Comprehensive Ransomware Detection: Optimization of Feature Selection through Machine Learning Algorithms and Explainable AI on Memory Analysis.

Lucas Leonel[1], **Diego Nunes Molinos**[1], **Rodrigo Sanches Miani**[1]

[1]Faculdade de Computação (FACOM) – Universidade Federal de Uberlândia (UFU)
Caixa Postal 593 – 38400-902 – Uberlândia – MG – Brazil

lucas.leonelcosta@gmail.com,{diego.molinos, miani}@ufu.br

Abstract. *The increase in ransomware attacks has underscored the need for robust cybersecurity measures. To combat these sophisticated threats, organizations must implement strong defenses, including cutting-edge technologies like machine learning to detect early signs of ransomware in their systems. This paper presents a comprehensive study on ransomware detection, highlighting the integration of machine learning algorithms and explainable artificial intelligence (XAI) techniques to enhance the transparency and reliability of predictive models in this field. Our focus relies on optimizing features within the CIC-MalMem-2022 dataset, which contains various memory-based malware samples. We also use the decision tree algorithm to identify influential features and uses the SHAP model for transparent decision-making. The results demonstrate that the algorithms can efficiently detect ransomware using only five optimized features.*

1. Introduction

Ransomware attacks represent a significant cybersecurity threat to society, not only corporations but also individuals, compromising the confidentiality and integrity of individuals and organizations, resulting in significant financial losses [Routray et al. 2023]. A recent study revealed a 33% increase in ransomware attacks in the Middle East between 2022 and 2024 [Aljabri et al. 2024]. These attacks have impacted approximately one in five companies (21%), with the average cost of repairing an attack amounting to US $1.85 million [Hornetsecurity 2022].

Ransomware often uses advanced obfuscation and polymorphism techniques, altering its code to avoid being detected by signature-based systems. It can also mimic legitimate behaviors such as file encryption, making it hard to differentiate between malicious activity and normal operations without generating false alarms. Furthermore, the speed with which ransomware can encrypt files often outpaces the ability of security methods to detect and respond effectively. In today's complex landscape, organizations must embrace dynamic and sophisticated strategies, including using artificial intelligence and machine learning technologies, to detect and counter ransomware threats effectively [Alraizza and Algarni 2023, Herrera-Silva and Hernández-Álvarez 2023, Bensaoud et al. 2024].

Memory forensic analysis techniques help efficiently detect ransomware by examining a system's physical memory for behavioral signs of ransomware. This process reveals malicious processes, Dynamic-Link Library (DLL) access, and threads that are not easily detected using other methods [Aljabri et al. 2024, Alraizza and Algarni 2023]. It was noted that the analysis of physical memory binaries tends to report a very large volume of information and application characteristics [Aljabri et al. 2024].

Under ransomware detection models based on machine learning, the volume of input data directly reflects the accuracy and performance of the model. Optimization in the feature selection process is crucial as it directly impacts the effectiveness and efficiency of detection models [Liu and Motoda 2007]. Adequate feature selection allows the model to focus on the most relevant characteristics of ransomware behavior, improving accuracy and speed in detecting malicious activity. In this context, the feature optimization process helps to eliminate redundant or irrelevant data that can confuse the model or cause *overfitting*, ensuring that the detection system is robust and agile. Furthermore, feature optimization reduces the computational complexity of the model, which is especially important in operational environments where hardware resources may be limited [Liu and Motoda 2007].

Several studies in the literature have explored various analysis techniques, both static and dynamic, using different data sources such as public datasets, Cuckoo Sandbox, Volatility framework for detecting ransomware using data from forensic memory analysis [Herrera-Silva and Hernández-Álvarez 2023, Malik et al. 2022, Aljabri et al. 2024]. In the feature selection process, we find that simple correlation helps identify the attributes that most influence ransomware detection. This leads to a logical reduction of the initial feature set. However, this process often neglects a more detailed analysis of the feature set, which could contribute to an even more significant reduction of the initial feature set, improving the model's precision by focusing on the highest-impact features.

In this context, this research aims to conduct a detailed investigation to enhance the selection of features for detecting ransomware through Machine Learning (ML) algorithms and Explainable Artificial Intelligence (XAI) techniques with memory data. The CIC-MalMem-2022 dataset was utilized for conducting the experiments. Our results indicate that refining and focusing on feature optimization using feature importance and XAI makes it possible to create reliable and high-performing models while still maintaining accuracy. It is important to note that due to the nature of the dataset, which consists of memory dump files, we performed an offline analysis. Our aim here is to present a case study on memory-based features that can be used to detect ransomware, rather than how to build end-to-end systems that would consume these features and detect malicious activities.

2. Theoretical Background

The following section will cover the essential conceptual framework for understanding ransomware detection through memory data analysis using ML techniques and XAI methods to enhance the transparency of predictive models. It will also discuss the critical role of feature optimization in the classification process.

2.1. Addressing Ransomware Detection Challenges

Detecting ransomware sets several significant challenges, primarily due to the constantly evolving nature of ransomware threats and the sophisticated methods used by cybercriminals. While static analysis effectively identifies samples with known signatures, it fails to detect new emerging threats. On the other hand, dynamic analysis can be more efficient, but running each suspicious sample requires specific time and computational resources [Dener et al. 2022].

The continuous development of new ransomware variants that bypass conventional security measures, including signature-based detection, has been one of the biggest challenges in detecting this threat. Additionally, ransomware often employs complex encryption techniques that activate before detection mechanisms can intervene, complicating efforts to prevent the attack [Beaman et al. 2021]. In general, ransomware can be a misleading threat that is difficult to detect using traditional security tools, even for the honeypot systems [Othman et al. 2024]. This is because its developers often use obfuscation tactics to mask their malicious code and evade detection.

As a result, it can be challenging for security professionals to identify and eliminate ransomware before it causes significant harm to computer systems and networks [Aslan and Samet 2020]. This means that more than advanced security measures may be needed to protect against ransomware attacks. [Naseer et al. 2021, Aslan and Samet 2020].

In response to those threats, there has been a significant evolution in approaches to utilize advanced analytics on in-memory binaries [Aljabri et al. 2024, Shafin et al. 2023, Nissim et al. 2019, Sihwail et al. 2021, Dener et al. 2022]. This approach greatly enhances the ability to identify and mitigate threats in real time. Memory binary analysis provides deeper insight into the operations that occur during the execution of malicious code, capturing suspicious activity that may go undetected by traditional signature-based methods or static analysis [Dener et al. 2022].

Memory analysis highlights anomalous patterns and unexpected modifications in the system, which might indicate ransomware behavior [Carrier 2021]. The integration of machine learning algorithms has notably improved the accuracy of this approach, enabling more effective detection and the implementation of automatic responses against evolving threats [Naseer et al. 2021].

Additionally, the application of XAI in these techniques supports the security of automated decisions, ensuring a clear and informed understanding of ransomware response actions. Therefore, using memory binaries represents a crucial crossroads in the evolution of ransomware detection strategies, promoting a more robust and adaptive defense against increasingly sophisticated cybercriminals. [Nasser and Nassar 2023].

2.2. Overcoming Challenges with Machine Learning and Explainable AI

Traditional malware detection methods face substantial criticism due to several intrinsic limitations. Statistical and signature-based approaches, for example, usually identify malware only after the attack has already begun or after the damage has already been done. This delay in detection is problematic, especially in an enterprise environment where rapid response time is crucial to mitigating the impacts of a malware infection [Galli et al. 2024].

Another significant drawback of traditional methods is their lack of explainability, especially in security scenarios [Nasser and Nassar 2023]. This limitation can have profound implications since the decision-making process of a detection system may not be transparent to the user [Nasser and Nassar 2023, Scalas et al. 2021, Galli et al. 2024]. This lack of transparency can deteriorate confidence in the system and inhibit regulatory observation, which often requires transparency in automated operations and decisions. Traditional machine learning models are often considered a *black box*, making it difficult for security experts to fine-tune and improve defense systems.

According to [Galli et al. 2024], XAI is revolutionizing the Artificial Intelligence (AI)-based malware detection process by making AI processes transparent and understandable, thereby increasing trust and adoption of these technologies. XAI helps to explain how decisions are made, boosting the confidence of users and security experts and allowing for the continual refinement of detection models. This is especially useful in identifying and investigating threats, enabling a more effective response to security incidents. Additionally, XAI drives significant progress in research by identifying key characteristics of malicious behaviors. These insights are crucial for developing more effective and customized ransomware detectors that can adapt to the constantly changing strategies of attackers [Scalas et al. 2021].

It is important to note that the effectiveness of machine learning and XAI algorithms in ransomware detection depends on the feature's quality and relevance. By integrating more influential features, ML models can be improved to ensure their versatility across different datasets. Additionally, in the field of XAI, the meticulous selection of optimized and well-understood features can significantly assist in interpreting models, eventually leading to more precise and effective results.

2.3. The Importance of Feature Selection in the Context of Ransomware Detection

When selecting features, it's essential to conduct a thorough exploratory analysis of the dataset to understand how the characteristics behave. It's crucial to have a clear and concise understanding of the features and to identify the most relevant ones that are likely to impact machine learning models positively or negatively. This helps minimize the number of redundant and irrelevant features for the problem at hand [Dener et al. 2022, Aslan and Samet 2020].

According to [Malik et al. 2022], selecting the correct features is essential for improving the model's accuracy and reducing computational complexity. Furthermore, behavioral features are particularly advantageous as they enable models to adapt effectively to new ransomware variations. This is because behavioral features are more reliable and flexible than specific variant signatures, which can change rapidly to evade detection. Although automated tools like K-best can help with feature selection by identifying the most impactful features through statistical analysis, manually analyzing the dataset enables uncovering nuances and correlations that automated tools may miss.

2.4. Related Work

The landscape of ransomware detection is constantly evolving, highlighting the need for continuous research and improved detection methods. [Shafin et al. 2023] proposed a method for detecting different types of malware in embedded devices used in smart city applications. They aimed to identify recent malware, even if it was obfuscated. They used the CIC-Malmem-2022 dataset and divided the dataset into an 80-20% ratio to generate training and testing sets. The train set contained 46876 samples, while the test set had

11,720 samples. According to the authors, their proposed model showed advancements in detecting obfuscated malware and presented an innovative design to enhance identifying individual attack types.

In [Bruna Moralejo 2023], the focus was detecting and classifying malware attacks. With the increasing sophistication and diversity of malware, including ransomware, more traditional methods like signature-based approaches are needed. To address this, the authors used various ML algorithms, specifically supervised algorithms, and the e CIC-MalMem-2022 dataset, which consists of 58596 records for detection. The best results regarding malware detection were obtained with the Random Forest algorithm. Other algorithms, like the Decision Tree, produced similar results to Random Forest but showed an increase in computation time.

[Balasubramanian et al. 2023] conducted an in-depth analysis of memory data from the CIC-MalMem-2022 dataset to investigate threat data by employing various ML models. The study reveals that volatile memory presents significant potential in extracting valuable information and insights about the characteristics and behavior of malware, specifically ransomware. The findings of their research suggest that volatile memory analysis has emerged as a promising approach for effectively detecting and mitigating cyber threats.

[Mezina and Burget 2022], utilized the latest CIC-MalMem-2022 dataset, which is up-to-date with current technologies. This dataset consists of benign cases, malware instances, and details of the malware type and family, enabling advanced experimentation. Initially, traditional ML techniques were tested, followed by a dilated convolutional network proposal. The findings indicate that all techniques have a detection accuracy of 0.99. However, the random forest method is the most precise for detection, while the proposed neural network architecture is the best for classifying the malware family, with an accuracy of 0.83.

In [Abualhaj and Al-Khatib 2024], a detailed study employed a Decision Tree (DT) classifier to identify malware in-memory data using the CIC-MalMem-2022 dataset. This dataset was chosen for its comprehensive coverage of various malware families, although the study focused exclusively on Trojan Horses. The classifier's performance was rigorously compared against other ML models, such as Gradient-Boosted Trees, Naive Bayes (NB), Support Vector Machines (SVM), and K-Nearest Neighbors, using a variety of metrics, including accuracy, recall, and precision, ensuring the validity and reliability of our findings. The results showed that the DT achieves an accuracy of 99.96%, further underscoring the potential of machine learning in cybersecurity.

According to [Galli et al. 2024], malware aims to steal sensitive data and break the normal functioning of computer systems. Traditional methods of detecting malware, such as signature-based and statistical analysis, have limitations in terms of accuracy and efficiency. However, recent advancements in AI techniques, such as ML approaches, have shown promising results in detecting ransomware by analyzing its behavior. Despite their success, the lack of transparency in the decision-making process of these AI models has raised concerns about their reliability and interpretability [Galli et al. 2024].

To improve the interpretability of AI-based malware detection systems, XAI methodologies and tools have been developed. These tools provide clear explanations for the results generated by such systems, thereby allowing users to make informed decisions and take necessary actions to safeguard their systems from malware attacks [Galli et al. 2024, Smith Jr 2023].

3. Methodology

The systemic view of the methodology involved the following steps: (a) Selection and preprocessing of the dataset, (b) Selection of algorithms and materializing a baseline, (c) Feature selection from the preprocessed dataset, and (d) Exploration of feature importance using the DT algorithm and rationalization with SHAP.

3.1. Selection and Preprocessing of the Dataset

The CIC-MalMem-2022 dataset, developed by the Canadian Institute for Cybersecurity (CIC), has been used in the malware detection field due to its relevance and number of features involving malicious processes. It contains up-to-date data on memory-based malware, including ransomware, trojans, and spyware. It has a relevant size, which is conducive to creating robust models. The reliability of these models is evidenced by the consistent results presented in several studies, consequently reinforcing their credibility. Additionally, transparent and accessible documentation facilitates research replication, making it a solid choice for developing advanced cybersecurity detection models.

The CIC-MalMem-2022 dataset comprises 55 distinct features and 58,596 records with an equal distribution of benign and malicious samples that contain various aspects of malware behavior and system operations. The dataset organizes malware samples into 20 categories, each representing a specific malware family such as trojans, ransomware, and spyware. The dataset includes a debug mode in the memory dump process, which simulates real-world malware attacks. [Canadian Institute for Cybersecurity 2022]. Preprocessing the dataset was necessary to separate ransomware-related records from benign data, which is crucial. Despite the dataset's extensive collection of malware records, our specific emphasis is on the analysis and detection of ransomware.

3.2. Selection of Algorithms and Materializing a Baseline

The baseline is a starting point for evaluating new models and comparing the performance of more complex or recently developed models to a simple baseline. This analysis helps to measure the actual improvement gained from advanced techniques. Using a simple classifier as a baseline allows us to understand the problem's complexity. If a simple model performs well, it may indicate that the problem doesn't require more sophisticated techniques or that it can save resources and time. In this study context, it is essential to prevent *overfitting*, as complex models like deep neural networks have a higher risk of *overfitting*, especially on small or unvaried data sets. The models were materialized using the following algorithms: Artificial Neural Networks (ANN), Decision Tree (DT), Naive Bayes (NB), and Support Vector Machines (SVM).

3.3. Features Selection

After the initial classification (baseline), we manually examined the dataset to identify fundamental features that could significantly impact the machine learning models. The goal was to improve the dataset by removing less significant features, simplifying the number of features, and enhancing the model. We used descriptive statistics calculations to identify the most influential features, including mean, median, mode, standard deviation, and correlations. We also used scatter plots to understand the data better and highlight differences between each class. We created histograms to visually represent data distribution and identify outliers. We made two histograms: one for ransomware-infected data and another for benign data, providing insights into the differences between the two.

Optimization with Feature Importance and Explainable Artificial Intelligence (XAI): The primary objective of this step is to determine the ideal number of features without impacting the model's performance. Initially, we used the Feature Importance technique and a DT classifier to assess the importance of each feature in the machine-learning model's decision-making process. This technique helped identify the less relevant features to the model by displaying their importance in a tree-like structure.

Through the XAI, specifically Shapley Additive Explanations (SHAP) method, the analysis elucidated the Decision Tree's results, making the decision-making process more transparent and enhancing the model's reliability and security. By improving the model's transparency, we can identify the relevant features and weed out those not essential to the decision-making process. All the machine learning algorithms were trained after selecting the relevant features. The goal was to assess accuracy and recall, iterating to enhance the dataset and model performance.

3.4. Evaluation Models

When evaluating ML algorithms, it was essential to ensure the reliability and accuracy of the results. The learning curves [1] of these algorithms were created to help determine if the models were *overfitting*, *underfitting*, or fitting the data well. Two learning curve graphs were generated to assess the accuracy and loss of the models, observing their performance on the training and test data throughout the process. According to the learning curve observations, the scales remained small, indicating no significant decay in the loss graph, and all the algorithms maintained high accuracy throughout the training period.

We utilized the k-fold cross-validation (k = 5) method to assess the performance of our ML models. We separated training sessions on each fold, which allowed us to calculate the average accuracy across all the folds and determine whether the models maintained consistent performance across different test and training data.

A confusion matrix was generated for each model, along with the True Positive Rate (TPR = TP/(TP + FN)) and False Positive Rate (FPR = FP/(FP + TN)) for each class in the dataset, to see how many data points were classified correctly and how many were classified incorrectly. The Accuracy was calculated using the equation ACC = (TP + TN)/ (TP + TN + FP + FN).

3.5. Instrumentation and Model Parameters

We used the Pandas library was to read, transform, and clean the data. The Matplotlib, Plotly, and Seaborn libraries were also employed for data visualization and graphic creation. The Keras library was used for Artificial Neural Networks (ANN), while Scikit-learn was employed for the Naive Bayes (NB), Decision Tree (DT), and Support Vector

[1] A learning curve is a graph that exhibits the progress of a specific learning metric during machine learning model training.

Machines (SVM) algorithms. All these ML algorithms experienced a new test section with the same configuration settings. It's worth mentioning that the volume of resources in each section is affected by the optimization process.

We used the following parameters in our experiments. The *ANN* classifier used 100 epochs and five layers. The first layer is dense with 15 neurons and a Rectified Linear Unit (ReLU) activation function. The second layer uses a dropout of 0.1 to prevent overfitting. The third layer has 15 neurons with the ReLU function, the fourth layer has 8 neurons, and the last layer, which consists of the output, has the sigmoid function. *DT* and *NB* used the default parameters and *SVM* was configured with the linear kernel parameter.

4. Experimental Evaluation, Results and Analysis

In this section, we describe the experimental evaluation proposed in Section 3. In summary, we trained all algorithms using the original features from the dataset to estimate their performance. Essential features are then selected through data analysis to optimize the dataset and models. This is followed by further feature optimization using manual selection, Feature Importance with DT, and XAI.

4.1. Baseline Classification

The CIC-MalMem-2022 dataset contains 9,791 records related to ransomware, additionally broken down as follows: a) 2,000 records for the AKO ransomware, b) 1,988 for the Conti ransomware, c) 1,958 for MAZE, d) 1,717 for the Pysa ransomware, and e) 2,128 for the Shade ransomware. It's important to note that the dataset also includes 29,298 benign records. For this work, all the ransomware registers from the dataset were used. The data was separated into training and test sets using the holdout approach to address this experiment. 20% of the data was assigned to the test set, while 80% was reserved for training all the algorithms selected for this work.

In the baseline experiments, all generated models demonstrated satisfactory performance in detecting ransomware. According to our learning curve plots, as the number of samples increased, the models stabilized and consistently showed exemplary performance. The confusion matrices revealed strong model performance, with most predictions being correct. Results concerning Acurracy, K-Fold Accuracy Average, and Confusion Matrix metrics can be seen in the section 5.

4.2. Feature Selection

In this step, the primary objective is to reduce the number of features used in creating the models while maintaining their performance. This work achieves this through two methods: a) manually reducing the initial set of features and b) analyzing the decision tree algorithm using the feature importance technique and providing transparency in this method with SHAP (XAI).

4.2.1. Results of the Manual Feature Selection

According to [Sihwail et al. 2021], some of the most important characteristics for memory-based ransomware detection are related to the following groups: a) *Terminated*

Processes, b) *DLL's Records*, c) *Registry Modifications*, d) *Active Network Connections*, e) *Running Services*, f) *Code Injections and Hooking*, g) *Forensic Memory Analysis*.

Although manual feature selection can be practiced through analyses conducted in works such as [Sihwail et al. 2021], it is important to conduct a detailed study of the dataset, along with an analysis of dataset attributes, histograms, and scatter plots, in order to achieve a refined selection of features.

Figure 1 shows all the dataset features used in the baseline analyses, with the manually selected characteristics highlighted in blue.

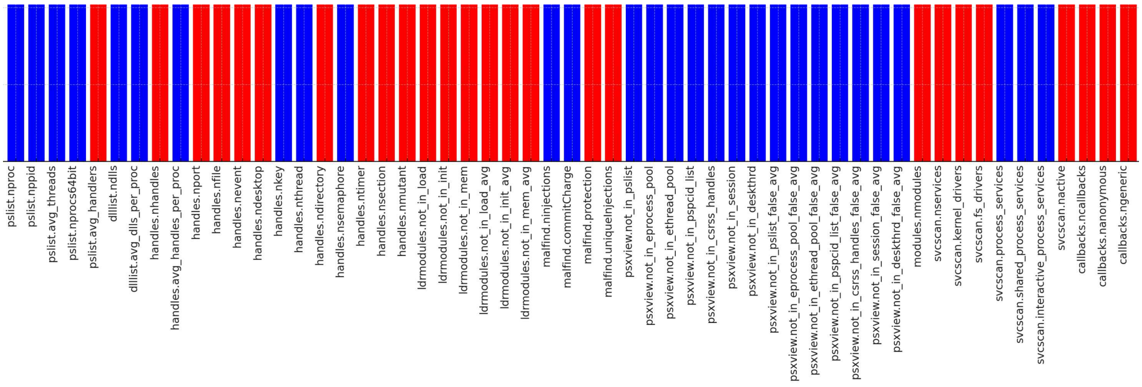


Figure 1. Baseline features and manual feature selection

After obtaining the results related to manual feature selection, the tests with the classifiers were conducted. Figure 2 illustrates the results obtained after manual feature selection.

When training and validating the (a) ANN with a smaller set of features, some fluctuations that indicated challenges in adjusting the validation data were observed. However, the curves consistently showed a decreasing trend and stability, suggesting the model performed well during training. The (b) DT also demonstrated good performance for detection, even with fewer features. Additionally, the learning curves of the (c) NB converged to a similar score, indicating that the model's performance stabilized with increasing training samples. Despite initial fluctuations, the (d) SVM tended to adjust as the number of samples increased. The confusion matrices suggested that all models performed well, with a high rate of true positives and low rates of false negatives and false positives. Both the graphs and matrices indicated that the models generalized well, showing good accuracy rates and stable learning curves after initial variations.

4.2.2. Results of the Feature Importance with Decision Tree and Shapley Additive Explanations (SHAP)

Feature importance was used to optimize the dataset by identifying and excluding less relevant features without affecting the model's performance. The DT was used for the method and was initially trained with the baseline data. The analysis of the feature importance comprehended metrics such as *Gini Impurity*, *Entropy*, and *Information Gain*, showing the weight of other features relevant to the model. Figure 3 shows the tree generated by the algorithm in (a) and a scatter plot in (b).

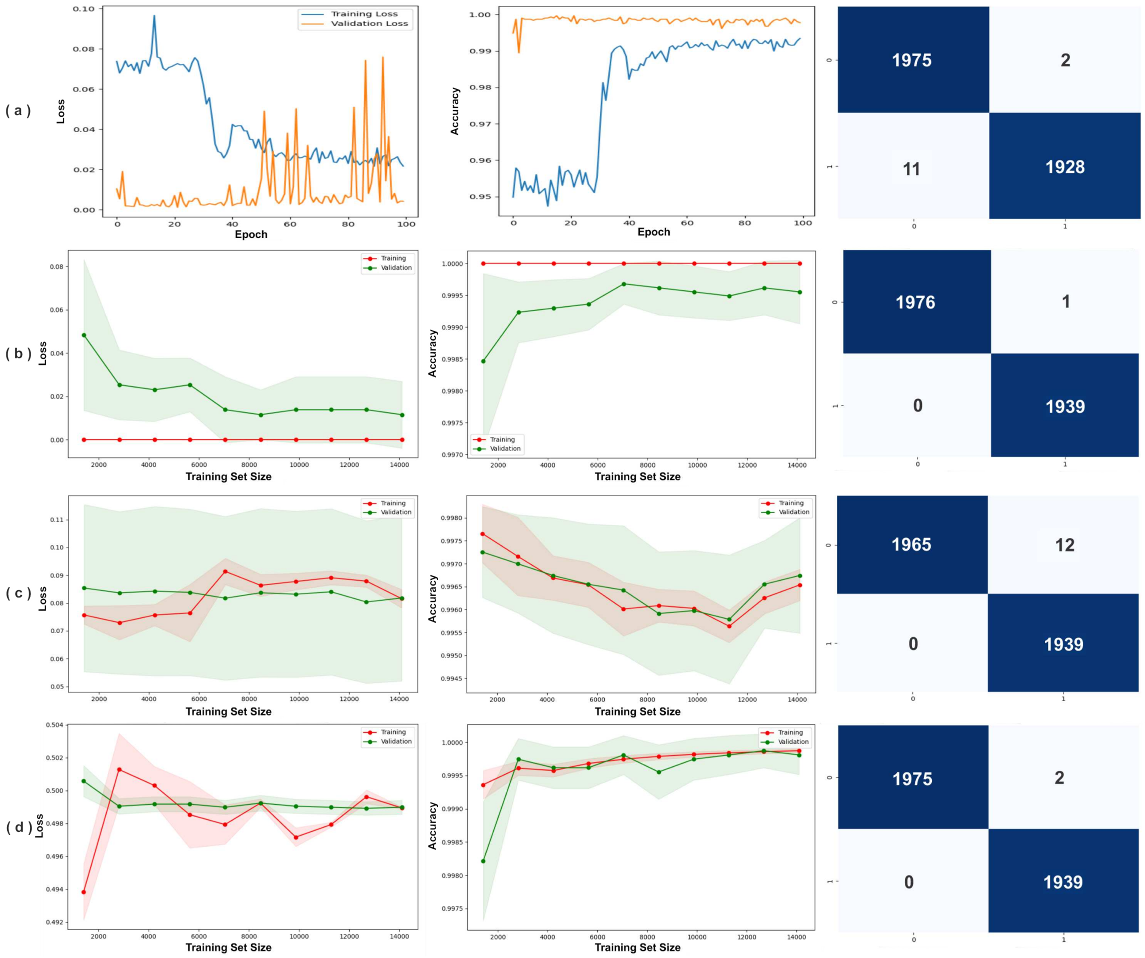


Figure 2. Models training and validation applied to the baseline after features selection manually.

According to Figure 3 (a), a rule was established for the most relevant feature, indicating that the *svcscan.shared_process_services* feature should be less than or equal to 116.5. This means that the model classifies the data as ransomware when the feature meets this condition. This feature refers to a metric captured during system analysis, specifically related to the total number of processes shared during the execution of a process in a system. An increase in the number of shared processes that do not typically use shared processes may indicate malicious activity.

In Figure 3 (b), it is evident that there is a specific threshold for data labeled as ransomware. Consequently, if the feature value exceeds 116.5, the data is considered benign; if it is 116.5 or lower, it is classified as ransomware.

Based on the analysis of the feature importance in the DT, illustrated by Figure 3, four more relevant features were selected from the 55 in the baseline; they are: (a) svscanṡhared_process_services, (b) svscan.process_services, (c) handles.nevent and (d)svscan.nservices.

Following the first step of optimization features, it was necessary to better understand their impact on the decision-making algorithms. The DT was selected because due to its simplicity. The SHAP was used to make the result of the DT more understand-

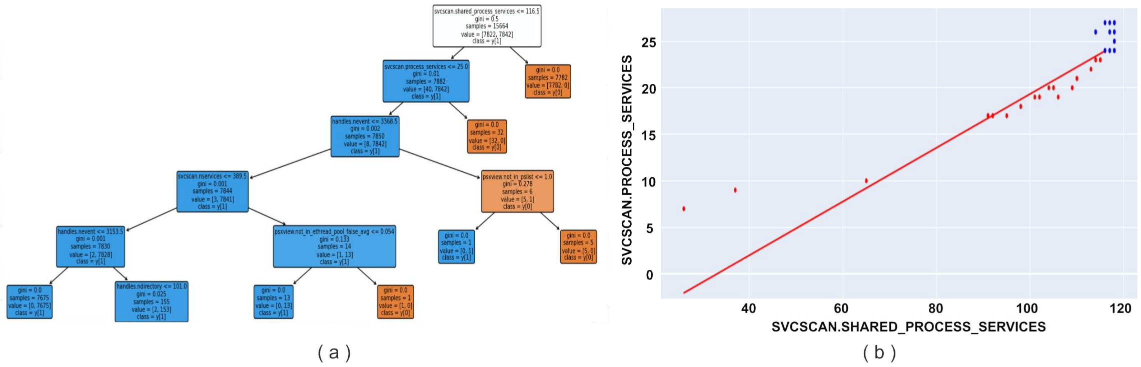


Figure 3. Generated Tree and Scatter Plot - Feature Importance.

able, improving the transparency of decision-making and increasing the confidence and security of the model.

For this experiment, 200 random samples of the training data from each trained algorithm were used, Figure 4 shows the most relevant features and how the model makes decisions. The higher the feature value, the stronger the shade of pink and the lower the shade of blue. In the dataset, the classes were divided into benign and ransomware data, where benign data is represented by 0 and ransomware data is represented by 1. When analyzing the model's impact output, it is notable that the most relevant features express negative values close to zero, which increases the possibility of these features impacting the model by increasing the probability of the model predicting class 0.

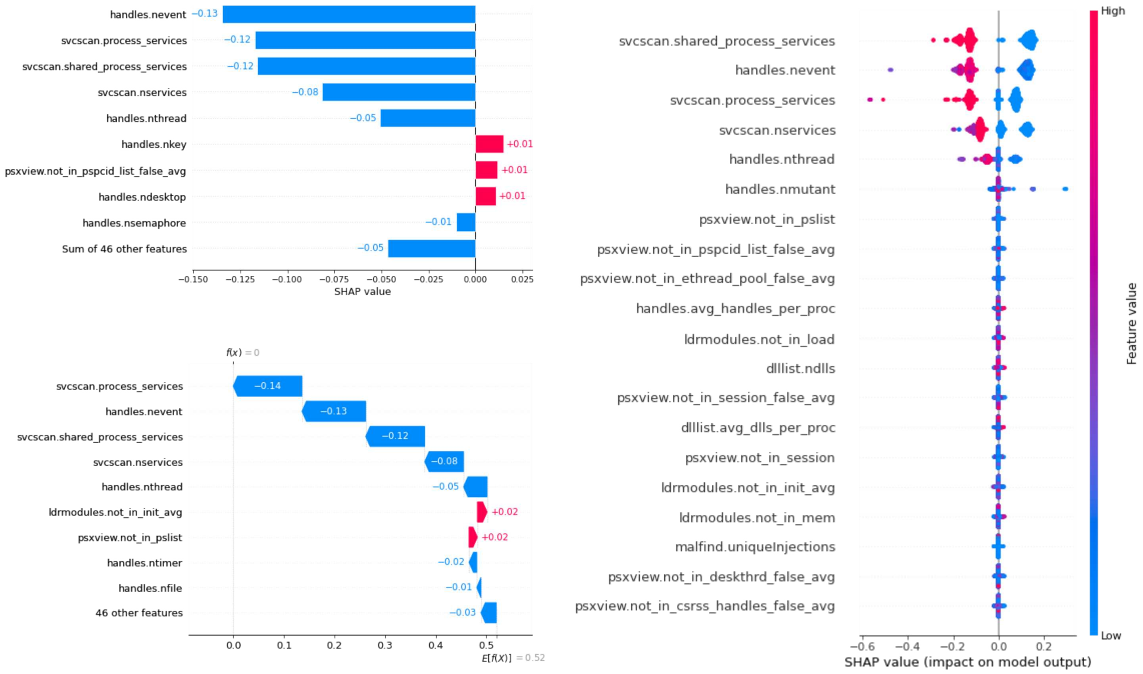


Figure 4. SHAP values using Decision Tree algorithm.

After analyzing Figure 4, it is evident that the features *svscan.shared_process_services*, *handles.nevent*, *svscan.process_services*, *svscan.nservices*, and *handles.nthread* have a positive impact on the model's prediction.

These features predominantly exhibit positive values on the horizontal axis, indicating their significant influence on the model's prediction. The distance of the points from the center is noticeable for all the mentioned features, suggesting a substantial impact on the model's prediction. Following a comprehensive analysis of the decision tree algorithm and SHAP, all models were retrained to evaluate the losses and accuracy of each model. Figure 5 shows these results.

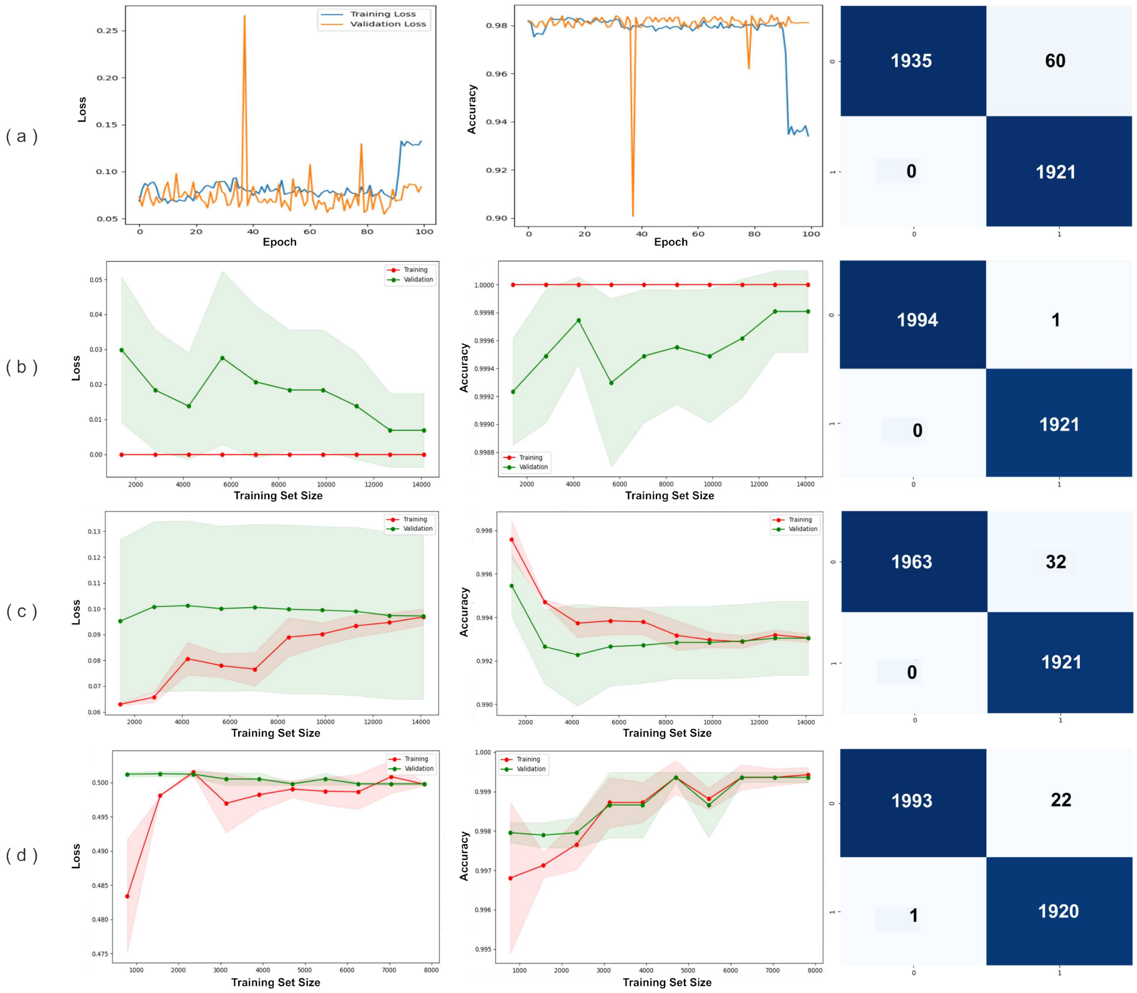
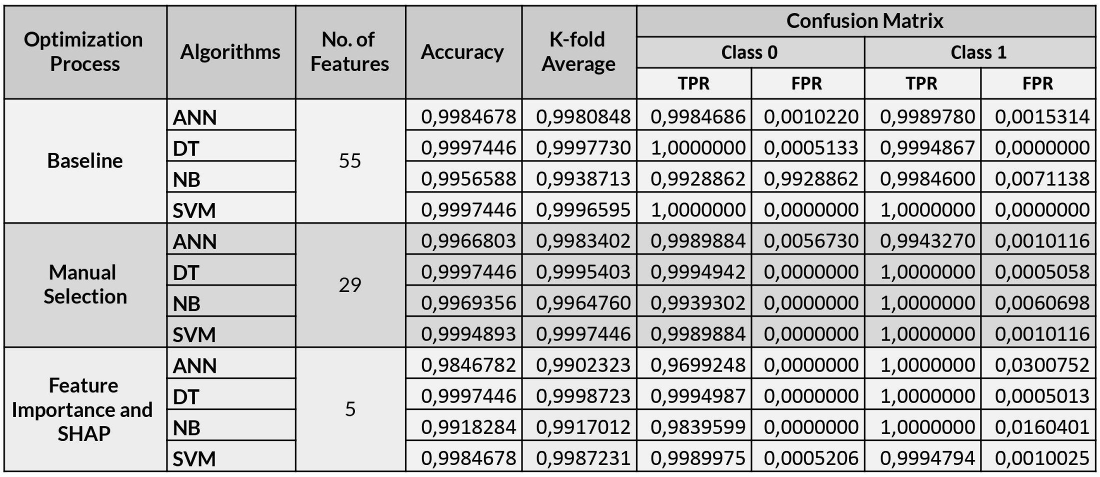


Figure 5. Models training and validation applied to the baseline after RFE and Decision Tree analysis.

According to Figure 5, it is evident that the (a) ANN exhibits occasional loss peaks during the validation stage. However, the model demonstrates learning and convergence as the number of epochs increases. The (b) DT displays stability in both validation and training accuracy with increased samples, indicating reliable model performance. Similarly, (c) NB shows improvement as the sample size grows. Additionally, despite encountering several convergence points, the (d) SVM effectively fits the data as the sample volume increases. The confusion matrices demonstrate robust performance for all algorithms, consistently achieving perfect class separation in all tests.

5. Analysis and Discussion of Results

Based on the experiments presented in this study, it can be seen that the number of features has been reduced. Originally, the dataset had 55 features, which has been reduced to 5. This implies that feature optimization techniques for ransomware, such as feature importance combined with SHAP, can greatly improve machine learning models. With these techniques, it was possible to select the most relevant features, reducing the size of the dataset and the complexity of models while maintaining accuracy. Figure 6 and Figure 7 illustrate the results obtained from the baseline experiments and the feature optimization.



Optimization Process	Algorithms	No. of Features	Accuracy	K-fold Average	Confusion Matrix			
					Class 0		Class 1	
					TPR	FPR	TPR	FPR
Baseline	ANN	55	0,9984678	0,9980848	0,9984686	0,0010220	0,9989780	0,0015314
	DT		0,9997446	0,9997730	1,0000000	0,0005133	0,9994867	0,0000000
	NB		0,9956588	0,9938713	0,9928862	0,9928862	0,9984600	0,0071138
	SVM		0,9997446	0,9996595	1,0000000	0,0000000	1,0000000	0,0000000
Manual Selection	ANN	29	0,9966803	0,9983402	0,9989884	0,0056730	0,9943270	0,0010116
	DT		0,9997446	0,9995403	0,9994942	0,0000000	1,0000000	0,0005058
	NB		0,9969356	0,9964760	0,9939302	0,0000000	1,0000000	0,0060698
	SVM		0,9994893	0,9997446	0,9989884	0,0000000	1,0000000	0,0010116
Feature Importance and SHAP	ANN	5	0,9846782	0,9902323	0,9699248	0,0000000	1,0000000	0,0300752
	DT		0,9997446	0,9998723	0,9994987	0,0000000	1,0000000	0,0005013
	NB		0,9918284	0,9917012	0,9839599	0,0000000	1,0000000	0,0160401
	SVM		0,9984678	0,9987231	0,9989975	0,0005206	0,9994794	0,0010025

Figure 6. Consolidated Results - Baseline and Feature Optimization

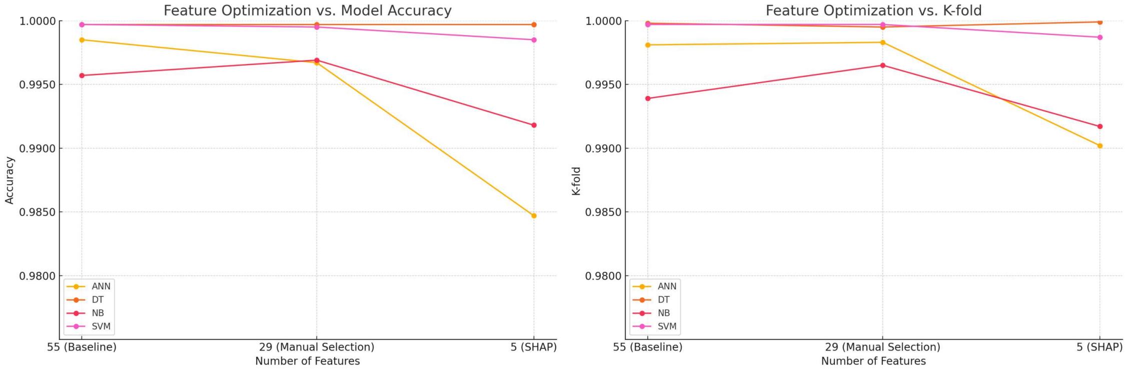


Figure 7. Consolidated Results - Feature Optimization and Model Accuracy

Looking at the performance of the accuracy and average accuracy in the K-fold in Figures 6 and 7, it was possible to see that all the algorithms obtained excellent results in different stages, as no biases were found since this evaluation method uses different data samples in order to check for possible biases in the models.

It is possible to see that the DT algorithm performed better than the others, even though all the algorithms expressed a high accuracy. The DT is an easy-to-implement algorithm, and its accuracy during training without registering major losses in the learning curve graph. Following these analyses, the NB algorithm showed a rise in the learning curve graph related to loss.

For the confusion matrix, all the algorithms performed well. The true positive rate, the proportion of records classified correctly, was high, while the false positive rate, which is the proportion of records classified incorrectly, was low. This shows that there were no problems with *overfitting* or *underfitting* and that the model is generalizing the two proposed classes very well.

There was also a slight decrease in the true positive rate during the Feature Importance and SHAP stage in class 0 in the Artificial Neural Networks (ANN) and Naive Bayes (NB) algorithms. However, despite this decrease, the true positive rates remained high and the false positive rate remained low, maintaining the quality and reliability of the classification models. Following this analysis, the DT and SVM algorithms maintained a high true positive rate and a low false positive rate, preserving the performance of the optimized models. Overall, all the models performed very well.

It is important to note that the feature importance method used in the DT algorithm is limited compared to the SHAP method. In this context, feature importance does not clearly explain how the features behave in the model and their relationship with the classes in each model. The results noted that the feature importance method applied to the decision tree algorithm selected four features as the most relevant, while the SHAP method showed five. The SHAP method validates the feature selection method, ensuring that the selected features are truly relevant to the models. Therefore, the SHAP method proved useful in two important situations: helping the feature selection process and providing a better understanding of the model's decisions.

6. Concluding Remarks

This work determined that integrating Machine Learning (ML) algorithms with Explainable Artificial Intelligence (XAI) techniques enables the identification of essential features in memory-based ransomware data and provides insight into the decision-making process of the models. By integrating XAI, this study improved detection models' accuracy, transparency, and reliability, facilitating the understanding of automatic decisions and supporting the security of threat responses. This advancement is essential for mitigating ransomware attacks.

The results of this study support the importance of choosing and optimizing features in ML models' performance. This allows for reducing the required features without compromising the model's effectiveness. It was also possible to observe which algorithms adapt better to a limited set of features. In this context, algorithms of greater complexity, such as Artificial Neural Networks (ANN), showed greater effort in training and slightly reduced accuracy. In contrast, algorithms such as the Decision Tree (DT), and Support Vector Machines (SVM) remained stable in training and maintained accuracy levels.

Looking ahead, we plan to replicate the presented method using other types of machine learning algorithms such as AdaBoost and Gradient Boosting Machines. In the context of detection, the feature optimization method used in this work can be applied to other classes of malware to observe the performance of models. Despite using the CIC-MalMem-2022, our goal is to evolve the set of features, comprising more features and a greater volume of memory records to support new experiments. Another area for future work would be using the insights we have gained about the most relevant memory features and designing a system to identify ransomware activity within the operating system.

References

Abualhaj, M. M. and Al-Khatib, S. N. (2024). Using decision tree classifier to detect trojan horse based on memory data. *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, 22(2):393–400.

Aljabri, M., Alhaidari, F., Albuainain, A., Alrashidi, S., Alansari, J., Alqahtani, W., and Alshaya, J. (2024). Ransomware detection based on machine learning using memory features. *Egyptian Informatics Journal*, 25:100445.

Alraizza, A. and Algarni, A. (2023). Ransomware detection using machine learning: A survey. *Big Data and Cognitive Computing*, 7(3):143.

Aslan, Ö. A. and Samet, R. (2020). A comprehensive review on malware detection approaches. *IEEE access*, 8:6249–6271.

Balasubramanian, K. M., Vasudevan, S. V., Thangavel, S. K., Kumar, G., Srinivasan, K., Tibrewal, A., and Vajipayajula, S. (2023). Obfuscated malware detection using machine learning models. In *2023 14th International Conference on Computing Communication and Networking Technologies (ICCCNT)*, pages 1–8. IEEE.

Beaman, C., Barkworth, A., Akande, T. D., Hakak, S., and Khan, M. K. (2021). Ransomware: Recent advances, analysis, challenges and future research directions. *Computers & security*, 111:102490.

Bensaoud, A., Kalita, J., and Bensaoud, M. (2024). A survey of malware detection using deep learning. *Machine Learning With Applications*, 16:100546.

Bruna Moralejo, L. (2023). Machine learning for malware detection and classification. Master's thesis, Universitat Politècnica de Catalunya.

Canadian Institute for Cybersecurity (2022). CIC-MalMem-2022 Dataset. `https://www.unb.ca/cic/datasets/malmem-2022.html`. Accessed: 10-01-2024.

Carrier, T. (2021). Detecting obfuscated malware using memory feature engineering.

Dener, M., Ok, G., and Orman, A. (2022). Malware detection using memory analysis data in big data environment. *Applied Sciences*, 12(17):8604.

Galli, A., La Gatta, V., Moscato, V., Postiglione, M., and Sperlì, G. (2024). Explainability in ai-based behavioral malware detection systems. *Computers & Security*, 141:103842.

Herrera-Silva, J. A. and Hernández-Álvarez, M. (2023). Dynamic feature dataset for ransomware detection using machine learning algorithms. *Sensors*, 23(3):1053.

Hornetsecurity (2022). Ransomware attacks survey 2022. Accessed: 05-31-2024.

Liu, H. and Motoda, H. (2007). *Computational methods of feature selection*. CRC press.

Malik, S., Shanmugam, B., Kannorpatti, K., and Azam, S. (2022). Critical feature selection for machine learning approaches to detect ransomware. *International Journal of Computing and Digital Systems*, 11(1):1168–1176.

Mezina, A. and Burget, R. (2022). Obfuscated malware detection using dilated convolutional network. In *2022 14th international congress on ultra modern telecommunications and control systems and workshops (ICUMT)*, pages 110–115. IEEE.

Naseer, M., Rusdi, J. F., Shanono, N. M., Salam, S., Muslim, Z. B., Abu, N. A., and Abadi, I. (2021). Malware detection: issues and challenges. In *Journal of Physics: Conference Series*, volume 1807, page 012011. IOP Publishing.

Nasser, Y. and Nassar, M. (2023). Toward hardware-assisted malware detection utilizing explainable machine learning: A survey. *IEEE Access*, 11:131273–131288.

Nissim, N., Lahav, O., Cohen, A., Elovici, Y., and Rokach, L. (2019). Volatile memory analysis using the minhash method for efficient and secured detection of malware in private cloud. *Computers & Security*, 87:101590.

Othman, H., AlHija, M. A., and Alsharaiah, M. A. (2024). Toward enhancing malware detection using practical swarm optimization in honeypot. *International Journal of Intelligent Engineering & Systems*, 17(1).

Routray, S., Prusti, D., and Rath, S. K. (2023). Ransomware attack detection by applying machine learning techniques. In *Machine Intelligence Techniques for Data Analysis and Signal Processing: Proceedings of the 4th International Conference MISP 2022, Volume 1*, pages 765–776. Springer.

Scalas, M. et al. (2021). Malware analysis and detection with explainable machine learning.

Shafin, S. S., Karmakar, G., and Mareels, I. (2023). Obfuscated memory malware detection in resource-constrained iot devices for smart city applications. *Sensors*, 23(11):5348.

Sihwail, R., Omar, K., and Arifin, K. A. Z. (2021). An effective memory analysis for malware detection and classification. *Computers, Materials & Continua*, 67(2).

Smith Jr, D. Q. (2023). *Exploring Machine Learning for Malware Detection With Feature Selection, Explainable AI, and Generative Adversarial Networks*. PhD thesis, North Carolina Agricultural and Technical State University.