electronics

MDPI

Article

Deep Learning for Cybersecurity Classification: Utilizing Depth-Wise CNN and Attention Mechanism on VM-Obfuscated Data

Sicheng Han, Heeheon Yun and Yongsu Park *

Department of Computer Science, Hanyang University, Wangshimriro 222, Seongdong-gu, Seoul 04763, Republic of Korea
* Correspondence: yongsu@hanyang.ac.kr

Abstract: With the increasing use of sophisticated obfuscation techniques, malware detection remains a critical challenge in cybersecurity. This paper introduces a novel deep learning approach to classify malware obfuscated by virtual machine (VM) code. We specifically explore the application of depth-wise convolutional neural networks (CNNs) combined with a spatial attention mechanism to tackle VM-protected cybersecurity datasets. To address the scarcity of obfuscated malware samples, the dataset was generated using VMProtect to ensure the models were trained on real examples of modern obfuscated malware. The effectiveness of our approach is demonstrated through extensive experiments on both regular malware and obfuscated malware, where our model achieved accuracies of nearly 100% and 93.55% in classifying the regular malware and the obfuscated malware, respectively.

Keywords: malware detection; code obfuscation; depth-wise CNN; attention; cybersecurity

Citation: Han, S.; Yun, H.; Park, Y. Deep Learning for Cybersecurity Classification: Utilizing Depth-Wise CNN and Attention Mechanism on VM-Obfuscated Data. *Electronics* **2024**, *13*, 3393. https://doi.org/10.3390/electronics13173393

Academic Editors: Yuanzhang Li, Yu-an Tan, Chen Liang and Hongfei Zhu

Received: 20 July 2024
Revised: 17 August 2024
Accepted: 21 August 2024
Published: 26 August 2024

1. Introduction

Malware detection has become more challenging with the rapidly evolving domain of cybersecurity, compounded by the increasingly sophisticated evasion techniques employed by adversaries. Among these, virtual machine (VM) code obfuscation stands out as a particularly complex method of evasion. By transforming executable malware into forms which are harder to analyze while preserving their functionality, VM code obfuscation significantly boosts the effectiveness of traditional detection systems. This paper introduces a novel approach to tackling the intricate problem of detecting VM code-obfuscated malware, leveraging the synergistic capabilities of depth-wise convolutional neural networks (CNNs) enhanced with spatial attention mechanisms.

VM code obfuscation alters executable code into a format which, while operational, becomes more challenging for both human analysts and automated tools to interpret. This can effectively shield malware from detection by conventional antivirus systems, posing a significant risk to cybersecurity infrastructures. Our work addresses this challenge by developing a robust deep learning model which precisely discerns various types of obfuscated malware. We utilize VMProtect, a leading software tool for code obfuscation, to create a diverse dataset of malware images. Our dataset comprises 10 classes of obfuscated malware, each represented by at least 1000 images derived from executable files. Notably, these classes include high-profile malware families such as Dark Hotel, Energetic Bear, Equation Group, and Gorgon Group. Given the restricted availability of malware samples and the rarity of specific variants, we applied data augmentation techniques through ImageDataGenerator.

Our methodology employs depth-wise CNNs to exploit the spatial hierarchies in the image data, alongside an attention mechanism which concentrates on the informative features of the images. This combination allows for effective and efficient feature extraction, which is essential for the classification of intricately obfuscated malware. The attention

mechanism in particular enhances the model's ability to focus selectively on areas of the image most likely to contain obfuscation artifacts. The effectiveness of our approach is demonstrated by a notable classification success rate of 93.55%, achieved through meticulous training and validation of our model. This success sets a new benchmark in the field and highlights the potential of combining depth-wise convolutions with attention mechanisms to advance the state of the art in malware detection. To the best of our knowledge, this is the first attempt to use CNNs to classify VM-based, code-obfuscated malware. Our dataset and the proposed method are available at https://github.com/AHSCone/Mal_CNN (accessed on 15 July 2024).

This paper is organized as follows. Section 2 reviews the related literature on malware detection. Section 3 details the dataset, the architecture of our attention-enhanced CNN, and our data augmentation strategy. Section 4 describes our experimental set-up and the metrics used for evaluation. Section 5 discusses the results and outlines potential areas for future research.

The principal contributions of this paper are as follows:

1. We utilize the commercial software VMProtect, available online at https://vmpsoft.com/products/vmprotect/ (accessed on 15 July 2024) to generate an unprecedented dataset of VM code-obfuscated malware specifically tailored for our analysis.
2. Faced with limited access to diverse malware samples, we employ advanced data augmentation and image enhancement techniques to ensure a comprehensive dataset.
3. Our model integrates a depth-wise CNN with a spatial attention mechanism designed to effectively identify and classify a wide range of VM code-obfuscated malware.

2. Related Work

2.1. VM-Based Code Obfuscation

VM-based code obfuscation protects software by transforming executable code into a form that is difficult to analyze while maintaining its original functionality. This method provides a robust shield against reverse engineering and unauthorized modifications, which is crucial in maintaining software integrity and security. VMProtect exemplifies this approach by converting standard bytecode into a proprietary virtual instruction set, which is decipherable only with its custom virtual machine [1,2]. This conversion process involves multiple layers of transformation, each introducing additional complexity and making the deobfuscation process challenging for attackers.

Historically, both legitimate software protection and malware developers aiming to evade detection have widely adopted VM-based obfuscation. Studies have demonstrated that VM obfuscated code can significantly hinder the efforts of static and dynamic analysis techniques, typically requiring specialized tools and expertise for reversal [3,4].

Further research has targeted the detection and analysis of VM-based obfuscated malware, uncovering patterns and weaknesses exploitable by security experts. These studies often involve a combination of machine learning techniques and manual analysis to identify the characteristic features of obfuscated code [5,6].

2.2. CNN and Attention Mechanism

Convolutional neural networks (CNNs) have revolutionized image processing and classification by prioritizing the most informative image parts. This capability has established CNNs as the cornerstone of numerous sophisticated visual recognition systems, being extensively applied in diverse fields such as medical imaging and autonomous vehicle technology [7,8]. The integration of attention mechanisms has further propelled the capabilities of CNNs, thereby improving a model's performance and interpretation ability [9,10].

A significant development in the CNN architecture is the creation of MobileNets, which utilize depth-wise separable convolutions. This design significantly reduces the computational cost and model size, making them particularly suitable for mobile and embedded vision applications where resources are constrained [11]. Following this, Mo-

bileNetV2 introduced an inverted residual structure which uses lightweight depth-wise convolutions to filter features in the intermediate expansion layer, boosting accuracy and efficiency in mobile vision tasks [12].

In the context of malware detection, attention-based CNNs represent a pivotal enhancement. Wang et al. illustrated how incorporating attention mechanisms can substantially improve the reliability of malware detection systems [13]. These models intensively focus on regions potentially containing malicious content, enabling the detection of subtle anomalies which might elude conventional CNNs. Furthermore, the introduction of the transformer architecture by Vaswani et al. marked a significant evolution in sequence modeling [14]. Transformers utilize self-attention mechanisms to recognize dependencies within data regardless of the positional distance in the input sequence.

2.3. Obfuscated Malware Detection and Classification

Detecting and classifying obfuscated malware presents significant challenges due to the complexity and variety of obfuscation techniques designed to bypass standard detection methods [15,16]. Advances in deep learning have demonstrated significant potential in detecting obfuscated malware by deciphering complex patterns and anomalies which indicate malicious behavior [17–19]. Nataraj et al. pioneered the use of image representations of binary malware for analysis, noting that visual textures in these images can distinguish malicious software from benign software [20].

In our previous works, we used the classic CNN for symmetric key-encrypted malware detection [21]. Recent research has investigated hybrid approaches. Suarez-Tangil et al. combined CNNs with random forests to analyze malware images, devising a detection system which capitalizes on the strengths of both techniques [22]. Li et al. employed generative adversarial networks (GANs) to generate synthetic malware images for training, thus bolstering a CNN's robustness against emerging malware types [23].

Efforts to enhance model interpretability and resilience against adversarial attacks have included studies on neural network robustness [24] and training on adversarial examples to augment model generalization and fortify defense mechanisms [25]. Lei et al. proposed a transfer learning method with feature fusion to achieve software defect detection [26]. Furthermore, recent works used MobileNets for IoT malware detection with opcode features [27]. These developments represent a significant advancement in malware detection strategies, utilizing sophisticated CNN architectures and image processing technologies. This work seeks to build upon these innovations and advance the field of malicious software detection and analysis.

3. Methods

3.1. Dataset Generation with VMProtect

This section introduces the methodology to develop a specialized dataset designed for the classification of VM-based obfuscated malware, as illustrated in Figure 1. Our dataset includes 10 distinct malware classes, with each containing at least 1000 variant samples, which is crucial for training a robust model and ensuring a diverse range of data.

The initial stage in the dataset creation process involves using VMProtect to obfuscate executable malware files. VMProtect analyzes the binary code and employs multiple obfuscation techniques, including code virtualization. This process transforms the original code into a pseudo-code that is only executable by VMProtect's proprietary virtual machine, concealing the malware's true characteristics from analytical tools.

To manage the obfuscation process, an automated script sets up directories for each malwarre category, such as "Dark Hotel", "Energetic Bear", "Equation Group", "Gorgon Group", and "Winnti". We use scripts for generation and execute VMProtect's obfuscation commands within the Linux system's shell, also logging each operation to facilitate monitoring and troubleshooting.

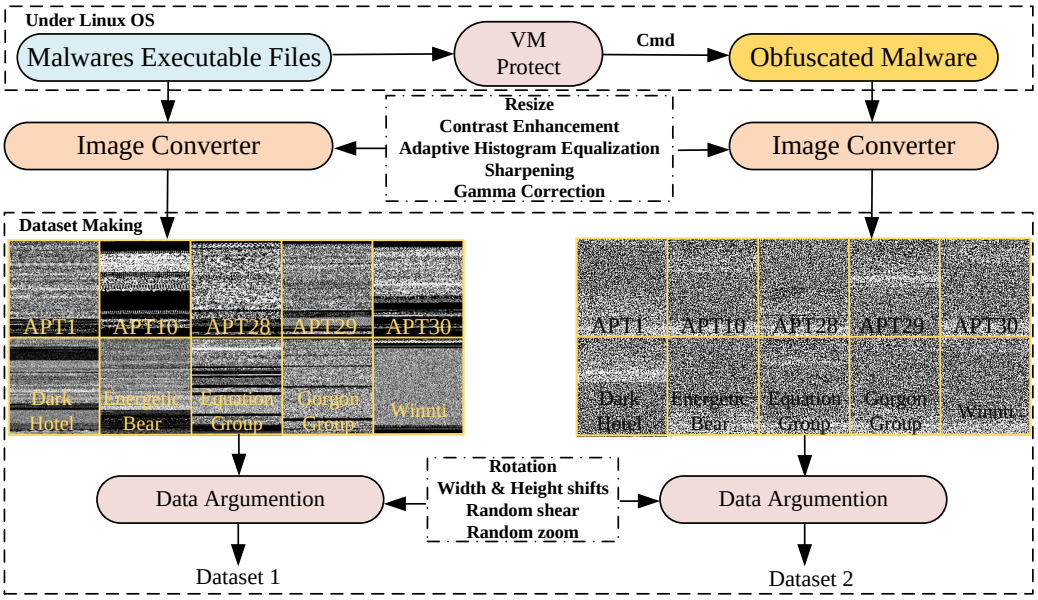


Figure 1. Illustration of the dataset generation procedure.

3.2. Image Conversion and Data Augmentation

Following obfuscation, binary files are converted into grayscale images. The conversion size s is determined as follows:

$$s = \max(\lceil \sqrt{L} \rceil, 128) \tag{1}$$

where L is the length of the binary file and the minimum dimension is set to 128 pixels. Padding is applied to ensure the binary data conform to $s \times s$ dimensions, reshaping the zero-padded array into a two-dimensional grid. These images are then standardized across the dataset, optimizing clarity and uniformity.

The images are subsequently enhanced to improve their analytical quality by employing several mathematical operations:

$$I' = (I - \mu) \times 3.0 + \mu \tag{2}$$

where I is the image intensity and μ is the mean intensity. This adjustment increases the contrast by a factor of three. Additionally, normalization scales the pixel values to a standard range of 0–255:

$$I_{\text{norm}} = \frac{I - \min(I)}{\max(I) - \min(I)} \times 255 \tag{3}$$

Adaptive histogram equalization is applied to enhance the contrast in localized areas:

$$I_{\text{eq}} = \text{equalize}(I) \tag{4}$$

Gamma correction is utilized to modify the luminance, and sharpness enhancement is achieved by increasing the edge definition:

$$I_{\gamma} = 255 \times \left(\frac{I}{255} \right)^{1.2} \tag{5}$$

$$I_{\text{sharp}} = \text{sharpen}(I, \text{factor} = 3.0) \tag{6}$$

Each processed image is then systematically archived and ready for model training.

To augment the diversity and volume of our dataset, we employed a script which enhanced the original set of images through various transformations. The script starts

by duplicating existing images if their count is below the desired threshold. Then, we apply transformations including rotation, width and height shifts, shear, zoom, and horizontal flipping during model training. This method ensures that the dataset is uniformly extensive and varied across all classes. Our goal was to create a dataset that was not only comprehensive but also meticulously optimized for the effective training of deep learning models.

3.3. Depth-Wise Convolutional Neural Networks with Spatial Attention Mechanism (Mal-CNN)

This subsection describes the architecture of the malware classification convolutional neural network (Mal-CNN), a hybrid deep learning model which synergistically integrates depth-wise CNNs with spatial attention mechanisms. Designed to enhance discriminatory power while maintaining computational efficiency, the Mal-CNN excels at processing complex image data derived from VM code-obfuscated malware, as depicted in Figure 2.

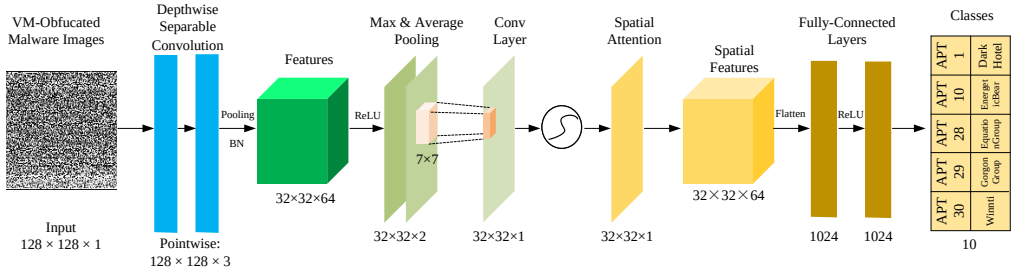


Figure 2. Our proposed Mal-CNN network structure.

Depth-wise convolution is an efficient adaptation of the standard convolution operation. Unlike traditional convolutions which combine inputs across all channels, depth-wise convolutions perform a separate convolution for each input channel, thereby significantly reducing both the computational load and the number of model parameters:

$$\text{Output}_k = \text{DepthConv}(\text{Input}_k) * \text{Filter}_k \tag{7}$$

Here, DepthConv signifies the depth-wise convolution applied to the kth channel of the input, Input_k, using the corresponding convolutional filter, Filter_k. Following this, a point-wise convolution aggregates these channel-wise outputs into comprehensive feature maps, effectively maintaining the network's capacity for sophisticated feature extraction.

By augmenting the efficiency of depth-wise convolutions, the Mal-CNN incorporates a spatial attention mechanism which selectively refines feature representations. This mechanism generates an attention map which highlights critical features for accurate classification while diminishing less important ones:

$$\text{Attention Map} = \sigma(\text{Conv}(\text{Concat}(\max(\text{Input}, \text{axis} = 3), \text{mean}(\text{Input}, \text{axis} = 3)))) \tag{8}$$

In this expression, σ represents the sigmoid activation function, Conv is a convolutional operation, and Concat denotes the concatenation of the channel-wise maximum and average of the input feature map. This focused mechanism directs the learning process toward the most informative elements within the feature map.

The integrated structure of the Mal-CNN optimizes both the spatial and channel-wise features of the input data:

$$\text{Enhanced Features} = \text{Attention Map} \odot \text{Depth-wise Convolved Features} \tag{9}$$

where $\odot$ indicates the element-wise multiplication of the depth-wise convolved features with the attention map, enhancing the model's responsiveness to critical features while diminishing the influence of irrelevant background noise.

The integration of depth-wise convolutions and spatial attention mechanisms positions the Mal-CNN as a significant advancement. Empirical evaluations highlight the Mal-CNN's effectiveness in precisely classifying VM code-obfuscated malware, showcasing superior performance and operational efficiency.

In the following Algorithm 1 N is the batch size, C is the channel count, and H and W are height and width, respectfully.

Algorithm 1 Depth-wise convolutional neural networks with spatial attention mechanism

Input: Image batch of size $N \times C \times H \times W$
Output: Class scores for each image in the batch
Initialize the network with the specified number of classes and input dimensions
$\text{DepthOutput}_k = \text{DepthConv}(\text{Input}_k) * \text{Filter}_k$
Apply point-wise convolution to combine channel outputs
$\text{Output} = \text{PointConv}(\text{DepthOutput})$
Calculate average and max pooling across channels
Apply convolution and sigmoid activation to obtain the attention map
$\text{Attention Map} = \sigma(\text{Conv}(\text{ConcatOutput}))$
Apply attention map to the original input by element-wise multiplication
$\text{Refined Output} = \text{Input} \odot \text{Attention Map}$
return Class probabilities for each image

4. Experiment and Decision

4.1. Dataset Description

The experimental evaluation utilized two distinct datasets to assess the efficacy of the proposed Mal-CNN model. The first dataset comprised images converted from original malware executable files, as depicted in Figure 3. The second dataset consisted of the images converted from the VM-obfuscated malware executable files (see Figure 4). These datasets were meticulously structured into 10 categories representative of various notable malware families. These categories were APT 1, APT 10, APT 28, APT 29, APT 30, Dark Hotel, Energetic Bear, Equation Group, Gorgon Group, and Winnti.

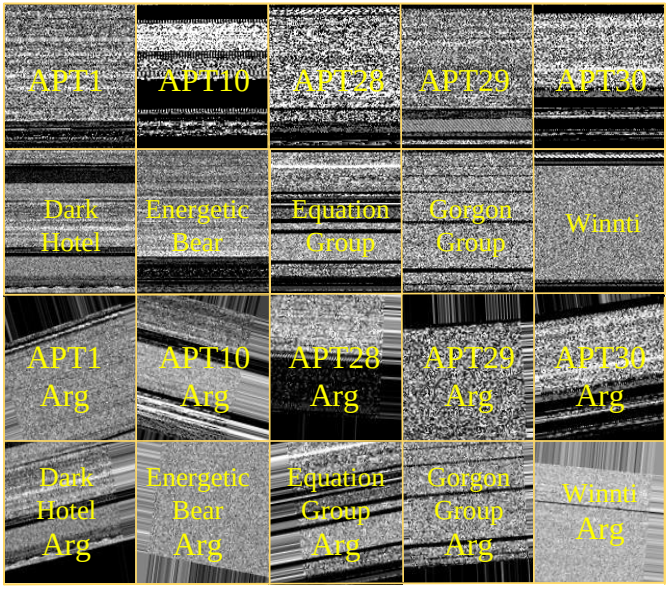


Figure 3. Dataset comprising original malware executable files, featuring a variety of malware families such as APT1, APT10, APT28, APT29, and APT30 enhanced through data augmentation to improve model training and robustness.

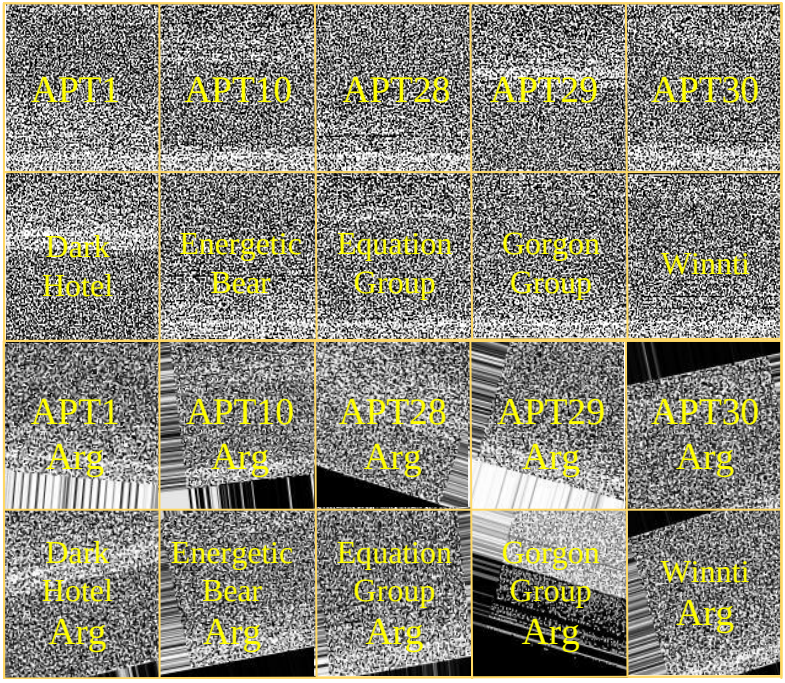


Figure 4. Dataset consisting of malware executable files processed through obfuscation techniques including VMProtect, encompassing malware families like APT1 and APT10, among others. This dataset was enhanced through data augmentation to improve model training and robustness.

Both datasets comprised a total of 10,000 images, with a division of 8000 images for training and 2000 for testing purposes. To ensure robustness and generalizability in the model's performance, each category within the datasets was augmented to contain at least 10,000 samples. This augmentation facilitated comprehensive training and testing, thereby enhancing the predictive capabilities of the Mal-CNN model in real-world scenarios.

The employment of two distinct datasets in our experimental framework served two purposes. The primary objective here was to assess the baseline performance of the model under controlled conditions with well-defined, non-obfuscated data. Moreover, the use of these datasets facilitated a direct and illustrative comparison between the feature representations of VM-obfuscated malware and their original counterparts.

4.2. Experimental Set-Up

The CNN architecture used in our experiments is detailed in the table below. The hyperparameters table (Table 1) outlines the specific settings used during the training phase of our model. The architecture of our CNN is detailed in the accompanying table, which specifies the optimizer, learning rate, batch size, and the choice of loss function, each of which is crucial for effective training. The reduction ratio for the spatial attention mechanism indicates the degree of compression applied in the attention layer to balance performance and computational cost. To mitigate the risk of overfitting, we incorporated dropout layers within the network to prevent co-adaptation of neurons by randomly dropping units during training. Additionally, we augmented the training dataset with transformations such as rotations and scaling to enhance the model's generalization capability. Early stopping was also employed to halt training once the improvement in model performance plateaued, ensuring the model did not continue to learn the idiosyncrasies of the training data. Without a dedicated validation set, hyperparameter tuning was approached by informally splitting the training set to allow for internal validation. A range of values for each hyperparameter was tested iteratively to find the optimal settings which resulted in the best performance on these informal validation trials.

Table 1. Training hyperparameters.

Parameter	Value
Optimizer	Adam
Learning Rate	0.0005
Batch Size	32
Number of Epochs	200
Loss Function	SGD
Reduction Ratio (Spatial Attention)	16

Additionally, the reduction ratio for the spatial attention mechanism specified the degree of compression in the attention layer, balancing performance and computational cost.

We present a structured overview of the Mal-CNN architecture specifically designed for effective malware detection (see Table 2). Each layer is described in terms of its type, input and output channels, kernel size, stride, and activation function.

Table 2. Detailed architecture of Mal-CNN.

Layer Type	Input	Output	Kernel Size	Stride	Activation
Depth-Wise Separable Conv.	1	32	3×3	1	ReLU
Batch Normalization	32	32	-	-	-
Max Pooling	-	-	2×2	2	-
Depth-Wise Separable Conv.	32	64	3×3	1	ReLU
Batch Normalization	64	64	-	-	-
Max Pooling	-	-	2×2	2	-
Depth-Wise Separable Conv.	64	128	3×3	1	ReLU
Batch Normalization	128	128	-	-	-
Max Pooling	-	-	2×2	2	-
Depth-Wise Separable Conv.	128	256	3×3	1	ReLU
Batch Normalization	256	256	-	-	-
Max Pooling	-	-	2×2	2	-
Spatial Attention	-	-	-	-	Sigmoid
Fully Connected	16,384	1024	-	-	ReLU
Fully Connected	1024	10	-	-	-

Figure 5 depicts a series of feature maps generated by the first convolutional layer of our CNN. Each map is a visual representation of the different filters' responses to a malware binary image. The variations in texture and intensity across these maps highlight the diverse aspects of the data which the network deemed significant for classification purposes. Some maps may show pronounced lines or stripes, possibly related to executable code segments or data definitions. In contrast, other maps appear more scattered and less uniform, suggesting areas where the binary structure might be employing obfuscation or encryption techniques to conceal its true nature.

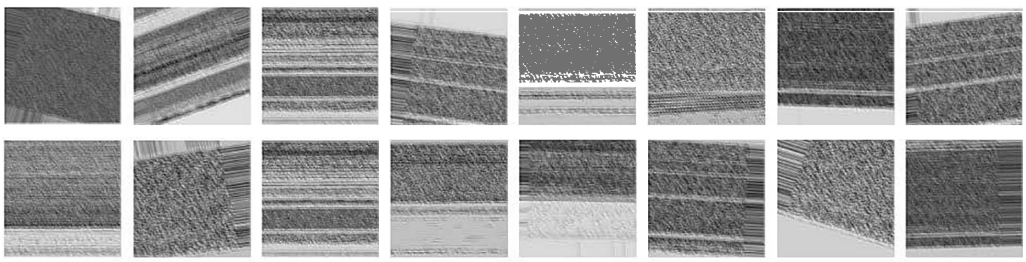


Figure 5. Feature maps extracted from a malware binary image by the first convolutional layer of the CNN.

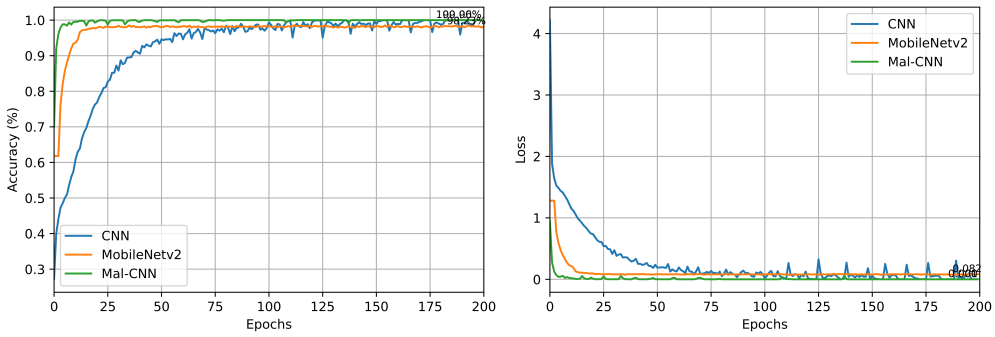
Furthermore, the brightness within these maps indicates regions with higher neural activity, where the CNN detected features strongly indicative of malicious content. Such features might include unusual binary structures, specific sequences of bytes, or anomalies

which differentiate malware from benign software. The ability of the CNN to detect these features and highlight them across various feature maps confirms the network's capacity to focus on pertinent aspects of the data, enhancing its effectiveness in malware detection.

4.3. Evaluation on Original Malware Dataset

The evaluation of the Mal-CNN model was meticulously performed using a dataset composed of images converted from original, non-obfuscated malware executables. This analysis involved a comparative assessment against a baseline CNN model which, while having a similar network depth, lacked the advanced features of depth-wise convolutional and spatial attention mechanisms present in the Mal-CNN. Furthermore, the Mal-CNN was also compared with MobileNetV2 [12], which also uses depth-wise convolutional layers.

The data reveal that the Mal-CNN achieved a training accuracy of 100.0%, indicative of its exceptional capability to capture and internalize the discriminative features inherent in the malware images (see Figure 6). Both the Mal-CNN and the baseline CNN exhibited similarly high training accuracies, evidencing that the inclusion of advanced architectural enhancements does not compromise the model's fit to the training data.



(**a**) (**b**)

Figure 6. Learning performance of the Mal-CNN on the original malware dataset, demonstrating the model's capacity for effective feature extraction and robust learning. (**a**) Learning curve for the training accuracy on the original dataset. (**b**) Learning curve for the training losses on the original dataset.

However, significant differences emerged in the testing phase. The Mal-CNN recorded a testing accuracy of 98.39% on the original dataset, substantially surpassing the baseline CNN, which achieved an accuracy of only 91.74% (see Figure 7). This disparity highlights the advantages conferred by the depth-wise convolutional layers and spatial attention mechanisms, which proves the Mal-CNN's generalization capabilities compared with a standard CNN. MobileNetV2 achieved an accuracy of 96.90%, higher than that of the CNN but slightly lower than that of the Mal-CNN. This phenomenon demonstrates the effectiveness of using depth-wise convolutional layers for regular malware detection. The marked improvement in testing accuracy reflects the Mal-CNN's superior ability to adapt and perform under varied conditions, affirming the architectural enhancements as crucial for dealing with complex real-world data.

Analysis of the features extracted by the CNN from malware binary images revealed efficient learning of distinctive textual patterns and shapes associated with different malware types. For instance, specific gradient patterns and texture changes indicative of malicious code were successfully recognized. The depth-wise convolutions helped capture nuanced variations within the malware images, which are often overlooked by simpler architectures.

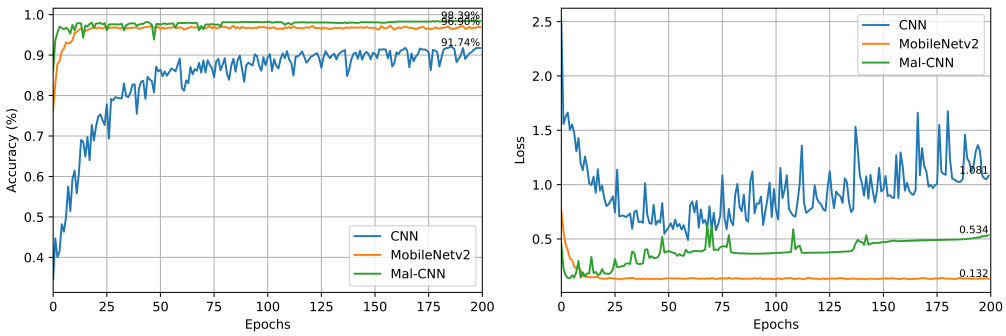


Figure 7. Testing results of the Mal-CNN using the original malware dataset, showcasing the model's resilience under operational conditions. (**a**) Testing accuracy curve. (**b**) Testing loss curve.

4.4. Evaluation on VM-Obfuscated Malware Dataset

Building upon the initial findings with the original dataset, we further explored the robustness and adaptability of the Mal-CNN model by testing it against a dataset composed of VM-obfuscated malware images. This dataset presents a more challenging environment designed to test the model's effectiveness against sophisticated evasion techniques.

The training phase for the VM-obfuscated dataset revealed that the Mal-CNN achieved an impressive accuracy of nearly 100% (99.99%) at the culmination of the training epochs, as demonstrated in Figure 8a. The rapid convergence of the training loss to a minimal value (Figure 8b) further confirms the model's efficient learning capabilities in the face of obfuscated data.

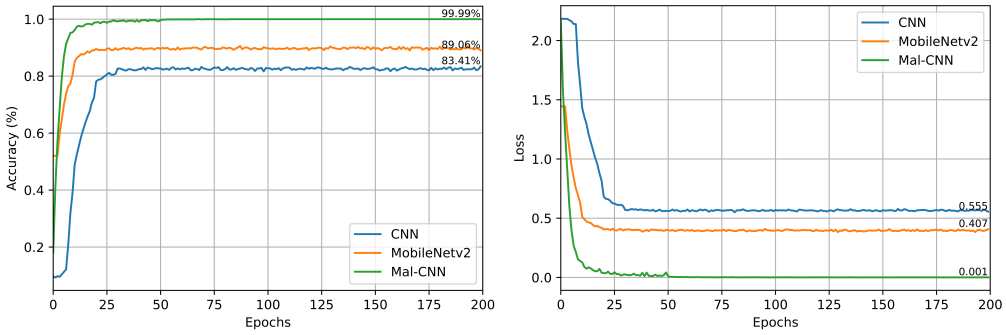
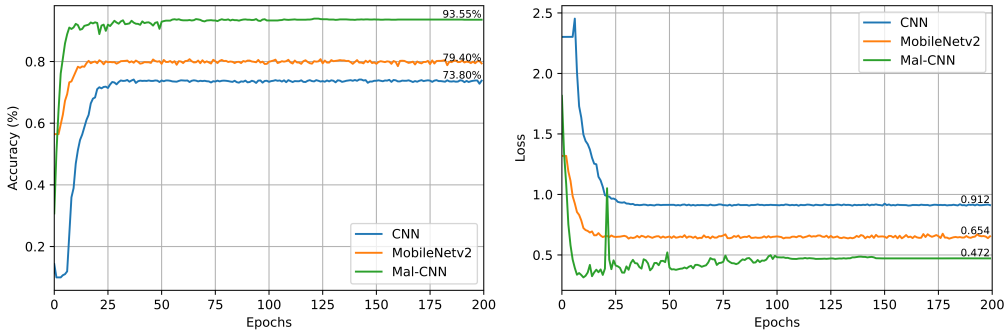


Figure 8. Learning performance of the proposed method using the obfuscation-processed malware dataset. This figure illustrates the model's ability to effectively adapt and learn under the added complexity introduced by VM obfuscation. (**a**) Curve for the training accuracy on the VM-obfuscated dataset. (**b**) Learning curve for the training losses on the VM-obfuscated dataset.

During testing, the Mal-CNN exhibited a robust accuracy of 93.55%, significantly outperforming the standard CNN, which managed only 73.80% under the same conditions (Figure 9a). To better explore how the Mal-CNN held up against existing solutions in the field, we choose MobileNetV2 as another baseline. However, MobileNetV2 only achieved an accuracy of 79.40%, despite the similar network structure. Notably, both the Mal-CNN and MobileNetV2 utilize depth-wise convolutional methods, and therefore, it can be concluded that the attention mechanism contributes greatly to the detection of VM-obfuscated malware. This stark contrast in testing performance, particularly in a complex scenario marked by VM obfuscation, underscores the efficacy of the depth-wise convolution and spatial attention mechanisms integrated within the Mal-CNN. Furthermore, the reduction in loss over epochs (Figure 9b) demonstrates the model's ability to

generalize well, suggesting that it can effectively handle the intricacies and challenges posed by obfuscation techniques.



(**a**) (**b**)

Figure 9. Testing results for the proposed method using the obfuscation-processed malware dataset. The results highlight the Mal-CNN's capacity to maintain a high level of performance, even when confronted with complex obfuscation techniques. (**a**) Testing accuracy curve for the VM-obfuscated dataset. (**b**) Testing loss curve for the VM-obfuscated dataset.

These findings articulate the substantial advantage conferred by the Mal-CNN architecture when dealing with advanced malware threats which employ obfuscation to evade detection. The model's superior performance in both the training and testing phases highlights its potential as a highly effective tool in cybersecurity capable of detecting even the most elusive threats. This phase of testing not only reinforced the model's robustness but also validated its applicability in real-world settings, where VM obfuscation is a prevalent challenge.

4.5. Evaluation Metrics

This evaluation covered two distinct datasets. The first dataset comprised images directly converted from the original malware executables, and the second dataset included images which were further processed using VM obfuscation techniques (see Table 3).

Table 3. Test recall and precision for CNN and Mal-CNN across two datasets.

Model	First Dataset		Second Dataset	
	Recall	**Precision**	**Recall**	**Precision**
CNN	82.74%	92.58%	73.80%	74.04%
MobileNetV2	89.85%	97.26%	79.39%	79.09%
Mal-CNN	93.65%	98.56%	93.55%	93.56%

The results from the datasets, as presented in Table 3, illustrate a distinct performance hierarchy among the evaluated models. The traditional CNN set-up, while performing well with a precision of 92.58% on unobfuscated malware, demonstrated considerable performance degradation when faced with the VM-obfuscated malware dataset. This highlights its limitations in coping with sophisticated obfuscation techniques. Meanwhile, MobileNetV2 not only showed enhanced efficiency but also better resilience against obfuscation compared with the traditional CNN, evidenced by its precision of 79.09% in the VM-obfuscated dataset. This improvement, particularly the precision reaching 79.39%, showcases the benefits of the depth-wise separable convolutions in dealing with complex malware forms.

In contrast, the Mal-CNN markedly outstripped the other models by combining both the depth-wise separable convolutions and the spatial attention mechanism. This model achieved the highest precision rates of 98.56% and 93.56% across the respective datasets.

The addition of the attention mechanism allowed the Mal-CNN to focus more adeptly on the relevant features within obfuscated malware images, substantially enhancing its detection capabilities.

These comparative results underscore the significant advantages of incorporating advanced architectural features such as spatial attention mechanisms alongside depth-wise convolutions, particularly in environments where obfuscation techniques complicate traditional detection methods.

4.6. Discussion

4.6.1. Dateset Limitations

Our dataset was assembled from a variety of sources which are known to encounter and catalog extensive ranges of malware threats. By incorporating data from multiple channels such as established malware repositories, real-time blacklists, and anonymized network traffic from corporate security perimeters, we gathered a dataset which is diverse and representative of the global threat landscape. Notably, this research primarily concentrated on a subset of well-known, high-profile malware families. This focus was driven by the availability of detailed behavioral data and the significant impact these malware types have on current threats. However, this emphasis may limit the model's ability to generalize to less-common or emerging malware types which exhibit different behavioral patterns or employ novel obfuscation techniques.

Our dataset comprised 10,000 images, each derived from executable malware files. Of these, a significant portion underwent various transformations to augment the dataset effectively and introduce variability which challenged the robustness of our models. Specifically, transformations such as rotation, scaling, and translation were applied to approximately 60% of the dataset. These transformations were intended to simulate variations in malware signatures which might occur due to changes in compilation environments or minor code alterations, thus providing a more rigorous test bed for our Mal-CNN model. The malware images in our dataset were categorized into 10 distinct classes, each representing a unique family of malware with specific characteristics and behaviors. Each category exhibited unique payload delivery mechanisms, propagation methods, and evasion techniques. For instance, ransomware typically encrypts user data and demands payment, while worms autonomously spread across networks. By training the Mal-CNN across such varied categories, the model learns to recognize and differentiate between complex patterns and signatures, enhancing its practical utility in real-world applications where diverse malware attacks are prevalent.

The use of image transformations on malware binaries, particularly rotations, may initially seem counterintuitive, since binary executable files are not naturally subject to orientation changes. However, the rationale behind such transformations is grounded in enhancing the model's ability to generalize from visual patterns rather than specific binary configurations. This approach is supported by the literature, which advocates for robust feature learning in visual recognition tasks [25], where models trained on varied orientations demonstrate improved performance in recognizing objects across different views [28]. In the context of malware detection, although the actual binary content remains orientation-invariant, the transformation encourages the model to focus on the invariant features within the data, thereby mitigating overfitting and improving the model's ability to generalize from complex, obfuscated malware samples [29].

4.6.2. Obfuscation Techniques and Limitations of Mal-CNN

The effectiveness of the Mal-CNN in dealing with VM-based obfuscation has been demonstrated throughout this study. However, VM-based techniques represent just one facet of the diverse obfuscation landscape. In VM-based obfuscation, the original executable's code is transformed into a form which can only be interpreted by a specific virtual machine. While the Mal-CNN has shown proficiency in learning and recognizing these transformed representations, it is essential to consider how it performs against other obfus-

cation methods, such as packing. Packing is a prevalent obfuscation technique wherein the executable's code is compressed or encrypted and then decompressed or decrypted at runtime by a stub. This method significantly alters the appearance of binary content when viewed as an image, posing different challenges from those presented by VM-based obfuscation. The ability of the Mal-CNN to detect malware obfuscated by packing was not the primary focus of this research, and thus it remains an area for further investigation.

The current configuration of the Mal-CNN was optimized for detecting VM-obfuscated malware, which might limit its effectiveness against other sophisticated obfuscation techniques like advanced packing, encryption, or polymorphic and metamorphic malware. These limitations highlight the necessity for continuous model evaluation and adaptation to incorporate a broader range of obfuscation techniques into the training regime. To address these challenges, future work will involve extending the training dataset to include a wider variety of obfuscated malware samples. Additionally, enhancing the Mal-CNN with adaptive learning capabilities to dynamically adjust to new obfuscation methods as they evolve remains a critical research direction.

Furthermore, the Mal-CNN utilizes depth-wise separable convolutions coupled with spatial attention mechanisms, which inherently increase the computational complexity of the model. This complexity manifests in longer training times and higher computational resource requirements compared with simpler models. However, the use of depth-wise separable convolutions optimizes the number of computations and parameters compared with standard convolutions, making it a more efficient choice for high-dimensional data processing. Despite these advantages, the real-time deployment of the Mal-CNN in systems with limited computational resources could be challenging. To mitigate this, model quantization techniques can be applied to reduce the precision of the numerical calculations, thereby decreasing the model size and speeding up inference without significantly sacrificing accuracy.

5. Conclusions

The experimental results from this study highlight the advanced capabilities of the Mal-CNN model, which notably exceeded the performance of a traditional CNN in both original and VM-obfuscated malware detection scenarios. Through the strategic incorporation of depth-wise convolutional layers and spatial attention mechanisms, the Mal-CNN demonstrated a robust ability to discern intricate patterns typical of sophisticated malware obfuscation techniques. This superior performance underscores the Mal-CNN's potential as an invaluable tool in the cybersecurity arsenal, particularly in environments where complex and evolving threats are prevalent.

Moreover, the Mal-CNN's efficacy in adapting to and accurately classifying data from diverse and challenging datasets confirms its architectural strengths and suitability for practical deployment in real-world settings. The success of the Mal-CNN in these tests encourages further exploration into its application across broader malware types and attack vectors.

Future research could focus on extending the model's architecture to incorporate multi-modal data sources and exploring the integration of additional neural network innovations to enhance its predictive accuracy and efficiency. Additionally, evaluating the Mal-CNN under real-time detection scenarios could provide insights into its performance and scalability in operational cybersecurity environments.

Author Contributions: Conceptualization, Y.P.; data curation, S.H.; formal analysis, S.H.; funding acquisition, S.H.; investigation, S.H.; methodology, S.H.; project administration, Y.P.; resources, S.H.; software, S.H.; supervision, H.Y. and Y.P.; validation, S.H. and H.Y.; visualization, S.H.; writing—original draft, S.H.; writing—review and editing, S.H. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The datasets and proposed Mal-CNN can be found at https://github.com/AHSCone/Mal_CNN (accessed on 15 July 2024).

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

References

1. Liu, Z.; Zheng, D.; Wu, X.; Chen, J.; Tang, X.; Ran, Z. VABox: A virtualization-based analysis framework of virtualization-obfuscated packed executables. In Proceedings of the 7th International Conference of the Advances in Artificial Intelligence and Security (ICAIS 2021), Dublin, Ireland, 19–23 July 2021; Proceedings, Part III 7; Springer: Cham, Switzerland, 2021; pp. 73–84.
2. Bang, C.H.; Suk, J.H.; Lee, S.J. VMProtect operation principle analysis and automatic deobfuscation implementation. *J. Korea Inst. Inf. Secur. Cryptol.* **2020**, *30*, 605–616.
3. Li, S.; Jia, C.; Qiu, P.; Chen, Q.; Ming, J.; Gao, D. Chosen-instruction attack against commercial code virtualization obfuscators. In Proceedings of the 29th Network and Distributed System Security Symposium, San Diego, CA, USA, 24–28 April 2022.
4. Zhu, H.; Kong, Q.; Xu, Z.; Chen, J.; Li, X. Research on Software Protection Technology Based on Driver. *Am. J. Inf. Sci. Technol.* **2020**, *4*, 46–50.
5. Lee, G.; Kim, M.; Yi, J.H.; Cho, H. Pinicorn: Towards Automated Dynamic Analysis for Unpacking 32-Bit PE Malware. *Electronics* **2024**, *13*, 2081. [CrossRef]
6. Abrath, B.; Coppens, B.; Broeck, J.V.D.; Wyseur, B.; Cabutto, A.; Falcarin, P.; Sutter, B.D. Code renewability for native software protection. *ACM Trans. Priv. Secur. TOPS* **2020**, *23*, 1–31. [CrossRef]
7. Alzubaidi, L.; Zhang, J.; Humaidi, A.J.; Al-Dujaili, A.; Duan, Y.; Al-Shamma, O.; Santamaría, J.; Fadhel, M.A.; Al-Amidie, M.; Farhan, L. Review of deep learning: Concepts, CNN architectures, challenges, applications, future directions. *J. Big Data* **2021**, *8*, 1–74.
8. Kattenborn, T.; Leitloff, J.; Schiefer, F.; Hinz, S. Review on Convolutional Neural Networks (CNN) in vegetation remote sensing. *ISPRS J. Photogramm. Remote Sens.* **2021**, *173*, 24–49. [CrossRef]
9. Salehi, A.W.; Khan, S.; Gupta, G.; Alabduallah, B.I.; Almjally, A.; Alsolai, H.; Siddiqui, T.; Mellit, A. A study of CNN and transfer learning in medical imaging: Advantages, challenges, future scope. *Sustainability* **2023**, *15*, 5930. [CrossRef]
10. Lu, J.; Tan, L.; Jiang, H. Review on convolutional neural network (CNN) applied to plant leaf disease classification. *Agriculture* **2021**, *11*, 707. [CrossRef]
11. Howard, A.G.; Zhu, M.; Chen, B.; Kalenichenko, D.; Wang, W.; Weyand, T.; Andreetto, M.; Adam, H. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. *arXiv* **2017**, arXiv:1704.04861.
12. Sandler, M.; Howard, A.; Zhu, M.; Zhmoginov, A.; Chen, L.C. MobileNetV2: Inverted Residuals and Linear Bottlenecks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 4510–4520.
13. Yakura, H.; Shinozaki, S.; Nishimura, R.; Oyama, Y.; Sakuma, J. Malware analysis of imaged binary samples by convolutional neural network with attention mechanism. In Proceedings of the Eighth ACM Conference on Data and Application Security and Privacy, Tempe, AZ, USA, 19–21 March 2018; pp. 127–134.
14. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.; Kaiser, L.; Polosukhin, I. Attention is all you need. In Proceedings of the Advances in Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; pp. 6000–6010.
15. Kim, J.Y.; Cho, S.B. Obfuscated malware detection using deep generative model based on global/local features. *Comput. Secur.* **2022**, *112*, 102501. [CrossRef]
16. Alani, M.M.; Mashatan, A.; Miri, A. XMal: A lightweight memory-based explainable obfuscated-malware detector. *Comput. Secur.* **2023**, *133*, 103409. [CrossRef]
17. Gopinath, M.; Sethuraman, S.C. A comprehensive survey on deep learning based malware detection techniques. *Comput. Sci. Rev.* **2023**, *47*, 100529.
18. Bensaoud, A.; Kalita, J.; Bensaoud, M. A survey of malware detection using deep learning. *Mach. Learn. Appl.* **2024**, *16*, 100546. [CrossRef]
19. Lu, T.; Du, Y.; Ouyang, L.; Chen, Q.; Wang, X. Android malware detection based on a hybrid deep learning model. *Secur. Commun. Netw.* **2020**, *2020*, 8863617. [CrossRef]
20. Nataraj, L.; Karthikeyan, S.; Jacob, G.; Manjunath, B. Malware Images: Visualization and Automatic Classification. In Proceedings of the 8th International Symposium on Visualization for Cyber Security, Pittsburgh, PA, USA, 20–21 July 2011.
21. Yang, W.; Park, Y. Identifying symmetric-key algorithms using CNN in Intel processor trace. *Electronics* **2021**, *10*, 2491. [CrossRef]
22. Suarez-Tangil, G.; Stringhini, G.; Cavallaro, L. Droids on Roids: Fortifying Scalable Malware Detection against Adversarial Attacks. *arXiv* **2019**, arXiv:1902.02354.
23. Li, Y.; Li, B. Enhancing Malware Detection via Generative Adversarial Networks. In Proceedings of the IEEE International Conference on Communications, Dublin, Ireland, 7–11 June 2020.
24. Papernot, N.; McDaniel, P.; Wu, X.; Jha, S.; Swami, A. Distillation as a defense to adversarial perturbations against deep neural networks. In Proceedings of the 2016 IEEE Symposium on Security and Privacy, San Jose, CA, USA, 22–26 May 2016; pp. 582–597.

25. Goodfellow, I.J.; Shlens, J.; Szegedy, C. Explaining and harnessing adversarial examples. In Proceedings of the International Conference on Learning Representations, Banff, AB, Canada, 14–16 April 2014.
26. Lei, T.; Xue, J.; Man, D.; Wang, Y.; Li, M.; Kong, Z. SDP-MTF: A Composite Transfer Learning and Feature Fusion for Cross-Project Software Defect Prediction. *Electronics* **2024**, *13*, 2439. [CrossRef]
27. Mai, C.; Liao, R.; Ren, J.; Gong, Y.; Zhang, K.; Zhang, C. MobileNet-Based IoT Malware Detection with Opcode Features. *J. Commun. Inf. Netw.* **2023**, *8*, 221–230. [CrossRef]
28. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. In Proceedings of the Advances in Neural Information Processing Systems, Lake Tahoe, NV, USA, 3–6 December 2012; Volume 25.
29. Szegedy, C.; Zaremba, W.; Sutskever, I.; Bruna, J.; Erhan, D.; Goodfellow, I.; Fergus, R. Intriguing properties of neural networks. *arXiv* **2013**, arXiv:1312.6199.