IAES International Journal of Artificial Intelligence (IJ-AI)
Vol. 14, No. 3, June 2025, pp. 2102~2110
ISSN: 2252-8938, DOI: 10.11591/ijai.v14.i3.pp2102-2110

A new wrapper feature selection approach for binary ransomware detection

Omar Chaieb[1], Nabil Kannouf[2], Mohammed Benabdellah[1]
[1]Arithmetic, Scientific Computation and Applications Laboratory, Faculty of Sciences, Mohamed Premier University, Oujda, Morocco
[2]Applied Sciences Laboratory, National School of Applied Sciences, Abdelmalek Essaadi University, Tetouan, Morocco

Article Info

Article history:

Received Feb 18, 2024
Revised Feb 11, 2025
Accepted Mar 15, 2025

Keywords:

Anomaly detection
Behavior analysis
Feature selection
Machine learning
Ransomware

ABSTRACT

Concerns about ransomware attacks have heightened in recent years for both individuals and organizations. Detecting these malicious attacks poses considerable challenges for cybersecurity professionals, particularly due to their ever-evolving nature. Although behavior-based detection methods show promise in recognizing new ransomware variants, they face significant hurdles, especially in managing the massive volumes of data generated from real-time malware behavior monitoring, leading to high dimensionality. This paper introduces a new feature selection approach specifically for binary ransomware detection. Our method emphasizes assessing the impact of feature categories on the effectiveness and speed of detection algorithms. It involves two stages: the first stage selects the most relevant groups (categories) of features, while the second ranks and identifies the important features within those categories. Experimental results indicate that our approach surpasses similar studies regarding accuracy and ability to minimize the original features set. Moreover, both computation speed and accuracy are notably enhanced when using the selected subset compared to the original features.

Corresponding Author:

Omar Chaieb
Arithmetic, Scientific Computation and Applications Laboratory, Faculty of Sciences
Mohamed Premier University
Oujda, Morocco
Email: omar.chaieb.men@gmail.com

1. INTRODUCTION

Universal cyberattacks have grown exponentially in recent years with new sophisticated and hard-to-detect cyberattacks such as malware [1], insider attacks [2], advanced persistent threats (APTs) [3], and distributed denial of service (DDOS) attacks [4]. Among all these destructive attacks, ransomware is considered as one of the most dangerous and harmful cyberattack [5]. The situation becomes more serious when attackers adopt the ransomware-as-a-service (RAAS) model, using identities stolen from the dark web to launch their attacks [6], [7]. In addition, the number of ransomware victims reported in March 2023 was 1.6 times higher than that in the peak month of 2022, proving that many cyber security players who believed they had succeeded in stopping ransomware attacks were mistaken [8].

We can define ransomware as a special kind of malware that silently encrypts a victim's data or lock down the system in order to stop the victim from using it [9]. Afterwards, the attackers request that the victim pays a ransom in digital currency so that he can get his data back. Like most malicious programs, ransomware can be contained through two main measures: detection and prevention. Of the two approaches, detection is more effective [10].

Furthermore, when it comes to variants and stealthy attacks like ransomware, machine learning (ML) techniques can surpass several limitations shown by signature-based detection methods [11] by successfully exploiting advanced capabilities such as bytecodes, API invocations [12] and behavioral reporting extracted through static or dynamic malware analysis [13], [14]. However, due to the large amount of data contained in behavior-based ransomware datasets, ML algorithms are not good at predicting whether an event is a normal or malicious activity [15]. For this reason, feature selection (FS) methods are used to decrease the volume of data by removing irrelevant features to enhance the efficiency of ML algorithms [16], [17].

The issue statement in this research study is that as ransomware attacks are destructive and irreversible, they need to be detected quickly and effectively. Besides, taking into consideration that large datasets limit the performance of ML-based solutions, especially when data comes from behavior-based malware analysis. By employing a new wrapper-based FS technique to decrease the volume of the original dataset, the main objective of this work is enhancing responsiveness, accuracy, and speed of classification algorithm.

We hypothesized that choosing the most relevant features categories while ignoring others would cast a shadow over the performance and speed of the ML algorithms. To validate this hypothesis, we suggest a new approach for evaluating the efficacy of distinct feature groups (categories), and then we built an optimal set containing the most relevant features from selected groups. The proposed method includes two stages: the first is devoted to select the most pertinent group of features, and the second is dedicated to rank and select the relevant features within the chosen groups. Additionally, when doing traffic behavior analysis and subsequent variable extraction, this new group-based feature reduction technique will enable us to focus only on the most crucial feature categories and dismiss those that are less important.

This article is divided into five sections: section 2 gives background information and evaluates related literature. The methodology is highlighted in section 3. Results and discussion are shown in section 4. Conclusions and upcoming projects are covered in the final part.

2. RELATED WORKS

This section provides some background information and related works. We first highlight the different types of ransomware attacks, detection techniques, ML and FS approach used to improve detection. Then, we will give a state of art of ransomware detection and FS approaches for data with high dimensions.

2.1. Types of ransomwares

We can classify ransomware based on a variety of factors, such as their method of encryption and how they are carried out. Ransomware attacks can be broadly divided into three types: scareware, locker ransomware, and crypto-ransomware. Scareware employes techniques such as social engineering or phishing to trick victims into providing valuable information, purchasing, or downloading dangerous software. It mostly serves for collecting data and displays warning messages to alert you that your device is the subject of an attack [18]. Locker-ransomware, which does not affect data but prevents users from using their system by displaying pop-up windows that never close or by locking their desktop [10], [19]. Crypto-ransomware is regarded as one of the most dangerous ransomwares, it encrypts files on the victim's device and renders them useless by employing powerful encryption algorithms that are difficult to detect [20], [21].

2.2. Ransomware detection approaches

To reduce the risk of ransomware attacks, we might use either a preventative or a detection approach [22]. The first seeks to prevent ransomware damage before it occurs by imposing strong access controls [23], [24], securing data and passwords and boosting user awareness [25] whereas the second train and test a detection model utilizing historical data and signatures from previous attacks. Furthermore, dynamic analysis can be used as a detection tool to locate ransomware by examining the behavior of the malware while it is running, or static analysis to detect ransomware by analyzing the contents of binaries without running the software [26]. However, because human log monitoring takes time and specific knowledge, data gathered during static or dynamic analysis such as network traffic and API invocations are used to feed ML algorithms.

ML techniques may overcome several limits of signature-based detection approaches, however given that ML algorithms are recognized for learning from previous data, this data needs to be prepared before using it, it is crucial to clean, normalize, and filter the data. One of the most important methods during the data preparation phase is the FS [16], [27]. This technique can be divided into filter and wrapper. Regardless of the data modeling algorithm employed, the former chooses features based on performance measurement, whereas the latter selects the relevant features by evaluating their performance through a learning algorithm [28]. In this study, to solve the FS problem for high-dimensional data, we combine the best properties of filters and wrappers by using feature group-based hybrid techniques to achieve high accuracy and high speed typical of wrappers, i.e., filters characteristics.

2.3. Related works

Sgandurra *et al.* [29] proposed a novel method named EldeRan to identify key dynamic characteristics of ransomware, which are then used for ransomware detection. The method consisted of two components: monitoring and a ML process. The first component dynamically analyzed traces of samples from two datasets in a sandbox environment: ransomware and goodware. From these records, EldeRan retrieved and parsed seven functional classes (Windows API calls, deleted files, and directory operations). The ML part consisted of two stages: FS and classification. For FS, the authors used mutual information (MI) as a filter-based FS technique. For classification issues, EldeRan used a logistic regression (LR) classifier. Furthermore, the authors compared the accuracy of LR algorithm to that of other ML algorithms, such as support vector machines (SVM) and naive Bayes (NB). The suggested method performs well over NB and rivals SVM. Moreover, as the authors mentioned, MI algorithm has proven to be a powerful method for automatic FS process. However, this method required a preset number of features to be selected as input, and uses the entire dataset as a starting point when performing FS instead of keeping only the most relevant groups of features, which we assume will increase the performance of validation algorithms. Furthermore, authors didn’t compare the results obtained using selected features with the original features in terms of accuracy and processing time.

Abbasi *et al.* [30] proposed an automatic wrapper-based FS approach for behavior-based ransomware detection. This approach used a new group-based strategy to address the issue of data with high dimension. The strength of this technique is that, unlike the filter method used in [29], it didn’t use a preset amount of features to be selected as input. Furthermore, to illustrate the influence of each feature category on classification accuracy, they used a group-oriented strategy consisting of two stages. Using the MI algorithm as a feature ranking technique, a similar amount of the best-ranked features is chosen from every feature category in the first stage. While in the second stage, a wrapper-based FS method is utilized to choose an ideal number of relevant features from every category using both LR and particle swarm optimization (PSO) algorithms. To avoid overfitting, this method used the LR as a wrapped algorithm during the FS process, while they used other algorithms such as SVM, K-nearest neighbor (KNN), random forest (RF), and decision tree (DT) to evaluate the generalizability of the selected features. Although the authors managed to fully automate the process of variable selection in this method, the number of variables selected in [29] (400 features) was less than in this method (823 selected features), and if this does not affect the performance of the algorithm, then doing so will definitely affect the response time, which is intolerable when it comes to ransomware detection. In addition, the method used the entire dataset as a starting point when performing FS instead of keeping only the most relevant groups of features, which we assume will increase the performance of detection algorithm. Furthermore, authors didn’t compare the results obtained using selected features with the original features in terms of accuracy and processing time.

3. METHOD

Starting with the group-based strategy for FS adopted by [30], which showed that individual feature groups have a significant influence on the classification performance. We assumed that choosing the most relevant feature groups while ignoring others would cast a shadow over the performance and speed of the ML algorithm. To validate this hypothesis, we suggest a new approach for evaluating the efficacy of distinct feature groups, and then we built an optimal set containing the most relevant features from chosen groups. The FS approach adopted in our paper is divided into two different stages. The first stage aims to rank and select the most relevant feature category using LR as an evaluation algorithm while the second one consists of ranking and then selecting an equal number of the most significant features from each category. Each stage will be composed of two different phases, one for the ranking and the other for the selection. The process of our method is presented in Figure 1.

3.1. Stage 1: category ranking and selection

This step of the approach is divided into two parts: one will be dedicated to rank feature groups, and the other to the selection of feature groups. Ranking feature groups involves applying the LR method to each group individually to classify them from the category with the highest performance to the one with the lowest score. After all groups have been rated, the LR algorithm will be applied again, but this time on group subsets, starting with the subset comprising the two best-ranked groups obtained in phase 1. During each iteration, a new group is added to the subset of groups while maintaining the ranking obtained during the first phase, then the new generated subset is evaluated using the LR algorithm. In addition, to measure the effectiveness of each sub-set of groups, the LR algorithm will be used as a wrapper algorithm to select the best sub-set. For the remaining phases of the process, only groups belonging to the selected sub-set and provided the best result will be preserved.

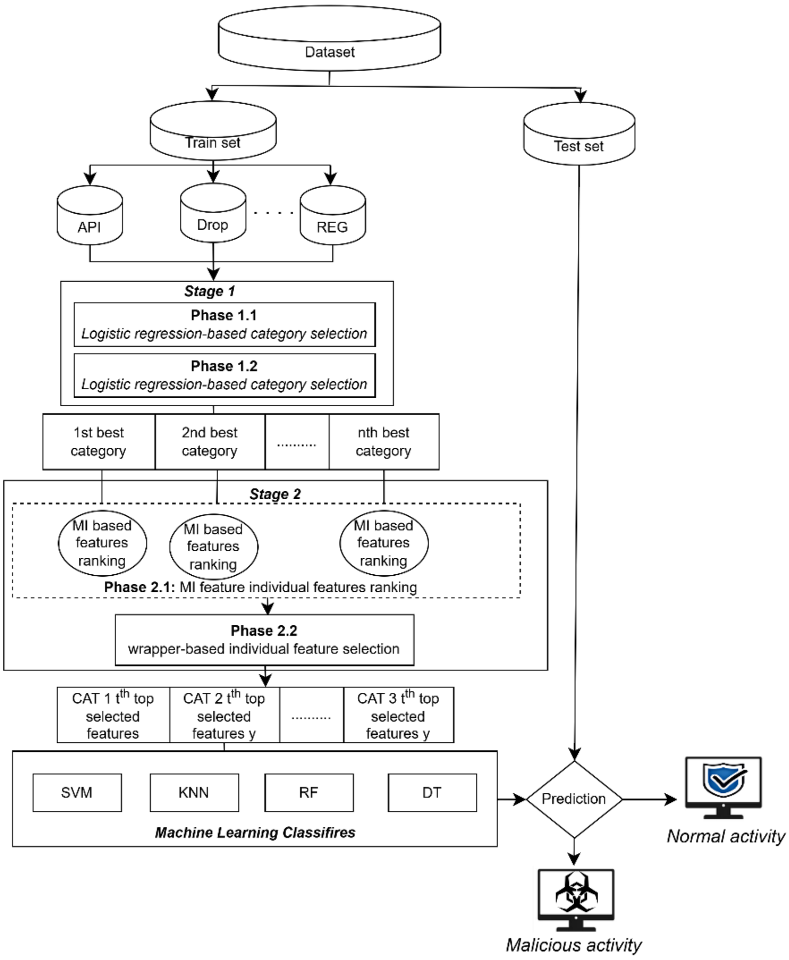

Dataset
Train set
Test set
API
Drop
. . . .
REG
Stage 1
Phase 1.1
Logistic regression-based category selection
Phase 1.2
Logistic regression-based category selection
1st best category
2nd best category
....
nth best category
Stage 2
MI based features ranking
MI based features ranking
MI based features ranking
Phase 2.1: MI feature individual features ranking
Phase 2.2
wrapper-based individual feature selection
CAT 1 tth top selected features
CAT 2 tth top selected features y
....
CAT 3 tth top selected features y
SVM
KNN
RF
DT
Machine Learning Classifires
Prediction
Normal activity
Malicious activity

Figure 1. An overview of the proposed method

3.2. Stage 2: individual feature ranking and selection

The individual feature ranking entails individually ranking features based on their relevance using the MI algorithm as a feature ranking method for each previously selected group. In our case, the classification is done separately for the four groups selected during stage 1 namely API, STR, REG, and FILES_EXT. For each group, we will sort the features in descending order based on their relevance as determined by the MI feature ranking method.

The wrapper-based individual FS phase consists of selecting an equal subset of the best-ranked features from each group of the category set after ranking and sorting the various variables in each category. This phase will be carried out in four actions: generating subset, assessing subset, defining stopping criteria, and evaluating results [31]. The first step consists of generating a subset of features to be evaluated. To this end, we adopted a forward search technique that begins with an empty set of features then adds one feature from each category at each iteration. In the second step, we defined the stopping criteria, which is 100 iterations. In the third step, we evaluated the selected subset of features using LR algorithm, which plays the role of a wrapped algorithm. The scores achieved by each subset will be compared with the best score achieved during the whole subset evaluation process until the stopping criteria (100 iterations) is reached. At the end of the FS process, we found that selecting 98 variables in each category would give the best result by applying the LR algorithm. Finally, to test the generalizability of our results, we validated the subset of selected features using different algorithms such as SVM, KNN, RF, and DT. All validation algorithms showed better results than previous similar methods in the literature. Figure 2 illustrates the phase 2 process of our FS technique, which begins with the selected feature groups and continues until the stopping criteria are reached.

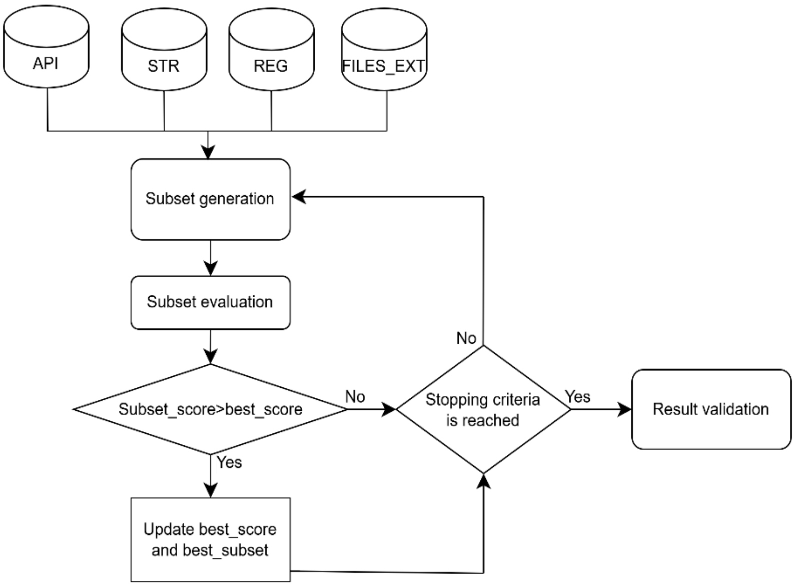


Figure 2. The wrapper-based FS process

4. RESULTS AND DISCUSSION

4.1. Experimental environments

The workstation used for all experiments has the following specifications: Intel Core i5-10500 (3.1 GHz), 8 GB of RAM. In addition, Python 3.9.7 and two primary libraries were utilized for the experiments. The first one is the 1.3.4 version of Pandas which was employed for data manipulation and analysis, while the second is the 0.24.2 version of SCIKIT-LEARN which was utilized for feature ranking and selection, classification strategies, and algorithm evaluation.

4.2. Dataset description

The dataset utilized in this article is Resilient Information Systems Security (RISS) [29], a dynamically generated dataset. It was retrieved in February 2016 and analyzed in a sandbox. The dataset contains a total of 1,524 examples, including 582 samples of ransomware from 11 categories and 942 goodware samples. This dataset also contains 30,970 features grouped into the 7 groups described in Table 1. In addition, each feature in the dataset has binary-quantitative values, therefore we did not have to perform the encoding or normalization operations.

Table 1. Description of the categories of features

Category code	Signification	Nb_features
API	API invocations	232
REG	Registry key operations	6,622
STR	Embedded strings	16,267
DROP	Extensions of the dropped files	346
DIR	File directory operations	2,424
FILE	File operations	4,141
FILES_EXT	File extensions involved in file operations	935

4.3. Hyper parameters setting

We employed many ML algorithms throughout our process, beginning with the LR algorithm as a wrapped algorithm and then moving on to the SVM, KNN, RF, and DT algorithms to generalize the findings obtained using the wrapped algorithm. In order to maintain the same conditions as the related works when performing our experiments, all algorithms cited previously were executed with default values and without any hyper parameter optimization. Additionally, to perform training and testing processes, we split the dataset using a stratified 4-fold cross validation to be sure that every malware family is represented in all folds, as recommended by [30].

4.4. Experimental results

This section displays the experimental findings for each phase of our approach, which includes ranking and selecting the most relevant category groups, ranking and selecting the most relevant features and validating the selected subset of features using several ML algorithms. Furthermore, regarding accuracy and processing time, we compared the validation results of the selected features to those produced by the entire dataset, and then we compared the experimental findings acquired by our method to those obtained by related works. As mentioned in the methodology section, the first step of our strategy was to rank groups by applying the LR algorithm on each group separately. Table 2 presents the accuracy evaluation metric obtained for each group when using LR algorithm.

After ranking the feature groups based on the scores they obtained, we used the LR algorithm to select the subset of groups that gave us the best score. The score reached by each subset of groups when measuring the accuracy evaluation metric is presented in Table 3. Among all subsets, the subset containing API, STR, REG, and FILES_EXT groups reached the best score.

Table 2. Group ranking scores

Group name	Score
API	94.42
DROP	84.00
REG	88.51
FILES	83.83
FILES_EXT	85.06
DIR	83.59
STR	90.56

Table 3. Groups subsets scores

Groups subset	Score
E1={API, STR}	97.12
E2={API, STR, REG}	97.62
E3={API, STR, REG, FILES_EXT}	97.78↑
E4={API, STR, REG, FILES_EXT, DROP}	97.70
E5={API, STR, REG, FILES_EXT, DROP, FILES}	97.70
E6={API, STR, REG, FILES_EXT, DROP, FILES, DIR}	97.45

The comparison between the results of our method and those obtained by related works [29], [30], is presented in Table 4. This table highlights the accuracy metric used to evaluate the performance of different validation algorithms including SVM, DT, RF, and KNN. Besides, we compare the number of selected features and feature reduction rate which show an outperforming result for the proposed approach.

Furthermore, the comparison between the obtained results and those reached by the entire dataset in terms of accuracy and speed is presented in Table 5. As shown in this table, the selected features by our proposed approach obtained outperforming results across all validation algorithms when using the accuracy evaluation metric. In addition, we calculate the computation time for the wrapper and validation algorithms which indicate a superiority of selected features compared to the original features.

Table 4. Selected features compared with original features

ML algorithm	LR	SVM	DT	RF	KNN	Nb selected features	Features reduction rate (%)
[29]	97.27	96.62	95.14	97.48	93.45	400	98.70
[30]	97.33	96.33	95.03	97.06	94.18	823	97.33
Our method	97.37↑	96.88↑	96.38↑	97.70↑	94.25↑	392↑	98.73↑

Table 5. The proposed method compared with related works

ML algorithm	LR		SVM		RF		DT		KNN	
	Accuracy	Time	Accuracy	Time	Accuracy	Time	Accuracy	Time	Accuracy	Time
Selected features	97.37	0.24↑	96.88↑	0.77	97.70↑	1.30↑	96.38↑	0.20↑	94.25↑	0.45↑
Original features	97.45↑	19.3	95.65	116.25	96.30	18.62	95.97	6.2	90.07	6.01

4.5. Discussion

We assumed at the beginning of this paper that selecting the most relevant feature groups and ignoring the rest would considerably affect the effectiveness and speed of detection algorithms. To validate this hypothesis, we proposed a new technique for measuring the effectiveness of individual feature groups, subsequently constructing an optimal set of these groups. As we will show in this section, the results achieved by our method showed a large superiority compared to those obtained by the entire dataset or related works.

The scores obtained for each group in phase 1.1 of our method show that the "API invocation" category received the highest score, which means that this group contains the largest number of relevant features. This result was concluded by [30], since the API invocation group contributed the most to the subset selected by their approach. According to the results obtained in phase 1.2 of our approach, the subset including the following categories: API, STR, REG, and FILES_EXT reached the highest score thus was employed in the next phase of our approach, which is consistent with what has been discovered in related works [29], [30], since the largest groups in their selected features are the same as those chosen in this step. Moreover, in the MI features ranking phase, the results collected for each category show that the API category has the most highly rated features among other groups. Therefore, it can be noted that the API group contains the features that, in accordance with other earlier research [32], may have the most influence on the decision of the ransomware detection algorithm.

Furthermore, as shown in Tables 4 and 5, our chosen set of features outperformed selected features of related works regarding accuracy and feature reduction rate. First, our approach reduced 98.73% of the overall dataset, whereas other FS methods reduced it to 97.33% [30] and 98.70% [29], impacting detection algorithm accuracy and speed. Second, our FS outperformed related works and original features in accuracy, scoring 97.37% with LR, 96.88% with SVM, 96.38 with DT, 97.70 with RF, and 94.25 with KNN.

5. CONCLUSION

According to the hypothesis proposed earlier in this article, choosing the most relevant feature groups while ignoring others would positively affect the performance and speed of the detection method. We suggested a new approach for evaluating the efficacy of distinct feature groups (categories) to verify this hypothesis, and then we built an optimal set of these groups. Regarding the feature reduction rate and accuracy, our approach outperformed related works. In addition, when we compare the outcomes of selected features with the original features in terms of detection rate and computation speed, the chosen features give better results. In conclusion, we note that all the obtained results support the hypothesis presented at the beginning of this paper. Our approach based on how the selected feature groups affect the performance of the detection algorithm will not only improve ransomware detection but also technically allow us to focus on the most important features categories and discard those less important when collecting data using behavioral analysis. Furthermore, we believe that this new FS method will also help improve other areas where datasets are high-dimensional, such as image processing, computer vision, and natural language processing. In the future, our method can be used to perform multi-class ransomware detection and evaluated on other real word datasets.

ACKNOWLEDGMENTS

There are no individuals, organizations, or funding bodies to acknowledge for this work. The research and preparation of the manuscript were conducted independently by the listed authors without external personal assistance requiring acknowledgment.

FUNDING INFORMATION

Authors state there is no funding involved.

AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Omar Chaieb	✓	✓	✓	✓	✓		✓	✓	✓	✓	✓			
Nabil Kannouf				✓	✓		✓	✓	✓	✓		✓	✓	
Mohammed Benabdellah				✓	✓		✓	✓	✓	✓		✓	✓	

C	: **C**onceptualization	I	: **I**nvestigation	Vi	: **Vi**sualization
M	: **M**ethodology	R	: **R**esources	Su	: **Su**pervision
So	: **So**ftware	D	: **D**ata Curation	P	: **P**roject administration
Va	: **Va**lidation	O	: Writing - **O**riginal Draft	Fu	: **Fu**nding acquisition
Fo	: **Fo**rmal analysis	E	: Writing - Review & **E**diting		

CONFLICT OF INTEREST STATEMENT

Authors state no conflict of interest.

INFORMED CONSENT

We have obtained informed consent from all individuals included in this study.

DATA AVAILABILITY

The data that support the findings of this study are openly available in Resilient Information Systems Security at https://doi.org/10.48550/arXiv.1609.03020, reference number [29].

REFERENCES

[1] D. Ucci, L. Aniello, and R. Baldoni, “Survey of machine learning techniques for malware analysis,” *Computers and Security*, vol. 81, pp. 123–147, 2019, doi: 10.1016/j.cose.2018.11.001.

[2] L. Daubner, M. Macak, R. Matulevičius, B. Buhnova, S. Maksović, and T. Pitner, “Addressing insider attacks via forensic-ready risk management,” *Journal of Information Security and Applications*, vol. 73, 2023, doi: 10.1016/j.jisa.2023.103433.

[3] A. Sharma, B. B. Gupta, A. K. Singh, and V. K. Saraswat, “Advanced persistent threats (APT): evolution, anatomy, attribution and countermeasures,” *Journal of Ambient Intelligence and Humanized Computing*, vol. 14, no. 7, pp. 9355–9381, 2023, doi: 10.1007/s12652-023-04603-y.

[4] S. Chaudhary and P. K. Mishra, “DDoS attacks in industrial IoT: a survey,” *Computer Networks*, vol. 236, 2023, doi: 10.1016/j.comnet.2023.110015.

[5] T. R. Reshmi, “Information security breaches due to ransomware attacks - a systematic literature review,” *International Journal of Information Management Data Insights*, vol. 1, no. 2, 2021, doi: 10.1016/j.jjimei.2021.100013.

[6] A. A. M. A. Alwashali, N. A. A. Rahman, and N. Ismail, “A survey of ransomware as a service (RaaS) and methods to mitigate the attack,” in *2021 14th International Conference on Developments in eSystems Engineering (DeSE)*, 2021, pp. 92–96, doi: 10.1109/DeSE54285.2021.9719456.

[7] P. H. Meland, Y. F. F. Bayoumy, and G. Sindre, “The ransomware-as-a-service economy within the darknet,” *Computers and Security*, vol. 92, 2020, doi: 10.1016/j.cose.2020.101762.

[8] Veeam, “2023 ransomware trends report,” *Veeam Software*. 2023. [Online]. Available: https://go.veeam.com/wp-ransomware-trends-report-2023.

[9] M. M. Ghonge, S. Pramanik, R. Mangrulkar, and D.-N. Le, *Cyber security and digital forensics: challenges and future trends*. Portsmouth, United States: Scrivener Publishing, 2022.

[10] C. Beaman, A. Barkworth, T. D. Akande, S. Hakak, and M. K. Khan, “Ransomware: recent advances, analysis, challenges and future research directions,” *Computers and Security*, vol. 111, 2021, doi: 10.1016/j.cose.2021.102490.

[11] F. A. Vadhil, M. L. Salihi, and M. F. Nanne, “Machine learning-based intrusion detection system for detecting web attacks,” *IAES International Journal of Artificial Intelligence*, vol. 13, no. 1, p. 711, Mar. 2024, doi: 10.11591/ijai.v13.i1.pp711-721.

[12] P. Bajpai and R. Enbody, “An empirical study of API calls in ransomware,” in *2020 IEEE International Conference on Electro Information Technology (EIT)*, 2020, pp. 443–448, doi: 10.1109/EIT48999.2020.9208284.

[13] S. Kumar, B. Janet, and S. Neelakantan, “Identification of malware families using stacking of textural features and machine learning,” *Expert Systems with Applications*, vol. 208, 2022, doi: 10.1016/j.eswa.2022.118073.

[14] M. Gopinath and S. C. Sethuraman, “A comprehensive survey on deep learning based malware detection techniques,” *Computer Science Review*, vol. 47, 2023, doi: 10.1016/j.cosrev.2022.100529.

[15] N. Kannouf, M. Labbi, Y. Chahid, M. Benabdellah, and A. Azizi, “A key establishment attempt based on genetic algorithms applied to rfid technologies,” *International Journal of Information Security and Privacy*, vol. 15, no. 3, 2021, doi: 10.4018/IJISP.2021070103.

[16] O. Chaieb, N. Kannouf, R. Amjoun, and M. Benabdellah, “Machine learning-based intrusion detection system: review and taxonomy,” in *Proceedings of the 6th International Conference on Big Data and Internet of Things*, 2023, pp. 10–21, doi: 10.1007/978-3-031-28387-1_2.

[17] M. Berhili, O. Chaieb, and M. Benabdellah, “Intrusion detection systems in IoT based on machine learning: a state of the art,” *Procedia Computer Science*, vol. 251, pp. 99–107, 2024, doi: 10.1016/j.procs.2024.11.089.

[18] S. Bagui and H. Brock, “Machine learning for android scareware detection,” *Journal of Information Technology Research*, vol. 15, no. 1, pp. 1–15, 2022, doi: 10.4018/jitr.298326.

[19] M. Humayun, N. Z. Jhanjhi, A. Alsayat, and V. Ponnusamy, “Internet of things and ransomware: evolution, mitigation and prevention,” *Egyptian Informatics Journal*, vol. 22, no. 1, pp. 105–117, 2021, doi: 10.1016/j.eij.2020.05.003.

[20] K. Begovic, A. Al-Ali, and Q. Malluhi, “Cryptographic ransomware encryption detection: survey,” *Computers and Security*, vol. 132, 2023, doi: 10.1016/j.cose.2023.103349.

[21] N. Rani, S. V. Dhavale, A. Singh, and A. Mehra, “A survey on machine learning-based ransomware detection,” in *Proceedings of the Seventh International Conference on Mathematics and Computing*, 2022, pp. 171–186, doi: 10.1007/978-981-16-6890-6_13.

[22] S. Maniath, P. Poornachandran, and V. G. Sujadevi, “Survey on prevention, mitigation and containment of ransomware attacks,” *Communications in Computer and Information Science*, vol. 969, pp. 39–52, 2019, doi: 10.1007/978-981-13-5826-5_3.

[23] T. McIntosh, A. S. M. Kayes, Y. P. P. Chen, A. Ng, and P. Watters, “Dynamic user-centric access control for detection of ransomware attacks,” *Computers and Security*, vol. 111, 2021, doi: 10.1016/j.cose.2021.102461.

[24] T. McIntosh, A. S. M. Kayes, Y. P. P. Chen, A. Ng, and P. Watters, “Applying staged event-driven access control to combat ransomware,” *Computers and Security*, vol. 128, 2023, doi: 10.1016/j.cose.2023.103160.

[25] X. Luo and Q. Liao, “Awareness education as the key to ransomware prevention,” *Information Systems Security*, vol. 16, no. 4, pp. 195–202, 2007, doi: 10.1080/10658980701576412.

[26] A. Damodaran, F. Di Troia, C. A. Visaggio, T. H. Austin, and M. Stamp, “A comparison of static, dynamic, and hybrid analysis for malware detection,” *Journal of Computer Virology and Hacking Techniques*, vol. 13, no. 1, pp. 1–12, 2017, doi: 10.1007/s11416-015-0261-z.

[27] D. Manikandan and J. Dhilipan, “Machine learning approach for intrusion detection system using dimensionality reduction,” *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 34, no. 1, pp. 430–440, 2024, doi: 10.11591/ijeecs.v34.i1.pp430-440.

[28] M. Mohammadi *et al.*, “A comprehensive survey and taxonomy of the SVM-based intrusion detection systems,” *Journal of Network and Computer Applications*, vol. 178, 2021, doi: 10.1016/j.jnca.2021.102983.

[29] D. Sgandurra, L. Muñoz-González, R. Mohsen, and E. C. Lupu, “Automated dynamic analysis of ransomware: benefits, limitations and use for detection,” *arXiv-Computer Science*, pp. 1–12, 2016.

[30] M. S. Abbasi, H. Al-Sahaf, M. Mansoori, and I. Welch, “Behavior-based ransomware classification: a particle swarm optimization wrapper-based approach for feature selection,” *Applied Soft Computing*, vol. 121, 2022, doi: 10.1016/j.asoc.2022.108744.

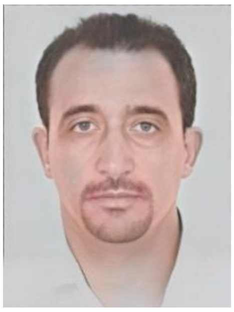
[31] A. Thakkar and R. Lohiya, “A survey on intrusion detection system: feature selection, model, performance measures, application perspective, challenges, and future research directions,” *Artificial Intelligence Review*, vol. 55, no. 1, pp. 453–563, 2022, doi: 10.1007/s10462-021-10037-9.

[32] A. Pektaş and T. Acarman, “Malware classification based on API calls and behaviour analysis,” *IET Information Security*, vol. 12, no. 2, pp. 107–117, 2018, doi: 10.1049/iet-ifs.2017.0430.

BIOGRAPHIES OF AUTHORS

Omar Chaieb is a Ph.D. student in the Laboratory of Arithmetic, Scientific Computing, and their Applications (LACSA) at the Faculty of Science, Mohamed First University, OUJDA. He received his master’s in 2013 in code, cryptography, and security of information from the Faculty of Sciences, University Mohamed V in Rabat, Morocco. His research interests include intrusion detection systems, artificial intelligence, and malware analysis. He can be contacted at email: omar.chaieb.men@gmail.com.

Nabil Kannouf is a professor of computer science at Abdelmalek Essaadi University in Morocco, a position he has held since 2020. He holds a Ph.D. in computer science and earned his master’s degree in computer engineering in 2010 from Faculty of Sciences, Mohamed Premier University’s, Oujda, Morocco. He can be contacted at email: nabil.kannouf@gmail.com.



Mohammed Benabdellah is a professor of computer science at Mohammed First University in Oujda, Morocco. He earned his Ph.D. in engineering sciences from Mohammed V University in Rabat, Morocco, in June 2007. Renowned for his internationally recognized research in cybersecurity, artificial intelligence, and image processing, he actively contributes to advancing these fields. He can be contacted at email: med_benabdellah@yahoo.fr.