

AN EVALUATION OF MACHINE LEARNING CLASSIFICATION ALGORITHMS FOR RANSOMWARE VARIANT CLASSIFICATION USING API CALL SEQUENCE CORRELATIONS

¹Gilbert George, ²Babatunde Usman Jimoh

¹Department of Cybersecurity, Nile University of Nigeria, Jabi, Abuja, Nigeria

²Department of Computer Science, Baze University, Jabi, Abuja, Nigeria

*Corresponding Author Email Address: batatunde.jimoh@bazeuniversity.edu.ng

ABSTRACT

Although considerable research has focused on ransomware variant classification, extracting robust features from malware samples remains challenging, which limits the accuracy of current methods. This study introduces a novel classification approach that uses the Pearson correlation coefficient between Application Programming Interface (API) group frequencies as input to Random Forest and Support Vector Machine (SVM) classifiers. The rationale for this approach is that distinct ransomware families exhibit unique behavioral patterns during dynamic execution that cannot be adequately captured by static analysis or simple feature counts alone. Empirical evaluations demonstrate that the Random Forest classifier achieved an accuracy of 99.9%, slightly outperforming the SVM classifier, which achieved an accuracy of 98.6%. These results indicate that the proposed feature extraction method allows for highly accurate ransomware family classification.

Keywords: Cybercrime, Digital Forensics, Machine Learning, Network Monitoring, Network Forensics, Ransomware.

INTRODUCTION

The volume of malware has grown exponentially year over year. Historically prominent strains, such as the Locky ransomware, highlighted the severe threat these vectors pose to global infrastructure. By accurately classifying ransomware variants, security analysts can develop targeted countermeasures for specific ransomware families, significantly improving incident response times and reducing analytical workloads. Unfortunately, due to the widespread use of code obfuscation techniques by ransomware developers, feature extraction via static analysis remains highly challenging (Singh & Singh, 2022). Consequently, dynamic analysis has emerged as a preferred alternative. In this paper, we introduce a behavioral identification solution based on the similarity of API call sequences during runtime. API calls are among the most effective metrics for representing malware actions in dynamic analysis environments (Aboaoja et al., 2022). Because ransomware relies heavily on specific operating system interactions to execute its payload, statistical analysis of Windows API calling sequences can reliably predict malicious behavior (Aboaoja et al., 2022). A higher correlation coefficient between system API calls typically signifies strong behavioral similarity between two samples, suggesting they belong to the same malware lineage (Demirkiran et al., 2022). Recent developments

The primary aim of this paper is to establish distinct ransomware behavioral patterns. To achieve this, we leverage the correlation coefficient between categorized API groups as our primary feature set. The specific objectives of this study are:

in dynamic ransomware and malware analysis have shifted the focus from simple API call frequencies to mapping the sequential, contextual, and structural relationships among these calls at runtime. For instance, Alhazmi (2025) demonstrated that extracting deep behavioral features from execution logs using sophisticated language models such as BERT—combined with conventional classifiers such as Random Forest and SVM—yields robust detection rates against evolving, polymorphic threats. Similarly, to overcome the limitations of linear numerical vectors, which often lose behavioral context, Ma (2025) proposed modeling API call sequences as directed graphs. Utilizing Graph Convolutional Networks (GCN) for structural processing and the Synthetic Minority Over-sampling Technique (SMOTE) for data balancing, this approach preserves sequential interactions while mitigating dataset imbalances in multi-class malware identification. These recent methodologies highlight the importance of considering contextual and relational dependencies of system calls, providing a benchmark against which straightforward statistical measures, such as the Pearson correlation coefficient, can be evaluated.

Despite these recent advancements, a significant research gap persists in the literature. Existing ransomware classification frameworks typically fall into two extremes: they either rely on simplistic API call frequency counts—which strip away the structural, sequential, and relational context of operating system interactions—or they deploy highly sophisticated deep learning architectures, such as Graph Convolutional Networks (GCNs) or Transformer-based language models. While these deep learning models achieve high accuracy, they introduce substantial computational overhead, require extensive preprocessing, entail complex hyperparameter tuning and lack interpretability.

Currently, there is a critical shortage of lightweight, intermediate methodologies capable of capturing behavioral interactions and cross-functional dependencies across different API domains (e.g., the inevitable relationship between file operations and cryptographic routines during a ransomware strike) without incurring prohibitive computational costs. Consequently, there is an urgent need for an efficient, resource-friendly statistical framework that preserves behavioral context for rapid forensic triage on resource-constrained endpoints.

- i. To evaluate multi-class malware classification based on dynamic API call sequences.
- ii. To quantify the descriptive contribution of specific API categories to ransomware family classification.

iii. To identify the optimal machine learning algorithm for ransomware classification using API correlation matrices.

MATERIALS AND METHODOLOGIES

Building upon prior work that utilizes cosine similarity between Dynamic Link Libraries (DLLs) (Singh et al., 2022), we propose that ransomware families can be effectively categorized by analyzing relationships among distinct operational API groups. API Groups Correlation Coefficient

Our methodology constructs a statistical profile by calculating the Pearson correlation coefficient between designated API groups (Li et al., 2022). To compute these coefficients, we monitored the calling frequencies of critical APIs across 11 prominent ransomware families: Critroni, CryptLocker, CryptoWall, KOLLAH, Kovter, Locker, MATSNU, PGPCODER, Reveton, TeslaCrypt, and Trojan-Ransom, as well as a control group of benign applications.

Because dynamic analysis relies on system calls interacting directly with the operating system (Demirkiran et al., 2022), we categorized the extracted APIs into five fundamental behavioral groups:

FileAPI: File system operations (e.g., creation, deletion, enumeration).

CryptAPI: Cryptographic operations and key management.

RegistryAPI: Registry key modifications and persistence mechanisms.

SocketAPI: Network connections and command-and-control (C2) communication.

ProcessAPI: Process injection, thread creation, and execution.

The Pearson correlation coefficient r between any two API groups is calculated using the formula:

$$r = \frac{\sum(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum(x_i - \bar{x})^2 \sum(y_i - \bar{y})^2}}$$

Where

x_i, y_i = Individual API group frequency values in the samples. $\bar{x}, \bar{y}$

The calculated mean values of the x and y variables
 n = Total number of features/samples

The specifics are listed in Table 1. After extracting the calling frequencies of the APIs in Figure 1, determine the correlation coefficients between API group columns.

JSON report, extracts the target indicators from the apistats block, and increments the respective group counters.

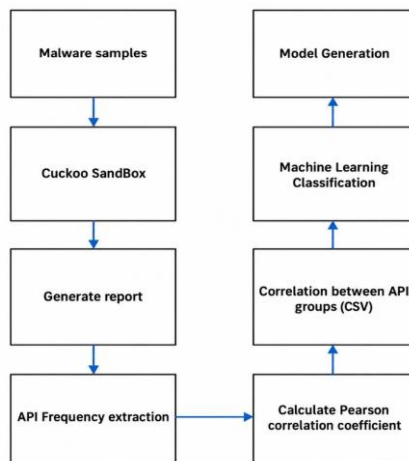


Figure 1 depicts the proposed method in our study.

Method Outline

The operational pipeline of our proposed method is structured as follows (illustrated conceptually in Figure 1)

Execute the ransomware sample within an isolated environment to generate a dynamic execution report. Parse the generated report to extract the call frequencies of the target APIs listed in Table 1.

Compute the Pearson correlation coefficients across the five groups to form a feature matrix.

Train and evaluate the machine learning classifiers using the resulting matrix. We utilized Cuckoo Sandbox, an automated malware analysis platform, to execute the samples and generate comprehensive JavaScript Object Notation (JSON) dynamic analysis reports. The parsing algorithm initializes the call frequencies of the five API groups to zero, traverses the

Feature Conversion and Labeling

The extracted frequency arrays are converted into matrices to calculate the Pearson correlation coefficients. The resulting values are compiled into a master feature vector (corrcoef list), labeled with the corresponding ransomware family name, and saved to a comma-separated values (CSV) format. This dataset serves as the direct input for our machine learning classifiers.

Experiment Setup

The experiments were conducted on a Windows 10 x64 environment powered by a 2.5 GHz processor and 8 GB of RAM. The machine learning workflows were executed using Weka. To ensure rigorous validation, the dataset was evaluated using a 10-fold cross-validation scheme alongside a traditional 70:30 train-test split.

Dataset

The experimental dataset comprised 2,000 instances gathered from public repositories

<https://www.kaggle.com/datasets/nkongolo/ugransome-dataset>. It included diverse Portable Executable (PE) files spanning 11 distinct ransomware families, as well as verified benign software.

RESULTS

Both machine learning models demonstrated strong performance metrics. The Random Forest classifier slightly outperformed the Support Vector Machine (SVM), achieving an overall classification accuracy of 99.90%. This underscores the viability of utilizing statistical API correlations for highly efficient ransomware identification.

Table 1. Shows the API features that were extracted from the dynamic analysis.

FileAPI	CryptAPI	RegistryAPI	SocketAPI	ProcessAPI
			Socket	
			InternetOpen	
			Shutdown	
			SendTo	
			Connect	
		RegCloseKey	Bind	CreateThreadCreateR
		RegCreateKey	Listen	oteThreadNtResumeTl
		RegQueryValueEx	Accept	adNtGetContextThrea
FindNext		RegSetValueEx	Recv	SetContextThreadCrea
FileGetFileType.		RegEnumKeyEx	Send	ProcessInternalWNtOp
FindFirstFile	CryptDeriveKey	RegOpenKeyExW	InternetOpenUrl	ProcessProcess32Nex
ExSetFilePointer	CryptEncrypt	NtQueryValueKey	InternetReadFil	Process32FirstWNtTer
GetFileSize	CryptDecrypt	NtOpenKey	InternetWriteF	nateProcess

Table 2. Performance Comparison of Classifier Models

Classifier Mode	Accuracy (%)	Precisio (%)	Recall (%)	F1-Score (%)	Weighted R AUC	Total Instances
Random Forest	99.90%	99.90%	99.90%	99.90%	0.999	2000
Support Vec Machine (SVM)	98.60%	98.50%	98.60%	98.55%	0.994	2000

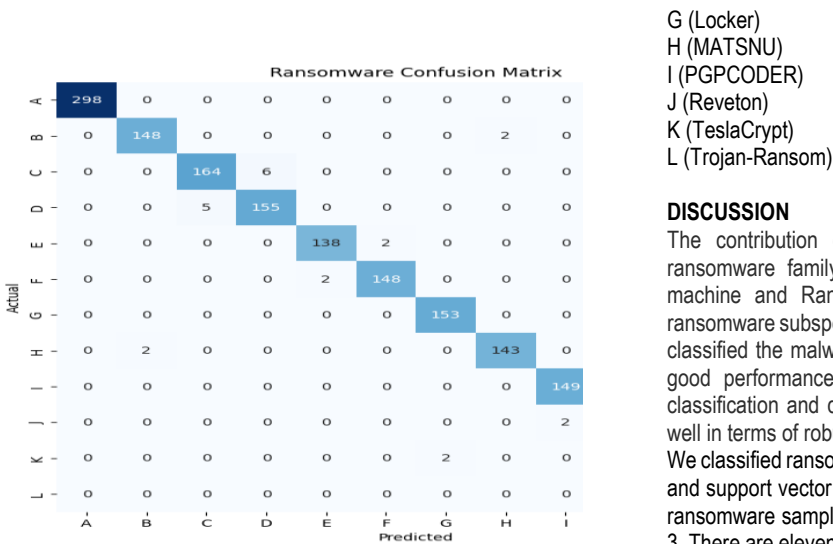


Figure 2: Confusion matrix
key
A (Benign)
B (Critroni)
C (CryptLocker)
D (CryptoWall)
E (KOLLAH)
F (Kovter)
G (Locker)
H (MATSNU)
I (PGPCODER)
J (Reveton)
K (TeslaCrypt)
L (Trojan-Ransom)

DISCUSSION

The contribution of each API to the classification of each ransomware family was ascertained using the support vector machine and Random Forest classifiers after classifying the ransomware subspecies. The Random Forest classifiers effectively classified the malware with an accuracy of up to 99% (indicating good performance in handling the use of APIs for malware classification and detection). The proposed classifiers performed well in terms of robustness, showing stable performance. We classified ransomware families using API frequency as features and support vector machines and Random Forest classifiers. The ransomware samples used in this experiment are shown in Table 3. There are eleven families in total, excluding the standard benign software; therefore, we set the cross-validation to 10. The test data is 30% of the total size of the ransomware samples. Using the API frequencies as feature quantities, the classification accuracy is 99%. The experimental finding suggests that APS frequency can be used as a feature.

Conclusions

In this paper, we proposed a method for efficiently categorizing ransomware using SVM and Random Forest classifiers by computing Pearson correlation coefficients, using them as features and extracting the calling frequency of an API from a report produced by dynamic analysis of the malware. We also discovered that it could identify the API categories that influence how each ransomware family is classified. It is anticipated that this finding would reduce the time other ransomware researchers must spend performing analysis. We plan to investigate more features and samples in the future and use them to enhance our suggested method.

REFERENCES

- Aboaoja, F. A., Zainal, A., Ghaleb, F. A., Al-rimy, B. A. S., Eisa, T. A. E., & Elnour, A. A. H. (2022). Malware detection issues, challenges, and future directions: A survey. *Applied Sciences*, 12(17), 8482.
<https://doi.org/10.3390/app12178482>
- Alhazmi, A. H. (2025). A robust and dynamic malware detection and classification model using behavioral-based analysis and BERT technique. *PLoS ONE*, 20(9), e0327604.
<https://doi.org/10.1371/journal.pone.0327604>
- Cuckoo Sandbox. (n.d.). *Cuckoo Sandbox—Automated malware analysis*. Retrieved November 9, 2022, from
<https://cuckoosandbox.org/download>
- Demirkiran, F., Çayır, A., Ünal, U., & Dağ, H. (2022). An ensemble of pre-trained transformer models for imbalanced multiclass malware classification. *Computers & Security*, 121, 102846.
<https://doi.org/10.1016/j.cose.2022.102846>
- Li, C., Cheng, Z., Zhu, H., Wang, L., Lv, Q., Wang, Y., Li, N., & Sun, D. (2022). DMalNet: Dynamic malware analysis based on API feature engineering and graph learning. *Computers & Security*, 122, 102872.
<https://doi.org/10.1016/j.cose.2022.102872>
- Ma, C. (2025). A malware classification method based on directed API call relationships. *PLoS ONE*, 20(3), e0299706.
<https://doi.org/10.1371/journal.pone.0299706>
- Singh, A., Ikuesan, R. A., & Venter, H. (2022). *Ransomware detection using process memory* (arXiv:2203.16871). arXiv.
<https://doi.org/10.48550/arXiv.2203.16871>
- Singh, J., & Singh, J. (2022). Assessment of supervised machine learning algorithms using dynamic API calls for malware detection. *International Journal of Computers and Applications*, 44(3), 270–277.
<https://doi.org/10.1080/1206212X.2020.1732641>