

Vision Based Malware Classification Using Deep Neural Network with Hybrid Data Augmentation

Md. Mahbubur Rahman^{1,3}^a, Md. Delwar Hossain¹^b, Hideya Ochiai², Youki Kadobayashi¹,
Tanjim Sakib³ and Syed Taha Yeasin Ramadan³^c

¹*Division of Information Science, Nara Institute of Science and Technology, Nara, Japan*

²*Graduate School of Information Science, The University of Tokyo, Tokyo, Japan*

³*Military Institute of Science and Technology, Dhaka, Bangladesh*

Keywords: Malware Classification, Deep Learning, Vision Transformer, DCGAN.


Abstract: Preventing malware attacks is crucial, as they can lead to financial losses, privacy breaches, system downtime, and reputational damage. Various machine learning and deep learning techniques have been proposed for malware classification. However, to evade detection, files from the same family are often altered by malware developers using various approaches so that they appear to be separate files. They may even appear as previously unidentified, commonly referred to as zero-day threats. These attacks can compromise the robustness of deep learning models trained for malware classification. In this research, we developed six fine-tuned Deep Neural Network (DNN) classifiers for classifying malware represented as images. A hybrid data augmentation technique based on Deep Convolutional Generative Adversarial Network (DCGAN) and traditional image transformation methods has been proposed to train the classifiers, enabling them to better handle malware variants. A subset of the publicly available Maling dataset, comprising six-class and the whole dataset, were used in the experiment. Additionally, both datasets were expanded using the proposed augmentation technique to train the developed classifiers. Experimental results reveal that vision transformer-based classifiers, trained with the proposed data augmentation technique, achieve a maximum accuracy of 99.94% for six-class classification and 99.79% for 25-class classification.


1 INTRODUCTION


Malware is any software that is expressly designed to harm, exploit or compromise computer systems, networks and devices. It includes a wide variety of dangerous applications such as viruses, worms, trojans, ransomware, spyware, adware and others. Traditional malware detection systems rely on signature-based approaches and heuristics to recognize known malware patterns or questionable activities. These systems compare the digital signatures or features of files and network traffic to a database of known virus signatures. While these methods have been efficient in detecting known malware, they have numerous limitations. Typical signature-based detection systems are reactive in nature, as they can only detect

and block malware for which signatures have already been developed and submitted to the database. This implies that new and previously unknown malware, known as zero-day threats, can simply evade these defenses leaving networks open to attacks. Second, malware developers have devised methods to avoid signature-based detection. They can obfuscate or encrypt their malicious code, making typical detection systems struggle to identify the malware based on its signature. To overcome these limitations, more advanced and adaptive approaches, such as machine learning and behavioral analysis are being employed to enhance malware detection capabilities.

Recent advancements in malware detection have seen the utilization of advanced Convolutional Neural Network (CNN) models. CNNs, originally designed for image recognition tasks, have proven to be effective in detecting patterns and features in complex data, including malware. By treating malware samples as images, CNN models may learn com-

^a <https://orcid.org/0000-0001-6525-2274>

^b <https://orcid.org/0000-0002-5968-0704>

^c <https://orcid.org/0000-0003-4275-6917>

plicated patterns and features that distinguish malicious software from benign software. In addition to CNN models, the usage of Vision Transformers has gained attention in malware detection. Vision Transformers use self-attentional mechanisms to identify interdependencies and linkages between various malware sample components. This enables the models to efficiently categorize and analyze different malware types based on their visual representations (Ma et al., 2021). On the other hand, by exposing the deep learning models to a wider variety of malware samples, it enables higher generalization and robustness. Image augmentation is commonly used in machine learning to artificially increase the diversity of a training dataset by applying various transformations to the existing images. Generative Adversarial Network (GAN) can also produce samples using its generator and discriminator networks that closely resemble the traits of novel and undiscovered malware variants, this strategy can be used to alleviate the difficulties caused by the dynamic and evolving nature of malware (Hu et al., 2021).

In this paper, we present a comprehensive methodology for malware classification, involving both preprocessing and augmentation of the dataset. The paper introduces following contributions in the the field of malware classification through the following key points-

1. The paper develops fine-tuned Transfer (BiT-M-R50x1, BiT-S-R50x1), Transformer (ViT-B/32 and ViT-B/16) and CNN (DenseNet121, VGG16) models based classifiers for vision oriented malware classification.
2. The paper introduces the usage of a hybrid data augmentation technique based on DCGAN and traditional image transformation methods for training robustness.
3. The paper demonstrates the usage of hybrid augmentation technique for dataset augmentation to train deep learning classifiers for performance enhancement.

The paper then compares the results of models trained on the augmented dataset with those trained on the original dataset demonstrating the impact of proposed augmentation technique on model performance. The results demonstrated the effectiveness of advanced machine learning models particularly the ViT-B/32 model with proposed augmentation technique, in accurately classifying and detecting malware.

2 RELATED WORKS

CNN models have been widely used in recent years for detecting and classifying malware. M. Yeo et al. (Yeo et al., 2018) suggested an automated malware detection approach based on CNN and other machine learning algorithms. Instead of relying on port numbers and protocols, the technique utilized 35 different features retrieved from packet flow. Using Stratosphere IPS project data, the study revealed that CNN and Random Forest (RF) obtained superior performance with over 85% accuracy, precision and recall for all classes. Arindam Sharma et al. (Sharma et al., 2019) offers a highly accurate and efficient malware detection solution based on 1-dimensional CNN. The system classifies binary files as dangerous or benign with minimum preprocessing, allowing the network to uncover features during training. The use of 1-dimensional convolutions distinguishes this approach from previous CNN-based approaches, resulting in better accuracy and training times when compared to state-of-the-art techniques. Mahmoud Abdelsalam et al. (Abdelsalam et al., 2018) provides an effective virus detection solution for cloud infrastructures based on CNN. The study employs a standard 2D CNN trained on metadata from virtual machine processes and improves accuracy with a unique 3D CNN that considers samples over a time interval. Experiments on randomly selected malware show that the 2D CNN model obtains an accuracy of roughly 79%, whereas the 3D CNN model greatly improves accuracy to over 90%.

Shun Tobiyama et al. (Tobiyama et al., 2016) offers a malware detection method based on process behavior that employs Deep Neural Networks (DNN). The suggested method entails training a Recurrent Neural Network (RNN) to extract features of process activity and a CNN to classify feature images created from the recovered features. The evaluation findings show great accuracy, with an AUC of 0.96 obtained in the best case scenario. Jixin Zhang et al. (Zhang et al., 2016) offers IRMD, a malware variant detection approach based on opcode image recognition. In this method, binary executables are broken down into opcode sequences and then transformed into pictures. The method detects harmful executables by using a CNN to compare the opcode images of target binaries with those of known malware samples. When the detection set contains a large number of binaries and the training set is limited, theoretical analysis and real-world testing show that the visualized analysis strategy improves detection accuracy by 15%.

Verma et al. (Verma et al., 2020) introduced an innovative approach that combines first-order and

second-order statistical texture features derived from grey level co-occurrence matrix (GLCM) for visualized malware classification using ensemble learning. They achieved an accuracy of 98.58%. Recently, the use of Vision Transformers (ViTs) also has gained attention in the field of malware detection (Belal and Sundaram, 2023).

In this paper, we introduce an efficient approach for malware detection by Big Transfers, Vision Transformers, CNNs and GANs. Our methodology leverages the strengths of each model and incorporates data augmentation techniques to enhance detection accuracy and robustness. The results obtained from our approach are compared with existing techniques, demonstrating its effectiveness in detecting and classifying malwares.

3 METHODOLOGY

The methodology used in this study is separated into two parts. Here we explored how the robustness of the deep learning models can be increased by exposing diversified examples to the models in vision-based malware classification. The initial dataset was created from publicly available popular data set – Maling (Nataraj et al., 2011) which has 25 diversified families of malware samples.

As we employ GAN model for new image sample generation, for high- quality generation of new image samples, a good number of samples are required. For our case, we opted for classes with over 300 samples, resulting in a total of six classes and 6560 image samples that constitute our base dataset. We augmented this base dataset using traditional, GAN and Hybrid augmentation techniques. In preprocessing stage, we employ techniques such as scaling and normalization to convert the raw dataset into a classification-ready state. The developed classifiers are trained on the pre-processed datasets and their performance is assessed using measures such as accuracy, loss, precision, recall and F1 score. In the next part of the methodology, we experimented using the whole Maling dataset following similar strategy. Figure 1 displays the overall methodology.

3.1 Malware Detection Using Deep Neural Network (DNN)

3.1.1 Big Transfer

BigTransfer or BiT introduced by Google stands out as an advanced transfer learning technique designed for image classification. Its utilization of pre-trained

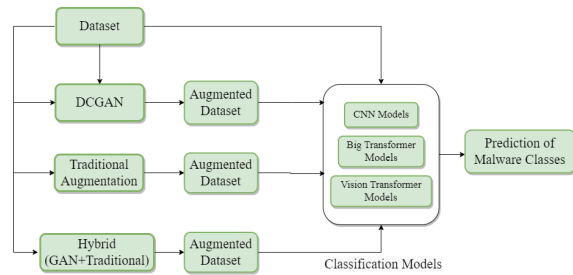


Figure 1: Overall methodology of the study.

representations enhances sample efficiency and simplifies the hyperparameter tuning process during the training of deep neural networks for visual tasks. BiT reevaluates the traditional approach of pre-training on expansive supervised datasets and subsequently fine-tuning the model for a specific target task.

3.1.2 Vision Transformer

Vision Transformer (ViT) (Dosovitskiy et al., 2020) is an advanced deep-learning model created exclusively for image identification tasks. It differs from typical convolutional neural networks (CNNs) in that it makes use of the capability of transformer-based architectures, which were originally created for natural language processing. ViT uses the self-attention method to images, allowing it to detect both local and global dependencies. The input image is divided into patches in ViT, each of which is represented by a fixed-size vector. These patches are then linearly embedded to get their embeddings. Positional embeddings are added to the patch embeddings to incorporate positional information.

3.1.3 CNN Models

The CNN model utilizes convolution and pooling layers to autonomously extract various hierarchies of features, ranging from basic attributes such as edges and corners to highly specific details. Typically, the feature maps undergo flattening through a flatten or global pooling operator, resulting in a 1-dimensional feature vector. This vector is subsequently fed as input through several fully-connected dense layers, culminating in the prediction of the output class using a softmax output layer. The goal of this hierarchical architecture, implemented across multiple stages, is to acquire spatial hierarchies of features while maintaining translation invariance. This is made feasible by means of the convolution and pooling layers, the two primary components in the CNN architecture. In this study, DenseNet121 and VGG16 are employed in our experimentation as CNN models.

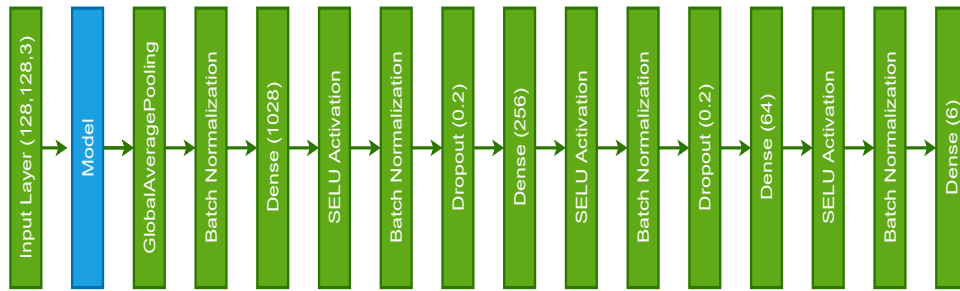


Figure 2: Layer architecture of deep learning models.

3.1.4 Proposed DNN Model

The proposed DNN layer architecture is shown in Figure 2. The input layer has the shape (128,128,3), indicating that it accepts images with 128x128 pixels and three color channels (RGB). Following that, the model (Transfer, Transformer or CNN) goes through multiple layers of operations to extract relevant features and classify the input images.

The GlobalAveragePooling layer is the first layer after the input layer and it decreases the spatial dimensions of the feature maps by computing the average of each feature map, resulting in a more compact representation of the data. Following that, a Batch Normalization layer is used to normalize the activations of the previous layer, enhancing the model’s stability and convergence during training. The Dense layer with 1028 units is followed by the SELU (Scaled Exponential Linear Units) activation function.

The activation functions known as Scaled Exponential Linear Units, or SELU, induce self-normalizing features is defined as:

$$SELU(x) = \lambda \cdot \begin{cases} x, & \text{if } x > 0 \\ \alpha \cdot (\exp(x) - 1), & \text{if } x \leq 0 \end{cases} \quad (1)$$

where, α and λ are two hyperparameters. In order to maintain the mean and variance of activations at or near 0 and 1, respectively, during training, SELU provides self-normalization.

Following the activation of the SELU, the subsequent layers are designed with combination of Batch Normalization, Dropout (0.2), SELU and Dense layers as depicted in Figure 2. The model hyperparameters are shown in Table 1.

Table 1: Hyperparameters used in the experiment.

Batch Size	Epochs	Optimizer	Learning Rate	Loss
128	50	Rectified-Adam	1.00E-3	Categorical Cross-entropy

3.2 DCGAN Data Augmentation

The DCGAN (Deep Convolutional Generative Adversarial Network) is a type of generative model that uses deep convolutional neural networks to generate realistic images. The architecture of DCGAN consists of two main components: the generator and the discriminator. The generator aims to generate realistic images from random noise, while the discriminator tries to distinguish between real and generated images by minimizing corresponding losses.

The generator and discriminator losses are given by

$$\mathcal{L}_{gen} = -\frac{1}{N} \sum_i \log(D(G(z_i))) \quad (2)$$

and

$$\mathcal{L}_{disc} = -\frac{1}{N} \sum_i [\log(D(x_i)) + \log(1 - D(G(z_i)))] \quad (3)$$

where, D represents the discriminator, G represents the generator, x_i represents the real images, z_i represents the random noise vectors and N represents the batch size.

3.3 Hybrid Augmentation

GAN has the capability to produce data that closely mimics the patterns present training data it has been exposed to. However, it generates images in an unsupervised manner and controlling specific augmentation parameters e.g., rotation angle are challenging. The quality of images generated by GANs may not always match the quality of real images in the training set. It may introduce artifacts and inconsistencies in the augmented data, Training GANs can be challenging and unstable. GANs involve training a generator and a discriminator simultaneously, and finding the right balance can be difficult. Instabilities in training may lead to poor-quality generated images or an ineffective augmentation process. For that we adopted hybrid augmentation approach for training the classifier. Here the data is augmented using DCGAN. Ad-

ditionally we augment the data using traditional augmentation method. Finally, original training data and augmented data from GAN and traditional augmentation are used for training.

4 RESULT ANALYSIS

In this study, we used Maling dataset (Nataraj et al., 2011) which comprises of malware samples classified into various kinds. Here, Windows malwares are treated as binary codes, with each 8 bits representing a pixel value. These values map to the grayscale image pixels ranging from 0 to 255. The dataset contains a total of 9339 samples. Each malware variant has a different number of samples, ranging from as little as 80 for Skintrim.N to as many as 2949 for Allaple.A.

Our initial dataset referred to as the base dataset consists of six classes extracted from the Maling dataset and comprises 6560 samples of images depicting malware. We augmented the image samples of six classes using DCGAN so that the total samples became 7200 with more images were generated for small classes. Table 2 depicts the distribution of the 6 malware families/classes of base dataset and their corresponding augmentation numbers. Though the original dataset Maling contains 25 families, in our initial study we have chosen 6 families, those who have 300 or more samples for DCGAN augmentation. The chosen families come from different worm, trojan, rogue types of malwares. Among them Allaple.A and Allaple.L are variants of Win32/Allaple family that represents a multi-threaded, polymorphic network worm with the ability to propagate across interconnected computers within a local area network (LAN) (Microsoft, 2023). The worm's file possesses polymorphic encryption, causing each instance of the worm to be distinct from one another. The rest of the families exhibit their own distinct traits for infecting systems through their unique methods of deception.

Figure 3 shows three distinct sets of samples: malware samples sourced from the Maling dataset, samples generated by the DCGAN model and samples created through image transformation techniques.

The base dataset for this work was augmented with DCGAN, resulting in an additional 640 samples comprising the GAug dataset. Augmentation of the base dataset samples were also performed equally (Table 2) using traditional augmentation technique, the resulting dataset is referred to as TAug. The augmentation procedure sought to improve the robustness and generalization capabilities of the malware detection models by increasing the diversity and variability of the dataset. Finally, the HAug dataset is formed

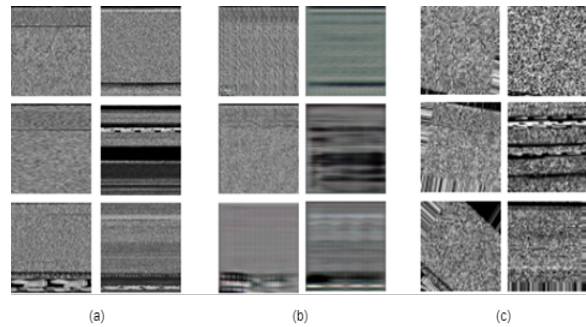


Figure 3: (a) Malware samples from Maling dataset. (b) DCGAN generated samples. (c) Samples generated using image transformation.

by combining samples from the base dataset, additional samples generated through DCGAN augmentation and traditional augmentation methods.

In our experiments, for each case 20 percent data were set aside for testing purposes and the developed classifiers were trained using rest 80 percent of the data. For classifiers trained with base data set, we noticed variable performance across the examined architectures for malware detection. Models' performances with respect to Precision, Recall, F1-score and loss values are furnished in Table 3. The performance values of various models after trained with augmented dataset in malware detection with respect to Precision, Recall, and F1 score are also furnished in Table 3. Some training scenarios are depicted in Figure 4 for six-class classification.

From Table 3, it is evident that augmentation positively affects the performance of different classifiers in six-class classification. For example, the DenseNet121-based classifier obtains precision values of 0.9832, 0.9885, 0.9888, and 0.9888 for the Non-augmented, TAug, GAug, and HAug datasets, respectively. Similarly, for the ViT-B/16-based classifier, recall values of 0.9984, 0.9972, 0.9986, and 0.9993 are obtained for the Non-augmented, TAug, GAug, and HAug datasets, respectively. In general, classifiers trained on GAN-augmented and Hybrid methods produced augmented datasets achieve better results than classifiers trained with non-augmented dataset. Even they perform better than the classifiers trained with datasets augmented with traditional techniques. It is to be noted that ViT-based classifiers exhibited a close performance when trained with a non-augmented dataset in comparison to their performance when trained with an augmented dataset.

Figure 5 provides an overview of the performance measures, specifically accuracy of various machine learning models when augmented with different techniques. DCGAN, No Augmentation, Traditional Augmentation and a Hybrid method are all studied.

Table 2: Augmented Dataset (DCGAN and Traditional augmentation.)

Malware Family/Class	Malware Type	Number of Samples	Additional Augmented samples	Augmented Dataset
Allaple.A	Worm	2949	40	2989
Allaple.L	Worm	1591	40	1631
Fakerean	Rogue	381	140	521
Instantaccess	Trojan	431	140	571
VB.AT	Worm	408	140	548
Yuner.A	Trojan	800	140	940
Total		6560	640	7200

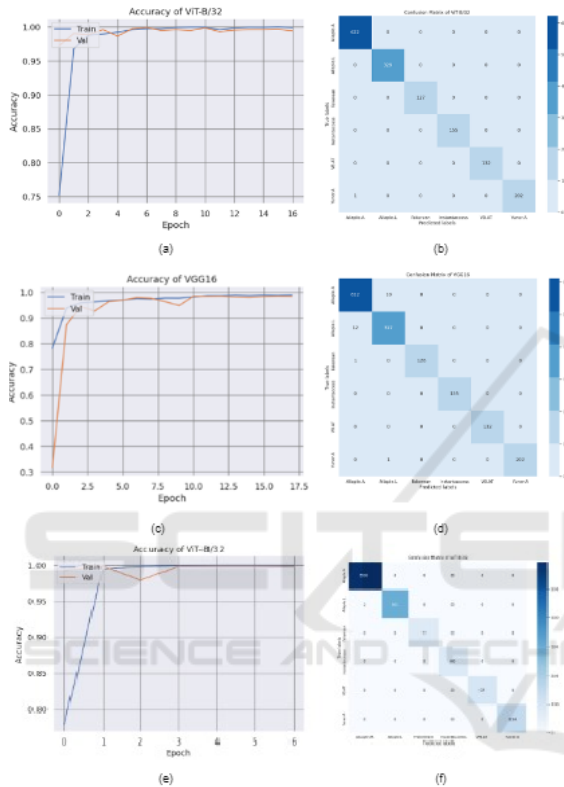


Figure 4: For six-class classification: (a) and (b) are ViT-B/32 model’s Epoch vs. Accuracy graph and confusion matrix after training with dataset with hybrid augmentation. (c) and (d) are VGG-16 model’s Epoch vs. Accuracy graph and confusion matrix after training with dataset with hybrid augmentation. (e) and (f) are ViT-B/32 model’s Epoch vs. Accuracy graph and confusion matrix where no augmentation is used.

BiT-M-R50x1, BiT-S-R50x1, ViT-B/32, ViT-B/16, DenseNet121 and VGG16 are among the models under consideration. Notably, the DCGAN augmentation technique consistently produces excellent accuracy across all models, with BiT-M-R50x1 scoring 98.4%, BiT-S-R50x1 scoring 97.91%, ViT-B/32 scoring 99.93%, ViT-B/16 scoring 99.86%, DenseNet121 scoring 98.89%, and VGG16 scoring 98.95%. On the other hand hybrid augmentation techniques perform best across models. This demonstrates the efficacy of GAN based augmentation and Hybrid augmenta-

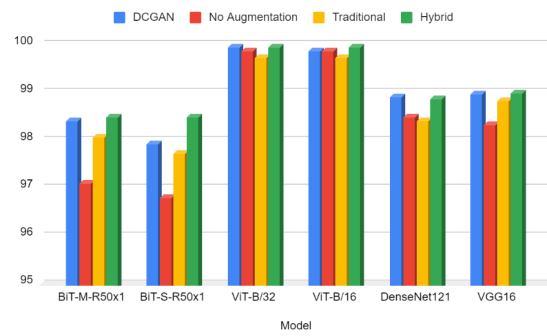


Figure 5: Comparison of accuracy for six-class classification.

tion technique in improving model performance.

Among the other models, ViT-B/32 and ViT-B/16 demonstrated exceptional performance with accuracy values of 99.93% and 99.86%, respectively. They both achieved near-perfect precision, recall and F1 scores, highlighting their ability to accurately classify malware instances. Comparatively, DenseNet121 and VGG16 achieved slightly lower accuracies ranging from 96.58% to 99.16%. However, they still exhibited reasonably high precision, recall and F1 scores, indicating their effectiveness in malware detection. BiT models also show improved performance after training with augmented dataset.

Models trained without augmentation perform somewhat worse in terms of accuracy. For example, BiT-M-R50x1 achieves 97.1%, BiT-S-R50x1 achieves 96.8%, ViT-B/32 achieves 99.85%, ViT-B/16 achieves 99.85%, DenseNet121 achieves 98.48% and VGG16 achieves 98.32%. Traditional augmentation methods, designated as "Traditional," produce competitive results, with BiT-M-R50x1 scoring 98.06%, BiT-S-R50x1 scoring 97.71%, ViT-B/32 scoring 99.72%, ViT-B/16 scoring 99.72%, DenseNet121 scoring 98.4% and VGG16 scoring 98.82%. The Hybrid approach, which combines DCGAN and classical augmentation, achieves substantial accuracy gains. BiT-M-R50x1 scores 98.47%, BiT-S-R50x1 scores 98.47%, ViT-B/32 scores 99.94%, ViT-B/16 scores 99.94%, DenseNet121 scores 98.85% and VGG16 scores 98.97%.

To observe the proposed augmentation effect on more number of classes, we employed hybrid augmentation on the entire Malimg dataset. The same hybrid augmentation was applied to the specified six classes of the Malimg dataset. Specifically, these six classes were augmented using both traditional and DCGAN augmentation techniques, following the details outlined in Table 2. For the remaining 19 classes, 140 samples were augmented using the traditional augmentation method. Once again, 20 percent of the

Table 3: Performance comparison of different models with and without augmentation for six-class classification. (No_Aug = Without Augmentation, GAug = Augmentation with DCGAN, TAUG=Traditional Augmentation, HAUG=Hybrid Augmentation).

Model	Precision				Recall				F1-score			
	GAug	No_Aug	TAug	HAug	GAug	No_Aug	TAug	HAug	GAug	No_Aug	TAug	HAug
BiT-M-R50x1	0.9839	0.9710	0.9805	0.9847	0.9840	0.9710	0.9805	0.9846	0.9839	0.9709	0.9805	0.9846
BiT-S-R50x1	0.9794	0.9688	0.9771	0.9847	0.9791	0.9680	0.9770	0.9846	0.9789	0.9675	0.9770	0.9846
ViT-B/32	0.9993	0.9984	0.9972	0.9993	0.9993	0.9984	0.9972	0.9993	0.9993	0.9984	0.9972	0.9993
ViT-B/16	0.9986	0.9984	0.9972	0.9993	0.9986	0.9984	0.9972	0.9993	0.9986	0.9984	0.9972	0.9993
Dense-Net121	0.9888	0.9849	0.9855	0.9885	0.9888	0.9848	0.9854	0.9885	0.9888	0.9847	0.9853	0.9885
VGG16	0.9895	0.9832	0.9895	0.9896	0.9895	0.9832	0.9895	0.9896	0.9895	0.9832	0.9895	0.9896

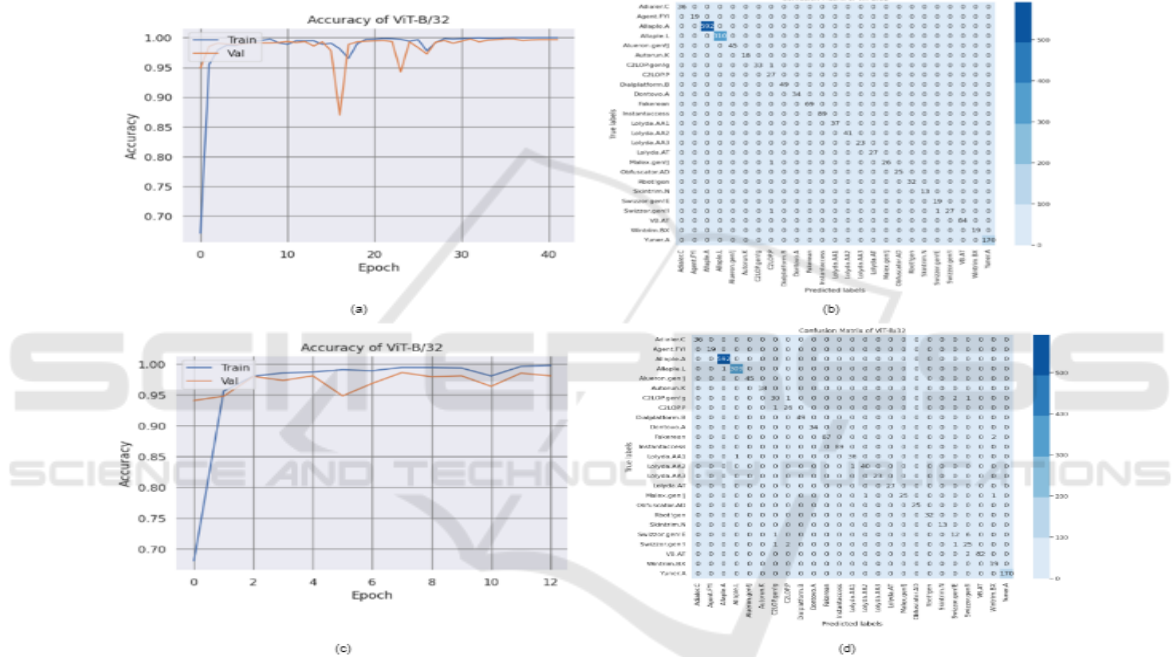


Figure 6: For Maling dataset: (a) and (b) are ViT-B/32 model’s Epoch vs. Accuracy graph and confusion matrix after training with dataset enhanced by hybrid augmentation. (c) and (d) model’s Epoch vs. Accuracy graph and confusion matrix for the same classifier when trained with dataset without augmentation.

data was set aside for testing.

The top two performers, namely the ViT-B/32 and ViT-B/16 models based classifiers were selected for classification experimentation. Both classifiers underwent training using hybrid augmented data and the non-augmented Maling dataset with 25 classes.

The results are summarized in Table 4. To compare, we included some recent results from the literature in Table 4. In this stage, we observed that classifiers based on the ViT-B/32 and ViT-B/16 models with the hybrid data augmentation technique, both achieve accuracies of 99.79%. Whereas ViT-B/16 and ViT-B/32 models based classifiers achieve accuracy of 98.29% and 98.71% respectively when

Table 4: Performance comparison of different approaches to malware classification using Maling dataset. (HyAug=Hybrid Augmentation, NAug=No Augmentation).

Model	Precision	Recall	F1 score	Accuracy
Belal et al.	0.9884	0.9808	0.9842	0.9932
Verma et al.	0.9804	0.9806	0.9805	0.9858
ViT-B/16(NAug)	0.9828	0.9828	0.9826	0.9829
ViT-B/32(NAug)	0.9871	0.9866	0.9866	0.9871
ViT-B/16(HyAug) (Proposed)	0.9978	0.9978	0.9978	0.9979
ViT-B/32(HyAug) (Proposed)	0.9980	0.9978	0.9978	0.9979

they are trained with non-augmented Maling dataset. Other measures like Precision, Recall and F1 score for the best performer classifier were found to be 0.9980,

0.9978 and 0.9978 respectively. The effectiveness of the proposed hybrid augmentation method is evident in the Precision, Recall and F1 scores provided in Table 4 for Vision Transformer based classifiers.

The training snapshots for the ViT-B/32 model, trained with and without hybrid augmentation are presented in Figure 6. The classifier exhibits fewer misclassifications when trained with the hybrid augmented dataset, underscoring the efficacy of proposed hybrid augmentation technique.

5 CONCLUSION AND FUTURE WORK

In this study, we developed a set of malware classifiers based on Transfer, Transformer and CNN models to identify different malware classes. We incorporated DC-GAN augmentation and classical augmentation methods alongside these classifiers. Furthermore, we introduced a hybrid augmentation technique that combines DCGAN and classical augmentation methods.

Initially, a dataset containing six-class malware samples was created, and the developed classifiers were trained to learn patterns from this dataset. The performance of these classifiers was evaluated using metrics such as accuracy, loss, precision, recall and F1 score. The initial dataset was then augmented using DCGAN, traditional and the proposed method. Experimental results indicated that the ViT-B/32 model-based classifier, trained with a dataset augmented with the proposed method, outperformed others, achieving the highest accuracy of 99.94%.

In the second phase of experiments, we utilized the publicly available Malimg dataset with the developed classifiers and the proposed augmentation technique. Here as well, classifiers trained with a hybrid augmentation-enhanced dataset outperformed those trained with a non-augmented dataset, achieving the highest accuracy of 99.79%.

Overall, the augmentation process aimed to enhance the resilience and generalizability of malware detection models by amplifying diversity and variability within the dataset. The experimental outcomes underscore that incorporating augmented training data into the dataset contributes to the enhancement of classifier performance. This study utilized only one dataset and future research could explore different datasets with model diversity for malware detection and classification.

ACKNOWLEDGMENT

This study was funded by the International Exchange Program of the National Institute of Information and Communications Technology (NICT), Japan.

REFERENCES

- Abdelsalam, M., Krishnan, R., Huang, Y., and Sandhu, R. (2018). Malware detection in cloud infrastructures using convolutional neural networks. In *2018 IEEE 11th international conference on cloud computing (CLOUD)*, pages 162–169. IEEE.
- Belal, M. M. and Sundaram, D. M. (2023). Global-local attention-based butterfly vision transformer for visualization-based malware classification. *IEEE Access*, 11:69337–69355.
- Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al. (2020). An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.
- Hu, H., Peng, S., He, Z., and Huang, L. (2021). Malgan: Gan-augmented dataset for malware detection. In *Proceedings of the 2021 International Conference on Computer Science and Network Technology*, pages 246–252. ACM.
- Ma, X., Yang, S., and Yang, H. (2021). Malware classification based on vision transformer. *Journal of Physics: Conference Series*, 1932(1):012024.
- Microsoft (2023). Microsoft security intelligence. Microsoft.
- Nataraj, L., Karthikeyan, S., Jacob, G., and Manjunath, B. S. (2011). Malware images: visualization and automatic classification. In *Proceedings of the 8th international symposium on visualization for cyber security*, pages 1–7.
- Sharma, A., Malacaria, P., and Khouzani, M. (2019). Malware detection using 1-dimensional convolutional neural networks. In *2019 IEEE European symposium on security and privacy workshops (EuroS&PW)*, pages 247–256. IEEE.
- Tobiyama, S., Yamaguchi, Y., Shimada, H., Ikuse, T., and Yagi, T. (2016). Malware detection with deep neural network using process behavior. In *2016 IEEE 40th annual computer software and applications conference (COMPSAC)*, volume 2, pages 577–582. IEEE.
- Verma, V., Muttoo, S. K., and Singh, V. (2020). Multiclass malware classification via first-and second-order texture statistics. *Computers & Security*, 97:101895.
- Yeo, M., Koo, Y., Yoon, Y., Hwang, T., Ryu, J., Song, J., and Park, C. (2018). Flow-based malware detection using convolutional neural network. In *2018 International Conference on Information Networking (ICOIN)*, pages 910–913. IEEE.
- Zhang, J., Qin, Z., Yin, H., Ou, L., and Hu, Y. (2016). Irm: malware variant detection using opcode image recognition. In *2016 IEEE 22nd International Conference on Parallel and Distributed Systems (ICPADS)*, pages 1175–1180. IEEE.