applied sciences

MDPI

Article

Binary and Multi-Class Malware Threads Classification

Ismail Taha Ahmed [1], Norziana Jamil [2,*], Marina Md. Din [2] and Baraa Tareq Hammad [1]

[1] College of Computer Sciences and Information Technology, University of Anbar, Anbar 55431, Iraq
[2] Institute of Informatics and Computing in Energy, Universiti Tenaga Nasional, Jalan Ikram-Uniten, Kajang 43000, Selangor, Malaysia
* Correspondence: norziana@uniten.edu.my

Abstract: The security of a computer system can be harmed by specific applications, such as malware. Malware comprises unwanted, dangerous enemies that aim to compromise the security and generate significant loss. Consequently, Malware Detection (MD) and Malware Classification (MC) has emerged as a key issue for the cybersecurity society. MD only involves locating malware without determining what kind of malware it is, but MC comprises assigning a class of malware to a particular sample. Recently, a few techniques for analyzing malware quickly have been put out. However, there remain numerous difficulties, such as the low classification accuracy of samples from related malware families, the computational complexity, and consumption of resources. These difficulties make detecting and classifying malware very challenging. Therefore, in this paper, we proposed an efficient malware detection and classification technique that combines Segmentation-based Fractal Texture Analysis (SFTA) and Gaussian Discriminant Analysis (GDA). The outcomes of the experiment demonstrate that the SFTA-GDA produces a high classification rate. There are three main steps involved in our malware analysis, namely: (i) malware conversion; (ii) feature extraction; and (iii) classification. We initially convert the RGB malware images into grayscale malware images for effective malware analysis. The SFTA and Gabor features are then extracted from gray-scale images in the feature extraction step. Finally, the classification is carried out by GDA and Naive Bayes (NB). The proposed method is evaluated on a common MaleVis dataset. The proposed SFTA-GDA is the effective choice since it produces the highest accuracy rate across all families of the MaleVis Database. Experimental findings indicate that the accuracy rate was 98%, which is higher than the overall accuracy from the existing state-of-the-art methods.

Keywords: malware detection; malware classification; SFTA; Gabor; GDA; energy security

Citation: Ahmed, I.T.; Jamil, N.; Din, M.M.; Hammad, B.T. Binary and Multi-Class Malware Threads Classification. *Appl. Sci.* **2022**, *12*, 12528. https://doi.org/10.3390/app122412528

Academic Editors: David Megías and Arcangelo Castiglione

Received: 29 September 2022
Accepted: 22 November 2022
Published: 7 December 2022

1. Introduction

The Internet has grown in importance in our day-to-day lives. We utilize it for a variety of business and non-business purposes, including banking, communication, entertainment, and shopping. Malicious programs and applications (often known as malware) are one of the biggest security risks the internet currently confronts. Malicious software, also known as malware, is created with the intention of causing harm or engaging in any type of undesirable activity on a computer system, including obstructing computer operations, gathering private information, getting around security measures, and displaying offensive advertisements. Every day, enormous volumes of malware are intentionally manufactured. The cost of harmful software has increased, and its market is always growing depending on how it functions, there are numerous types of malware, including adware, spyware, bot, virus, trojan, ransom wares, worm, and backdoor, among others [1], [2,3]. Therefore, Malware detection and classification has emerged as one of the most pressing issues in the security field. To fully comprehend the aim and components of the malware, a further classification can be created to identify the types and family classes of malware [4,5].

Malware analysis entails both the detection and classification of malware. Malicious or benign malware can be distinguished through detection. In contrast, classification entails

determining the specific malware family for a particular type of malware. There are two basic categories of malware analysis: static and dynamic. Figure 1 displays a regularly employed malware analysis taxonomy. Both manual and automated analysis is possible. Manual analysis necessitates subject expertise. On other hand, automatic analysis needs highly developed data science programming skills [6]. The primary mechanism of static analysis is to find binary files without running any software. It operates by taking malware binary's style signatures. Static analysis is one of the finest methods for identifying typical malware because it is quick and secure [7]. In contrast, during a dynamic analysis, a software's behavior is investigated, and from these findings, the software's intentions or purposes are inferred. Although it can detect sophisticated malware, it is time-consuming and prone to security threats [8]. In comparison to static analysis, dynamic analysis is a superior method, but it uses more time and memory and has scaling problems. The focus of our work is on static analysis.

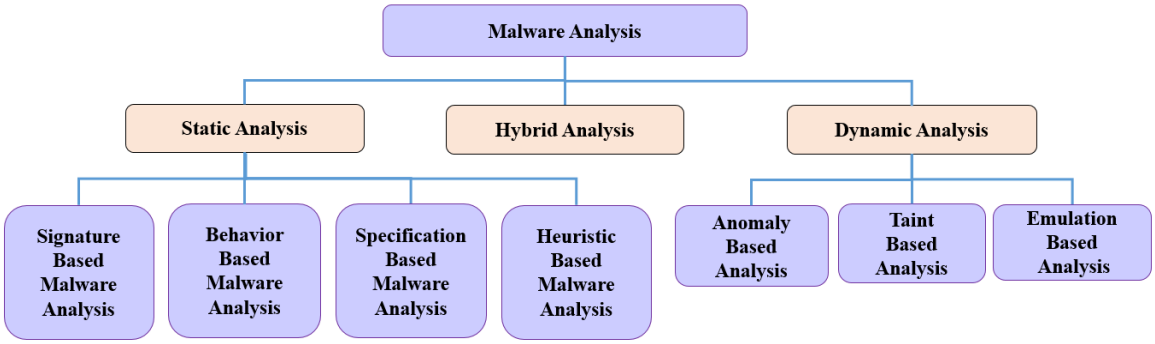

Figure 1. Existing Taxonomy of Malware Analysis.

Recently, numerous studies have been conducted to identify malware utilizing image processing techniques including texture analysis, entropy, and image matrix. It has been noted that textural analysis continues to be used actively in malware detection via imaging techniques. An essential component of computer vision is texture analysis. Most surfaces have some roughness to them. Malware images from the same family tend to have fairly similar patterns and textures throughout most malware databases. It is clear that while the malware photos do not technically have repeating patterns, they do have a lot of "texture," which can be used for automatic classification. There are striking visual similarities across malware images from the same family in terms of image texture [9]. Nataraj et al. [10] was the first work to propose malware classification based on textural features. In order to compute texture features, they employ GIST [11,12] which utilizes a wavelet decomposition of an image. Additionally, they demonstrated that texture analysis approaches using image processing may categorize malware faster than other malware classification approaches.

However, the vast majority of MC approaches, which rely on texture analysis, have a number of fundamental flaws, including a low classification rate since they classify malware using inaccurate and onerous features. The huge feature vector dimension results in a significant computational burden [13] and consumption of resources. The requirement for discovering precise and practical features for increasing the MC performance following malware has been detected. Additionally, this is to identify the top malware classification techniques. Segmentation-based fractal texture analysis (SFTA) and Gabor filters are two widely used computational techniques for texture analysis that are effective for classifying and segmenting textures. Therefore, the proposed method would leverage relevant texture features, namely SFTA and Gabor as well as GDA and NB as Classifiers.

Three steps make up the proposed malware analysis method: (a) malware conversion; (b) feature extraction; and (c) classification. We initially convert the RGB malware images into grayscale malware images for effective malware analysis. The SFTA and Gabor features are extracted from gray-scale images in the feature extraction step. Lastly, the classification is carried out by GDA and NB.

The following are the contributions of the proposed method:

1. To present an effective malware detection and classification method, SFTA and Gabor are extracted as distinctive feature vectors.
2. The usage of a malware visualization method that transforms binary files to 8 bit vectors for create grayscale graphics.
3. SFTA-GDA minimizes processing times and enhances overall detection/classification accuracy through texture features.
4. Experiment findings demonstrate that the proposed technique can accurately classify malware families.
5. Experimental results show that our proposed method can classify malware families with a low rate of false positives and false negatives.

The remainder of this article is organized as follows. The related works are discussed in Section 2. The selected features are described in Section 3. The proposed method is described in Section 4. Results and analyses are discussed in Section 5. Finally, in Section 6 a conclusion is drawn.

2. Related Works

This section provides an overview of earlier studies on malware detection (MD) and malware classification (MC) techniques. It can be seen that textural analysis is still used actively to find malware using image techniques.

Makandar et al. [4] presented the MC method as reliant on Gabor Wavelet, GIST and DWT. Malware is categorized using a Support Vector Machine (SVM) classification technique. On the Malimg Dataset, the proposed algorithm underwent testing. Verma et al. [5] presented malware classification as reliant on the first-order and GLCM-based second-order statistical texture features. The public Malimg malware Dataset was used to test the presented method. ELM is a classifier that has been used in the classification phase. Gandotra et al. [14] presented the MC method as dependent on static and dynamic features. Multiple classification algorithms were used, including IB1, decision tree, and random forest. Han et al. [15] presented the MC method as reliant on visualized images and entropy graphs. Determining the similarities of entropy graphs has been used to find and classify malware. Vinayakumar et al. [16] presented the MC method by using textural features which consisted of wavelet transform and Gabor transform. The KNN classifier was used in the classification stage. Fang et al. [17] presented the MC method as reliant on dynamic, static, and content-oriented features. In the classification scenario, a fuzzy random forest and an SVM are classifiers that have been applied. Kong et al. [18] presented the MC method by using structural information. The Assemble classifier was used in the classification stage. They employ the call graph method, which collects the features of each malware sample. Kosmidis and Kalloniatis [19] presented malware detection (MD) based on GIST feature extraction technique. The model had a detection accuracy of 91.6%. The malware is categorized using a random forest classification technique. On the MaleVis Dataset, the proposed algorithm underwent testing. Ban et al. [20] presented malware detection (MD) based on B2M (Binary mapping to image) algorithm, the SURF algorithm and the Local sensitive hashing (LSH) algorithm. The method had an 85% classification accuracy rate. Liu et al. [21] presented the MC method as reliant on GIST and multi-layer LBP features. The proposed method experimented on the Malimg Database. The RF classifier was used in the classification stage. Fu et al. [22] presented MC by using the global features and local features combined. Multiple classification algorithms were used, including support vector machine, random forest, and K-nearest neighbor. Liu and Wang [23] proposed the MC method based on local mean method. The ensemble learning classifier was used in the classification stage. Bozkir et al. [24] presented the MC method as reliant on GIST, HOG (Histogram of Gradients) descriptors and their combination. Multiple classification algorithms were used, including j48, RBF kernel-based SMO, Random Forest, XGBoost and linear SVM.

The majority of the MC techniques previously discussed are based on texture analysis techniques. In contrast to these MC techniques, the proposed method would leverage relevant texture features, namely SFTA and Gabor. These approaches' fundamental drawback is that they have a low classification rate since they classify malware using unreliable and cumbersome features. Another factor is a large feature vector dimension. As a result, the proposed technique lowers the risk of misclassification and increases classification accuracy. In addition, our method places more emphasis on machine learning classification to cut down on computing costs.

3. Multiple Features

Numerous image processing applications have had exceptional success with texture analysis methods. Malware images from the same family tend to have fairly similar patterns and textures throughout most malware databases. It is clear that the malware photos do not technically have repeating patterns, they do have a lot of "texture," which can be used for automatic classification. There are striking visual similarities across malware images from the same family in terms of image texture [9]. Therefore, the texture analysis plays a distinct role in the field of the malware classification.

The requirement to finding accurate and convenient features to increase the malware classification performance following malware has been detected. Because of their reliability and low computational cost, the texture feature descriptors SFTA and Gabor feature are utilized for texture feature extraction. The next subsection provides an explanation for each texture descriptor.

3.1. Segmentation-Based Fractal Texture Analysis (SFTA)

Segmentation-based fractal texture analysis (SFTA) is one of the popular texture approaches [25]. The most notable aspect of an image that is used to recognize and classify malware images and find similarities across images from different virus families is its texture. SFTA is used for texture feature extraction due to its dependability and affordable computation.

The SFTA extraction method can depend on two steps. Firstly, the set of binary images was created by applying the input grayscale image decomposition. The data were divided using the Two-Threshold Binary Decomposition (TTBD) technique [26].

Secondly, SFTA feature vectors are calculated as the average gray level, fractal dimension size, and additional SFTA feature vector. The complexity of malware image structures that are fractured in the input image are depicted using fractal estimations, as seen in Figure 2. For more details, see [25]. In order to extract the SFTA features, the following mathematical expression (Equation (1)) is employed.

$$\varnothing_{sfta}(U) = \begin{cases} 1 \;\; if \; \exists (i', j') \in N_8[(i,\, j)] : \\ \qquad \varnothing_e(i', j') = 0 \\ \qquad \varnothing_e(i,\, j) = 1 \\ \qquad 0 \; Otherwise \end{cases} \tag{1}$$

where N_8 $[(i, j)]$ represents the number of connected pixels initialized as 8 in this work. $\varnothing_e(i,\, j)$ is Binary image.

The size of the characteristics vector depends on how many thresholds are selected. For instance, seven binary images will be generated when we were considered them equal to three. Therefore, for each image, 21 features were created using the SFTA method.

As we already discussed, the majority of MD/MC methods suffer from limitations including a huge number of feature vectors and a high time complexity. However, due to their sturdiness and inexpensive processing, SFTA are preferred among texture image analysis techniques. Therefore, using SFTA features extraction in MD/MC is intriguing.

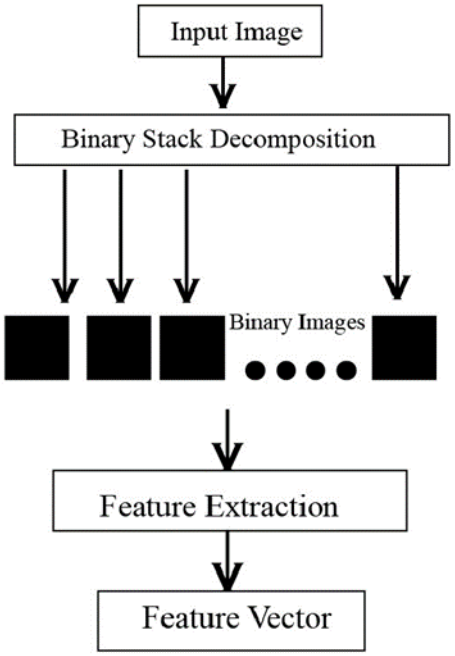

Figure 2. SFTA Extraction process.

3.2. Gabor Features

In image processing, Gabor filters [27] have indeed been widely employed for feature extraction. A coefficient matrix provided by Gabor filters allows for multi-resolution analysis. Thus, a 2D Gabor filter has been applied in order to extract features. A 2D Gabor can be generated in the time and frequency domain [28]. In the time domain, Gaussian function and a sinusoidal wave are produced. In the frequency domain, it is a convolution of the transformations of the Gaussian and sinusoid. In order to extract the 2D Gabor features, the following mathematical expression (Equation (2)) is employed.

$$\begin{aligned} G_{\theta,f,\sigma1,\sigma2}(x,y) &= exp\left[\frac{-1}{2}\left(\frac{x'^2}{\sigma_1^2}+\frac{y'^2}{\sigma_2^2}\right)\right]\ \cos(2\pi f x'+\varphi) \\ x' &= x\ \sin\theta + y\ \cos\theta \\ y' &= x\ \cos\theta + y\ \sin\theta \end{aligned} \tag{2}$$

where:

f = the spatial frequency of the wave at an angle θ with the x axis,
$\sigma1$ and $\sigma2$ = the standard deviations of the 2D Gaussian envelope,
φ = the phase.

In a number of image analysis and classification applications, Gabor filters are often used. Two frequent Gabor features [29], namely Mean Squared Energy and Mean Amplitude, are recovered across a range of orientations and sizes.

At different scales and orientations, Gabor features can be retrieved. Figure 3 shows the 2D Gabor filters in a variety of eight orientations and five scales [30].

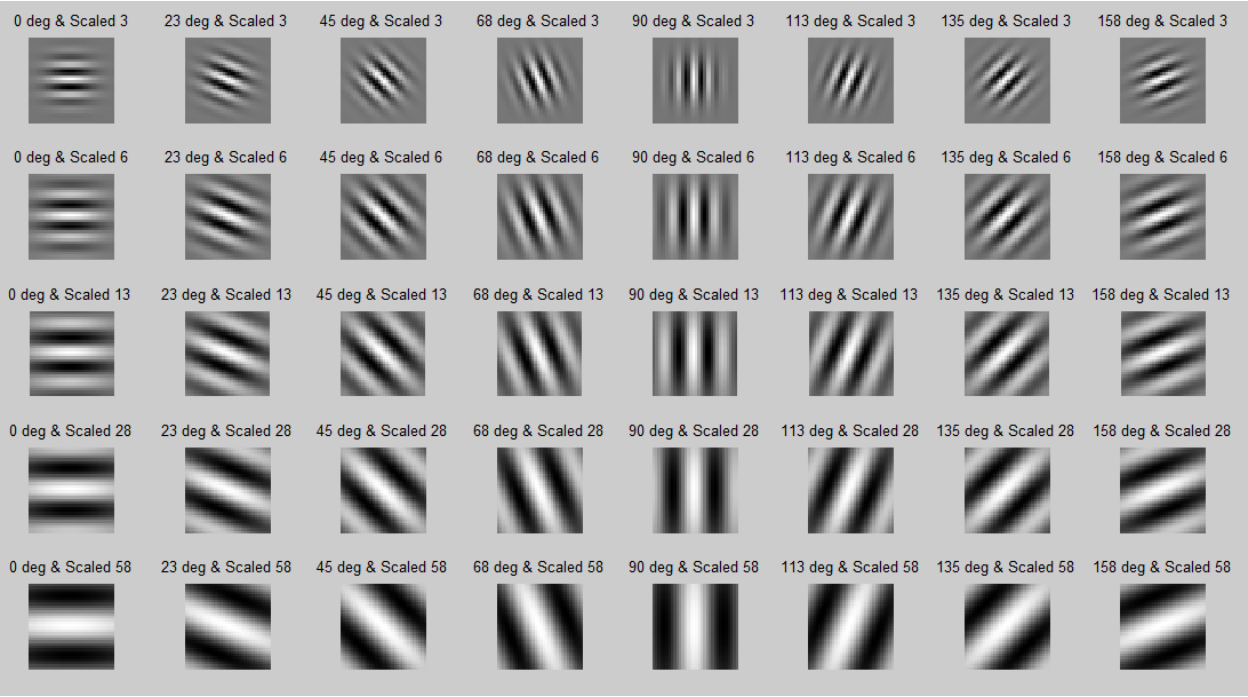

Figure 3. A two-dimensional Gabor filter with eight orientations and five scales [31].

Finally, feature vector [32] is obtained by extracting Mean Squared Energy and Mean Amplitude as feature vectors from the Response Matrices. Mean Squared Energy is calculated by adding the squared values of each matrix value in a response matrix. The mean amplitude of a response matrix is calculated as the sum of the absolute values of each matrix value. If you want to understand further about Gabor Features, I suggest reading [32].

4. The Proposed Methods

Three main processes make up the suggested technique for malware analysis: (a) malware conversion; (b) feature extraction; and (c) classification. For proper malware analysis, we first transform the RGB malware images into grayscale malware images. In the feature extraction step, the SFTA and Gabor features are extracted from grayscale images. Lastly, the classification is carried out by GDA and NB. Figure 4 depicts the proposed method's flowchart. Additionally, the Algorithm 1 was developed. In the subsections below, each step's full details are presented.

Step 1: **Malware Conversion**

The PE binary files (malware or Non-Malware) are often visualized and provided as input for malware analysis (detection and classification task). In the majority of malware detection and classification task, each PE binary file is converted into a 2D array and visualized as a grayscale image. While there is a significant variance between distinct families, the image textures of the same families are very similar [10]. However, the PE binary files in MaleVis dataset are visualized as RGB byte images that belong to 26 malware classes, including 25 malware and 1 Non-Malware. Therefore, it is necessary to convert these RGB images into grayscale images. Figure 5 demonstrates the conversion of RGB malware images into grayscale malware images.

Algorithm 1: Proposed MC_based GDA and NB Classifier.

Input: RGB Malware Image.
Output: Non-Malware/Malware Image.
Begin
For
1: Use the "Imread ()" function to read each image;
2: Convert the RGB image to the gray-scale image using Matlab function such as "rgb2gray ( )";
3: Then, the SFTA features {*Sftaf1*, *Sftaf2*, *Sftaf3*, *Sftaf4*, . . . *Sftaf21*} are extracted to obtain 1 × 21-dimension feature vector;
4: Extract the Gabor features vector:

A. Apply 2D Gabor filters to each image that has been converted.
B. Extract the mean squared energy and mean amplitude as the Gabor features {Gaborf1, Gaborf2, Gaborf3, Gaborf4 . . . Gaborf12} to obtain a 12-dimension feature vector.

5: Training:

A. Employ the above feature vectors to train the GDA classifier;
B. Employ the above feature vectors to train the NB classifier;

6: Testing:

A. The trained GDA model are tested to identify whether the image is non-Malware or Malware;
B. The trained NB model are tested to identify whether the image is non-Malware or Malware.

End for
End

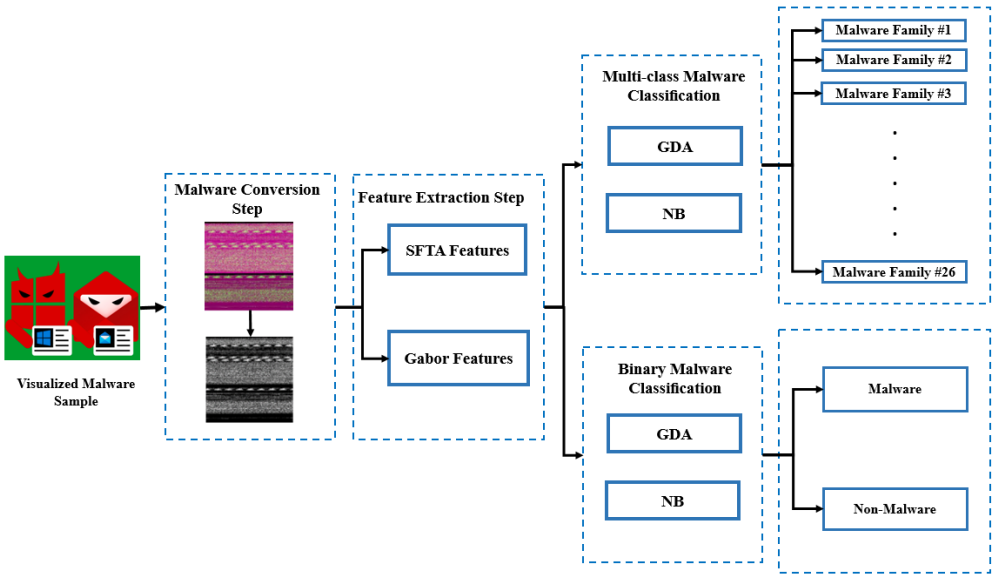

Figure 4. Proposed method Flowchart.

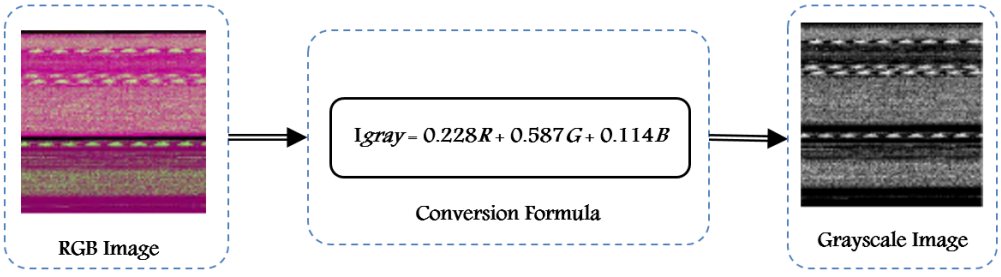

Figure 5. The Conversion Process Diagram.

Step 2: **Feature Extraction**

Following malware visualization, features are extracted for malware analysis. Both the machine learning (ML) and computer security (CS) communities have looked into feature extraction for malware analysis. As can be observed, malware classification (MC) frequently employs the same set of features as malware detection (MD).

There are typically two ways to extract image features: the first type includes extracting the global features from the entire image; in the second type, local feature points are extracted and then described using pertinent features.

The image's primary global features are texture, color, shape, and space of the image. We came to the conclusion that textural features were much more suitable and adequate as the global characteristics of malware after studying the traits and contained data of malware images.

SFTA and Gabor are two commonly used texture feature extraction methods. Due to its resilience and lower computational complexity compared to other methods, SFTA is the greatest fit for our purposes. Since Gabor is the best choice to reduce feature dimension when compared to the various texture methods, it was chosen to extract mean squared energy and mean amplitude features.

Step 2.1: **SFTA Features Extraction**

The malware may be easily recognized due to a texture-based feature that was generated from malware that could be seen. As shown in Algorithm 2, SFTA Texture features are extracted by hand-engineering methods. The SFTA feature vector that was obtained has a 1×21 dimension.

Algorithm 2: Compute SFTA textures features.

Input: Visualized Malware Image.
Output: 1×21 features vector dimension.

1. Open the malware image that was visualized.
2. Compute the SFTA using the Equation (1).
3. Twenty One features vector are produced.

Step 2.2: **Gabor Features Extraction**

Algorithm 3 illustrates the application of a 2D Gabor filter to extract features. Equation (2) is used to obtain the mean squared energy and mean amplitude Gabor features. The feature vectors' dimensions are 1×12.

Algorithm 3: Compute Gabor textures features.

Input: Visualized Malware Image.
Output: 1×12 features vector dimension.

1. Open the malware image that was visualized.
2. Apply 2D Gabor filters to each image that has been converted.
3. Calculate the mean squared energy and mean amplitude Gabor features using the Equation (2).
4. Twelve features vector are produced.

Step 3: **Classification**

It is usually worthwhile to assess how good the chosen features are and how good the model is before we get started with the classification step. In general, features and models are regarded as being a decent representation when we are able to correctly categorize the malware families using the chosen features and classifiers.

A review of the literature revealed that several studies using KNN, RF, NB, ELM, GDA, NN, and SVM showed improved accuracy findings. In this paper, we employed NB and GDA as useful methodologies for malware analysis.

Step 3.1: **Naive Bayes (NB) Classifier**

A probability-based classification technique called the Naive Bayes Algorithm counts the frequencies and permutations of values found in a dataset to create a set of likelihood. The top rated sample in the applicable class is included in Naive Bayes Classifier's system learning, which is based on test data [33].

Step 3.2: **Gaussian discriminant analysis (GDA) Classifier**

A specific generative learning method called GDA [34] attempts to separately fit a Gaussian distribution to every class of data in order to produce the distribution of several classes [26].

5. Results and Discussion

We take the presented methods to the test using a number of indications and then analyze the outcomes. Datasets, performance assessment measures, assessment outcomes, and comparing with certain other approaches are the four subsections that make up this section. The experiment was performed on a select few properties; for more details, view Table 1.

Table 1. Experimentation Properties Description.

Hardware	Properties
PC	HP laptop
Operating system	Microsoft Windows 10 64-bit (OS)
RAM	8 GB
Processor	Intel(R) Core(TM) i7-6500U CPU @ 2.50 GHz 2.60 GHz
Software	MATLAB version R2020a
Graphics Card	Intel® HD Graphics 520 (NVIDIA GTX 950M)

5.1. Datasets

MaleVis (Malware Evaluation with Vision) dataset [35] was utilized to gauge the effectiveness of the proposed method. The MaleVis dataset consists of 14,226 RGB byte images belonging to 26 malware classes which include 25 malware and 1 cleanware as shown in Table 2. These 14,226 RGB byte images were divided into 9100 samples for training and 5126 samples for testing. There are 350 images total throughout all classes, which are evenly distributed. The Malware classes included Adposhel, Agent-fyi, Allaple. A, Amonetize, Androm, AutoRun-PU, BrowseFox, Dinwod! rfn, Elex, Expiro-H, Fasong, HackKMS. A, Hlux! IK, Injector, InstallCore. C, MultiPlug, Neorekla-mi, Neshta, Regrun. A, Sality, Snarasite. D!tr, Stantinko, VBA/Hilium. A, VBKrypt, and Vilsel. The distribution of samples among the different malware classes contained in the datasets is shown in Figure 6. The images resolutions range between 224 × 224 and 300 × 300 pixels. The various malware classes in the MaleVis dataset are displayed in Figure 7.

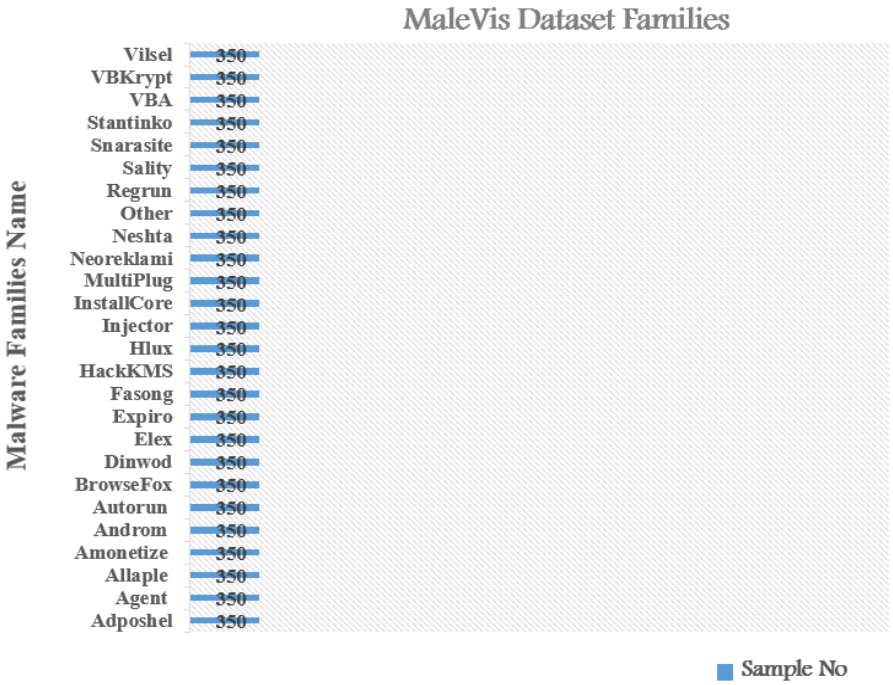

Figure 6. The distribution of MaleVis datasets samples among different malware classes.

Figure 7. Various Samples collected from the MaleVis Dataset [36].

Table 2. Explanation of the MaleVis Dataset Categories.

Class ID	Family	Details	
		Malware Category	Sample No.
#1	Adposhel	Adware	350
#2	Agent	Trojan	350
#3	Allaple	Worm	350
#4	Amonetize	Adware	350
#5	Androm	Backdoor	350
#6	Autorun	Worm	350
#7	BrowseFox	Adware	350
#8	Dinwod	Trojan	350
#9	Elex	Trojan	350
#10	Expiro	Virus	350
#11	Fasong	Trojan	350
#12	HackKMS	Riskware	350
#13	Hlux	Worm	350
#14	Injector	Trojan	350
#15	InstallCore	Adware	350
#16	MultiPlug	Adware	350
#17	Neoreklami	Adware	350
#18	Neshta	Virus	350
#19	Other	-	350
#20	Regrun	Trojan	350
#21	Sality	Virus	350
#22	Snarasite	Trojan	350
#23	Stantinko	Trojan	350
#24	VBA	Macro Malwares	350
#25	VBKrypt	Trojan	350
#26	Vilsel	Trojan	350
	Total	-	9100

5.2. Performance Evaluation Metric

In order to evaluate the proposed method, Classification accuracy is the percentage of samples that are correctly classified to all categories. The accuracy rate is less effective when classes are not balanced. It does give crucial information when the classes are balanced. In order to calculate Classification Accuracy, the following equation has been used [37]:

$$\text{Accuracy} = \frac{(\text{TP} + \text{TN})}{(\text{TP} + \text{TN} + \text{FP} + \text{FN})} \times 100\ \% \tag{3}$$

The terms of TP, TN, FP and FN can be calculated by the following formulas [38]:

1. The term "TP" (True Positive) refers to the variety of malware types that may be considered to be positive.
2. The term "TN" (True Negative) refers to the variety of malware types that may be considered to be negative.

3. The term "FP" (False Positive) refers to the variety of malware types that may be considered to be negative and positive.
4. The term "FN" (False Negative) refers to the variety of malware types that may be considered to be negative and positive.

5.3. Evaluation Results

This research employs a 10-fold cross-validation procedure to guarantee the precision and dependability of the experimental outcomes [39]. This paper used k-fold cross-validation (k-fold CV). This technique enables performance assessment using numerous distinct dataset combinations to reduce bias [40]. The dataset can be separated into ten sections. In each testing procedure, nine sections are chosen for the training set, and one section is utilized for the test set. Ten tests in total are carried out, and the method is ultimately evaluated by combining the outcomes of all the experiments.

The outcomes are divided into two category levels: (a) binary classification, and (b) multiclass classification. We extract vector features based on the two features SFTA and Gabor for both binary classification and multiclass classification outcomes. The performance of various machine learning classifiers that have been trained using Naive Bayesian (NB) and GDA is then evaluated through experiments. Tables 3 and 4 display the proposed method's accuracy rate using two classifiers on the MaleVis dataset.

Table 3. Detection Accuracy Rate of two classifiers on two features across MaleVis dataset.

Classifier	Detection Accuracy (%)	
	SFTA Feature	Gabor Feature
NB	84	83
GDA	97	93

5.3.1. Binary Malware Classification Results

For Binary Classification Results, we employ binary classification of malware against Non-malware (benign), where the malware class is easily generated by combining all MaleVis families into a single malware set. Whereas, another class is configured to be benign (non-malware).

After using the two texture descriptors SFTA and Gabor features on the MaleVis datasets for feature extraction, the NB and GDA classifiers are used in this instance for binary classification. In other words, the experiments were carried out for binary classification (malware or Non-malware (benign)) under the MaleVis dataset. The Detection Rate results of the proposed SFTA-GDA, SFTA-NB, Gabor-GDA, and Gabor-NB are shown in Table 3.

All classifiers use the exact same feature vector, however they all produce different results. This is so because every classifier has a distinct set of characteristics. The accuracy findings on MaleVis dataset in Table 3 indicate that the performance of the SFTA-GDA attained a high accuracy rate of 97%. Whereas, the accuracy rate of Gabor-GDA was 93%. Furthermore, it is evident that the SFTA-NB performed much better on the MaleVis dataset, achieving a high classification accuracy of 84% compared to Gabor-NB which was 83%.

Table 4. Classification Accuracy rate of two classifiers on two features across MaleVis Dataset.

Class ID	Family Name	Classification Accuracy (%)			
		Naive Classifier		GDA Classifier	
		SFTA Feature	Gabor Feature	SFTA Feature	Gabor Feature
#1	Adposhel	97	90	99	94
#2	Agent	89	52	99	96
#3	Allaple	76	78	98	97
#4	Amonetize	89	62	99	96
#5	Androm	84	52	97	95
#6	Autorun	84	94	98	96
#7	BrowseFox	61	95	99	95
#8	Dinwod	89	55	97	96
#9	Elex	95	80	98	95
#10	Expiro	66	58	99	95
#11	Fasong	98	94	96	95
#12	HackKMS	97	99	99	98
#13	Hlux	99	98	99	99
#14	Injector	73	88	99	95
#15	InstallCore	99	99	96	96
#16	MultiPlug	88	66	97	96
#17	Neoreklami	87	70	99	96
#18	Neshta	95	95	99	97
#19	Regrun	94	91	99	95
#20	Sality	95	80	99	95
#21	Snarasite	99	99	99	95
#22	Stantinko	83	85	99	93
#23	VBA	85	95	99	98
#24	VBKrypt	62	96	97	95
#25	Vilsel	99	99	99	99
Average		87	82%	98%	95%

For both of classifiers, SFTA-GDA and SFTA-NB had the highest detection accuracy. However, the performance of the SFTA-GDA attained a high accuracy rate of 97%. In contrast, the accuracy rate of SFTA-NB was 84 percent as shown in Figure 8. For this reason, we considered the suggested technique based on SFTA and GDA classifier in the initial findings section of this work. The above results are presented for the Binary Classification, while the findings from the multiclass classification are presented in the next section.

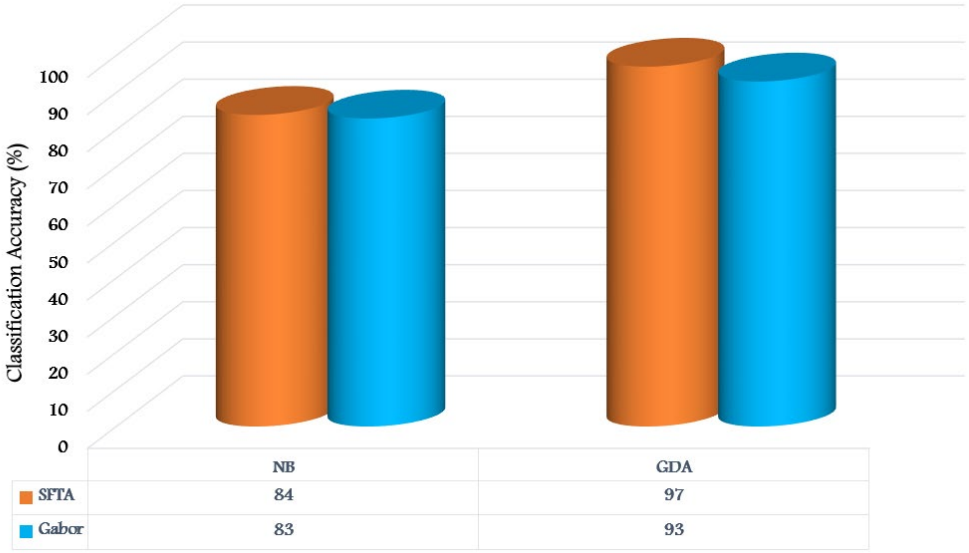

Figure 8. Detection Accuracy Rate of Proposed Method.

5.3.2. Multi-Class Malware Classification Results

We make an effort to group the malware samples into the appropriate families in order to produce multiclass classification findings. The MaleVis dataset contains 25 malware families, thus we have 25 classes to use for this classification issue. Table 4 shows the collected results for each malware family, as well as the overall average of accuracy metric that was determined after applying each classifier. The multi-class malware classification results using the SFTA-GDA, SFTA-NB, Gabor-GDA, and Gabor-NB are shown in Table 4.

According to the average accuracy results on the MaleVis dataset, the performance of the SFTA-GDA attained a high average accuracy rate of 98%. Meanwhile, the average accuracy rate for Gabor-GDA was 95%. Additionally, it is clear that the SFTA-NB outperformed Gabor-NB on the MaleVis dataset by reaching a high classification average accuracy of 87% as opposed to 82%. The maximum classification accuracy was achieved by SFTA-GDA and SFTA-NB for both classifiers. However, the performance of the SFTA-GDA attained a high accuracy rate of 98%. Meanwhile, the accuracy rate of SFTA-NB was 87 percent as shown in Figures 9 and 10.

As can be seen, when the NB classifier is employed with SFTA and Gabor texture descriptors, its performance suffers when classifying malware images. For ten different families in the MaleVis Dataset, the accuracy ranges from 52% to 73%. However, when the GDA classifier was employed with SFTA and Gabor texture descriptors, its performance significantly improved when classifying malware images. For all families in the MaleVis Dataset, the accuracy ranges from 94% to 99%.

As can be observed, the performance of SFTA-NB ranges from 61% to 73% for the four different families (BrowseFox, Expiro, VBKrypt, and Injector). Additionally, the performance of Gabor-NB for seven different families (Agent, Amonetize, Androm, Dinwod, Expiro, MultiPlug, and Neoreklami) ranges from 52% to 70%. However, the performance of SFTA-GDA has greatly increased and currently varies from 96% to 99% for all families. Moreover, the performance of Gabor-GDA has greatly improved and now ranges from 94% to 99% for all families as demonstrated in Table 4.

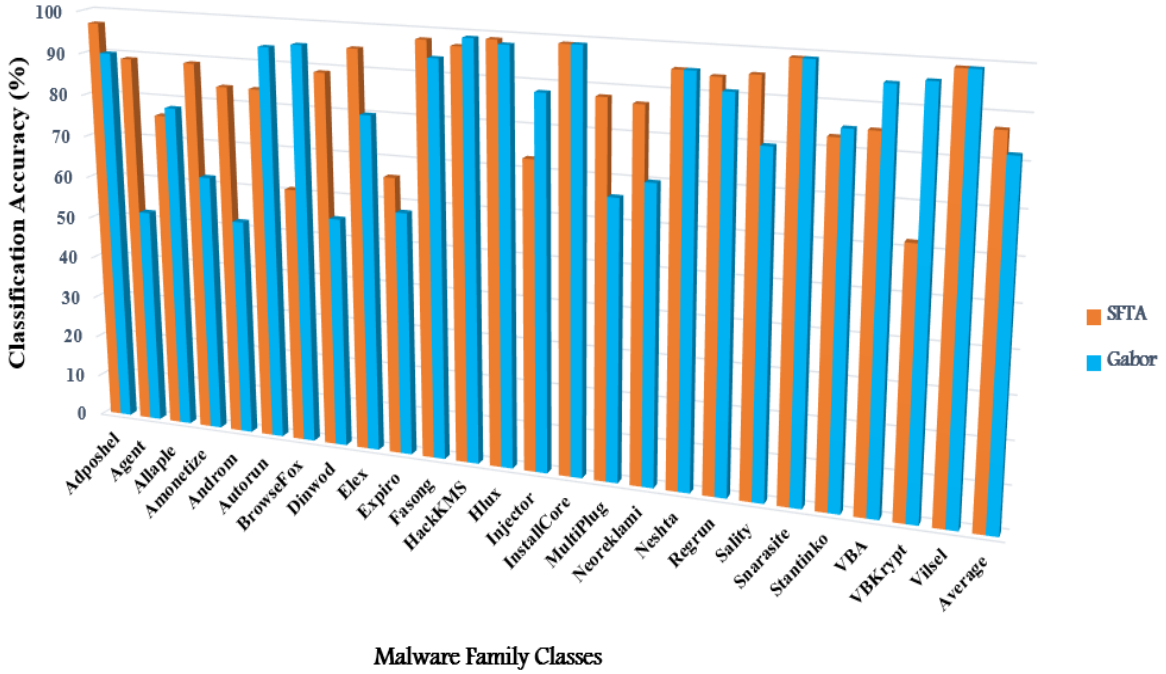

Figure 9. The classification Accuracy-based SFTA and Gabor features across NB Classifier.

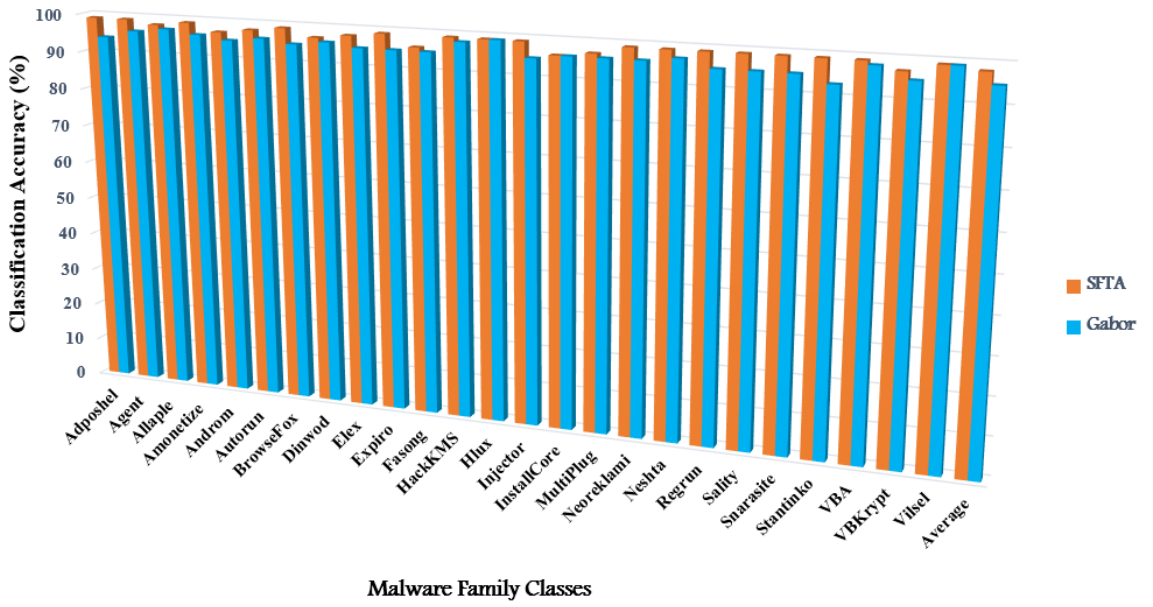

Figure 10. The classification Accuracy-based SFTA and Gabor features across GDA Classifier.

In the majority of recent works such [36], the Neshta class, which is a member of the virus family, had the lowest accuracy. However, the SFTA-NB, Gabor-NB, SFTA-GDA, and Gabor-GDA methods achieved better classification accuracy, specifically for the Neshta class, at 87%, 82%, 98%, and 95%, respectively.

The findings demonstrate that the GDA classifier, when combined with SFTA features, has an important effect, particularly in some classes, such as (BrowseFox, Expiro, VBKrypt, and Injector), whose overall accuracy increased from 61%, 66%, 73%, 62% to 99%, 99%, 99%, 97%. It is obvious that using a GDA classifier greatly enhances the results.

It can be concluded that employing SFTA-GDA, the accuracy is seen to be greatly enhanced for all families of the MaleVis Dataset.

5.4. Existing Methods Comparison Results

In order to evaluate the effectiveness of our proposal, we compare it to other techniques in this section. The key component of the majority of these algorithms for classifying malware is the extraction of textural features. As shown in Table 5, the proposed technique is compared with a variety of other state-of-the-art malware classification techniques [4,36,41–45] that are based on Hand-crafted Features.

Table 5. Comparative Findings of current MD/MC Methods.

Methods	Data Analysis	Feature Kind	Classifier Kind	Dataset	Accuracy (%)
Kang et al. [41]	Static	creator information	SVM	Malware	90
Makandar et al. [4]	Static	Gabor GIST DWT	KNN	Malimg	98
Aziz et al. [42]	Static	DWT	SVM	Mahenhuer	92
Hashemi et al. [43]	Static	LBP	KNN	Malimg	91
Liu et al. [21]	Static	GIST	RF	Malimg	91
Nisa et al. [44]	Static	SFTA	SVM	Malimg	95
Nisa et al. [44]	Static	Fused SFTA and DNN features	cubic SVM	Malimg	99
Patil et al. [36]	Static	-	Random f	MaleVis	93
Mohammed et al. [45]	Static	DCT	CNN	MaleVis	96
Proposed (SFTA-GDA)	Static	SFTA	GDA	MaleVis	98

Refs. [21,41,43,44] Malware detection methods are based on the spatial domain. The techniques [21,43,44] are based on LBP, GIST, and SFTA, respectively.

The transform domain is utilized by both [4,42] Malware detection techniques. The [4,42] approaches are based on the discrete wavelet transform (DWT). The current methods in [4,42] have the highest levels of accuracy. However, as seen in Table 5, they have the drawback of requiring a lot of time. The usage of the transform domain is the primary explanation.

The [44,45] approaches are based on the merging of deep features with handmade features. The existing techniques in [44,45] provided successful outcomes. However, the combined feature of handcrafted features and deep features, as shown in Table 5, required greater time consumption.

Other spatially based malware classification approaches [21,41,43] perform worse than our proposed method. In terms of classification accuracy rates, the proposed method, which does not use the transform domain and deep features, beats other existing methods.

The accuracy rate of the proposed method was 98%, which is higher than the overall accuracy from the existing state-of-the-art methods.

6. Conclusions

The main goal of the proposed method would be to use infected photos to extract a strong feature that will increase the classification performance. The proposed malware analysis approach consists of three steps: malware conversion, feature extraction, and classification. For efficient malware analysis, we first transform the RGB malware images to grayscale versions. In the feature extraction step, gray-scale images are used to extract the SFTA and Gabor features. Finally, naïve Bayes (NB) and Gaussian Discriminant Analysis (GDA) are used as the classifier. A typical MaleVis dataset is used to assess the proposed

method. Due to its superior accuracy rate when compared to all other families in the MaleVis Dataset, the proposed SFTA-GDA was the best option. The experiment findings show that the proposal can accurately and efficiently classify malware samples to their appropriate families by combining Segmentation-based fractal texture analysis (SFTA) and Gaussian Discriminant Analysis (GDA). The accuracy rate of the proposed method was 98%, which is higher than the overall accuracy of the currently available state-of-the-art methods. Even if our method yields the high classification accuracy rate, it is still necessary to extract more potent malware features. Future work includes using deep learning models including CNN because of its powerful capability in characterizing features. This will reduce the amount of manual participation. Additionally, only the Malevis dataset is used to evaluate the suggested technique. We would then prefer to evaluate our method on more datasets in the future.

Author Contributions: Conceptualization, I.T.A., B.T.H. and N.J.; writing—original draft preparation, I.T.A.; writing—Original Draft Preparation, B.T.H.; review & editing, N.J.; Validation, M.M.D.; Software, I.T.A.; Validation, N.J.; Funding acquisition, M.M.D. All authors have read and agreed to the published version of the manuscript.

Funding: This research is supported and funded by the Publication Fund under the Tan Sri Leo Moggie Chair of Energy Informatics, Institute of Informatics and Computing in Energy, Universiti Tenaga Nasional.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Poudyal, S.; Akhtar, Z.; Dasgupta, D.; Gupta, K.D. Malware analytics: Review of data mining, machine learning and big data perspectives. In Proceedings of the 2019 IEEE Symposium Series on Computational Intelligence (SSCI), Xiamen, China, 6–9 December 2019; pp. 649–656.
2. Hemalatha, J.; Roseline, S.A.; Geetha, S.; Kadry, S.; Damaševičius, R. An efficient densenet-based deep learning model for malware detection. *Entropy* **2021**, *23*, 344. [CrossRef] [PubMed]
3. O'Brien, D. Internet Security Threat Report-Ransomware 2017. *Symantec* **2017**, *11*, 203–214.
4. Makandar, A.; Patrot, A. Malware class recognition using image processing techniques. In Proceedings of the 2017 International Conference on Data Management, Analytics and Innovation (ICDMAI), Pune, India, 24–26 February 2017; pp. 76–80.
5. Verma, V.; Muttoo, S.K.; Singh, V.B. Multiclass malware classification via first-and second-order texture statistics. *Comput. Secur.* **2020**, *97*, 101895. [CrossRef]
6. Aslan, Ö. Performance comparison of static malware analysis tools versus antivirus scanners to detect malware. In Proceedings of the International Multidisciplinary Studies Congress (IMSC), Solin, Croatia, 20–21 April 2017.
7. Naeem, H.; Guo, B.; Naeem, M.R.; Ullah, F.; Aldabbas, H.; Javed, M.S. Identification of malicious code variants based on image visualization. *Comput. Electr. Eng.* **2019**, *76*, 225–237. [CrossRef]
8. Bayer, U.; Moser, A.; Kruegel, C.; Kirda, E. Dynamic analysis of malicious code. *J. Comput. Virol.* **2006**, *2*, 67–77. [CrossRef]
9. Nataraj, L.; Karthikeyan, S.; Jacob, G.; Manjunath, B.S. Malware images: Visualization and automatic classification. In Proceedings of the 8th International Symposium on Visualization for Cyber Security, Pittsburgh PA, USA, 20 July 2011; pp. 1–7.
10. Nataraj, L.; Yegneswaran, V.; Porras, P.; Zhang, J. A comparative assessment of malware classification using binary texture analysis and dynamic analysis. In Proceedings of the 4th ACM Workshop on Security and Artificial Intelligence, Chicago, IL, USA; 2011; pp. 21–30.
11. Torralba, A.; Murphy, K.P.; Freeman, W.T.; Rubin, M.A. Context-based vision system for place and object recognition. In Proceedings of the Ninth IEEE International Conference on Computer Vision, Nice, France, 13–16 October 2003; Volume 2, p. 273.
12. Oliva, A.; Torralba, A. Modeling the shape of the scene: A holistic representation of the spatial envelope. *Int. J. Comput. Vis.* **2001**, *42*, 145–175. [CrossRef]
13. Han, K.; Kang, B.; Im, E.G. Malware analysis using visualized image matrices. *Sci. World J.* **2014**, *2014*, 132713. [CrossRef]
14. Gandotra, E.; Bansal, D.; Sofat, S. Integrated framework for classification of malwares. In Proceedings of the 7th International Conference on Security of Information and Networks, Scotland, UK, 9–11 September 2014; pp. 417–422.
15. Han, K.S.; Lim, J.H.; Kang, B.; Im, E.G. Malware analysis using visualized images and entropy graphs. *Int. J. Inf. Secur.* **2015**, *14*, 1–14. [CrossRef]

16. Vinayakumar, R.; Alazab, M.; Soman, K.P.; Poornachandran, P.; Venkatraman, S. Robust intelligent malware detection using deep learning. *IEEE Access* **2019**, *7*, 46717–46738. [CrossRef]
17. Li, F.-Q.; Wang, S.-L.; Liew, A.W.-C.; Ding, W.; Liu, G.-S. Large-Scale Malicious Software Classification With Fuzzified Features and Boosted Fuzzy Random Forest. *IEEE Trans. Fuzzy Syst.* **2020**, *29*, 3205–3218. [CrossRef]
18. Kong, D.; Yan, G. Discriminant malware distance learning on structural information for automated malware classification. In Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Chicago, IL, USA, 11–14 August 2013; pp. 1357–1365.
19. Kosmidis, K.; Kalloniatis, C. Machine learning and images for malware detection and classification. In Proceedings of the 21st Pan-Hellenic Conference on Informatics, Larissa, Greece, 28–30 September 2017; pp. 1–6.
20. Xiaofang, B.; Li, C.; Weihua, H.; Qu, W. Malware variant detection using similarity search over content fingerprint. In Proceedings of the 26th Chinese Control and Decision Conference (2014 CCDC), Changsha, China, 31 May–2 June 2014; pp. 5334–5339.
21. Liu, Y.; Lai, Y.-K.; Wang, Z.-H.; Yan, H.-B. A new learning approach to malware classification using discriminative feature extraction. *IEEE Access* **2019**, *7*, 13015–13023. [CrossRef]
22. Fu, J.; Xue, J.; Wang, Y.; Liu, Z.; Shan, C. Malware visualization for fine-grained classification. *IEEE Access* **2018**, *6*, 14510–14523. [CrossRef]
23. Liu, L.; Wang, B. Malware classification using gray-scale images and ensemble learning. In Proceedings of the 2016 3rd International Conference on Systems and Informatics (ICSAI), Shanghai, China, 19–21 November 2016; pp. 1018–1022.
24. Bozkir, A.S.; Tahillioglu, E.; Aydos, M.; Kara, I. Catch them alive: A malware detection approach through memory forensics, manifold learning and computer vision. *Comput. Secur.* **2021**, *103*, 102166. [CrossRef]
25. Costa, A.F.; Humpire-Mamani, G.; Traina, A.J.M. An efficient algorithm for fractal analysis of textures. In Proceedings of the 2012 25th SIBGRAPI Conference on Graphics, Patterns and Images, Ouro Preto, Brazil, 22–25 August 2012; pp. 39–46.
26. Hammad, B.T.; Ahmed, I.T.; Jamil, N. A Steganalysis Classification Algorithm Based on Distinctive Texture Features. *Symmetry* **2022**, *14*, 236. [CrossRef]
27. Daugman, J.G. Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters. *JOSA A* **1985**, *2*, 1160–1169. [CrossRef]
28. Song, X.; Liu, F.; Zhang, Z.; Yang, C.; Luo, X.; Chen, L. 2D Gabor filters-based steganalysis of content-adaptive JPEG steganography. *Multimed. Tools Appl.* **2017**, *76*, 26391–26419. [CrossRef]
29. Zheng, D.; Zhao, Y.; Wang, J. Features extraction using a Gabor filter family. In Proceedings of the sixth Lasted International Conference, Signal and Image Processing, Hawaii, HI, USA, 23–25 August 2004.
30. SwagotaBera, D.; Sharma, M.; Singh, B. Feature extraction and analysis using Gabor filter and higher order statistics for the JPEG steganography. *Int. J. Appl. Eng. Res.* **2018**, *13*, 2945–2954.
31. Ahmed, I.T.; Hammad, B.T.; Jamil, N. Common Gabor Features for Image Watermarking Identification. *Appl. Sci.* **2021**, *11*, 8308. [CrossRef]
32. Kamarainen, J.-K. Gabor features in image analysis. In Proceedings of the 2012 3rd International Conference on Image Processing Theory, Tools and Applications (IPTA), Istanbul, Turkey, 15–18 October 2012; pp. 13–14.
33. Lowd, D.; Domingos, P. Naive Bayes models for probability estimation. In Proceedings of the 22nd International Conference on Machine Learning, Bonn, Germany, 7–11 August 2005; pp. 529–536.
34. Sharifi, K.; Leon-Garcia, A. Estimation of shape parameter for generalized Gaussian distributions in subband decompositions of video. *IEEE Trans. Circuits Syst. Video Technol.* **1995**, *5*, 52–56. [CrossRef]
35. Bozkir, A.S.; Cankaya, A.O.; Aydos, M. Utilization and comparision of convolutional neural networks in malware recognition. In Proceedings of the 2019 27th Signal Processing and Communications Applications Conference (SIU), Sivas, Turkey, 24–26 April 2019; pp. 1–4.
36. Patil, S.; Varadarajan, V.; Walimbe, D.; Gulechha, S.; Shenoy, S.; Raina, A.; Kotecha, K. Improving the Robustness of AI-Based Malware Detection Using Adversarial Machine Learning. *Algorithms* **2021**, *14*, 297. [CrossRef]
37. Hammad, B.T.; Jamil, N.; Ahmed, I.T.; Zain, Z.M.; Basheer, S. Robust Malware Family Classification Using Effective Features and Classifiers. *Appl. Sci.* **2022**, *12*, 7877. [CrossRef]
38. Ahmed, I.T.; Hammad, B.T.; Jamil, N. A comparative analysis of image copy-move forgery detection algorithms based on hand and machine-crafted features. *Indones. J. Electr. Eng. Comput. Sci.* **2021**, *22*, 1177–1190.
39. Ahmed, I.T.; Hammad, B.T.; Jamil, N. Image Copy-Move Forgery Detection Algorithms Based on Spatial Feature Domain. In Proceedings of the 2021 IEEE 17th International Colloquium on Signal Processing & Its Applications (CSPA), Langkawi, Malaysia, 5–6 March 2021; pp. 92–96.
40. Ahmed, I.T.; Der, C.S.; Jamil, N.; Mohamed, M.A. Improve of contrast-distorted image quality assessment based on convolutional neural networks. *Int. J. Electr. Comput. Eng.* **2019**, *9*, 5604–5614. [CrossRef]
41. Kang, H.; Jang, J.; Mohaisen, A.; Kim, H.K. Detecting and classifying android malware using static analysis along with creator information. *Int. J. Distrib. Sens. Netw.* **2015**, *11*, 479174. [CrossRef]
42. Makandar, A.; Patrot, A. Wavelet statistical feature based malware class recognition and classification using supervised learning classifier. *Orient. J. Comput. Sci. Technol.* **2017**, *10*, 400–406. [CrossRef]
43. Hashemi, H.; Hamzeh, A. Visual malware detection using local malicious pattern. *J. Comput. Virol. Hacking Tech.* **2019**, *15*, 1–14. [CrossRef]

44. Nisa, M.; Shah, J.H.; Kanwal, S.; Raza, M.; Khan, M.A.; Damaševičius, R.; Blažauskas, T. Hybrid malware classification method using segmentation-based fractal texture analysis and deep convolution neural network features. *Appl. Sci.* **2020**, *10*, 4966. [CrossRef]
45. Mohammed, T.M.; Nataraj, L.; Chikkagoudar, S.; Chandrasekaran, S.; Manjunath, B.S. Malware detection using frequency domain-based image visualization and deep learning. *arXiv* **2021**, arXiv:2101.10578.