SEMARAK ILMU PUBLISHING
202103268166(003316878-P)

Journal of Advanced Research in Applied Sciences and Engineering Technology

Journal homepage:
https://semarakilmu.com.my/journals/index.php/applied_sciences_eng_tech/index
ISSN: 2462-1943

SEMARAK ILMU PUBLISHING
Journal of
ADVANCED RESEARCH IN APPLIED SCIENCES AND ENGINEERING TECHNOLOGY
ISSN 2462 - 1943

Ransomware Early Detection using Machine Learning Approach and Pre-Encryption Boundary Identification

Wira Z. A. Zakaria[1,*], Mohd Faizal Abdollah[2], Othman Abdollah[2], S.M. Warusia Mohamed S.M.M[2]

1 MyCERT, Cybersecurity Malaysia, Menara Cyber Axis, Jalan Impact, 63000 Cyberjaya, Selangor, Malaysia
2 Fakulti Teknologi Maklumat dan Komunikasi, Universiti Teknikal Malaysia Melaka, Hang Tuah Jaya, 76100 Durian Tunggal, Melaka, Malaysia

ARTICLE INFO

Article history:
Received 22 June 2023
Received in revised form 1 December 2023
Accepted 28 March 2024
Available online 25 June 2024

Keywords:

Ransomware; Early detection; Pre-encryption; Pre-encryption boundary; Crypto-ransomware; Cryptographic ransomware

ABSTRACT

The escalating ransomware threat has catalysed the formation of a sophisticated network of cybercriminal enterprises. Addressing this issue, our research provides a detailed exploration of the ransomware menace and an evaluation of contemporary detection methodologies. A successful ransomware attack leverages many factors: robust encryption methods that defy decryption, the anonymity of cyber currencies, and the widespread availability of ransomware kits that enable even inexperienced actors to launch attacks. Such dynamics have cultivated a niche for cybercriminal specialists in the digital underworld. In response to these challenges, our study proposes a detection framework based on machine learning, a domain where regression algorithms have gained popularity without yielding a definitive protective model. We employ API call analysis as the foundation to assess various machine learning classifiers' efficiency in identifying ransomware. The evaluation demonstrates that the Naive Bayes classifier underperforms due to suboptimal accuracy, making it unsuitable for this application. Conversely, Logistic Regression, with an AUC of 0.951, minimal training time, and substantial efficacy gains, emerges as a strong contender. The Decision Tree and Random Forest classifiers exhibit comparable proficiency; however, the Decision Tree's interpretability and Random Forest's computational swiftness present unique advantages. Superior still, SVM and Gradient Boosted Trees command the highest AUC and gains, albeit at the cost of increased training duration. Our findings affirm the pivotal role of API call analysis in ransomware detection and the potency of machine learning approaches in learning from extensive datasets to identify novel malware strains. Given the continual evolution of malware, detection methodologies must adapt correspondingly. This study's comparative analysis elucidates the trade-offs between accuracy, computational speed, and training time, guiding the selection of the optimal machine learning algorithm for robust ransomware detection.

1. Introduction

Ransomware attacks can have severe consequences, including data loss, monetary losses, operational disruptions, reputational harm, and even threats to human safety [1-3]. A ransomware

[*] *Corresponding author.*
E-mail address: wirazanoramy@gmail.com

https://doi.org/10.37934/araset.47.2.121137

attack can result in the loss of personal data, such as photos, documents, and financial information, for individuals. A ransomware attack can disrupt business operations, result in financial losses, and harm an organization's reputation. In certain instances, a ransomware attack can pose a threat to vital infrastructure, such as hospitals or utilities. Ransomware attacks pose a threat to society because they can disrupt the operation of essential services, such as healthcare and transportation systems [4-6]. Moreover, ransomware attacks are frequently conducted by criminal organizations or state-sponsored actors, posing threats to national security [7-11]. The significant impact of ransomware attacks highlights the significance of early detection, prevention, and mitigation strategies for protection against these threats. In recent years, ransomware has become increasingly prevalent, targeting individuals, organizations, and even critical infrastructure. This essay will provide a comprehensive explanation of ransomware, its operation, and its effects on victims. We will also discuss strategies for preventing ransomware attacks and mitigating their damage.

Ransomware is a type of malicious software that encrypts a victim's files or prevents access to their computer system, rendering it inoperable [12-14]. The attacker then demands a ransom payment, typically in the form of a cryptocurrency such as Bitcoin, in exchange for the decryption key. A system can be infected with ransomware through various channels, including email attachments, malicious websites, and software vulnerabilities. Once ransomware infects a victim's computer, the malware begins encrypting files and displaying a message, typically demanding payment within a specified time frame. The ransom amount can range from a few hundred to millions of dollars, depending on the perceived value of the victim and the attacker's methods. Attacks by ransomware can have devastating effects on individuals and organizations. A ransomware attack can result in the loss of personal data, such as photos, documents, and financial information, for individuals. A ransomware attack can disrupt business operations, result in financial losses, and harm an organization's reputation. In certain instances, a ransomware attack can pose a threat to vital infrastructure, such as hospitals or utilities.

A ransomware attack can have far-reaching and long-lasting consequences. There is no assurance that the attacker will provide the decryption key or restore the victim's data, even if the victim pays the ransom. In addition, paying the ransom encourages additional attacks and supports criminal activity. A multilayered approach to cybersecurity is essential for preventing and mitigating the damage caused by ransomware attacks [15-17]. This strategy includes:

i. Implementing robust cybersecurity policies: Organizations should have in place a comprehensive cybersecurity policy that includes routine backups of critical data, software updates, and employee training on best practices for email and web browsing.
ii. Antivirus and anti-malware software can detect and prevent ransomware attacks before they infect a computer system.
iii. Utilizing firewalls and intrusion prevention systems: Firewalls and intrusion prevention systems can detect and prevent malicious activity by blocking unauthorized network access.
iv. Regular backups of critical data can ensure that a victim can restore their data even if it is encrypted by ransomware.
v. Developing a response plan for ransomware attacks: Organizations should have a clear response plan that includes isolating infected systems, notifying customers and stakeholders of the attack, and contacting law enforcement.

In addition to these preventive measures, organizations should be aware of the most recent ransomware techniques and be prepared to adapt their cybersecurity strategies accordingly. Some ransomware attackers, for instance, are now employing machine learning and artificial intelligence to evade detection by conventional antivirus and anti-malware software. Ransomware poses an increasing threat to individuals, businesses, and society. The severity and duration of a ransomware attack highlights the importance of early detection, prevention, and mitigation strategies [17-19]. By adopting a multilayered approach to cybersecurity and keeping abreast of the most recent ransomware techniques, organizations can better defend themselves against this evolving threat.

2. Chronology of a Ransomware Attack

Below are the sequences involving in a ransomware attack [20-23]:

i. Initial compromise:
    - Phishing email: The perpetrator sends a phishing email containing a malicious attachment or link to the target organization.
    - Exploitation of vulnerabilities: The perpetrator gains access to the target system by exploiting unpatched software or network vulnerabilities.

ii. Privilege escalation: The perpetrator exploits vulnerabilities or uses stolen credentials to obtain elevated privileges on the compromised system.

iii. Reconnaissance: The perpetrator collects data about the target network to identify valuable data and systems.

iv. Infection spread: The perpetrator traverses the network laterally, infecting other systems and devices.

v. Downloading and execution: The adversary downloads and executes the ransomware payload on compromised systems.

vi. Encryption: The ransomware employs strong encryption algorithms to encrypt the victim's data.

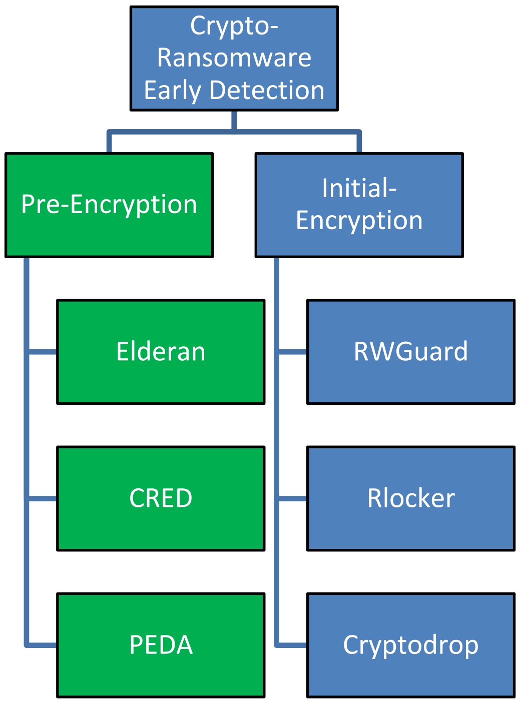
vii. Ransom notes: The perpetrator leaves a ransom note with payment instructions and a deadline on infected systems.

viii. Communication with the assailant: The victim contacts the assailant to negotiate the ransom payment or request additional information.

ix. Payment: The victim pays the ransom, typically in cryptocurrencies such as Bitcoin, to receive the decryption key. The perpetrator provides the decryption key, and the victim attempts to decrypt and recover their data.

3. Related Works on Early Detection

EldeRan is a proposed framework for identifying significant ransomware dynamic features and using them to detect ransomware. The Mutual Information criterion was employed to select the most relevant dynamic features from a large set. By utilizing a small set of features without compromising the performance of the machine learning classifier, EldeRan is well-suited for detecting new ransomware families. In terms of accuracy, Regularized Logistic Regression was compared to other machine learning classifiers, such as Support Vector Machine (SVM) and Naive Bayes. The results showed that Logistic Regression outperformed Naive Bayes and was competitive with SVM. Furthermore, Logistic Regression was found to be easier to train and adapt compared to SVM. The regularization technique applied to Logistic Regression helped the classifier generalize better to unseen samples by preventing overfitting.



This research proposes an interesting and multi-layered approach to detecting ransomware attacks, focusing specifically on crypto-ransomware, which locks files using encryption. The use of the Pre-Encryption Detection Algorithm (PEDA) aims to catch ransomware activity at its most early, pre-encryption stage, which is crucial to prevent irreversible damage to files. The two-stage approach to detection adds depth and robustness to the process [24,25].

The first level involves signature comparison using the SHA-256 Secure Hashing Algorithm. This technique, also known as signature-based detection, is a widely used method for identifying known threats in cybersecurity. Its speed and accuracy make it a reliable first line of defence, but it has a key limitation: it struggles to detect new, unknown threats (zero-days). The second level of detection involves a Learning Algorithm (LA) based on pre-encryption application program interface (API) calls. Machine learning algorithms have proven highly effective at detecting ransomware based on API calls because they can learn from the behaviours of known ransomware and apply this learning to identify suspicious activity, even from previously unseen ransomware variants. The high recall rates achieved by this approach are impressive. Recall is a key performance metric in this context because it measures the proportion of actual positive cases (in this case, ransomware attacks) that were correctly identified. A high recall rate means that very few actual ransomware attacks went undetected. The research has identified specific API calls that differentiate between ransomware (malicious software) and goodware (benign software). The fact that they have found API calls that are common in ransomware and less common in goodware, and vice versa, is particularly valuable. This type of information can be used to train more effective machine learning models and to create more robust signature-based detection methods. This approach presents a multi-layered detection system that can potentially be very effective at detecting crypto-ransomware at its earliest stages. The integration of both signature-based detection and machine learning enhances the strength of the detection system and makes it more adaptable to evolving threats. However, the effectiveness of this method in a real-world scenario would need to be tested and validated.

4. Pre-Encryption Boundary

In ransomware, the "pre-encryption boundary" is the point in an attack right before the bad software starts to encrypt the victim's files [26]. Ransomware attacks usually happen in stages, starting with infiltration, moving laterally, establishing persistence, and ending with data encryption (and possibly data theft). The pre-encryption boundary would be when all the prep work is done, and the ransomware is about to start encrypting the files. Finding the pre-encryption boundary is so important because if you step in at this point, you can stop a ransomware attack from having disastrous effects. If you can find ransomware before it encrypts your data and stop it, you might be able to save valuable data and avoid a lot of money loss and trouble. From a technical point of view, finding the pre-encryption boundary might involve looking for certain indicators of compromise (IOCs). This could include strange network traffic, the presence of known ransomware-related files or processes, unauthorized attempts to get higher privileges, or sudden, unexplained changes in file permissions. Finding ransomware before it starts encrypting data is hard, but there are a few ways to do it:

i. Anomaly Detection: Advanced cybersecurity systems often use machine learning algorithms to learn normal system behaviour and identify any unusual activity. If a piece of software starts trying to access and modify many files in a short time, this could be flagged as potential ransomware.
ii. Signature-Based Detection: This approach involves maintaining an up-to-date database of known ransomware signatures—specific characteristics of the ransomware code—and scanning for these regularly.
iii. Behaviour-Based Detection: Unlike signature-based detection, which relies on prior knowledge of specific ransomware families, behaviour-based detection focuses on identifying actions that are typical of ransomware. For instance, an attempt to contact a known ransomware command-and-control server would be flagged.
iv. Sandboxing: This is a method where suspicious programs are executed in a safe, isolated environment (a "sandbox") to observe their behaviour. If they behave like ransomware, they can be neutralized before reaching the encryption stage.
v. Threat Hunting: A proactive approach where cybersecurity professionals actively look for indicators of compromise (IOCs) or other signs of intrusion in their systems.

5. Methodology

This section discusses the research methodology for the early detection framework for crypto ransomware. The detection framework includes data acquisition, data preparation, feature selection, and classification modules. Multiple experiments were conducted on the three modules comprising this framework to quickly determine the most effective technique for detecting ransomware attacks in advance. The evaluations of performance utilized in this study are true positive (TP) and accuracy (A). A methodology is a strategy or action plan that produces results. In this chapter, each of the methodology's procedures for implementing the detection framework are detailed. In this research methodology model, there are four phases. Figure 1 depicts the phases as Data Collection, Framework Development, Experimentation, and Evaluation.

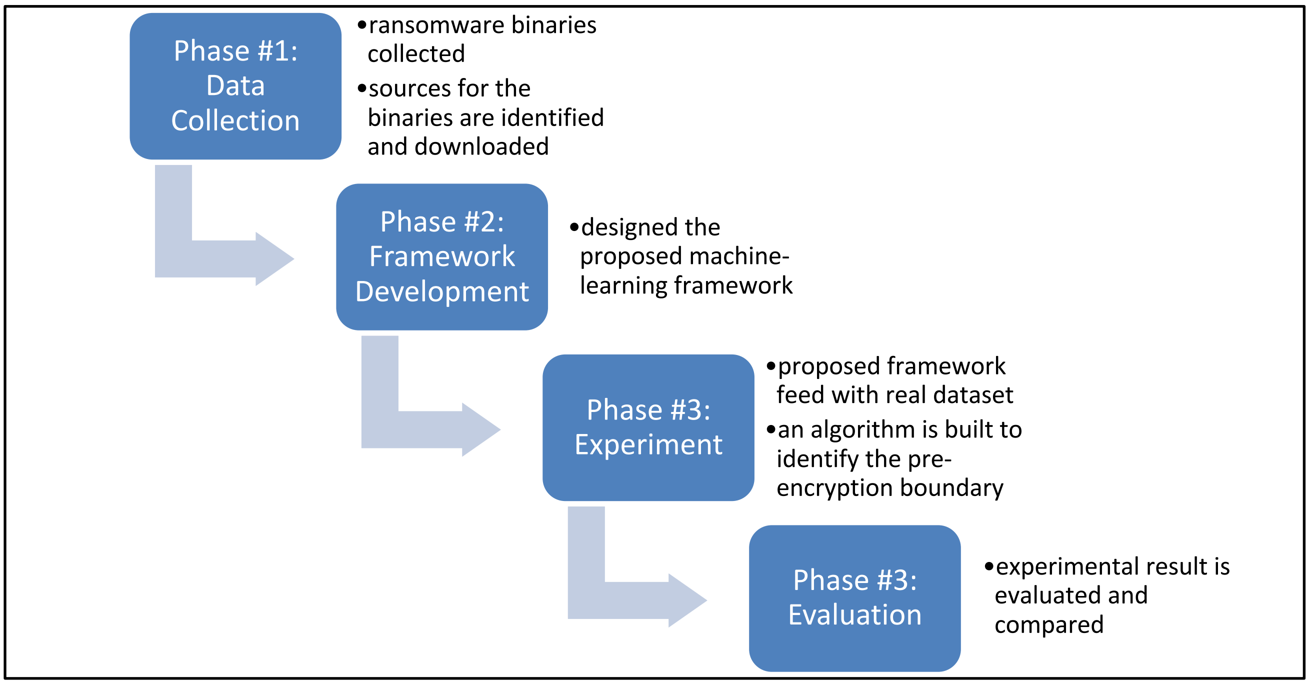

Phase #1: Data Collection
•ransomware binaries collected
•sources for the binaries are identified and downloaded
Phase #2: Framework Development
•designed the proposed machine-learning framework
Phase #3: Experiment
•proposed framework feed with real dataset
•an algorithm is built to identify the pre-encryption boundary
Phase #3: Evaluation
•experimental result is evaluated and compared

Fig. 1. Research design

5.1 Phase #1: Ransomware Data Collection

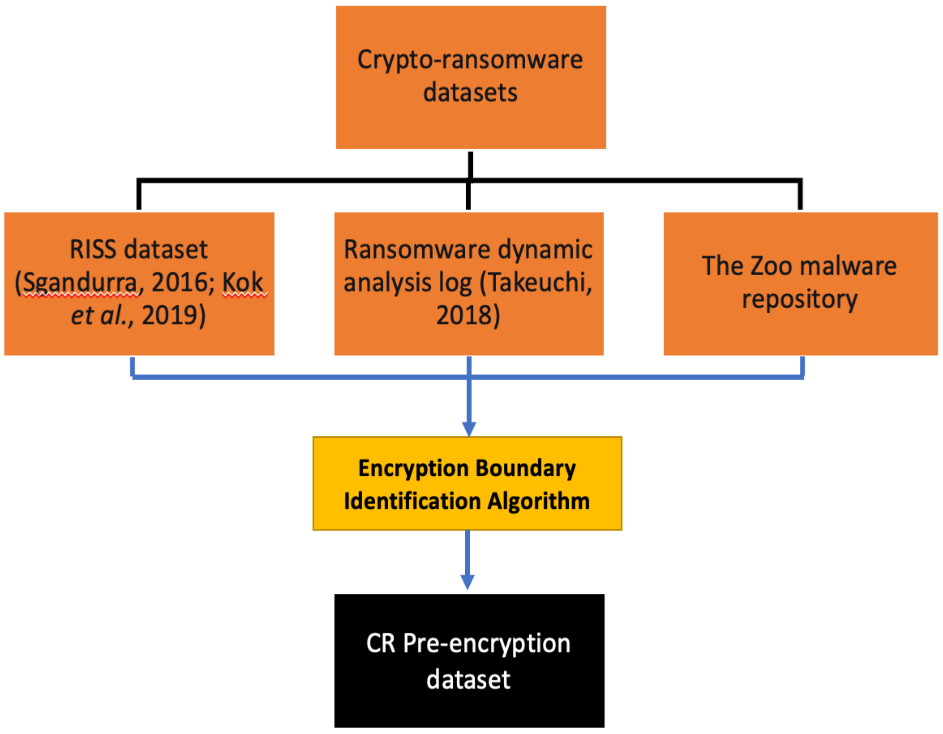

Crypto-ransomware datasets
RISS dataset (Sgandurra, 2016; Kok *et al.*, 2019)
Ransomware dynamic analysis log (Takeuchi, 2018)
The Zoo malware repository
Encryption Boundary Identification Algorithm
CR Pre-encryption dataset

Fig. 2. The datasets involved in this study

5.2 Phase #2: Framework Development

The proposed framework contains two important modules:

i. The RENTAKA-algorithm – this algorithm is developed using PHP. It identifies the ransomware's pre-encryption boundary and group the features into multiple ransomware subphases such as pre-encryption, during encryption and post-encryption.
ii. Pre-encryption features extraction – once the boundary has been identified and flagged by the algorithm, pre-encryption features are identified, extracted, and stored in a comma separated value (CSV) file.

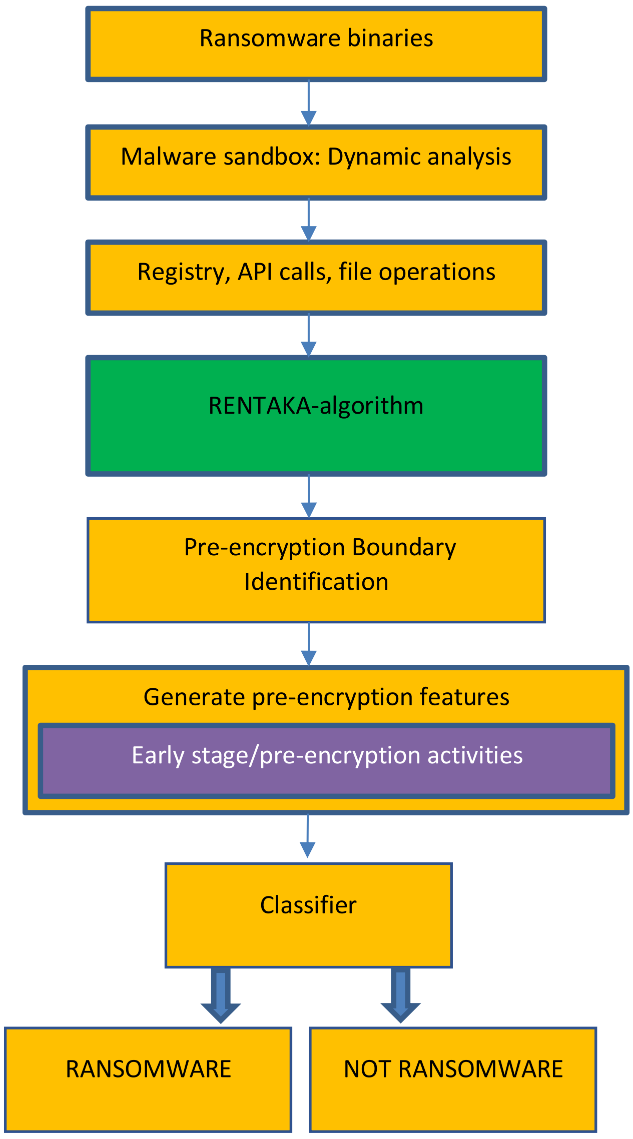

Ransomware binaries
Malware sandbox: Dynamic analysis
Registry, API calls, file operations
RENTAKA-algorithm
Pre-encryption Boundary Identification
Generate pre-encryption features
Early stage/pre-encryption activities
Classifier
RANSOMWARE
NOT RANSOMWARE

Fig. 3. The proposed framework

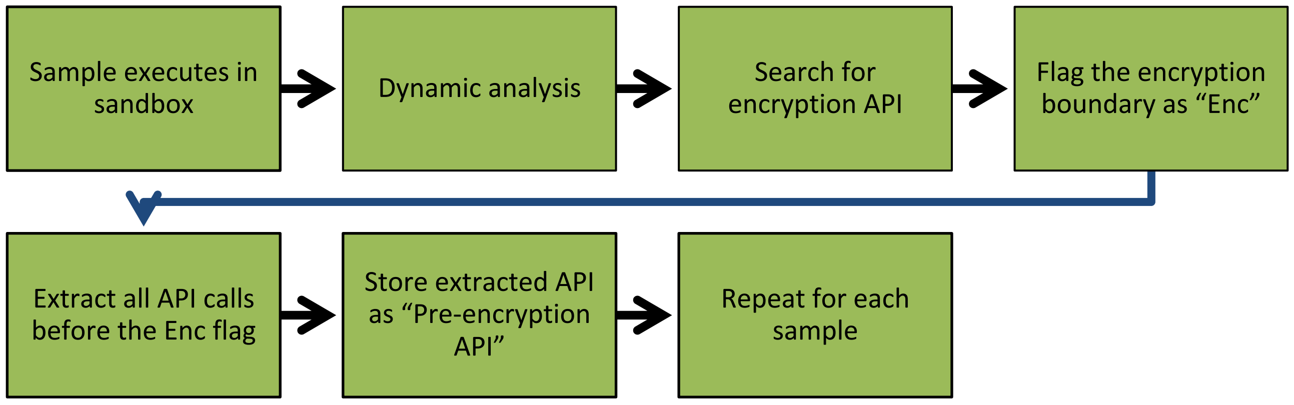

Sample executes in sandbox
Dynamic analysis
Search for encryption API
Flag the encryption boundary as "Enc"
Extract all API calls before the Enc flag
Store extracted API as "Pre-encryption API"
Repeat for each sample

Fig. 4. Flow of the proposed algorithm

6. Machine Learning

Machine learning is a subfield of artificial intelligence (AI) concerned with the development of algorithms and models that enable computers to learn from data and make predictions or decisions based on that data [27-31]. It is a data analysis technique that automates the construction of analytical models, enabling computers to adapt to new data and enhance their performance over time without being explicitly programmed. There are a number of different machines learning methodologies, including:

i. Supervised learning [28,32]: In this method, the algorithm is trained on a labelled dataset in which input and output data are paired. This data is used to train the model, which is then used to make predictions on new, unobserved data.
ii. Unsupervised learning [33]: The algorithm is given an unlabelled dataset and must independently identify patterns and relationships within the data. Clustering, in which the model clusters similar data points, and dimensionality reduction, which reduces the number of variables in a dataset, are common techniques.
iii. In reinforcement learning, the model learns by interacting with its environment and receiving rewards or punishments as feedback [34]. The objective is to determine the optimal course of action in each circumstance to maximize the cumulative recompense.

Natural language processing, computer vision, speech recognition, recommendation systems, medical diagnosis, malware detection, fraud detection, and self-driving vehicles are just a few of the many domains in which machine learning has applications [35]. In the field of cybersecurity, machine learning is becoming increasingly essential. It assists organizations in detecting and responding to hazards more efficiently and effectively. Examples of machine learning applications in cybersecurity include:

i. Malware detection: Machine learning algorithms can be taught to analyse and classify files as malicious or benign based on their characteristics, including file size, API calls, and behavioural patterns [31,36,37]. This enables the detection of malware, including previously unknown or zero-day threats, to occur more quickly and precisely.

ii. Anomaly detection: Machine learning can be utilized to establish a baseline of normal system or network behaviour. By perpetually monitoring and analysing data, the algorithms can detect deviations from this baseline and flag potential security incidents such as unauthorized access, data breaches, and distributed denial-of-service (DDoS) attacks.
iii. Machine learning models can be trained to identify fraudulent emails by analysing sender information, email content, and URLs, among other features [29,38]. This protects users from phishing attacks, which are frequently designed to take sensitive information such as login credentials or personal data.
iv. Spam filtering: By analysing the content and patterns of emails, machine learning can be used to improve spam filtering algorithms, enabling for the identification, and filtering of spam messages more effectively.
v. Network traffic analysis can be applied with machine learning to detect intrusions or unauthorized activities [39-42]. By analysing network traffic patterns, the algorithms can detect anomalous activity, such as data exfiltration or unauthorized access attempts, and notify security teams.
vi. By analysing historical data on security incidents, vulnerabilities, and threat intelligence, machine learning can assist businesses in assessing the risk associated with their assets and operations. This allows organizations to effectively prioritize their security efforts and allocate resources.

Machine learning improves cybersecurity by automating the detection, response, and prevention of threats. It enables organizations to identify and respond to potential security incidents with greater speed and efficiency, thereby reducing the risk of intrusions and the associated costs.

7. API Calls as Features

API calls can be extracted from malware in a scalable and automated manner, making them simple to process and analyse in large datasets. This is crucial because ransomware attacks are becoming more prevalent and sophisticated, and machine learning-based approaches must be able to process large amounts of data rapidly in order to detect and respond to attacks in real-time. Using API calls as features in machine learning-based ransomware detection offers several advantages over code-based features, including capturing the malware's behaviour, being more resistant to obfuscation techniques, and being consistent across different ransomware versions. This approach is easily scalable and can aid organisations in detecting and responding swiftly to ransomware attacks, thereby minimising their impact on operations and finances. Using API queries as features in machine learning-based ransomware detection has several advantages [43-49].

API calls are a potent feature for detecting ransomware because they capture the behaviour of the malware as it runs, providing a wealth of information for differentiating ransomware from benign samples. This is significant because ransomware is designed to modify the system's behaviour to accomplish its objectives, such as encrypting files or demanding payment. By using API calls as features, we can identify ransomware-induced changes in behaviour and distinguish them from innocuous behaviour.

By this approach it is more resistant to obfuscation techniques.[7,50-52] Ransomware authors frequently use obfuscation techniques, such as packing, encryption, and polymorphism, to conceal their malicious code from detection. However, these techniques cannot conceal the malware's behaviour while it is running, making API calls a more reliable method for detecting ransomware. By analysing the API calls made by malware, regardless of the obfuscation techniques employed, we can determine its underlying behaviour. Consistent across ransomware variants: ransomware typically arrives in multiple versions, each with its own code. API calls are more effective than code-based features for detecting ransomware due to the consistency of the malware's behaviour between various versions. By focusing on the malware's behaviour rather than its code, we can detect novel ransomware variants that may not be included in signature-based antivirus software.

8. Related Works

This paper provides an overview of various static analysis techniques that can be used to detect and analyse ransomware[31,53,54]. One of the techniques that the authors discuss is analysing the API calls made by ransomware. By analysing the API calls, it is possible to identify suspicious behaviour, such as accessing sensitive system resources or communicating with a remote server. The authors note that while API call analysis can be effective, it may not always be sufficient on its own to detect ransomware, as some malware may use legitimate APIs in their malicious activities. Overall, this paper highlights the importance of API call analysis as a component of a comprehensive ransomware detection and analysis approach.

The papers propose a ransomware detection technique based on analysing the API calls made by an application. The authors use machine learning algorithms to classify API calls as either malicious or benign, based on a dataset of known ransomware samples. The proposed technique achieves high accuracy and can detect new ransomware variants that were not present in the training dataset. However, the authors note that the technique may not be effective against ransomware that uses advanced evasion techniques, such as code obfuscation or polymorphism. Presents a machine learning-based approach to classify malicious API calls used by ransomware. The authors extract a set of features from the API calls, such as the frequency and duration of calls, and use them to train several machine learning classifiers. The proposed approach achieves high accuracy and outperforms other state-of-the-art techniques. The authors note that the proposed technique can be extended to detect other types of malwares beyond ransomware.

This paper [44]proposed a behaviour-based malware detection approach that analyses the API calls made by an application. The authors extract a set of features from the API calls, such as the sequence of calls and the parameters passed to them, and use them to train several machine learning classifiers. The proposed approach achieves high accuracy and can detect previously unknown malware samples. The authors note that the proposed technique can be used in conjunction with other detection techniques, such as signature-based detection, to improve detection rates.

9. Results

The accuracy of a model is the proportion of correct predictions relative to the total number of predictions. In ransomware detection, accuracy measures how accurately the EldeRan framework identifies ransomware samples as ransomware and non-ransomware samples as non-ransomware. True Positive Rate (TPR), also known as sensitivity or recall, is the proportion of ransomware samples the algorithm correctly classifies as ransomware. A high TPR indicates that the EldeRan framework can detect ransomware samples without incorrectly classifying them as non-ransomware.

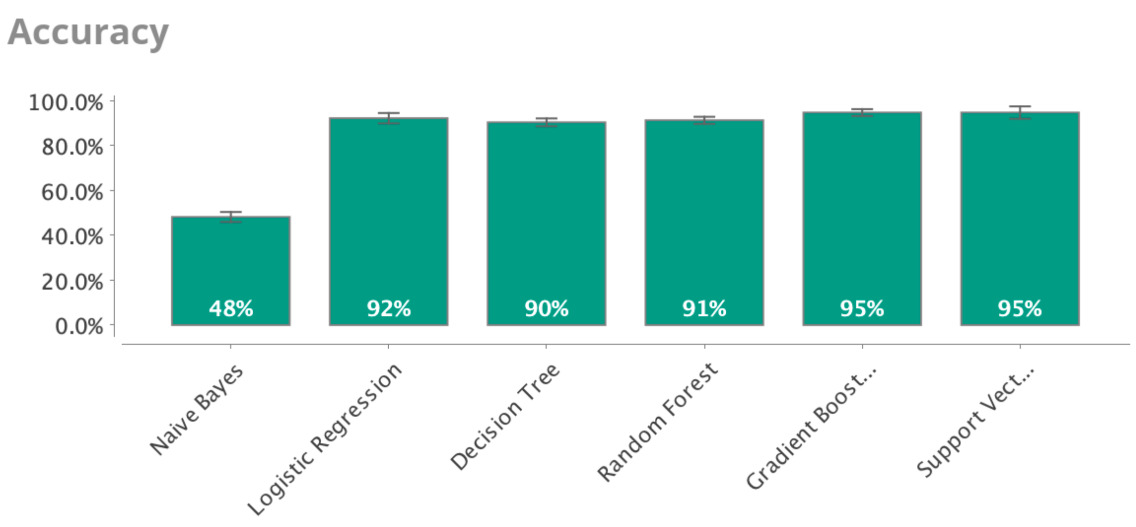
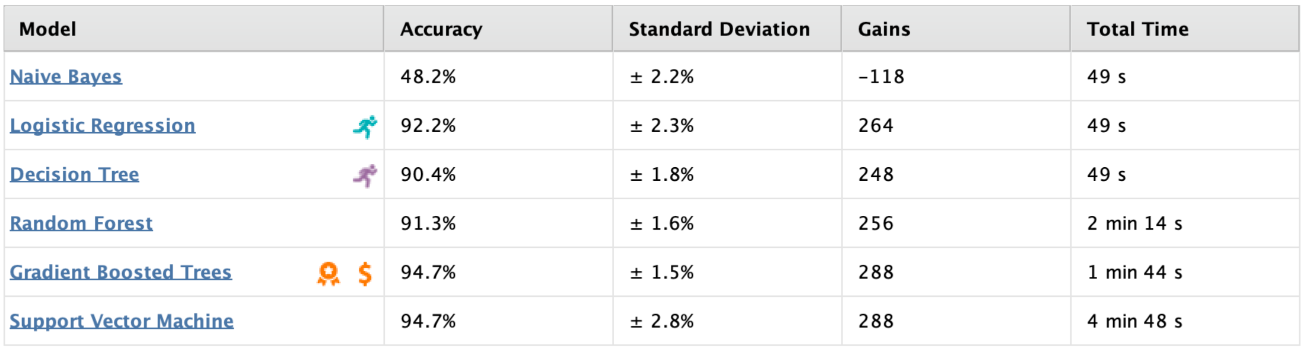


Fig. 5. Result for accuracy of each classifier



Model	Accuracy	Standard Deviation	Gains	Total Time
Naive Bayes	48.2%	± 2.2%	-118	49 s
Logistic Regression	92.2%	± 2.3%	264	49 s
Decision Tree	90.4%	± 1.8%	248	49 s
Random Forest	91.3%	± 1.6%	256	2 min 14 s
Gradient Boosted Trees	94.7%	± 1.5%	288	1 min 44 s
Support Vector Machine	94.7%	± 2.8%	288	4 min 48 s

Fig. 6. Results from Rapidminer

i. Naive Bayes (48.00%): Naive Bayes has the lowest accuracy among the classifiers mentioned. Its performance might be affected by the independence assumption, where it assumes that the features are independent of each other. This assumption might not hold true for the dataset being used, leading to a lower accuracy.
ii. Logistic Regression (92.21%): This linear model performs significantly better than Naive Bayes. It tries to model the probability of an instance belonging to a particular class using a logistic function. With 92.21% accuracy, it is a decent classifier for many applications.
iii. Decision Tree (90.37%): Decision Trees work by recursively splitting the data based on the feature that provides the highest information gain. The accuracy is slightly lower than Logistic Regression but still considered a good classifier for many use cases.
iv. Random Forest (91.28%): This ensemble method uses multiple Decision Trees to make predictions. It is generally more accurate and robust than a single Decision Tree. In this case, its accuracy is higher than the Decision Tree but lower than Logistic Regression.
v. Gradient Boosted Trees (94.73%): This is another ensemble method that combines weak learners (Decision Trees) to form a strong learner. It works by iteratively adding trees to the model while correcting the errors made by previous trees. In this comparison, Gradient Boosted Trees have the highest accuracy along with the Support Vector Machine.
vi. Support Vector Machine (94.73%): SVM is a powerful and flexible classifier that tries to find the best decision boundary between classes by maximizing the margin. It performs

well in high-dimensional spaces and has the same highest accuracy as Gradient Boosted Trees in this comparison.

Table 1
Experimental results from all classifiers

Model	AUC	Gains	Total Time	Training Time (1,000 Rows)
Naive Bayes	0.614	-118.0	49269.0	113.5
Logistic Regression	0.951	264.0	48624.0	326.8
Decision Tree	0.931	248.0	49017.0	133.2
Random Forest	0.978	256.0	134933.0	402.2
Gradient Boosted Trees	0.984	288.0	104700.0	1436.4
Support Vector Machine	0.983	288.0	288208.0	647.0

The Area Under the Curve (AUC) is a performance metric used to evaluate the performance of a classification model. Specifically, AUC refers to the area under the Receiver Operating Characteristic (ROC) curve, which is a plot of the True Positive Rate (TPR) against the False Positive Rate (FPR) at various classification thresholds. The AUC value ranges from 0 to 1, with a value of 0.5 indicating that the model's performance is no better than random chance, and a value of 1 indicating that the model perfectly classifies all samples. Classification error in machine learning is the difference between the predicted class labels (output) and the actual class labels (ground truth) for a given dataset. In other words, it quantifies the number of inaccurate predictions generated by a classification algorithm. Classification error is essential because it aids in evaluating a machine learning model's performance. A model with a high classification error may not be accurate or trustworthy, whereas a model with a low classification error is more likely to be accurate and useful for the given task. Table 2 below shows the classification errors for all classifiers:

Table 2
Classification error and training time for all classifiers

Model	Classification Error	Gains	Total Time	Training Time (1,000 Rows)
Naïve Bayes	0.5184	-118.0	49269.0	113.5
Logistic Regression	0.0779	264.0	48624.0	326.8
Decision Tree	0.0963	248.0	49017.0	133.2
Random Forest	0.0872	256.0	134933.0	402.2
Gradient Boosted Trees	0.0527	288.0	104700.0	1436.4
Support Vector Machine	0.0527	288.0	288208.0	647.0

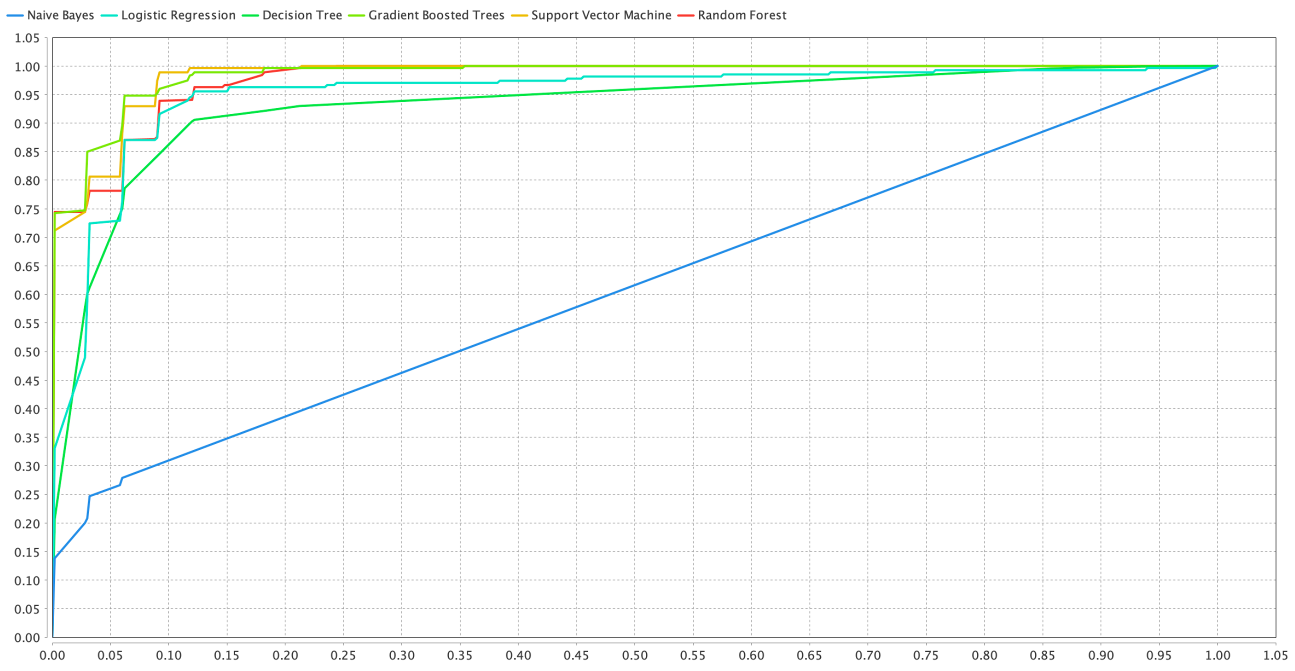


Fig. 7. ROC comparison for each classifier

Naïve bayes classifier has the lowest AUC (0.614) and the maximum standard deviation (0.1), indicating relatively poor performance and greater variation in the results. In addition, it has negative gains (-118.0), indicating that it does not add substantial value. Despite its short training period, Naïve Bayes does not appear to be an appropriate candidate for this specific task. With an AUC of 0.951, this classifier demonstrates excellent performance using logistic regression. In addition, it has a low training duration for 1,000 rows and respectable gains (264.0). Given its performance and computational efficiency, Logistic Regression seems to be a suitable candidate for this endeavour. This classifier's AUC is 0.931, which is lower than that of Logistic Regression. Despite this, it has a brief training period and reasonable gains (248.0). It could be a suitable option if interpretability and training speed are more crucial than AUC maximisation. With an AUC of 0.978, Random Forest demonstrates strong performance. However, it requires more training time and yields marginally fewer gains (256,0) than Logistic Regression. If the improved precision outweighs the reduction in computational efficiency, it may be a suitable option.

Gradient Boosted Trees classifier has the highest AUC (0.984) and gains (288.0), indicating outstanding performance, alongside SVM. However, its training time for 1,000 rows is the longest, making it computationally costly. GBT may be the finest option if precision is prioritised over training duration. SVM has a slightly lower AUC (0.983) than GBT, but its efficacy is still impressive. It shares the highest gains (288.0) with GBT, despite its significantly longer total duration. The training time for SVM for 1,000 rows is shorter than that of GBT but longer than that of other classifiers. If high gains and a strong AUC are prioritised and a longer total duration is acceptable, it may be a suitable option.

Gradient Boosted Trees and Support Vector Machines have the highest AUC and gains among classifiers, signifying superior performance. However, they require lengthier training periods, which may be a concern for some applications. Logistic Regression and Random Forest offer competitive performance and superior computational efficiency, making them viable alternatives when training time and total time are crucial.

10. Conclusion

Naïve Bayes classifier shows relatively poor performance with the lowest AUC (0.614), maximum standard deviation (0.1), and negative gains (-118.0), making it unsuitable for the task. Logistic Regression, on the other hand, performs well with an AUC of 0.951, low training time for 1,000 rows, and notable gains (264.0), making it a suitable candidate for this project. Decision Tree classifier has a slightly lower AUC (0.931) compared to Logistic Regression, but with a short training time and reasonable gains (248.0), it could be a suitable choice if interpretability and training speed are prioritized over AUC maximization. Random Forest has a high AUC (0.978) and decent gains (256.0), but it requires more training time. It may be a suitable option if improved accuracy is more important than computational efficiency. Gradient Boosted Trees (GBT) classifier boasts the highest AUC (0.984) and gains (288.0), indicating excellent performance, similar to SVM. However, GBT has the longest training time for 1,000 rows, making it computationally expensive. It may be the best choice if accuracy is prioritized over training time. SVM has a marginally lower AUC (0.983) but shares the highest gains (288.0) with GBT. Its training time for 1,000 rows is shorter than GBT but longer than other classifiers. It may be suitable if high gains, strong AUC, and longer total duration are acceptable.

Gradient Boosted Trees and Support Vector Machines offer the highest AUC and gains, signifying superior performance, but require longer training times, which may be a concern for some applications. Logistic Regression and Random Forest provide competitive performance with better computational efficiency, making them viable alternatives when training time and total time are crucial factors.

This research demonstrates the significance of API call analysis for ransomware detection and analysis. By analysing an application's API interactions, it is possible to detect and potentially prevent ransomware attacks. Particularly effective are machine learning-based approaches, which can learn from large datasets of known malware samples and identify previously unseen variants. It is essential to note, however, that malware authors are constantly adapting their techniques, and detection methods must also adapt to maintain pace with new threats.

Acknowledgement
This research was not funded by any grant.

References

[1] Moussaileb, Routa, Benjamin Bouget, Aurélien Palisse, Hélène Le Bouder, Nora Cuppens, and Jean-Louis Lanet. "Ransomware's early mitigation mechanisms." In *Proceedings of the 13th international conference on availability, reliability and security*, pp. 1-10. 2018. https://doi.org/10.1145/3230833.3234691

[2] Moussaileb, Routa, Benjamin Bouget, Aurélien Palisse, Hélène Le Bouder, Nora Cuppens, and Jean-Louis Lanet. "Ransomware's early mitigation mechanisms." In *Proceedings of the 13th international conference on availability, reliability and security*, pp. 1-10. 2018. https://doi.org/10.1145/3230833.3234691

[3] Beaman, Craig, Ashley Barkworth, Toluwalope David Akande, Saqib Hakak, and Muhammad Khurram Khan. "Ransomware: Recent advances, analysis, challenges and future research directions." *Computers & security* 111 (2021): 102490. https://doi.org/10.1016/j.cose.2021.102490

[4] Mattei, Tobias A. "Privacy, confidentiality, and security of health care information: Lessons from the recent Wannacry cyberattack." *World neurosurgery* 104 (2017): 972-974. https://doi.org/10.1016/j.wneu.2017.06.104

[5] Gagneja, Kanwalinderjit K. "Knowing the ransomware and building defense against it-specific to healthcare institutes." In *2017 Third International Conference on Mobile and Secure Services (MobiSecServ)*, pp. 1-5. IEEE, 2017. https://doi.org/10.1109/MOBISECSERV.2017.7886569

[6] Mansfield-Devine, Steve. "Leaks and ransoms–the key threats to healthcare organisations." *Network Security* 2017, no. 6 (2017): 14-19. https://doi.org/10.1016/S1353-4858(17)30062-4

[7] Alazab, Mamoun, Sitalakshmi Venkatraman, Paul Watters, Moutaz Alazab, and Ammar Alazab. "Cybercrime: the case of obfuscated malware." In *Global Security, Safety and Sustainability & e-Democracy: 7th International and*

4th e-Democracy, Joint Conferences, ICGS3/e-Democracy 2011, Thessaloniki, Greece, August 24-26, 2011, Revised Selected Papers, pp. 204-211. Springer Berlin Heidelberg, 2012. https://doi.org/10.1007/978-3-642-33448-1_28

[8] Pathak, P. B., and Yeshwant Mahavidyalaya Nanded. "A dangerous trend of cybercrime: ransomware growing challenge." *International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)* 5, no. 2 (2016): 371-373.

[9] Adam, Kujawa, Reed Thomas, Orozco Armando, Collier Nathan, Segura Jérôme, McNeil Adam, Tsing William, and Boyd Christopher. *Cybercrime tactics and techniques Q1 2017*. Malwarebytes, Tech. Rep., 2017.[Online]. Available: https://www. malwarebytes. com/pdf/labs/Cybercrime-Tactics-and-Techniques-Q1-2017. pdf.

[10] Hopkins, Michael, and Ali Dehghantanha. "Exploit Kits: The production line of the Cybercrime economy?." In *2015 second international conference on Information Security and Cyber Forensics (InfoSec)*, pp. 23-27. IEEE, 2015. https://doi.org/10.1109/InfoSec.2015.7435501

[11] Connolly, Lena Y., and David S. Wall. "The rise of crypto-ransomware in a changing cybercrime landscape: Taxonomising countermeasures." *Computers & Security* 87 (2019): 101568. https://doi.org/10.1016/j.cose.2019.101568

[12] Torres, P. E. P., and S. G. Yoo. "Detecting and neutralizing encrypting Ransomware attacks by using machine-learning techniques: A literature review." *Int. J. Appl. Eng. Res* 12, no. 18 (2017): 7902-7911.

[13] Urooj, Umara, Mohd Aizaini Bin Maarof, and Bander Ali Saleh Al-rimy. "A proposed adaptive pre-encryption crypto-ransomware early detection model." In *2021 3rd International Cyber Resilience Conference (CRC)*, pp. 1-6. IEEE, 2021. https://doi.org/10.1109/CRC50527.2021.9392548

[14] Dion, Y., and Sarfraz N. Brohi. "An experimental study to evaluate the performance of machine learning alogrithms in ransomware detection." *Journal of Engineering Science and Technology* 15, no. 2 (2020): 967-981.

[15] Jethva, Brijesh, Issa Traoré, Asem Ghaleb, Karim Ganame, and Sherif Ahmed. "Multilayer ransomware detection using grouped registry key operations, file entropy and file signature monitoring." *Journal of Computer Security* 28, no. 3 (2020): 337-373. https://doi.org/10.3233/JCS-191346

[16] Poudyal, Subash, Dipankar Dasgupta, Zahid Akhtar, and Kishor Gupta. "A multi-level ransomware detection framework using natural language processing and machine learning." In *14th International Conference on Malicious and Unwanted Software" MALCON*, no. October 2015. 2019.

[17] Zimba, Aaron, Zhaoshun Wang, and Hongsong Chen. "Multi-stage crypto ransomware attacks: A new emerging cyber threat to critical infrastructure and industrial control systems." *Ict Express* 4, no. 1 (2018): 14-18. https://doi.org/10.1016/j.icte.2017.12.007

[18] Patyal, Manveer, Srinivas Sampalli, Qiang Ye, and Musfiq Rahman. "Multi-layered defense architecture against ransomware." *International Journal of Business and Cyber Security* 1, no. 2 (2017).

[19] Kiru, Muhammad Ubale, and Aman Jantan. "Ransomware Evolution: Solving Ransomware Attack Challenges." In *The Evolution of Business in the Cyber Age*, pp. 193-229. Apple Academic Press, 2020. https://doi.org/10.1201/9780429276484-9

[20] Nguyen, Duc Thang, and Soojin Lee. "Lightgbm-based ransomware detection using api call sequences." *International Journal of Advanced Computer Science and Applications* 12, no. 10 (2021). https://doi.org/10.14569/IJACSA.2021.0121016

[21] Loman, Mark. "How ransomware attacks." *PDFs/technical-papers/sophoslabs-ransomware-behavior-report. pdf* (2019).

[22] Bansal, Urvashi. "A review on ransomware attack." In *2021 2nd International Conference on Secure Cyber Computing and Communications (ICSCCC)*, pp. 221-226. IEEE, 2021.

[23] Kumar, P. Ravi, and Hj Rudy Erwan Bin Hj Ramlie. "Anatomy of ransomware: attack stages, patterns and handling techniques." In *Computational Intelligence in Information Systems: Proceedings of the Computational Intelligence in Information Systems Conference (CIIS 2020)*, pp. 205-214. Springer International Publishing, 2021. https://doi.org/10.1007/978-3-030-68133-3_20

[24] Kok, S. H., Azween Abdullah, and N. Z. Jhanjhi. "Early detection of crypto-ransomware using pre-encryption detection algorithm." *Journal of King Saud University-Computer and Information Sciences* 34, no. 5 (2022): 1984-1999. https://doi.org/10.1016/j.jksuci.2020.06.012

[25] Kok, S. H., A. Azween, and N. Z. Jhanjhi. "Evaluation metric for crypto-ransomware detection using machine learning." *Journal of Information Security and Applications* 55 (2020): 102646. https://doi.org/10.1016/j.jisa.2020.102646

[26] Al-Rimy, Bander Ali Saleh, Mohd Aiziani Maarof, Mamoun Alazab, Fawaz Alsolami, Syed Zainudeen Mohd Shaid, Fuad A. Ghaleb, Tawfik Al-Hadhrami, and Abdullah Marish Ali. "A pseudo feedback-based annotated TF-IDF technique for dynamic crypto-ransomware pre-encryption boundary delineation and features extraction." *IEEE Access* 8 (2020): 140586-140598. https://doi.org/10.1109/ACCESS.2020.3012674

[27] Cusack, Greg, Oliver Michel, and Eric Keller. "Machine learning-based detection of ransomware using SDN." In *Proceedings of the 2018 ACM international workshop on security in software defined networks & network function virtualization*, pp. 1-6. 2018. https://doi.org/10.1145/3180465.3180467
[28] Jiang, Tammy, Jaimie L. Gradus, and Anthony J. Rosellini. "Supervised machine learning: a brief primer." *Behavior Therapy* 51, no. 5 (2020): 675-687. https://doi.org/10.1016/j.beth.2020.05.002
[29] Jordan, Michael I., and Tom M. Mitchell. "Machine learning: Trends, perspectives, and prospects." *Science* 349, no. 6245 (2015): 255-260. https://doi.org/10.1126/science.aaa8415
[30] Torres, P. E. P., and S. G. Yoo. "Detecting and neutralizing encrypting Ransomware attacks by using machine-learning techniques: A literature review." *Int. J. Appl. Eng. Res* 12, no. 18 (2017): 7902-7911.
[31] Almousa, May, Sai Basavaraju, and Mohd Anwar. "Api-based ransomware detection using machine learning-based threat detection models." In *2021 18th International Conference on Privacy, Security and Trust (PST)*, pp. 1-7. IEEE, 2021. https://doi.org/10.1109/PST52912.2021.9647816
[32] Adamu, Umaru, and Irfan Awan. "Ransomware prediction using supervised learning algorithms." In *2019 7th International Conference on Future Internet of Things and Cloud (FiCloud)*, pp. 57-63. IEEE, 2019. https://doi.org/10.1109/FiCloud.2019.00016
[33] Zahoora, Umme, Asifullah Khan, Muttukrishnan Rajarajan, Saddam Hussain Khan, Muhammad Asam, and Tauseef Jamal. "Ransomware detection using deep learning based unsupervised feature extraction and a cost sensitive Pareto Ensemble classifier." *Scientific Reports* 12, no. 1 (2022): 15647. https://doi.org/10.1038/s41598-022-19443-7
[34] Fang, Zhiyang, Junfeng Wang, Jiaxuan Geng, and Xuan Kan. "Feature selection for malware detection based on reinforcement learning." *IEEE Access* 7 (2019): 176177-176187. https://doi.org/10.1109/ACCESS.2019.2957429
[35] Poudyal, Subash, Dipankar Dasgupta, Zahid Akhtar, and Kishor Gupta. "A multi-level ransomware detection framework using natural language processing and machine learning." In *14th International Conference on Malicious and Unwanted Software" MALCON*, no. October 2015. 2019.
[36] Scalas, Michele, Davide Maiorca, Francesco Mercaldo, Corrado Aaron Visaggio, Fabio Martinelli, and Giorgio Giacinto. "On the effectiveness of system API-related information for Android ransomware detection." *Computers & Security* 86 (2019): 168-182. https://doi.org/10.1016/j.cose.2019.06.004
[37] Obeis, Nawfal Turki, and Wesam Bhaya. "Malware analysis using APIs pattern mining." *International Journal of Engineering & Technology* 7, no. 3.20 (2018): 502-506.
[38] Chan, Philip K., and Richard P. Lippmann. "Machine learning for computer security." (2006).
[39] Moussaileb, Routa, Nora Cuppens, Jean-Louis Lanet, and Hélène Le Bouder. "Ransomware network traffic analysis for pre-encryption alert." In *Foundations and Practice of Security: 12th International Symposium, FPS 2019, Toulouse, France, November 5–7, 2019, Revised Selected Papers 12*, pp. 20-38. Springer International Publishing, 2020. https://doi.org/10.1007/978-3-030-45371-8_2
[40] Modi, Jaimin. "Detecting ransomware in encrypted network traffic using machine learning." PhD diss., 2019.
[41] Alhawi, Omar MK, James Baldwin, and Ali Dehghantanha. "Leveraging machine learning techniques for windows ransomware network traffic detection." *Cyber threat intelligence* (2018): 93-106. https://doi.org/10.1007/978-3-319-73951-9_5
[42] Atifi, Adil, and Elias Bou-Harb. "On correlating network traffic for cyber threat intelligence: A Bloom filter approach." In *2017 13th International Wireless Communications and Mobile Computing Conference (IWCMC)*, pp. 384-389. IEEE, 2017. https://doi.org/10.1109/IWCMC.2017.7986317
[43] Qin, Bin, Yalong Wang, and Changchun Ma. "API call based ransomware dynamic detection approach using textCNN." In *2020 International Conference on Big Data, Artificial Intelligence and Internet of Things Engineering (ICBAIE)*, pp. 162-166. IEEE, 2020. https://doi.org/10.1109/ICBAIE49996.2020.00041
[44] Alazab, Mamoun, Sitalakshmi Venkataraman, and Paul Watters. "Towards understanding malware behaviour by the extraction of API calls." In *2010 second cybercrime and trustworthy computing workshop*, pp. 52-59. IEEE, 2010. https://doi.org/10.1109/CTC.2010.8
[45] Alazab, Manoun, Robert Layton, Sitalakshmi Venkataraman, and Paul Watters. "Malware detection based on structural and behavioural features of api calls." (2010).
[46] Qiao, Yong, Yuexiang Yang, Lin Ji, and Jie He. "Analyzing malware by abstracting the frequent itemsets in API call sequences." In *2013 12th IEEE international conference on trust, security and privacy in computing and communications*, pp. 265-270. IEEE, 2013. https://doi.org/10.1109/TrustCom.2013.36
[47] Chen, Zhi-Guo, Ho-Seok Kang, Shang-Nan Yin, and Sung-Ryul Kim. "Automatic ransomware detection and analysis based on dynamic API calls flow graph." In *Proceedings of the international conference on research in adaptive and convergent systems*, pp. 196-201. 2017. https://doi.org/10.1145/3129676.3129704

[48] Ki, Youngjoon, Eunjin Kim, and Huy Kang Kim. "A novel approach to detect malware based on API call sequence analysis." *International Journal of Distributed Sensor Networks* 11, no. 6 (2015): 659101. https://doi.org/10.1155/2015/659101

[49] Kim, Chan Woo. "Ntmaldetect: A machine learning approach to malware detection using native api system calls." *arXiv preprint arXiv:1802.05412* (2018).

[50] You, Ilsun, and Kangbin Yim. "Malware obfuscation techniques: A brief survey." In *2010 International conference on broadband, wireless computing, communication and applications*, pp. 297-300. IEEE, 2010. https://doi.org/10.1109/BWCCA.2010.85

[51] Fan, Wenqing, Xue Lei, and Jing An. "Obfuscated malicious code detection with path condition analysis." *Journal of Networks* 9, no. 5 (2014): 1208. https://doi.org/10.4304/jnw.9.5.1208-1214

[52] Xu, Dongpeng, Jiang Ming, and Dinghao Wu. "Cryptographic function detection in obfuscated binaries via bit-precise symbolic loop mapping." In *2017 IEEE Symposium on Security and Privacy (SP)*, pp. 921-937. IEEE, 2017. https://doi.org/10.1109/SP.2017.56

[53] Sheen, Shina, and Ashwitha Yadav. "Ransomware detection by mining API call usage." In *2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, pp. 983-987. IEEE, 2018. https://doi.org/10.1109/ICACCI.2018.8554938

[54] Bajpai, Pranshu, and Richard Enbody. "An empirical study of api calls in ransomware." In *2020 IEEE International Conference on Electro Information Technology (EIT)*, pp. 443-448. IEEE, 2020. https://doi.org/10.1109/EIT48999.2020.9208284