THE JOURNAL OF
DIGITAL FORENSICS,
SECURITY AND LAW

Journal of Digital Forensics, Security and Law

Volume 14 | Number 2 Article 3

6-30-2019

DF 2.0: An Automated, Privacy Preserving, and Efficient Digital Forensic Framework That Leverages Machine Learning for Evidence Prediction and Privacy Evaluation

Robin Verma
Indraprastha Institute of Information Technology Delhi, robinv@iiitd.ac.in

Jayaprakash Govindaraj Dr
Jayaprakash Govindaraj, jayaprakash_govindaraj@mcafee.com

Saheb Chhabra
Indraprastha Institute of Information Technology Delhi, New Delhi, India, sahebc@iiitd.ac.in

Gaurav Gupta
Ministry of Electronics and Information Technology, New Delhi, India, gupta.gaurav@meity.gov.in

Recommended Citation

Verma, Robin; Govindaraj, Jayaprakash Dr; Chhabra, Saheb; and Gupta, Gaurav (2019) "DF 2.0: An Automated, Privacy Preserving, and Efficient Digital Forensic Framework That Leverages Machine Learning for Evidence Prediction and Privacy Evaluation," *Journal of Digital Forensics, Security and Law*: Vol. 14 : No. 2 , Article 3.
Available at: https://commons.erau.edu/jdfsl/vol14/iss2/3

DF 2.0: AN AUTOMATED, PRIVACY PRESERVING, AND EFFICIENT DIGITAL FORENSIC FRAMEWORK THAT LEVERAGES MACHINE LEARNING FOR EVIDENCE PREDICTION AND PRIVACY EVALUATION

Robin Verma[1], Jayaprakash Govindaraj[2], Saheb Chabra[3], Gaurav Gupta[4]

[1]University of Texas at San Antonio, Texas, USA
[2]McAfee Software India Private Limited, Bangaluru, India
[3]Indraprastha Institute of Information Technology Delhi, New Delhi, India
[4]Ministry of Electronics and Information Technology, New Delhi, India
robin.verma@utsa.edu[1]; *jayaprakash_govindaraj@mcafee.com*[2]; *sahebc@iiitd.ac.in*[3]; *gupta.gaurav@meity.gov.in*[4]

ABSTRACT

The current state of digital forensic investigation is continuously challenged by the rapid technological changes, the increase in the use of digital devices (both the heterogeneity and the count), and the sheer volume of data that these devices could contain. Although data privacy protection is not a performance measure, however, preventing privacy violations during the digital forensic investigation, is also a big challenge. With a perception that the completeness of investigation and the data privacy preservation are incompatible with each other, the researchers have provided solutions to address the above-stated challenges that either focus on the effectiveness of the investigation process or the data privacy preservation. However, a comprehensive approach that preserves data privacy without affecting the capabilities of the investigator or the overall efficiency of the investigation process is still an open problem. In the current work, the authors have proposed a digital forensic framework that uses case information, case profile data and expert knowledge for automation of the digital forensic analysis process; utilizes machine learning for finding most relevant pieces of evidence; and maintains data privacy of non-evidential private files. All these operations are coordinated in a way that the overall efficiency of the digital forensic investigation process increases while the integrity and admissibility of the evidence remain intact. The framework improves validation which boosts transparency in the investigation process. The framework also achieves a higher level of accountability by securely logging the investigation steps. As the proposed solution introduces notable enhancements to the current investigative practices more like the next version of Digital Forensics, the authors have named the framework 'Digital Forensics 2.0', or 'DF 2.0' in short.

Keywords: Digital Forensics Framework, Automation, Data Privacy, Machine Learning

1. INTRODUCTION

Digital forensic science has evolved a lot since the first Digital Forensics Research Workshop (Palmer et al., 2001). However, there have been some research problems that are continuously challenging the researchers and practitioners till date.

The first and foremost challenge is the ever growing data storage capacity of digital devices (Quick & Choo, 2014). The large volume of data increases the time requirements for the data acquisition and the data analysis processes (Lillis, Becker, O'Sullivan, & Scanlon, 2016). Moreover, since the number of cases that involve digital evidence in some form is on the rise all over the world, the digital forensic investigators are facing a pressing need for reducing the investigation time per case (Al Awadhi, Read, Marrington, & Franqueira, 2015).

The second challenge is thrown by the increasing diversity of digital devices that are becoming available in the market (Hossain, Fotouhi, & Hasan, 2015). Digital forensic personnel have to continuously strive for finding new ways (through software as well as hardware means) to acquire and analyze such devices (Inspectorate, 2015). The software diversity deals with variety of file-types, ever evolving Operating Systems, the huge pool of innovative applications, and other software advancements aimed at contemporary digital devices. On the hardware front, diversity of sensors, chips, circuit modules and other hardware units that produce unique data streams presents a challenge for digital forensics. Although providing a solution to each instance of the above-stated diversity challenges is a one-time effort for the practitioners and researchers; however, overall the rate at which these parameters change keeps them on their toes.

Furthermore, people tend to use separate devices for communication, entertainment and productivity purposes. Hence the number of individuals who own and use more than one digital devices at a time is increasing (Facebook-Business, 2014). Another study by Facebook in 2016 reveals that 94% teens in France and 98% teens in Germany own multiple devices (Facebook-IQ, 2016). The Pew Research Center published a report in 2015 stating that around 36% of US adults own all three devices, namely a smartphone, a computer, and a tablet (Anderson, 2015). Another survey by Pew in January 2017 has revealed that 77% of US adult population owns a smartphone, 78% owns a desktop or laptop, and 51% owns a tablet computer (Pew-Research, 2017). Although the survey presents separate figures for the three devices, one can safely assume that individuals who own multiple devices are a significant part of the US population today. The people in other regions of the world either share similar statistics or would achieve the same trends in the near future. The rise in the number of devices owned per person would increase the average number of exhibits seized in a new case, thus increasing the respective investigation time and efforts.

Even after finding their ways to acquire and analyze the new digital devices, the digital forensic examiners face the third challenge from rapid technological advancements that change the rules of the game now and then (Garfinkel, 2010). The technological progress that poses a challenge to investigators is the increasing list of devices that are going digital every day, thanks to the novel hardware innovations. The devices in everyday use which get equipped with com-

The current work is an extension of the paper which was presented at "*The ADFSL 2018 Conference on Digital Forensics, Security and Law*" on May 17, 2018 at the *University of Texas at San Antonio*, TX, USA.

putational, communication and digital storage capability, commonly referred to as Internet of Things (IoT), pose new investigative challenges to the digital forensic process (Oriwoh, Jazani, Epiphaniou, & Sant, 2013). Any investigation involving such devices would require knowledge about how the data is produced, stored and communicated to or from these devices.

The fourth challenge, which is not directly connected to the functioning of the digital forensic investigation, is data privacy protection during the digital forensic investigation (Aminnezhad, Dehghantanha, & Abdullah, 2012). The Digital forensic investigators always get full access to the contents of seized storage media which according to them is necessary for achieving completeness. Apart from containing potential evidence files, the seized storage media also contain owner's private data which may be sensitive at times like private/family pictures and videos, business related digital documents, medical diagnostic or treatment reports, commercial software with license information, and much more. Investigator's open access to these private files is a threat to owner's data privacy (Verma, Govindaraj, & Gupta, 2016).

The data privacy protection is also related to need for transparency in the digital forensic investigation that ensures only case-relevant data are accessed from the seized media and remaining private files are not affected (Dehghantanha & Franke, 2014). There is a pressing need for finding means to fix accountability of the investigator in case a data privacy breach happens during the investigation. The two sister agencies that work in close collaboration with digital forensic personnel, namely the Police and the regular forensic laboratories, are facing difficulties related to transparency and accountability. The case of Annie Dookhan is a good example of the same (Driscoll, 2014). To the best of authors' knowledge, there are no reported instances of professional misconduct against digital forensic investigators till date; however, it is high time that the community should adopt self-regulatory ways to improve the transparency as well as the accountability of the digital investigation process.

Apart from the challenges listed above, some researchers have predicted that moving forward the field of digital forensic would start diverging into more specialized sub-fields (Garfinkel, 2015). The subfields would require the investigators to possess expert knowledge of the respective subdomain. The digital forensic laboratories would need an investigation mechanism that could allow different experts to work together on a given case. There is one more aspect to learning for digital forensic examiners which aims to capture the psychological, cultural and social characteristics of the people who commit computer related crimes (M. K. Rogers, 2011). Researchers have been trying to capture such parameters that could help in digital forensics investigation process (M. Rogers, 1999; M. K. Rogers, Seigfried, & Tidke, 2006).

The digital forensic frameworks to date have focused on addressing the above-stated challenges either in separation or well-defined scenarios with controlled environmental conditions. In the current work, the authors have proposed a new digital forensic framework that incorporates forensic image preprocessing, tool-independent automation, machine learning based filtration of most relevant evidence and their privacy level evaluation to address the above-stated challenges. The framework proposes a new way in which the state of the art digital forensic research and systems could be combined in one place to realize the following.

- Increased investigative efficiency by reducing the investigation time and efforts

- Improved investigative accuracy by using multiple tools at the same time
- Better investigative planning via automation
- Improved validation
- Data privacy protection for forensically non-relevant private files
- Enhanced transparency and accountability
- Building expert knowledge for forensic investigation, education, training, and multi-agency collaborations

2. PROPOSED SOLUTION

The framework takes forensic exhibits and images (of desktops, laptops, smartphones, tablets, or other devices that store data), network logs, memory dumps, and all other sources of digital storage as input (refer to figure 1). As the inputs proceed to the next phase of ***'Forensic Preprocessing'***, the investigator fills in all case related facts into a document called *Current Case Information (CCI).* The document consists of forensically relevant data that is unique to the case under investigation, like individual keywords, timelines, and other useful information. After that, the investigator also provides the list of digital forensic tools, with their respective version numbers. All input images are processed to remove forensically irrelevant data like files listed in NSRL (Seo, Lim, Choi, Chang, & Lee, 2009) and duplicate files (Neuner, Mulazzani, Schrittwieser, & Weippl, 2015; Scanlon, 2016). The forensic image formatting is also changed, without breaking the integrity of the input, to enable fast and parallel operations in successive investigation phases. In case physical devices (exhibits) are available, then the imaging for these seized devices is started simultaneously with the data removal and reformatting. The authors call the above procedure 'forensic preprocessing' as it precedes the actual processing for finding evidence files (the analysis phase). The preprocessing aims to rearrange and consolidate the data available in all of the submitted forensic images (provided in any of the popular formats) so that forensic tools could read the data concurrently. However, all preprocessing techniques and methods should ensure that the output produced by them is compatible with all digital forensic software tools. The section 3 discusses preprocessing in details.

The next step runs the ***'Automated Digital Forensic Processing'*** module. The module takes inputs from the CCI document, a case-specific command list, and some already known exception commands. The '*Case Profile Commands (CPC)*' database contains a list of commands that a specific digital forensic tool would require while performing a case specific job under a particular hardware deployment. These commands listed in CPC-database ensure that the planning of investigative steps is complete and consistent with respect to a particular type of case. For example, in the case of a financial fraud investigation, the CPC-database will contain commands for say Encase tool, version 7.0 running on a Windows 8.1 workstation, to perform a keyword search job (with a list of unique operations, called job-sections, refer figure 2) on a Linux machine's forensic-image that has an EXT4 file-system. The CPC-database contains the comprehensive collection of commands and scripts, to complete distinct tasks, which are executed by the list of forensic software tools already provided by

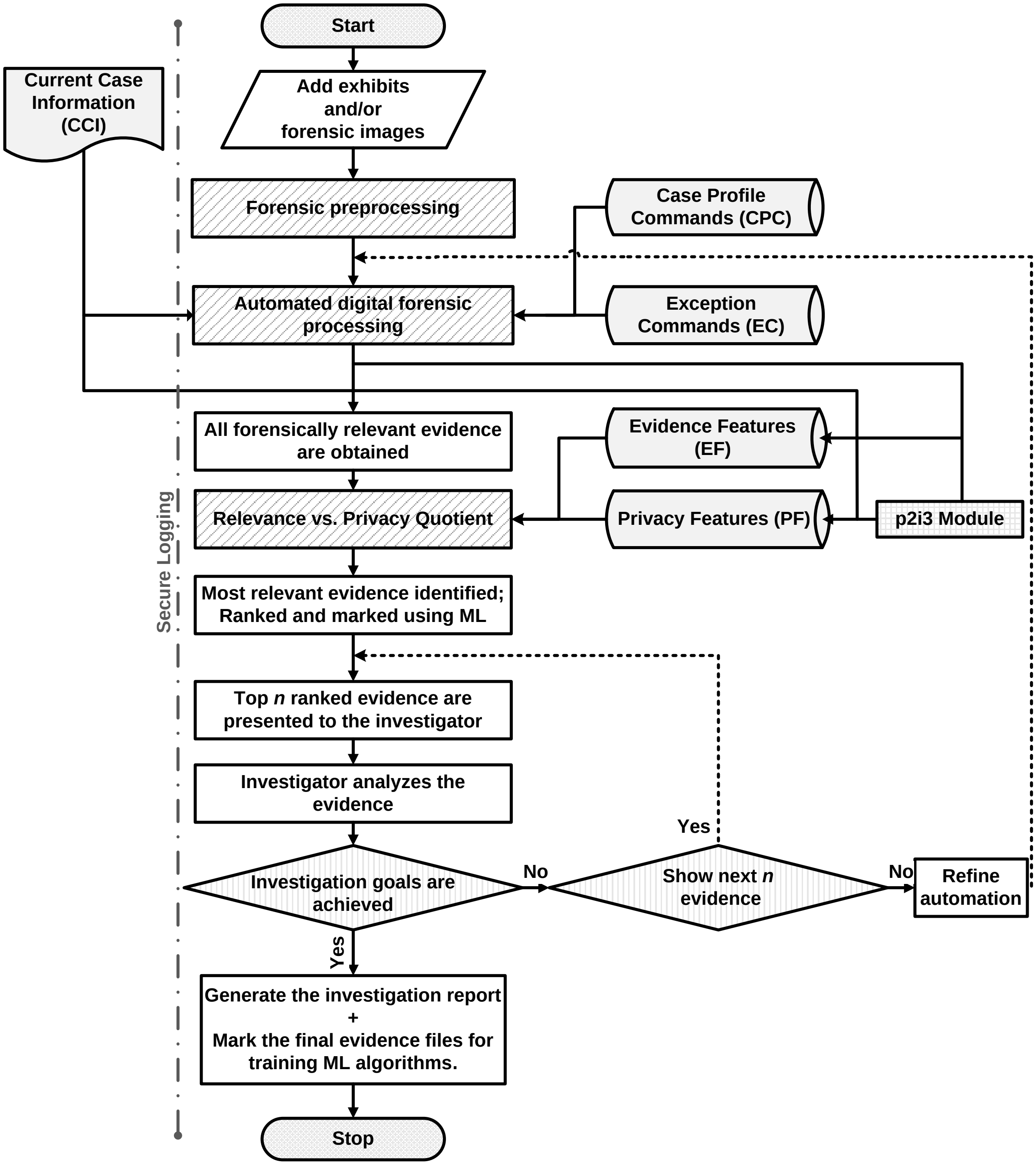

Start
Add exhibits and/or forensic images
Current Case Information (CCI)
Forensic preprocessing
Case Profile Commands (CPC)
Automated digital forensic processing
Exception Commands (EC)
All forensically relevant evidence are obtained
Evidence Features (EF)
Relevance vs. Privacy Quotient
Privacy Features (PF)
p2i3 Module
Secure Logging
Most relevant evidence identified; Ranked and marked using ML
Top n ranked evidence are presented to the investigator
Investigator analyzes the evidence
Yes
Investigation goals are achieved
No
Show next n evidence
No
Refine automation
Yes
Generate the investigation report + Mark the final evidence files for training ML algorithms.
Stop

Figure 1. Digital Forensic 2.0 framework flowchart

the investigator.

The *Exception Commands (EC)* database consists of command structure similar to that of the CPC-database with a distinction that these commands aim to find evidence files that could otherwise be missed during the initial run of forensic tools. For example, all PDF attachments received on Gmail while being viewed by the receiver's browser generates one PNG image for each page of the attached document (Verma, Gupta, Sarkar, & Gupta, 2012). So, when the user login into their account and check their emails with PDF files as attachments, the PNG images corresponding to each page of the viewed PDF document get loaded into the browser cache. These images could be extracted from any of these three sources; the cache on hard-disk drive, the RAM dump or the Hibernation file of the system. A digital forensic investigator should fill in command (or scripts) to parse these PNG files, from the sources described above, in the EC-database.

The EC-database is a collection of all such exclusive commands which can find targeted content. In other words, the database contains expert knowledge which has been acquired over time from individual experience, careful observations, and novel research efforts. In case two forensic labs enjoy a considerable amount of trust and mutual understanding, they could share their EC-databases. The sharing will give the examiners on both sides the opportunity to upgrade their knowledge and enhance their capabilities. In case all forensic labs in a province or state agree to share their EC-databases, it could become a good collection of valuable regional (*demographic*) forensic insights.

Depending on the investigation needs and the availability of forensic tools, the automation module can work with both the open source as well as commercial digital forensic tools. The framework requires that the forensically relevant files processed by the tools have a uniformly high level of data abstraction. For example, the tools should expands all compound files (at a lower level of data abstraction) to extract the contained files (at a higher level of data abstraction) before these files could be passed on to the next level of scrutiny by the framework. Section 4 discusses this in more details.

The results of Automated Digital Forensic Processing are passed on to the next step (***Relevance vs. Privacy Quotient***). Here, with the help of machine learning algorithms, a relevance score for all potential evidence files (obtained from the automation module) is calculated. Similarly, the privacy quotient for these files is also calculated simultaneously. The investigator is then presented with a finite list of the top scoring relevant files. The investigator can analyze these files to decide whether these evidence files are sufficient to prove or disprove the case. If the investigator wants, she could keep on requesting the next lot of most relevant files for further examination, till the list of potential evidence gets exhausted. As soon as the investigator gets sufficient evidence from the relevance list, she may stop the investigation and generate the case report. However, if the investigator feels that the artifacts enlisted in relevance list are not sufficient, she is free to override the filters and start over the automation module.

The framework also incorporates a ***Secure Logging System*** (from start of the investigation till it stops) where all actions and decisions of the investigator are chronologically logged into a secure place. The safe storage for these logs could either be a hardened local server or a reliable cloud space where the investigator has no chance of tampering with them (Barik, Gupta, Sinha, Mishra, & Mazumdar, 2007; Verma, Govindaraj, & Gupta, 2014). Since the investigator may be required to explain her actions in case any privacy breach or some foul play is either doubted or reported. The secure logging ensures that the accountability of the investigator is fixed when such a situation arises. A brief discussion on the same is presented in section 7.

Automation used in the framework simplifies repeatability of the investigation process, which proves to be very helpful in validating the investigation outcomes. Especially, for the ***Technical Validation*** which aims to check whether all steps followed by the investigator fulfill the goals of the investigation. Automation together with the secure-logging will help the digital forensic community to study and optimize the investigative techniques followed by examiners. Repeatability and easy validation could improve the overall transparency of the investigative process. The framework also ensures a three-way error reduction mechanism using automation. Firstly, the automation reduces the chances of human error that may happen at any time. Secondly, the automation ensures that no step is missed from the investigative planning which remains consistent for a particular type of case. Thirdly, the automation ensures that no evidence file is missed due to limitations of a particular tool since results from different forensic tools are combined to present a comprehensive list of potential pieces of evidence. The above solution will keep the investigative powers of the investigator intact with good chances that her overall efficiency gets improved.

2.1 Setup

The proposed framework needs a hardware infrastructure that could provide both high-performance computational power as well as high-speed data storage and access. A robust and capable software should also support the hardware to realize both an efficient parallel processing and a powerful data management mechanism. Another requirement for the software component of the framework is its compatibility to run applications and programs from all publicly available software platforms. So, all state of the art Operating System dependent and Operating System independent digital forensic tools, which are capable of working on various digital devices, irrespective of whether they are closed source (commercial) or open source could be deployed on the proposed framework.

All the forensic tools and applications that are installed on the framework should be able to receive command-line instructions. Since most of the open source digital forensic tools take command-line inputs, they can easily be attached to the framework. Since all commercial tools are closed source, it is the responsibility of their developers to provide a command-line support for their respective tools. Although there are some tools like EnCase, which accept scripts to automate some investigative tasks, there is still a segment of commercial tools that do not support automation. The tools that do not provide any support for automation can not be used with the proposed framework.

Depending on the requirement, the proposed framework can be set up on any of the following configurations:

1. *Beowulf Cluster in a laboratory*- best suited for digital forensic laboratory environments where a suitable number of processing nodes could be selected based on the budget and investigative load (Ayers, 2009). A Beowulf cluster file system provides support for high-performance data access and storage. The processing speed and efficiency of a Beowulf cluster in a laboratory setting are better as compared to a distributed systems deployment or a cloud deployment of the same configuration.

2. *On the Cloud* - a private cloud with a strict access control could be a useful option for an investigation agency, which has multiple departments located at same or different geographical locations (Van Baar, van Beek, & van Eijk, 2014). Alternatively, an agency could also rent virtual machines on a public cloud having comparatively high processing and data storage capabilities.

The catch with cloud-based deployment is the dependency on limited upload and download speeds. However, if the network speeds are favorable, the cloud-hosted framework could enhance remote investigations capabilities where investigators could simultaneously work on the same case.

3. *Distributed Systems* - could also be used to deploy the framework with the processing power comparable to abovementioned deployment models. However, the data access speed, the parallelization in processing, and the file system capabilities would be relatively more complicated and hard to manage (Richard III & Roussev, 2006).

3. PREPROCESSING

The Forensic Preprocessing module is the first component of the proposed framework that operates on the forensic images. The authors call the module 'forensic preprocessing' as it precedes the process of finding evidence files (the analysis phase). The preprocessing aims to rearrange and consolidate the data available in all of the submitted forensic images so that forensic tools could read the data concurrently.

Before preprocessing could begin, the investigator is required to fill in all case related details into the *Current Case Information* (CCI) document. The document consists of forensically relevant information about the case under investigation, like the type of case, the name of the case, suspect's information, keywords of interest, timelines of interest, targeted file types, and other valuable information(refer figure 2). After filling the CCI document, the investigator also provides the list of digital forensic tools, with their respective version numbers, which are installed on her forensic system and best suit the analysis requirements of the given case. The information from the CCI document and the tools list is used by the preprocessing module to fine-tune its operations.

The primary aim of the preprocessing module is to change the data formatting of the forensic images (without breaking their integrity) so that the digital forensic tools attached to the framework could perform highly efficient parallelized operations. The secondary aim is to remove forensically irrelevant data from the forensic images which include files listed in NSRL (Seo et al., 2009) and duplicate files (Neuner et al., 2015; Scanlon, 2016).

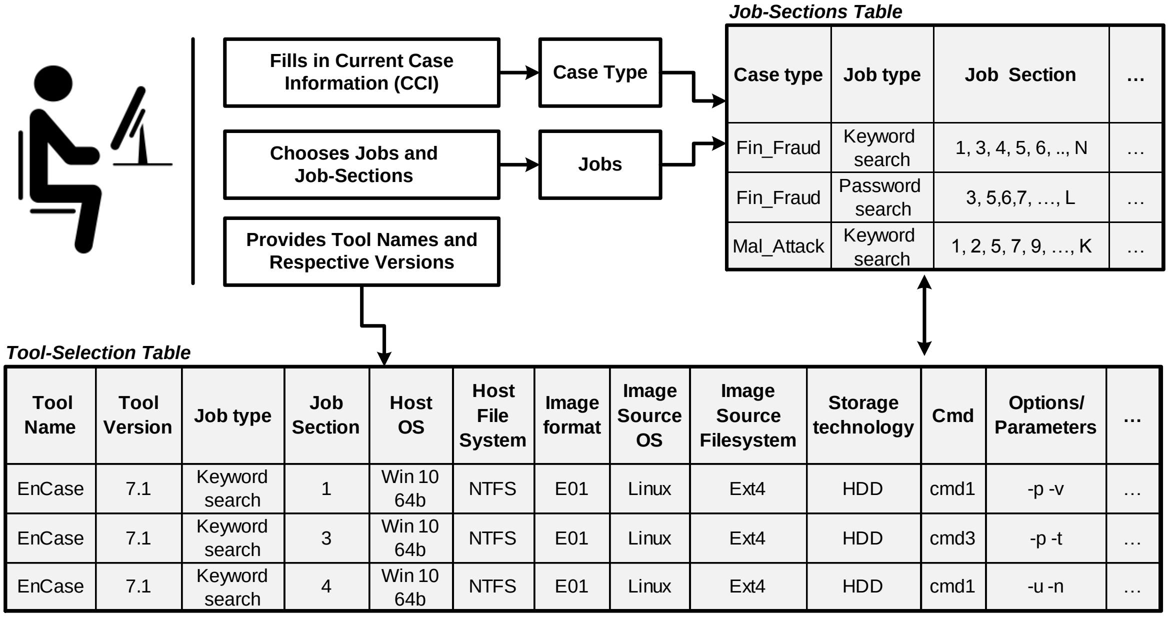
In case physical devices (exhibits) are submitted instead of their forensic images, then the imaging for these seized devices is started simultaneously with the data reformatting and redundancy removal. All preprocessing techniques and methods should ensure that the output produced by them is compatible with the digital forensic software tools due to be used in the automation phase.

The data formatting operation should keep the integrity of the forensic images intact, and hence there should be no impact on the admissibility of the forensic evidence extracted out of the newly formatted data.

4. AUTOMATION

The Automated Digital Forensic Processing module aims to carry out a thorough forensic analysis of the forensic images to collect all case related potential pieces of evidence without any human intervention. The module uses the *Current Case Information* (CCI) document and queries both the *Case Profile Commands* (CPC) database as well as the *Exception Commands* (EC) database (refer figure 3).

The CPC-database is populated by querying two tables, namely the *Job-Sections* table and the *Tool-Selection* table (positioned


Fills in Current Case Information (CCI)
Chooses Jobs and Job-Sections
Provides Tool Names and Respective Versions
Case Type
Jobs

Job-Sections Table

Case type	Job type	Job Section	…
Fin_Fraud	Keyword search	1, 3, 4, 5, 6, .., N	…
Fin_Fraud	Password search	3, 5,6,7, …, L	…
Mal_Attack	Keyword search	1, 2, 5, 7, 9, …, K	…

Tool-Selection Table

Tool Name	Tool Version	Job type	Job Section	Host OS	Host File System	Image format	Image Source OS	Image Source Filesystem	Storage technology	Cmd	Options/ Parameters	…
EnCase	7.1	Keyword search	1	Win 10 64b	NTFS	E01	Linux	Ext4	HDD	cmd1	-p -v	…
EnCase	7.1	Keyword search	3	Win 10 64b	NTFS	E01	Linux	Ext4	HDD	cmd3	-p -t	…
EnCase	7.1	Keyword search	4	Win 10 64b	NTFS	E01	Linux	Ext4	HDD	cmd1	-u -n	…

Figure 2. Investigator's input to the framework

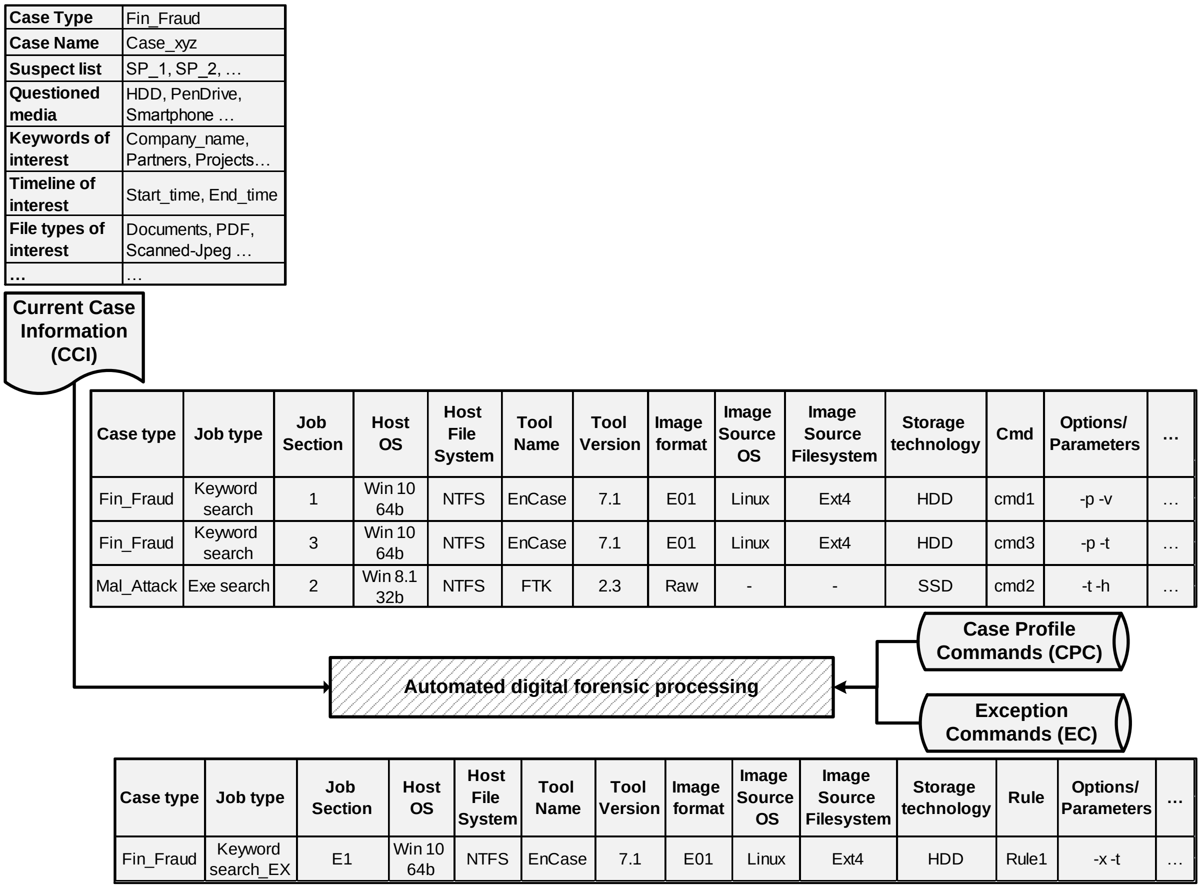
at top right and bottom of figure 2 respectively). The Job-Sections table contains information about various jobs and sub-jobs (the author calls them **job-sections**) that are carried out by the digital forensic tools. The job name specifies a particular task of forensic importance which is used in a digital investigation, for example 'keyword search'. The keyword search can further be divided into small tasks, like searching keywords in all text files (let us call it job-section 1). Similarly, searching for keywords in PDF files is another sub-task (let us call it job-section 2). Likewise, a comprehensive list of well-defined subtasks for a particular job can be populated. If we consider the keyword search job with reference to a particular case (say Financial Fraud), the investigator can identify the list of job-sections that are useful for the investigation of that case.

The Job-Sections table contains this mapping for all type of known case types, respective jobs that are needed to be performed for these case types and the comprehensive list of job-sections for the same.

The Tool-Selection table contains tool version specific commands or scripts to implement job-sections from the Job-Sections table. All of the instructions are stored with respective parameters.

The CPC-database is populated with case-specific commands recognized by the tools, specified by the investigator, for completing a collection of small investigative jobs. The values obtained from the CCI document include specific terms including names of the suspects, names of the companies they are associated with, names of their partners, names of the projects they have handled, and more.

The CPC-database holds all job specific directives that may belong to more than one type of case profiles; for example, keyword search is one job which has application in a variety of cases. The keyword search job can be performed by various digital forensic software tools. However, the search technique implementation along with the key-



Case Type	Fin_Fraud
Case Name	Case_xyz
Suspect list	SP_1, SP_2, …
Questioned media	HDD, PenDrive, Smartphone …
Keywords of interest	Company_name, Partners, Projects…
Timeline of interest	Start_time, End_time
File types of interest	Documents, PDF, Scanned-Jpeg …
…	…

Case type	Job type	Job Section	Host OS	Host File System	Tool Name	Tool Version	Image format	Image Source OS	Image Source Filesystem	Storage technology	Cmd	Options/ Parameters	…
Fin_Fraud	Keyword search	1	Win 10 64b	NTFS	EnCase	7.1	E01	Linux	Ext4	HDD	cmd1	-p -v	…
Fin_Fraud	Keyword search	3	Win 10 64b	NTFS	EnCase	7.1	E01	Linux	Ext4	HDD	cmd3	-p -t	…
Mal_Attack	Exe search	2	Win 8.1 32b	NTFS	FTK	2.3	Raw	-	-	SSD	cmd2	-t -h	…

Case Profile Commands (CPC)
Automated digital forensic processing
Exception Commands (EC)

Case type	Job type	Job Section	Host OS	Host File System	Tool Name	Tool Version	Image format	Image Source OS	Image Source Filesystem	Storage technology	Rule	Options/ Parameters	…
Fin_Fraud	Keyword search_EX	E1	Win 10 64b	NTFS	EnCase	7.1	E01	Linux	Ext4	HDD	Rule1	-x -t	…

Figure 3. Automated digital forensic processing module.

word list(s) would differ depending on the tool specifications and the case profile respectively.

The collection of all jobs that are performed for a particular case type is in public knowledge. Moreover, how a particular job could be carried out by various digital forensic software tools could also be documented. There are tool-specific commands for performing a particular job which could take specific parameters and options based on the case type and information from the CCI document.

All of the above information is captured in the databases, as shown in Figure 3 that makes the automation possible. For example, if the job requirement is keyword search for a Financial Fraud case type where a Windows 10 machine with EnCase version 7.1 installed on it is available, and the forensic image is a Hard Disk Drive with Linux installation needs to be examined, then the first database entry for keyword search could fetch the command(s) with corresponding parameters and options (if applicable). For simplicity of understanding the authors have all columns of the databases in Figure 3; otherwise, the databases could be normalized further.

Even after processing the forensic image with a variety of digital forensic software tools, there are some crucial evidence that might escape the examiner's scrutiny. For example, with the surge in mobile phone usage people have started taking pictures of various documents that they use in their daily lives. Examples include tickets, different identity cards, business cards, bank checks, mark-sheets and sometimes username and passwords for important on-line accounts. The forensic tools that search for keywords only focus on files that have textual data, and would not be able to search for images that have some written content until and unless they are instructed to do so. Experienced investigators have knowledge of such intricate details, like running OCR on suspected images along with keyword search, or filtering out the potential pictures by their metadata in case the OCR engine fails. These approaches could help the investigation by obtaining crucial evidence on the first run. The proposed framework stores these intricate details in the EC-database. The commands include implementation tricks and techniques that come from knowledge gathered by forensic experts over time as well as research breakthroughs. Structurally the database is similar to CPC-database (refer figure 3).

The working of the automation module (especially the structure of CPC-database) which is presented above is inspired by the work of (Karabiyik & Aggarwal, 2014). However, to the best of authors' knowledge, the conceptualization of the Exception Commands database is a fresh contribution.

4.1 Design

An Expert System could be used to design the automation engine. The rules of conducting forensic analysis could be stored in the CPC-database. Different variables that need to be considered like case type, job specification, device type, respective OS and File-System versions, forensic tool's name/version, and respective commands/parameters/options could be modeled into the system.

4.2 Relevant vs. Non-Relevant Files: First level of data privacy preservation

The outcome of the automated digital forensic processing would give a list of files from the forensic image(s) which are potential pieces of evidence for the case under investigation.

The automation module operations segregate all files present in the forensic image(s) into two classes, namely ***Forensically Relevant Files (FRF)*** and ***Forensically Irrelevant Files (FIF)***. The FRF advance to the next stages of the investigation, whereas the FIF is made inaccessible to the investigator.

The denial of access to all files (including the private files) which are present in FIF group, is the **first level of data privacy preservation** ensured by the proposed framework.

5. FORENSIC RELEVANCE VS. DATA PRIVACY

The data privacy aims to protect owner's personal information from falling into hands of unauthorized people (Fischer-Hübner, 2001) (OECD, 2002). Whereas, a digital forensic investigation seeks to find all potential pieces of evidence that indicate a malicious activity carried out in digital space (Pollitt, 2004).

All files that are selected/ highlighted/ exported at the completion of the automation module fall into the Forensically Relevant Files (FRF) group. The number of files in the FRF is still large enough for the investigators to examine individually. Moreover, a considerable number of owner's private files that do not qualify as concrete evidence are also included in the FRF collection. Hence, finding actual evidence files from the FRF group is undoubtedly a massive manual effort, which further involves a significant risk of data privacy violations for the private files that do not have much of evidential value.

The proposed framework uses machine learning to determine the degree of relevance (details in subsection 5.1) as well as the level of privacy (details in subsection 5.2) for all files present in the FRF group. The investigator is presented with the top most relevant files (say, a bunch of top 20 or top 50) for examination, with their respective level of privacy also marked on them.

The next set of most relevant files is not presented to the investigator until she examines the first bunch and feels that further investigation is needed. Only after the investigator raises an explicit request to the system, the next bunch (succeeding 20 or 50) of files is presented for her scrutiny. The process of request and grant continues until the investigator finds all actual evidence needed to resolve the case or the list of FRF gets exhausted. In a rather unusual situation when the examiner feels that the automation module should be rerun, the framework provides a provision of doing so too.

The above-stated mechanism, for presenting most relevant files in a bunch until the investigator finds concrete evidence to prove or disprove the case, also prevents privacy breach to an extent. The process could also be understood as the **second level of data privacy preservation** which is ensured by the proposed framework. Although the data privacy protection in this filtration process is not absolute, however, the data privacy of a large number of files belonging to FRF is significantly preserved.

5.1 Degree of Relevance

The proposed framework classifies files based on their degree of relevance to the current case under investigation. The classification process needs to process data available in the Evidence Features (EF) database (Figure 4). The EF-database takes information about each file that is selected into FRF, and some case specific information from the Current Case Information (CCI) document.

5.1.1 Feature selection

The aim is to classify each file in the FRF into a potentially-conclusive or a potentially-indecisive piece of evidence. The information stored in the EF-database corresponding to each file, belonging to the FRF for a particular case under investigation, acts as a feature-set for a machine learning implementation. The features can come from:

1. The file's metadata: includes information like - File-Type; Time-Stamps; File-Size; File-Address; File Containing Folder Name; File Containing Folder Depth; Access Control Permissions; and Owner(s) of the File.

2. Source image and the automation module: includes information like - Forensic Tool that selected the file; More than one Tool selected the file (Y/N); Job-Type; Job-Section; Level of Data Abstraction; Did the file got extracted from a compound file (Y/N); Source Image Format; Source Image File-System; Source Image Operating-System; Source Image Storage Technology.

3. Use of the Exception Commands: includes information like - Is a result of Exception Command (Y/N); Number of Exception Commands used; Exception Command IDs.

4. The associated Current Case Information: includes information like - Case-ID; Case-type; Has Keywords of Interest (Y/N); Has Name from Suspect

List (Y/N); Is File Type of Interest (Y/N); Does Fall into Timeline of Interest (Y/N).

It is worthy to note that, the list of above-stated features is not exhaustive and may contain more features in each category. Also, the order of features mentioned above does not reflect their respective significance.

5.1.2 Data collection

The data collection happens when a case is investigated using the framework. Two options that may be used by the investigating agencies while doing the data collection are discussed below:

1. Data collection for a particular type of case: It includes collecting data while investigating cases of the same kind. For example, If an investigative agency analyzes only Financial Frauds cases, then all features collected in the Evidence Features database will help in forming a machine learning prediction model most suited for financial fraud cases. Creating a model for a particular kind of case is considerably easy because each case shares a high degree of commonality in their respective feature sets.

2. Data collection for all type of cases: It includes collecting data while investigating cases of all kinds. The features collected in the Evidence Features database will form a machine learning prediction model that could find potentially-conclusive evidence for any given case. Creating a generic model that can make predictions for any case at hand is a difficult task as compared to the previous option because the feature sets will have many variations.

5.1.3 Machine learning approach for relevance

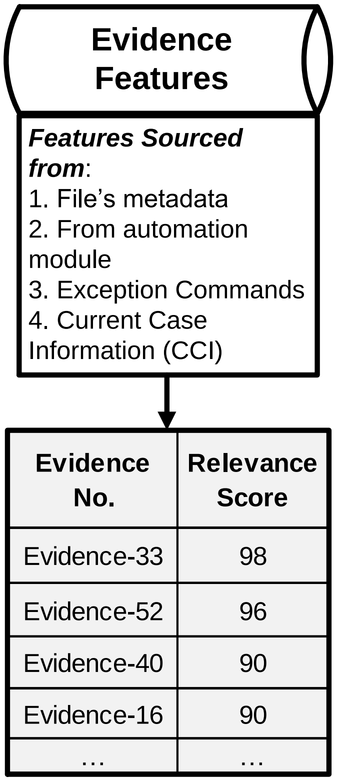
As already stated before, the machine learning solution aims to classify each file in the FRF into either a ***Potentially Conclusive (PC)*** or a ***Potentially Indecisive (PI)*** evidence. Hence, to put it formally -

1. The machine learning approach addresses a two-class classification problem (a *supervised learning technique*). The reason for choosing a supervised learning approach is to learn from the experience of the investigators who have already solved similar cases. The framework needs access to the case related artifacts like the case information document, the forensic image associated with that case, information about the tools that were used to solve the case, and the list of actual evidence files that concluded the investigation.

   The first three artifacts (*mentioned in the previous paragraph*) could be used by the framework to collect feature information about all the FRF files, whereas the last object would act as the ground truth for training. All actual evidence that the investigator marks at the completion of each case investigation help populating the last feature column that is helpful in training.

   After training on some examples of solved cases of the same type, the machine learning solution could start predicting for a new case. However, for a generalized solution, the training set should contain a considerable number of examples of each type of cases that have been solved by the investigative agency before the solution could start predicting.

2. The supervised learning approach could be implemented using an ensemble


Evidence Features
Features Sourced from:
1. File's metadata
2. From automation module
3. Exception Commands
4. Current Case Information (CCI)

Evidence No.	Relevance Score
Evidence-33	98
Evidence-52	96
Evidence-40	90
Evidence-16	90
…	…

Figure 4. Degree of relevance for forensically relevant files.

learning method like Decision tree or Random Forest that give considerably good results when the training data set is less, and the feature set is relatively strong.

The authors think the above-stated learning methods are suitable for the classification task (PC vs. PI) when developing a prediction model for the same type of cases with a relatively small training dataset. However, if an investigation agency that has a collection of a substantial number of cases of the same type say hundred or more cases of financial fraud, then they could try other algorithms like Support Vector Machine (SVM) and k-Nearest Neighbors (kNN).

When a generic solution needs to be created, an ample number of cases of each type that the investigation agency works on is required. However, if multiple agencies agree to share their EF-databases and list of conclusive evidence for respective cases, the aim of making a generic prediction solution could be achieved.

The machine learning approach finds PC files and calculates a relevance score for each of them. The files are then arranged from highest relevance score to the lowest. The framework ensures that only a bunch of most relevant files are presented to the investigator and rest of the files are masked from her. The investigator asks for the next bunch of files if required. The process continues till the investigator finds all conclusive pieces of evidence or the list of FRF gets exhausted. The machine learning solution's efficiency increases with the number of solved cases getting incorporated into the training set.

5.1.4 Mathematical Formulation of Relevance Score

Let the number of input cases be $\boldsymbol{n}$ and the number of features corresponding to an individual file be $\boldsymbol{x}$ (from the EF-database).

$$\mathbf{C} = \{C_1, C_2, C_3, ..., C_n\}$$

Where, $\mathbf{C}$ represent the case vector. The case instance C_i can be represented as a collection of its respective Forensically Relevant Files group (FRF).

$$C_i = \{F_{i1}, F_{i5}, F_{i7}, \ldots, F_{ij}, \ldots\}$$

$$where,\ i \in (1\ to\ n)$$

And, F_{ij} is the j^{th} file in C_i's FRF. Every file in the above set can have a maximum of x features, and the feature vector for F_{ij} can be represented as:

$$\mathbf{f}_{F_{ij}} = \{f_{ij}^1, f_{ij}^2, f_{ij}^3, ..., f_{ij}^x\}$$

So, the case C_i together with its FRF and respective feature vectors can be represented in matrix form as:

$$C_i = \begin{bmatrix} F_{i1} \\ F_{i2} \\ F_{i3} \\ . \\ . \\ F_{ij} \\ . \end{bmatrix} = \begin{bmatrix} f_{i1}^1 & f_{i1}^2 & f_{i1}^3 & f_{i1}^4 & f_{i1}^5 & . & f_{i1}^x \\ f_{i2}^1 & f_{i2}^2 & f_{i2}^3 & f_{i2}^4 & f_{i2}^5 & . & f_{i2}^x \\ f_{i3}^1 & f_{i3}^2 & f_{i3}^3 & f_{i3}^4 & f_{i3}^5 & . & f_{i3}^x \\ . & . & . & . & . & . & . \\ . & . & . & . & . & . & . \\ f_{ij}^1 & f_{ij}^2 & f_{ij}^3 & f_{ij}^4 & f_{ij}^5 & . & f_{ij}^x \\ . & . & . & . & . & . & . \end{bmatrix}$$

The input cases ground-truth evidence can be represented as

$$E = \begin{bmatrix} E_1 \\ E_2 \\ E_3 \\ . \\ E_i \\ . \\ E_n \end{bmatrix}$$

and E_i accounts for the evidence vector corresponding to the i^{th} case which was declared solved after finding files having conclusive evidence. For example, the evidence vector will have a collection of files like

$$E_i = \{F_{i1}, F_{i3}, F_{i5}, \ldots\}$$

$$where,\ Files\ in\ E_i \subset Files\ in\ C_i$$

Here, the feature vector corresponding to the evidence E_i would consist of the union of all prominent features of files mentioned above.

$$\mathbf{f}_{E_i} = f_{F_{i1}} \cup f_{F_{i3}} \cup f_{F_{i5}} \cup \ldots$$

Let us assume that the features which get selected are following

$$\mathbf{f}_{E_i} = \{f_{i1}^1, f_{i1}^5, f_{i3}^9, f_{i5}^{15}, f_{i3}^{19}, f_{i5}^{21}, \ldots, f_{i1}^x\}$$

Since we have x input features, hence the weight vector **W** can be represented as

$$W = \begin{bmatrix} W_1 \\ W_2 \\ W_3 \\ . \\ . \\ . \\ W_x \end{bmatrix}$$

and,

$$W = fn_1(FeaturesMatrix, EvidenceVector)$$

The Relevance Score (RS) for each file present in FRF can be computed as

$$RS = fn_2(WeightVector, FeaturesMatrix)$$

The computation of RS is followed by sorting of the Potentially Conclusive(PC) files from the highest relevance score to lower. The files get clustered into various sets say **p** number of sets and each set has **m** number of files which can be represented as -

$$S = \{S_1, S_2, S_3, \ldots, S_k, \ldots, S_p\}, and$$

$$S_k = \{F_1, F_2, F_3, \ldots, F_l, \ldots F_m\}$$

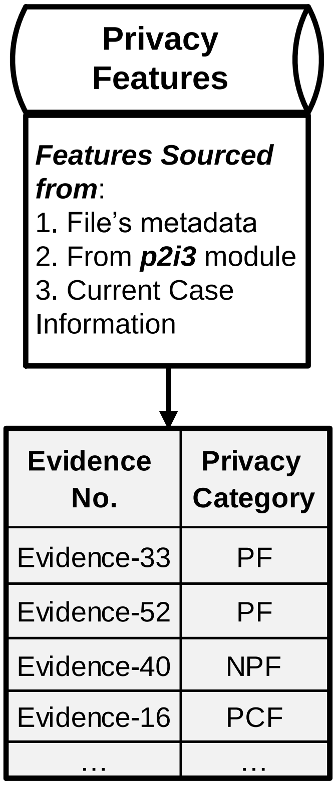
The subsection 6.1.5 provides further information about how machine learning algorithms' scored probabilities can be used for sorting the PC files.

5.2 Privacy Quotient

The framework also identifies whether a file is private or it contains any Personally Identifiable Information (PII) about the suspect. The aim is to correlate the data privacy information for each file with their respective evidence rating (from the previous subsection). The privacy information of each file will not restrict the investigative capabilities of the forensic examiner in any way. However, the privacy quotient of the individual file would enable both the suspect and the legal authorities to assess the scale of data privacy violation, if it happens during the investigation process.

A specific module named Private and PII Identification (**p2i3**) runs on all files belonging to FRF (refer figure 1). The authors have marked the p2i3-module as a separate entity in the flow diagram; however, the module could be a part of the automation engine if some of the forensic tools support the required functionality. For example, the tool EnCase (Version 7 and up) has the provision of finding files that contain personal information as well as artifacts containing Personally Identifiable Information.

All files in the FRF group are examined to determine whether they are private to the suspect or contain any of her PII.


Privacy Features
Features Sourced from:
1. File's metadata
2. From ***p2i3*** module
3. Current Case Information

Evidence No.	Privacy Category
Evidence-33	PF
Evidence-52	PF
Evidence-40	NPF
Evidence-16	PCF
...	...

Figure 5. Privacy quotient for forensically relevant files.

5.2.1 Feature selection

The information stored in the Privacy Features (PF) database acts a feature-set for machine learning implementation to find each file's privacy quotient. The features are described below:

1. Features from file's metadata (same as in the EF-database): It captures information like - File-Type; Time-Stamps; File-Size; File-Address; File containing folder name; File containing folder depth; Access-Control permissions; Owner(s) of the file.

2. Features from the source image and the **p2i3** module: It captures information like - Source image format; Source image File-System; Source image Operating-System; Source image storage technology; Is the file a private file (Y/N); Type of the private information identified; More than one type of private information present (Y/N); Does the file contain any PII (Y/N); Type of PII identified; More than one PII present (Y/N).

3. Features from the CCI document: it captures information like - Case-ID; Case-type; Has keywords of interest (Y/N); Has name(s) from the suspects list (Y/N); Is the File-Type of interest (Y/N); Does the file fall into Timeline of Interest (Y/N).

It may be noted that, the list of above-stated features is not exhaustive and may contain more features in each category. Also, the order of the features does not reflect their respective significance.

The data collection part of the privacy rating solution is same as that of the evidence rating solution (refer sub-subsection 5.1.2).

5.2.2 Machine learning approach for privacy quotient

The aim of the machine learning implementation in the privacy solution is to categorize files from the FRF group into three groups; namely, the Private Files (**PF**), PII Containing Files (**PCF**), and Non-Private Files (**NPF**). Hence, to put it formally -

1. The machine learning approach addresses a clustering problem. An unsupervised machine learning approach is used to categorize the files into one of three clusters (*PF, PCF, and NPF*) as described above.

2. The unsupervised learning approach can use a k-means algorithm to segregate the files into these three clusters. However, there are good chances that the third cluster NPF could get more than 35% of sample population (files from FRF), making the k-means cluster analysis unfruitful. In such a situation the solution needs one extra level of processing.

   The k-means algorithm should be started with a higher value, preferably 3 to 4 times the value of the number of required clusters "n" (*which is currently 3*). An inflated value of n would produce 9 to 12 clusters, each of which would comply with the condition of having the sample population between 5 - 35%.

   A secondary level of clustering on top of these results (using the Hierarchical Clustering) will club them into the final three clusters namely, PF, PCF, and NPF.

6. MACHINE LEARNING IMPLEMENTATION

The authors present a prototype (proof of concept) implementation of ML techniques for predicting the evidential value of a file (subsection 6.1), as well as assessing the privacy quotient of the file (subsection 6.2).

6.1 Prediction of evidential value: Classification

In the absence of a real-world digital forensic case, the authors decided to choose the 'Hacking-Case' for the prototype implementation. The Hacking-Case files are available on NIST's *Computer Forensics Reference Data Sets* (CFReDS) project website (CFReDS, acc. Mar'18).

6.1.1 The setup

The authors downloaded the EnCase images (*two .E01 files*) from the Hacking Case page on CFReDS website. The rest of the steps are enumerated below:

1. The authors collected the metadata information about all the files contained in the forensic image of the given case. The authors used the EnPack *'flat-file-export-(v4-0-0).enpack'* (Key, acc. Mar'18) in EnCase V7 to export around sixty six columns of metadata information corresponding to each file. The table 1 provides selected fields produced by the above-mentioned EnPack. The authors collect all metadata values in a CSV-file and name it as All-File-Dataset (**AFD**).

2. The authors asked five digital forensic investigators working in a private digital forensic laboratory to find answers to 20 investigative questions out of the 31

(listed in *Appendix:A*), which are mentioned on the website. The authors reduced the number of questions for simplifying the analysis process for the investigators. The authors asked each investigator to mark a set of files as potential evidence. The authors collected all five sets of marked evidence, where one investigator's set may have a slight difference in the number of entries as compared to entries in the marked sets of her colleagues. The above-stated scenario is ideal to collect all potential evidence files for the given case because the union of all marked sets from multiple investigators would provide a comprehensive list of answers for each investigative question.

3. The authors asked the investigators to align their tools (EnCase) with the time-settings of the image before they start looking for answers. The same time settings would ensure that marked files collected from all the investigators have consistent time values.

4. The investigators were asked to note down the total time they spent on the case during the investigation.

5. After the investigation process got over, the authors asked the investigators to export the metadata information of their respective marked evidence files (using the *flat-file-export-(v4-0-0).enpack*) into respective Potential Evidence Dataset (**PED**).

6. The authors first collated all values generated by the five investigators in one place and removed the multiple entries. In other words, all PEDs were merged into one CSV file, and duplicate rows were removed. The authors named this file as All-Potential-Evidence-Dataset (**APED**). The rows in APED are unique and present the union of all the files that were marked by the investigators as potential pieces of evidence.

7. The authors added a new column to APED named '*IsEvidence*'; which contained a binary value '1' for all the rows signaling that all entries in the table were potential evidence files.

8. At the same time, the authors also added 'IsEvidence' column to the AFD, and made all entries '0'; asserting that all entries in the AFD were non-evidence files.

9. Finally, the authors merged all entries of the APED into the AFD, and removed the duplicate rows where the 'IsEvidence' value was '0'. The final CSV became the dataset which was used in ML implementation.

It is worthy to note that some files, marked by the investigators which were registry values, while exported to their respective PED's did not had any of their timestamps (*like Accessed, Acquired, Created, Modified, or Deleted*) except Written. The authors populated the missing timestamps of these registry entries using the time details of their parent files in the final dataset.

6.1.2 The dataset

As already mentioned in the previous subsection, the authors have used *flat-file-export-(v4-0-0).enpack* (Key, acc. Mar'18) to export all files present in the Hacking-Case EnCase images. There were a total of 12,190 files present in the case. After exporting the metadata of all these files using the EnPack, the authors found that only 11, 937 entries were populated in the output file (AFD). The exporting script ignored 98 files which

Table 1. The reduced set of columns present in the dataset.

Type	Columns
Time	Accessed, Acquired, Created, Deleted, Modified, Written
String	Category, Description, EvidenceFile, Extension, Extraction Status, ItemType, Name, ShortName, PrimaryDevice, Protected, Signature, SignatureResult, SignatureTag
Numeric	ExtentCount, FileID, InitializedSize, LogicalSize, PhysicalLocation, PhysicalSector, PhysicalSize, UniqueOffset
Addresses	FullPath, Matching File Path, OriginalPath, SymLink
Alpha-Numeric	GUID, MD5Hash, SHA1Hash, StartingExtent
Binary	HasAttributeList, HasPermList, IsCompressed_B, IsDeleted_B, IsDisk_B, IsDuplicate_B, IsEncrypted_B, IsFolder_B, IsHardLinked_B, IsHidden_B, IsIndexed_B, IsInternal_B, IsMountedVolume_B, IsOverwritten_B, IsPicture_B, IsSparse_B, IsStream_B, IsVolume_B, WasProcessed_B, **IsEvidence**

had a physical size of zero bytes. Moreover, 120 archive/composite files were also skipped by the script during the process.

The authors intentionally removed one row from the database, which had the address 'Dell Latitude CPi\C.' Encase adds all images in a case under an imaginary 'C' folder, which acts as the root folder for that case. Since the 'C' folder entry does not have any evidential value or actual existence in the real case, the authors decided to remove the same. Hence, the total count of entries in the AFD database decreases to 11, 972.

The authors then carried out the EnCase processing on case image and exported the metadata again. The processing of the image recovered metadata entries corresponding to 120 archive/composite files that were missed earlier.

After combining the metadata entries (PEDs) obtained from the five digital forensic investigators, the authors got a total of 259 metadata entries for all marked potential evidence files in the APED.

It may be noted that the forensic investigators also marked registry entries as evidence files, whose metadata information are not present in the initial forensic image. The EnCase expands the registry files (like SAM) and enables the investigator to mark the entries within. There are a total of 23 registry entries that are included in the APED.

Finally, the authors merged the entries from the APED into the AFD, and obtained the final dataset, with a total of 12,115 entries.

The number of columns exported by the EnPack is 66; however, the authors removed some of the columns that hold information specific to EnCase or the EnPack. For example, columns like 'Codepage', 'Complexity', 'Entropy', 'Tags', and 'Recepiant' are EnCase specific columns which are not so tightly related to the actual file. Similarly, columns like 'Output filename' is an example of the EnPack specific column which is not related to the file.

Moreover, there are some other columns

that hold redundant information; like the columns 'Full Path', 'Item Path', and 'True Path' have the same content. The authors removed such types of columns as well, and reduced the set of columns to 55 (refer table 1 for details). The 56th column 'IsEvidence' is populated by the values received from the digital forensic examiners. The value '1' in the column means that the file is potential evidence in the given case, and a '0' means it is not.

6.1.3 Experiments and results

The authors conducted experiments on the dataset using various baseline algorithms, that include promising Machine Learning (ML) algorithms suited for two class classification, results of which are presented in the next section.

Digital forensic investigation process aims to capture all potential pieces of evidence, and could not afford to lose any possible evidence that may slip through the investigator's scrutiny as a benign file. The similar scenario happens in the ML results, where the False Negatives (also called Error type 2 in ML) are the files that are actually potential evidence files but have been wrongly predicted as innocuous. Considering the harm that False Negatives values can have on the outcome of the investigation, the authors used the technique of 'Bagging' to reduce their values. Results of the same are presented in the next section.

Experimental protocol: The authors divided the dataset into training and testing in the proportion of 80% and 20% respectively. Hence the training dataset contains 9,692 records, and the testing dataset includes 2,423 records.

6.1.4 Baseline performance

The authors have used some popular ML algorithms which are known to be good performers on the two-class classification problems. The authors have used seven algorithms, namely - Support Vector Machine (SVM), Two-class Logistic Regression, Deep SVM, Decision Forest, Decision Jungle, Boosted Decision Tree, and Neural Networks. There is no specific reason for the selection of these seven algorithms, and other algorithms can also be used on the dataset to accomplish the required classification job.

The confusion matrix of each of these algorithms is presented in table 2. The positive labeled entries (marked evidence files) in the dataset are significantly less than the negative labeled entries (where 'IsEvidence' value is '0'), so the accuracy values of the algorithms do not reflect each algorithm's actual performance. Hence, the authors have also incorporated the *Equal Error Rate* (**EER**) of the respective algorithms in the results (table 2). The lower the EER, the better the performance. The ROC curves of all these algorithms are plotted in figure6 for easy comparison.

The baseline algorithms results show that all algorithms are not performing good when it comes to tackling the type 2 errors; the False Negatives (**FN**). The high values of the FN are not right from the digital forensic perspective too, as they allow actual evidence files to slip through the investigator's scrutiny as innocuous files. However, the False Positives (**FP**), called type 1 errors in ML terms, on the other hand, could also be problematic for digital investigator as they mark innocent files as potential evidence files. However, since all the files predicted by the proposed ML solution are presented to the investigator for final decision making, all the FP would be easily identified at that time.

The ML prediction could be meaningful in digital forensics scenario if the FN are reduced to a minimum. The authors have applied 'Bagging' technique to achieve the same goal; which is explained in the subsec-

Table 2. The baseline performance of various ML algorithms on the dataset.

Algorithm	Confusion Matrix		Confusion Matrix Format		Accuracy	EER
Support Vector Machine	28	28	TP	FN	98.37	0.1071
	12	2379	FP	TN		
Two-Class Logistic Regression	36	20	TP	FN	98.81	0.0954
	9	2382	FP	TN		
Two-Class Locally-Deep SVM	7	49	TP	FN	97.67	0.2857
	8	2383	FP	TN		
Two-Class Decision Forest	0	56	TP	FN	97.75	0.1045
	0	2392	FP	TN		
Two-Class Decision Jungle	0	56	TP	FN	97.75	0.2592
	0	2392	FP	TN		
Two-Class Neural Network	44	12	TP	FN	97.96	0.0954
	38	2353	FP	TN		
Two-Class Boosted D-Tree	44	12	TP	FN	99.31	0.0276
	5	2386	FP	TN		

tion 6.1.6.

6.1.5 Relevance score

As mentioned in the mathematical formulation section 5.1.4, the framework would present the potential evidence files, sorted in order of their relevance score, to the investigator. The framework uses the 'scored probabilities values' as the relevance score for a particular file (*the table 3 shows some examples of the same*).

In the table 3, the second column '*IsEvidence*' is the labeled column which belongs to the input dataset (AFD). The third column contains the scored values predicted by the trained ML model (*here, the SVM model*). The fourth column holds the respective probability scores with which the ML model has predicted the classification.

It can be observed from the table that the entry 5, is a False Positive **(FP)**; a non-evidential file marked as potential evidence. Whereas, the entries 7 and 8 are False Negatives **(FN)**; actual evidence files marked as

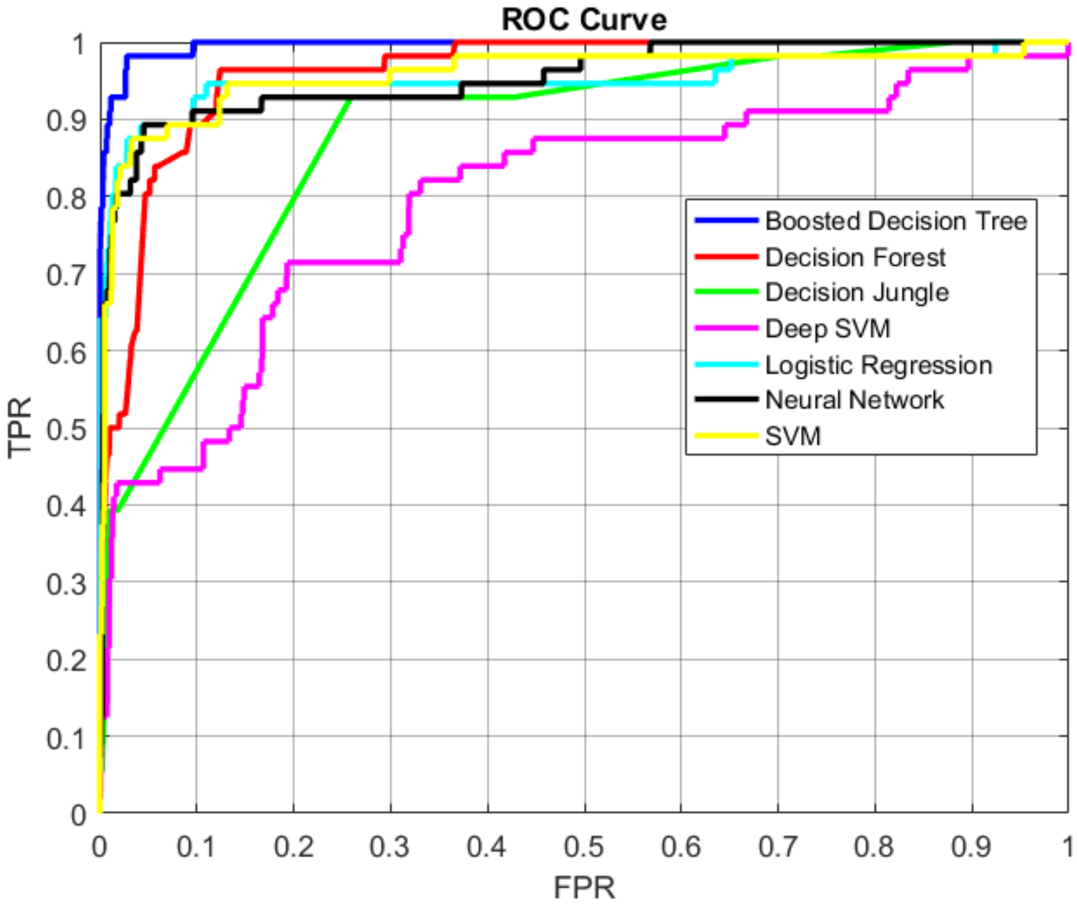


Figure 6. The ROC for baseline algorithms.

Table 3. The relevance scores of files.

S. No	IsEvidence	Scored Labels	Scored Probabilities
1	1	1	0.9996124506
2	0	0	0.0007406375
3	0	0	0.0027789336
4	1	1	0.9765605330
5	**0**	**1**	0.9028829932
6	0	0	0.0036880332
7	**1**	**0**	0.1220335960
8	**1**	**0**	0.0965592340

benign ones. The baseline implementation results show large numbers of FN, which are not suitable for the digital forensic investigation. Hence the authors resorted to 'Bagging' technique to reduce the same.

Table 4. The performance of 'Bagging' on the dataset.

Number of Classifiers	Confusion Matrix		Confusion Matrix Format		Accuracy	EER
15	49	3	TP	FN	91.5	0.0599
	203	2168	FP	TN		
30	50	2	TP	FN	91.79	0.0627
	197	2174	FP	TN		
45	50	2	TP	FN	90.3	0.0585
	233	2138	FP	TN		
60	47	5	TP	FN	91.62	0.0606
	198	2173	FP	TN		
75	51	1	TP	FN	86.83	0.0611
	318	2053	FP	TN		

6.1.6 Bagging

The ML technique 'Bagging' solves a given problem by creating multiple weak ML models that take almost equal portions of positive and negative label samples for training.

Once ready, all these ML models give their respective predictions for a given test sample. The final decision on that sample is taken through a majority voting over all of these predicted values.

The authors took a 40 to 60 ratio of positive to negative labeled samples (selected with replacement) to train the groups of two-class classifiers using the Neural Networks ML algorithm. The authors tried with five different group sizes, namely 15, 30, 45, 60 and 75; and tested these groups of classifiers to predict for the current case. The results of these 'Bags' of classifiers are provided in the table 4. The ROC curves corresponding to bag-level results are presented in the figure 7.

It can be observed from the confusion matrix of these groups that the False Negatives (**FN**) have been reduced to low values; for example, the FN for the '75-Classifiers' group is just 1. Although the accuracy value for the classifier groups varies, the authors observed that the '45-Classifiers' group gives the best performance in terms of a low FN (2), a low EER (0.0585), and reasonably high Accuracy (90.3).

6.2 Determining the privacy quotient: Clustering

After creating a prototype ML model for predicting the evidential relevance of files, the authors also implemented another ML model that could cluster files based on their privacy quotient.

The clustering implementation aims to segregate all files present in the input digital

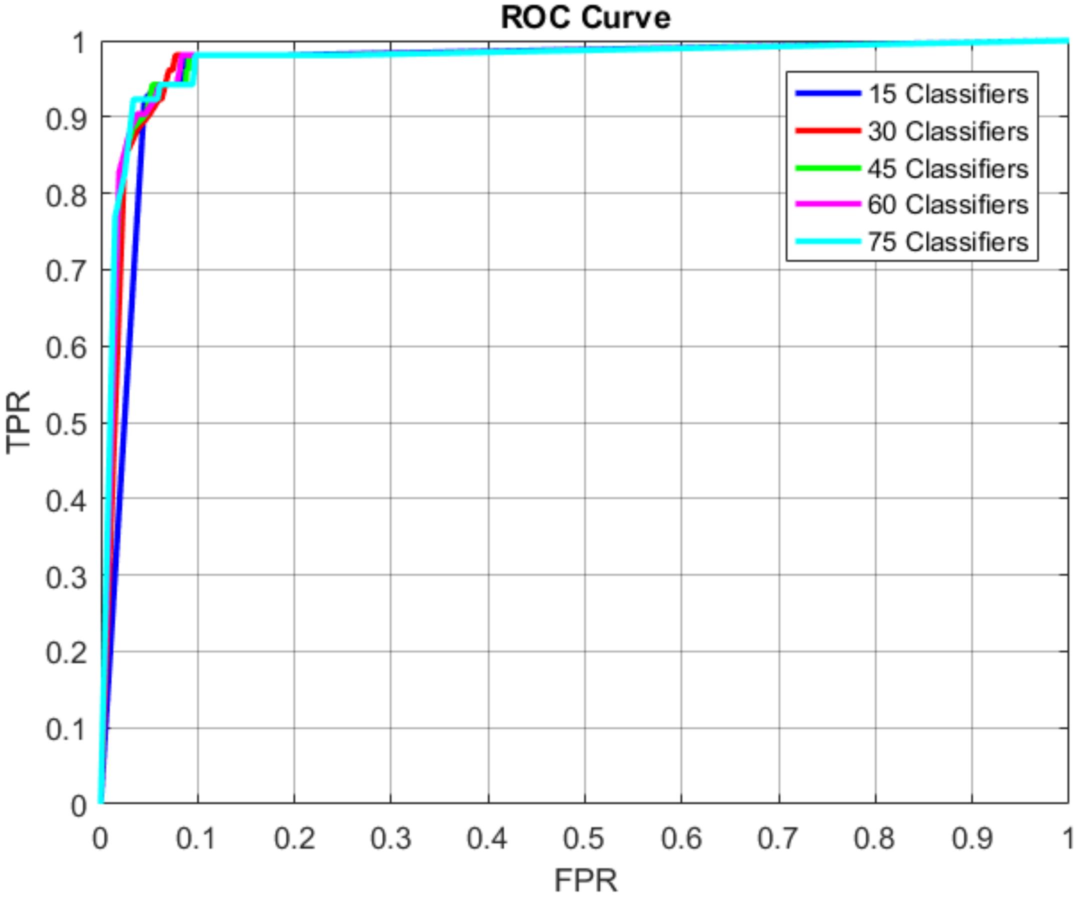


Figure 7. The ROC for the bagging approach.

forensic image into different classes. These classes can then be labeled as either Private Files (**PF**), PII Containing Files (**PCF**), or Non-Private Files (**NPF**).

6.2.1 Dataset

The authors have used the same digital forensic image as discussed in the section 6.1; which is the 'Hacking-Case', available on the CFReDS website (CFReDS, acc. Mar'18).

The authors have used the same dataset for the privacy ML prototype implementation. Since the 'Hacking Case' is a generated case which has been developed by CFReDS for training purposes, it does not have much private information that could be clustered out. Hence, for the sake of simplicity and prototype implementation, the authors has assumed all media files (pictures and multimedia category) as PF, all documents files as PCF, and rest of the files as NPF.

6.2.2 Dataset processing

The ML clustering algorithms use higher levels of numerical calculations in the background, before they could assign a clustering label to given entries; hence they prefer more numeric valued columns in the input datasets.

The current dataset (*introduced in the previous sub-section 6.1.2*) has a plentiful of string-valued columns like 'Category', 'Extention', 'Full-Path', 'SignatureTags', and others. So, in order to get fruitful clustering results, the authors carried out data-manipulation and transformation for several non-numeric columns. For example, the authors changed the string-valued binary columns (*containing 'TRUE' or 'FALSE' values*) into binary-valued columns (*'1' for 'TRUE', and '0' for 'FALSE'*). The columns in the binary category of table 1 with names like 'IsDisk_B', 'IsDuplicate_B', and 'IsPicture_B' are examples of the same.

For feeding data to clustering algorithms, the authors dropped some columns from the input dataset which were not helping with the clustering process. The insights about which columns should be dropped, and which data-manipulation and transformation techniques should be used came from extensive experimentation.

6.2.3 Experiments and results

The authors used K-Means and Hierarchal clustering algorithms for grouping the PF, PCF, and NPF files. The authors used a data transformation on the 'Category' column of the dataset, which maps numerical values (1 to 9) to the string values of the column. The above-stated mapping is provided in the table 5.

Table 5. The data translation of the 'Category' column.

Numeric Code	Categories
1	Library
2	Windows
3	Executable
4	**Picture, Multimedia, Multimedia-Video**
5	**Document, Document-Presentation, Document-Spreadsheet**
6	None
7	Folder
8	Archive
9	Script, Unknown, Email, Database, Communication, Plug In, Internet, Code, Font, Application

Table 6. The K-Means clustering results.

Purity of cluster	Dominating class
0.677364865	1
0.999285204	**4**
0.553819444	1
0.487112046	2
0.461617195	2
0.321135991	6
0.993489583	7
0.682042834	1
0.707509881	1

For the k-means clustering to work, every potential cluster should have between 5% to 35% of the sample population respectively. Since the number of samples in NCF are more than 35% of total samples, the authors segregated the NCF into seven sub-groups (7 numeric codes in total). Also, since all media files are in PF, the authors assigned one numeric code (*code-4*) to them. All the documents files are in PCF are assigned one numeric code (*code-5*). Therefore, the authors chose nine numeric codes for data translation as mentioned in table 5. The segregation keeps the labeled samples in check and clustering algorithm is able to perform better.

All the clustering experiments aim to get a maximum purity value for code-4 and code-5 groups. The 'purity' of a particular cluster with respect to an input class refers to the probability with which the samples of that class map into that cluster after clustering. The higher the purity value, the better is the

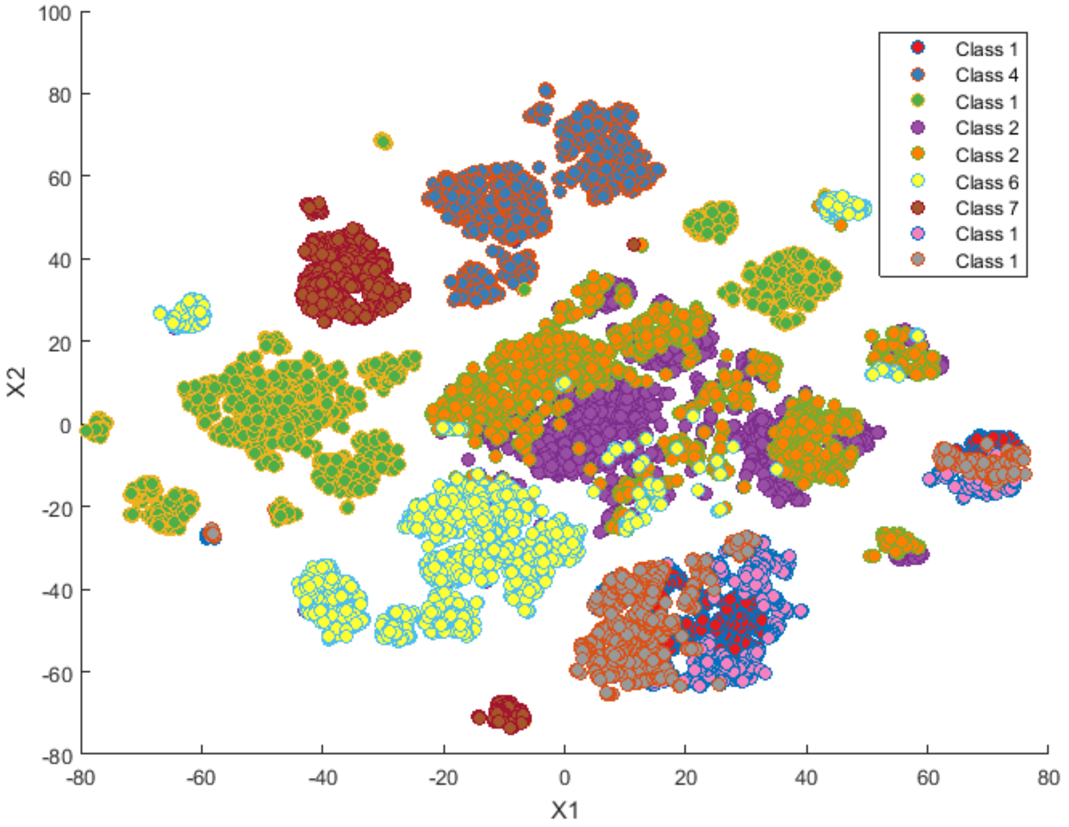


Figure 8. The t-SNE plot for K-Means clustering results.

clustering result for that class.

The results of the k-means algorithm are shown in table 6. It can be noticed that the purity of code-4 (PF files) in cluster 2 is '0.9992' (*very close to 1*). However, the clustering results for code-5 (PCF) are not so good, as there is not a single output cluster where code-5 dominates the results.

The same results can be visualized in a better way using the **t-SNE** plot (*a dimensionality reduction method*); which is shown in the figure 8. Here, the second cluster which is dominated by code-4 (class-4) can be seen as a dark patch in the top center of the plot.

Table 7. The Hierarchical clustering results.

Purity of cluster	Dominating class
1	9
1	9
1	7
0.923076923	9
1	9
1	9
1	9
1	9
0.254199420	1

However, when the authors applied hierarchical clustering on the same dataset, the results were not so encouraging. The hierarchical clustering was not able to clearly distinguish either code-4 (PF) or code-5 (PCF) in any of the nine output clusters. The results for hierarchical clustering are stated in table 7.

7. SECURE LOGGING SYSTEM

The logging process ensures that all operations from the starting state in the proposed framework (refer the flowchart in figure 1) till the state when the investigation stops are recorded. The logging also ensures that all actions of the examiner starting from the time when she begins the analysis process till all conclusive evidence get identified are listed. All system operations and investigator actions need logging because of two reasons; firstly, to resolve conflicting situations like allegations of data privacy violations; secondly, for studying investigation styles of examiners for learning and training purposes.

The logging system could fulfill both of the above-stated requirements only when the logs are complete as well as tamper-proof. The first requirement of completeness, which is relatively easy to achieve, refers to logging all activities of the system and the investigator.

However, the second requirement of ensuring that the logs become tamper-proof is a difficult problem. The first possible solution could capture the activity logs with the help of a dedicated application running on the forensic system. This solution assumes that the examiner is cooperative and honest enough not to interfere with the logging application. After the investigation process is complete, the logging application should transfer the logs to an external storage place which is safe from tampering. Any tampering attempt during its operation would cause the application to stop prematurely, invalidating the captured logs.

The second possible solution should try to capture examiner's activities at the operating system level (with a system level application or module) and save the logs in a safe location. The safe storage for these logs could either be a hardened local server or a reliable cloud space where the investigator has no chance of tampering with them Barik et al. (2007).

Since the investigator may be required to explain her actions in case any privacy breach or some foul play is either doubted or reported. The secure logging fixes the accountability of the investigator for her actions, in case such a situation arises.

8. RELATED WORK

A digital forensic process model denotes the way in which an investigation should proceed from the time of first response, to an incident till the investigation is completed. It acts as a user manual for the investigators, to guide them on how to collect and analyze potential evidence from devices.

8.1 Data privacy protection in Digital Forensics

Although there are plenty of digital forensic process models discussed in digital forensic literature, only a few incorporates privacy of data into the digital forensic investigation process. There are some excellent papers that have provided solutions to the data privacy protection problem in the digital forensic scenario. However, their solutions are either designed for a specific environment and not generic in nature; or the privacy protection works as a separate module that has performance implications.

Staden (2013) proposes a framework that protects the privacy of third party during a digital forensic investigation with the help of a profiling and filtering mechanism. Depending on the sensitivity of data being queried, a decision is taken whether the data should be presented to the examiner or not. The paper focuses on enhancing the privacy in multi user environments,that are subjected to post incident investigations.

Dehghantanha and Franke (2014) have defined the same as a cross-disciplinary field of research and named it as ‘privacy-respecting digital investigation’. They also talk about the present challenges and opportunities that the field has to offer.

Aminnezhad et al. (2012) state that digital forensic investigators face a dilemma whether they should protect suspects’ data privacy or achieve completeness in their investigation. The paper also states that there is a lack of awareness among professional digital forensic investigators regarding suspects’ data privacy, which could result in an unintentional abuse. There have been attempts to protect data privacy during digital forensic investigation using cryptographic mechanisms. Law et al. (2011) have proposed a way to protect the data privacy using encryption. The authors talk of encrypting data set on an email server and indexing the case related keywords, both at the same time. The investigator gives keyword input to the server owner, who has the encryption keys, to get back the emails that contain the keyword.

Hou, Uehara, Yiu, Hui, and Chow (2011b) propose a mechanism to protect the privacy of data on third party service provider’s storage center form the investigator using homomorphic and commutative encryption. At the same time, the mechanism also ensures that the service provider does not get to know the queries that were fired by the investigator. Hou, Uehara, Yiu, Hui, and Chow (2011a) talk of a similar solution on a remote server.

Shebaro and Crandall (2011) use Identity Based Encryption to carry out a network traffic data investigation in privacy preserving setting. Guo, Jin, and Huang (2011) put forward generic privacy policies for network forensic investigations.

Croft and Olivier (2010) have proposed a mechanism where data is compartmentalized into layers of sensitivity, less private data on lower layers and highly private data on higher layers. Investigator’s access to private information is controlled by initially restricting his access to the lower layers first. The investigator is required to prove his knowledge of the low-level layers, to get access to higher level information.

The Df 2.0 framework ensures that the data privacy protection is incorporated into the digital forensic model and hence does not have any impact on the efficiency of the investigation process.

8.2 Next generation of digital forensics

The subsection discusses some notable research works that proposed the next level of digital forensics. They either incorporated high levels of hardware performance or advocated the use of automation as a performance enhancement measure; or both.

Ayers (2009) enlists the limitation of the first generation of digital forensic tools that are struggling with the huge volumes of data involved in modern day investigations. The author proposes several parameters to measure efficiency, together with the requirements that need to be incorporated into the second generation of digital forensic tools. The author also proposed processing architecture of second generation tools which utilizes Beowulf clusters, supercomputers, distributed systems, and grid computing. The evidence storage, workflow management and software reliability of the second generation tools are also discussed. The paper provides requirements and high-level characteristics of the system that was under development.

Garfinkel (2010) also talks about the requirement for data standardization and modular mechanisms in the field for digital forensics and digital forensic research.

Van Baar et al. (2014) have brilliantly moved the digital forensic processing on a cloud where high-end machines could speed up processing and help different actors involved in a digital forensic investigation to collaborate on a particular case.

Carrier, Spafford, et al. (2005) proposed a way to automate searches in digital forensic investigations. Richard III and Roussev (2006) suggested a way to handle large-scale digital investigations with the use of distributed computing. They proposed the use of a cluster of distributed computers to facilitate processing and store the images and results at a central data store. The authors suggested the use of automation by all forensic tools so that they may handle the challenges of tomorrow.

Abbott, Bell, Clark, De Vel, and Mohay (2006) proposed an automated way to correlate events for digital forensic investigation. The authors also demonstrate implementation using publically available digital forensic scenarios and data.

9. CONCLUSION AND FUTURE WORK

The authors have proposed a new digital forensic framework that brings efficiency in digital forensic processing with the help of automation while preserving data privacy for the suspect. The framework ensures that the automation supports a range of digital forensic software tools and produces effective outcomes by incorporating the current case information, case profile data, the knowledge of experienced digital forensic investigators. The investigator is presented with the most relevant evidence that are sorted with the help of machine learning algorithms. The framework balances the investigative requirements of the case with the data privacy protection of suspect's forensically irrelevant private files. The framework ensures that the efficiency of investigation is enhanced, without compromising on the outcomes of the investigation or affecting the investigative powers of the examiner. However, since the system is securely logging all actions of the investigator, she experiences a greater sense of accountability for avoiding unwanted data privacy violations. The automation and secure logging encourage a better validation check, hence bringing a higher level of transparency into the investigation process.

The current work also exhibits a prototype ML implementation that predicts the evidential relevance of a given file that is present in the forensic image of a case under investigation. The algorithm predicts whether a given file is potential evidence or not. The prediction task has been modeled as a supervised learning problem (two-class classification) where the ML algorithm aims first to get training on a labeled dataset, followed by making predictions on the records of the testing dataset. Firstly, the performance of seven baseline ML algorithms was tested on the CFReDS's 'Hacking Case' dataset ((CFReDS, acc. Mar'18)). In spite of giving a reasonable accuracy the results from these seven algorithms show a high rate of False Negatives, which is not acceptable in the digital forensic investigative scenario. So, the authors used the 'Bagging' technique, that takes the predictive decision by taking a majority voting over the predictions of a bunch of weak machine learning models that are trained on small portions of nearly equal parts of positive and negative labeled samples from the dataset. The use of bagging significantly reduced the number of False Negatives, making the ML predictions more usable for a digital forensic investigator. The implementation of ML techniques for assessing the privacy quotient showed encouraging results. The k-means algorithm implementation produced an exclusive output cluster that was dominated by the PF class. However, the results for the PCF class were not so promising.

9.1 Future work

The authors would like to deploy the ML solution in real life digital forensic cases. It would require the authors to have access to these real-life forensic images, which are available in various digital forensic laboratories. Since sharing the data may be difficult for the agencies who possess the forensic images of such cases. So, the authors would like to make an independent standalone system that could be used by the digital forensic personnel working in a laboratory environment to extract the datasets from the cases they have in their possession and build ML trained models on them.

As the dataset holds only the metadata information of the files contained in the digital forensic image; it would be easy for the forensic personnel to share their respective datasets and the ML trained models with the research community as well as their colleagues in other digital forensic laboratories. In the long term, the authors would like to combine the ML models trained on cases from different laboratories in one geographical region with the models from neighboring laboratories (*state-level or country-level*) to find how the prediction pattern varies; and how these learnings could be used in making a universal prediction system. Another exciting extension could be applying the prediction models trained on data in one language to predict cases containing different languages.

Since the *General Data Protection Regulation* (**GDPR**) has come into force from May 2018; the authors would also like to incorporate all privacy compliance measures into the DF 2.0.

ACKNOWLEDGEMENTS

The Research efforts of the first author are supported by Tata Consultancy Services (TCS) Limited, under the ***TCS Research Scholarship Program***.

REFERENCES

Abbott, J., Bell, J., Clark, A., De Vel, O., & Mohay, G. (2006). Automated recognition of event scenarios for digital forensics. In *Proceedings of the 2006 acm symposium on applied computing* (pp. 293–300).

Al Awadhi, I., Read, J. C., Marrington, A., & Franqueira, V. N. (2015). Factors influencing digital forensic investigations: Empirical evaluation of 12 years of dubai police cases. *The Journal of Digital Forensics, Security and Law: JDFSL*, *10*(4), 7.

Aminnezhad, A., Dehghantanha, A., & Abdullah, M. T. (2012). A survey on privacy issues in digital forensics. *International Journal of Cyber-Security and Digital Forensics (IJCSDF)*, *1*(4), 311–323.

Anderson, M. (2015, November). *Smartphone, computer or tablet? 36% of americans own all three.* http://www.pewresearch.org/fact-tank/2015/11/25/device-ownership/. (Accessed: 2018-01-14)

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

Barik, M. S., Gupta, G., Sinha, S., Mishra, A., & Mazumdar, C. (2007). An efficient technique for enhancing forensic capabilities of ext2 file system. *digital investigation*, *4*, 55–61.

Carrier, B. D., Spafford, E. H., et al. (2005). Automated digital evidence target definition using outlier analysis and existing evidence. In *Dfrws.*

CFReDS. (acc. Mar'18). *Hacking case.* `https://www.cfreds.nist.gov/Hacking_Case.html`. (Accessed: 2018-02-03)

Croft, N. J., & Olivier, M. S. (2010). Sequenced release of privacy-accurate information in a forensic investigation. *Digital Investigation*, *7*(1), 95–101.

Dehghantanha, A., & Franke, K. (2014). Privacy-respecting digital investigation. In *Privacy, security and trust (pst), 2014 twelfth annual international conference on* (pp. 129–138).

Driscoll, S. K. (2014). I messed up bad: lessons on the confrontation clause from the annie dookhan scandal. *Ariz. L. Rev.*, *56*, 707.

Facebook-Business. (2014, March). *Finding simplicity in a multi-device world.* `https://www.facebook.com/business/news/Finding-simplicity-in-a-multi-device-world`. (Accessed: 2018-01-14)

Facebook-IQ. (2016, February). *The multidevice movement: Teens in france and germany.* `https://www.facebook.com/iq/articles/the-multidevice-movement-teens-in-france-and-germany/`. (Accessed: 2018-01-14)

Fischer-Hübner, S. (2001). *It-security and privacy: design and use of privacy-enhancing security mechanisms.* Springer-Verlag.

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

Garfinkel, S. L. (2015). The expanding world of digital forensics.

Guo, H., Jin, B., & Huang, D. (2011). Research and review on computer forensics. In *Forensics in telecommunications, information, and multimedia* (pp. 224–233). Springer.

Hossain, M. M., Fotouhi, M., & Hasan, R. (2015). Towards an analysis of security issues, challenges, and open problems in the internet of things. In *2015 ieee world congress on services* (pp. 21–28).

Hou, S., Uehara, T., Yiu, S., Hui, L. C., & Chow, K. (2011b). Privacy preserving multiple keyword search for confidential investigation of remote forensics. In *Multimedia information networking and security (mines), 2011 third international conference on* (pp. 595–599).

Hou, S., Uehara, T., Yiu, S.-M., Hui, L. C., & Chow, K. (2011a). Privacy preserving confidential forensic investigation for shared or remote servers. In *Intelligent information hiding and multimedia signal processing (iih-msp), 2011 seventh international conference on* (pp. 378–383).

Inspectorate, G. S. (2015). *Changing policing in ireland.* November.

Karabiyik, U., & Aggarwal, S. (2014). Audit: Automated disk investigation toolkit. *The Journal of Digital Forensics, Security and Law: JDFSL*, *9*(2), 129.

Key, S. (acc. Mar'18). *Flat file export.* `https://www.guidancesoftware.com/app/flat-file-export`. (Accessed: 2018-03-10)

Law, F. Y., Chan, P. P., Yiu, S.-M., Chow, K.-P., Kwan, M. Y., Tse, H. K., & Lai, P. K. (2011). Protecting digital data privacy in computer forensic examination. In *Systematic approaches to digital forensic engineering (sadfe), 2011 ieee sixth international workshop on* (pp. 1–6).

Lillis, D., Becker, B., O'Sullivan, T., & Scanlon, M. (2016). Current challenges and future research areas for digital forensic investigation. *arXiv preprint arXiv:1604.03850*.

Neuner, S., Mulazzani, M., Schrittwieser, S., & Weippl, E. (2015). Gradually improving the forensic process. In *Availability, reliability and security (ares), 2015 10th international conference on* (pp. 404–410).

OECD. (2002). *Oecd guidelines on the protection of privacy and transborder flows of personal data.* OECD Publishing.

Oriwoh, E., Jazani, D., Epiphaniou, G., & Sant, P. (2013). Internet of things forensics: Challenges and approaches. In *Collaborative computing: Networking, applications and worksharing (collaboratecom), 2013 9th international conference conference on* (pp. 608–615).

Palmer, G., et al. (2001). A road map for digital forensic research. In *First digital forensic research workshop, utica, new york* (pp. 27–30).

Pew-Research. (2017). Mobile fact sheet [Blog]. *Pew Research Center: Internet, Science & Tech*(January 12). `http://www.pewinternet.org/fact-sheet/mobile/`. (Accessed: 2018-01-14)

Pollitt, M. M. (2004). A brief history of computer forensics. *Unpublished manuscript*.

Quick, D., & Choo, K.-K. R. (2014). Impacts of increasing volume of digital forensic data: A survey and future research challenges. *Digital Investigation*, *11*(4), 273–294.

Richard III, G. G., & Roussev, V. (2006). Next-generation digital forensics. *Communications of the ACM*, *49*(2), 76–80.

Rogers, M. (1999). Psychology of computer criminals. In *annual computer security institute conference, st. louis, missouri.*

Rogers, M. K. (2011). The psyche of cybercriminals: A psycho-social perspective. In *Cybercrimes: A multidisciplinary analysis* (pp. 217–235). Springer.

Rogers, M. K., Seigfried, K., & Tidke, K. (2006). Self-reported computer criminal behavior: A psychological analysis. *digital investigation*, *3*, 116–120.

Scanlon, M. (2016). Battling the digital forensic backlog through data deduplication. *arXiv preprint arXiv:1610.00248*.

Seo, K., Lim, K., Choi, J., Chang, K., & Lee, S. (2009). Detecting similar files based on hash and statistical analysis for digital forensic investigation. In *2009 2nd international conference on computer science and its applications, csa 2009.*

Shebaro, B., & Crandall, J. R. (2011). Privacy-preserving network flow recording. *digital investigation*, *8*, S90–S100.

Staden, W. v. (2013). Protecting third party privacy in digital forensic investigations. In *Advances in digital forensics ix* (pp. 19–31). Springer.

Van Baar, R., van Beek, H., & van Eijk, E. (2014). Digital forensics as a service: A game changer. *Digital Investigation*, *11*, S54–S62.

Verma, R., Govindaraj, J., & Gupta, G. (2014). Preserving dates and timestamps for incident handling in android smartphones. In *Ifip international conference on digital forensics* (pp. 209–225).

Verma, R., Govindaraj, J., & Gupta, G. (2016). Data privacy perceptions about digital forensic investigations in india. In *Ifip international conference on digital forensics* (pp. 25–45).

Verma, R., Gupta, A., Sarkar, A., & Gupta, G. (2012, December). *Forensically important artifacts resulting from usage of cloud client services.* Presented as a Case Study at 2012 Annual Computer Security Applications Conference, Orlando, Florida, USA. `https://www.acsac.org/2012/program/case/Gupta.pdf`. (Accessed: 2018-01-14)

APPENDIX

A: The Hacking Case Questionnaire

1. What operating system was used on the computer?
2. When was the install date?
3. What is the time-zone settings?
4. Who is the registered owner?
5. What is the computer account name?
6. What is the primary domain name?
7. When was the last recorded computer shutdown date/time?
8. How many accounts are recorded (total number)?
9. What is the account name of the user who mostly uses the computer?
10. Who was the last user to logon to the computer?
11. A search for the name of "Greg Schardt" reveals multiple hits. One of these proves that Greg Schardt is Mr. Evil and is also the administrator of this computer. What file is it? What software program does this file relate to?
12. List the network cards used by this computer. This same file reports the IP address and MAC address of the computer. What are they?
13. Find some 'installed programs' that may be used for hacking.
14. What is the SMTP email address for Mr. Evil?
15. List some newsgroups that Mr. Evil has subscribed to?
16. A popular IRC (Internet Relay Chat) program called MIRC was installed. What are the user settings that was shown when the user was on-line and in a chat channel?
17. This IRC program has the capability to log chat sessions. List some IRC channels that the user of this computer accessed.
18. Ethereal, a popular "sniffing" program that can be used to intercept wired and wireless Internet packets was also found to be installed. When TCP packets are collected and re-assembled, the default save directory is that users 'My Documents' directory. What is the name of the file that contains the intercepted data?
19. Viewing the file in a text format reveals much information about who and what was intercepted. What type of wireless computer was the victim (person who had his Internet surfing recorded) using?
20. How many files are actually reported to be deleted by the file system?