Princess Sumaya University for Technology

King Hussein School for Computing Sciences

Malware Analysis in PDFs

Dr. Ammar Odeh

Software Engineering Course (13477)

Fall - 2022/2023

Prepared By:

Rand Abdel Fattah 20190231

Anton Bahou 20190144

Hassan Abualhaj 20200500

Mohammed Madi 20200386

Contents

Table of Figures

Table of Tables

Chapter 1: Problem Definition

1.1 Introduction

Malware, short for malicious software, has a self-explanatory name. Whether that malicious intent was to cause damage, leak information, or gain unauthorized access, it is without a doubt unwelcome. Just as technology has grown exponentially in its innovation and importance, the world of malware hasn't been slacking behind.

With cybersecurity risks being on the rise, the purposes and intents of malicious attack perpetrators became more varied. Cyberattacks over the last few years uncovered many culprits targeting governments and corporations. The motive for many of these attacks resides in growing the black market of activities including selling leaked private information, identity theft, bank fraud, etc.

Analyzing malware generally falls into two categories. The first is analyzing them statically, which means that the malware is examined without executing the malware itself. The second approach is analyzing them dynamically, which means that the malware is studied by looking at its behavior as it is being executed. The static approach is much simpler than the dynamic approach. Which approach to take is also dependent on the nature of the malware that is being analyzed. A malware being a virus, ransomware, or trojan etc. can have an effect or limitation on the way it will be treated in the analyzation process.

The rampant growth of the internet, cloud services, digital communication services and many more technologies has given malware a new medium to emerge through. The vast amounts of Portable Document Format (PDF) files being transferred between individuals makes them appealing to malicious attackers and threatening to unaware individuals to the same extent.

This paper will discuss the importance of analyzing malware and current systems that analyze malware in PDFs, and propose a potential system based on the shortcomings of the current ones.

1.2 System Description

Our system's main aim is to detect any Portable Document Format (PDF) file that is embedded with malware using machine learning.

Our system uses CNN and Random Forest algorithms and contains a model that detects embedded malware in PDF files.

In our approach supervised machine learning will be used, which means our dataset is labeled and will accurately classify previously unseen data into two labels; malicious or safe.

For our CNN model, the weights of the nodes will be tuned using back propagation. As for our random forest models, they will be tuned using a Grid Search algorithm.

Our models will be trained using 80% of the dataset, and their performance will be tested using the remaining 20%.

1.3 System Purpose

Rapid development of technology resulted in an increased use of PDFs. The side-effects of such an increase in use of this format brought forward new malware attacks on people. The attackers conceal their preferred method of malware into the PDF using steganography and wait for the user to fall into the trap.

The purpose of our research is to provide a method that prevents future cyberattacks through the use of Artificial Intelligence practices, these practices will bring forward a mechanism that understands the behavior of such malware as well as its purpose. These understandings will be established through

datasets of such attacks, then incorporated into machine learning models and statistical methods for a well-rounded mechanism.

Our intention is to deliver the best f1-score possible, as we believe security is one of the most important aspects of software engineering. We also want to deliver a method that can be utilized by many devices, no matter their capabilities, as people these days use an enormous range of devices each with a different processing power forcing our method to be executed in an acceptable complexity.

1.4 Problem Statement

PDF files are some of the most common files for an attacker to use as a “trojan”. Simply because it's very easy to convince a victim to use it through social engineering. Maybe it looks like a research paper, or some class notes from a friend. Luckily A Lot of these attacks can be stopped if the attack is using a previously seen “Signature”. A signature is a pattern in the malware that allows security software to detect malicious software. It could be a sequence of bytes over a network or a hidden sequence of code within a file, in this case within a PDF. The issue arises when an attack uses a never-before-seen signature, probably an undiscovered vulnerability; a so-called Zero-day attack.

This is a big problem because of the huge space of possible different signatures which have never been seen before. 80% of all successful data breaches in 2019 were the result of a zero-day attack. It's estimated that 42% of all attacks in 2021 were zero-day attacks.[3]

The focus of this paper is to offer a machine learning based alternative which performs better against zero-day attacks while also performing well against previously seen signatures.

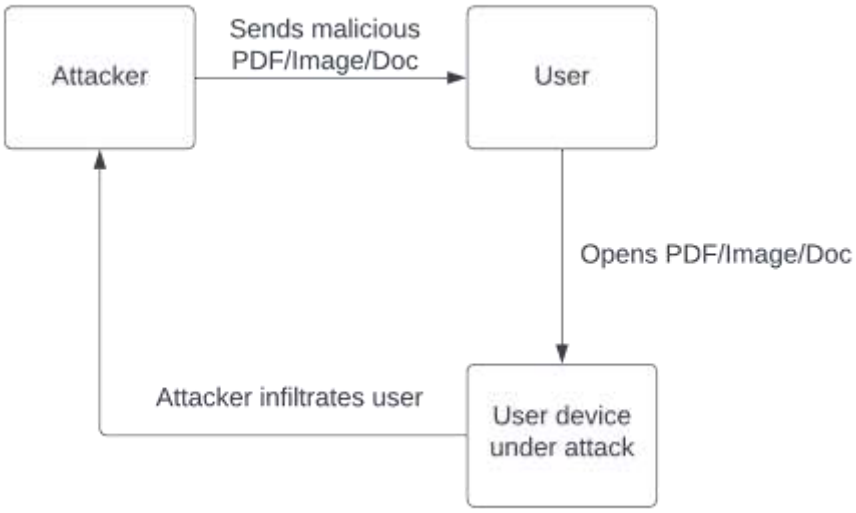

Figure 1: Malicious PDF/Image/Doc Attack Scenario

1.5 System Context View

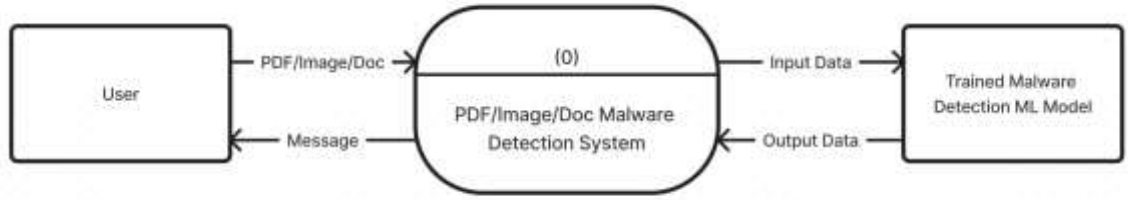

Figure 2: The System Context View

1.6 Literature Review

Dr. Robert Chun, Dr. Mike Wu, and Mr. Navneet Goel studied the structure of PDFs and why it's a sweet spot for attackers to embed several viruses and malware. They also discovered some techniques that detect malicious files such as forensic analysis. They talked about the importance of using machine learning to analyze and stop attacks. They also discussed the need for continuous enhancements of machine learning techniques, as attackers will exploit the vulnerabilities in these techniques sooner or later.[1]

Priyansh Singh, Shashikala Tapaswi, and Sanchit Gupta mentioned in their paper that in order to attack a document you can either target the document itself, target a software that opens this document, or target both of them. To target a certain document, you need to exploit some of the features of this document. When creating a document with interactive features or visual elements like audio or video, you can easily embed a visual element that loads and executes malware without the user noticing any foreign actions as these elements are normally provided by the environment. This method is called the exploitation of programming and interactive application features. While the structural attack uses the Crafted-Content method to target the software that opens certain documents, it changes the structure of the document which causes the software to react unexpectedly. Piggybacking on this reaction helps embedding malicious code in the memory for future execution. This might lead to a buffer overflow which is considered usually as a starting point in an attack.[2]

In A. Corum, D. Jenkins, and J. Zheng's paper, PDFs were transformed into a grayscale version using byte values and Markov plots to allow the extraction of important features for the development of the classification model. These features were based on keypoint descriptors and texture features. This model was created by a learning algorithm that used Random Forest, Decision Trees and K-Nearest Neighbor incorporated with mathematical functions in the field of vectors to detect malware in PDFs. Their performance was similar to the level of popular antivirus scanners. The dataset used was from Contagio.[4]

Another interesting research was done by F. Mercaldo and A. Santone. The researchers also converted the images into a PNG grayscale version; they used a script on a GitHub repository to achieve that.[5] The grayscale image was then converted into a histogram of pixel intensity values that was entered into a 3-layer deep neural network. The neural network was responsible for detecting if the image is a malware and finding out which family of malware it belonged to. The dataset used belonged to AMD. 17 different Algorithms were used and compared, 12 of them were deep learning algorithms.[6]

This paper, which was written by H. D. Samuel, M. S. Kumar, R. Aishwarya and G. Mathivanan, focused on the problems caused by multiple algorithms for steganography. A special software for template matching was used to solve this issue. Grayscale conversion was utilized again in combination with a machine learning algorithm to identify the spyware hidden in the RGB layer. The system requires a lot of resources however a high accuracy is achieved. The image goes through a process called regression

to separate it and classify it correctly according to the type of malware as part of the detection process and handled with accordingly.[7]

Since images are closely related to PDFs, papers [6] and [7] were used to find more viable ways of analysis for PDFs to be used in our approach.

In his paper, Sultan S. Alshamrani talked about how modern antivirus software have become outdated and don't offer enough protection against malicious PDFs. He proposes a Random Forest ML model that can identify JavaScript and malicious API calls in PDFs by analyzing the documents statically and dynamically using a non-signature method in order to perform well against zero-day malware attacks, as well as alternative Logistic Regression, Stochastic Gradient Boosting and Support Vector classifiers. The proposed model achieved an F1 score of 0.986, making it more efficient that all the tools referenced in the paper.[8]

This paper by K. O. Babaagba and S. O. Adesanya talks about the effects of feature selection and the ML approach used on the yielded results. The paper tests various supervised and unsupervised classification and clustering ML algorithms with and without feature selection. The paper concludes the machine learning approach in detecting malware is effective, and that using feature selection based on Information Gain leads to better values in performance metrics. The detection conducted in this paper is focused on viruses, and states that similar work should be done on other types of malwares to observe the process more accurately.[9]

Ke He and Dong Kim explain that RNN (Recurrent Neural Network), the most widely used neural network for detecting malware, is vulnerable to redundant API injections. They propose an alternative technique which utilizes a CNN (Convolutional Neural Network), it works by turning the Malware information into an image, then processing it through the network. Doing this, as they explained, solves the redundant API injection problem.[10]

Seong Il Bae, Gyu Bin Lee and Eul Gyu Im explain that ransomware is different from normal malware, most normal signature-based detection methods have an issue detecting zero-day ransomware attacks. They offer an alternative machine learning based approach. First, they process the executable file and turn it into N-gram vectors using CF-NCF, which is similar to TF-ITF. Then, they use several distinguishable machine learning algorithms, the most notable was Random Forest which yielded an f1-score of 99.52.[11]

1.7 Challenges

- Finding, preprocessing, and merging the appropriate datasets. Getting data from multiple sources and ensuring that the data is preprocessed correctly to ensure optimal performance from the model.
- Choosing the correct ML models. Different models use various ML techniques. Finding the suitable model
- Fine tuning the models and their hyperparameters and checking their performance metrics.

1.8 Projection

Table 1: Phase 1 Tasks

Activity	Predecessor	Time (Days)
Planning and Splitting Tasks	-	1
Introduction	Planning and Splitting Tasks	2
System Description	Literature Review	1
System Purpose	Literature Review	1
Problem Statement	Planning and Splitting Tasks	2
System Context View	System Description	1
Literature Review	Introduction AND Problem Statement	3
Challenges	System Description	1
Projection	-	1

Table 2: Phase 2 Tasks

Activity	Predecessor	Time (Days)
Brainstorming	Finish Phase 1	3
Functional Requirements	Brainstorming	2
Non-Functional Requirements	Brainstorming	2
Class Diagram	FR and NFR	1
Use Cases	FR and NFR	2
Activity Diagram	FR and NFR	1

Table 3: Phase 3 Tasks

Activity	Predecessor	Time (Days)
Sequential Diagram	Finish Phase 2	1
Brainstorming	Sequential Diagram	5
Coding	Brainstorming	5
Prepare Simulation Results	Coding	2
Conclusion	Simulation Results	1

1.8.1 Gantt Charts

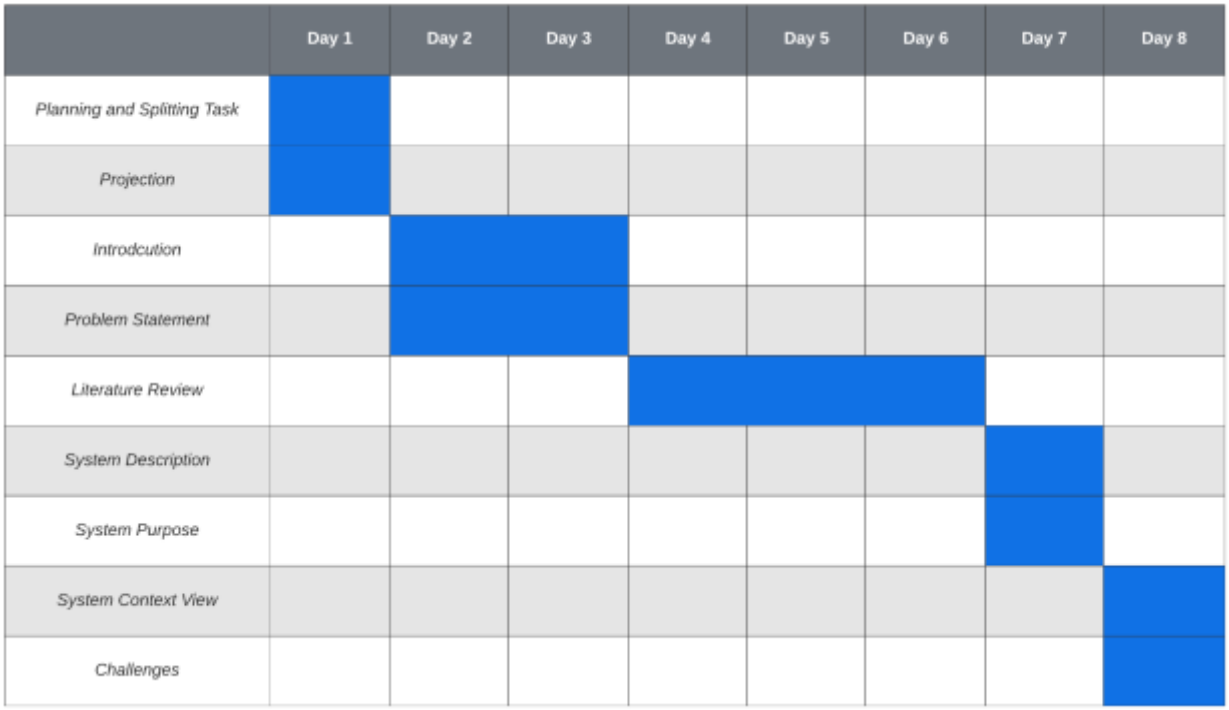

Figure 3: Phase 1 Gantt Chart

Figure 4: Phase 2 Gantt Chart

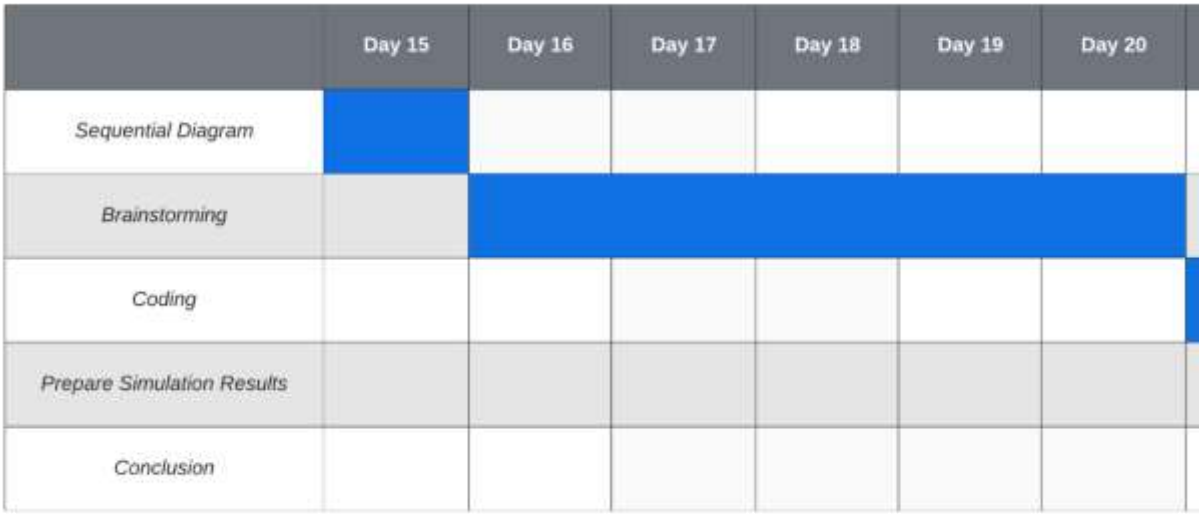

Figure 5: Phase 3.1 Gantt Chart

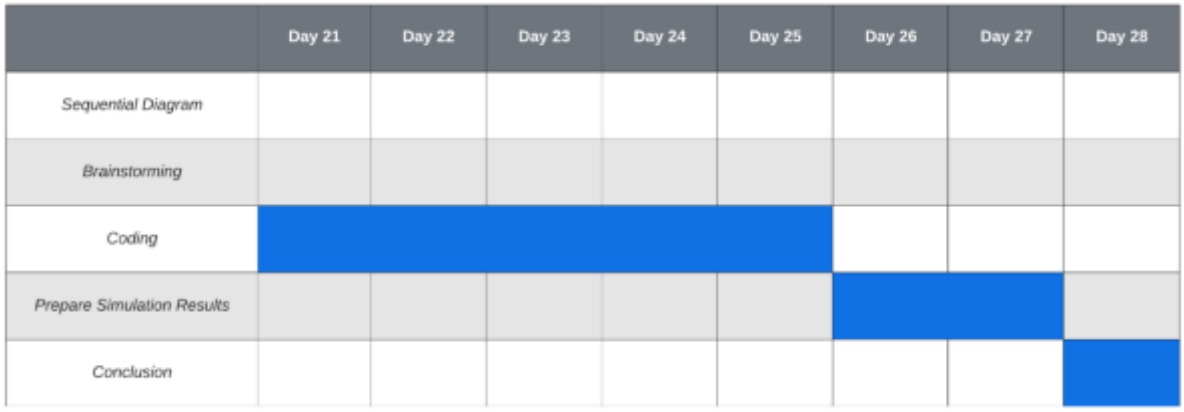

Figure 6: Phase 3.2 Gantt Chart

1.8.2 Network Diagrams

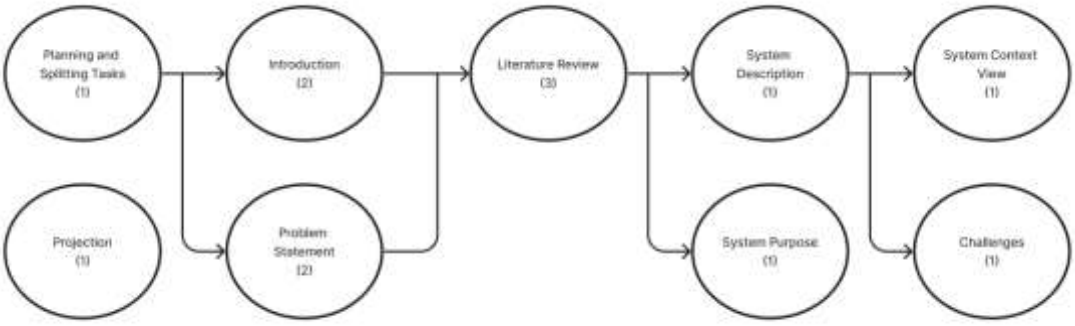

Figure 7: Phase 1 Network Chart

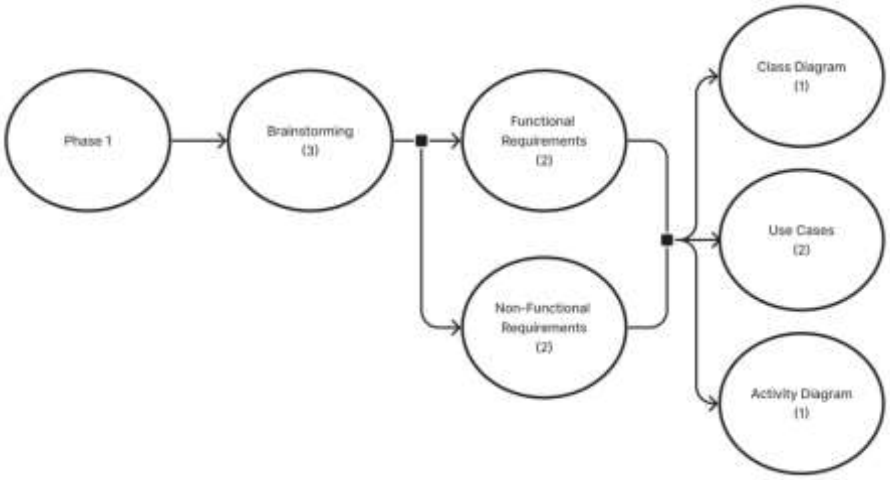

Figure 8: Phase 2 Network Chart

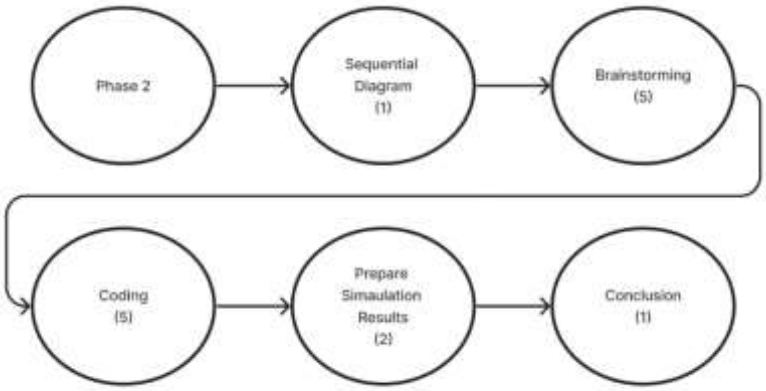

Figure 9: Phase 3 Network Chart

REFERENCES

[1] S. S. Pachpute, “Malware Analysis on PDF.” *San Jose State University Library*. doi: 10.31979/etd.pf8d-htjh.

[2] P. Singh, S. Tapaswi, and S. Gupta, “Malware Detection in PDF and Office Documents: A survey,” *Information Security Journal: A Global Perspective*, vol. 29, no. 3. Informa UK Limited, pp. 134–153, Feb. 13, 2020. doi: 10.1080/19393555.2020.1723747.

[3] “2022 cyber security statistics trends & data,” PurpleSec, 17-Oct-2022. [Online]. Available: https://purplesec.us/resources/cyber-security-statistics/#ZeroDay. [Accessed: 13-Nov-2022].

[4] A. Corum, D. Jenkins and J. Zheng, "Robust PDF Malware Detection with Image Visualization and Processing Techniques," *2019 2nd International Conference on Data Intelligence and Security (ICDIS)*, 2019, pp. 108-114, doi: 10.1109/ICDIS.2019.00024.

[5] Leeroybrun, “Leeroybrun/bin2png: Convert a binary file to a PNG image and then back to binary.,” GitHub. [Online]. Available: https://github.com/leeroybrun/Bin2PNG. [Accessed: 15-Nov-2022].

[6] F. Mercaldo and A. Santone, “Deep learning for image-based mobile malware detection,” *Journal of Computer Virology and Hacking Techniques, vol. 16, no. 2. Springer Science and Business Media LLC*, pp. 157–171, Jan. 13, 2020. doi: 10.1007/s11416-019-00346-7.

[7] H. D. Samuel, M. S. Kumar, R. Aishwarya and G. Mathivanan, "Automation Detection of Malware and Stenographical Content using Machine Learning," *2022 6th International Conference on Computing Methodologies and Communication (ICCMC)*, 2022, pp. 889-894, doi: 10.1109/ICCMC53470.2022.9754063.

[8] S. S. Alshamrani, “Design and Analysis of Machine Learning Based Technique for Malware Identification and Classification of Portable Document Format Files,” *Security and Communication Networks*, vol. 2022. Hindawi Limited, pp. 1–10, Sep. 21, 2022. doi: 10.1155/2022/7611741.

[9] K. O. Babaagba and S. O. Adesanya, “A Study on the Effect of Feature Selection on Malware Analysis using Machine Learning,” *Proceedings of the 2019 8th International Conference on Educational and Information Technology. ACM*, Mar. 02, 2019. doi: 10.1145/3318396.3318448.

[10] He, K. and Kim, D.-S. (2019) “Malware detection with malware images using Deep Learning Techniques,” 2019 18th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/13th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE) [Preprint]. Available at: https://doi.org/10.1109/trustcom/bigdatase.2019.00022.

[11] S. I. Bae, G. B. Lee, and E. G. Im, “Ransomware detection using machine learning algorithms,” *Concurrency and Computation: Practice and Experience*, vol. 32, no. 18, 2019.