INSTITUTE FOR HOMELAND SECURITY

Sam Houston State University

DEEPGRAY:

A NOVEL APPROACH TO MALWARE CLASSIFICATION USING GRAYSCALE IMAGES WITH DEEP LEARNING

Institute for Homeland Security

Sam Houston State University

Harshitha Polsani
Haodi Jiang

DeepGray: A Novel Approach to Malware Classification Using Grayscale Images with Deep Learning

1st Harshitha Polsani
Department of Computer Science
Sam Houston State University
Huntsville, TX 77341, USA
hxp029@shsu.edu

2nd Haodi Jiang
Department of Computer Science
Sam Houston State University
Huntsville, TX 77341, USA
hxj024@shsu.edu

Abstract

In the ever-evolving landscape of cybersecurity, the threat posed by malware continues to loom large, necessitating innovative and robust approaches for its effective detection and classification. In this paper, we introduce a novel method, DeepGray, for multi-class malware classification utilizing grayscale images and the power of deep learning. Our dataset combines the malware sample from the BODMAS dataset and the benign sample from the DikeDataset. Our approach involves transforming executable files into a format suitable for deep learning by converting them into grayscale images while retaining the essential data characteristics. During the data preprocessing step, applied Principal Component Analysis (PCA) was applied to distill the most significant features. To achieve state-of-the-art results in multi-class malware classification, we harnessed the power of deep learning and transfer learning, employing well-established neural network architectures such as a customized Convolutional Neural (CNN), VGG16, EfficientNet, and Vision Transformers (ViT). The models were meticulously trained and rigorously evaluated using a 5-fold cross-validation methodology. Notably, our approach yielded remarkable results, with ViT achieved an impressive accuracy of 0.95. This research underscores the potential of grayscale image analysis and deep learning within the domain of multi-class malware classification. The insights derived from this study contribute significantly to the field of cybersecurity and pave the way for further advancements in the realm of malware detection and classification.

***Index Terms*— Malware Images, Transfer Learning, Convolutional Neural Networks**

IV. INTRODUCTION

The continuous evolution of malware of malware presents formidable challenges to cybersecurity, demanding ingenious solutions for the timely and precise detection and classification of these threats. Malicious software, commonly referred to as malware, represents a grave peril to computer systems as it exploits vulnerabilities to gain unauthorized access and inflict damage. Traditional signature-based detection methods often struggle to keep pace with the continually mutating and obfuscating malware. In response, there is growing interest in exploring innovative approaches for effective malware analysis and classification. Leveraging the success of deep learning in image recognition tasks, deep learning methods have been proposed for malware classification using malware grayscale images.

To develop a resilient model capable of precision in the classification of malware samples across a multitude of categories, effectively tackling the rich diversity within the realm of malware types. In times past, traditional signature-based detection methods, typified by Support Vector Machine (SVM) algorithms, have long held sway in the field of malware classification. However, given the nuanced and intricate variations inherent in malware, these conventional methods may falter in their quest for optimal accuracy. In this evolving landscape, deep learning techniques, particularly the prowess of Convolutional Neural Networks (CNNs), have emerged as a beacon of promise. They offer the capacity to discern complex patterns and glean meaningful representations directly from raw data. This potential presents an exciting avenue for elevating the efficacy of malware classification through the prism of image-based approaches.[3]

This work is supported by the Institute for Homeland Security (IHS) at Sam Houston State University

The incorporation of grayscale images not only simplifies our input data but also opens doors to harnessing the benefits of deep learning within the domain of image recognition tasks. Grayscale images adeptly encapsulate the visual characteristics of malware samples, imbuing our deep learning model with the capability to differentiate and identify malware across a broad spectrum of categories. This image-centered analytical approach paves the way for pioneering avenues to strengthen our malware detection capabilities, thereby delivering valuable insights and contributions that extend beyond the realm of computer security.[2]

As we proceed with this research, we aim to demonstrate the effectiveness of our approach by conducting comprehensive experiments and evaluations on the chosen dataset. By leveraging the capabilities of deep learning and grayscale image analysis, we seek to develop a highly efficient and accurate malware classification model that can contribute significantly to the ongoing efforts in combating security threats. Our evaluation includes employing well-established deep learning architectures, such as VGG16, EfficientNetV2B0, InceptionV3, and Vision Transformers (ViT) as our base models for transfer learning. By fine-tuning these pre-trained models on our augmented dataset, our system achieves accurate malware classification results, showcasing the effectiveness and adaptability of our approach.

The rest of this paper is organized as follows. Section II provides an overview of related work in the field of malware detection, highlighting existing methodologies and techniques used for detecting and classifying malware. Section III presents details about the dataset used in our study, including its composition, data preprocessing, and class distribution. Section IV introduces the proposed methodology, outlining the steps involved in malware classification, feature extraction, and model selection. In Section V, the experimental results are presented, providing a comprehensive analysis of the model's performance and its effectiveness in accurately classifying diverse malware samples. Finally, Section VI serves as the conclusion of our work, summarizing the key findings and implications of our research

V. Related Work

There has been significant research focused on malware analysis as it has emerged as a critical challenge due to the proliferation of malware families with increasing complexity. Deep learning techniques, particularly convolutional neural networks (CNNs), have shown great promise in addressing this problem. Pant et al. and Bista et al. present an experimental study on classifying grayscale malware images into their respective families with high accuracy and low loss using the Malimg malware dataset. They explored the effectiveness of transfer learning using pretrained VGG16, ResNet-18, and InceptionV3 models, along with a custom CNN model. Their custom CNN model achieves best remarkable accuracy of 98.2in classifying malware, outperforming the rest of models using transfer learning. The research highlights the potential of CNNs in image-based malware classification, emphasizing the need for further exploration and fine-tuning to enhance performance on varying datasets [4].

Kalash et al. present an innovative CNN-based framework for malware classification. Their approach involves converting malware binaries into grayscale images and leveraging CNNs to automatically learn discriminative representations from the data, eliminating the need for hand-crafted features. Remarkably, the proposed method achieves outstanding accuracy of 98.52% and 99.97% on the Malimg and Microsoft malware datasets, respectively, surpassing state-of-the-art results and even outperforming the winning team of the Microsoft Malware Classification Challenge [8]. These compelling experimental findings underscore the efficacy of CNNs in addressing the growing complexity and volume of malware threats [5].

In the work conducted by Deepa et al. and Vinod et al., the authors proposed a deep learning-based approach for malware classification using a combination of Convolutional Neural Networks (CNNs) and different classifiers. The methodology involves converting malware binaries into grayscale images and then employing the pre-trained VGG16 model as a feature extractor to obtain 21,055-dim feature vectors. These extracted features are used to train multiple classifiers, including Support Vector Machine (SVM), Random Forest (RF), XGBoost, and Deep Neural Network (DNN), for the classification task into various malware

families. The experiments are performed on the 'Malimg' dataset, which contains 9,339 samples from 25 different malware families. The results showcase impressive accuracy for the classifiers, with SVM achieving 98.51%, RF achieving 97.60%, XGBoost achieving 97.39%, and DNN achieving 98.51% accuracy [6].

Sandip Shinde et al. presented a novel approach utilizing the Dike Dataset, which contains labeled samples of both malicious and benign portable executables (PEs) and object-linking and embedding (OLE) files, to train their model. To handle the challenge of detecting malware in diverse and ever-changing forms, they adopted an image-based classification technique. The process involved converting the executable files into grayscale images and subsequently applying a deep convolutional neural network (CNN) model for feature extraction. Notably, they employed the EfficientNet architecture, specifically implementing models B0 to B3. The trained models were then evaluated on the Dike Dataset, yielding impressive accuracy results. Specifically, EfficientNetB0 achieved an accuracy of 94.62%, EfficientNetB1 attained 95.46%, and EfficientNetB2 demonstrated the highest accuracy of 95.63%. However, EfficientNetB3 showed a slightly lower accuracy of 94.28%, potentially due to the limitations of the dataset size. The study successfully showcased the potential of using EfficientNet for effective and efficient binary classification of malware and benign [7].

Vasundhara et al. proposed a novel approach for malware classification using the EfficientNet-B1 model. They utilized the MMCC dataset, which consists of 10,868 samples from 9 different malware families. The methodology involved reading the malware binary files as 8-bit unsigned integer vectors and converting them into 2D grayscale images for visualization. The dataset was split into train and test sets, and the images were resized to 260x260 and normalized. The EfficientNet-B1 model pretrained on the ImageNet dataset was used with the last layer adapted for malware family classification. The experimental results showed that the proposed model outperformed other pretrained models, achieving an impressive accuracy of 98.57% in classifying unknown malware samples from the test set, effectively detecting subtle variations in the malware families. The study demonstrated the efficacy of deep learning and the EfficientNet-B1 model for robust and accurate malware classification [9].

Lok Man Kwan et al. proposed a method for enhancing the accuracy of malware detection and classification. The study involved two datasets: the Microsoft Malware Classification Challenge's dataset with 10,868 samples spread across 9 classes, and the REWEMA dataset containing 3,136 benign and 3,136 malware data. The proposed method consisted of two parts: Markov image generation based on byte-level transfer frequency and transfer learning with VGG19 for deep learning. The Markov image was generated using three steps: counting byte value transitions, calculating probabilities, and forming the Markov image. Additionally, three extra steps were proposed to enhance the Markov image's performance. For transfer learning, only the convolutional layer of VGG19 was used, and additional layers were added. The proposed method achieved an accuracy of 0.987 and a loss of 0.062 for malware classification, and an accuracy of 0.973 with a loss of 0.076 for malware detection. The comparative study with the grayscale method showed that the proposed method demonstrated better performance in malware classification, while the grayscale method performed better in malware detection [10].

VI. DATASET

In this section, we present the dataset used in our study, its composition, and the data preparation process for the malware classification task. The dataset consists of two primary sources: the BODMAS dataset and the DikeDataset Benign. We have meticulously collected and curated data from these sources to ensure its quality and relevance for our research.

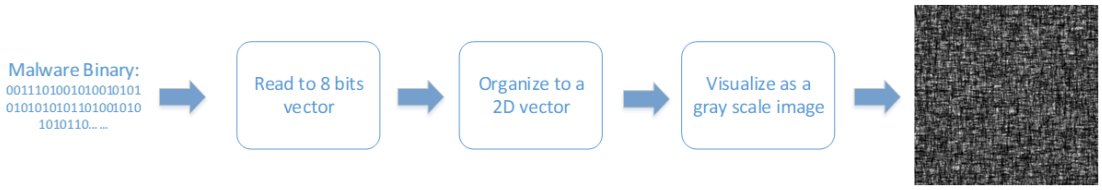

Fig. 1. Extract malware grayscale image feature from the binary.

A. *The BODMAS and DikeDataset dataset*

The BODMAS (Blue Hexagon Open Dataset for Malware Analysis) dataset is a comprehensive collection of 57,293 Windows PE files, including both disarmed malware binaries and their corresponding metadata. From this extensive dataset, we have carefully selected a subset of 24,813 executable files, representing 14 different categories of malware. The categories include 'trojan,' 'worm,' 'backdoor,' 'dropper,' 'ransomware,' 'pua,' 'downloader,' 'virus,' 'cryptominer,' 'informationstealer,' 'exploit,' 'rootkit,' 'p2p-worm,' and 'trojan-gamethief.' [1]

The DikeDataset [14] plays a pivotal role in our study, as it provides us with samples of benign executable files, including both PE (Portable Executable) and OLE (Object Linking and Embedding) files. This dataset consists of 1,083 samples representing benign executables, which serve as non-malicious samples in our research. By incorporating the DikeDataset, we are able to complement our analysis of malware samples with benign files, facilitating a comprehensive evaluation of our proposed approach.

B. *Data Preparation and Grayscale Conversion*

To facilitate image-based analysis for malware classification, we converted the extracted malware binaries into grayscale images. The data preparation process shown in the Fig.1 involved reading the binary content of each executable file and reshaping it into a 2D vector representation. We then transformed this 2D vector into a grayscale image using the Python Imaging Library (PIL). The grayscale images serve as a concise and informative representation of the malware samples, retaining essential features for classification.

C. *Principal Component Analysis*

Given the variability in file sizes between malware and benign samples in terms of binary data, we opted to utilize Principal Component Analysis (PCA) [15] as a pivotal technique. Our objective was to efficiently compress the grayscale images and derive latent feature representations from the malware images sourced from both the BODMAS dataset and the DikeDataset. PCA serves as a powerful tool for dimensionality reduction in this context. It endeavors to curtail the dimensionality of the images while preserving the utmost essential information indispensable for the precise classification process. The practical execution of PCA involved the following sequential steps:

1) Data Loading and Grayscale Conversion: We began by loading the grayscale images obtained from the previous data preparation stage. These images were represented as 2D arrays, where each pixel's intensity value corresponded to a grayscale value ranging from 0 (black) to 255 (white). The grayscale images were then converted to 2D vectors using the numpy library to facilitate PCA analysis.

2) Applying Principal Component Analysis: For PCA analysis, we used the scikit-learn library, which provides a simple and efficient implementation of PCA. In particular, we set the number of principal components to 1, as we aimed to capture the most dominant direction of variance in each image, effectively compressing the data into a single row. This compression allowed us to retain the essential information necessary for classification while significantly reducing the image's dimensionality.

3) Determining the Variance Threshold: After performing PCA on each grayscale image, we calculated

the explained variance ratio. The explained variance ratio indicates the proportion of the dataset's total variance explained by each principal component. We then computed the cumulative variance as the sum of explained variances in descending order. By setting a variance threshold of 0.95, we selected the minimum number of principal components required to retain at least 95% of the variance. This threshold allowed us to strike a balance between reducing dimensionality and preserving crucial information.

4) Compressing the Images: With the variance threshold determined, we compressed the images by projecting them onto the selected principal components. The compression process involved transforming the original 2D vector representation of the grayscale image into a lower-dimensional vector, effectively representing each image in a more compact form.

5) Resizing the Images: Finally, we resized the compressed images to a target size of 512x512 pixels to maintain consistency in the input dimensions for our deep learning models.

By conducting Principal Component Analysis on the grayscale images of both malware and benign samples, we achieved an efficient and informative representation of the data. The compressed images, with significantly reduced dimensionality, will serve as the fundamental input for our deep learning-based classification model. These compressed grayscale images will allow our model to distinguish and classify diverse malware types effectively, enabling us to proceed to the subsequent stages of our research with a concise and representative dataset.

D. Data Selection for Multiclass Classification

Due to the significant class imbalance in the original dataset, where some categories had very few samples, we decided to focus on a subset of classes for our multiclass classification problem. Our goal was to ensure a balanced representation of classes while maintaining a sufficient number of samples for effective model training.

To address this issue, we selected seven major categories from the BODMAS and DikeDataset subsets for our multiclass classification. These selected categories include 'trojan,' 'worm,' 'backdoor,' 'dropper,' 'ransomware,' 'downloader,' and 'virus.' These categories were chosen based on their prevalence within the dataset and their representation of diverse malware types. Additionally, we incorporated the 'benign' category as the eighth class in our multiclass classification problem. The benign category consists of grayscale images extracted from non-malicious executable files obtained from the DikeDataset. The selected classes include 'trojan,' 'worm,' 'backdoor,' 'dropper,' 'ransomware,' 'downloader,' 'virus,' and 'benign.' To summarize, by performing data selection after PCA, we aimed to strike a balance between class representation and sample count, to ensure multiclass classification model is capable of effectively distinguishing between various malware types, including benign files.

E. Data Augmentation

After performing the data selection process, we observed the class imbalance issue is still existing in the dataset, where some categories had a large number of samples while others had very few. To address this issue, we utilized data augmentation [16] techniques to increase the number of samples in each class and create a more balanced dataset. The Keras ImageDataGenerator was utilized to apply various transformations such as rotation, width and height shift, shear, zoom, and horizontal flip to generate augmented images from each original image, to the images while preserving their underlying characteristics. The augmentation was carried out differently for each class based on the initial number of samples and the desired number of samples to achieve a more balanced dataset.

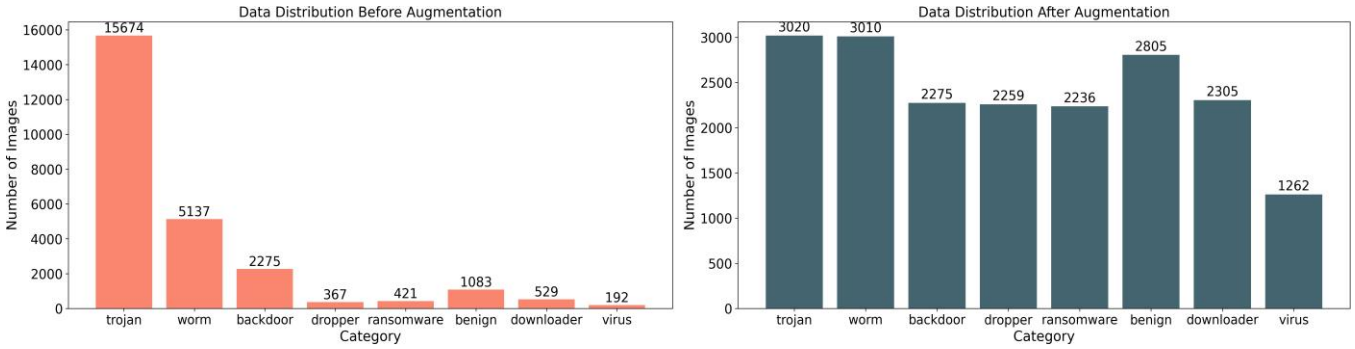

Fig. 2. Data distribution before and after augmentation, respectively.

Following the augmentation process, the data distribution in each class became more equitable, enhancing the representation of the different malware types. The number of images in each class after augmentation is as follows: trojan (3020 images), worm (3010 images), backdoor (2275 images), dropper (2259 images), ransomware (2236 images), benign (2805 images), downloader (2305 images), and virus (1262 images). By increasing the number of samples, we have created a balanced dataset that facilitates effective training of our multiclass classification model. After the augmentation process, the dataset achieves a heightened balance, rendering it well-suited for effective training of a multiclass classification model. The augmentation significantly bolsters the sample count within each class, effectively addressing the challenge of class imbalance and promoting a more comprehensive understanding of each class's distinctive features during the training phase.

Fig. 2 visually illustrates the transformation in data distribution both before and after the augmentation process. As evident from Fig. 2, the augmentation procedure has notably augmented the sample counts within each class, yielding a substantially more balanced representation of the various malware categories. This equilibrium within the dataset holds significance when training a multiclass classification model, as it ensures the model's ability to effectively learn from a diverse array of examples and make precise predictions when faced with unseen data. The augmented dataset facilitates a more holistic comprehension of each class, empowering the model to capture an extended spectrum of features and patterns. Consequently, this leads to enhanced performance in our malware classification task.

This augmented dataset, now well-balanced and enriched, lays foundation for our subsequent deep learning-based malware classification model. The augmentation process has significantly fortified the sample counts within each class, effectively mitigating the class imbalance challenge and fostering a more comprehensive understanding of each class's unique features during the model's training phase.

In summary, the process of converting malware binaries into grayscale images, followed by Principal Component Analysis (PCA), data selection, and strategic augmentation, has led to the creation of a meticulously curated dataset. This dataset includes comprehensive grayscale images of both malware and benign files, thoughtfully chosen and augmented to rectify class imbalances and offer a comprehensive representation of various malware types. The resulting dataset serves as a robust and invaluable asset, equipping our model to effectively differentiate between diverse malware categories and benign files. This groundwork lays the foundation for our subsequent endeavors in training a highly accurate and efficient malware classification model, holding the potential to drive significant advancements in the field of cybersecurity.

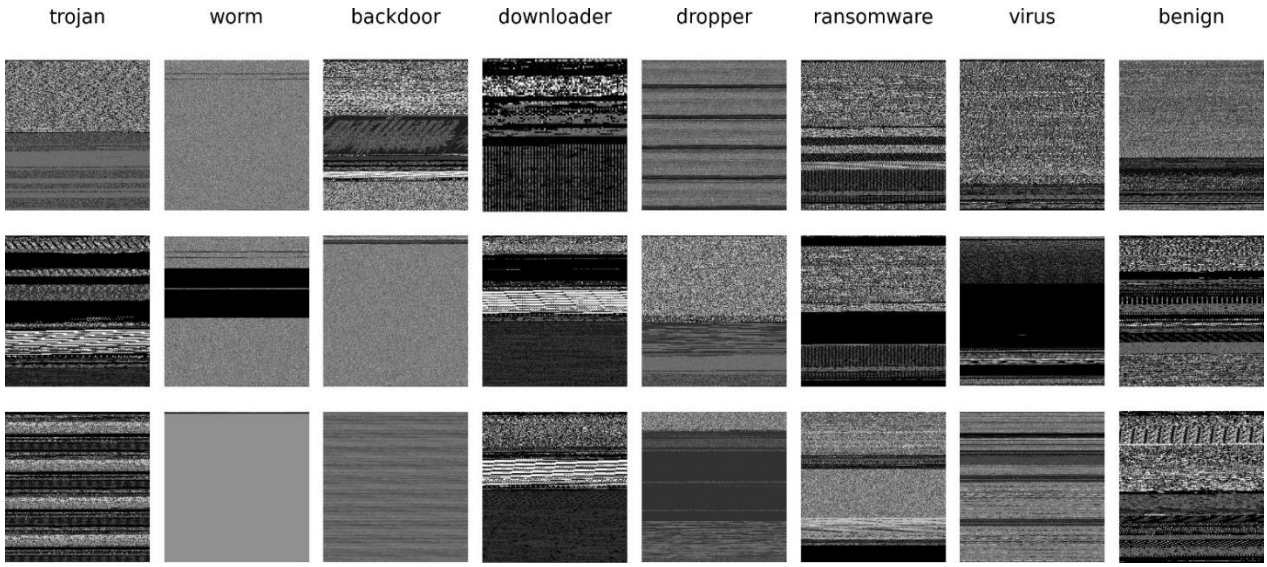

Fig 3. Malware and benign images used in this work.

VII. Proposed Methodology

In this section, we outline the proposed methodology for training the multiclass malware classification model using deep learning techniques. The methodology encompasses data preprocessing, model architecture selection, model training, and evaluation.

A. Data Collection and Preprocessing

As discussed in Section III, The dataset used in this study consists of grayscale images of both malware and benign samples from the BODMAS dataset and the DikeDataset, respectively. The dataset has been carefully curated to include diverse categories of malware, such as 'trojan,' 'worm,' 'backdoor,' 'dropper,' 'ransomware,' 'downloader,' 'virus,' and 'benign' as shown in Fig. 3. To address class imbalance, a data augmentation process was thoughtfully applied, thereby bolstering the sample counts within each class.

The augmented and preprocessed dataset is split into training, validation, and testing sets in the ratio of 60%, 20%, and 20%, respectively. The data splitting ensures that the model is trained on a large portion of the dataset, validated on a separate subset for hyperparameter tuning, and tested on unseen data to evaluate its generalization performance. The dataset distribution after splitting is as follows: Training Set contains 11501 images belonging to 8 classes. Validation Set contains 3835 images belonging to 8 classes. Test Set contains 3836 images belonging to 8 classes. Before feeding the data into the deep learning model, we perform data normalization to rescale the pixel values to the range [0, 1]. The normalization process involves dividing each pixel value by 255, the maximum pixel value in grayscale images. By doing so, we ensure that the pixel values are in a standardized range, allowing the model to converge faster during training and preventing any feature from dominating the learning process.

B. Model Building

The primary objective of our study is to develop accurate and efficient models capable of distinguishing different types of malwares. This core process is visually depicted in Fig. 4, providing a sequential representation of the stages from raw malware images to the conclusive classification results. A comprehensive examination of each element within this process follows, offering insights into our approach and the distinct deep learning models we have employed.

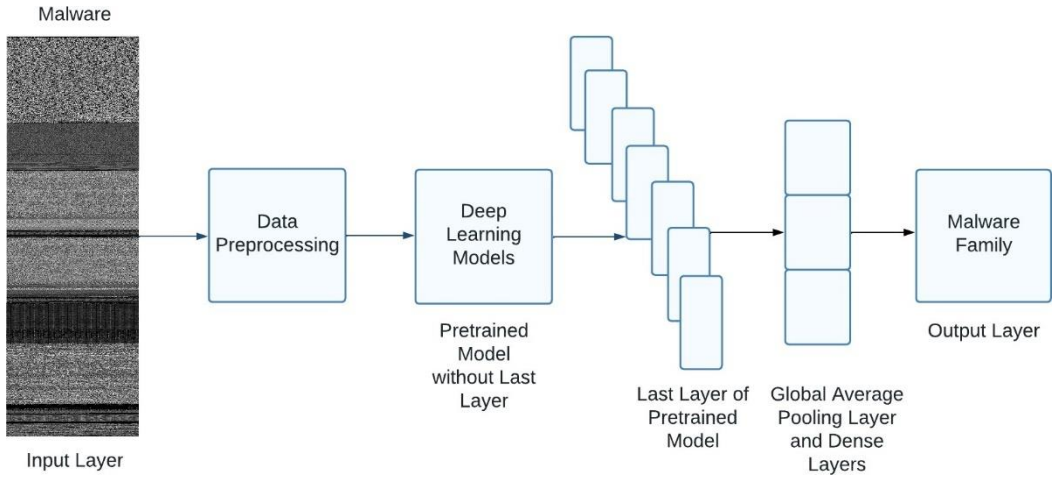

Fig. 4. Overview of proposed model for the malware classification.

The Malware Images (Input Layer) represent the unprocessed grayscale images of malware samples, serving as the initial input data for our classification models. Prior to feeding these images into our deep learning models, a sequence of data preprocessing steps is meticulously applied. These steps encompass operations such as pixel value normalization and the transformation of grayscale images into an RGB-like representation by replicating the single-channel grayscale image three times.

The Deep Learning Model (Pretrained Model without the Last Layer) serves as the cornerstone architecture of our deep learning models. We leverage pretrained models, including but not limited to VGG16, InceptionV3, EfficientNetV2B0, and Vision Transformers (ViT B32), with the final classification layer removed. These pretrained models have already gleaned valuable features from extensive datasets like ImageNet.

The Last Layer of the Pretrained Model plays a pivotal role in generating predictions based on the features extracted by the preceding layers. Typically, it includes a softmax activation function that furnishes class probabilities for various malware categories. After the feature extraction process from the pretrained model, these features undergo Global Average Pooling 2D, which serves to reduce the spatial dimensions of the feature maps and yield a fixed-length feature vector. This feature vector is subsequently linked to a fully connected Dense layer with ReLU activation. Ultimately, the output of the model provides a prediction regarding the malware family to which the input image belongs, thereby concluding the classification process.

Throughout our study, we have undertaken experimentation with an array of architectural models, each aligned with a distinct deep learning approach.

1) Customized CNN Model: Our first approach involves building a customized Convolutional Neural Network (CNN) model specifically tailored for malware classification. The architecture consists of three convolutional layers with max-pooling followed by fully connected layers. The model aims to learn hierarchical features from the input grayscale images to classify malware into distinct categories. The CNN model was trained for 25 epochs using the Adam optimizer with a learning rate of 0.001. To mitigate overfitting, we applied a dropout rate of 20% to the first fully connected layer.

2) VGG16 Transfer Learning Model: Our second approach involved using the VGG16 architecture for transfer learning to enhance the malware classification task [11]. The VGG16 model was pre-trained on the ImageNet dataset, making it adept at recognizing complex features in images. To adapt the VGG16 model for grayscale images, we replicated the single-channel grayscale image to create an RGB-like representation as mentioned previously. The VGG16-based model comprised two parts: the frozen base

model and the classification head. We froze the base model's layers to retain the knowledge gained from ImageNet [12] pre-training, allowing the model to focus on learning malware-specific patterns.

The architecture of the VGG16-based model starts with a Lambda layer to concatenate the grayscale image channel three times, effectively creating an RGB representation. Next, the frozen VGG16 base model followed, extracting relevant features from the input images. A Flatten layer converted the output to a 1D tensor, which was then connected to a fully connected Dense layer with 128 units, using the ReLU activation function. To prevent overfitting, we added a Dropout layer with a 0.3 rate before the final Dense layer. The output Dense layer employed the softmax activation function to provide class probabilities. The model was compiled with categorical cross-entropy loss and optimized using the Adam optimizer. During training, the model underwent 50 epochs.

3) InceptionV3 Model: Our third approach involved utilizing transfer learning with the InceptionV3 architecture, a powerful deep learning model that has been pre-trained on the ImageNet dataset. The InceptionV3 model was originally designed for RGB images, but to accommodate our grayscale images. Again, we replicated the single-channel grayscale image three times, effectively creating an RGB-like representation. Simialr, the InceptionV3-based model also consists of two main parts: the frozen base model and the classification head. By freezing the base model's layers, we retained the valuable knowledge gained from ImageNet pre-training. This enabled the model to focus on learning malware-specific patterns, leading to improved classification accuracy.

The model architecture begins with a Lambda layer to concatenate the grayscale image channel three times, transforming it into an RGB representation. Next, the frozen InceptionV3 base model extracts relevant features from the input images. A GlobalAveragePooling2D layer is used to reduce the spatial dimensions of the output and obtain a fixed-length feature vector. This feature vector is then passed through a fully connected Dense layer with 128 units and the ReLU activation function. To prevent overfitting, a Dropout layer with a rate of 0.2 was introduced before the final Dense layer. The output Dense layer employs the softmax activation function to provide class probabilities for the different types of malware. The model was compiled using categorical cross-entropy loss and optimized with the Adam optimizer. During training, the model underwent 30 epochs.

4) EfficientNet: For our fourth approach, we employed transfer learning with the EfficientNetV2B0 architecture, a state-of-the-art deep learning model known for its exceptional performance in computer vision tasks. While EfficientNetV2B0 was originally designed for RGB images, we adapted it to grayscale images by replicating the single-channel grayscale image three times to create an RGB-like representation as done before.

In the configuration of the EfficientNetV2B0-based model, we used a Sequential model. The input grayscale images were transformed into RGB representations using a Lambda layer followed by a Conv2D layer with three filters and a ReLU activation function. The EfficientNetV2B0 base model was then integrated, with 255 layers frozen to retain the valuable knowledge obtained from pre-training on the ImageNet dataset. By doing so, the model focused on learning malware-specific features, leading to improved classification accuracy.

The model architecture further consisted of a GlobalAveragePooling2D layer, which reduced the spatial dimensions of the output to obtain a fixed-length feature vector. This feature vector was connected to a fully connected Dense layer with 128 units and the ReLU activation function. To prevent overfitting, we introduced a Dropout layer with a rate of 0.3 before the final Dense layer. The output Dense layer utilized the softmax activation function to provide class probabilities for the different types of malware. The model was compiled with categorical cross-entropy loss and optimized using the SGD (Stochastic Gradient Descent) optimizer. During training, the model underwent 30 epochs to achieve convergence.

5) Vision Transformers: In our pursuit of achieving precise and efficient malware classification models, we delved into the domain of Vision Transformers (ViT), leveraging the potency of a ViT model with a b32 configuration [13]. ViT represents a state-of-the-art deep learning architecture widely acknowledged for its

exceptional performance in the domain of computer vision tasks.

The architectural design of our ViT model was meticulously curated, with careful consideration given to grayscale image inputs, each having dimensions of 512x512 pixels. To enable seamless processing, we conducted a preprocessing step that thoughtfully transformed single-channel grayscale images into a more intricate three-channel representation.

The ViT-based framework featured a well-structured ensemble of essential components, encompassing input preprocessing, the robust Vision Transformer backbone, critical flattening layers, and fully connected layers. A holistic approach to training and optimization was adopted, incorporating the Rectified Adam optimizer and categorical cross-entropy loss with label smoothing. To uphold training stability and combat overfitting, a suite of diverse callbacks were seamlessly integrated into the training process for 25-epoch.

C. Evaluation Metrics

For the multi-class classification of malware, the evaluation metrics used to assess the performance of the deep learning models are as follows:

1) Accuracy: Accuracy measures the proportion of correctly classified samples out of the total number of samples in the dataset. It is a widely used metric for classification tasks and provides an overall indication of how well the model performs in classifying different types of malware.

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN}$$

2) Precision: Precision measures the proportion of correctly predicted instances of a specific malware class out of all instances that were classified as that class by the model. It is calculated by dividing the number of true positive predictions (correctly classified instances of a particular class) by the sum of true positive and false positive predictions for that class.

$$Precision = \frac{TP}{TP + FP}$$

3) Recall: Recall, also known as sensitivity, measures the proportion of correctly predicted instances of a specific malware class out of all actual instances of that class in the dataset. It is calculated by dividing number of true positive predictions for that class by the sum of true positive and false negative predictions for that class.

$$Recall = \frac{TP}{TP + FN}$$

4) F1 Score: The F1-score is the harmonic mean of precision and recall.

$$F1\ Score = \frac{2 \times Precision \times Recall}{Precision + Recall}$$

where, TP (True Positive): The number of correctly predicted instances of a specific malware family, indicating the model's ability to accurately classify that specific type of malware.

FP (False Positive): The instances where the model incorrectly classifies samples as a certain malware family when they belong to a different class or benign.

TN (True Negative): The number of correctly identified samples that do not belong to the predicted class.

FN (False Negative): The instances where the model incorrectly classifies samples as benign or other malware families when they actually belong to the predicted class.

By analyzing these metrics for each malware family, we can evaluate the model's accuracy, precision, recall, F1-score, and other performance measures for the multi-class classification task. This thorough evaluation process will provide valuable insights into the model's ability to distinguish between different types of malware and its overall effectiveness in classifying diverse categories of malware accurately.

VIII. Experimental Results

In this section, we present the results achieved from our experimentation on malware classification using various deep learning models. The models were trained, validated, and tested on a carefully curated dataset encompassed grayscale images of both malware and benign samples.

Table I offers a comprehensive overview of performance metrics, encompassing accuracy, precision, recall, and F1-score, for each of the evaluated models. These experimental results provide crucial insights into the performance of each model. Notably, the CNN, VGG16, InceptionV3, Efficientnet, and ViTB32

TABLE I
Peformance Metrics of Each Model

Model	Accuracy	Score	Precision	Recall	F1 Score
CNN	0.78	Macro	0.77	0.76	0.77
		Weighted	0.78	0.78	0.78
VGG16	0.82	Macro	0.83	0.8	0.8
		Weighted	0.83	0.82	0.82
InceptionV3	0.9	Macro	0.9	0.9	0.9
		Weighted	0.9	0.9	0.9
Efficientnet V2B0	0.93	Macro	0.93	0.93	0.93
		Weighted	0.93	0.93	0.93
VisionTransformers Vit B32	**0.95**	**Macro**	**0.96**	**0.96**	**0.96**
		Weighted	**0.95**	**0.95**	**0.95**

models achieved accuracies of 78%, 82%, 90%, 93%, and 95%, respectively. Amongst all the models, ViTB32 demonstrated the highest Macro (Weighted) precision, recall, and F1-score, achieving 96% (95%), 96% (95%), and 96% (95%), respectively. This establishes ViTB32 as the best choice for malware classification within the scope of our selected dataset.

A. Comprehensive Analysis of the VGG16 Model

Table II provides an overview of the classification results achieved by the VGG16 Transfer Learning Model across various malware categories, highlighting its considerable potential in the field of malware identification. Notably, the model exhibited exceptional performance in backdoor and ransomware detection, boasting precision and recall values of 0.90 and 0.95, respectively, for these specific categories. These results underscore the VGG16 model's proficiency in harnessing pre-trained knowledge and capturing intricate features, rendering it a valuable asset in our quest to develop a precise malware classification model.

Nevertheless, it is crucial to acknowledge the scope for further improvement. The model's performance in categories such as viruses, where both precision and recall were comparatively lower, suggests the need for ongoing refinement. One avenue worth exploring is extending the training process by running the model for more epochs and fine-tuning its parameters. These strategies hold the potential to enhance its identification capabilities across a broader spectrum of malware categories, solidifying its position as a valuable model in our continuous efforts to achieve accurate and robust malware classification.

TABLE II
CLASSIFICATION RESULTS ON EACH CATEGORY OF MALWARE WHEN CLASSIFIED BY VGG16

Malware Family	Precision	Recall	F1-Score
Backdoor	0.90	0.90	0.90
Downloader	0.79	0.82	0.80
Dropper	0.72	0.80	0.76
Ransomware	0.95	0.87	0.91
Trojan	0.87	0.82	0.84
Virus	0.80	0.43	0.56
Worm	0.84	0.93	0.88
Benign	0.73	0.81	0.76

B. Comprehensive Analysis of the InceptionV3 Model

Table III provides a detailed breakdown of precision, recall, and F1-score metrics for each malware family, as classified by the InceptionV3 model following 5-Fold Cross Validation. The InceptionV3 model showcases robust performance across various malware categories, attaining notably high precision and recall scores for backdoor, dropper, and ransomware families. These results indicate its effectiveness in accurately categorizing these malware types, highlighting its capacity to generalize effectively across a diverse array of malware families.

While certain categories, such as trojan and virus, exhibit slightly lower precision scores in comparison to other malware categories, the model consistently exhibits exceptional performance in distinguishing between various types of malware. These findings underscore the model's proficiency in addressing a wide spectrum of malware families, emphasizing its ability to leverage pre-trained knowledge and capture intricate features, thus substantiating its effectiveness in the malware classification task.

Although the InceptionV3 transfer learning model demonstrates remarkable capabilities, as with any model, there may be opportunities for further fine-tuning or exploration of alternative architectures to potentially enhance its capabilities even further.

TABLE III
MALWARE FAMILY CLASSIFICATION METRICS WHEN CLASSIFIED BY INCEPTIONV3

Malware Family	Precision	Recall	F1-Score
Backdoor	0.96	0.91	0.93
Downloader	0.92	0.86	0.89
Dropper	0.92	0.90	0.91
Ransomware	0.98	0.95	0.96
Trojan	0.85	0.87	0.86
Virus	0.86	0.86	0.86
Worm	0.87	0.92	0.90
Benign	0.85	0.90	0.88

C. Comprehensive Analysis of the EfficientNet Model

Table IV presents a comprehensive overview of precision, recall, and F1-score metrics for each malware family when classified by the EfficientNetV2B0 model. This model showcases exceptional performance in effectively discriminating among various types of malware, as indicated by consistently high precision and recall scores across most categories. It is worth highlighting that the EfficientNetV2B0 model was trained with 255 frozen layers, a factor contributing to its robust classification capabilities.

However, Table IV also reveals that the EfficientNetV2B0 model exhibits lower recall and F1-score values specifically for the trojan malware category. Although it maintains high precision, it appears to face challenges in accurately identifying instances of trojan malware, resulting in reduced recall and F1- score values for this particular class. Further investigation, along with potential adjustments to the model's parameters or architecture, may offer avenues for improving its performance in correctly classifying trojan malware.

TABLE IV
MALWARE FAMILY CLASSIFICATION METRICS WHEN CLASSIFIED BY EFFICIENTNETV2B0

Malware Family	Precision	Recall	F1-Score
Backdoor	0.92	0.93	0.93
Downloader	0.97	0.93	0.95
Dropper	0.95	0.97	0.96
Ransomware	0.98	0.96	0.97
Trojan	0.92	0.85	0.89
Virus	0.91	0.91	0.91
Worm	0.87	0.94	0.94
Benign	0.94	0.94	0.94

D. Comprehensive Analysis of the Vision Transformer (ViT) Model

Table V provides a comprehensive breakdown of precision, recall, and F1-score metrics for each malware family, as classified by the Vision Transformers (ViT) B32 model. This model demonstrates exceptional proficiency in accurately categorizing various malware families, particularly excelling in precision and recall for families such as backdoor, downloader, dropper, and ransomware.

However, it encounters challenges when classifying trojan malware, resulting in lower recall and F1- score values for this specific category. Additionally, there is a slight decrease in precision when handling worm malware. Nevertheless, the ViT B32 model consistently delivers robust performance in distinguishing between viruses and benign samples.

Overall, the ViT B32 model impressively distinguishes different malware types but may benefit from further refinement or exploration of alternative approaches to enhance its performance, especially in accurately classifying trojan and worm malware. The ViT B32 model consistently exhibits superior classification capabilities across a diverse array of malware types within our selected dataset, with notable strengths in several categories.

TABLE V
CLASSIFICATION RESULTS ON EACH CATEGORY OF MALWARE WHEN CLASSIFIED BY VISION TRANSFORMERS VIT B32

Malware Family	Precision	Recall	F1-Score
Backdoor	0.94	0.94	0.94
Downloader	0.98	0.98	0.98
Dropper	0.97	0.99	0.98
Ransomware	0.99	0.99	0.99
Trojan	0.93	0.86	0.89
Virus	1.00	0.97	0.98
Worm	0.88	0.96	0.92
Benign	0.99	0.97	0.98

IX. Conclusion

This paper introduces" DeepGray," a deep learning model that underscores the importance of grayscale image representations in the field of malware classification. Our extensive investigation reveals how grayscale images play a pivotal role in capturing essential malware characteristics, enabling deep learning systems to effectively discern intricate malicious patterns.

We conducted experiments involving several deep learning models, including VGG16, InceptionV3, EfficientNetV2B0, Vision Transformers ViT B32, and a custom CNN model, for malware classification. The results demonstrate that ViT B32, InceptionV3, and EfficientNetV2B0 outperformed other models. Among these, the Vision Transformers model achieved the highest accuracy and demonstrated strong classification capabilities across various malware families. Leveraging transfer learning with these models significantly improved classification performance. Moreover, our findings highlight the potential of deep learning in enhancing malware classification, particularly when combined with grayscale image analysis. This fusion offers substantial promise for strengthening cybersecurity measures and enhancing malware detection accuracy.

Looking ahead, our future research will focus on refining deep learning models, particularly the custom CNN model, and expanding grayscale image analysis to further enhance our model's performance. This ongoing work will fortify our defenses against the ever-evolving landscape of malware threats, ultimately bolstering the protection of digital ecosystems.

References

[1] L. Yang, A. Ciptadi, I. Laziuk, A. Ahmadzadeh and G. Wang," BODMAS: An Open Dataset for Learning based Temporal Analysis of PE Malware," 2021 IEEE Security and Privacy Workshops (SPW), San Francisco, CA, USA, 2021, pp. 78-84, doi: 10.1109/SPW53761.2021.00020.

[2] Nataraj, Lakshmanan, Karthikeyan, Shanmugavadivel, Jacob, Gre´goire, Manjunath, B. (2011). Malware Images: Visualization and Automatic Classifica- tion. 10.1145/2016904.2016908.

[3] W. Hardy, L. Chen, S. Hou, Y. Ye, and X. Li, "Dl4md: A deep learning framework for intelligent malware detection," in Proceedings of the International Conference on Data Mining (DMIN), 2016.

[4] Pant, D., Bista, R. (2021). Image-based Malware Classification using Deep Convolutional Neural Network and Transfer Learning. In 2021 3rd International Conference on Advanced Information Science and System (AISS 2021) (pp. 1-6). November 26–28, 2021, Sanya, China. ACM, New York, NY, USA. DOI: 10.1145/3503047.3503081.

[5] M. Kalash, M. Rochan, N. Mohammed, N. D. B. Bruce, Y. Wang and F. Iqbal," Malware Classification with Deep Convolutional Neural Networks," 2018 9th IFIP International Conference on New Technologies, Mobility and Security (NTMS), Paris, France, 2018, pp. 1-5, doi: 10.1109/NTMS.2018.8328749.

[6] K. Deepa, K. S. Adithyakumar and P. Vinod," Malware Image Classification using VGG16," 2022 International Conference on Computing, Communication, Security and Intelligent Systems (IC3SIS), Kochi, India, 2022, pp. 1-6, doi: 10.1109/IC3SIS54991.2022.9885587.

[7] S. Shinde, A. Dhotarkar, D. Pajankar, K. Dhone and S. Babar," Malware Detection Using Efficientnet," 2023 International Conference on Emerging Smart Computing and Informatics (ESCI), Pune, India, 2023, pp. 1-6, doi: 10.1109/ESCI56872.2023.10099693.

[8] "Microsoft malware classification challenge (big 2015)," https://www.kaggle.com/c/malware-classifiation, 2017, accessed: 2017-01-30

[9] V. Acharya, V. Ravi and N. Mohammad," EfficientNet-based Convolutional Neural Networks for Malware Classification," 2021 12th Interna- tional Conference on Computing Communication and Networking Technologies (ICCCNT), Kharagpur, India, 2021, pp. 1-6, doi: 10.1109/ICC- CNT51525.2021.9579750.

[10] L. M. Kwan," Markov Image with Transfer Learning for Malware Detection and Classification," TENCON 2022 - 2022 IEEE Region 10 Conference (TENCON), Hong Kong, Hong Kong, 2022, pp. 1-6, doi: 10.1109/TENCON55691.2022.9977916.

[11] S. Albawi, T. A. Mohammed and S. Al-Zawi," Understanding of a convolutional neural network," 2017 International Conference on Engineering and Technology (ICET), Antalya, Turkey, 2017, pp. 1-6, doi: 10.1109/ICEngTechnol.2017.8308186.

[12] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens and Z. Wojna," Rethinking the Inception Architecture for Computer Vision," 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 2016, pp. 2818-2826,

doi: 10.1109/CVPR.2016.308.
[13] Alexey, Dosovitskiy., Lucas, Beyer., Alexander, Kolesnikov., Dirk, Weissenborn., Xiaohua, Zhai., Thomas, Unterthiner., Mostafa, Dehghani., Matthias, Minderer., Georg, Heigold., Sylvain, Gelly., Jakob, Uszkoreit., Neil, Houlsby. (2021). An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale.
[14] I. George-Andrei," DikeDataset", Github.com, [online] Available: https://github.com/iosifache/DikeDataset.
[15] Jolliffe Ian T. and Cadima Jorge, Principal component analysis: a review and recent developments, Phil. Trans. R. Soc. A. http://doi.org/10.1098/rsta.2015.0202.
[16] Luis Perez and Jason Wang, The Effectiveness of Data Augmentation in Image Classification using Deep Learning. CoRR abs/1712.04621 (2017)

INSTITUTE FOR HOMELAND SECURITY

Sam Houston State University

The Institute for Homeland Security at Sam Houston State University is focused on building strategic partnerships between public and private organizations through education and applied research ventures in the critical infrastructure sectors of Transportation, Energy, Chemical, Healthcare, and Public Health. The Institute is a center for strategic thought with the goal of contributing to the security, resilience, and business continuity of these sectors from a Texas Homeland Security perspective. This is accomplished by facilitating collaboration activities, offering education programs, and conducting research to enhance the skills of practitioners specific to natural and human caused Homeland Security events.

Institute for Homeland Security
Sam Houston State University

Polsani, H., Jiang, H. (2023) DeepGray: A Novel Approach to Malware Classification Using Grayscale Images with Deep Learning. (Report No. IHS/CR-2023-1014). The Sam Houston State University Institute for Homeland Security.
https://doi.org/10.17605/OSF.IO/EADBM