted by the test cases and source devices; however, the tool functionalities were not affected. The collected data indicated that the hardware resource usage and time consumption of each forensic tool were inconsistent between different test cases. The test results from test cases 'FIT-ID-01' and 'FIT-ID-04' showed how the forensic tool performance changed with and without the use of a write blocker, such as the tools used less time without a write blocker (Figure 3a,d), DC3DD used more CPU resources without a write blocker and even finished the imaging process faster than DCFLDD. On average, among the four test cases, DCFLDD used the most memory resources compared to other tools, whereas DC3DD utilised the most CPU resources. Even though DCFLDD used more memory than DC3DD or DCFLDD and always used more than 60% of CPU, we argue that with the hardware specifications of most of computers nowadays, this level of hardware usage can be accepted.

Based on our literature review, other researchers had evaluated different forensic imaging tools based on the tool functionalities; however, no one had compared the hardware resource usage and time consumption. This research contributed by comparing the functionalities of popular open-source forensic tools based on the NIST framework and also evaluated in detail the hardware resource usage and time consumption for each tool. Following the experiments, the best tool among the selected tools is DCFLDD. This tool meets most of the requirements in different test cases, except for the functions 'DI-RO-07' which is featured in test case 'FIT-ID-01'. In addition, Guyamger outperforms other tools in forensic image creation speed and hash values calculation speed, and all log files include more information (e.g., source hard drive information, and time consumption for creating the forensic image).

5.3. Findings of Collected Data from the Second-Hand USB Hard Drives

In this research, 17 second-hand USB external hard drives were used to experiment with the availability and analysis of any remnant data in these hard drives. The output of the experiment revealed that 15 hard drives (88.23%) out of 17 hard drives had data after the recovering process, and only two hard drives had removed all the remnant data and were properly formatted. For detailed information from the experiments with 17 hard drives

(presented in Appendix E (https://github.com/imdadullahunsw/forensics, accessed on 7 June 2022)), the summary results are listed as follows:

- None of the USB external hard drives (0%) were damaged or unable to create the forensic image.
- Two hard drives (i.e., 11.76% namely 'USB-HD-08 and USB-HD-11') were formatted by using forensic methods, and no data were found from these hard drives.
- Seventeen of the purchased USB hard drives (100%) had traces of remnant data.
- Data from 15 hard drives (88.23%) were able to be recovered. Ten of these hard drives (58%) contained personal information which could help to identify the previous owners, and six hard drives (35.29%) contained the information that identified the organisations that used to own these hard drives. In addition, three hard drives (17.65%) contained viruses and malware.

Figure 6 illustrates the type of data and information collected from the purchased hard drives.

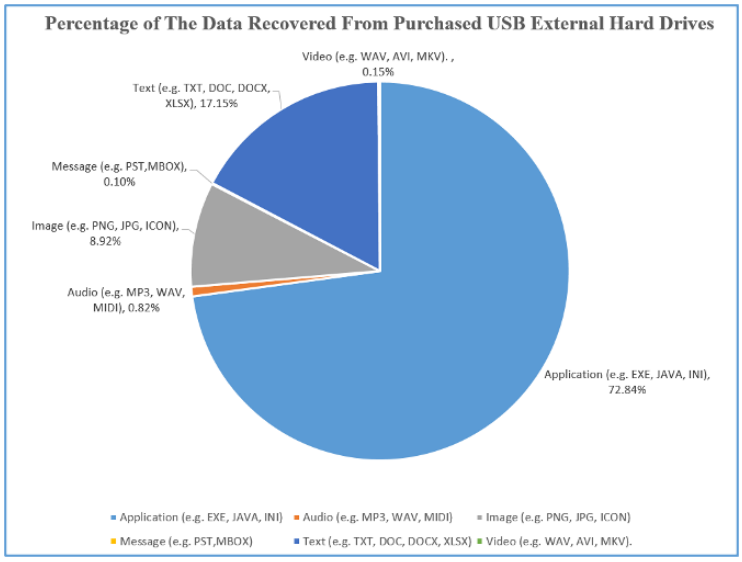

Figure 6. Proportion of data type identified.

Based on our findings, we note that most of the purchased hard drives still contained the remnant data (e.g., bank transactions, driver licenses, and contracts). This is a high-security risk due to the amount of information related to a person's identity and the organisation's confidentiality. Only data from two hard drives out of 17 hard drives were securely deleted, which means the users in New Zealand, along with other users mentioned in previous research, are not yet aware of the security risks and threats. On the other hand, data from two hard drives were securely deleted, which means nearly 11% of the users i.e., 2 out of 17, understand the risks, and they also know the method to format the hard drive or delete data securely. The reasons for the vast majority of sensitive data being left on the storage devices are due to the lack of security awareness, organisation policies, disposal methods of second-hand storage devices, and users being unaware of the existence or the method of properly using secure erased tools [17,47]. Therefore, there is a need to raise awareness and educate individuals and businesses about data security issues when using storage devices. These devices must be appropriately managed and processed following appropriate methods before being sold or stored.

5.4. Secure Data Deletion Recommendation

Users could physically destroy entire storage devices or send the command to the storage device controller to overwrite every block on the device. Each storage device has a controller which can translate the analog format into a data format. The researchers

mentioned that to securely delete data through a physical approach, users should request the controller to overwrite a single block or overwrite all blocks on the storage devices with new random values. This process ensures that data on the storage device are irrecoverable [40,47,48]. The researchers also mentioned that users could use different software to delete data securely: first, users could command the device's drivers and low-layer interfaces to apply secure delete by using free Secure Erase Utility (https://cmrr.ucsd.edu/resources/secure-erase.html, accessed on 7 June 2022) software or Linux's MMC driver (https://www.kernel.org/doc/html/latest/driver-api/mmc/index.html, accessed on 7 June 2022) [48]. Second, users could use file overwriting tools to overwrite files or delete all old data in a selected folder and load new unimportant data within the same folder [32,49]. Third, users could use free-space filling tools to fill in all the free spaces in a storage device and ensure the device does not contain sensitive data rather than filler material [32].

Many researchers had studied to understand the challenges and difficulties of securely erasing data on storage devices. The researchers mentioned that data deletion methods could be broken down into two categories: Low-level formatting and high-level formatting. The researchers mentioned that the deleted data from low-level formatting methods could be quickly recovered. On the other hand, it is challenging to retrieve data from high-level formatting methods [40,47,50]. However, to be able to apply high-level formatting methods requires more time and effort, compared to usual methods [51].

5.5. Our Research Limitations

Many open-source forensic imaging tools exist on the market; however, because of the time limitation, this study only focuses on three popular open-source tools, and there were no commercial tools evaluated in this research. Additionally, the selected tools tested in the experiment could not test several functions (as in Appendix A), due to the difficulty of finding suitable hard drives. Owning to the limited budget and time, there were not many forensic workstations with different specifications and configurations, and the test results were based on one specific forensic workstation. To replicate the experiment with the same test result, other researchers need to set up the same forensic workstation (hardware and software specification) and use the same second-hand USB external hard drive model. For this reason, the test cases for tool functionalities evolution were designed to ensure that the collected result was independent of the hardware specification to improve the accuracy or reliability of the test result. However, the tool test results for hardware usage and time consumption also depend on the USB external hard drive's specification.

For this reason, although we used only a USB 2.0 interface to test the tool's image creation speed, many types of connection interfaces need to be tested (e.g., SATA, M2, and Thunderbolt) to speed up the process of creating forensic images of each tool. It is also possible to spend more time searching for external drives that meet specific requirements to test the remaining functions in the NIST list that have not been tested in the experiments. In addition, in this research, only three open-source tools were used to collect and process data, and no commercial tools were used due to the limited budget. In addition, only 17 USB external hard drives with different storage capacities were used in this research, and the findings may not reflect the actual situation or represent only a very small fraction of the actual market of hard drives usage in New Zealand. To broaden the study, future experiments may need to be performed on the data recovery on more USB external hard drives (e.g., from 100 to 200 hard drives) and must experiment with the different hard drives of higher capacities, such as from 100 GB to 1 TB, 2 TB or even larger.

6. Related Work

We note that there are limited studies where the researchers [12–14,44] evaluated the functionalities of imaging tools based on the NIST framework. NIST created different test environments to evaluate various versions of forensic tools, e.g., `X-Ways Forensics 16.2 SR 5` [52], `Image MASSter Solo 4 Forensic` [53], `Fast Dist Acquisition System`

(FDAS) [54], DCFLDD 1.3.4 1 [20], DC3DD v7.2.641 [19], Guymager v0.8.1 [21], FTK Imager v3.4.2.6 [55], WiebeTech Ditto Forensic FieldStation v2016Mar01a [56], and Tableau TD3 Forensic Imager v2.0.0 [57], for different storage media SATA, and ATA hard drives, USB thumb drives and different types of disk partitions (e.g. exFAT, ext2, ext4, f32). The objective of this initiative is to provide measurable metrics and to assist the practitioners, researchers, and other users with the tools used in computer forensics investigations with different scenarios and for providing accurate results.

The authors in [1] proposed a digital forensic workflow model for various tasks involved in the digital forensic investigation process that helps enable the identification and management of risks error mitigation during each stage of the workflow. Another work [2] presented a peer review methodology for the digital forensic investigation that is a six-stage approach consisting of investigative tasks, forensic activities, and forensic analysis processes. The authors in [44] evaluated and compared EnCase (version 6.8) and LinEn (version 6.1) based on NIST requirements functions, such as creating an image on one operating system. The authors found a difference between the Encase and Linen performance since the LinEn will stop creating an image process when it finds some reading errors or unacceptable partition on a source device because of the incompatibility on some Linux kernel. Cusack and Liang [13] evaluated FTK Imager (version 2.90), Helix3 Pro, and Automated Image and Restore (AIR) (version 2.0.0) based on their functions to create images and generate log types. However, they did not mention the time each tool needed in order to create and verify images. The authors found that compared to both FTK and Helix3 Pro, the AIR tool supported more features in creating images, such as the ability to verify images that were created in the network storage and the capability to create images where the export destination had inadequate storage space. Similarly, Shah and Paradise [14] evaluated EnCase (version 7.04.01), FTK Imager (version 3.1.1.8) and SANS SIFT Workstation (version 2.14) tools to observe how useful these tools were in creating and verifying images and how these tools used hardware resources. The authors in [12] evaluated FTK Imager (version 3.1.4), Forensic Toolkit (version 5.1), EnCase Imager (version 7.09), EnCase (version 7.05.01), Open Source Suite (version 1.3.2.20110401), and Paladin (version 4.0) tools based on the tool feature lists mentioned by NIST. They found that the FTK (version 5.1) tool supported more functions to create images. All the tested tools had the function to compress image size. Table 7 summarises the commercial, free and open-source tools used by the researchers in the previously published research articles.

Table 7. The digital forensic tools tested by previous works for various functionality.

Articles	Commercial Tools	Free Tools	Open-Source Tools
[44]	EnCase (version 6.8)	N/A	LinEn (version 6.1)
[13]	Helix3 Pro, Automated Image and Restore (AIR) (version 2.0.0)	FTK Imager (version 2.90)	N/A
[52]	X-Ways Forensics (version 16.2 SR-5)	N/A	
[53]	Image MASSter Solo-4 Forensic		
[54]	Fast Disk Acquisition System (FDAS) (version 2.0.2)		
[20]	N/A		DCFLDD (version 1.3.4-1)
[14]	EnCase (version 7.04.01)	FTK Imager (version 3.1.1.8), SANS SIFT Workstation (version 2.14)	N/A
[12]	Forensic Toolkit (version 5.1), EnCase (version 7.05.01), Paladin (version 4.0)	FTK Imager (version 3.1.4), EnCase Imager (version 7.09)	Open Source Suite (version 1.3.2.20110401)

Table 7. *Cont.*

Articles	Commercial Tools	Free Tools	Open-Source Tools
[19]	N/A	N/A	DC3DD (version 7.2.641)
[21]			Guymager (version 0.8.1)
[55]		FTK Imager (version 3.4.2.6)	N/A
[56]	WiebeTech Ditto Forensic FieldStation (version 2016Mar01a)	N/A	
[57]	Tableau TD3 Forensic Imager (version 2.0.0)		

Many researchers studied remnant data on storage devices purchased from the second-hand market, e.g., in the United Kingdom (UK), North America, Germany, France [22], the United Arab Emirates (UAE) [8], Indonesia [7], Australia [6] and New Zealand [18]. The authors in [7,8] used devices provided by the supplier which were marked with a unique serial number, ensuring that the researchers could not determine how or where the devices were purchased. Other researchers purchased second-hand storage devices from online auction websites within the same country. The authors in [6,36,58] used different accounts to buy the memory cards and USB thumb drives from different sellers to prevent anybody from detecting an ongoing investigation and also to avoid unintended effects if someone wanted to change or influence the results of the research. After collecting storage devices, the researchers created the images of all devices by using a variety of open-source and commercial forensic imaging tools. The acquisition tools used in their studies included `FTK Imager` (version 1.42 to 3.4.1) ([7,8,36]), `Encase imager` (version 7.10) ([34]), `X-ways` (version 18.8) ([34]) and `Tableau TD1` ([18]). The images were then securely stored to ensure confidentiality.

Few works have studied searching for remaining data in second-hand storage devices and understanding the process and tools used by the previous researchers in their experiments. For example, the authors in [15] examined the second-hand 43 old USB thumb drives, purchased over the eBay website in England, for remaining data and the level of security risks based on the examined data. The authors used different types of forensic tools, such as Forensic Tool Kit (FTK), to create images of the USB devices, and `Autopsy` (version 2.08) and `Helix` software (Linux based) were used to analyse the created images. The authors in [22] analysed different hard drivers (e.g., 174 hard drives from the UK, 74 hard drives from North America, 39 hard drives from Germany, 17 hard drives from France, and 42 hard drives from Australia) by using an authenticated tool such as `Autopsy` (version 2.2.4) and the `Sleuth Kit` (version 3.1.3). Similarly, other works [6–8,16,18,36] use evaluate different types of secondary storage devices by first creating their images, and analysed them for sensitive information using various forensic tools.

Hence, we note that very limited works [12–14] carried out an in-depth evaluation of free, commercial, and open-source imaging forensic tools. In addition, the majority of the works did not fully evaluate the functional and optional requirements of imaging tools based on the NIST CFTT framework (see Appendix A). For example, Shah and Paradise [14] tested four functions such as 'DI-RM-01', 'DI-RM-02', 'DI-RM-04' and 'DI-RM-08' (as shown in Appendix A). However, Cusack and Liang [13] tested most of the functional requirements listed by the CFTT framework, except the optional requirement functions. Furthermore, we note that specifically in New Zealand, since our experiments were based on the secondary storage devices purchased from New Zealand, the research on data remanence in second-hand hard disks was conducted by Roberts and Wolfe [18] in 2011. However, their research did not focus on USB devices and was conducted 10 years ago. Other researchers also conducted research on data remanence in second-hand USB devices [6,7,15,36]. However, their research only used USB thumb drives to collect the remnant data. Therefore, the proposed research fills in the research gap by testing all functional requirements as recommended by NIST in order to identify the best open-source

forensic imaging tool currently available. In addition, we tested the majority of the NIST optional requirements; Table 8 summarises the NIST function requirements that were tested in our and previously published research articles. Another significant contribution of this study is that it discovered what sort of data are available on second-hand USB external hard drives sold online in New Zealand.

Table 8. NIST functional and optional requirements tested in previous works.

NIST Tools	**Articles**													
Req. ID	**[20]**	**[53]**	**[52]**	**[54]**	**[19]**	**[21]**	**[55]**	**[56]**	**[57]**	**[13]**	**[12]**	**[14]**	**[44]**	**Ours**
DI-RM-01	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
DI-RM-02	✓	✓	✓	✗	✗	✗	✓	✗	✗	✓	✓	✓	✓	✓
DI-RM-03	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
DI-RM-04	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
DI-RM-05	✓	✗	✓	✗	✗	✗	✗	✗	✗	✓	✗	✗	✓	✓
DI-RM-06	✓	✓	✗	✓	✓	✓	✓	✗	✗	✓	✓	✗	✓	✓
DI-RM-07	✓	✓	✓	✓	✓	✓	✓	✗	✗	✓	✗	✗	✗	✓
DI-RM-08	✗	✗	✗	✗	✗	✗	✓	✗	✗	✓	✗	✓	✗	✓
DI-RO-01	✗	✓	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓
DI-RO-02	✓	✗	✗	✗	✗	✗	✗	✗	✓	✗	✗	✗	✗	✓
DI-RO-03	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓
DI-RO-04	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓
DI-RO-05	✗	✗	✗	✗	✓	✓	✗	✓	✓	✗	✗	✗	✗	✓
DI-RO-06	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
DI-RO-07	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓
DI-RO-08	✗	✗	✗	✗	✗	✗	✓	✗	✗	✗	✗	✗	✗	✗
DI-RO-09	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
DI-RO-10	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
DI-RO-11	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
DI-RO-12	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
DI-RO-13	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
DI-RO-14	✗	✗	✗	✓	✗	✗	✗	✗	✓	✗	✗	✗	✗	✗
DI-RO-15	✗	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
DI-RO-16	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
DI-RO-17	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓
DI-RO-18	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓

7. Conclusions

We presented a research gap for studying forensic imaging tools for forensic investigations and the necessity to carry out analyses on second-hand storage devices in the local market in New Zealand. We described the experimental quantitative research methodology, research model, and the processes to evaluate the forensic tools (mainly the `DC3DD`, `DCFLDD`, and `Guymager`) for NIST functional and optional requirements and to collect the remnant data from purchased hard drives. In our findings, we noted that `DC3DD` and `Guymager` met most of the tool functionalities in different test cases. However, there is only one function that these tools failed ('DI-RO-07'), apart from that, all other functionalities worked as expected. The information saved in the log files generated by `Guymager` include more data compared to `DCFLDD` and `DC3DD`. In addition, `DC3DD` and `DCFLDD` did not have a user interface, and all requests must be inputted via the command line except for `Guymager`, which has a user-friendly GUI. Overall, `Guymager` was the best tool in terms of tool functionalities and tool performance (less time consumption compared to other tools). After recovering

data from purchased USB external hard drives, we found the vast majority of remnant data and information (e.g., bank transactions, contracts, CVs, photos, and videos), which could help to determine the identity of the previous owner. We noted that roughly 90% of the hard drives contained personal/organisational sensitive data, specifically 6 out of 17 hard drives contained organisation information, and 10 out of 17 hard drives contained personal information. Data collected from hard drives indicate that users in New Zealand are not fully aware of data security.

Author Contributions: Conceptualisation: Z.S., A.K. and A.L.; Methodology: H.P.T. and Z.S.; Investigation: Z.S., H.P.T. and A.L.; Resources: Z.S. and A.K.; Writing—original draft preparation: Z.S., I.U., H.P.T. and A.L.; Writing—review and editing: Z.S., I.U. and A.K.; Supervision: Z.S. and A.K. All authors have read and agreed to the published version of the manuscript.

Funding: There are no internal or external funds available for this research work. We further declare that the corresponding author will fully pay the APC charges.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: This submission does not include human or animal research.

Data Availability Statement: There are no data associated with this article.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A. NIST 2005 Functional and Optional Requirements

Appendix A.1. Primary Features of Digital Evidence

The following are the primary requirements following the NIST (2005) recommendations (DI-RM stands for digital imaging requirement mandatory):

- **DI-RM-01**: The tool will be able to identify or recognise a physical storage device.
- **DI-RM-02**: The tool will be able to create a forensic image or clone from the source device or forensic image. The tools will enable the user to select outputs such as creating a clone or forensic image from a source device.
- **DI-RM-03**: The tool will be able to run on at least one operating system and create the image or clone from the source device on that operating system.
- **DI-RM-04**: The tool will be able to procure all observable sectors from the source device.
- **DI-RM-05**: The tool will be able to procure all concealed sectors from the source device.
- **DI-RM-06**: The tool will be able to procure all sectors or data from the source device accurately.
- **DI-RM-07**: The tool will be able to send out notifications to the user of error locations and types if any errors occur when reading the source device.
- **DI-RM-08**: The tool will be able to fill in a specific object in the inaccessible data if any errors occur when reading the source device.

Appendix A.2. Optional Features of Digital Evidence

The following are the optional requirements following NIST (2005) recommendations (DI-RO stands for digital imaging requirement optional):

- **DI-RO-01**: If the tool supports image file creation and this image is selected, the tool will be able to create the image in the selected file format which includes all data acquired from the source device.
- **DI-RO-02**: If the tool supports an image file creation and is selected and if there is an error when creating an image file, the tool will inform the user of the error.
- **DI-RO-03**: If the tool supports creating an image file and allows for image selection, if the output location has insufficient storage space for the image, the tool will inform this error to the user.
- **DI-RO-04**: If the tool supports the creation of an image file and allows for image selection and allows dividing the image into multiple files with selected size, then

the tool will be able to create a multi-file image of the same file size and acquire the correct data from the source device.

- **DI-RO-05**: If the tool supports creating an image file and allows to select an image file to create, and additionally, if the tool allows checking the integrity of the created image and the option for integrity check after creating the image was selected, then the tool will be able to announce to the user if there is any change on the image file and mention the changed location on the image file.
- **DI-RO-06**: If the tool allows converting an image file to another image file format, then the tool will be able to convert an original image file to a new image file format and ensure the data in the new image file are the same as the original image file.
- **DI-RO-07**: If the tool allows to switch the output location and if the output location has excess storage capacity when creating an image file, then the tool will be able to allow the user to change to another output location and continue the image creation process on the new output location such that the resulting multi-file image denotes the same data that the tool obtains.
- **DI-RO-08**: If the tool provides replication during the acquisition process and the replicate function was selected, then the tool will create a replication from the source device.
- **DI-RO-09**: If the tool provides replication from an image file and then the replicate function was selected, then the tool will create a replication from the image file.
- **DI-RO-10**: If the tool provides partial replication that is a part of the source device data and the replicate function was selected, then the tool will be able to create a clone with a specific subset of the source device.
- **DI-RO-11**: If the tool allows to create a bit-for-bit clone (unaligned clone) and if the option was selected, then the tool will be able to collect every bit (except the pun) from the source device and create a bit-for-bit clone.
- **DI-RO-12**: If the tool allows to create a bit-stream duplicate clone (cylinder clone) and if the option was selected, then the tool will be able to create a cylinder clone.
- **DI-RO-13**: If the tool provides replication from an image file and if the replicate function was selected and there are surplus sectors on the clone output location, then the tool will be operated as a default setting or operated based on the user request; it will either not modify the surplus sectors or write a benign fill to the surplus sectors following the user request.
- **DI-RO-14**: If the tool provides replication from an image file and the replicate function was selected, and if there is not enough storage space on the destination location to store all the sectors collected from the source device, then the tool will announce to the user and create a summarised clone by using all available sectors on the clone destination location.
- **DI-RO-15**: If the tool provides replication from an image file and the replicate function was selected and there is a fault when creating the clone, then the tool will be able to announce to the user that there is a write error.
- **DI-RO-16**: If the tool provides a logging hash for each block and the function was selected, then the tool will be able to log the correct hash value for the block size which is required from digital sources.
- **DI-RO-17**: If the tool allows creating a log file, then the tool will be able to log at least one of the following information, such as tool version, the setting of the tool, imaging date, imaging time, and source device size.
- **DI-RO-18**: If the tool allows imaging from a source that is insecure by a write blocker hardware-based or software-based, then the tool will not change the source data throughout the imaging process.

References

1. Horsman, G.; Sunde, N. Unboxing the digital forensic investigation process. *Sci. Justice* **2022**, *62*, 171–180 . [CrossRef]
2. Sunde, N.; Horsman, G. Part 2: The Phase-oriented Advice and Review Structure (PARS) for digital forensic investigations. *Forensic Sci. Int. Digit. Investig.* **2021**, *36*, 301074. [CrossRef]

3. Talib, M.A. Testing closed source software: Computer forensic tool case study. *J. Comput. Virol. Hacking Tech.* **2018**, *14*, 167–179. [CrossRef]
4. Horsman, G. The different types of reports produced in digital forensic investigations. *Sci. Justice* **2021**, *61*, 627–634. [CrossRef]
5. Cert, U. Computer Forensics. 2008. Available online: https://www.us-cert.gov/sites/default/files/publications/forensics.pdf (accessed on 7 June 2022).
6. Robins, N.; Williams, P.A.; Sansurooah, K. An investigation into remnant data on USB storage devices sold in Australia creating alarming concerns. *Int. J. Comput. Appl.* **2017**, *39*, 79–90. [CrossRef]
7. Lim, C.; Meily, N.; Ahmadi, H. Forensics Analysis of USB Flash Drives in Educational Environment. In Proceedings of the International Conference on Information, Communication Technology and System (ICTS), Surabaya, Indonesia, 24 September 2014.
8. Jones, A.; Martin, T.; Alzaabi, M. The 2012 analysis of information remaining on computer hard disks offered for sale on the second hand market in the UAE. In Proceedings of the 10th Australian Digital Forensics Conference, ADF 2012, Perth, Australia, 3–5 December 2012; pp. 47–53.
9. Jones, A.; Martin, T.; Alzaabi, M. The 2016 Analysis Of Information Remaining On Computer Hard Disks Offered For Sale On The Second Hand Market In the UAE. *J. Digit. Forensics, Secur. Law* **2016**, *11*, 6.
10. U.S. Department of Commerce Technology Administration National Institute of Standards and Technology. The Nist Visiting Committee on Advanced Technology. Gaithersburg, MD 20899-1060. 2004. Available online: https://www.nist.gov/system/files/documents/2017/05/09/report04.pdf (accessed on 7 June 2022).
11. U.S. Department of Commerce Technology Administration National Institute of Standards and Technology. The Nist Visiting Committee on Advanced Technology. Gaithersburg, MD 20899-1060. 2005. Available online: https://www.nist.gov/system/files/documents/2017/05/09/report05.pdf (accessed on 7 June 2022).
12. Sonnekus, M.H. A Comparison of Open Source and Proprietary Digital Forensic Software. Ph.D. Thesis, Rhodes University, Makhanda, South Africa, 2014.
13. Cusack, B.; Liang, J. Comparing the performance of three digital forensic tools. *J. Appl. Comput. Inf. Technol.* **2011**, *15*, A11.
14. Shah, M.; Paradise, D. *Tool Comparison*; Research Champlain College: Dublin, Ireland, 2013.
15. Jones, A.; Valli, C.; Dabibi, G. *The 2009 Analysis of Information Remaining on USB Storage Devices Offered for Sale on the Second Hand Market*; School of Computer and Information Science: Perth, Australia, 2009.
16. Sansurooah, K.; Szewczyk, P. A study of remnant data found on USB storage devices offered for sale on the Australian second hand market in 2011. In Proceedings of the 10th Australian Information Security Management Conference, Perth, Australia, 3–5 Decmber 2012; Citeseer: State College, PA, USA, 2012; Volume 82.
17. Robins, N.; Williams, P.A.; Sansurooah, K. I know what you did last summer... An Investigation into Remnant Data on USB Storage Devices Sold in Australia in 2015. In Proceedings of the Australasian Computer Science Week Multiconference, Canberra, Australia, 2–5 February 2016; pp. 1–8.
18. Roberts, D.; Wolfe, H. Data remanence in New Zealand: 2011. In Proceedings of the 9th Australian Digital Forensics Conference, Perth, Australia, 5–7 December 2011.
19. *DC3DD-v7.2.641. Test Results for Digital Data Acquisition Tool*; Homeland Security: Washington, DC, USA, 2016. Available online: https://www.dhs.gov/sites/default/files/publications/1490_508_Test%20Report_NIST_Disk%20Imaging_dc3dd%20v7.2.641_October_14_2016.pdf (accessed on 9 June 2022).
20. *DCFLDD-1.3.4-1. Test Results for Digital Data Acquisition Tool*; Homeland Security: Washington, DC, USA, 2013. Available online: https://www.dhs.gov/sites/default/files/publications/DCFLDD%201%203%204-1%20Test%20Report_updated.pdf (accessed on 7 June 2022).
21. *Guymager-v0.8.1. Test Results for Digital Data Acquisition Tool*; Homeland Security: Washington, DC, USA, 2016. Available online: https://www.dhs.gov/sites/default/files/publications/1492_508_Test%20Report_NIST_Disk%20Imaging_Guymager%20v0.8.1_October_14_2016.pdf (accessed on 7 June 2022).
22. Jones, A.; Valli, C.; Dardick, G.S.; Sutherland, I.; Dabibi, G.; Davies, G. The 2009 analysis of information remaining on disks offered for sale on the second hand market. *J. Digit. Forensics, Secur. Law* **2010**, *5*, 3. [CrossRef]
23. Adam, C. *Forensic Evidence in Court: Evaluation and Scientific Opinion*; John Wiley & Sons: Hoboken, NJ, USA, 2016.
24. Dimpe, P.M.; Kogeda, O.P. Impact of using unreliable digital forensic tools. In Proceedings of the World Congress on Engineering and Computer Science, San Francisco, CA, USA, 25–27 October 2017; Volume 1, pp. 118–225.
25. Sammons, J. *The Basics of Digital Forensics: The Primer for Getting Started in Digital Forensics*; Elsevier: Amsterdam, The Netherlands, 2012.
26. Horsman, G. "I couldn't find it your honour, it mustn't be there!"—Tool errors, tool limitations and user error in digital forensics. *Sci. Justice* **2018**, *58*, 433–440. [CrossRef] [PubMed]
27. Kuharev, J.; Navarro, P.; Distler, U.; Jahn, O.; Tenzer, S. In-depth evaluation of software tools for data-independent acquisition based label-free quantification. *Proteomics* **2015**, *15*, 3140–3151. [CrossRef] [PubMed]
28. Bhat, W.A.; AlZahrani, A.; Wani, M.A. Can computer forensic tools be trusted in digital investigations? *Sci. Justice* **2021**, *61*, 198–203. [CrossRef] [PubMed]
29. Casey, E. *Digital Evidence and Computer Crime: Forensic Science, Computers, and the Internet*; Academic Press: Cambridge, MA, USA, 2011.
30. Casey, E. *Digital Evidence and Computer Crime: Forensic Science, Computers and the Internet*; Academic Press: Cambridge, MA, USA, 2018.
31. Taylor, R.W.; Fritsch, E.J.; Liederbach, J. *Digital Crime and Digital Terrorism*; Prentice Hall Press: Hoboken, NJ, USA, 2014.

32. Shaw, A.; Browne, A. A practical and robust approach to coping with large volumes of data submitted for digital forensic examination. *Digit. Investig.* **2013**, *10*, 116–128. [CrossRef]
33. Flandrin, F.; Buchanan, W.J.; Macfarlane, R.; Ramsay, B.; Smales, A. Evaluating digital forensic tools (DFTs). In Proceedings of the 7th International Conference: Cybercrime Forensics Education & Training, Canterbury, UK, 10–11 July 2014; pp. 1–16.
34. Horsman, G. Framework for Reliable Experimental Design (FRED): A research framework to ensure the dependable interpretation of digital data for digital forensics. *Comput. Secur.* **2018**, *73*, 294–306. [CrossRef]
35. Lee, J.; Un, S.; Hong, D. High-speed search using Tarari content processor in digital forensics. *Digit. Investig.* **2008**, *5*, S91–S95. [CrossRef]
36. Szewczyk, P.; Sansurooah, K. The 2012 investigation into remnant data on second hand memory cards sold in Australia. In Proceedings of the 10th Australian Digital Forensics Conference, Perth, Australia, 3–5 December 2012.
37. Hasan, R.; Mahmood, S.; Raghav, A. Overview on Computer Forensics tools. In Proceedings of the 2012 UKACC International Conference on Control, Cardiff, UK, 3–5 September 2012; IEEE: Piscataway, NJ, USA, 2012; pp. 400–403.
38. Kessler, G.C.; Carlton, G.H. A study of forensic imaging in the absence of write-blockers. *J. Digit. Forensics, Secur. Law* **2014**, *9*, 51. [CrossRef]
39. Talib, M.A. Towards early software reliability prediction for computer forensic tools (case study). *SpringerPlus* **2016**, *5*, 1–12. [CrossRef]
40. Zareen, M.S.; Aslam, B.; Akhlaq, M. Criteria for validating secure wiping tools. In Proceedings of the IFIP International Conference on Digital Forensics, Orlando, FL, USA, 26–28 January 2015; Springer: Berlin/Heidelberg, Germany, 2015; pp. 321–339.
41. Albanna, F.; Riadi, I. Forensic Analysis of Frozen Hard Drive Using Static Forensics Method. *Int. J. Comput. Sci. Inf. Secur.* **2017**, *15*, 173–178.
42. Chassanoff, A.; Woods, K.; Lee, C.A. Digital preservation metadata practice for disk image access. In *Digital Preservation Metadata for Practitioners*; Springer: Berlin/Heidelberg, Germany, 2016; pp. 99–109.
43. Guo, Y.; Slay, J. A function oriented methodology to validate and verify forensic copy function of digital forensic tools. In Proceedings of the 2010 International Conference on Availability, Reliability and Security, Krakow, Poland, 15–18 February 2010; IEEE: Piscataway, NJ, USA, 2010; pp. 665–670.
44. Byers, D.; Shahmehri, N. A systematic evaluation of disk imaging in EnCase® 6.8 and LinEn 6.1. *Digit. Investig.* **2009**, *6*, 61–70. [CrossRef]
45. Marra, M.; Di Biccari, C.; Lazoi, M.; Corallo, A. A gap analysis methodology for product lifecycle management assessment. *IEEE Trans. Eng. Manag.* **2017**, *65*, 155–167. [CrossRef]
46. Zhu, J. *Quantitative Models for Performance Evaluation and Benchmarking: Data Envelopment Analysis with Spreadsheets*; Springer: Berlin/Heidelberg, Germany, 2014; Volume 213.
47. Reardon, J.; Basin, D.; Capkun, S. On secure data deletion. *IEEE Secur. Priv.* **2014**, *12*, 37–44. [CrossRef]
48. Reardon, J. Related work on secure deletion. In *Secure Data Deletion*; Springer: Berlin/Heidelberg, Germany, 2016; pp. 11–31.
49. Diesburg, S.; Feldhaus, C.A.; Fardan, M.A.; Schlicht, J.; Ploof, N. Is Your Data Gone? Comparing Perceived Effectiveness of Thumb Drive Deletion Methods to Actual Effectiveness. *arXiv* **2015**, arXiv:1512.08986.
50. Dhillon, A. An Algorithm for Secure Formatting of Memory. *Int. J. Comput. Distrib. Syst.* **2012**, *1*, 66–71.
51. Onarlioglu, K.; Robertson, W.; Kirda, E. Eraser: Your Data Won't Be Back. In Proceedings of the 2018 IEEE European Symposium on Security and Privacy (EuroS&P), London, UK, 24–26 April 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 153–166.
52. *X-Ways Forensics 16.2 SR-5. Test Results for Digital Data Acquisition Tool*; Homeland Security: Washington, DC, USA, 2013. Available online: https://www.dhs.gov/sites/default/files/publications/X-Ways%20Forensics%2016%202%20SR-5%20TestReport_0.pdf (accessed on 7 June 2022).
53. *Image-MASSter-Solo-4-Forensic. Test Results for Digital Data Acquisition Tool*; Homeland Security: Washington, DC, USA, 2013. Available online: https://www.dhs.gov/sites/default/files/publications/Image%20MASSter%20Solo-4%20ForensicTestReport_1_Final_0.pdf (accessed on 7 June 2022).
54. *Fast-Dist-Acquisition-System-(FDAS). Test Results for Digital Data Acquisition Tool*; Homeland Security: Washington, DC, USA, 2013. Available online: hhttps://www.dhs.gov/sites/default/files/publications/508_Test%20Report_FDAS%202%200%202_October%202015_Final_0.pdf (accessed on 7 June 2022).
55. *FTK-Imager-v3.4.2.6. Test Results for Digital Data Acquisition Tool*; Homeland Security: Washington, DC, USA, 2016. Available online: https://www.dhs.gov/sites/default/files/publications/1491_508_Test%20Report_NIST_Disk%20Imaging_FTK%20Imager%20v3.4.2.6_October_14_2016.pdf (accessed on 7 June 2022).
56. *WiebeTech-Ditto-Forensic-FieldStation-v2016Mar01a. Test Results for Digital Data Acquisition Tool*; Homeland Security: Washington, DC, USA, 2016. Available online: https://www.dhs.gov/sites/default/files/publications/1498_508_Test%20Report_NIST_Disk%20Imaging_WiebeTech%20Ditto%20Forensic%20FieldStation%20v2016Mar01a_October_14_2016.pdf (accessed on 7 June 2022).
57. *Tableau-TD3-Forensic-Imager-v2.0.0. Test Results for Digital Data Acquisition Tool*; Homeland Security: Washington, DC, USA, 2018. Available online: https://www.dhs.gov/sites/default/files/publications/508_Test%20Report_NIST%20Disk%20Imaging%20Tool%20Tableau%20TD3%20Forensic%20Imager%20v2.0.0%20August%202018_Final.pdf (accessed on 7 June 2022).
58. Sansurooah, K.; Hope, H.; Almutairi, H.; Alnazawi, F.; Jiang, Y. An investigation into the efficiency of forensic data erasure tools for removable Usb flash memory storage devices. In Proceedings of the 11th Australian Digital Forensics Conference, Perth, Australia, 2–4 December 2013.