Evidence Identification in Heterogeneous Data Using Clustering

Hussam Mohammed[1]
Centre for Security, Communications and Network Research
University of Plymouth
hussam.mohammed@plymouth.ac.uk

Nathan Clarke[2]
Centre for Security, Communications and Network Research
University of Plymouth
N.Clarke@plymouth.ac.uk

Fudong Li[3]
Centre for Security, Communications and Network Research
University of Plymouth
fudong.li@port.ac.uk

ABSTRACT

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

KEYWORDS

Digital forensics, heterogeneous data, clustering algorithms, FCM, K-means.

1 INTRODUCTION

Digital forensics has become an important tool in the fight against cyber and computer-assisted crimes. Recently, due to the increasing volume of data and the heterogeneity of digital evidence, more time and effort are required to conduct digital forensic examinations [1]. The large amount of data has a direct impact on investigators as they find themselves having to examine and analyze more files per case [2]. In addition, due to the heterogeneity of the evidence, cases may contain multi-resources and applications, posing difficulties for investigators to find evidence across all these devices in a consistent manner, without placing a significant cognitive load on the investigator [3].

With the significant increase in computing, individuals have increasingly become to own several devices (e.g. PC, laptop, tablet, and Smartphone) with each using different applications across various platforms [4]. Additionally, companies producing electronic devices need to choose an operating system (OS) either open source or commercial for their core technology [5]. Consequently, the file structure will be formatted according to the OS, resulting in a variety of file systems such as NTFS, FAT, HFS, and Ext4 [6]. The applications that sit on top of these OSs also operate across platforms with similar purposes. For example, platforms can have several web browsers (e.g. Google Chrome, Mozilla Firefox, and Apple's Safari), and messaging (e.g. SMS, Viber, and WhatsApp) both within a device and across devices. Being able to examine and analyze data from across many systems and applications based on the category of data at the same time is currently impossible [7]. A wide range of tools and techniques both commercially or via open source (including Encase, AccessData FTK, and Autopsy) have been developed to investigate cyber and computer-assisted crimes [8]. However, the forensic examination and analysis are further complicated as most existing tools are designed to work with a single device (e.g. a workstation and/or a smartphone) and a relatively small volume of data [9]. Indeed, tools are already struggling to deal with individual cybercrime cases that have a large size of evidence (e.g. between 200 Gigabyte and 2 Terabyte of data) [7].

Several methods have utilized to overcome these issues to find the evidential artifacts in an automated way such as unsupervised machine learning algorithms (e.g. Clustering algorithms) [10]. Clustering algorithms group data into clusters containing objects sharing common characteristics [11]. The algorithms divide the data without any prior knowledge about it. This precisely exists within forensic cases containing data which are not labelled. Therefore, there is a need for intelligence to reduce the volume of data to an acceptable level – where acceptable would be defined as identifying all artifacts of interest and leaving behind all benign files. This can lead to group only the suspicious data thereby minimize the burden upon the investigators. However, it is difficult to apply clustering algorithms on files directly and therefore metadata categories can be used instead [12]. Data categories, including databases, documents, pictures, and web browsers, hold valuable information

[1] University of Anbar/ Ramadi, Anabr, Iraq.
[2] Security Research Institute/Edith Cowan University/Perth, Western Australia, Australia.
[3] School of Computing, University of Portsmouth

that can be used to answer some of the questions of a forensic investigation. Examples of the questions include, who did what to a file, when they did it and where it was carried out. To this end, this paper proposed and developed a technique to identify the evidential artifacts by grouping them togethers using FCM and K-means algorithms. Building upon the authors prior work on harmonizing heterogeneous resources [13], the paper presents a series of experiments to empirically explore the viability of clustering of heterogeneous data across a number of cases.

The remainder of the paper is structured as follows: Section 2 presents a literature review of the existing research which uses metadata and clustering methods in forensic investigations to identify the evidence. Section 3 illustrates the datasets and research methodology deployed in the proposed experiment. Section 4 shows the experimental results of the clustering approach and Section 5 presents a comprehensive discussion of the proposed technique. The conclusion and future works are highlighted in Section 6.

2 RELATED WORK

Several of the published studies in the field of digital forensic analysis have focused on the use of metadata to reconstruct past events. These studies have utilized metadata to achieve a particular purpose such as data reduction [13], data clustering [14], and identification of evidential artifacts [15]. Regarding to data reduction using metadata, Rowe and Garfinkel [13] developed a tool (called Dirim) to automatically determine anomalous or suspicious files in a large corpus by analyzing the directory metadata of files (e.g. the filename, extensions, paths and size) via a comparison of predefined semantic groups and comparison between file clusters. Their experiment was conducted on a corpus consisting of 1,467 drive images with 8,673,012 files. The Dirim approach found 6,983 suspicious files based on their extensions. However, the approach analyses the data in each drive individually which leads to repeating the process multiple times. Similarly, [16] proposed an approach to create a centralized hashed library of benign files based. The approach utilized nine automated methods such as path, filename, timestamp, hash value, unusually busy weeks for a corpus, file size, directories containing mostly-known files, known uninteresting directories, and extensions. By using the combination of methods, a total of 8.4 million hash values of uninteresting files were created and the hashes could be used for different cases. By using an 83.8-million-file international corpus, 54.7% of files were eliminated as they were matched with two of the nine selected methods. In addition, false negative and false positive rates of their approach were 0.1% and 19% respectively. However, the remaining volume of data is still large which requires additional approaches to make it as workable for investigators as possible.

In [14], they proposed a forensic analysis approach for computer systems through the application of clustering algorithms to discover useful information in documents. The approach consists of two phases: a pre-processing step (which is used for reducing dimensionality) and running clustering algorithms (i.e. K-means, K-medoids, Single Link, Complete Link, and Average Link). The approach was evaluated by using five different datasets seized from computers in real-world investigations. According to the results, both of the Average Link and Complete Link algorithms gave the best results in determining relevant or irrelevant documents; whilst K-means and K-medoids algorithms presented good results when there is suitable initialization. However, the computational costs of hierarchical algorithms (i.e. Average Link and Complete Link) are usually high. Therefore, they are not suitable for datasets with a large volume of data. From a similar perspective, [17] carried out an examination for clustering digital forensics text string search output. Four clustering techniques were evaluated, including K-Means, Kohonen Self-Organizing Map (SOM), LDA followed by K-Means, and LDA followed by SOM. This study utilized more than two million search hits which were found in approximately 50,000 allocated files and unallocated blocks. The results showed that LDA followed by K-means obtained the best performance: more than 6,000 relevant search hits were retrieved after reviewing less than 0.5% of the search hit result. In addition, when performed individually, both K-Means and SOM algorithms, gave a poorer performance than when they were combined with LDA. However, this evaluation was achieved with only one synthetic case, which was small in size comparing with real-world cases.

With the aim to find the evidential artifacts in an automated way, [15] proposed an automated approach for identifying the evidence and speeding up the analysis process for computer forensics. Their approach consisted of three general steps: metadata extraction, clustering and automated evidence identification. Real forensic datasets were utilized to evaluate their approach, and four file metadata categories (i.e. File system, Email, EXIF and Internet history) were chosen and extracted individually. They then used unsupervised pattern recognition to cluster evidential artifacts to aid the investigators to focus on the evidential files thereby saving their time and efforts. The SOM was utilized for automatically grouping the input data without any supervision. The investigator determined the number of clusters before the process starts. Afterward, the Automated Evidence Profiler (AEP) algorithm was applied to analyze and identity the related artifacts across all metadata SOMs. The AEP contain two steps: first is to identify the first cluster based on prior work achieved in profiling criminal behavior; the second step is to identify subsequent clusters using the timeline analysis of each file in the first cluster. Their experiment was conducted by using four forensic cases, where each case includes a single forensics image. The experiment based on clustering has shown that 93.5% of interesting artifacts were grouped in the top five clusters. While the AEP algorithm has presented acceptable results and shown that the algorithm can reduce the investigator's time to analyze the cases and present the relevant evidence in a report. However, their approach was only applied to single images with a limited number of metadata categories.

In terms of dealing with heterogenous resources, [18] proposed a framework to analyze heterogenous data using K-means algorithm. The framework tried to uniform the format of all datasets to identify the hidden features. This process, as they revealed, makes the datasets homogenous and easy to analyze. For instance, the process converts the files such as pdf, ppt, and text files to text files and then

to csv files. Afterwards, they applied the K-means algorithm on these files. Although the framework was applied on real life heterogeneous datasets, there was no clear results illustrated. In addition, the process of dataset's uniform was not explained with various datasets to know its accuracy.

As demonstrated above, a number of technologies, such as data clustering and data reduction, have the potential capacity to save digital investigators' time and effort, were examined. Data clustering techniques have been widely used to eliminate uninteresting files and thus speed up the investigation process by determining relevant information more quickly. However, these studies have some limitations in find the evidential data especially with heterogeneous data.

3 MATERIALS AND METHODS

3.1 Datasets

Although the need of accessing real forensic data is essential to make the entire experiment more reliable, the availability of real forensic datasets is limited especially containing heterogeneous data. However, to examine the ability of grouping evidential artifacts using the proposed approach, three forensics cases (two privates and one public) were identified. The cases contained images from multiple resources such as smartphones, computers, and external hard drives. The reasons for using both public and private cases were due to the limited number of real forensic cases. The public case (Case 1) was generated by the National Institute of Standards and Technology (NIST)[20]. This case is an artificial case describing the scenario of a suspected person who tried to leak sensitive information related to the newest technology in his company. The other cases were obtained from Iraq, and contain information of crimes committed by convicted criminals. Table1 illustrated the details of these cases.

Table 1: Case Details

Case ID	Source ID	Image Size	Total Artifacts	Notable	Total Notables
1	1	20 GB	143180	151	177
	2	4 GB	1085	11	
	3	700 MB	867	15	
2	1	42.8 GB	24669	545	1638
	2	40.8 GB	7274	1093	
3	1	30 GB	260914	501	505
	2	2 GB	324	4	

Table 2: Overview of Experimental Datasets

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

During the metadata extraction phase, various metadata were obtained from these resources as illustrated in Table 2 such as file systems and applications. A number of fields within metadata categories contains missing features because they were extracted from devices or applications which do not support these features. For instance, the EXIF metadata, which was extracted from smartphone datasets, has completed metadata features such as filename, timestamp, camera manufacturer and model, size of image file, size of the image (width x height), IOS, latitude, longitude, and GPS timestamp. The EXIF metadata within computer datasets, however, contained missing features such as IOS, latitude, longitude, and GPS timestamp. Similarly, the internet browsing metadata is differentiated across the forensic images based on platforms and applications. In the computer images, there were two browsers (IE and Chrome) which have features such as URL, visit count, visit timestamp, referrer URL, title, and profile. Whereas the smartphone browsers only have (URL, visit count, visit timestamp). The smartphones images contain SMS and Viber application, and both of them serve to send and receive messages. Many features between SMS and Viber are similar such as account number, sending timestamp, delivery timestamp, message body, status, seen, and recipient number; as well as they contain binary-based data such as opened, deleted, seen, etc.

3.2 Merging Datasets

In order to overcome the heterogeneity issues, a prior work by the authors [13] was applied. This section will briefly describe the process in order to aid understanding of the datasets used in the experiment.

The characterization process is achieved by using a rule-based system with a high level of fundamental conditions and rules. Rule-based systems are a method that is used to manipulate the knowledge to interpret information in a useful manner [21]. An analysis of the metadata showed there was a limited number of the

fundamental conditions utilized such as string, consistency, numerical, Boolean, and timestamp. The characterization algorithm uses these rules and conditions which contain all the appropriate knowledge for matching similar categories. The final output of the characterization process is a record that contains all similar metadata categories as shown in Fig. 1.

Dataset 1 (D1)

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

Dataset 2 (D2)

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

Record File

Database No.	Column No.	Type	Database No.	Column No.	Type	Database No.	Column No.	Type	Database No.	Column No.	Type	Database No.	Column No.	Type
D1	C1	String (S.S)	D1	C3	Timestamp	D1	C4	String (SN)	D1	C2	String (S'S.S)	D2	C3	String (NzN)
D2	C1	String (S.S)	D2	C2	Timestamp	D2	C4	String (SN)	-	-	-	-	-	-

Figure 1: Characterization Process (D: Dataset; C: Column)

The harmonization process is utilized to merge the similar categories based on the characterization record. It can adjust the differences and inconsistencies among different measurements, methods, procedures, schedules, specifications, or systems to make them uniform or mutually compatible. Many fields within the metadata categories are stored in various forms across heterogeneous systems (i.e. timestamp, phone number, and file size). Fig. 2 shows the output of this process.

Output Dataset

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

Figure 2: Harmonization Process

3.3 Experimental Set-up

Clustering is the most powerful method for analyzing the data which can divide a dataset into a number of distinguished groups [10]. However, clustering algorithms generally have no internal way to handle textual data and missing values. Instead, a common solution is to represent each string feature by a numerical value and fill-in the missing values in a pre-processing step. Consequently, the traditional way for numerating leads to the two main problems: huge dimensionality and sparse distribution. While the filled-in values are inherently less reliable than the observed data. However, this paper proposes an approach to solve these problems as shown in Fig. 3.

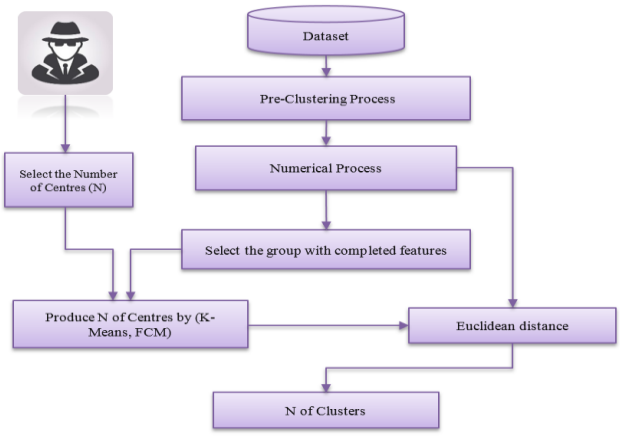

Figure 3: Clustering Process

3.3.1 The Pre-clustering process is to split up the dataset vectors into groups that are filled-in similar features. This leads to identify the group containing vectors with completed features.

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

1. The extra spaces from the strings are removed.
2. Extra Spaces are added to the end of the string containing fewer characters to make the length of two strings equal
3. The circular shift operation will be applied on one of these values in order to obtain all string probabilities as a tuple and produce several strings in order to match them with another string. The circular shift is a special kind of cyclic permutation, which in turn is a special kind of permutation. Formally, a

circular shift is a permutation X of n characters in the tuple such that:

$$X(i) = X(n - i) \quad (1)$$

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

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

$$Score(i) = \sum_{j=1}^{j=n} \frac{(tj - sj)}{n} \quad (2)$$

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

5. If the maximum score is greater than 0.7, the target string is given a numerical value as following:

$$Sn = Nq + (1 - MaxScore) \quad (3)$$

Where Sn is the numerical value of target string, Nq is the numerical value of source string.

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

$$Sn = Ln + 1 \quad (4)$$

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

3.3.3 Centers generation: the filled-in group with completed features will be selected to generate centers by using one of the current methods such as K-means, Fuzzy c-means (FCM) clustering. The investigator will select the number of centers before the process begins, where these clustering algorithms are only used to predict the centers of the clusters.

3.3.4 Euclidean distance (ED) [22]: ED is matrices of the squared distances between points. The centers will be used to find the other vectors using ED. Each pre-cluster group contains specific features which will only be calculated with same features of centers. Afterwards, the shortest distance between a vector and a center, the vector will be assigned to this particular cluster. ED can be calculated by using following equation:

$$d = \sum_{i=1}^{n} (x_i - y_i)^2 \quad (5)$$

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

4 EXPERIMENTAL RESULTS

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

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

For each category within the three cases, the clustering procedure ran three times to ensure the stability of the developed process. In this paper, five clustering sizes were selected (15, 25, 50, 75, and 100) to obtain a view of clustering performance across all categories using FCM, K-means algorithms. Tables 3, 4, and 5 show the details of experimental results of three cases. These results obtained based on top five clusters containing a high number of notable artifacts. These results illustrated a proportion of notable versus the benign data rather than the actual number of artifacts.

Table 3: Experimental results of Case 1 (✓: Notable; ×: Benign)

Centers Generation		FCM					K-Means				
Category	Cluster size	15	25	50	75	100	15	25	50	75	100
File List	✓ (%)	100	100	98.8	98.8	96.5	100	100	97.7	95.4	96.5
	× (%)	14.1	12.4	11.5	11.4	10.9	12.4	12	11.3	11	10.8
Email	✓ (%)	100	100	100	100	100	100	100	100	100	100
	× (%)	100	100	100	100	100	100	100	100	100	100
Internet Data	✓ (%)	97.7	97.7	97.7	80	80	100	86.6	40	33.3	31.1
	× (%)	47.4	52.2	53.8	26.1	28.4	76.7	38	17	13.8	10.9

Table 4: Experimental results of Case 2

Centers Generation		FCM					K-Means				
Category	Cluster size	15	25	50	75	100	15	25	50	75	100
File List	✓ (%)	100	100	100	100	100	100	100	100	98.9	79.4

EXIF Data	× (%)	16.5	12.9	13	13	12.3	99.8	13	12.3	11.3	10.2
	✓ (%)	100	100	100	100	100	100	100	98.3	96.2	95
	× (%)	12.9	13	10.6	10.5	14.8	12.9	14.8	9.7	8	7.8

Table 5: Experimental results of Case 3

Centers Generation		FCM					K-Means				
Category	Cluster Size	15	25	50	75	100	15	25	50	75	100
File List	✓ (%)	100	100	94.4	86.6	90	100	100	97.7	91	92.2
	× (%)	5.2	4.8	0.7	0.65	0.5	15.2	4.9	2.8	1.2	1.17
EXIF Data	✓ (%)	100	100	100	100	100	100	100	100	100	100
	× (%)	17.6	15.4	14.4	14.4	14.4	22	15.3	14.4	14.4	14.4
Internet Data	✓ (%)	100	100	100	100	100	100	100	100	100	100
	× (%)	100	100	100	100	100	100	100	100	100	100
SMS-Viber	✓ (%)	98.8	94.4	94.4	91.1	98.8	98.8	92.2	68.8	40	32.2
	× (%)	79.2	80.1	96.3	92.3	94.2	76.5	69.3	22	4.1	0.5

Table 3 shows the results of Case 1 across two algorithms with five configurations of clusters size. Noticeably, the clustering based upon File List with 15 and 25 cluster size provided successful isolating for the notable artifacts with 100% proportion across FCM and K-means within the top 5 clusters. Also, a good proportion of the benign data with at least 85.9% was eliminated. With increasing the cluster size (i.e. 50, 75, and 100), the proportion of benign and notable artifacts that were presented within the top 5 clusters decreased comparing with their counterparts from 15 and 25 cluster sizes. Indeed, more than 1.2% and 88.5% of notable and benign artifacts respectfully were grouped in other clusters. The results of the email category showed that all artifacts of notable and benign were grouped in one cluster for both algorithms across all cluster sizes. This phenomenon happened because there were only 19 files included in the email category. Regarding the internet data category, FCM showed that there is a stability in a proportion of notable artifacts in first three cluster sizes (15, 25, 50) with 97.7% and then the results dropped to 80% in larger cluster sizes (75,100). But, the proportion of benign artifacts was relativity high within first three sizes which reached to 53.8%, demonstrating that more half of the benign artifacts were grouped in top five clusters. While the proportion of benign artifacts also dropped to reach to 26.1% with the cluster size of 75 configuration. In contrast, K-means algorithm showed a difference in term of the proportion of notable and benign artifacts. The 25-cluster configuration showed the best as there was a high proportion of notable artifacts with a relatively small number of benign data in top five clusters. While the 15-cluster configuration contained the high proportion of both notable and benign artifacts. This happened because File List contained a high number of carved files with missing features such as a timestamp. In addition, the small size of cluster configuration led to group most artifacts in top-five clusters.

The results of Case 2 are presented in Table 4. This case showed that 100% of notable artifacts were grouped within the top 5 clusters across all cluster sizes for both the File List and EXIF data categories by using the FCM algorithm. For the same configurations and categories, the proportion of non-notable artifacts was also low with a range of 10.5% -16.5% being presented in top five clusters. Regarding K-means, the majority of the artifacts (both notable and benign) were grouped within top five clusters by using cluster size 15, indicating the ineffectiveness of this setting and most investigation on this configuration is required. In comparison, results from the configuration with larger cluster sizes (e.g. 25, 50, 75) show that most notable artifacts were grouped within the top five clusters with a small amount of benign data being present. This probably occurred because the timestamps of evidential files were relatively close. Clustering based EXIF data achieved good results as at least 95% of notables were grouped within the top five clusters across all configurations; while the benign data was relatively low with a maximum of 14.8% being grouped within the top five clusters under all setups.

Table 5 illustrats the results of Case 3. Regarding to the File List, FCM and K-means showed the best result in grouping all notable artifacts within the top five clusters by using cluster sizes 15 and 25; under the same configurations, the amount of benign data being allocated to the top five clusters for FCM was smaller to its K-means counterpart. While the amount of benign data being grouped within the top five clusters is much smaller across the rest of cluster sizes (i.e. 50, 75 and 100) for FCM and K-means, the number of notable artifacts also reduced (with up to 13.4% of reduction). Clustering based EXIF data showed that 100% of the notables founded within top five clusters for all cluster sizes and across both algorithms. This could happen due to the pictures were taken in one location where GPS data was relatively similar. In contrast, the proportion of benign data being gathered within the top five clusters decreased slightly as the cluster size increased. The internet and messaging categories reflected most critical results of this case due to the high proportion of benign data that were found within top five clusters. Regarding internet data, all notable and benign artifacts can be observed within top five clusters across all cluster

sizes and algorithms. While the results based on messaging category illustrated that there was a difference in the proportion of notable and benign between the two algorithms (FCM, K-means). By using FCM algorithm, more than 91% of notable artifacts founded within top five clusters, but more than 80% of benign data also grouped with top five clusters. In contrast, by using K-means, a large proportion of notable and benign obtained using a small size of clusters. Additionally, the proportion of notable and benign reduced significantly using large sizes of clusters.

5 DISCUSSIONS

From the aforementioned results, the proposed approach of clustering has the ability to group the evidential artifacts within top five clusters. Therefore, the approach can correlate the related artifacts in the same category. Indeed, each case contains more than one evidential source with various categories. These categories were classified into file system and applications. Within each case, there are similar categories such as File List, messaging, and internet data. The process of merging the similar categories has successfully achieved without any effect on the clustering process. The clustering based on File List showed the best results across the three cases with 100% of notables being grouped with the top five clusters using FCM and K-means with a relative small amount of benign data being included. This was due to a large number of files contained in these categories as the clustering works well with large volume of data. However, the results based File List category in Case 1 and Case 2 were relatively similar where all notable were grouped within rank five clusters using small sizes of cluster configurations (15,25) with only 12% and 5% of benign data in Case1 and Case 3 respectively. Whenever the size of cluster configuration increased, the proportion of notable and benign decreased. Meaning, the small size of clusters configuration comparatively contained a large number of both notable and benign while large size of clusters configuration comparatively contains a less proportion of both notable and noise. In contrast, in Case 2, the clustering based the File List illustrated that there was no influence apparently in the results in terms of notable and benign when changing the cluster size using FCM algorithm. Moreover, the proportion of benign is relatively constant and small. While the clustering based K-means has proven to be challenging in Case 2 where most notable and benign in the File List were obtained in the rank-five clusters using 15 as a cluster size.

With regard to the clustering based applications categories, it was revealed that the performance of grouping the evidential artifacts with minimum a proportion of benign was less efficient compared with clustering based on the File List. This could be due to the small number of files within the applications categories. However, the clustering based EXIF category presented the best results among applications categories in terms of grouping the notable in Case 2 and Case3 using FCM and K-means. However, it is notable that the proportion of benign files using K-means clusters was less than the proportion of benign using FCM. The results of email category within the Case 1 and internet category within Case 3 showed the worst results because all notable and benign files were grouped in one cluster. This was due to the small number of files that provided to cluster procedure (e.g. only 19 files in total in the email category).

6 CONCLUSION AND FUTURE WORK

This paper has examined the possibility of using clustering algorithms in the digital forensics analysis. The proposed approach of clustering is working upon the merged datasets which come from various resources within a single case. The experimental results proved that the evidence can be correlated within a dataset, and the evidential artifacts can be grouped in the rank-five cluster. The results of identifying notable artifacts within rank-five cluster revealed that clustering based file systems were more accurate than clustering based applications. The results also illustrated that there is a slight difference between FCM and K-means algorithms, but the FCM showed a stability in the results across various configurations of cluster size.

Future research will focus upon developing a method using AI applications to identify the clusters containing the evidence in an automated way.

References

[1] D. Quick, and K. K. R. Choo, “Big forensic data management in heterogeneous distributed systems: quick analysis of multimedia forensic data,” *Software: Practice and Experience,* vol. 47, no. 8, pp. 1095-1109, 2017.

[2] V. Roussev, C. Quates, and R. Martell, “Real-time digital forensics and triage,” *Digital Investigation,* vol. 10, no. 2, pp. 158-167, 2013.

[3] H. Mohammed, N. Clarke, and F. Li, “An automated approach for digital forensic analysis of heterogeneous big data,” *The Journal of Digital Forensics, Security and Law: JDFSL,* vol. 11, no. 2, pp. 137, 2016.

[4] D. Bennett, “The challenges facing computer forensics investigators in obtaining information from mobile devices for use in criminal investigations,” *Information Security Journal: A Global Perspective,* vol. 21, no. 3, pp. 159-168, 2012.

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

[6] A. S. Tanenbaum, *Modern operating system*: Pearson Education, Inc, 2009.

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

[8] D. Ayers, “A second generation computer forensic analysis system,” *digital investigation,* vol. 6, pp. S34-S42, 2009.

[9] S. Almulla, Y. Iraqi, and A. Jones, “Feasibility of Digital Forensic Examination and Analysis of a Cloud Based Storage Snapshot,” *Journal of Digital Information Management,* vol. 15, no. 1, pp. 19, 2017.

[10] V. S. Harichandran, F. Breitinger, I. Baggili, and A. Marrington, “A cyber forensics needs analysis survey: Revisiting the domain's needs a decade later,” *Computers & Security,* vol. 57, pp. 1-13, 2016.

[11] R. Xu, and D. Wunsch, “Survey of clustering algorithms,” *IEEE Transactions on neural networks,* vol. 16, no. 3, pp. 645-678, 2005.

[12] S. C. Guptill, “Metadata and data catalogues,” *Geographical information systems,* vol. 2, pp. 677-692, 1999.

[13] H. Mohammed, N. Clarke, and F. Le, “Automating the Harmonisation of Heterogeneous Data in Digital Forensics,” in 17th European Conference on Cyber Warfare and Security, Oslo, Norway, 2018.

[14] N. C. Rowe, and S. L. Garfinkel, "Finding anomalous and suspicious files from directory metadata on a large corpus." pp. 115-130.
[15] L. F. da Cruz Nassif, and E. R. Hruschka, "Document clustering for forensic computing: An approach for improving computer inspection." pp. 265-268.
[16] M. Al Fahdi, N. L. Clarke, F. Li, and S. M. Furnell, "A suspect-oriented intelligent and automated computer forensic analysis," *Digital Investigation*, vol. 18, pp. 65-76, 2016.
[17] N. C. Rowe, "Identifying forensically uninteresting files using a large corpus." pp. 86-101.
[18] N. L. Beebe, and L. Liu, "Clustering digital forensic string search output," *Digital Investigation*, vol. 11, no. 4, pp. 314-322, 2014.
[19] M. Kalra, N. Lal, and S. Qamar, "K-Mean Clustering Algorithm Approach for Data Mining of Heterogeneous Data," *Information and Communication Technology for Sustainable Development*, pp. 61-70: Springer, 2018.
[20] NIST, "The CFReDS project," 2015, 2015.
[21] T. P. Liang, E. Turban, and J. E. Aronson, "Decision Support Systems and Intelligent Systems," *Yogyakarta: Penerbit Andi*, 2005.
[22] P.-E. Danielsson, "Euclidean distance mapping," *Computer Graphics and image processing*, vol. 14, no. 3, pp. 227-248, 1980.