Methodology for the Automated Metadata-Based Classification of Incriminating Digital Forensic Artefacts

Xiaoyu Du*
xiaoyu.du@ucdconnect.ie
University College Dublin
Dublin, Ireland

Mark Scanlon*
mark.scanlon@ucd.ie
University College Dublin
Dublin, Ireland

ABSTRACT

The ever increasing volume of data in digital forensic investigation is one of the most discussed challenges in the field. Usually, most of the file artefacts on seized devices are not pertinent to the investigation. Manually retrieving suspicious files relevant to the investigation is akin to finding a needle in a haystack. In this paper, a methodology for the automatic prioritisation of suspicious file artefacts (i.e., file artefacts that are pertinent to the investigation) is proposed to reduce the manual analysis effort required. This methodology is designed to work in a human-in-the-loop fashion. In other words, it predicts/recommends that an artefact is likely to be suspicious rather than giving the final analysis result. A supervised machine learning approach is employed, which leverages the recorded results of previously processed cases. The process of features extraction, dataset generation, training and evaluation are presented in this paper. In addition, a toolkit for data extraction from disk images is outlined, which enables this method to be integrated with the conventional investigation process and work in an automated fashion.

KEYWORDS

Digital Forensics, Automatic Forensic Investigation, Artefact Relevancy, Machine Learning

ACM Reference Format:
Xiaoyu Du and Mark Scanlon. 2019. Methodology for the Automated Metadata-Based Classification of Incriminating Digital Forensic Artefacts. In *Proceedings of the 14th International Conference on Availability, Reliability and Security (ARES '19), August 26–29, 2019, Canterbury, United Kingdom.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3339252.3340517

1 INTRODUCTION

"Big digital forensic data" is a challenge faced by law enforcement agencies around the world as the prevalence of digital devices ever increases [19]. The volume of data requiring analysis during digital forensic investigations is ever increasing. How to quickly detect pertinent file artefacts is a problem calling to be solved. A variety of solutions such as automation, data reduction, centralised digital evidence processing, data deduplication, triage, etc., have been proposed and developed in recent years.

Beebe et al. [2] outlined how the automation of forensic processes is important. Numerous automation tools have been developed by researchers in the field. However, independent automatic tools are showing very little impact on the efficiency of the investigative process. As the formats used by these tools vary, this leads to the issue that investigators are left to manually figure out the clues from the analyses results [3]. Merging the disparate formatted results from various tools is also overly arduous and time consuming [5]. The Cyber-Investigation Analysis Standard Expression (CASE)[1], a community-developed standard format, is attempting to bridge this gap.

For reducing the digital forensic backlogs, the first operational Digital Forensics as a Service (DFaaS) system, Hansken, was implemented by Netherlands Forensics Institute (NFI) in 2014 [26, 27]. DFaaS is a cloud based solution, analysing forensic data and sharing the results on a centralised sever. It enables easier information sharing between investigators and detectives, which is one manner that improves overall efficiency over traditional systems. Based on the DFaaS paradigm, data deduplication was proposed for reducing the time repeatedly acquiring and analysing known file artefacts [25]. Several experiments have been performed that proves the increased efficiency created by a deduplication system [8, 21, 28]. Creation of centralised artefact whitelisting eliminates the known, benign operating system and application files, and also eliminates known, benign user created files. In addition, known illegal file artefacts can be detected at an earliest stage possible, i.e., during the acquisition phase.

Machine learning approaches offer a data driven solution, which results in a more flexible solution compared with hard coded scripts. In recent years, libraries such as `scikit-learn` [2] facilitate simple and efficient data mining and data analysis. Researchers in many fields look to machine learning techniques to improve the effectiveness and efficiency of their solutions; the same is true in the digital forensics and cybersecurity domains. For example, machine learning has been employed in malware classification [18], establishing forensic analysis priorities [11], automated categorisation of digital media [20], etc.

The research presented in this paper aims to improve the automation of the digital forensic investigative process. To achieve this goal, a methodology is proposed to automatically determine suspicious/relevant file artefacts encountered during the investigation. Machine learning models were trained to perform this prediction

*UCD Forensics and Security Research Group - https://www.forensicsandsecurity.com

ARES '19, August 26–29, 2019, Canterbury, United Kingdom

ACM ISBN 978-1-4503-7164-3/19/08…$15.00
https://doi.org/10.1145/3339252.3340517

[1] https://caseontology.org/
[2] https://scikit-learn.org/

task. A DFaaS system saved the expert human artefact classification, i.e., illegal or benign, on the server, which provides training data for the models. The experiments presented in this paper prove the viability of this methodology.

1.1 Contribution of this Work

The contribution of this work can be summarised as:

- Design of an automatic digital forensic data processing approach to prioritise the investigation of suspicious file artefacts;
- Verification of the proposed methodology through an example scenario;
- Creating a toolkit for data extraction form disk images, associated dataset generation,and pre-processing. This enables the method to be performed automatically during the investigation;
- Analysis and discussion of the proposed solution in the field for accelerating the processing of large volumes of digital evidence.

2 RELATED WORK

Section 2.1 to 2.4 presents the existing methods and techniques on expediting the process of digital forensic investigation and combating the backlogs caused by the big data volumes. In addition, how these techniques can be combined during the investigation are discussed. In Section 2.5, the significant value of metadata and timeline analysis to digital forensic investigation is outlined, and associated tools and frameworks towards automatic metadata and timeline analysis are presented.

2.1 Automatic Analysis

Automating the investigative process is challenging. Garfinkel [10] pointed out that automation comes at great expense and has had limited success. The path followed in a digital forensic investigation can vary substantially due to the investigation's purpose, the type of case, and the variety of devices encountered. A number of process models have been defined, yet there is still no global standard procedure in practice. Hence, there is no existing, completely automated approach to conduct investigations.

Numerous automatic tools have been developed in the research and commercial fields that can assist the investigation process. These tools aim to only reduce part of manual work during evidence processing. Many automatic tools focus on data extraction. However, automatic artefact examination is more challenging.

Attempting to chain existing automated tools achieves very limited progress to achieve this goal. This is due to the tools requiring specific formats for input data, and the format of generated output data can be incompatible with the next desired tool in the chain. Automatic processing is needed to improve the efficiency and reduce inadvertent errors in digital forensic investigation [15]. In addition, as described in the introduction, the large volume of data leads to individual examiners being unable to fully understand it or use it effectively. One arduous task that investigators have to perform is connecting the dots [3].

Chen [5] outlines that merging analyses results from various tools into a single case report is difficult. Hence, the automation should be designed on a framework level, which means from the start to the end of the process, the tools work together automatically.

Efforts have been made toward a higher level of automatic digital evidence processing. CASE aims to aggregate the extracted information by various forensic tools in a standardised manner. `log2timeline`[3] is a tool that generates a "super timeline", which consists of digital events from the various files and logs discovered. `log2timeline` uses parsers to extract digital events from a variety of file formats (e.g., NTFS `$MFT`, Microsoft IIS log files, SQLite databases, Firefox Cache, Chrome preferences, etc.). However, existing analysis plug-ins provide very limited options.

2.2 DFaaS and Data Deduplication

The term *Big Data Forensics* was proposed as a new branch of digital forensics by Zawoad et al. [29]. In the age of Big Data, there are both new challenges and opportunities for digital forensics. Lillis et al. [19] outlined the challenges faced by investigators; and one of the most impactful is the ever increasing data volumes requiring forensic analysis. Significant potential benefits for digital evidence processing can be seen from a number of techniques such as data mining, machine learning, and big data analytics.

In 2014, the Hansken DFaaS system was described by the NFI [26]. The system provides a service that processes high volumes of digital material in a forensic context. This system had processed over a petabyte of data, as of 2015 [27]. As the benefits outlined in [27], DFaaS improves the efficiency of the investigation through offering better resource management, collaboration and sharing knowledge, etc.

One question to be noted is how DFaaS can work with the existing forensic process models. The DFaaS system from the NFI is based on the *Integrated Digital Forensic Process Model* [17]. In 2017, Du et al. [7] discussed the evolution of digital forensic process models, as well as an overview of the benefits of DFaaS to existing process models. DFaaS is no longer a new paradigm and is compatible with traditional evidence processing frameworks.

Employing data deduplication techniques to combat the big data volume and expedite digital evidence processing has been discussed in the literature [25, 28, 29]. For applying deduplication techniques, a centralised database recording the known file artefacts is required. This technique can work on top of the DFaaS framework. The employment of data deduplication techniques effectively reduces the data required to be processed during the investigation, and potentially blacklisting enables a faster illegal file artefact detection [21, 25]. Experimentation has proven the significant savings of the storage and the volumes of data processed in [21] and [8].

2.3 Prioritised Analysis

Even with the large number of devices and data involved in modern investigations, the forensic value of each artefact is not equal. Conducting comprehensive analysis on all seized digital devices is overly time consuming. How to quickly determine which data has more pertinent value for manual examination is an open research question in existing literature.

Triage is a term originally used in medical field; in digital forensics, triage ranks seized digital devices in terms of importance or priority [23]. Digital forensic triage is a method usually used for getting faster responses to an incident for time-sensitive cases, such

[3] https://github.com/log2timeline/plaso/wiki

as child abuse, kidnapping, and terrorist threats [14]. When the volume of seized material is very large and only few devices might be considered relevant for the investigation, how to quickly determine which are the devices with the higher forensic value is paramount.

Prioritised analysis on file artefacts has become necessary in a variety of cases. As the storage of each device is increasing, the number of file artefacts on a device could be huge. Manual examination of each file artefact could take a long time. Conventional methods rely on the hash-based filtering and keyword indexing to search for relevant file artefacts. In recent years, researchers have proposed metadata-based clustering methods for grouping similar file artefacts to assist the investigators. Document clustering has also been used for forensic analysis [6].

2.4 Machine Learning in Digital Forensics

Machine learning-based techniques have been widely applied across diverse fields. Deep learning, as another subcategory of artificial intelligence, really shines when dealing with complex problems such as image classification, nature language processing, and speech recognition. A number of methods have been proposed to develop intelligent methods for problem solving in digital forensics.

An approach for computer user identification was presented by Grillo et al. in order to quickly classify seized devices [11]. The proposed method for identifying the individual computer user leveraged the user's habits, computer skill level, online interests, etc. User profiling resulted in five categories being identified: occasional users, Internet chat users, office worker users, experienced users, and hacker users. This approach prioritised the analysis of seized hard drives; forensic examiners could examine only potentially relevant hard drive images resulting in a reduced analysis time.

Triage in forensic investigation has been discussed in Section 2.3 and using machine learning technique to conduct triage on seized devices was discussed. Marturana et al. [20] presented a triage method for the categorisation of digital media. The features extracted from the seized devices were based on the connections to specific crimes under investigation. Two use cases were presented in this paper; one involved copyright infringement, while the other involved child sexual abuse investigation. The crime-related features were defined by the researchers, which resulted in models being relevant for the case categories defined, but may not be applicable to real cases directly.

2.5 Metadata Forensics and Timeline Examination

Typical questions often asked in digital forensic investigation include: when, where, what, why, and how the incident happened [4]. Metadata refers to data about data and it usually plays significant role in digital forensic investigation. The precise information contained in metadata varies depending on the file system and file type. Generally, metadata consists of file modification time stamps, file ownership information, and data units allocated to this metadata unit [1]. In digital forensic investigation, file system metadata is typically used for 1) keyword searching to find out a specific type of file artefact, e.g., based on the file type; 2) filtering the file artefacts based on file size, creation time, and so on [10].

In recent years, metadata has also been applied for automatic forensic investigation. Rowe and Garfinkel [24] outlined a methodology that uses directory metadata (file names, extensions, paths, size, access/modification times, fragmentation, status flags and hash codes) on a large corpus to find out anomalous and suspicious file artefacts. In 2013, Raghavan and Raghavan [22] demonstrated the use of metadata associations to determine the origin of downloaded files.

Digital forensic analysis attempts to reconstruct the events that have occurred on the seized device(s) pertaining to a case. Creating a timeline of system activity is capable of assisting the investigators in the discovery of useful traces for the case. Inglot et al. claimed that there is a need for a comprehensive timeline analysis tool [16].

Hargreaves et al. [13] proposed an automated timeline reconstruction approach for generating high-level events, which reduced the number of events to be analysed. One problem highlighted by these authors is that separate neural networks are required to be trained for different applications and a new data set of file system activity would be required for newer versions of applications. More general purpose analysers are needed for digital event correlation.

`log2timeline and plaso` [4] [12] appears to be the tool of choice for timeline generation and analysis over recent years. It is a tool designed to extract timestamps (of digital events) from various of log files found on a typical computer system and aggregate them. `log2timeline` analysers conduct the preliminary discovery of the digital events contained on a device. However, analysers generating useful information through digital events correlation must be developed to enable an automated evidence analysis process.

3 METHODOLOGY

Traditional, hash-based white-listing or blacklisting methods are usually the go-to automatic solution for finding known file artefacts during a digital forensic investigation. If nothing pertinent has been detected, the investigators will have to manually perform keyword search or filter to examine the artefacts. Using hash matching to detect illegal files can only detect the precisely same file artefacts or artefacts with a minor change (i.e., by using approximate matching).

This research presents an approach that enables automatic determination of suspicious file artefacts by training machine learning models. The associated metadata and digital events occurred are employed to extracting features of each file artefact. Combining this with a centralised, deduplicated digital evidence processing framework, illegal file artefacts encountered in previous cases are labelled as such in the database. These files on the blacklist can be used to train classifiers for detecting previous unencountered suspicious file artefacts in new cases.

3.1 The Relevancy Determination Process

Figure 1 illustrates the designed work-flow of this methodology as could be applied in an investigation. From a raw format image copy to the prediction result, the process is completely automated. The processing is broken down in four steps:

[4] https://github.com/log2timeline/plaso/wiki

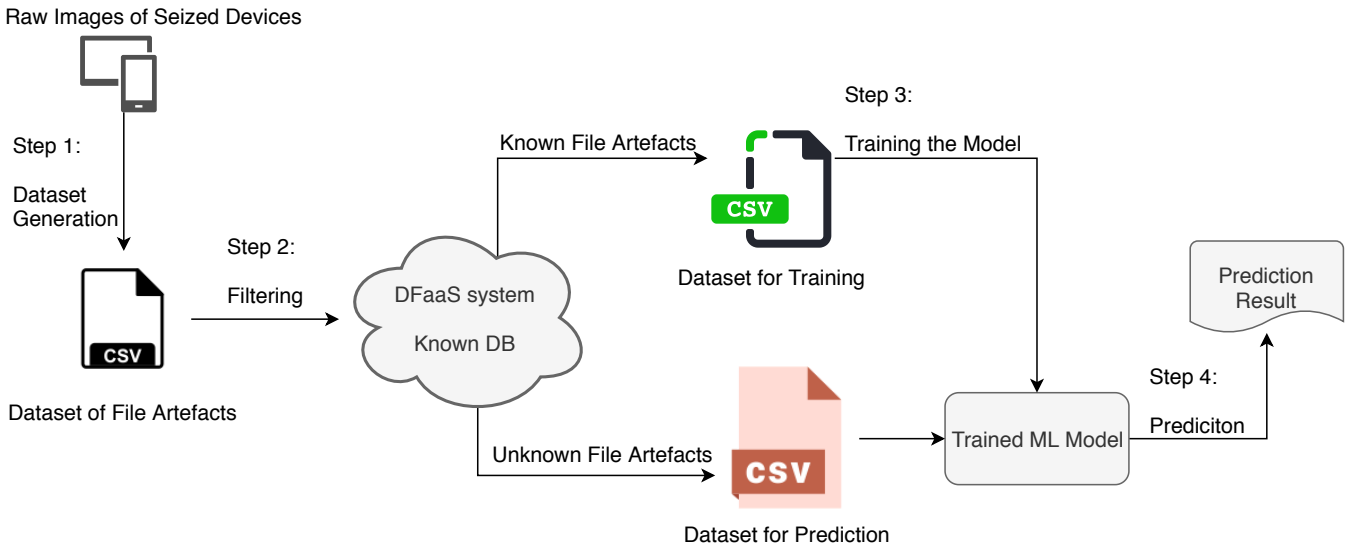

Figure 1: Overview of Methodology: the work-flow of the proposed methodology in an investigation.

(1) **Dataset Generation** - The developed toolkit is used to generate the training dataset in *csv* format for machine learning-based modelling. The details on how this dataset is generated is presented in Section 3.2.
(2) **Filtering** - It works through comparing the hash with the known database, which will split the file artefacts into two categories; known file artefacts and unknown file artefacts.
(3) **Training the Model** - Training a model using known file artefacts. The features used will be described in Section 3.4.
(4) **Prediction** - The categorisation of the previously unencountered files are predicted by the trained model (i.e., if it is relevant or suspicious to the investigation).

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

3.2 Toolkit for Data Extraction and Processing

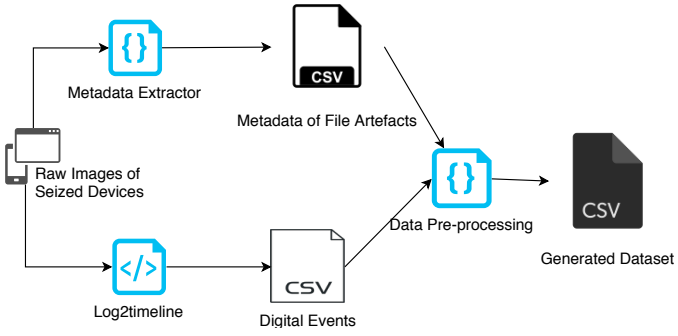

Figure 2: Toolkit for Data Extraction and Processing

Figure 2 shows how the dataset for training is generated from a raw disk image. There are three parts at this stage: 1) metadata extraction using the python library `pytsk` [5]; 2) digital events extraction using `log2timeline`; 3) merging the metadata and digital events to a dataset for file artefacts. The tools for metadata and digital events extraction operate directly on raw disk image. The input is disk image in *raw* format. The output is a file in *csv* format.

The metadata extraction tool uses `pytsk`. `Pytsk` is a Python binding for the Sleuth Kit. In this research, it was used to extract the file system metadata. The timeline generated using `log2timeline` consists of the digital events from the file system, windows registry, browsing history, download history, etc. Each event describes source, the file name, event type, etc.

In the generated timeline, each row represents a digital event and the `inode` can identify which file artefact it is related to. The `inode` can be used to collate all of the digital events associated with each file artefact. If multiple partitions are involved, a combination of hash and `inode` must be used to keep this unique identifier. The combination of metadata and timeline results in an abundance of information related to each file artefacts. More details on the extracted data is presented in Section 3.3.

3.3 File System Metadata and Timeline

The importance of metadata analysis in an investigation process has been discussed in Section 2.5. The specific file system metadata considered useful and exploited in this research are:

- File name;
- File size;
- `inode`;
- File hash value (MD5/SHA1/SHA256);
- Physical address of file;
- The number of blocks of the file;
- The block number(s) allocated to the file;

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

- Creation time of the file;
- Last access time of the file;
- Last modification time of the file.

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

- Fields describing the event:
  - **date** - Date that the event occurred;
  - **time** - Time that the event occurred;
  - **MACB** - Modification, access, creation and birth times;
  - **desc** - A description of the timestamp object;
  - **short** - A shorter version description of the timestamp;
  - **filename** - The file object on which the event occurred;
  - **sourcetype** - Description of the source type, e.g., "Opera Browser History";
  - **source** - A shorted form of the source, e.g., "WEBHIST";
  - **type** -Timestamp type, e.g., "Last Time Executed".
- Fields describing the source:
  - **inode** - The inode or MFT number of the parsed artefacts;
  - **user** - The user owns the parsed artefacts;
  - **host** - The host that the data came from.
- Fields describing the tool used:
  - **version** - the version of the tool;
  - **timezone** - Timezone where applying the tool generating the timeline;
  - **format** - The parsing module.
- Fields describing other information:
  - **notes** - An operational field;
  - **extra** - A reference to a hash that stores all additional fields that might be used.

3.4 Feature Extraction from Metadata and Timeline

In training machine learning models, not every feature available proves valuable. For example, randomly generated numerical features, artefact hash values, or `inode` values are not helpful to the prediction task. Feature manipulation is usually needed for a specific task or purpose for each machine learning model. Trough feature transformation, the information input for model training can be added, changed or removed as desired for each task. In fact, the resultant model is a way of constructing a new feature that solves the task at hand [9].

Standalone metadata, such as filename and timestamp, are likely not suitable to use directly for training classification models. Table 1 lists the useful metadata and the corresponding features that can be extracted, manipulated, and transformed.

Table 1: Valuable Feature Extracted

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

3.5 Evaluation Matrix

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

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

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

$$precision = TP/(TP + FP);$$

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

$$recall = TP/(TP + FN);$$

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

$$F1\text{-}Score = 2*recall*precision.$$

3.6 Comparison with the Existing Methodology

Differentiating this approach from previous research on using metadata to cluster the file artefacts, this research leverages expert human analysis results from the manual processing previous investigations. The hypothesis that the features of analysed file artefacts can enable trained machine learning models to determine how relevant newly encountered file artefacts are to a specific type of investigation. The experiment presented in Section 4 explores this hypothesis.

One advantage of the proposed method is performing supervised machine learning tasks on the investigated file artefacts; the automated categorisation can more directly steer the investigator's focus towards pertinent data at the earliest possible stage. Lastly, this research presents a complete overview of how this method can be applied in real-world investigative scenario.

4 EXPERIMENTATION AND RESULTS

This section outlines the dataset generation, the structure of the dataset, the machine learning models trained and the performance of the results are evaluated.

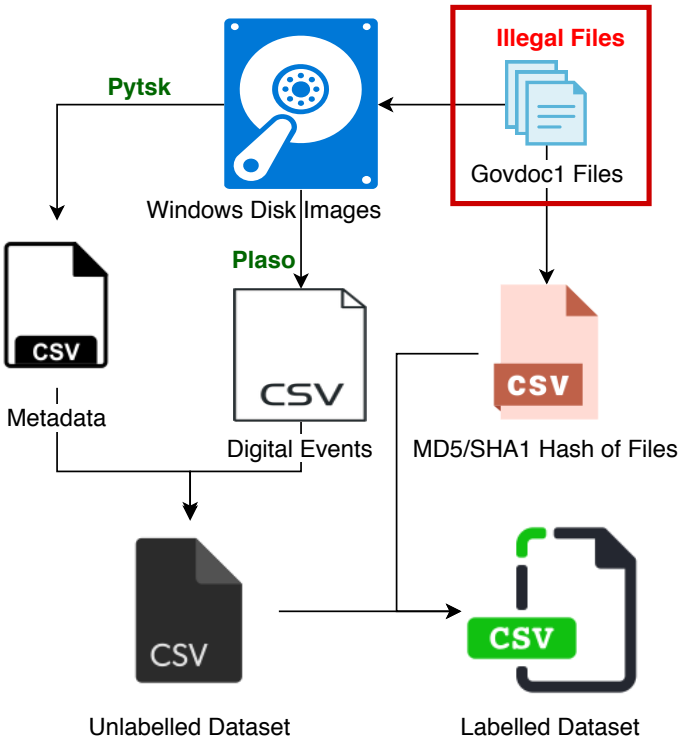

Figure 3: Dataset Generation - 1) Disk image creation; 2) Metadata and timeline generation; 3) Merging the two sources (each sample in the dataset represents a single file artefact); 4) Labelling the data (file artefacts from Govdoc1 Corpus are defined as "illegal").

4.1 Experimentation Dataset

In this section, the creation of dataset used in the experiment is outlined. As shown in figure 3, the process of experimental dataset generation includes:

(1) Emulating wear-and-tear of the device on a virtual environment and planting the "illegal" file artefacts on the test virtual machine;
(2) Using the developed tool to extract filesystem metadata and a "super timeline" from the disk image;
(3) Merging the information about file artefacts from two sources (i.e., the extracted timeline and metadata);
(4) Labelling the file artefacts on the dataset based on the hash of file artefacts.

The disk images used in this research are Windows 7 disk images. Emulated wear and tear actions (surfing the Internet, installing software, downloading files from browsers, etc.) were conducted in a virtual environment. The Govdocs1[6] dataset were defined as "known illegal" file artefacts (Govdocs1 consists of almost 1 million freely-redistributable files of various formats and sizes).

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

4.2 Example Scenario

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

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

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

4.3 Result Discussion

Several machine learning classification algorithms including Logistic Regression, k-NN, Support Vector Machine, Decision Tree, Gaussian Naïve Bayes are tested in the experimentation. The datasets used in the experiment were generated using the toolkit presented in Section 3.2. Table 2 shows the generated datasets used in this experiment; mainly differing on the percentage of illegal artefacts. The dataset is split into training and testing data.

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

Table 2: Datasets Used in the Experiment

Precision-Recall curves summarise the trade-off between the true positive rate and the positive predictive value for a predictive model using different probability thresholds. For getting an overall understanding of the models' performance, Precision-Recall curve (RP curve) is visualised and average precision score (AP score) is calculated. Figure 4 shows the RP curve of the employed algorithms on dataset1. Comparison of the average precision score, the performance of models from best to worst are Decision Tree, Gaussian Naïve Bayes, Support Vector Machine, k-NN, and Logistic Regression respectively.

As the practical usage of the trained model is for digital forensic investigation, more concern should be put on the classifying of

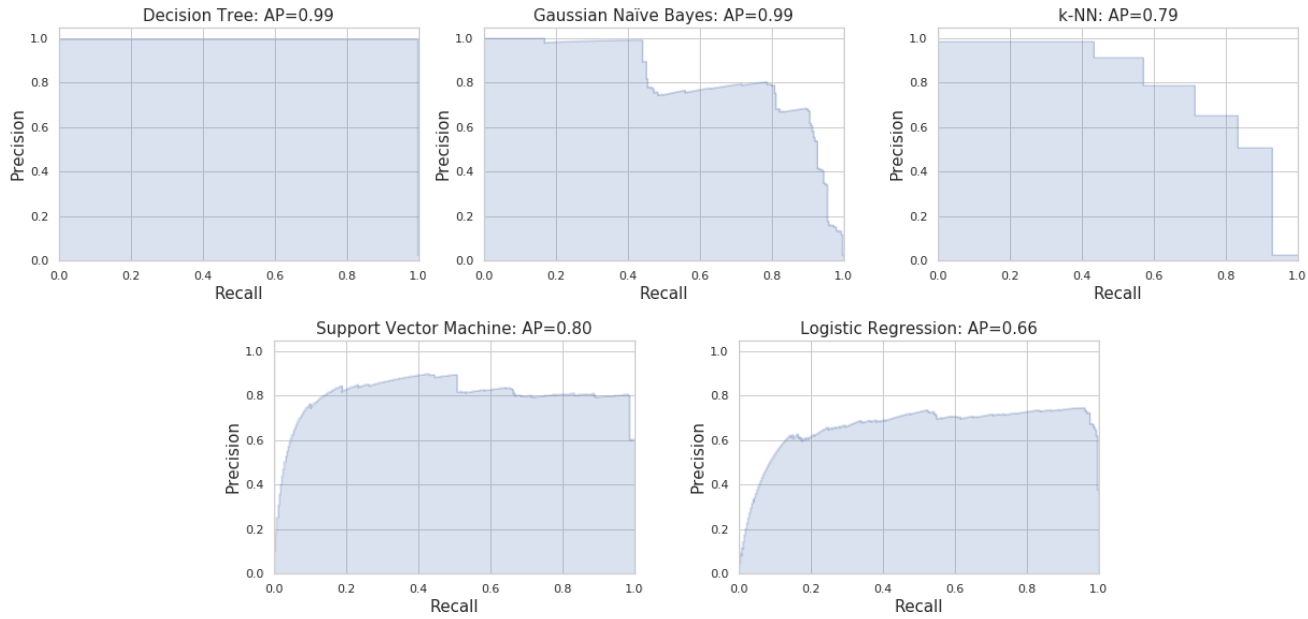

Figure 4: Precision-Recall Curves per Classifier with Corresponding Average Precision (AP) Scores

illegal files (class 1). Precisely, the precision, recall and F1 score on class 1, instead of the average value on class 0 and class 1, should be used to represents the performance of the models. The scores shown in Table 3 are on the class 1. This table shows an evaluation matrix from two datasets.

As shown in Table 3, it is obvious that the performance with `dataset2` is significantly better than `dataset1`. This happens is because the number of "illegal" file samples in `dataset2` is more than `dataset1`. The imbalanced class problem is apparent in these datasets. The datasets were created in this manner due to the assumption that only a subset of the "illegal" files are classified as known illegal/known benign from the centralised database.

An overview on the models' settings and performance analyses, as shown in Figure 4, is listed below:

- **Decision Tree** - Observing from the AP score and precision, recall and F1 scores, the Decision Tree classifier performs best for both datasets compared against other models. In addition, the performance is stable on the different datasets.
- **Gaussian Naïve Bayes** - The performance of Gaussian Naïve Bayes on `dataset1` is worse than `dataset2`. Even though the AP score is good, the scores on class 1 is much worse. For this severely imbalanced dataset, the score for the class with less samples is very low.
- **k-NN** - One important parameter for k-NN is the selection of k. In this experiment, the model achieved its best accuracy when k was set to be 5.
- **Support Vector Machine** - Through experimentation, the model realises a poor performance when the given data is not normalised. Hence, the dataset is normalised before it is fed into the model whose parameters are left default for the experiment.
- **Logistic Regression** requires quite large sample sizes, which could be the reason it scores lower comparing with the other models.

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

5 CONCLUSION

This paper outlines an automatic approach to use machine learning models to predict which file artefacts are likely pertinent to an investigation (i.e. which file artefacts are likely more suspicious than others). It is designed to integrate with a DFaaS framework, rather than a stand alone experiment on an individual device. The associated toolkit was introduced that was developed for supporting the automation of some the digital forensic process. An example scenario was outlined and tested indicating the feasibility and effectiveness of the proposed methodology. The promising experimental results for suspicious artefact detection provides motivation for further research.

5.1 Future Work

Future work in this ares will focus on:

- Generating more scenarios - This experimental scenario used in this paper was child abuse material possession and distribution investigation. The purpose of this case type is to find out files with similar size, directory, creation time, etc. More

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

Table 3: Evaluation Matrix of Different Classification Algorithms and Datasets

investigative scenarios will be designed and evaluated. For example, training models for determining suspicious files through each origin source (from email attachment, Cloud account, USB device, etc.), third party owner, etc.

- More exploration is required on the extent of imbalanced classes influences each model's performance. Form the conducted experiment in this paper, it was shown that the dataset can influence the performance of the models. In the future work, more diverse datasets will be generated for testing. The generation of these datasets should provide a broad variety on the ratios of the benign/illegal files.
- Exploring appropriate feature selection and feature engineering approaches for the defined machine learning tasks. In this paper, the features are selected by the common knowledge of digital forensic investigation. One avenue of exploration that could improve the performance is extracting as many features as possible initially, and subsequently selecting from them by using techniques such as *SelectBest*, *RFE (Recursive Feature Elimination)*, and *PCA (Principal Component Analysis)*.
- Tuning the model so that it can fit needs of real investigative scenarios better. For example, to focus on the recall scores, rather than insisting on a higher precision. Because, in any digital forensic investigation, any pertinent inculpatory or exculpatory file artefact can not be inadvertently overlooked.

REFERENCES

[1] Cory Altheide and Harlan Carvey. 2011. *Digital forensics with open source tools.* Elsevier.

[2] Nicole Beebe. 2009. Digital forensic research: The good, the bad and the unaddressed. In *IFIP International Conference on Digital Forensics*. Springer, 17–36.

[3] Andrew Case, Andrew Cristina, Lodovico Marziale, Golden G Richard, and Vassil Roussev. 2008. FACE: Automated digital evidence discovery and correlation. *Digital Investigation* 5 (2008), S65–S75.

[4] Eoghan Casey. 2011. *Digital evidence and computer crime: Forensic science, computers, and the internet.* Academic Press.

[5] Lei Chen, Hassan Takabi, and Nhien-An Le-Khac. 2019. *Security, Privacy, and Digital Forensics in the Cloud.* John Wiley & Sons.

[6] Luís Filipe da Cruz Nassif and Eduardo Raul Hruschka. 2013. Document clustering for forensic analysis: an approach for improving computer inspection. *IEEE Transactions on Information Forensics and Security* 8, 1 (2013), 46–54.

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

[8] Xiaoyu Du, Paul Ledwith, and Mark Scanlon. 2018. Deduplicated Disk Image Evidence Acquisition and Forensically-Sound Reconstruction. In *2018 17th IEEE International Conference on Trust, Security and Privacy in Computing and Communications/12th IEEE International Conference on Big Data Science and Engineering (TrustCom/BigDataSE)*. IEEE, 1674–1679.

[9] Peter Flach. 2012. *Machine learning: the art and science of algorithms that make sense of data.* Cambridge University Press.

[10] Simson L Garfinkel. 2010. Digital forensics research: The next 10 years. *Digital Investigation* 7 (2010), S64–S73.

[11] Antonio Grillo, Alessandro Lentini, Gianluigi Me, and Matteo Ottoni. 2009. Fast user classifying to establish forensic analysis priorities. In *IT Security Incident Management and IT Forensics, 2009. IMF'09. Fifth International Conference on*. IEEE, 69–77.

[12] Kristinn Guðjónsson. 2010. Mastering the super timeline with log2timeline. *SANS Institute* (2010).

[13] Christopher Hargreaves and Jonathan Patterson. 2012. An automated timeline reconstruction approach for digital forensic investigations. *Digital Investigation* 9 (2012), S69–S79.

[14] Ben Hitchcock, Nhien-An Le-Khac, and Mark Scanlon. 2016. Tiered forensic methodology model for Digital Field Triage by non-digital evidence specialists. *Digital Investigation* 16 (2016), S75–S85.

[15] Ronald In de Braekt, Nhien-An Le-Khac, Jason Farina, Mark Scanlon, and Mohand-Tahar Kechadi. 2016. Increasing Digital Investigator Availability through Efficient Workflow Management and Automation. (04 2016), 68–73.

[16] Bartosz Inglot, Lu Liu, and Nick Antonopoulos. 2012. A framework for enhanced timeline analysis in digital forensics. In *2012 IEEE International Conference on Green Computing and Communications*. IEEE, 253–256.

[17] Michael Donovan Kohn, Mariki M Eloff, and Jan HP Eloff. 2013. Integrated digital forensic process model. *Computers & Security* 38 (2013), 103–115.

[18] Quan Le, Oisín Boydell, Brian Mac Namee, and Mark Scanlon. 2018. Deep learning at the shallow end: Malware classification for non-domain experts. *Digital Investigation* 26 (2018), S118–S126.

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

[20] Fabio Marturana and Simone Tacconi. 2013. A Machine Learning-based Triage methodology for automated categorization of digital media. *Digital Investigation* 10, 2 (2013), 193–204.

[21] Sebastian Neuner, Martin Mulazzani, Sebastian Schrittwieser, and Edgar Weippl. 2015. Gradually improving the forensic process. In *2015 10th International Conference on Availability, Reliability and Security*. IEEE, 404–410.

[22] Sriram Raghavan and SV Raghavan. 2013. Determining the origin of downloaded files using metadata associations. *Journal of Communications* 8, 12 (2013), 902–910.

[23] Marcus K Rogers, James Goldman, Rick Mislan, Timothy Wedge, and Steve Debrota. 2006. Computer forensics field triage process model. *Journal of Digital Forensics, Security and Law* 1, 2 (2006), 2.

[24] Neil C. Rowe and Simson L. Garfinkel. 2012. Finding Anomalous and Suspicious Files from Directory Metadata on a Large Corpus. In *Digital Forensics and Cyber Crime*, Pavel Gladyshev and Marcus K. Rogers (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 115–130.

[25] Mark Scanlon. 2016. Battling the Digital Forensic Backlog through Data Deduplication. In *Proceedings of the 6th IEEE International Conference on Innovative Computing Technologies (INTECH 2016)*. IEEE, Dublin, Ireland.

[26] RB Van Baar, HMA Van Beek, and EJ van Eijk. 2014. Digital Forensics as a Service: A game changer. *Digital Investigation* 11 (2014), S54–S62.

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

[28] Kathryn Watkins, Mike McWhorte, Jeff Long, and Bill Hill. 2009. Teleporter: An analytically and forensically sound duplicate transfer system. *Digital Investigation* 6 (2009), S43–S47.

[29] Shams Zawoad and Ragib Hasan. 2015. Digital forensics in the age of big data: Challenges, approaches, and opportunities. In *2015 IEEE 17th International Conference on High Performance Computing and Communications, 2015 IEEE 7th International Symposium on Cyberspace Safety and Security, and 2015 IEEE 12th International Conference on Embedded Software and Systems*. IEEE, 1320–1325.