Fuzzy-Import Hashing: A Malware Analysis Approach

Nitin Naik[1], Paul Jenkins[1], Nick Savage[1], Longzhi Yang[2], Tossapon Boongoen[3] and Natthakan Iam-On[3]

[1]School of Computing, University of Portsmouth, United Kingdom
[2]Department of Computer and Information Sciences, Northumbria University, United Kingdom
[3]School of Information Technology, Mae Fah Luang University, Thailand
Email: {nitin.naik, paul.jenkins, nick.savage}@port.ac.uk, longzhi.yang@northumbria.ac.uk,
tossapon.boo@mfu.ac.th, natthakan@mfu.ac.th

***Abstract*—Malware has remained a consistent threat since its emergence, growing into a plethora of types and in large numbers. In recent years, numerous new malware variants have enabled the identification of new attack surfaces and vectors, and have become a major challenge to security experts, driving the enhancement and development of new malware analysis techniques to contain the contagion. One of the preliminary steps of malware analysis is to remove the abundance of counterfeit malware samples from the large collection of suspicious samples. This process assists in the management of man and machine resources effectively in the analysis of both unknown and likely malware samples. Hashing techniques are one of the fastest and efficient techniques for performing this preliminary analysis such as fuzzy hashing and import hashing. However, both hashing methods have their limitations and they may not be effective on their own, instead the combination of two distinctive methods may assist in improving the detection accuracy and overall performance of the analysis. This paper proposes a Fuzzy-Import hashing technique which is the combination of fuzzy hashing and import hashing to improve the detection accuracy and overall performance of malware analysis. This proposed Fuzzy-Import hashing offers several benefits which are demonstrated through the experimentation performed on the collected malware samples and compared against stand-alone techniques of fuzzy hashing and import hashing.**

***Index Terms*—Malware Analysis; Fuzzy-Import Hashing; Fuzzy Hashing; Import Hashing; Fuzzy C-Means Clustering; Ransomware.**

I. Introduction

Malware is a short form of MALicious softWARE, which is a collective phrase for all software developed for disrupting, damaging or gaining access to data and systems in an unauthorised manner. Categorised into several categories depending on factors such as creator, variant, code, activity and severity to aid in identification and classification. Providing a taxonomy does not detract from the challenge of enhancing and evolving counter measure techniques capable of analysing malware, especially as a result of changing attack surfaces and vectors, and hence attack techniques, combined with the substantial growth in malware. One of the preliminary steps of malware analysis is to remove the abundance of counterfeit malware samples from the large collection of suspicious samples. This preliminary step is a crucial task in determining the success of subsequent steps of the malware analysis, as it concentrates resources on analysing the most probable malware samples. This step can be performed either in a static mode or dynamic mode, where the static process does not run the suspicious samples and is safer, while the dynamic process executes the malware and is more sensitive [1]. There are several analysis techniques available, however, every technique has its own strengths and limitations and may not be successful in analysing every type of malware. Therefore, attempting to select a single analysis method that can be successfully applied on every type of malware is a non trivial task.

Hashing techniques are one of the fastest and efficient techniques for conducting this preliminary analysis such as fuzzy hashing and import hashing [2], [3]. However, both hashing methods have their own limitations; for example, fuzzy hashing performs well when similarity exists between the structure of files, and import hashing when similarity exists between the import address tables of files. Consequently, they may not be effective on their own, however, the combination of these two distinctive methods may assist in improving the detection accuracy and overall performance of the analysis. Therefore, this paper proposes a fuzzy-import hashing technique which is the combination of fuzzy hashing and import hashing to improve the detection accuracy and overall performance of the analysis.

Import hashing is faster than fuzzy hashing, therefore, their combined operation is performed in such a way that import hashing is applied on samples initially and if it cannot find any match then and only then is fuzzy hashing applied. In this way, fuzzy hashing is only utilised on the remaining samples which could not be matched by import hashing, thus saving the computational overheads of fuzzy hashing. Their combination offers several benefits such as they can complement each other, so that when one method cannot detect a match, then the other technique can; and an import hashing result can be easily fused with a fuzzy hashing result to obtain final similarity scores, which may be very useful in improving the result of further clustering or classification. Therefore, the proposed fuzzy-import hashing can improve the detection accuracy and overall performance whilst simultaneously maintaining its speed. These benefits are demonstrated through the experimentation performed on the collected ransomware samples

978-1-7281-6932-3/20/$31.00 ©2020 IEEE

and compared against stand-alone fuzzy hashing and import hashing. Additionally, due to the fuzzy nature of the fuzzy-import hashing result, it is employed in Fuzzy C-Means (FCM) clustering [4], [5], [6], and compared against its corresponding fuzzy hashing based FCM result to demonstrate its further benefits.

The paper is divided into the following sections: Section II discusses the chosen analysis methods import hashing and fuzzy hashing. Section III explains the collection and verification process of malware (ransomware) samples. Section IV discusses analysis of malware (ransomware) employing the chosen analysis methods fuzzy hashing and import hashing. Section V discusses the analysis of malware (ransomware) employing the proposed fuzzy-import hashing. Section VI compares the FCM results of fuzzy-import hashing against their corresponding fuzzy hashing method. Section VII presents some of the main benefits of the proposed fuzzy-import hashing method. Lastly, Section VIII presents the summary of the research work and suggests some future work.

II. Background

A. Fuzzy Hashing

Cryptographic hash and fuzzy hash techniques are utilised in security analysis in an attempt to detect malware when investigating both the integrity and similarity of files of interest. Of these two techniques it is the similarity which is of greater importance as malware developers base their code on previous examples leading to the development of new strains [7]. In fuzzy hashing analysis, the file of interest is divided into multiple blocks and a hash value is calculated for each block, with the final step being the concatenation of all hash values of the blocks to generate the fuzzy hash value as shown in Fig. 1. A number of factors affect the length of the fuzzy hash value, including the block size, the size of the file and the output size of the selected hash function [8]. Fuzzy hashing methods can be classified into several categories: Context-Triggered Piecewise Hashing (CTPH), Statistically-Improbable Features (SIF), Block-Based Hashing (BBH) and Block-Based Rebuilding (BBR) [9], [10], [11]. Forensic analysis of malware requires a thorough understanding of the degree of similarity between known malware samples and inert files to assess files for their threat potential. This is especially important when considering the analysis and clustering of suspected malware in order to identify new variants. As a result the use of the similarity preserving property of fuzzy hashing is useful in forensic investigation when comparing unknown files with known malware families for their triage and clustering, where samples have the same functionality, yet different cryptographic hash values [12].

1) SSDEEP: The SSDEEP fuzzy hashing method was initially developed for locating spam emails [7]. This method divides a file into number of blocks based on the content of that file. The endpoint points of these blocks are determined by a rolling hash method utilising the Adler32 function [8]. Generating the SSDEEP fuzzy hash value for the file, consists

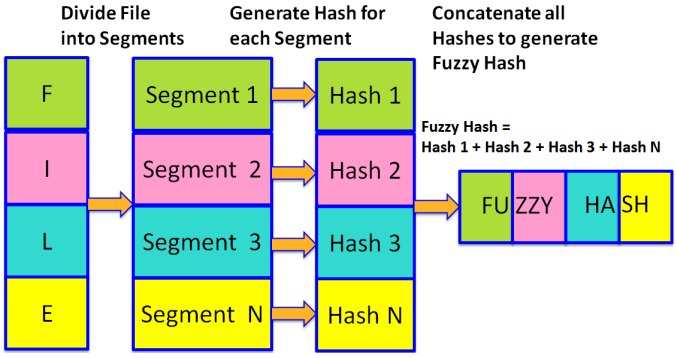

Fig. 1. Generation of Fuzzy Hash Value in Fuzzy Hashing Method

of calculating an individual hash value for each block and concatenating these into a single hash value. Similarity between the two files is calculated by utilising Damerau-Levenshtein distance function.

2) SDHASH: The SDHASH fuzzy hashing method finds common and rare features in a file and matches the rare features in another file to determine the degree of similarity between the two files [13]. Generally, a feature is a 64-byte string and is found using an entropy calculation. It employs the cryptographic hash function SHA-1 and Bloom filters to calculate the SDHASH fuzzy hash value of a file [14]. A Bloom filter is a space-efficient probabilistic data structure to find whether the element is definitely not present in the set or may be present in the set. Similarity between the two files is calculated by utilising a Hamming distance function.

3) mvHASH-B: The mvHASH-B fuzzy hashing method is slightly different from SDHASH fuzzy hashing method, which focuses on keeping the data unchanged even if there is a small change in it. Thus resulting in the same hash value being generated in the case of a minor change, thus preserving the similarity. However, mvHASH-B transforms the input data based on the concept of majority votes, then encodes the majority vote bit sequence with RLE (Run-Length Encoding - a type of lossless data compression approach), and finally generates mvHASH-B fuzzy hash value utilising Bloom filters [15]. Moreover, it uses a self-defined hash function which has a higher run time efficiency and its complexity is equivalent to the cryptographic hash function SHA-1.

B. Import Hashing - IMPHASH

Import hashing is one of a number of methods used to ascertain the similarity of two files. This method utilises import libraries (function calls from other software), where the order in which they are called and the functions themselves are utilised to generate a hash value (see Fig. 2). This IMPort HASH (IMPHASH) is based upon the Import Address Table (IAT), which is a list of the software (relocatable) and their functions including all the DLL, EXE and SYS files required to be bound and linked with the relocatable code of the original software to build the final application [16]. Thus, two pieces of software that were compiled with similar code except with a different order of functions will generate different IMPHASH

values. This method is analogous to Fuzzy Hashing with regard to its speed, computation, complexity and hash size, however, it is noteworthy that IMPHASH provides a binary similarity result, rather than the degree of similarity of two files.

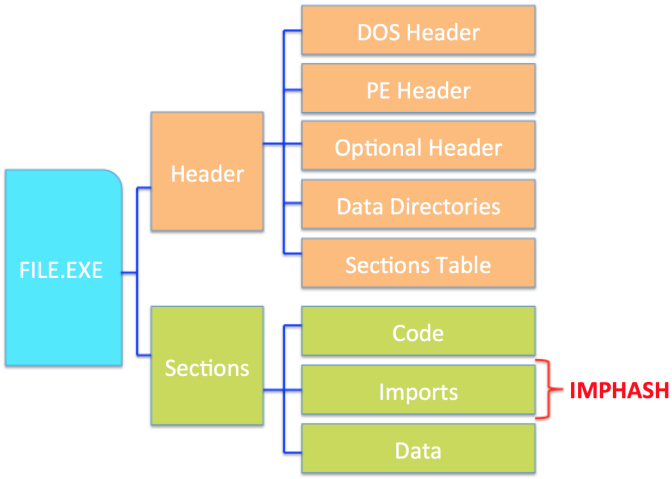

Fig. 2. Generation of IMPHASH Value from the Import Address Table (IAT) of a Portable Executable (PE) File

III. Collection of Malware Samples

In this implementation, one of the most prevalent malware classes, ransomware was selected to perform all analyses and utilised in evaluating the performance of the proposed fuzzy-import hashing method. Ransomware was selected for the experiment as it is one of the most relevant and damaging malware that exploits victims for financial gain, business disruption and market share [17]. Numerous types of ransomware have been created and used in cyberattacks, though, some ransomware categories were worthy of greater focus due to their severity of attack and financial loss. Based on primary research, four ransomware categories were targeted for this work WannaCry, Locky, Cerber and CryptoWall [18], [19], [20]. Thousands of malware samples were acquired from the two sources *Hybrid Analysis* [21] and *Malshare* [22]. Later, these samples were verified for their credibility as numerous samples were simply bogus samples. It was critical to select only credible samples of a specific category as a reference to test all chosen analysis methods and the proposed fuzzy-import hashing successfully. These samples were investigated based on the information available on *VirusTotal* [23]. To determine that every sample was indeed genuine malware or ransomware and were members of a specific ransomware category, the criteria was set that it must be identified as malware by at least 40 or more detection engines on *VirusTotal*. To check the ransomware category of collected samples, their category from WannaCry, Locky, Cerber and CryptoWall was verified manually on the recognized detection engines on *VirusTotal*. This sample collection and verification process was both lengthy and time consuming, leading to 1000 ransomware samples being selected out of several thousand samples, these were equally divided into 250 samples of four ransomware categories WannaCry, Locky, Cerber and CryptoWall. The four different categories of ransomware were chosen to evaluate how each analysis method works on the different categories of ransomware.

IV. Malware Analysis Using Fuzzy Hashing and Import Hashing

The malware analysis is performed to determine the success of import hashing and fuzzy hashing based on their similarity detection rate while applied to all the different types of ransomware samples WannaCry, Locky, Cerber and CryptoWall. The samples of a specific ransomware type possess certain similarity with other samples of the same type, and all the samples were carefully checked during the collection of ransomware samples. Thus, this analysis will determine the similarity detection success rate of import hashing and fuzzy hashing for each ransomware type based on whether they can match every sample with one or more samples of the same ransomware type.

A. Malware Analysis Approach: Fuzzy Hashing

When fuzzy hashing is applied on an unpacked ransomware sample, it generates a fuzzy hash value for that ransomware sample. This fuzzy hash value can be matched against either already identified ransomware samples or their fuzzy hash values. If the fuzzy hash of a sample in question matches with any of the pre-identified ransomware samples or its fuzzy hash value, then, the fuzzy hash result is generated as a degree of similarity between the two. This fuzzy similarity result is presented in the range of 1% (least matched) to 100% (exactly matched), however, it is entirely at the discretion of security experts how this value is interpreted depending on their analysis requirement. Generally, a threshold value can be set to accept or ignore the fuzzy similarity score and to determine as matched or not matched scenarios respectively. The fuzzy hashing should only be used as an initial investigation that may assist in any further analysis but not as a conclusive result [24].

B. Malware Analysis Results: Fuzzy Hashing

In this experiment, the SSDEEP, SDHASH and mvHASH-B fuzzy hashing methods were used to detect similarity for each ransomware category separately. It was important to assess the performance of these three methods in different threshold conditions for comparison purposes; therefore, their similarity detection results were evaluated in four different conditions: 1) when all the fuzzy similarity scores were considered (1-100%), 2) when those fuzzy similarity scores were considered which are greater than 10%, 3) when those fuzzy similarity scores were considered which are greater than 20%, and 4) when those fuzzy similarity scores were considered which are greater than 30%. The four evaluation results for the four ransomware categories are presented in Table I. One of the most important findings in all four evaluation results is that the detection rate of SDHASH and mvHASH-B fuzzy hashing methods decreased and in some cases quite significantly as the

similarity threshold value increased. The detection rate of the SSDEEP fuzzy hashing method is lower, however, consistent in all four experiments. At the final similarity threshold limit of 30%, most SSDEEP results are superior to the other two fuzzy hashing methods. This finding is crucial when utilising these similarity results in further analysis as they can affect the next stage (e.g., clustering or classification) result significantly.

C. Malware Analysis Approach: Import Hashing

Similarly, when import hashing is applied on an unpacked ransomware sample, it generates an IMPHASH value for that ransomware sample. Moreover, this IMPHASH value can be matched against either existing identified ransomware samples or their IMPHASH values. If the IMPHASH value of the investigated sample matches with any of the pre-identified ransomware samples or its IMPHASH value, then, the result is generated as a matched sample with one or more samples. However, it does not provide a degree of similarity, rather a binary output (i.e. either matched or not matched). The import hashing should only be used as an initial investigation that may help in any further analysis but not as a conclusive result [12].

D. Malware Analysis Results: Import Hashing

In this experiment, the import hashing method was used to detect similarity for each ransomware category separately. The similarity detection results for all the four ransomware categories are shown in Table II. The import hashing result is a mixed result when compared with the fuzzy hashing results. In one case it is somewhat better, however, in other cases it is slightly lower. It is worth noting that import hashing can only be used on PE file formats, therefore, its effectiveness depends on the type of samples investigated.

In summary, the detection rate of three fuzzy hashing (SSDEEP, SDHASH and mvHASH-B) and import hashing methods were not very good. Therefore, these methods cannot be used as a reliable malware analysis method and require further enhancement. Import hashing and fuzzy hashing are both fast and compact methods, and may be combined together to apply two different detection methods to improve the detection rate without affecting the performance significantly.

V. Malware Analysis Using the Proposed Fuzzy-Import Hashing

A. Malware Analysis Approach: Fuzzy-Import Hashing

Every malware analysis method has certain limitations due to its typical working procedure and not every method works well with every type of malware. Consequently, it may be useful to combine two different detection methods to enhance the detection capability, provided this does not affect the performance of the analysis significantly. Fuzzy-import hashing is the combination of fuzzy hashing and import hashing which applies both techniques to detect the similarity between two files. Both fuzzy hashing and import hashing are a compact, fast and resource-optimised method employed for analysis which may not be effective on their own, nonetheless they can complement each other and may improve the overall detection accuracy without affecting the overall performance significantly [12]. Fuzzy hashing attempts to find structural similarity between the two files in their entirety, whereas import hashing attempts to find similarity between import address tables of files. Therefore, they can complement each other in finding a missed opportunity by one of the methods. Thus, the combined search result can increase the detection accuracy and confidence level of the overall analysis.

Fuzzy hashing provides the result as a degree of similarity of each matched sample and import hashing only reveals whether the sample is matched or not. The two different types of results require a suitable alignment to combine these two results to be utilised as one result in the advanced analysis. In this fuzzy-import hashing method, an import hashing matched result is considered similar to 1 or 100% of fuzzy hashing result (exact match of fuzzy hashing) and an unmatched result is considered similar to 0 of fuzzy hashing result (no match of fuzzy hashing). The reason for considering import hashing matched result as 1 or 100% is: if the two IATs of two files are same then it is very likely that they hold a very strong similarity due to similar sequencing of function calls. This enables the fuzzy-import hashing method to generate its combined results as a degree of similarity in the range of 1 to 100%, in similar way to fuzzy hashing results. Import hashing is faster than fuzzy hashing, therefore, their combined operation is performed in such a way that import hashing is applied on samples initially, and if a match cannot be found then fuzzy hashing is applied. In this way, fuzzy hashing can only be applied on the remaining samples which could not be matched by import hashing, thus saving computational overheads of the fuzzy hashing method. The logical approach for this implementation is shown using the pseudocode in Algorithm 1.

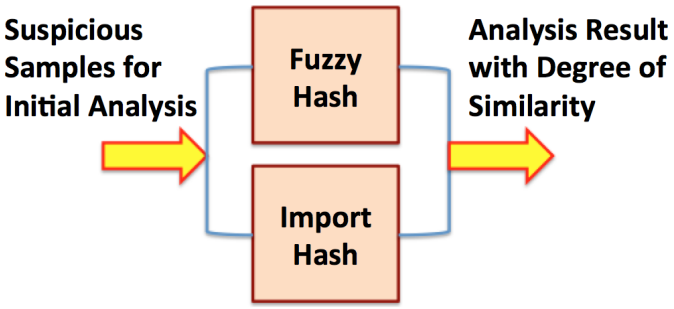

Fig. 3. Fuzzy-Import Hashing: A Malware Analysis Approach

B. Malware Analysis Results: Fuzzy-Import Hashing

The performance of fuzzy-import hashing was compared against the performance of import hashing and three fuzzy hashing methods SSDEEP, SDHASH and mvHASH-B. This evaluation was to determine whether this integration was successful or not, and if successful, then which fuzzy hashing method produced greater accuracy in the results. The similarity detection result of fuzzy-import hashing utilising

TABLE I
MALWARE ANALYSIS RESULTS OF THE SSDEEP, SDHASH AND MVHASH-B FUZZY HASHING METHODS FOR WANNACRY, LOCKY, CERBER AND CRYPTOWALL RANSOMWARE SAMPLES

Fuzzy Hashing Matching Criteria	WannaCry Ransomware			Locky Ransomware			Cerber Ransomware			Cryptowall Ransomware		
	SSDEEP Detection Rate	SDHASH Detection Rate	mvHASH-B Detection Rate	SSDEEP Detection Rate	SDHASH Detection Rate	mvHASH-B Detection Rate	SSDEEP Detection Rate	SDHASH Detection Rate	mvHASH-B Detection Rate	SSDEEP Detection Rate	SDHASH Detection Rate	mvHASH-B Detection Rate
Fuzzy Similarity Scores (1-100%)	91.2%	93.6%	90%	42%	58.4%	72.4%	33.6%	71.2%	94.8%	28%	52.4%	83.6%
Fuzzy Similarity Scores >10%	91.2%	93.6%	90%	42%	38.4%	64%	33.6%	62.8%	90.4%	28%	32.8%	56.8%
Fuzzy Similarity Scores >20%	91.2%	90%	84.4%	41.6%	35.6%	36.4%	33.6%	37.6%	36.8%	28%	24%	20.8%
Fuzzy Similarity Scores >30%	90.8%	90%	84.4%	41.6%	30.4%	33.6%	33.6%	28.4%	36%	28%	20.4%	20.4%

TABLE II
MALWARE ANALYSIS RESULTS OF IMPORT HASHING FOR WANNACRY, LOCKY, CERBER AND CRYPTOWALL RANSOMWARE SAMPLES

Ransomware Category	Import Hashing Detection Rate
WannaCry Ransomware	87.6%
Locky Ransomware	31.6%
Cerber Ransomware	61.6%
CryptoWall Ransomware	27.2%

Algorithm 1: Pseudocode of Fuzzy-Import Hashing to determine Malware Similarity by combining Fuzzy Hash with Import Hash

$\mathbb{S}$, Set of Samples for Investigation
$\mathbb{I}$, Set of Import Hashes of Known Malware
$\mathbb{F}$, Set of Fuzzy Hashes of Known Malware
S, Similarity Score
I, Import Hash Value
F, Fuzzy Hash Value
δ_T, Fuzzy Hash Similarity Threshold
Δ, Degree of Similarity
for $(i = 1; i < |\mathbb{S}|; i++)$ **do**
  for $(j = 1; j < |\mathbb{I}|; j++)$ **do**
    if $I_{\mathbb{S}_i} == I_j$ **then**
      $S_{i,j} = 1$
  if $I_{\mathbb{S}_i} \notin \mathbb{I}$ **then**
    for $(k = 1; k < |\mathbb{F}|; k++)$ **do**
      if $\Delta(F_{\mathbb{S}_i}, F_k) \geq \delta_T$ **then**
        $S_{i,k} = \Delta(F_{\mathbb{S}_i}, F_k)$
return $S[\ ]$

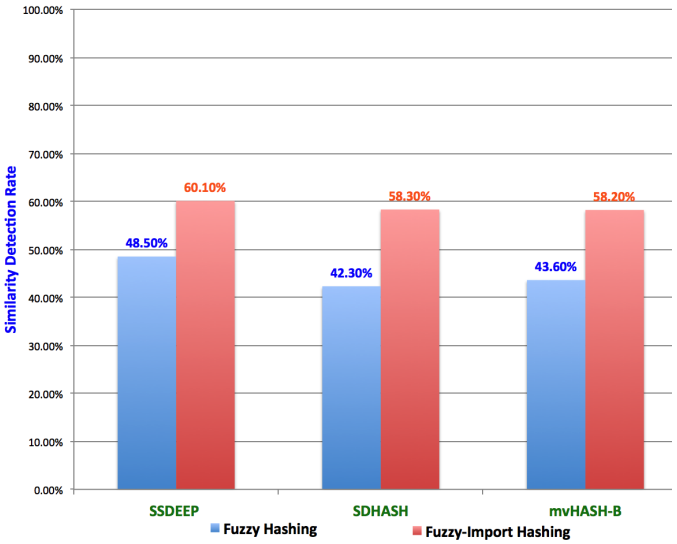

Fig. 4. Comparative Analysis of the Overall Similarity Detection Rate of Fuzzy-Import Hashing (SSDEEP, SDHASH and mvHASH-B) and their corresponding Fuzzy Hashing (SSDEEP, SDHASH and mvHASH-B) for the collected Ransomware Samples

three different fuzzy hashing methods for all the four ransomware categories is shown in Table III, where, the fuzzy similarity scores greater than 30% were utilised for all the three fuzzy hashing methods (i.e. the fuzzy hash similarity threshold was set at 30%). Noticeably, fuzzy-import hashing with all the three fuzzy hashing methods showed improvement (at least >11%), but SSDEEP fuzzy hashing based fuzzy-import hashing produced comparatively better overall results of analysis (see Fig. 4). On the basis of this experimentation, fuzzy-hashing based analysis results were slightly improved in almost all the cases.

TABLE III
COMPARISON OF MALWARE ANALYSIS RESULTS OF FUZZY-IMPORT HASHING (SSDEEP, SDHASH AND MVHASH-B), FUZZY HASHING (SSDEEP, SDHASH AND MVHASH-B) AND IMPORT HASHING (IMPHASH) FOR THE COLLECTED RANSOMWARE SAMPLES

Ransomware Category	**IMPHASH Import Hashing Detection Rate**	**SSDEEP Fuzzy Hashing Detection Rate**	**SSDEEP Fuzzy-Import Hashing Detection Rate**	**SDHASH Fuzzy Hashing Detection Rate**	**SDHASH Fuzzy-Import Hashing Detection Rate**	**mvHASH-B Fuzzy Hashing Detection Rate**	**mvHASH-B Fuzzy-Import Hashing Detection Rate**
WannaCry Ransomware	87.6%	90.8%	92.8%	90%	92.8%	84.4%	92.8%
Locky Ransomware	31.6%	41.6%	48.8%	30.4%	45.2%	33.6%	44.4%
Cerber Ransomware	61.6%	33.6%	61.6%	28.4%	61.6%	36%	61.6%
CryptoWall Ransomware	27.2%	28%	37.2%	20.4%	33.6%	20.4%	34%
Overall Detection Rate of Each Hashing Method	**52%**	**48.5%**	**60.1%**	**42.3%**	**58.3%**	**43.6%**	**58.2%**

VI. CLUSTERING OF MALWARE SAMPLES USING FCM BASED ON SIMILARITY SCORES OF FUZZY-IMPORT HASHING AND THEIR CORRESPONDING FUZZY HASHING

As previously mentioned, the fuzzy-import hashing results can be further utilised for advanced analysis such as clustering, and due to its fuzzy similarity results it can be directly utilised with fuzzy c-means clustering [4], [5], [6]. Here, both fuzzy-import hashing and its corresponding fuzzy hashing results were utilised for FCM to compare if fuzzy-import hashing could improve the clustering results. Evaluation of the clustering was undertaken using the fuzzy c-means clustering results of four fuzzy indexes *Fuzzy Silhouette Index, Partition Coefficient, Modified Partition Coefficient and Partition Entropy*, which were computed and collectively compared for both fuzzy-import hashing and its corresponding fuzzy hashing to determine the optimal clustering value of that category. Here, the higher value of the first three evaluation metrics signifies better clustering results and the lower value of the fourth evaluation metric signifies better clustering results. This computation was based on *fclust* package of **R** [25]. The comparative results are shown in Tables IV to VII for four different ransomware categories: WannaCry, Locky, Cerber and CryptoWall respectively. The majority of fuzzy-import based FCM results (8 out 12) were improved, however, some FCM results were not, indicating the requirement for further analysis of unimproved results and the possible reasons.

VII. BENEFITS OF THE PROPOSED FUZZY-IMPORT HASHING

Combining fuzzy hashing with import hashing to develop fuzzy-import hashing could offer several benefits, however some of the most notable benefits are:

- **Performance Sustainability:** Import hashing is one of the fastest analysis methods as it only generates the hash of a part of a file (i.e., IAT) and does not affect the overall performance of the combined analysis process.
- **Detection Rate Improvement:** Both hashing methods can sometimes complement each other when one hashing method fails to find similarity due to its particular limitations. Therefore, fuzzy-import hashing can detect greater malware samples than any single method alone.
- **Overheads Minimisation:** Import hashing is faster than fuzzy hashing, therefore, if it is applied on samples prior to fuzzy hashing then all the matched samples would not need to be processed again through fuzzy hashing as only unmatched samples require checking by fuzzy hashing. This avoids performing fuzzy hashing on all the samples thus reducing overheads required by fuzzy hashing alone.
- **Result Alignment:** Import hashing binary results can be readily aligned with fuzzy hashing results where the matching result could be treated similarly to 1 or 100% of a fuzzy hashing result (exact match of fuzzy hashing) and an unmatched result could be treated similarly to 0 of a fuzzy hashing result (no match of fuzzy hashing). Therefore, the two results can be easily aligned together in the form of fuzzy similarity scores.
- **Accuracy Improvement:** In case of import hashing found matched sample(s), the strong similarity score 1 or 100% is added to the final similarity result of fuzzy-import hashing, which increases the accuracy of the overall result and the further processing results of clustering or classification.

VIII. CONCLUSION

This paper proposed a fuzzy-import hashing method for malware analysis to improve the similarity detection accuracy and performance of the analysis. This method was developed combining fuzzy hashing and import hashing methods; where, both methods are a compact and fast hashing methods, however, they are not always capable of producing reliable and positive results as a stand-alone method. In this implementation, the combined operation of the two methods was performed in such a way that import hashing was applied on samples initially due to its speed and if it could not find any match then and only then, was fuzzy hashing applied. This ensured that fuzzy hashing could only be

TABLE IV
COMPARISON OF FCM RESULTS BASED ON SIMILARITY SCORES OF FUZZY-IMPORT HASHING (SSDEEP, SDHASH AND MVHASH-B) AND FUZZY HASHING (SSDEEP, SDHASH AND MVHASH-B) FOR THE COLLECTED WANNACRY RANSOMWARE SAMPLES

Cluster Validity Index	**SSDEEP Fuzzy Hashing**	**SSDEEP Fuzzy-Import Hashing**	**SDHASH Fuzzy Hashing**	**SDHASH Fuzzy-Import Hashing**	**mvHASH-B Fuzzy Hashing**	**mvHASH-B Fuzzy-Import Hashing**
Fuzzy Silhouette Index	0.78324	0.793863	0.6958656	0.8228494	0.7994093	0.826524
Partition Coefficient	0.6433042	0.6691064	0.7055717	0.8488053	0.4005461	0.7275929
Modified Partition Coefficient	0.5719651	0.6029277	0.646686	0.8185663	0.2806553	0.6731114
Partition Entropy	0.8016553	0.7308573	0.5792176	0.3254954	1.278439	0.6301614

TABLE V
COMPARISON OF FCM RESULTS BASED ON SIMILARITY SCORES OF FUZZY-IMPORT HASHING (SSDEEP, SDHASH AND MVHASH-B) AND FUZZY HASHING (SSDEEP, SDHASH AND MVHASH-B) FOR THE COLLECTED LOCKY RANSOMWARE SAMPLES

Cluster Validity Index	**SSDEEP Fuzzy Hashing**	**SSDEEP Fuzzy-Import Hashing**	**SDHASH Fuzzy Hashing**	**SDHASH Fuzzy-Import Hashing**	**mvHASH-B Fuzzy Hashing**	**mvHASH-B Fuzzy-Import Hashing**
Fuzzy Silhouette Index	0.9085124	0.9300699	0.8325851	0.8326483	0.8816986	0.8443886
Partition Coefficient	0.8376619	0.838944	0.7522258	0.7536781	0.9988531	0.9408663
Modified Partition Coefficient	0.8051943	0.8053328	0.702671	0.7044138	0.9986237	0.9290395
Partition Entropy	0.3675082	0.347518	0.5703733	0.5602319	0.005407051	0.1397954

TABLE VI
COMPARISON OF FCM RESULTS BASED ON SIMILARITY SCORES OF FUZZY-IMPORT HASHING (SSDEEP, SDHASH AND MVHASH-B) AND FUZZY HASHING (SSDEEP, SDHASH AND MVHASH-B) FOR THE COLLECTED CERBER RANSOMWARE SAMPLES

Cluster Validity Index	**SSDEEP Fuzzy Hashing**	**SSDEEP Fuzzy-Import Hashing**	**SDHASH Fuzzy Hashing**	**SDHASH Fuzzy-Import Hashing**	**mvHASH-B Fuzzy Hashing**	**mvHASH-B Fuzzy-Import Hashing**
Fuzzy Silhouette Index	0.8559951	0.6945895	0.6917668	0.6937094	0.7052969	0.7406375
Partition Coefficient	0.7772775	0.6930008	0.7838876	0.7951774	0.6131531	0.656417
Modified Partition Coefficient	0.732733	0.631601	0.7406651	0.7542129	0.5357837	0.5877004
Partition Entropy	0.4904145	0.6716732	0.4853877	0.4553734	0.8616182	0.7794824

TABLE VII
COMPARISON OF FCM RESULTS BASED ON SIMILARITY SCORES OF FUZZY-IMPORT HASHING (SSDEEP, SDHASH AND MVHASH-B) AND FUZZY HASHING (SSDEEP, SDHASH AND MVHASH-B) FOR THE COLLECTED CRYPTOWALL RANSOMWARE SAMPLES

Cluster Validity Index	**SSDEEP Fuzzy Hashing**	**SSDEEP Fuzzy-Import Hashing**	**SDHASH Fuzzy Hashing**	**SDHASH Fuzzy-Import Hashing**	**mvHASH-B Fuzzy Hashing**	**mvHASH-B Fuzzy-Import Hashing**
Fuzzy Silhouette Index	0.9863146	0.4704871	0.9988991	0.4587643	0.7775503	0.7939686
Partition Coefficient	0.7826071	0.6088555	0.9091084	0.1666667	0.5108229	0.822717
Modified Partition Coefficient	0.7391285	0.5306266	0.89093	0.5232862	0.4129875	0.7872604
Partition Entropy	0.4428341	0.8592991	0.1518233	1.791759	1.058915	0.4041951

applied on the remaining samples which could not be matched by import hashing, thus saving the computational overheads of fuzzy hashing. The similarity detection performance of fuzzy-import hashing was compared against stand-alone fuzzy hashing (SSDEEP, SDHASH and mvHASH-B) and import hashing, which demonstrated an improvement in similarity detection rate for each fuzzy hashing method. Subsequently, the FCM clustering result based on fuzzy-import hashing was compared against the stand-alone fuzzy hashing method (SSDEEP, SDHASH and mvHASH-B) to determine its success for advanced clustering analysis. This comparison indicated some positive results, however, further investigation is required of some unimproved cases. This proposed fuzzy-import hashing demonstrated some improvements in overall detection rates; however, this is still not a significant improvement to consider this proposed method as a generic analysis method and requires further analysis and improvement in the future.

ACKNOWLEDGEMENT

The authors gratefully acknowledge the support of *Hybrid-Analysis.com, Malshare.com* and *VirusTotal.com* for this research work.

REFERENCES

[1] C. Harrell. (2013) Finding Malware: Like Iron Man. [Online]. Available: https://digital-forensics.sans.org/summit-archives/DFIR_Summit/Finding-Malware-Like-Iron-Man-Corey-Harrell.pdf

[2] N. Naik, P. Jenkins, J. Gillett, H. Mouratidis, K. Naik, and J. Song, "Lockout-Tagout Ransomware: A detection method for ransomware using fuzzy hashing and clustering," in *IEEE Symposium Series on Computational Intelligence (SSCI)*, 2019.

[3] N. Naik, P. Jenkins, N. Savage, L. Yang, K. Naik, and J. Song, "Augmented YARA rules fused with fuzzy hashing in ransomware triaging," in *IEEE Symposium Series on Computational Intelligence (SSCI)*, 2019.

[4] J. C. Bezdek, R. Ehrlich, and W. Full, "FCM: The fuzzy c-means clustering algorithm," *Computers & Geosciences*, vol. 10, no. 2-3, pp. 191–203, 1984.

[5] R. Krishnapuram and J. M. Keller, "A possibilistic approach to clustering," *IEEE transactions on fuzzy systems*, vol. 1, no. 2, pp. 98–110, 1993.

[6] N. R. Pal, K. Pal, J. M. Keller, and J. C. Bezdek, "A possibilistic fuzzy c-means clustering algorithm," *IEEE transactions on fuzzy systems*, vol. 13, no. 4, pp. 517–530, 2005.

[7] J. Kornblum, "Identifying almost identical files using context triggered piecewise hashing," *Digital investigation*, vol. 3, pp. 91–97, 2006.

[8] A. Tridgell, "Efficient algorithms for sorting and synchronization," Ph.D. dissertation, Australian National University Canberra, 1999.

[9] F. Breitinger and H. Baier, "A fuzzy hashing approach based on random sequences and hamming distance," in *Annual ADFSL Conference on Digital Forensics, Security and Law. 15*, 2012. [Online]. Available: https://commons.erau.edu/adfsl/2012/wednesday/15

[10] C. Sadowski and G. Levin, "Simhash: Hash-based similarity detection," 2007. [Online]. Available: www.webrankinfo.com/dossiers/wp-content/uploads/simhash.pdff

[11] V. Gayoso Martínez, F. Hernández Álvarez, and L. Hernández Encinas, "State of the art in similarity preserving hashing functions," 2014. [Online]. Available: http://digital.csic.es/bitstream/10261/135120/1/Similarity_preserving_Hashing_functions.pdf

[12] N. Naik, P. Jenkins, N. Savage, and L. Yang, "Cyberthreat Hunting-Part 1: Triaging Ransomware using Fuzzy Hashing, Import Hashing and YARA Rules," in *IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*. IEEE, 2019.

[13] V. Roussev, "Data fingerprinting with similarity digests," in *IFIP International Conference on Digital Forensics*. Springer, 2010, pp. 207–226.

[14] ——, "An evaluation of forensic similarity hashes," *digital investigation*, vol. 8, pp. S34–S41, 2011.

[15] F. Breitinger, K. P. Astebøl, H. Baier, and C. Busch, "mvhash-b-a new approach for similarity preserving hashing," in *2013 Seventh International Conference on IT Security Incident Management and IT Forensics*. IEEE, 2013, pp. 33–44.

[16] Mandiant. (2014) Tracking malware with import hashing. [Online]. Available: https://www.fireeye.com/blog/threat-research/2014/01/tracking-malware-import-hashing.html

[17] N. Naik, P. Jenkins, and N. Savage, "A ransomware detection method using fuzzy hashing for mitigating the risk of occlusion of information systems," in *2019 IEEE International Symposium on Systems Engineering (ISSE)*, 2019.

[18] K. Savage, P. Coogan, and H. Lau, "The evolution of ransomware - Symantec," pp. 1–57, 2015.

[19] Y. Klijnsma. (2019) The history of Cryptowall: a large scale cryptographic ransomware threat. [Online]. Available: https://www.cryptowalltracker.org/

[20] Malwarebytes. (2019) Ransomware. [Online]. Available: https://www.malwarebytes.com/ransomware/

[21] Hybrid-Analysis. (2019) Hybrid Analysis. [Online]. Available: https://www.hybrid-analysis.com/

[22] Malshare. (2019) A free Malware repository providing researchers access to samples, malicious feeds, and YARA results. [Online]. Available: https://malshare.com/index.php

[23] VirusTotal. (2019) Virustotal. [Online]. Available: https://www.virustotal.com/#/home/upload

[24] N. Naik, P. Jenkins, N. Savage, and L. Yang, "Cyberthreat Hunting-Part 2: Tracking Ransomware Threat Actors using Fuzzy Hashing and Fuzzy C-Means Clustering," in *IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*. IEEE, 2019.

[25] P. Giordani and M. B. Ferraro, "Package FCLUST: Fuzzy Clustering," *CRAN R studio*, 2015.