Received: November 17, 2024. Revised: December 10, 2024.

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

Effective Obfuscated Malware Detection Leveraging Cutting-edge Machine and Deep Learning Approaches

Reyadh Hazim Mahdi[1, 2]* Hafedh Trabelsi[1]

[1]*Department of Electrical Engineering, National Engineering School of Sfax (ENIS), Sfax University, Tunisia*
[2]*Department of Computer Science, College of Science, Mustansiriyah University, Baghdad, Iraq*
* Corresponding author's Email: reyadh.hazim@uomustansiriyah.edu.iq

Abstract: Detecting obfuscated malware concentrates on specifying any malicious software (malware) that has been intentionally disguised or concealed to avoid conventional detection systems. Typically, obfuscation methods like runtime obfuscation, packing, and code encryption have been utilized by malware authors to change the appearance of malware while preserving its malicious functionality. As these methods have evolved, conventional detection systems have become less effective, requiring further sophisticated solutions like machine and deep learning approaches. Machine and deep learning approaches work on obfuscated malware detection via auto-learning features and patterns from massive datasets. Additionally, these approaches can specify formerly unseen malware by analyzing various features. In this paper, the Independent Component Analysis (ICA) is first utilized for isolating relevant features from obfuscated data to be prepared for binary classification. For further data analysis, essential feature selection, and a deeper comprehension of the relationships inside the dataset, the Pearson correlation coefficient is applied to the dataset to be prepared for multiclass classification. This dual scheme improves the feature extraction process depending on the classification type, enhancing the system's versatility and performance. Then, a proposed One-Dimensional Convolutional Neural Network (1D-CNN) is leveraged for extracting efficient features from memory traces, and providing an accurate system for classifying obfuscated memory malware. The combination of ICA or Pearson correlation with 1D-CNN in a unified system offers a scalable and inclusive settlement for binary or multiclass classification and contributes to the progress of malware classification systems. Besides the proposed 1D-CNN approach, two machine learning approaches are trained and assessed on the CIC-MalMem-2022 dataset, and the attained results depicted that the performance of the proposed 1D-CNN approach was superior, with accuracies of 99% and 88% for binary and multiclass classification, respectively.

Keywords: Obfuscated memory malware, Independent component analysis, Correlation technique, Proposed 1D-CNN, Random forest, Extreme gradient boosting.

1. Introduction

In the past few years, there has been a substantial increase in more sophisticated malware attacks, with the first half of 2022 seeing nearly 2.8 billion malware instances. These exceedingly effective malware attacks are utilized by malicious actors that can deliver diverse payloads and achieve various goals. With 560,000 new malware instances detected daily and a 62% year-over-year increase in malware variants, malware is becoming one of the most serious threats to global cybersecurity [1, 2].

Obfuscated memory malware represents a kind of malicious software geared towards operating in the memory of computers, preventing exposure via conventional security systems over diverse obfuscation methods [3]. These methods conceal the actual behavior of malware while it is running in memory without leaving any traces on the disk and usually remain hidden until it is executed [4]. Since manual detection systems' time consumption and complexity are extremely high, diverse machine and deep learning approaches were exploited that could auto-produce wise insights from data [5]. Machine learning approaches take a set of features that are

capable of being considered a large sample size for comparing differences. Additionally, the key factor in determining the appropriate machine learning approach to use is the input features which come in different formats. While some machine-learning approaches concentrate on speed, others concentrate on precision and accuracy. Thus choosing approaches that match the goal and type of input has a huge influence on the system results [6, 7]. Deep learning has substituted conventional classification approaches as an important issue in detecting malware [8]. Deep learning approaches, specifically Convolutional Neural Networks (CNNs) are capable of simulating the brain of humans for learning, thus experts have implemented these approaches in almost every field [9, 10].

The latest advances in machine and deep learning have been accompanied by the emergence of various approaches for detecting memory-hogging malware. These approaches have revealed encouraging results (between 93% and 99%), specifically in the scenarios of binary classification. Such results highlight the ability of machine learning to distinguish between two classes (malware or benign). However, implementing these approaches to the scenarios of multiclass classification reveals a substantial issue. When the goal is to distinguish between malware and benign classes, these approaches are effective, but their performance will diminish when the goal is to distinguish among different families [11]. This performance disparity can be attributed to the increased complexity of malware families, creating a more challenging scenario for classification approaches originally designed to make binary decisions. The complex behavioral patterns and delicate differences in traits that distinguish one family of malware from another necessitate more advanced analytical approaches, capable of dealing with the complex and overlapping features that characterize different types of malware [12].

Compared with the other deep learning approaches, 1D-CNN is perfectly suitable for constrained resources and real-time applications because it significantly reduces the cost of computation by simplifying 1D feature processing. It is capable of recognizing patterns irrespective of their temporal position in the input. The low parameter space of 1D-CNN enhances generalization, diminishes overfitting issues, and facilitates the visual representation of the learned filters due to their interpretability [13].

In this paper, the obfuscated malware detection issue has been tackled which represents malicious software intended to evade detection by conventional tools of cybersecurity. Besides the proposed One-Dimensional Convolutional Neural Network (1D-CNN), sophisticated machine learning approaches specifically Random Forest and Extreme Gradient Boosting are leveraged for analyzing and interpreting covertly residing memory dumps. This obfuscated malware classification system allows many essential contributions:

1. Utilizing ICA to minimize the memory data dimensionality, segregate unrelated signals from assorted data, and de-obfuscate hidden (independent) components.
2. Utilizing one of the most common correlation techniques to specify the linear relationships between diverse features in the memory malware dataset.
3. Proposing 1D-CNNs to auto-learn features and specify essential patterns directly from memory traces dataset.
4. Combining ICA with machine and deep learning approaches allows the system to handle better the difficulties posed by malware obfuscation methods, leading to enhance the binary classification performance.
5. Combining the Pearson correlation with machine and deep learning approaches makes the system capable of specifying significant feature relationships throughout multi-classes, enhancing the selection of essential features, capturing delicate inconsistencies between malware families, and improving performance of a multiclass classification system.
6. Implementing diverse assessment indicators to exhibit the performance of the approaches, and the results depicts that the proposed 1D-CNN approach is superior when contrasted to the machine learning approaches in classifying obfuscated malware.

Besides this section, the structure of this paper begins with the recently presented related works in Section Two. In Section Three, the proposed malware classification system is exhibited in detail. The utilized dataset, experiments, and attained results are presented in Section Four. Eventually, the conclusion and recommendation for new works are given in Section Five.

2. Related works

Many research studies have addressed the topic of obfuscated memory malware due to its increasing harmful influence, leading to the need to detect diverse malware effectively, however, most of these studies were focused on malware classification as either malicious or benign [14].

Mezina and Burget [15] presented the utilization of a dilated CNN (DCNN) approach for binary and multiclass malware classification. Expanding the receptive domain of convolutional layers improved the extraction of essential features from memory dumps, making the system capable of capturing wider context and long-range dependencies. This presented approach was implemented using the latest CIC-MalMem-2022 dataset, and the achieved accuracy results for binary and multiclass classification were 99% and 83%, respectively. The results showed that the system performance was decreased when dealing with more complex classes. While this system stands out in binary classification, it should concentrate on enhancing its multiclass abilities and decreasing the computational load to be more flexible for the real world scenarios.

Hasan and Dhakal [16], investigated various machine-learning approaches (Multi-Layer Perceptron, Extreme Gradient Boosting, k-Nearest Neighbors, and Random Forest) for binary and multiclass classification of obfuscated memory malware. The CIC-MaIMem-2022 dataset was utilized to assess the proposed classification framework, and all of these implemented approaches achieved promising results of 99% for malware binary classification. The main disadvantage of this system was the computational cost accompanying memory analysis.

Shafin et al. [17], presented a binary and multiclass malware classification system using the CIC-Malmem-2022 dataset. In this hybrid system, the feature-learning abilities of CNNs were combined with the temporal modeling benefit of bidirectional long short-term memory. This presented system surpassed conventional machine learning approaches and achieved 99% and 84% accuracy in detecting binary and family attacks. However, despite the optimization of hybrid architecture for resource-constrained IoT environments, the memory and computation cost of deep learning approaches still pose potential difficulties.

Roy et al. [18], proposed a hybrid classification system to detect obfuscated malware. In this system, before delving into the classification process, sixteen essential features were selected out of fifty-five features utilizing the Pearson correlation technique. In the first phase of classification, several machine-learning approaches (Extremely Randomized Trees, Random Forests, and Extreme Gradient Boosting) were implemented as base learners, and the second phase involved the utilization of a deep-learning classifier (of dense layers) as a meta-learner. The CIC-MalMem-2022 dataset was utilized for assessing the system performance, and the superior findings were achieved with accuracies of 99% and 85% for binary classification and multiclass classification, respectively. Despite the utilization of feature reduction, the hybrid nature of the system with multi-layers (involving machine and deep learning approaches) could increase the computational requirements in comparison with simpler systems.

Maniriho et al. [19], utilized deep auto-encoders to provide optimal automatic feature extraction, with various ensemble learning approaches (Binary Logistic Regression, Stochastic Gradient Descent, Support Vector Machine, and Cat Boosting) to perform the binary classification. In this system, the MemMal-d2024 dataset was enhanced by introducing two timestamp features, bringing the total to 58 features. This enhanced dataset enables the malware classification system to be assessed without exhibiting temporal bias. The experiments disclosed that the proposed system achieved high performance when specifying unseen malware with an accuracy of 98%. However, the system's training and deployment need computational resources (for instance, deep auto-encoders and various ensemble learning approaches), which could be a drawback for resource-constrained systems.

kumar et al. [20], employed diverse machine-learning approaches for auto-distinguishing malicious and benign software via learning patterns from the samples of the CIC-MalMem-2022 dataset. The correlation technique was utilized in this system to extract and select the essential features, and eighteen features were acquired out of fifty-seven features. Among the implemented approaches, Random Forest and Extreme Gradient Boosting performed the best in malware classification, attaining high accuracy results (99%) and low rates of false positives. This system concentrated on conventional machine learning approaches for malware binary classification and ignored the exploration of deep learning approaches that can provide promising results, especially in multiclass classification.

Sihwail et al. [21] used different optimization algorithms with machine learning to select essential features and enhance the efficiency and accuracy of the malware detection system. In this system, among the optimization algorithms used, the enhanced whale optimization algorithm (EWOA) was superior in efficiently reducing the number of non-essential features, and the K-nearest neighbors were implemented for the binary malware classification task. Despite the enhancements, this system acknowledged that high-dimensional spaces still pose a challenge for EWOA, especially in preventing local

optima and preserving the balance between exploitation and exploration. The system performance was assessed using the CIC-MalMem-2022 dataset, and the classification accuracy obtained was 99%. While Abualhaj et al. [22], utilized the Firefly Algorithm (FA) as an optimization algorithm for selecting the essential features and implemented the Random Forest for binary and multiclass malware classifications. The CIC-MaIMem-2022 dataset is also used in assessing the system performance, and the achieved accuracy results were 99% and 87% for binary and multiclass classification, respectively. The presented feature selection mechanism, although innovative, provides further computational costs that may not be justified in comparison with its performance gains.

Table 1. Comparison between relevant malware classification systems

Ref.	Methodology and Utilized Dataset	Classification	Accuracy Results
[15]	DCNN approach using CIC-MalMem-2022 dataset	Binary classification	99%
		Multiclass classification	83%
[16]	Several Machine learning approaches using CIC-MalMem-2022 dataset	Binary classification	99%
[17]	CNN and bidirectional long short-term memory using CIC-MalMem-2022 dataset	Binary classification	99%
		Multiclass classification	84%
[18]	Pearson correlation and Machine learning approaches using CIC-MalMem-2022 dataset	Binary classification	99%
		Multiclass classification	85%
[19]	Deep auto-encoders and several machine learning approaches using MemMal-d2024 dataset	Binary classification	98%
[20]	Random Forest and Extreme Gradient Boosting using CIC-MalMem-2022 dataset	Binary classification	99%
[21]	EWOA and K-nearest neighbors using CIC-MalMem-2022 dataset	Binary classification	99%
[22]	FA and Random Forest using CIC-MalMem-2022 dataset	Binary classification	99%
		Multiclass classification	87%

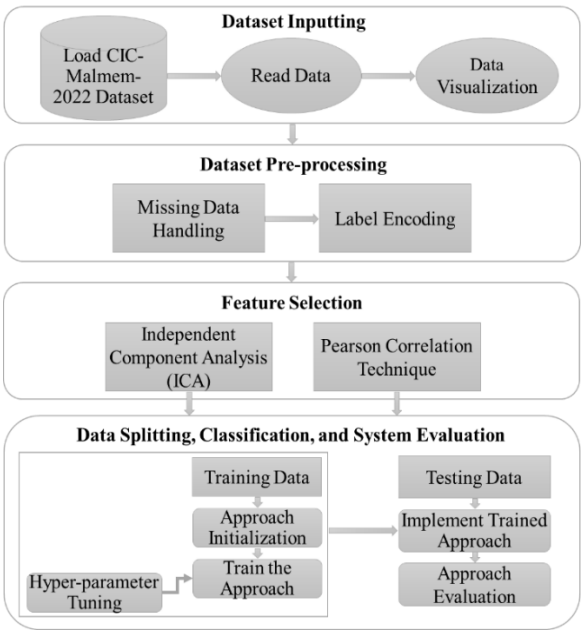

Figure. 1 Proposed system Pipeline

Table 1 illustrates a summary of the relevant previous works.

According to the prevalent studies analysis in the field, it is noticeable that inadequate attention has been provided to designing and assessing accurate multiclass classification systems for obfuscated memory malware that can function inside limited system requirements. Therefore, besides binary classification, the proposed system contemplates an accurate multiclass classification to detect malware classes and families with reduced computational cost.

3. Proposed system

The proposed system constitutes advanced machine and deep learning approaches to improve malware analysis and detection. These approaches classify memory activity into two classes (benign and malware) or multi-classes (benign, Trojan, Spyware, and Ransomware), as detailed in Fig. 1.

This system incorporates cutting-edge machine and deep learning approaches, comprising Random Forest, Extreme Gradient Boosting, and proposed 1D-CNN, each adapted to handle the issue of obfuscated memory malware binary and multiclass classification. In binary classification, by utilizing a combination of ICA with the machine and deep learning approaches, the system can concentrate on the significant and relevant features for better classification performance. In multiclass classification, the Pearson correlation is combined with the machine and deep learning approaches to specify significant feature relationships throughout multi-classes and acquire delicate inconsistencies between malware families.

3.1 Dataset input and pre-processing

In this proposed system, the dataset input and pre-processing stages are carefully constructed to guarantee that the dataset is ready for the applied machine and deep learning approaches. Here, the CIC-Malmem-2022 dataset is employed to train and test these approaches to classify memory activities into benign and malware or multiple malware classes. It was chosen to be utilized in this system because evaluating new models on outdated datasets does not accurately reflect real-world effectiveness. This dataset is first cleansed of infinite and missing values to guarantee data integrity.

The dataset involves various malware families that are represented as strings. Machine and deep learning approaches inherently need numerical input, therefore, certain dataset features, specifically the 'Category' column should be encoded due to their categorical nature. Thus, the categories; Malware and Benign could be encoded as 1 and 0, respectively, and the categories; Trojan, Spyware, Ransomware, and Benign could be encoded as 3, 2, 1, and 0, respectively.

3.2 Feature selection

The computational process involved in analyzing real-world patterns is often jeopardized by the high dimensionality of data, a common difficulty in many machine and deep learning-based applications. Therefore, in this proposed system, Independent Component Analysis (ICA) and the Pearson Correlation Technique are exploited to eliminate unnecessary features, hence, minimizing the complexity of computation and improving the system performance.

For binary classification, within the limits of the CIC-Malmem-2022 dataset, which involves features taken from memory traces, ICA can be applied for extracting and segregating essential statistically independent features. Before starting to apply this technique, the dataset should be pre-processed by subtracting the mean from each feature and de-correlating the variables using Singular Value Decomposition (SVD), in other words, the dataset should be centered and whitened. The linear transformation (L) should then be calculated so that the components are made as statistically independent of each other as possible. These pre-processes can be specified using Eqs. (1)-(3):

$$I_{center} = I - \mu_I \tag{1}$$

$$I_{white} = D_{eigen}{}^{-1/2} M_{eigen}{}^{\top} I_{center} \tag{2}$$

$$C = L\, I_{white} \tag{3}$$

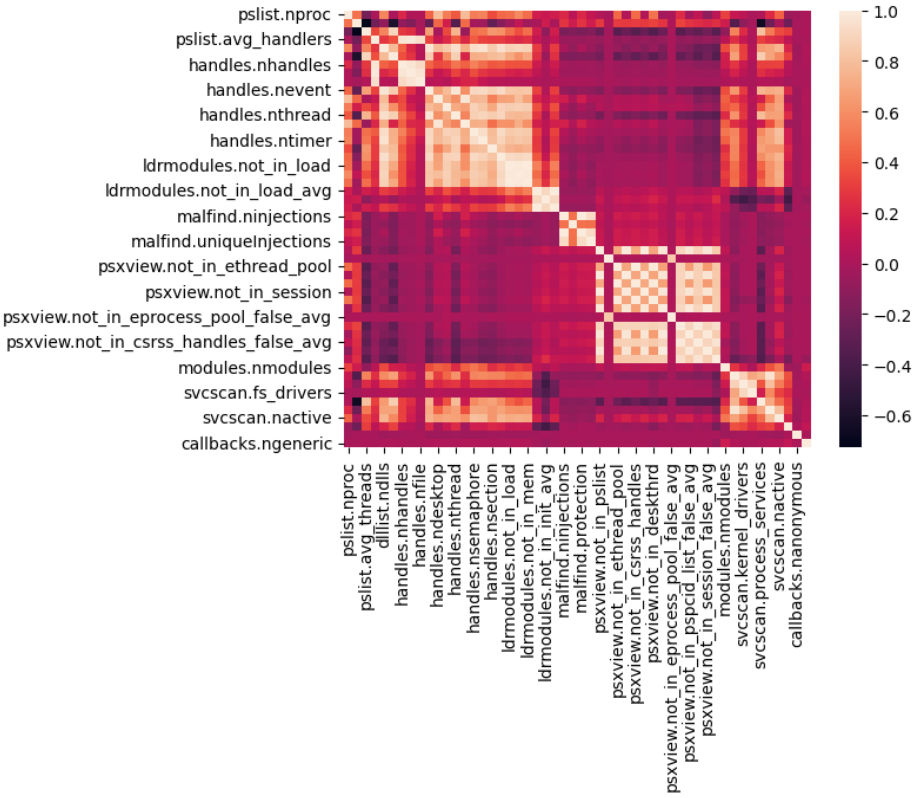

Figure. 2 Heatmap of features correlation using the Pearson correlation technique

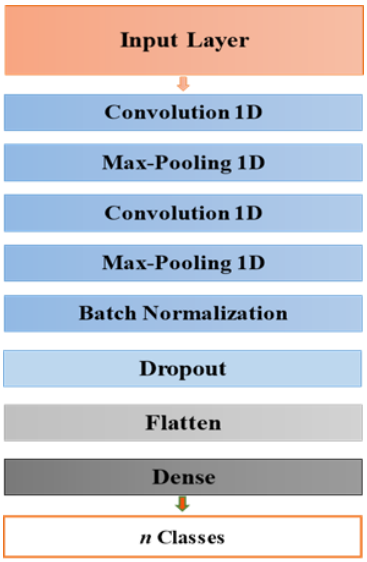

Figure. 3 The proposed 1D-CNN architecture

Where I_{center} denotes the input features, μ_I denotes the dataset mean vector, D_{eigen} denotes the diagonal matrix of eigenvalues, M_{eigen} denotes the eigenvectors matrix, and C denotes independent components.

After that, the Fast ICA method is applied to obtain the independent components to be utilized for dimensionality reduction, as specified in Eq. (4):

$$C^{(t+1)} = \mathrm{E}[I\, tanh(C^{(t)}I)] - \mathrm{E}[tanh'(C^{(t)}I)]C^{(t)} \quad (4)$$

After applying ICA on the CIC-Malmem-2022 dataset, it has the potential to obtain independent components corresponding with normal, suspicious, and hidden behaviors of memory operations. These components are fed into deep and machine-learning approaches to enhance binary classification accuracy. Here, only five essential features were selected out of the existing features (54 features).

For multi-class classification, possessing excessive features can make distinguishing between classes more difficult, especially if there are duplicate or irrelevant features, therefore, to make a malware detection system more effective, a Pearson correlation technique can be exploited to reduce the dimensionality of the dataset by specifying highly correlated features (redundant features) that could not add much new information. By specifying relationships between features, this technique assists in comprehending how specific memory regions or measures in the dataset are associated with each other. It is possible to remove those strongly correlated features to simplify the system without losing predictive ability.

The coefficient (C) of the Pearson correlation between N and M variables can be computed using Eq. (5):

$$C = \frac{\sum(N_i-\bar{N})(M_i-\bar{M})}{\sqrt{\sum(N_i-\bar{N})^2\sum(M_i-\bar{M})^2}} \quad (5)$$

Where N_i and M_i denote the individual values of N and M, and $\overline{N}$ and $\overline{M}$ denote the means of the two variables. The numerator and denominator in this formula denote the covariance and standard deviations of N and M variables.

A heatmap of the correlation values between features in the dataset used is depicted in Fig. 2, with various colors indicating the correlations' direction and strength. Here, only ten essential features out of the existing features (54 features) were selected.

3.3 Implementing deep and machine learning approaches

In this work, a deep learning approach is designed and trained to learn and extract essential representations of selected features and classify malicious files dependent on generated features. The proposed 1D-CNN approach fundamentally involves several convolutions and fully connected layers, the vital features are extracted by slipping one filter on the input data in the convolutions and flattening it in the fully connected layers. Fig. 3 depicts the proposed approach's architecture, encompassing two 1D convolutions followed by 1D Max-pooling, batch normalization, one flatten, and dense (with Sigmoid in binary classification or Softmax for multiclass classification). The convolution-1D can be mathematically formulated using Eq. (6):

$$I_y^h = f\left(\sum_{x=1}^{M} I_y^{h-1} \times H_{xy}^h + b_y^h\right) \quad (6)$$

Where a feature map can be constructed with an input (I), a convolution filter (H), a function of activation (f), and a bias (b).

To raise the computation speed and minimize the length of parameters, the convolutions are downsampling in 1D max-pooling layers. Batch normalization is utilized to normalize the proposed approach and to evade the issue of overfitting. Besides the downsampling and batch normalization layers, the dropout can also be utilized to handle the overfitting. The output of the layers above is then flattened in the fully connected layer to form a single vector. While the final layer output represents probabilities for each label.

Table 2. 1D-CNN layers' names, shapes, and relevant parameters

Layers	Shapes Output	No. of Parameter	Shapes Output	No. of Parameter
Convolution 1D	(None, 4, 32)	96	(None, 4, 32)	96
Max-Pooling 1D	(None, 2, 32)	0	(None, 2, 32)	0
Convolution 1D	(None, 2, 64)	12352	(None, 2, 64)	12352
Max-Pooling 1D	(None, 1, 64)	0	(None, 1, 64)	0
Batch Normalization	(None, 1, 64)	256	(None, 1, 64)	256
Dropout	(None, 1, 64)	0	(None, 1, 64)	0
Fully Connected	(None, 64)	0	(None, 64)	0
Dense	(None, 1)	65	(None, 4)	260
	Parameters; Total: 12,769 Trainable: 12,641 Non-trainable: 128		Parameters; Total: 12,964 Trainable: 12,836 Non-trainable: 128	

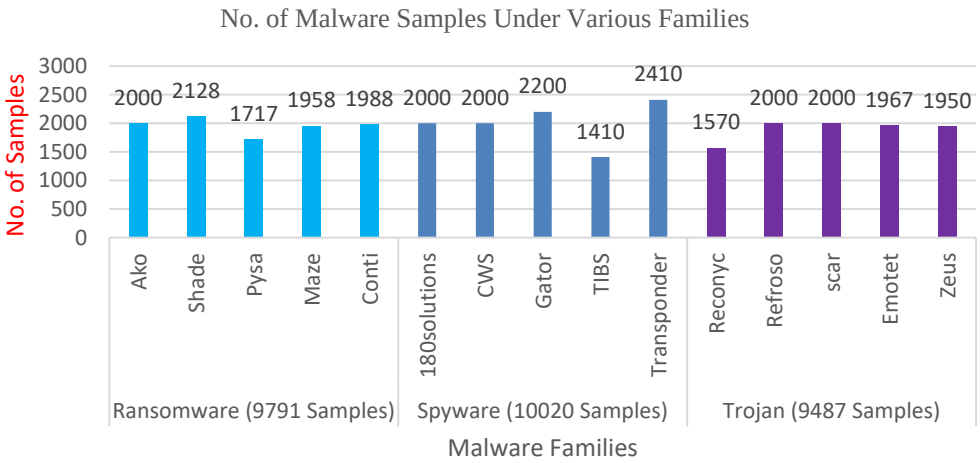

Figure. 4 Malware samples distribution of the CIC-MalMem-2022 dataset

The rectified linear unit (ReLU) function is utilized for network activation at hidden layers by making output values vary from zero to infinity. The final layer of the 1D-CNN approach is activated utilizing the sigmoid function in the binary classification scenario and the softmax function in the multiclass classification scenario. Table 2 summarizes the layers of the proposed 1D-CNN in detail.

In this proposed system, the first implemented machine learning approach is the Random Forest, which is employed in memory dump classification and distinguishing between benign and diverse malware families. This approach is revered for its accuracy and robustness, specifically in addressing complicated datasets holding many features. Random Forest merges multi-decision trees, minimizing the overfitting issue while seizing an extensive range of data attributes. Every tree contributes a specific vision, and the cooperative decision-making from all trees presents an extensive and balanced classification. The effectiveness and adaptability of Random Forest make it a significant approach to be compared with the proposed 1D-CNN, enhancing the malware classification capabilities.

The second implemented approach is Extreme Gradient Boosting. In essence, Extreme Gradient Boosting relies on iterative gradient boosting. It utilizes a multi-threaded mechanism that effectively utilizes the machine's CPU cores, which leads to improved system performance and speed. It is noteworthy that Extreme Gradient Boosting is superior to other machine-learning approaches for resolving regression, classification, and other predictive issues, especially when utilizing structured or tabular datasets.

4. Experimental results and discussion

This section exhibits the utilized dataset, assessments based on various metrics, and experimental outcomes using two experiments (for binary and multiclass classification). The experimental outcomes of carrying out the proposed obfuscated memory malware classification system

utilizing various machine and deep learning approaches Random Forest, Extreme Gradient Boosting, and Proposed 1D-CNN) will be presented and discussed in detail. The results for these approaches are exhibited and compared using the CIC-MalMem-2022 dataset, to depict the proposed classification system efficiency. The training process for these approaches is the same.

The utilized dataset represents the cybersecurity dataset named "CIC-Malmem-2022" [23], which concentrates on detecting malicious memory activities. This dataset was generated in a dominated environment by the Canadian Institute for Cybersecurity to assist researchers in obfuscated malware detection that operates mainly in memory and is difficult to detect via traditional antivirus software because it leaves no traces on the disk.

The CIC-Malmem-2022 dataset was created by executing both malicious and benign programs, with memory dumps collected at various stages to acquire the evolution of malicious activity over time. It involves 58,596 samples (29,298 benign and 29,298 malware), each with 56 features. There are different families of malware such as Ransomware, Spyware, and Trojans that each depict 15 distinct common memory-based attacks, and the sample distribution of these families is depicted in Fig. 4.

The efficacy of the proposed malware classification system is assessed by adopting several metrics extensively utilized in the malware classification field. These metrics encompass Recall (R_e), Precision (P_r), and F1-score ($F1_s$) with their macro and weighted average, and Accuracy (A_c) metric as well. Careful examination of these metrics can provide valuable insights into the strengths and counterfeits of the proposed approaches. R_e refers to the system's ability to properly specify actual malware instances, which is particularly significant when malware utilizes methods like memory obfuscation to avoid detection, and P_r refers to the measure of malware instances that are classified as malware. When there is an uneven distribution for classes, $F1_s$ is used, which refers to the harmonic mean of recall and precision [24]. These metrics are formulated using Eqs. (7)-(9) [25]:

$$R_e = \frac{T_{Positives}}{T_{Positives}+F_{Negatives}} \tag{7}$$

$$P_r = \frac{T_{Positives}}{T_{Positives}+F_{Positives}} \tag{8}$$

$$F1_s = 2 \times \frac{R_e \times P_r}{R_e + P_r} \tag{9}$$

In the task of obfuscated memory malware classification, the fundamental metric in measuring overall classification propriety which is A_c refers to the ratio of the properly classified instances to the whole instances, is formulated using Eq. (10) [26]:

$$A_c = \frac{T_{Positives}+T_{Negatives}}{Total_{Instances}} \tag{10}$$

Table 3. Results for deep and machine learning approaches in a binary classification scenario

Classes	**Random Forest**			**Extreme Gradient Boosting**			**Proposed 1D-CNN**			**Support**
	P_r	R_e	$F1_s$	P_r	R_e	$F1_s$	P_r	R_e	$F1_s$	
0	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99	8749
1	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99	8830
A_c		0.99			0.99			0.99		17579
Macro Avg.	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99	17579
Weight Avg.	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99	17579

Table 4. Results for deep and machine learning approaches in multiclass classification scenario

Classes	**Random Forest**			**Extreme Gradient Boosting**			**Proposed 1D-CNN**			**Support**
	P_r	R_e	$F1_s$	P_r	R_e	$F1_s$	P_r	R_e	$F1_s$	
Benign	1.00	1.00	1.00	1.00	0.99	0.99	0.99	1.00	0.99	7324
Ransomware	0.73	0.72	0.73	0.73	0.73	0.73	0.75	0.73	0.74	2448
Spyware	0.79	0.79	0.79	0.78	0.79	0.79	0.79	0.81	0.80	2505
Trojan	0.72	0.72	0.72	0.73	0.73	0.73	0.75	0.75	0.75	2372
A_c		0.87			0.87			0.88		14649
Macro Avg.	0.81	0.81	0.81	0.81	0.81	0.81	0.82	0.82	0.82	14649
Weight Avg.	0.87	0.87	0.87	0.87	0.87	0.87	0.88	0.88	0.88	14649

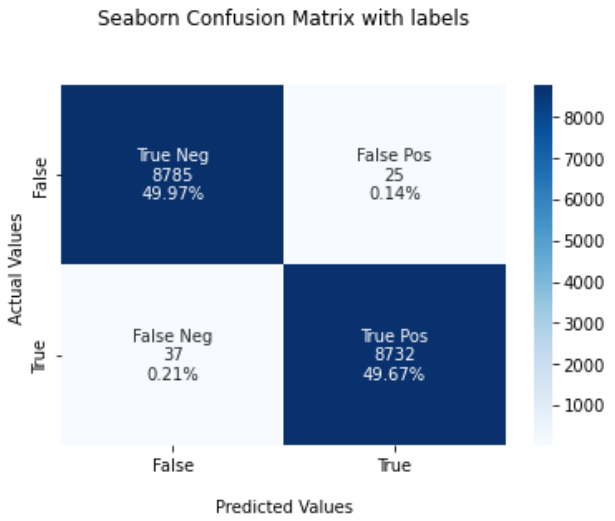

Figure. 5 Confusion matrix for Random Forest in binary classification scenario

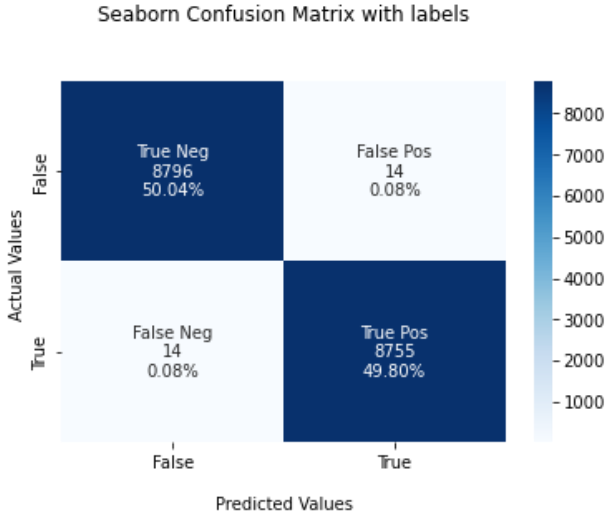

Figure. 6 Confusion matrix for Extreme Gradient Boosting in binary classification scenario

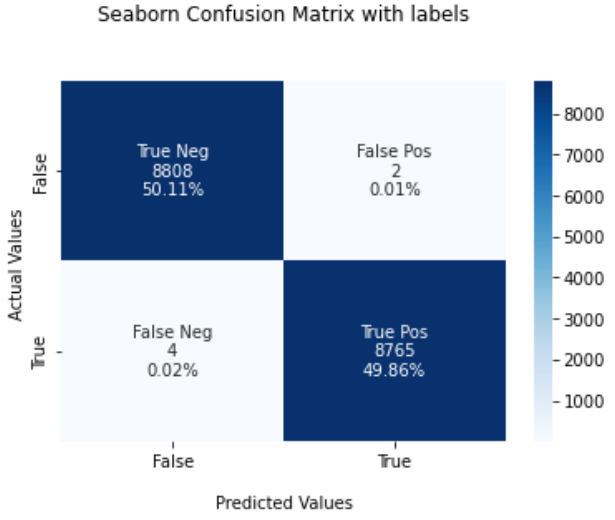

Figure. 7 Confusion matrix for Proposed 1D-CNN in binary classification scenario

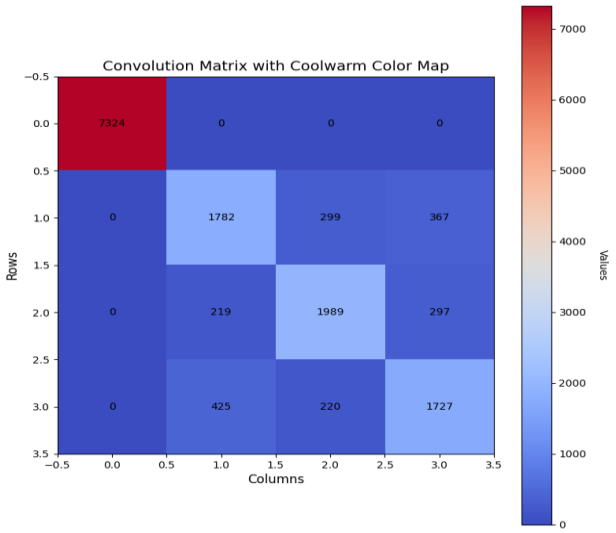

Figure. 8 Confusion matrix for Random Forest in multiclass classification scenario

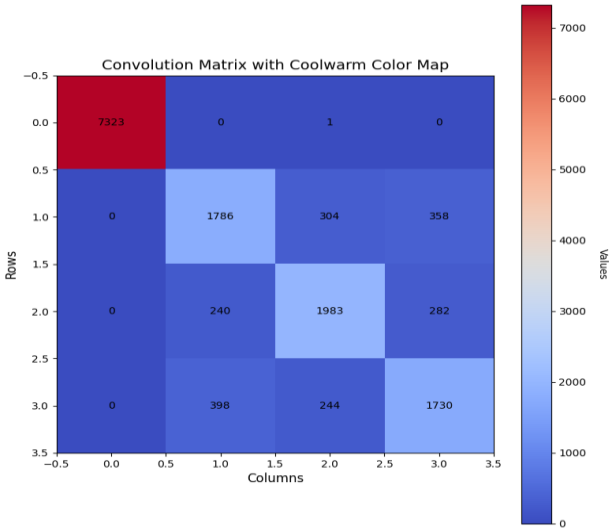

Figure. 9 Confusion matrix for Extreme Gradient Boosting in multiclass classification scenario

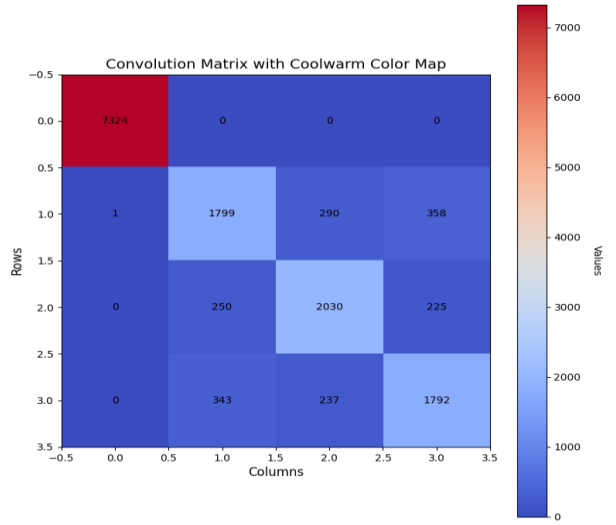

Figure. 10 Confusion matrix for Proposed 1D-CNN in multiclass classification scenario

Figure. 11 Training and validation accuracy/loss for the proposed 1D-CNN in binary classification scenario

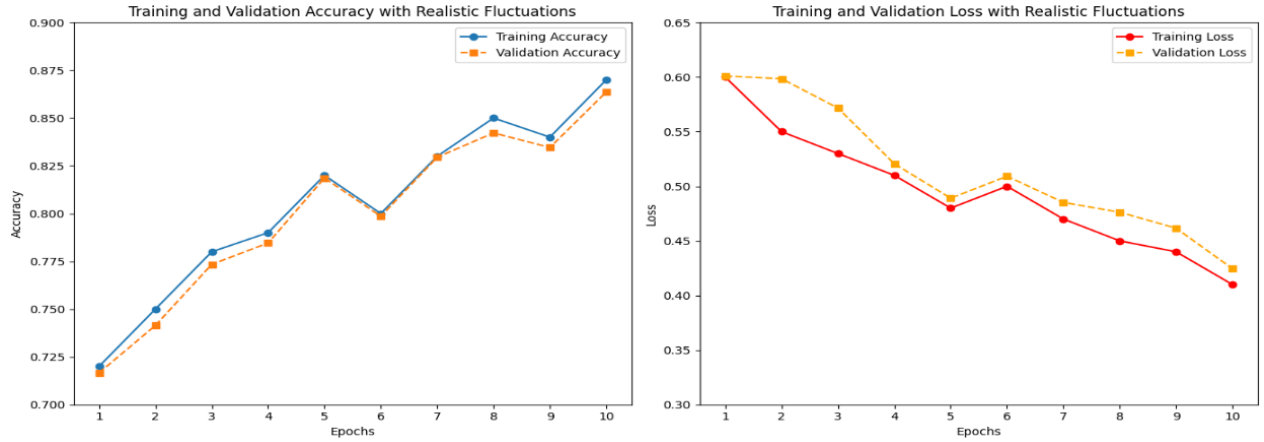

Figure. 12 Training and validation accuracy/loss for the proposed 1D-CNN in multiclass classification scenario

Table 5. A comparison between the proposed multiclass classification system and relevant works

Ref.	Feature Selection	Detection Approaches	Attained Results			
			P_r	R_e	$F1_s$	A_c
[17]	None	CNN and bidirectional long short-term memory	85%	85%	84%	84%
[18]	Pearson Correlation	Stacked Machine learning approaches	85%	85%	84%	85%
[22]	FA	Random Forest	87%	87%	87%	87%
Proposed multiclass system	Pearson Correlation	Random Forest	87%	87%	87%	87%
		Extreme Gradient Boosting	87%	87%	87%	87%
		1D- CNN	88%	88%	88%	88%

Where $T_{Positives}$ denotes the count of properly classified obfuscated malware samples, $F_{Positives}$ denotes the count of benign samples improperly classified as obfuscated malware, $T_{Negatives}$ denotes the count of properly classified benign samples, and $F_{Negatives}$ denotes the count of obfuscated malware samples improperly classified as benign software.

The classification results of the implemented deep and machine learning approaches assessed on the CIC-MalMem-2022 dataset are depicted in Tables 3 and 4. The proposed 1D-CNN reports the highest results among the others, and the accuracy achieved is 99% and 88%, for binary and multiclass classification, respectively.

In the binary and multiclass classification scenarios, among all the implemented approaches, the Proposed 1D-CNN achieved the highest results with accuracies of 99% and 88%, respectively.

Figs. 5 to 7 illustrate the confusion matrices using the CIC-MalMem-2022 dataset in binary scenario, and Figs. 8 to 10 in multiclass classification scenario.

It is observed that even when the approaches implemented in this proposed system do not match the right class, this wrong class can be classified to a closer one. This indicates that the proposed system is robust and effective.

In Figs. 5 to 7, the values in the diagonal represent the true positive and negative for the two classes, showing how well the approaches properly specify instances of every class, specifically, the approach's performance gets better for each class when diagonal values get closer to the support instances. While in Fig. 8, Fig. 9, and Fig. 10, the diagonal values denote the true positives for the four classes, showing how well the approaches properly specify instances of every class. The best performance was achieved by proposed 1D-CNN approach. Fig. 11 and Fig. 12 show proposed 1D-CNN approach's accuracy and computational loss.

The proposed classification system can provide better results in binary classification because the dataset is balanced, where half of the dataset used is old and contemporary obfuscation-based attacks and exactly the other half contains benign memory dumps. Meanwhile, multiclass classification achieved acceptable and superior results compared to other relevant works. An in-depth comparison between the proposed multiclass classification system and systems in [17, 18, 22] using the CIC-MalMem-2022 dataset is depicted in Table 5.

5. Conclusion

Cutting-edge machine and deep learning approaches in the cybersecurity field were presented in this work, concentrating on the binary and multiclass classification for the obfuscated memory malware. For binary classification, the utilization of ICA with the machine and deep learning approaches made the system more generalized over various obfuscated malware types, as the ICA concentrated on significant signals and revealed independent components, and the machine and deep learning approaches were capable of learning intricate and non-linear patterns in these signals. And with this scenario (for machine and deep learning approaches), the achieved accuracy, recall, precision, and F1-score were 99% using the CIC-MalMem-2022 dataset. Furthermore, for multi-class classification, combining the Pearson correlation technique for core feature selection with the proposed 1D-CNN for feature extraction and classification enhances the accuracy obtained on different malware families. By segregating the independent obfuscation traces, the proposed 1D-CNN could concentrate more on the actual malware signal, hence providing promising results and enhancing the classification accuracy, recall, precision, and F1-score to 88% using the CIC-MalMem-2022 dataset.

However, future works can explore more developed Recurrent Neural Networks (RNNs) and their effectiveness can be assessed on various malware families for enhancing complicated pattern learning substantially and capturing long-term dependencies.

Conflicts of Interest

The authors declare no conflict of interest.

Author Contributions

Conceptualization, R. H. Mahdi and H. Trabelsi; methodology, R. H. Mahdi; software, R. H. Mahdi; validation, R. H. Mahdi; formal analysis, R. H. Mahdi; investigation, R. H. Mahdi; resources, R. H. Mahdi; data curation, R. H. Mahdi; writing—original draft preparation, R. H. Mahdi; writing—review and editing, R. H. Mahdi and H. Trabelsi; visualization, H. Trabelsi; supervision, H. Trabelsi; project administration, R. H. Mahdi and H. Trabelsi.

References

[1] M. M. Alani, A. Mashatan, and A. Miri, "XMal: A lightweight memory-based explainable obfuscated-malware detector", *Computers & Security*, Vol. 133, 2023.

[2] R. H. Mahdi, and H. Trabelsi, "Detection of Malware by Using YARA Rules", In: *Proc. of 2024 21st International Multi-Conf. on Systems, Signals & Devices (SSD)*, Erbil, Iraq, pp. 1-8, 2024.

[3] F. Biondi, T. Given-Wilson, A. Legay, C. Puodzius, and J. Quilbeuf, "Tutorial: An Overview of Malware Detection and Evasion Techniques", *Lecture Notes in Computer Science, Leveraging Applications of Formal Methods, Verification and Validation, Modeling*, Springer, Cham, Vol. 11244, pp. 565-586, 2018.

[4] M. R. Naeem, M. Khan, A. M. Abdullah, F. Noor, M. I. Khan, M. A. Khan, I. Ullah, and S. Room, "A Malware Detection Scheme via Smart Memory Forensics for Windows Devices", *Mobile Information Systems*, pp. 1-16, 2022.

[5] T. Carrier, P. Victor, A. Tekeoglu, and A. H. Lashkari, "Detecting Obfuscated Malware using Memory Feature Engineering", In: *Proc. of International Conf. on Information Systems Security and Privacy*, pp. 177-188, 2022.

[6] A. J. Abdullah, T. M. Hasan, and J. Waleed, "An Expanded Vision of Breast Cancer Diagnosis Approaches Based on Machine Learning Techniques", In: *Proc. of 2019 International Engineering Conf. (IEC)*, Erbil, Iraq, pp. 177-181, 2019.

[7] X. Zhou, Y. Leng, A. K. Dutta, N. Juraev, A. Alkhayyat, and Y. Elmasry, "Machine learning analysis/optimization of auxetic performance of a polymeric meta-hybrid structure of re-entrant and meta-trichiral", *European Journal of Mechanics - A/Solids*, Vol. 109, 2025.

[8] H. Wang, B. Cui, Q. Yuan, R. Shi, and M. Huang, "A review of deep learning based malware detection techniques", *Neurocomputing*, Vol. 598, 2024.

[9] J. Waleed, S. Albawi, H. Q. Flayyih, and A. Alkhayyat, "An Effective and Accurate CNN Model for Detecting Tomato Leaves Diseases", In: *Proc. of 2021 4th International Iraqi Conf. on Engineering Technology and Their Applications (IICETA)*, Najaf, Iraq, pp. 33-37, 2021.

[10] R. J. Kolaib, and J. Waleed, "Crime Activity Detection in Surveillance Videos Based on Developed Deep Learning Approach", *Diyala Journal of Engineering Sciences*, Vol. 17, No. 3, pp. 98-114, 2024.

[11] Md. A. Hossain, and Md. S. Islam, "Enhanced detection of obfuscated malware in memory dumps: a machine learning approach for advanced cybersecurity", *Cybersecurity*, Vol. 7, No. 16, 2024.

[12] N. Z. Gorment, A. Selamat, and O. Krejcar, "Obfuscated Malware Detection: Impacts on Detection Methods", *Recent Challenges in Intelligent Information and Database Systems. ACIIDS 2023. Communications in Computer and Information Science*, Vol. 1863, pp. 55-66, 2023.

[13] H. Parineh, M. Sarvi, and S. A. Bagloee, "Detecting emergency vehicles With 1D-CNN using fourier processed audio signals", *Measurement*, Vol. 223, 2023.

[14] J.-Y. Kim, and S.-B. Cho, "Obfuscated Malware Detection Using Deep Generative Model based on Global/Local Features", *Computers & Security*, Vol. 112, 2022.

[15] A. Mezina, and R. Burget, "Obfuscated malware detection using dilated convolutional network", In: *Proc. of 2022 14th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT)*, Valencia, Spain, pp. 110-115, 2022.

[16] S. M. R. Hasan, and A. Dhakal, "Obfuscated Malware Detection: Investigating Real-World Scenarios Through Memory Analysis", In: *Proc. of 2023 IEEE International Conf. on Telecommunications and Photonics (ICTP)*, Dhaka, Bangladesh, pp. 1-5, 2023.

[17] S. S. Shafin, G. Karmakar, and I. Mareels, "Obfuscated Memory Malware Detection in Resource-Constrained IoT Devices for Smart City Applications", *Sensors*, Vol. 23, No. 11, 2023.

[18] K. S. Roy, T. Ahmed, P. B. Udas, Md. E. Karim, and S. Majumdar, "MalHyStack: A hybrid stacked ensemble learning framework with feature engineering schemes for obfuscated malware analysis", *Intelligent Systems with Applications*, Vol. 20, 2023.

[19] P. Maniriho, A. N. Mahmood, and M. J. M. Chowdhury, "MeMalDet: A memory analysis-based malware detection framework using deep autoencoders and stacked ensemble under temporal evaluations", Computers & Security, Vol. 142, 2024.

[20] S. kumar, Shersingh, S. kumar, and K. Verma, "Malware Classification Using Machine Learning Models", *Procedia Computer Science*, Vol. 235, pp. 1419-1428, 2024.

[21] R. Sihwail, M. Al Ghamri, and D. Ibrahim, "An Enhanced Model of Whale Optimization Algorithm and K-nearest Neighbors for Malware Detection", *International Journal of Intelligent Engineering and Systems*, Vol.17, No.3, pp. 606-621, 2024, doi: 10.22266/ijies2024.0630.47.

[22] M. M. Abualhaj, M. Al-Zyoud, A. Alsaaidah, A. Abu-Shareha, and S. Al-Khatib, "Enhancing Malware Detection through Self-Union Feature Selection Using Firefly Algorithm with Random Forest Classification", *International Journal of Intelligent Engineering and Systems*, Vol.17, No.4, pp. 376-389, 2024, doi: 10.22266/ijies2024.0831.29.

[23] T. Carrier, P. Victor, A. Tekeoglu, and A. H. Lashkari, "Detecting Obfuscated Malware using Memory Feature Engineering", In: *Proc. of the 8th International Conf. on Information Systems Security and Privacy (ICISSP 2022)*, pp. 177-188, 2022.

[24] C. S. Banumathi, and A. B. Rajendra, "Hybridized Least Absolute Shrinkage Selection Operator and Cross-Validation Algorithm for Classification of Malware", *International Journal of Intelligent Engineering and Systems*, Vol.16, No.5, pp. 329-338, 2023, doi: 10.22266/ijies2023.1031.28.

[25] J. Waleed, A. T. Azar, S. Albawi, W. K. Al-Azzawi, I. K. Ibraheem, A. Alkhayyat, I. A. Hameed, and N. A. Kamal, "An Effective Deep Learning Model to Discriminate Coronavirus Disease From Typical Pneumonia", *International Journal of Service Science, Management, Engineering, and Technology (IJSSMET)*, Vol. 13, No. 1, pp. 585-600, 2022.

[26] M. M. Alani, A. Mashatan, and A. Miri, "XMal: A lightweight memory-based explainable obfuscated-malware detector", *Computers & Security*, Vol. 133, 2023.