Journal of COMPUTER AND FORENSIC SCIENCES

CFS 2024, Vol. 3, Issue 2, pp. 43–56
https://doi.org/10.5937/jcfs3-55513
Original research paper

A Clustering Approach to Malware Dataset Analysis

Slaviša Ilić[1*], Kristijan Kuk[2], Vladica Stojanović[2], and Igor Petrović[3]

[1] Department of Military Electronic Engineering, University of Defense, Belgrade, Serbia; ilic.slavisa@gmail.com

[2] Department of Information Technology, University of Criminal Investigation and Police Studies, Belgrade, Serbia;

kristijan.kuk@kpu.edu.rs; vladica.stojanovic@kpu.edu.rs

[3] Ministry of Defense of the Republic of Serbia; igor.petrovic@mod.gov.rs

* Corresponding author: ilic.slavisa@gmail.com

Received: December 17, 2024 • Accepted: December 25, 2024 • Published: December 30, 2024

Abstract: The research in the area of malware analysis is very popular, with an accent on machine learning algorithms that help automate this subject. One of the leading portals that help researchers with dataset problems is VirusTotal, providing free academic accounts with hundreds of thousands of malware samples with metadata. This work contributes with the analysis of 429,058 malware samples from VirusTotal in terms of overcoming the problem of inconsistent labeling of the antivirus scan results from different vendors. Two methods were used, LSA and LDA, both with automatic calibration of parameters, with the purpose of finding the optimal number of clusters – both resulting in 5. The graphical representation of the clusters was done by k-means clustering in two-dimensional space. Additional research on the most informative words in each cluster showed that 4 similar clusters could be reported as a result from both methods and one cluster per method (LSA and LDA) that was not related to the cluster in the opposite method. The showed results prove that the clustering approach to malware data analysis with automatic calibration of the parameters is a good method when dealing with inconsistent labels in the dataset.

Keywords: malware; analysis; clustering; unlabeled; virus total; LSA; LDA.

1. INTRODUCTION

The low effectiveness of traditional antivirus solutions in dealing with advanced computer viruses is a challenge for researchers [1, 2, 3, 4]. To avoid antivirus software, various techniques are used, such as code obfuscation, evasion techniques, fileless execution, etc. [5]. In order to overcome the weaknesses of antivirus solutions, researchers resort to various mechanisms of malware analysis, which can be roughly divided into static analysis (file or code of malicious software) and dynamic analysis of their behavior in specially prepared environments for analysis called "sandbox" systems [4].

In both types of research, machine learning and deep learning techniques are often used, with the aim of classifying and clustering malicious files into different malware categories. For this type of research, datasets with samples of malicious and benign software are of

Citation: S. Ilić, K. Kuk, V. Stojanović, and I. Petrović, "A Clustering Approach to Malware Dataset Analysis," *Journal of Computer and Forensic Sciences*, vol. 3, no. 3, pp. 44–57, 2024, https://doi.org/10.5937/jcfs3-55513.

great importance. One of the leading portals that helps malware researchers by providing the samples is VirusTotal (virustotal.com) [6]. For the submitted samples on the website, it provides over 70 results of antivirus checks on whether the requested file or "url" domain is malicious. VirusTotal provides a free service of sharing virus samples with an academic account, which was used to acquire the dataset of 429,058 samples described in this paper. This paper describes samples that are not labeled in terms of the category to which they belong (malicious or benign) or in terms of virus class but contain descriptions of antivirus companies according to their naming convention, which is mostly related to the behavior of the sample or its static code investigation. Main difficulties in the VirusTotal dataset analyses are the uncommon sample labeling policy from different antivirus companies, as well as the belonging of certain viruses to several categories (an example of labeling can be seen in Figure 1, as a value after "result:" keyword).

Researchers tried to overcome this problem in different ways. In [7], the 328 million VirusTotal reports for 235 million samples were analyzed with a yearly diversity perspective. The authors used different clustering methods and focused on the discovery of the malicious samples that were not properly labeled by VirusTotal. Even showing good results, their approach of creating 29,000 clusters does not categorize malware samples in well-know, human-understandable manner, and the authors focused on the set of features from VirusTotal and particular antivirus vendor, that might not be suitable for other researchers. In [8], the authors gave a clustering approach that leads to zero-day malware discovery, working on a publicly available dataset of the static malware file analysis. Contrary to the approach in this research, we tried to collect a custom dataset with the results of the antivirus vendors that are based on static file analysis and possibly the reverse malware engineering and dynamic analysis performed by different antivirus vendors collected together, all given through antivirus labels on the VirusTotal portal. Some authors used a similar approach to ours – in [9], the self-organizing maps (Kohonen maps) are used for clustering of the virus total dataset, in [10] the authors offered the novel clustering mechanism that predicts the number of the clusters on different datasets, and in [11] the BIRCH (Balanced Iterative Reducing and Clustering using Hierarchies) method is used on features extracted from both static and dynamic malware analysis.

While many authors recognized the need for clustering as one of the solutions for malware classification, very few of them offered the analysis of the inconsistently labeled data and results that belong to human-readable clusters. In this paper we tried to overcome this problem by contributing with the interpretation of the dataset and a two staged classification experiment that could be applied to similar VirusTotal or other datasets to better determine classes contained in the dataset. To further explore the established clusters, the most informative words are extracted for each cluster in order to have human understanding of the classes. Whole research is conducted on a workstation with an Intel i5-1145G7 CPU and 32GB memory), while JSON files and the database files were sorted on an SSD drive. The average computation time was around 6 hours.

The following section, MATERIALS AND METHODS, describes the dataset, followed with automatic calibration of latent semantic analysis (LSA) [12] and latent Dirichlet allocation (LDA) [13] parameters in the grid search process to determine the optimal number of clusters and to make the best classification of the samples. In the section RESULTS, the graphical and tabular results of the clustering are presented in the form of two-di-

mensional color figures and the list of words with the highest informational gain for each cluster. The section DISCUSSION gives additional class proposition based on the word description of the clusters, finding similarities between LSA and LDA clustering results. The CONCLUSION section summarizes the research and explains the possible future research direction.


```
"scans":
{
    "MicroWorld-eScan": {"detected": false, "version": "14.0.297.0", "result": null, "update": "20180313"},
    "nProtect": {"detected": false, "version": "2018-03-13.02", "result": null, "update": "20180313"},
    "CMC": {"detected": false, "version": "1.1.0.977", "result": null, "update": "20180313"},
    "CAT-QuickHeal": {"detected": true, "version": "14.00", "result": "Trojan.IGENERIC", "update": "20180313"},
    "McAfee": {"detected": false, "version": "6.0.6.653", "result": null, "update": "20180313"},
    "Cylance": {"detected": true, "version": "2.3.1.101", "result": "Unsafe", "update": "20180313"},
    "AegisLab": {"detected": true, "version": "4.2", "result": "Troj.W32.SchoolGirl.tnxl", "update": "20180313"},
    "TheHacker": {"detected": false, "version": "6.8.0.5.2523", "result": null, "update": "20180311"},
    "K7GW": {"detected": false, "version": "10.41.26485", "result": null, "update": "20180313"},
    "K7AntiVirus": {"detected": true, "version": "10.41.26487", "result": "Riskware ( 0040eff71 )", "update": "20180313"},
    "Baidu": {"detected": true, "version": "1.0.0.2", "result": "Win32.Trojan.WisdomEyes.160704O1.9500.9796", "update": "20180313"},
    "F-Prot": {"detected": true, "version": "4.7.1.166", "result": "W32/Betload.A.gen!Eldorado", "update": "20180313"},
    "Symantec": {"detected": true, "version": "1.5.0.0", "result": "Trojan.Gen.2", "update": "20180313"},
    "ESET-NOD32": {"detected": false, "version": "17050", "result": null, "update": "20180313"},
    "TrendMicro-HouseCall": {"detected": false, "version": "9.950.0.1006", "result": null, "update": "20180313"},
    "Paloalto": {"detected": true, "version": "1.0", "result": "generic.ml", "update": "20180313"},
    "Microsoft": {"detected": true, "version": "1.1.14600.4", "result": "Trojan:Win32/Tiggre!rfn", "update": "20180313"},
    "NANO-Antivirus": {"detected": false, "version": "1.0.100.21498", "result": null, "update": "20180313"},
    "ViRobot": {"detected": false, "version": "2014.3.20.0", "result": null, "update": "20180313"},
    "Avast": {"detected": true, "version": "18.2.3827.0", "result": "Win32:Malware-gen", "update": "20180313"},
    "Tencent": {"detected": false, "version": "1.0.0.1", "result": null, "update": "20180313"},
    "Ad-Aware": {"detected": true, "version": "3.0.3.1010", "result": "Trojan.GenericKD.40156721", "update": "20180313"},
    "Emsisoft": {"detected": true, "version": "4.0.2.899", "result": "Trojan.GenericKD.40156721 (B)", "update": "20180313"},
    "Comodo": {"detected": false, "version": null, "result": null, "update": "20180313"},
    "F-Secure": {"detected": false, "version": "11.0.19100.45", "result": null, "update": "20180313"},
    "DrWeb": {"detected": false, "version": "7.0.28.2020", "result": null, "update": "20180313"},
    "Zillya": {"detected": true, "version": "2.0.0.3510", "result": "Downloader.Betload.Win32.51", "update": "20180313"},
    "Invincea": {"detected": false, "version": "6.3.4.26036", "result": null, "update": "20180121"},
    "McAfee-GW-Edition": {"detected": false, "version": null, "result": null, "update": "20180313"},
    "Sophos": {"detected": false, "version": null, "result": null, "update": "20180313"},
    "Ikarus": {"detected": true, "version": "0.1.5.2", "result": "Trojan.Win32.Tiggre", "update": "20180313"},
    "Cyren": {"detected": true, "version": "5.4.30.7", "result": "W32/Betload.A.gen!Eldorado", "update": "20180313"},
    "Jiangmin": {"detected": true, "version": "16.0.100", "result": "Trojan.Generic.arpbl", "update": "20180313"},
    "Webroot": {"detected": true, "version": "1.0.0.400", "result": "W32.Malware.Gen", "update": "20180313"},
    "Avira": {"detected": false, "version": "8.3.3.6", "result": null, "update": "20180313"},
    "Kingsoft": {"detected": false, "version": "2013.8.14.323", "result": null, "update": "20180313"},
    "Endgame": {"detected": false, "version": "2.0.4", "result": null, "update": "20180308"},
    "Arcabit": {"detected": true, "version": "1.0.0.830", "result": "Trojan.Generic.D264BE31", "update": "20180313"},
    "SUPERAntiSpyware": {"detected": false, "version": "5.6.0.1032", "result": null, "update": "20180313"},
    "AhnLab-V3": {"detected": false, "version": "3.12.0.20130", "result": null, "update": "20180313"},
    "ZoneAlarm": {"detected": false, "version": "1.0", "result": null, "update": "20180313"},
    "Avast-Mobile": {"detected": false, "version": "180313-04", "result": null, "update": "20180313"},
    "TotalDefense": {"detected": false, "version": "37.1.62.1", "result": null, "update": "20180313"},
    "VBA32": {"detected": false, "version": "3.12.28.0", "result": null, "update": "20180313"},
    "AVware": {"detected": false, "version": "1.5.0.42", "result": null, "update": "20180313"},
    "MAX": {"detected": true, "version": "2017.11.15.1", "result": "malware (ai score=97)", "update": "20180313"},
    "Zoner": {"detected": false, "version": "1.0", "result": null, "update": "20180313"},
    "Rising": {"detected": false, "version": "25.0.0.1", "result": null, "update": "20180313"},
    "Yandex": {"detected": false, "version": "5.5.1.3", "result": null, "update": "20180313"},
    "SentinelOne": {"detected": true, "version": "1.0.15.206", "result": "static engine - malicious", "update": "20180225"},
    "eGambit": {"detected": false, "version": "v4.3.5", "result": null, "update": "20180313"},
    "Fortinet": {"detected": true, "version": "5.4.247.0", "result": "W32/PossibleThreat", "update": "20180313"},
    "AVG": {"detected": true, "version": "18.2.3827.0", "result": "Win32:Malware-gen", "update": "20180313"},
    "Panda": {"detected": true, "version": "4.6.4.2", "result": "Tri/CI.A", "update": "20180313"},
```


Figure 1. *Part of the description from the .json files in the dataset for one of the samples.*

2. MATERIALS AND METHODS

2.1. Dataset

The dataset was obtained from the virustotal.com portal in the form of multiple password-protected archives, organized annually, from 2017 to 2021. The decompression was done in a Python script (due to two types of passwords used in the compression process and a large number of archives), resulting in a dataset organized by year with subdirectories for the file extension. Although a large number of data types are represented (46 in total), 74% of the data set consists of .exe files (shown in Figure 2), which makes this dataset imbalanced. This is expected since users are most suspicious of executable files and most likely to check them the most.

The total number of instances in the dataset is 429,058, and they are located on the computer disk in the form of files without an extension (for security reasons), only with a file name that is the SHA256 representation of the file. A description file with the same name

but with the extension ".json" (hereinafter referred to as JSON files) is attached to the sample. It contains a description of the sample and the results of the analysis with VirusTotal (shown in Figure 1). Besides the scan results that provide the detection information (which is used for clustering), the following attributes from the JSON files are also interesting:

– FileTypeExtension – information about the sample extension;
– Positives – the number of antivirus vendors that marked the sample as malicious;
– Total – the number of all antivirus vendors that were in use at the time of submission;
– Dates – date of submission;
– Permalink – permalink to download sample from the VirusTotal portal.

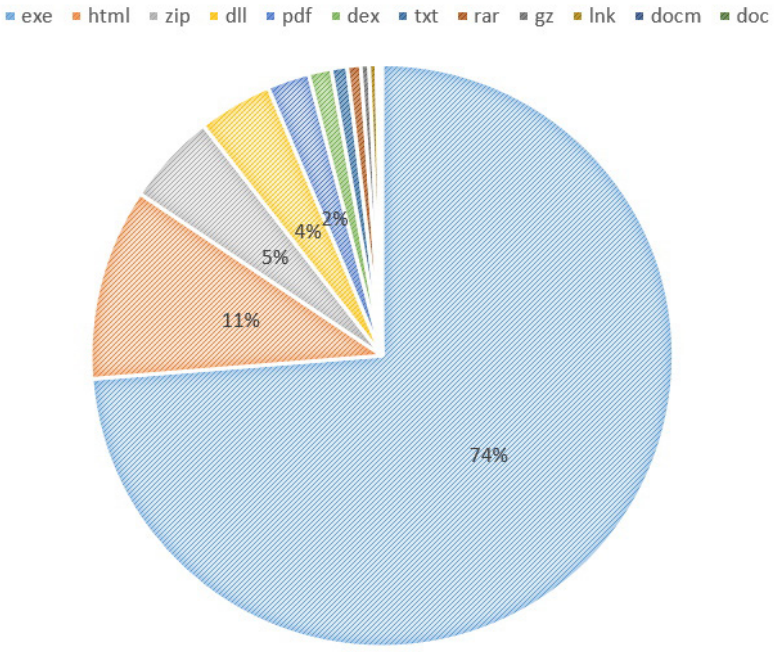

Figure 2. *Balance of the data set.*

In order to better analyze the dataset, two forms of data representation were used:

– The database with basic data about the samples, which later could be quickly consulted during the processing,

– Pandas data frames that could be easily loaded with the data from the database and used in memory, as well as saved in the form of "pickle" files on the computer disk for later faster loading, without parsing.

2.2. Database

In order to not always collect data from JSON files describing a malicious sample (due to slowness), it was more convenient to save the once found data set (obtained by regular expressions on JSON files one by one) in such a way that it could be quickly accessed. In order to realize this, the database was created with the attributes whose explanations are shown in Table 1.

Table 1. *Column names in the database with explanations.*

Column name	Description
SHA256	Hash function (SHA256) value of the sample file
ScanId	VirusTotal unique scan identifier
ScanDate	The date of the scanning results described in JSON file
FileTypeExtension	The extension of the sample scanned
TimeSubmited	The number of times the sample was submitted to virus total
Positives	The number of antivirus vendor that marked the sample as malign
Total	The number of antivirus vendors available when the sample was submitted
PermaLink	The link to the VirusTotal scanning information
AlYac	The name of antivirus vendor (each available vendor has its own column with the results of scanning).
APEX	
AVG	
Acronic	
AdAware	
…	

The most important attributes of the "scans" part (Figure 1) are the name of the antivirus vendor (for example, AVG, Avast, Kaspersky, Trend Micro, etc.), the value "detected" (true or false), and "results" with a string containing "null" if not detected as malicious or a string describing the malware (for example: unsafe, malware, risky, malicious, Win32. Trojan.WisdomEyes.16070401.9500.9796, Bot, Botload, Malwaregen, etc.).

The way in which the data was entered into the database is the deserialization of the JSON files with, i.e., conversion into objects and filling the database. The main challenge here was the fact that the JSON format was not the same for all files, so two types of deserialization were used with two types of regular expressions.

The content of each .json file with the description of the sample scan is loaded into memory, after which the "re" library in the Python programming language is applied, which allows using the following regular expression in the findall() function:

results = re.findall(r"result\": [a-zA-Z0-9\.\s:\-\"\+\[\]()/@!=]*, \"update", json_file_content)

A regular expression returns all strings that satisfy the condition that it starts with "result," followed with a string that could contain letters or numbers, dots, spaces, characters like - " + [ ] ( ) @ ! = , all present zero or multiple times, written as [a-zA-Z0-9\.\s:\-\"\+\[\]()/@!=]*, and finishing with "update".

With additional preprocessing that removes quotation marks, each string is added to the list of strings that has from 57 up to 70 elements, depending on the number of antivirus vendors that reported the sample, which is later added to the database.

Besides the described attributes in the database, an interesting parameter that can be calculated is the number of positives compared to the number of total detections [4]. When positives are divided with total detections, the result gives the value of sample maliciousness. The top ten maliciousness values are given in Table 2, which can propose the idea of the file types that are mostly used to carry malware. In the future research, if the full year dataset could be provided by VirusTotal, the whole year trend could be calculated, which could give more useful information about the extensions mostly used by malware developers to spread the malware.

Table 2. *Top ten maliciousness values per file type.*

	Average maliciousness values	Standard deviation	File type extension	Number of samples
1.	0.75	null	pptx	1
2.	0.73	0.17	exe	320,767
3.	0.71	0.16	html	46,407
4.	0.70	0.19	dll	17,925
5.	0.66	0.11	pdf	10,148
6.	0.59	0.09	docm	704
7.	0.58	0.15	pl	7
8.	0.56	0.15	doc	613
9.	0.55	0.15	lnk	1,794
10.	0.55	0.04	gif	36

From Table 2, it can be concluded that the samples with the highest maliciousness are hidden in the types: .exe, .html, .dll, etc. When interpreting the results, types that do not have a sufficient number of representatives in the corpus (e.g., .pptx, .pl, etc.) should be omitted.

The database represents a good starting point for future research, given that it can quickly answer questions about the most important characteristics of the sample.

2.3. Preprocessing

When loading data from the database using the "pandas" library in the Python programming language, the dataset was created that consists of SHA256 identifiers and a list of results from the antivirus manufacturer. A part of the data frame can be seen in Table 3.

When loading data from the database, the results of 17 antivirus manufacturers were deliberately omitted as useless for clustering: APEX, Acronis, CrowdStrike, Cybereason, Cylance, Cynet, Endgame, FireEye, Invincea, MAX, MicroWorldeScan, Paloalto, Sangfor, SentinelOne, Sophos, Trapmine, and eGambit. The listed manufacturers do not identify the type of maliciousness in their description but describe the degree of danger or similar.

Table 3. *Tabular representation of the contents of the part of the database in which only the name of the sample, scan results of the first two and the last two antivirus vendors are displayed.*

SHA256	ALYac	APEX	…	Zoner	eGambit
252b851489b1824d16f7b744d083faeaf-729e5bdadfdca54652fb4232e80cc48	Generic.Malware.SFYd.E4353219	Malicious	…	NULL	NULL
8276880249a1bb51652a219840d57f642fc-de71ac8b6775f735e5a44f713185f	NULL	NULL	…	NULL	NULL
2567580043e224dd764ae0186241cd-99d9e401a7953fd1d9dd4cdb8452d6a1ff	Win32.Virlock.Gen.1	Malicious	…	Packer.Win32.Virlock	Unsafe.AI_Score_93%
3ab9cd3ca0795be208a120ed7d-e29f051062d04b3841815d58ad-f5402b59672f	NULL	NULL	…	NULL	NULL
cb124719117d8a8f401c7eacea93aea-741bab39491bd856271ceaf50fd21b42b	NULL	NULL	…	NULL	NULL
54d6904f4566842b4bd68632904c-cf1a3cf99d626029d758c4440409e5b3a43a	Win32.Virlock.Gen.1	Malicious	…	Packer.Win32.Virlock	Unsafe.AI_Score_98%
3b52694ff8831f787d565cb3b3199355f-4396c3c4fd5ffab269f9815c7f65675	NULL	NULL	…	NULL	NULL
2bf7ca3014ac62f3206ed6b3953a1400d-a9c368ca08332b3290b4686de99caf5	Gen:Variant.Graftor.473934	NULL	…	NULL	NULL
815d4c03173befb3fcc3a471889e33cb-395c615878eb902c8e73d5db0e5e3877	Gen:Variant.Razy.173713	NULL	…	NULL	NULL
7bc685215e357bea7eaca7dea953b2e283c-2805cca923813a5c3e8b543cfa649	NULL	NULL	…	NULL	NULL
6ca02d8610d50bb06f1c21d4f6e947f91b-481b0396a551b61de35b39937ccd6e	Gen:Variant.Zusy.279528	NULL	…	NULL	NULL
0843a17c1a3d1582959da8962b34a6f-fb8071b488620ed7e9feb2da04316bcbe	Trojan.GenericKD.40409647	NULL	…	NULL	NULL
0b17c081060be4975e1fc41f929b-27c30077b471da401aef273ebb96631f1e96	Trojan.Ransom.TeslaCrypt	NULL	…	NULL	NULL
56a8f5aec963c428667e04f92ddbe4c52df-78f766ad72c8ebcb3fcf69302112f	Trojan.HTML.Ramnit.A	NULL	…	TrojanDownloader.VBS	NULL
55a54ba1a84344779a5369b4d6e575f-c498a0169589aa0a4f223d1d8422aee2d	Win32.Virlock.Gen.8	NULL	…	NULL	NULL
3ba48db41d493335ce40dde-aeca9e3eed78b319b5d70d5c-458d11146787146ce	Gen:Variant.Adware.Dropper.103	NULL	…	NULL	malicious_confidence_99%
4a90dc08b5dac88cbe01fb9388d9e771dc-87d76c11baa894cedb37c73370f862	Win32.Viking.AY	NULL	…	Win32.Viking	NULL
30abed3c3e61f39c4ad-2ac01b9dd7208df97f6bdac6bd93a91cfc-59687d6aaa9	Win32.Virlock.Gen.1	Malicious	…	Packer.Win32.Virlock	Unsafe.AI_Score_99%

After obtaining the pandas data set (data frame), additional editing of the data set was performed by converting it into a string and eliminating certain parts of the text with the next regular expression function:

```
def clean_text(text):
```

```
text = re.sub(r'[^a-zA-Z\s]+|nan|\b[a-zA-Z]\b', ' ', text)
clean_text = re.sub(r'\s+', ' ', text)
very_clean_text = clean_text.strip()
return very_clean_text
```

The mentioned function removes everything in the text that is not letter marks, as well as strings of characters "nan" and any letter that stands alone – since it has no informative value for the clustering. After that, spaces were also removed. The text obtained in this way was vectorized using "TF-IDF" (term frequency – inverse document frequency) of the scikit-learn library [14], which is a common way of extracting features in the Python programming language. TF-IDF is a numerical statistic that describes the importance of words in a document that is part of a collection and is described in detail in [14].

In addition to the pre-processing by which individual strings and parts of strings were removed, the words that have no informational gain in the data set were noticed and removed by using the concept of "stop words," which is applied in the next step – vectorization, i.e., distinguishing features.

By configuring the feature extraction on all features (max_features = None) and by setting the parameter max_df = 0.98, the words found in over 98% of the corpus are not taken into account. There is also a min_df parameter, often called a cut_off parameter in literature, which could be used to indicate that words that appear in less than a certain percentage of documents are not used. This parameter is not used because such words could have greater informational gain related to clustering.

2.4. LSA with Automatic Calibration Method

The LSA (latent semantic analysis) method transforms the multidimensional textual representation (TF-IDF matrix) into a smaller set of dimensions that better represent the essence and hidden themes in the data. This technique is useful for reducing data complexity and improving clustering results. LSA uses singular value decomposition (SVD) on the TF-IDF matrix to identify latent semantic structures in the data. LSA assumes that documents and concepts can be represented in a reduced number of dimensions, where the dimensions are combinations of the original concepts. It emphasizes reducing dimensions and highlighting semantic similarity between documents and concepts. LSA is a linear model that attempts to determine hidden structures in data and is based on term-document matrix analysis.

In this paper, the automatic calibration of LSA parameters is implemented, which means that the best values for the number of dimensions in LSA and the number of clusters in the k-means algorithm [15] were automatically searched. The basic idea is to experiment with different numbers of dimensions and clusters, measure the clustering quality, and choose the best combination based on a certain metric.

By iterating through different numbers of clusters for each number of dimensions in LSA, the different numbers of clusters in the k-means algorithm were taken, from 1 to 14 in steps of 1. More than 14 malware families were not expected to be in the dataset (based on the domain knowledge of usual malware families). By evaluating the clustering for

each combination of LSA dimensions and the number of clusters, the validity of the silhouette (silhouette score) was calculated, which measures how similar documents within one cluster are to each other in relation to documents in other clusters. The value ranges from -1 to 1, where higher values indicate better clustering. The configuration that gives the highest silhouette value was considered to be the best. The graphical presentation of the results obtained by this method can be seen in Figure 3.

2.5. LDA with Automatic Calibration Method

LDA (latent Dirichlet allocation) is a statistical model used to discover latent (hidden) topics in a corpus of documents. In LDA, it is assumed that documents originate from a certain number of latent topics. It uses Bayesian methods to estimate the probability distribution of topics and words in documents and represents each document as a mixture of those topics (the percentage of each topic appearing in the document). These probabilities reflect which words are most typical for given topics.

Similar to the previous experiment (LSA), the range of the number of clusters is from 1 to 14. The goal is to find the optimal number of clusters that best separate the data. To search for the optimal number of clusters, the silhouette score was used again.

3. RESULTS

3.1. LSA Method Results

The results of the LSA method are shown in Figure 3, where it can be seen that the algorithm determined the optimal number of 5 clusters and that the clusters are quite well separated except for two smaller clusters shown in orange and red, which can be claimed by visual inspection to be a subset of the purple cluster.

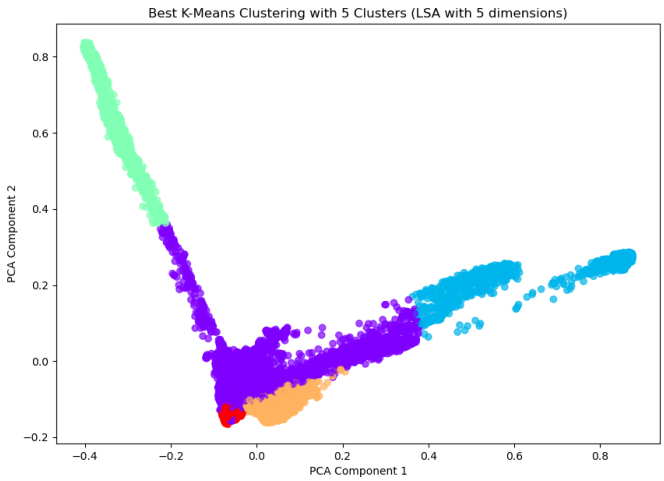

Figure 3. *Graphic representation of five clusters created by automatic calibration of LSA algorithm parameters.*

The research is further conducted to show which words are the most related to which cluster, with sorting words by the highest information gain for a particular cluster (among the given five). The most informative words per cluster are given in Column 2 of Table 4, while the number of samples per cluster is given in Column 3 of Table 4.

Table 4. *LSA method – words with the highest information gain for each cluster (Column 2) and number of samples per cluster (Column 3).*

Cluster	Words with the highest information gain per cluster	Samples per cluster
0	worm, virut, agent, virus, variant, hematite, generickd, malware, heur, adware	306,726
1	ramnit, vbs, html, dropper, virus, bp, script, nimnul, js, htm	34,266
2	virlock, virus, polyransom, nabucur, virransom, fa, pe, gena, crypt, ransom	29,284
3	js, exploit, fakejquery, script, html, pdfka, pdf, iframe, redirector, downloader	37,324
4	android, androidos, riskware, adware, agent, shedun, pup, smsreg, triada, smspay	21,458

By visual inspection of the database content (described in Subsection 2.2.), it could be concluded that the word "Trojan" is dominant in most samples, but according to Table 4, the algorithm apparently omits that word, since it is too common and its information gain is small. Also, it could be seen that in cluster 0, which has the most samples, the word selection of the algorithm includes words that can refer to a large number of samples, and that cluster is placed in the central part of the graphic representation in Figure 3, while the other clusters are marked with words that are rare and are further away from the center on the graphic display.

3.2. LDA Method Results

From Figure 4, which represents the LDA method with auto calibration of parameters, it could be concluded that the optimal number of clusters determined by the automatic calibration of the LDA parameters is 5, similar to the previous method. However, the difference in relation to the LSA method could be seen in both the graphical representation and in Table 5. The difference in the visual graphic representation comes from the mathematical and conceptual differences of the algorithms.

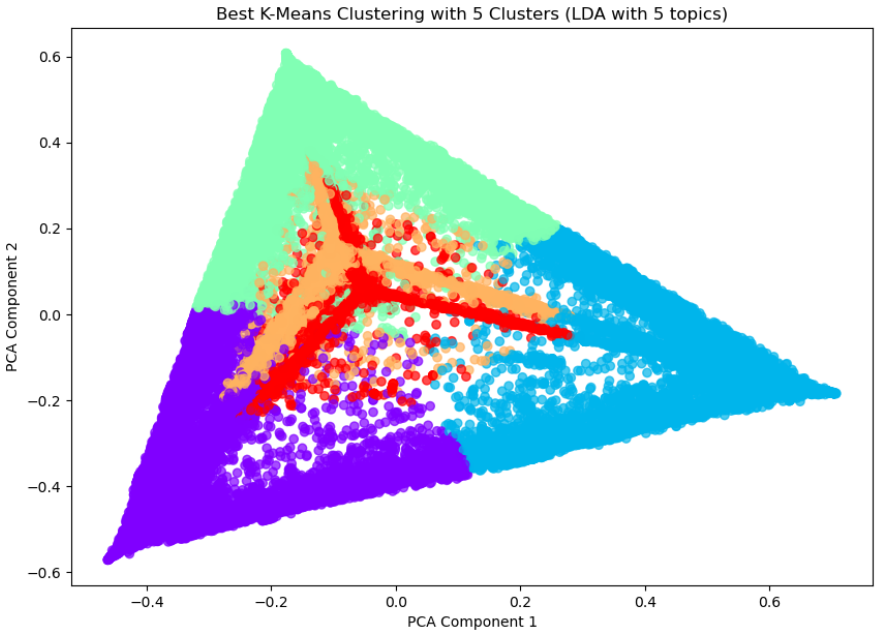

Figure 4. *Graphic representation of five clusters created by automatic calibration of LSA algorithm parameters.*

The most informative words per cluster are given in Column 2 of Table 5 (different from the LSA algorithm), while the number of samples per cluster is given in Column 3 of the same table with a more balanced distribution among clusters.

Table 5. *LDA method – words with the highest information gain for each cluster (Column 2) and number of samples per cluster (Column 3).*

Cluster	Words with the highest information gain per cluster	Samples per cluster
0	js, exploit, variant, agent, pdf, msil, kryptik, downloader, heur, fakejquery	94,685
1	virlock, android, virus, worm, vobfus, polyransom, androidos, nabucur, agent, adware	103,409
2	generickd, variant, agent, sality, dinwod, adware, allaple, razy, malware, downloader	92,942
3	virus, hematite, backdoor, js, agent, coinminer, qukart, worm, infector, wabot	75,856
4	ramnit, vbs, html, virut, js, virus, dropper, script, bp, nimnul	62,166

4. DISCUSSION

This research tried to overcome the problem of inconsistent labels of the antivirus reports by applying the clustering method. It is shown that the optimal number of clusters could be found with automatic parameter calibration for LSA (latent semantic analysis) as well as LDA (latent Dirichlet allocation). Both methods resulted in the number 5 as the optimal number of clusters. Additionally, the two-dimensional space plotting was applied for human perceptions of the clusters.

By analyzing the results shown in Tables 4 and 5, it stands that both methods omitted some words (i.e., "Trojan") as words with little informational gain, which is present in most samples and could be determined by visual inspection of the database. By further comparison of the two tables, additional information about each cluster could be derived in the next form:

– The biggest cluster determined from the LSA method (cluster number 0) and the cluster number 2 in the LDA method both represent a general class that contains words that describe malicious software in a general sense and that could be the subject of further analysis.

– Viruses of the "Rammnit" class, which represent light Trojans (usually later weaponized with additional download of "heavier" components), are intended for stealing the victim's banking and other information or for simple ransomware requests after encrypting the data. This "class" is cluster number 1 in the LSA method and number 4 in the LDA method.

– Viruses of the "Ransomware" class that encrypt the victim's system, demanding a ransom. With the LSA method, cluster number 2 belongs to this class, and cluster number 1 in the LDA method. The word "Nabsur" represents one such virus.

– Several classes that are included together in a cluster, and refer to viruses that use java script, html script, and similar are presented in cluster number 3 provided with the LSA method and with number 0 in the LDA method.

– In the last class, the algorithms gave different results. According to the word values, it could be said that this is a cluster of Android viruses as far as the LSA algorithm is concerned, as well as a class of Trojans and "miners" as far as the LDA algorithm is concerned. With the LSA method, this is cluster numbered 4, and with the LDA method, it is cluster 3.

5. CONCLUSION

This work presented the VirusTotal dataset of 429,058 malware samples and gave an idea of how to collect useful data and preprocess it before making a data frame out of it. Two clustering mechanisms, LSA and LDA, both with automatic calibration of parameters, were applied with the purpose of optimal number of clusters determination, resulting in both with 5 clusters as optimal (Figures 3 and 4).

Additional research on the most informative words in each cluster (Tables 4 and 5) showed that for some clusters, both LSA and LDA gave words with similar meaning, from which the existence of the real (enough distinct) classes in the dataset could be concluded. The first such class was described as a general class with most samples in it, and 3 additional classes of malware were described as "Rammnit," "Ransomware," and "java-html". In one

cluster case, methods gave different results where LSA most deterministic words described it as "Android" in general and LDA as "Trojans" and "miners" in general.

This paper contributes with an approach to classify the malware data, which is inconsistently labeled, and classify it into some basic classes. Future research could be conducted that treats the different words with similar meanings in the same way, which could lead to even better clustering.

FUNDING:
This research received no external funding.

INSTITUTIONAL REVIEW BOARD STATEMENT
Not applicable.

INFORMED CONSENT STATEMENT
Not applicable.

CONFLICTS OF INTEREST:
The authors declare no conflict of interest.

REFERENCES

[1] J. Greig, "Cybercriminals raking in $1.5 trillion every year," (2020). TechRepublic. [online], available at: https://www.techrepublic.com/article/cybercriminals-raking-in-1-5-trillion-every-year/. [Accessed: 24 December 2024].

[2] PurpleSec company, "The Ultimate List of Cybersecurity Stats Data, & Trends," PurpleSec. [online], available at: https://purplesec.us/resources/cybersecurity-statistics/. [Accessed: 24 December 2024].

[3] M. McGuire, Into the web of profit, An in-depth study of cybercrime, criminals and money, Book, Project funded by Bromium, Inc., available at: https://www.bromium.com/wp-content/uploads/2018/05/Into-the-Web-of-Profit_Bromium.pdf . [Accessed: 24 December 2024].

[4] S. Ilić, M. Gnjatović, B. Popović, N. Maček (2022). A pilot comparative analysis of the Cuckoo and Drakvuf sandboxes: an end-user perspective, Military Technical Courier, https://doi.org/10.5937/vojtehg70-39196

[5] S. Ilić, M. Gnjatović, B. Popović, I. Tot, B. Jovanović, N. Maček, and M. Gavrilović Božović. (2024). "Going beyond API Calls in Dynamic Malware Analysis: A Novel Dataset," *Electronics* 13, no. 17: 3553. https://doi.org/10.3390/electronics13173553

[6] G. Sood, Virus Total web portal [online]. Available at: https://www.VirusTotal.com. [Accessed: 24 December 2024].

[7] K. van Liebergen, J. Caballero, P. Kotzias, C. Gates. A Deep Dive into VirusTotal: Characterizing and Clustering a Massive File Feed. (2022). doi:10.48550/ARXIV.2210.15973

[8] O. Jurečková, M. Jureček, M. Stamp, Classification and online clustering of zero-day malware. J Comput Virol Hack Tech 20, 579–592 (2024). https://doi.org/10.1007/s11416-024-00513-5

[9] R. S. Pirscoveanu, M. Stevanovic and J. M. Pedersen, "Clustering analysis of malware behavior using Self Organizing Map," 2016 International Conference on Cyber Situational Awareness, Data Analytics and Assessment (CyberSA), London, UK, 2016, pp. 1–6, doi: 10.1109/CyberSA.2016.7503289.

[10] V. Petrosyan and A. Proutiere, "Viral Clustering: A Robust Method to Extract Structures in Heterogeneous Datasets", presented at the Thirtieth AAAI Conference on Artificial Intelligence (AAAI-16), February 12–17, Phoenix, USA, 2016, pp. 1986–1992. https://kth.diva-portal.org/smash/record.jsf?pid=diva2%3A898710&dswid=8986

[11] G. Pitolli, G. Laurenza, L. Aniello, L. Querzoni, and R. Baldoni. MalFamAware: automatic family identification and malware classification through online clustering. (2021). International Journal of Information Security, 20(3), 371–386. https://doi.org/10.1007/s10207-020-00509-4

[12] T. K. Landauer, P. W. Foltz, and D. Laham, Introduction to Latent Semantic Analysis. (1998).Discourse Processes, 25, 259–284.

[13] D. M. Blei, Y. Ng. Andrew, and M. I. Jordan. Latent Dirichlet allocation. 2003. J. Mach. Learn. Res. 3, null (3/1/2003), 993–1022.

[14] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, and V. Dubourg. Scikit-learn: Machine Learning in Python. J. Mach. Learn. Res. 2011, 12, 2825–2830. Available online: http://jmlr.org/papers/v12/pedregosa11a.html [Accessed: 24 December 2024]

[15] J. Ortega, N. Almanza-Ortega, A.Vega-Villalobos, A. R. Pazos-Rangel, R. Diaz, Z. Diaz, J. Crispin, and A. Martínez-Rebollar. The K-Means Algorithm Evolution. (2019). doi: 10.5772/intechopen.85447