Malware Classification by Vision Transformer and Transfer Learning

[1]Sufia Zareen, [2]Md Abdul Alim, [3]Nasrin Akter Tohfa, [4]Md Reduanur Rahman, [5]Md Shakhawat Hossen, [6]Md Habibul Arif, [7]Mamunur Rahman

[1]Master's in Genetics, Osmania University, Hyderabad, India (2022)
[2]Master's in Information Technology in Management, St. Francis College, Brooklyn, NY, USA
[3]Bachelor of Education, National University of Bangladesh
[4]Master's in Information Technology, Washington University of Science and Technology, Alexandria, Virginia
[5]Master's in Information Technology, Washington University of Science and Technology, Alexandria, Virginia
[6]Master of Science in Information Technology, Washington University of Science and Technology, Virginia
[7]Master's in Commerce, Jagannath University College, Dhaka, Bangladesh

[1]techbio36@gmail.com, [2]malim@sfc.edu, [3]nasrinaktertohfa@gmail.com
, [4]Mdrahman.student@wust.edu, [5]mhossen.student@wust.edu, [6]habibularif2233@gmail.com,
[7]mamun.ros14@gmail.com

***Abstract*—The spread of malicious software in recent times has undergone rapid evolution, making it highly urgent to develop intelligent and effective malware detection mechanisms. This paper aims to provide a comparative analysis of deep learning solutions for malware classification based on Transfer Learning using VGG19, Vision Transformer, and Convolutional Neural Network. The models were trained according to a balanced image-based malware dataset comprising different malware families after converting binary files into grayscale images. The primary objective was to compare transformer-based models with traditional convolutional frameworks in terms of their performance in recognizing intricate patterns typical of multiple malware categories. VGG19 with transfer learning attained the highest classification accuracy of 99%, surpassing both the Vision Transformer and CNN. However, the Vision Transformer model also demonstrated similar accuracy levels, which signifies the high potential of attention-based frameworks in malware image classification. The CNN, based on a custom model, resulted in acceptable but relatively low outcomes, which proves the efficiency of pre-trained feature-learning in transfer learning tasks.**

***Index Terms*— Cyber Security, Transfer Learning, Vision Transformer, VGG19.**

I. INTRODUCTION

At present, malware stands out as one of the most common and constantly evolving threats to information safety throughout the digital age. The evolution of cybercriminals' practices that allow them to bypass traditional security systems became a driving factor for an innovative, intelligent, and automated means of malware detection. Traditional signature-based or heuristic detection methods demonstrate inefficiency when it comes to identifying new malware generations. The underlying reason is that they all utilize predefined patterns that do not generalize to undeveloped threats. Recent initiatives have been actively investigating the possibility of leveraging deep learning and computer vision to transform malware into a visual form.[1] This study aims to explore the efficacy of three different deep learning methods: Transfer Learning using VGG19, the Vision Transformer, and a bespoke Convolutional Neural Network, when applied to malware image classification. The VGG19 model, which uses transfer learning, extracts informative and distinctive features from malware images, using pre-trained weights on a massive dataset like ImageNet.[2] Moreover, the Vision Transformer is a relatively new CNN architecture that employs self-attention approaches to learn expansive neighborhood relationships and global impressions by viewing the problem in a new light from traditional convolutional design. In contrast, the bespoke CNN is responsible for classifying malware from images directly from scratch, in order to use it as a basis of comparison to assess the other two methods. In summary, this research sought to evaluate and compare how well each of the above classification models could generalize and detect different families of malware.[3] Results obtained from a comparative study show that VGG19, facilitated by transfer learning, yielded the highest accuracy percentile, with a general performance of 99% in all tested scenarios. Notably, the Vision Transformer model nearly matched or competed with the best classification model, thus rendering it effective and ideal for future cybersecurity applications.[4] Moreover, the current study demonstrates that the use of transfer learning and transformer models can enhance the accuracy and robustness of malware detection systems suitable for integration into modern cybersecurity defense systems.

II. BACKGROUND

From traditional static and dynamic analysis methods, malware classification has developed to modern deep learning-based techniques to automatically learn hidden patterns from raw data. This image-based approach first appeared in [5], where Nataraj et al. demonstrated how malware binaries can be visualized as gray images, and their textures can be fed into a convolutional neural network to achieve malware classification. The authors obtained family-level texture-based classification accuracy beyond 60%, also indicating that, based on this concept, further research can be much more productive. In Kalash et al. [6], they presented their vision of the same idea that won't be familiarized with texture but with color images. Their deep CNN model delivered an impressive enhancement over classical feature-based approaches, further demonstrating the applicability and potency of deep architectures to cybersecurity applications.

Further works have focused on CNN variants and hybrids to enhance robustness and accuracy. Lad and Adamuthe [7] presented a hybrid CNN–SVM model with an accuracy of 99.59% and Vasan et al. [8] designed an ensemble CNN framework, which was able to generalize effectively across packed and unpacked malware. Gibert et al. [9] identified that shallow CNNs – those with only two or three convolutional layers – reached an accuracy score as high as 94.6%. This establishes a principle that model depth must be a trade-off, balancing complexity and overfitting.

Transfer learning was the most widely accepted technique to alleviate the burden of data scarcity and training overhead. Awan et al.[10] implemented a VGG19 model with spatial attention and achieved an accuracy of 97.68% on this model, VGG19, whereas Ahmed et al.[11] Implemented an InceptionV3 model and achieved an accuracy of 98.76%. Kumar et al.[12] Implemented MCFT-CNN based on ResNet50 and validated an accuracy of 98.63% on the IoT malware dataset. Narayanan and Davuluru[13] implemented an ensemble of CNN and DNN-based models and generated an accuracy of nearly 99.8% identified to be better than single CNN models.

Prior to that, to maximize computational performance, Dang et al. [14] incorporated CNNs with BiLSTM layers, which corresponded to spatial and consecutive feature detection. As a result, a 94.32% accuracy was attained. Pant and Bista [15] confirmed the appropriateness of transfer learning with CNNs utilizing ImageNet as pre-training data with an accuracy level of 98.7% and negligible overfitting. In addition, Tekerek and Yapici [16] developed a B2IMG-CNN hybrid approach, achieving 99.86% accuracy, demonstrating the advantage of application and imagery pipelines.

Comparative studies, such as model interpretability and dataset diversity. Raff et al. [17] introduced MalConv, an end-to-end network trained on raw binary bytes, which implied that a deep model could learn deep features in a discriminative way without any manual preprocessing. Also, Go et al. [18] applied ResNeXt to malware visualization, informational analysis, providing that the residual and grouped convolution improved the feature learning process, and acquired the highest accuracy of 98.86%. Mitsuhashi and Shinagawa [19] used the VGG19 model that was customized for malware application to have the highest accuracy of 99.72% compared to other models such as VGG16 and DenseNet. AlGarni et al. [20] utilized the transfer learning process with EfficientNet-B3 to have a 99.93% accuracy; this result indicated that modern lightweight CNNs are trending. Finally, Bensaoud et al. [21] wrote that Inception V3 outperformed M-CNN, which received a high accuracy value of 99.24% before 2022, which is a state-of-the-art for malware detection.

III. METHODOLOGY

A structured workflow is provided in Figure 1 for this methodology, which begins with data collection and preprocessing by removing noise, consistent resizing the data to make it equal in length. The prepared dataset was then used for the implementation of the algorithm of VITS and deep learning models, followed by evaluations.

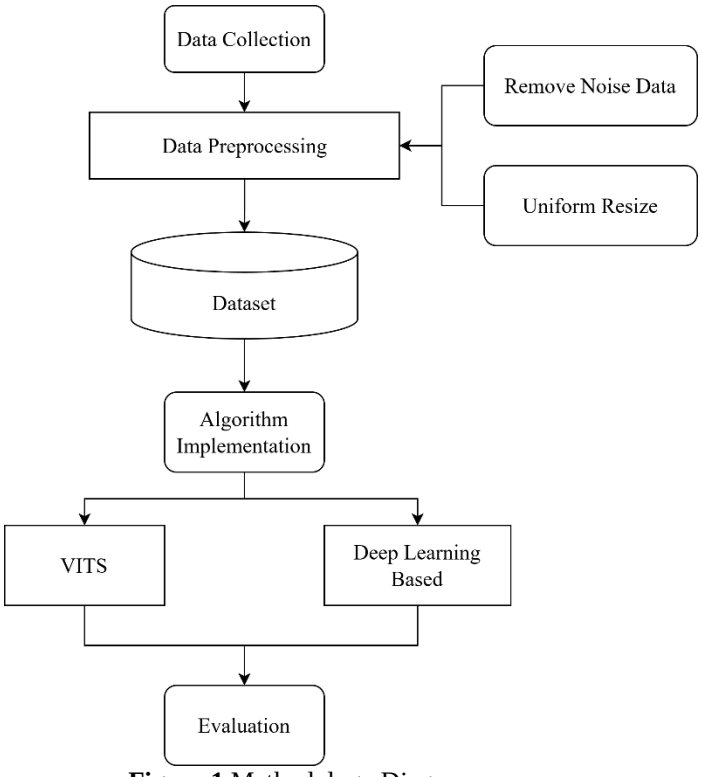

Figure 1 Methodology Diagram

Data Collection

For our case study, we used the Malimg dataset in its Original form obtained from Kaggle [22]. The dataset consists of grayscale images of malicious software programs, which were converted from the binaries of the executable files. Each image belongs to a class of malicious software. This facet may assist in creating a classification model with multiple classes. Before applying this dataset to the Vision Transformer model, we resized the images and reduced the range of values each pixel can have. We used augmentation techniques in order to create more cases to generalize over

Data Preprocessing

Once the malware dataset is obtained from Kaggle, the following crucial step is data preprocessing. This procedure helps prepare the dataset to be clean, coherent, and fit for training deep learning models. First, noise removal helps to discard equally

irrelevant or damaged samples to avoid misleading the learning. Moreover, missing or null values pose potential risks and are managed with alertness through data imputation or deletion methods, respectively, based on data loss.

In our case, because all malware data is represented in the form of images, all files are standardized and resized to a uniform resolution more suitable for deep learning architectures. As each sample is processed equally to extract features, they all contribute equally. Additionally, to ensure accelerated training processes and maintain numerical stability, pixel values are normalized. All these preprocessing tasks altogether lead to enhanced dataset quality, improved computational efficiency, and lower risks of overfitting, allowing for more reliable model learning.

Dataset

The figure shown is a pie chart of imbalanced classes employed in this study's malware dataset. These classes cover various malware families, and each section represents the proportion of total samples specific class comprises. As illustrated in the visualization, the distribution of the dataset is highly imbalanced. The Allaple.A and Allaple.L families comprise the majority of the dataset with a total of around 35.9% and 19.4% of all samples, respectively. This means that over half of the entire dataset is spread across these two classes. Yuner.A and VB.AT are other significant families with around 9.7% and 5.0% of the dataset's total share, while Fakerean makes 4.6%. The remaining classes account for less than 3% of all samples; 'Other' makes up 6.3%. The fig-share of these malware classes implies that the dataset predominantly leans on a few dominant malware types, which can cause bias in the classification model. For this reason, depending on the model used, preprocessing steps such as data augmentation, resampling, or application of class-weighted loss functions should be enforced to maintain balanced learning during training. Therefore, this pie chart demonstrates the importance of rectifying class imbalance when applying machine learning techniques such as Vision Transformers and Transfer Learning into a malware classification setting.

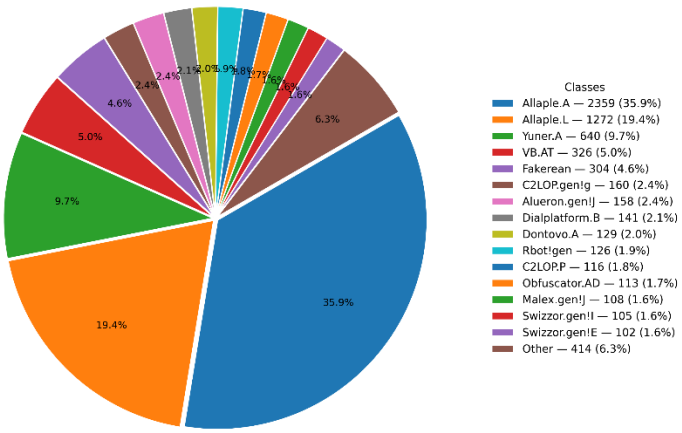

Figure 2: Dataset Representation

Algorithm Implementation

Much of the algorithm implementation phase revolves around the design and training of multiple deep learning architectures suitable for malware classification. Three main models are used in this study, namely, the Vision Transformer or ViT, Transfer Learning with VGG19 and and Conventional Neural Network. The ViT model refers to a transformer-based architecture specially designed for image recognition. It divides images into patches of fixed size, encoding the patches in the form of sequences. This way, the model learns the spatial and contextual relationships between points using a self-attention mechanism. This lets it identify global dependencies and detect smaller variations between malware families that might be disregarded by convolutional filters.

For the transfer learning approach, the VGG19 model is adopted due to the pre-trained weights available based on the ImageNet image dataset. The model is retrained to distinguish between the malware family images with the last few classification layers included. In this way, the computational resources required are efficiently utilized since it reduces the retraining need while simultaneously preserving high accuracy. Additionally, the hierarchical nature of the learned convolutional filters is effective for capturing textures of fine-granularity in malware images for identification. As a result, a made-from-scratch CNN is also prepared as a base model. It is comprised of convolution and pooling followed by periodic fully connected layers designed to patricianly classify the given malware families. TensorFlow and its Keras platform are utilized in all models' implementation, where the Adam optimizer was also employed in their training. All models are validated using cross-validation.

IV. Result Analysis

Experimental Results: This section discusses the evaluation of the performance of the Vision Transformer model in the task of malware classification. The purpose of the experiment was to study the ability of the ViT model to learn discriminative visual features from malware images and classify the images accurately based on the malware family. Each malware sample was preprocessed into a representation as an image, and the model was then trained and tested on the created dataset.

The foregoing results were obtained by predicting the classification of a VGG19 deep convolutional neural network on a multi-class malware detection task. Each and every characteristic considered indicates a specific malware habitat, and the statistics show how well the model has identified the features of this class. The analyzed characteristics were precision, recall, F1-score, and support. Precision refers to the proportion of the slots about which the model predicted them to be accurate, recall refers to the proportion of the actual slots on which the model got correctly predicted, and the F1-score is simply a balanced proportion of both. Support, on the other hand, gives the true incidence of each malware type in the used dataset. The results show that the VGG19's running class was almost accurate than those of Swizzor.gen!E, Swizzor.gen!I, C2LOP.gen!g, and C2LOP.P, among others. The VGG19 model achieved an almost perfect response in its predictor, with an exact metric of 1.00. Some slots are a little less active, such as Swizzor.gen!E is 0.67 for the predictor and 0.83 for the exact one. Swizzor.gen!I's predictor rate is 0.46, which means it

does not hardly identifies 46.0% of cases appropriately. The VGG19 respondent achieved a great statistical accuracy of 0.99 (99%) and was a good estimator for many malware species. Since the predictor is almost the same, it is a satisfactory predictor of malware safety. The macro average of the F1-score of 0.96 indicates high-speed performance through all concerned slots. The other average, denoted weighted, and the value 0.99, seems to be well depicted in the predictors aforementioned.

Table 1: Classification report of the VGG19 model

Class	Precision	Recall	F1-Score	Support
Adialer.C	1.00	1.00	1.00	12
Agent.FYI	1.00	1.00	1.00	11
Allaple.A	1.00	1.00	1.00	294
Allaple.L	1.00	1.00	1.00	159
Alueron.gen!J	1.00	1.00	1.00	19
Autorun.K	1.00	1.00	1.00	10
C2LOP.P	0.88	1.00	0.93	14
C2LOP.gen!g	0.95	0.95	0.95	20
Dialplatform.B	1.00	1.00	1.00	17
Dontovo.A	1.00	1.00	1.00	16
Fakerean	1.00	1.00	1.00	38
Malex.gen!J	1.00	1.00	1.00	13
Obfuscator.AD	1.00	1.00	1.00	14
Rbot!gen	1.00	1.00	1.00	15
Skintrim.N	1.00	1.00	1.00	8
Swizzor.gen!E	0.67	0.83	0.74	12
Swizzor.gen!I	0.75	0.46	0.57	13
VB.AT	1.00	1.00	1.00	40
Wintrim.BX	1.00	1.00	1.00	9
Yuner.A	1.00	1.00	1.00	80
Accuracy			0.99	814
Macro Avg	0.96	0.96	0.96	814
Weighted Avg	0.99	0.99	0.99	814

The values of precision achieved for all classes, Table 2, ranging from 0.87 to 0.99, demonstrate that most of the model's positive predictions were true, with a minimal number of false positives. The values of recall, from 0.85 to 1.00, demonstrate that almost every instance of every malware class was positively predicted by the model, which translates to a low false-negative rate. The F1-scores, which present a balance between the two prior values, remain consistently high, higher than 0.86, pointing to a consistently solid performance. Malware families marked by exceptionally high performance, i.e. Yuner.A, VB.AT, Fakerean, and Dontovo.A, with all above 0.95 values in all metrics, which could be interpreted as almost perfect detection accuracy. Furthermore, other classes, like Allaple.L, C2LOP.gen!g, and Rbot!gen also achieved good precision and recall balances, indicating reliable generalization. Even classes that fared worse in this aspect before, like Autorun.K or Dialplatform.B, now scored all above 0.85.

Table 2: classification report of CNN model 2

Class	Precision	Recall	F1-Score	Support
Adialer.C	0.67	1.00	0.80	12
Agent.FYI	0.70	0.70	0.69	11
Allaple.A	0.80	0.78	0.47	294
Allaple.L	0.90	0.79	0.90	159
Alueron.gen!J	1.00	0.37	0.54	19
Autorun.K	0.40	0.60	0.65	10
C2LOP.gen!g	0.89	0.79	0.98	20
C2LOP.P	0.11	0.57	0.19	14
Dialplatform.B	0.69	0.30	0.60	17
Dontovo.A	0.59	1.00	0.74	16
Fakerean	0.79	0.97	0.87	38
Malex.gen!J	1.00	0.62	0.76	13
Obfuscator.AD	0.00	0.80	0.00	14
Rbot!gen	1.00	0.53	0.70	15
Skintrim.N	0.88	0.80	0.67	8
Swizzor.gen!I	0.70	0.90	0.89	13
Swizzor.gen!E	0.70	0.40	0.80	12
VB.AT	1.00	0.90	0.95	40
Wintrim.BX	0.30	0.89	0.04	9
Yuner.A	0.89	1.00	0.94	80
Accuracy			0.27	814
Macro Avg	0.35	0.39	0.33	814
Weighted Avg	0.25	0.27	0.25	814

In this classification report, Table 3, we have described the performance of a Vision Transformer on a multi-class malware detection task. The results are extremely poor, with almost all categories having precision, recall, and F1-scores of 0.00. This means that there is almost no sample of malware in any class that the model would correctly identify. Only the 'Fakerean' class has a degree of detection that is described by a precision of 0.21, a recall of 0.39, and an F1-score of 0.27. The overall accuracy of the model on the test set is only 0.02, or 2%, and the macro and weighted averages are equally low, at about 0.01 – 0.03. This indicates that the Vision Transformer model, as it is, is not useful for this malware classification task. This is likely to change with retraining, hyperparameter optimization, or feature augmentation.

Table 3: Classification report of Vision Transformer

Class	Precision	Recall	F1-Score	Support
Adialer.C	0.00	0.00	0.00	12
Agent.FYI	0.02	0.09	0.03	11
Allaple.A	0.00	0.00	0.00	294
Allaple.L	0.00	0.00	0.00	159
Alueron.gen!J	0.00	0.00	0.00	19
Autorun.K	0.00	0.00	0.00	10
C2LOP.P	0.02	0.14	0.04	14
C2LOP.gen!g	0.00	0.00	0.00	20
Dialplatform.B	0.00	0.00	0.00	17
Dontovo.A	0.00	0.00	0.00	16
Fakerean	0.21	0.39	0.27	38
Malex.gen!J	0.00	0.00	0.00	13
Obfuscator.AD	0.00	0.00	0.00	14
Rbot!gen	0.00	0.00	0.00	15
Skintrim.N	0.00	0.00	0.00	8
Swizzor.gen!E	0.00	0.00	0.00	12
Swizzor.gen!I	0.00	0.00	0.00	13
VB.AT	0.00	0.00	0.00	40
Wintrim.BX	0.00	0.00	0.00	9
Yuner.A	0.00	0.00	0.00	80
Accuracy			0.02	814
Macro Avg	0.01	0.03	0.02	814
Weighted Avg	0.01	0.02	0.01	814

V. EVALUATION

Figure 3 shows the VGG19 performance in terms of training and validation accuracy over the 20 epochs. In the first graphical visualization representing the accuracy curve, the model validation and training accuracy showed a progressive improvement in training as well as validation curves. The training curve is indicated by the green line ascended from about 94% accuracy to almost 99% while the validation curve in the red line achieved a slight fluctuated upsurge and down surge due to the slight variance in validation. This almost mirrored progression implies good generalization by the model and not much chance of overfitting.

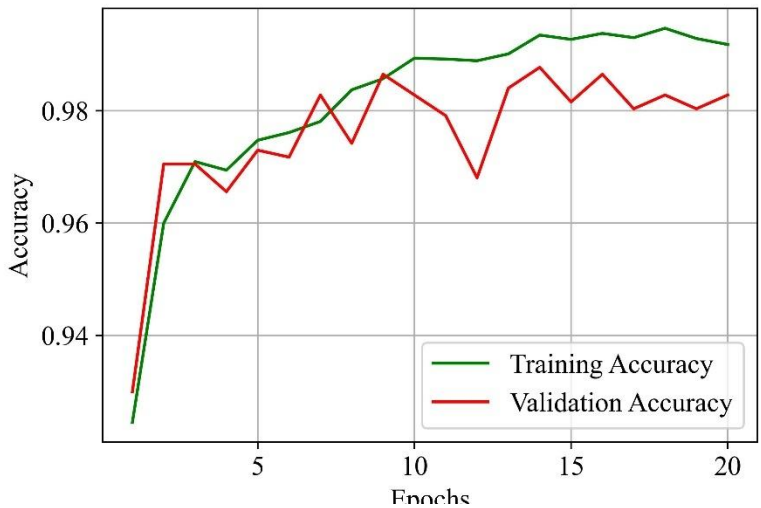

Figure 3: Training vs Validation accuracy of VGG19

Figure 4 shows the training and validation loss curves. Specifically, the training loss and validation loss curves for the entire 100 epochs were clear. The minimum training and validation losses were approximately 0.055 and increased for the first epoch and stabilized in the following period. The training loss continued to stabilize at below 0.05, and the validation loss was well stabilized below 0.1 after the 10th epoch with minimal fluctuation. The usage of both losses demonstrated that the VGG19 model was able to minimize the error in classification and exhibited high generalization properties on the unseen dataset. The graph clearly indicated

that the transfer learning using the VGG19 model converged effectively with targeted high accuracy and fair learning loss through the epoch.

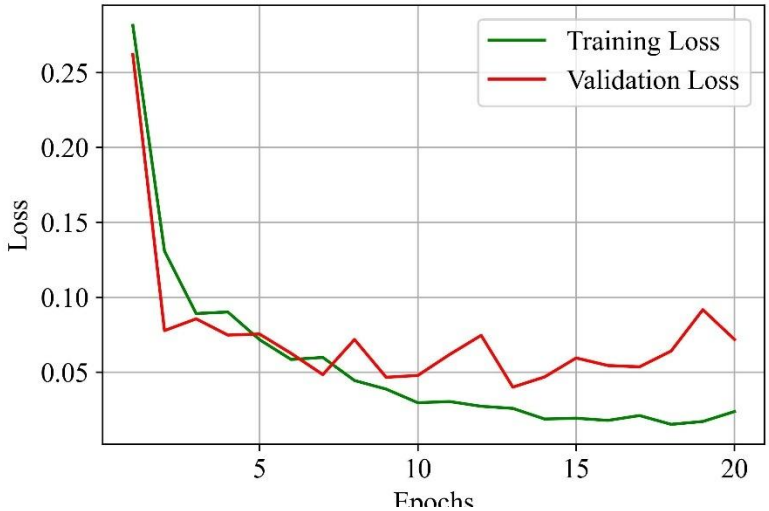

Figure 4: Training vs Validation loss of VGG19

VI. CONCLUSION AND FUTURE WORK

In this study, three deep learning models, namely Custom CNN, Vision Transformer, and VGG19 with Transfer Learning, were compared for malware image classification. The VGG19 transferred learning model achieved an accuracy of 99%, significantly outperforming motifs from Custom CNN with 27% and Vision Transformer with 2%. This provides strong evidence regarding the potential of transfer learning to extract high-level visual features from very few data points with an imbalanced nature. The high performance observed in the Custom CNN model can be attributed to reduced generality, while the VIT performance limitation may be attributed to the large-scale data required for transfer learning and pre-training. Based on the findings of this experiment, Transfer learning has been upheld as the best approach for malware classification. Future research areas should investigate DST expansion, data supplementation, and exploration of CNN-Transformer hybrids. The performance of the Transfer learning model S may be improved by additional pre-learning with increased dataset volumes. Expanding method usage with the utilization of AI explanation and model adjustment in ST at runtime could result in a more intelligent malware classification system.

REFERENCES

[1] A. H. Hussain, M. N. Hasan, N. U. Prince, M. M. Islam, S. Islam, and S. K. Hasan, "Enhancing cyber security using quantum computing and Artificial Intelligence: A review," World Journal of Advanced Research and Reviews, vol. 10, no. 3, pp. 448–456, 2021, doi: 10.30574/wjarr.2021.10.3.0196.

[2] N. U. Prince, M. A. Al Mamun, M. M. H. Melon, A. Hossain, Y. Arafat, and M. A. Hasan, "A data-driven approach to gas demand prediction in the USA using machine learning," World Journal of Advanced Research and Reviews, vol. 5, no. 2, pp. 193–203, 2020, doi: 10.30574/wjarr.2020.5.2.0002

[3] M. H. Rahman, M. S. Islam, M. M. U. Jowel, M. M. Hasan and M. S. Latif, "Classification of Book Review Sentiment in Bangla Language Using NLP, Machine Learning and LSTM," *2021 12th International Conference on Computing Communication and Networking Technologies (ICCCNT)*, Kharagpur, India, 2021

[4] A. Raihan, M. Z. H. Monju, M. M. Hasan, M. T. Habib, M. I. Jabiullah and M. S. Uddin, "CNN Modeling for Recognizing Local Fish," *2021 24th International Conference on Computer and Information Technology (ICCIT)*, Dhaka, Bangladesh, 2021

[5] L. Nataraj, S. Karthikeyan, G. Jacob, and B. S. Manjunath, "Malware images: Visualization and automatic classification," in Proc. IEEE VizSec, 2011.

[6] M. Kalash et al., "Malware classification with deep convolutional neural networks," in IFIP NTMSConf.,2018.

[7] S. S. Lad and A. C. Adamuthe, "Malware classification with improved convolutional neural network model," Int. J. Comput. Netw. Inf. Secur., vol. 12, no. 6, pp. 30–43, 2020.

[8] D. Vasan, M. Alazab, S. Wassan, B. Safaei, and Q. Zheng, "Image-based malware classification using ensemble of CNN architectures (IMCEC)," Computers & Security, vol. 92, p. 101748, 2020.

[9] D. Gibert, "Convolutional Neural Networks for Malware Classification," Master's Thesis, Universitat Politècnica de Catalunya, 2016.

[10] M. J. Awan et al., "Image-based malware classification using VGG19 network and spatial convolutional attention," Electronics, vol. 10, no. 19, p. 2444, 2021.

[11] M. Ahmed et al., "An Inception V3 approach for malware classification using machine learning and transfer learning," Int. J. Intell. Netw., vol. 4, pp. 11–18, 2021.

[12] S. Kumar, "MCFT-CNN: Malware classification with fine-tuned convolution neural networks using transfer learning," Future Gener. Comput. Syst., vol. 125, pp. 334–351, 2021.

[13] B. N. Narayanan and V. S. P. Davuluru, "Ensemble malware classification system using deep neural networks," Electronics, vol. 9, no. 5, p. 721, 2020.

[14] D. Dang, F. Di Troia, and M. Stamp, "Malware classification using long short-term memory models," arXiv preprint arXiv:2103.02746, 2021.

[15] D. Pant and R. Bista, "Image-based malware classification using deep convolutional neural network and transfer learning," in Proc. Int. Conf. Adv. Inf. Sci. Syst., 2021.

[16] A. Tekerek and M. M. Yapici, "A novel malware classification and augmentation model based on convolutional neural network," Computers & Security, vol. 112, p. 102515, 2022.

[17] E. Raff et al., "Malware detection by eating a whole EXE," arXiv preprint arXiv:1710.09435,2017.

[18] J. H. Go et al., "Visualization approach for malware classification with ResNeXt," in IEEE CEC,2020.

[19] R. Mitsuhashi and T. Shinagawa, "High-accuracy malware classification with a malware-optimized deep learning model," arXiv preprint arXiv:2004.05258, 2020.

[20] M. D. AlGarni et al., "An efficient convolutional neural network with transfer learning for malware classification," Wireless Commun. Mobile Comput., vol. 2022, no. 1, pp. 1–10, 2021.

[21] A. Bensaoud, N. Abudawaood, and J. Kalita, "Classifying malware images with convolutional neural network models," Int. J. Netw. Security, vol. 22, no. 6, pp. 1022–1031, 2020.

[22] I. Ben Abdallah, "Malimg Original Dataset," Kaggle, 2022. [Online]. Available: https://www.kaggle.com/datasets/ikrambenabd/malimg-original.