IVMCT: Image Visualization based Multiclass Malware Classification using Transfer Learning

[1]Manish Goyal, [2]Raman Kumar
Research Scholar, Department of Computer Science and Engineering, I K Gujral Punjab Technical University, Kapurthala, Punjab, Pincode:144603, India., Email-id: er.manishgoyal.ghudda@gmail.com
Assistant Professor, Department of Computer Science and Engineering, I K Gujral Punjab Technical University, Kapurthala, Punjab, Pincode:144603, India., Email-id: er.ramankumar@aol.in

Article Info
Page Number: 42 – 50
Publication Issue:
Vol 71 No. 2 (2022)

Article History
***Article Received**: 05 December 2021*
***Revised**: 12 January 2022*
***Accepted**: 02 February 2022*
***Publication**: 10 March 2022*

Abstract

Computer systems have made it possible to transfer human life from the real world to virtual reality. This process has been accelerated by the Covid-19 virus. Cybercriminals have also switched from a real-life to a virtual one. Online, committing a crime is far easier than in real life. Cybercriminals often use malicious software (malware), to launch cyber-attacks. Apart from this polymorphic and metamorphic malware are used that use obfuscation techniques to create new malware variants. To effectively battle new malware types, you'll need to employ creative approaches that depart from the conventional. Traditionally signature-based techniques are used with machine learning algorithms to detect malware that is unable to catch its variants. Deep learning (DL), which differs from typical machine learning methods, might be a potential approach to the challenge of identifying all varieties of malware. In the present study, an IVMCT framework is introduced which classifies malware using transfer learning. For this purpose, the MalImg dataset is used which is based on grayscale images converted from binaries of malware. The comparison of IVMCT is done with existing techniques which shows that our technique is better than existing techniques.

Keywords: Malware classification, Deep learning, CNN, Resnet, Alexnet, Densenet, transfer learning.

1. Introduction

Malware is software and code that can cause damage to systems. There are many kinds of malware today that aim to infect all information systems. It is crucial to be able to recognize them and avoid them to reduce the risk. There are several types of malware: trojans, worms, spyware and ransomware. Kaspersky has reported a dramatic increase in malware infections in recent years. In 2020, Kaspersky experienced 666,809,967 attacks worldwide. Also, according to Kaspersky 549,301 attacks were only ransomware [1]. This huge number of attacks makes everyone concerned about the security of data. Malware detection has become an important field of research. Malware detection/classification is a critical stage. Mainly malware analysis is done statically and dynamically[2]. Static analysis allows you to identify malware signatures without actually running malicious code. Although this method is relatively inexpensive, it can be very difficult to analyze malware that uses polymorphism or obfuscation packaging. Dynamic analysis refers to the execution and evaluation of malware behavior. Malware is executed in a controlled environment, such as sandboxes and virtual Computers[3], to guard against an attack.

The phrase "deep learning" refers to the techniques used to learn in multi-layer networks[4]. If more than 2 layers are added to the neural network it is converted into a deep neural network. These networks have gained great popularity recently. Well, there are two basic reasons behind this: (1) the amount of computing power required to train this network and (2) the problem of vanishing gradients

Deep learning's effectiveness is largely due to the inadequate of a manually designed feature extractor. Deep learning methods can automatically extract the characteristics or representations needed for classification from raw data[5]. To put it in other words, the feature extractor is created by the algorithm. These feature extractors minimize resources, are less likely to make mistakes, and, most importantly, can extract specific high-dimensional qualities that humans cannot envisage[6].

A visualization approach is also useful for malware classification. Malware visualization has been utilized in recent research to categorize malicious software [5]. This method analyzes every file in the malware executable. In the case of visualization, binaries of malware are arranged into bytes then is converted into an image[7]. A specific pattern is observed in each malware family which can be helpful in training. The training classification might be applied once the malware image has been visualized, utilizing the texture characteristics from the malware image. In image classification, convolutional neural networks (CNNs), one of the most well-known neural network models, are utilized. The input image is transformed to a pixel array first. To generate a predicted output, several layers of convolutional processing are applied to the image. The use of well-illustrated datasets is required to train the CNN model [8]. The CNNs' knowledge can be used to improve the detection and classification of malware[9].

The organization of the paper is as followed by the introduction section 2 contains a literature survey in which work done by various authors for malware classification is explained. Then MalImg dataset is briefly explained in Section 3 after the proposed architecture of IVMCT is explained in section 4 which is followed by the result section. Finally, a conclusion and future scope are given to conclude the paper.

2 Literature Survey

Rezende et al. [10] used ResNet-50 to develop a neural network design that included transfer learning. They used RGB images with a dimension 224 and 224 respectively, with 10 folds. They used the Glorot uniform approach. After 750 training sessions, the model was accurate to 98.62%. GIST tools were used with kNN, where k = 4, to obtain a precision of 97.48%. Additionally, bottleneck features were employed to achieve 98.0% accuracy. Gibert et al. [11] To create a file-agnostic deep learning scheme to detect patterns in visuals of malware binaries. The proposed scheme uses patterns to identify malicious software and allows it to be classified in a live environment. The malware visualization process could include other features, beyond patterns. Vasan et al [12] proposed another combination of CNN models. In this case, the authors have used ResNet50 to extract features. In the CNN architecture ensemble, SoftMax and Multiclass SVMs were deployed as classifiers. To reduce the dimensionality, a PCA process (principal component assessment) was applied. A fusion process was then used. The authors of [13] looked into an alternative scale of categorization. They turned malicious APK files into colorful graphics in this case. Other approaches, like Long Short-Term Memory [14], can be used in collaboration with CNN. Proposed a technique that implemented an ensemble classification scheme using both the compiled and assembled malware files. Convolutional and recurrent neural networks were used. CNN was utilized to classify the malware image, whereas the LSTM was used to classify harmful software assembly files. The authors of [15] developed a visual malware categorization method based on ANN. The suggested classification approach employed characteristics collected from the Malimg Database to train ANN.

Nataraj et al. [16] used KNN with Euclidean distance to extract GIST texture characteristics from displayed grayscale images and classify malware. For malware classification, their methodology was less computationally expensive than the n-gram method. Han et al. [17]suggested a technique for producing entropy graphs from grayscale images using an automated analysis approach. But their technique lacks in detecting packed malware as patterns were not observable due to high entropy measures. Kancherla et al. [9] used binary images to extract Gabor, intensity, and wavelet characteristics. They used a method that was resistant to code obfuscation.

To achieve better ensembling, Liu et al. [18] suggested a method in which they used the local mean method and lowered the size of an image. Fu et al. [19] used RGB images to display malware and extract global texture and color properties. Local characteristics were also retrieved from the code and data segments. To achieve successful malware categorization, they used a combination of global and local characteristics. Conti et al. [20] integrated various intriguing visualization tools within a hex editor-like interface. The three visuals work together to improve the analysts' workflow. The 'Byteview' visualization shows a whole file at a single glance, with each byte represented by a pixel. Even though both code bytes and image pixels are in a hexadecimal range from 00 to FF, this is possible. The hex value of the matching byte determines the intensity of each pixel. As a result, pictures produced by comparable code sequences are similar.

Ni, Qian, and Zhang [21] also present a deep learning-based method for classifying malware. For malware categorization, their program uses SimHash and CNN approaches. The program converts malware codes into grayscale pictures using the SimHash technique and then uses CNN to classify their families. Various approaches, such as multi-hash, bilinear interpolation, and major block selection, can be used to increase performance.

Cui et. al. [22] offers a strategy for improving model accuracy by combining CNN with the Bat algorithm. Malicious code is converted into grayscale images using the implemented approach. The CNN and Bat techniques are used to classify images. This is done to balance data amongst virus families. This method has one flaw: they only tested it using one assessment criterion. Tobiyama et. al. [23] deploy a two-stage deep learning neural network for infection detection. The authors first created an image using the behavioral characteristics derived from the trained recurrent neural network. They employed CNN afterward to categorize the feature images.

Chen et. al. [24] investigated the weaknesses of CNN-based malware detectors in depth. The authors offered two ways for attacking malware detectors that had just been created. The authors also performed experiments with a pre-detection system to reject hostile instances, demonstrating its usefulness in enhancing malware detection with accuracy and speed. Omer Aslan et. al. [6] offers an optimized framework to help with malware classification. They used a hybrid model by combining two deep neural networks. The framework used by the authors is shown in Figure 2, is divided into four phases namely input, feature collection, training and testing phase. To construct a feature vector, the authors combine two pre-trained models with equal weights. The training procedure was then carried out to acquire a high accuracy rate. The following is a description of each stage: The pre-training procedure begins with the Resnet and Alex Net architectures being trained on the ImageNet dataset [25]. The features obtained from the Resnet and Alex Net designs are then concatenated in the second stage to form a feature vector. The dimensions of this produced vector are 4096. Then fully connected layers were applied to this concatenated feature vector. Finally, utilizing the comprehensive datasets as inputs to the trained model, experimental analysis of the proposed model was done. On the MalImg dataset, they were 97.78 percent accurate.

Muhammad Asam et al [26] selected CNNs, ResNet-18 & DenseNet201 for the DFSMC framework. The hybrid learning capacity of these two customized CNNs has been utilized in the Deep Boosted Feature Space-Based Malware Classification framework (DBFS-MC). The DBFS-MC technique customizes ResNet-18 (and DensNet-201) as well. They're TL-based, and they've been fine-tuned for our issue space. Deep features from the second layer classification layer are combined to produce a boosted feature set and provide SVM with them. Figure 4 shows an overview of the framework. Figure 3. This scheme uses the deep feature learning capability and discrimination power of SVM. The accuracy achieved using the DBFS-MC framework was 98.61%. Vasan et al. [27] firstly arrange malware binaries into 8bit vectors. Then organized this 8-bit vector into a 2-dimensional array. This two-dimensional array of 8-bit vectors is equivalent to a grayscale image. Afterward, the authors apply a color map on this 2 D array so that patterns can become visible. Then they used CNN on this color map image. The authors used the dropout layer and fully connected layer to get an output of malware families of 25 classes. IMCFN has a 98.82% accuracy rate.

3. MalImg Dataset

MalImg data set is firstly introduced by Natraj et. al. [16]. They were the first to convert malware binaries into images. They observed some specific patterns in images. MalImg is made up of 9339 Images and samples don't require any pre-processing to be applied to the image-based analysis. This dataset includes 25 malware families, which are shown in Figure 1.

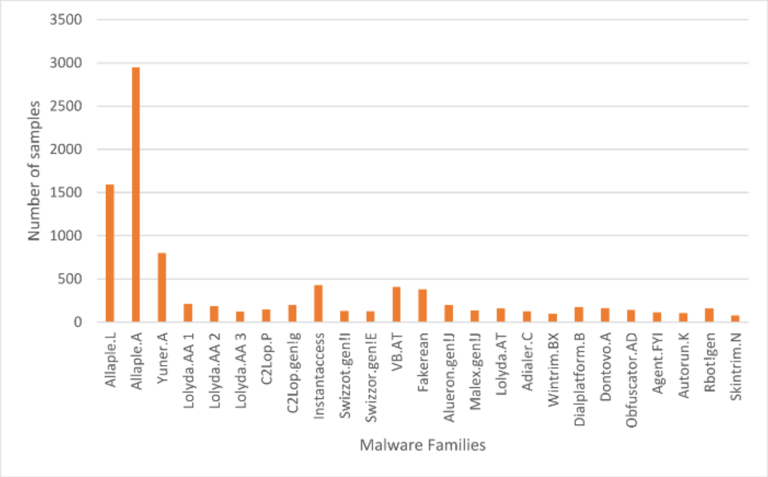

Figure 1: Number of samples in each malware family of MalImg dataset

4. Proposed Work

This section presents a framework for transfer learning for multiclass classification of malware. The proposed work provides a hybrid deep neural network[26]. This section is divided into 2 subsections namely malware visualization and architecture of the model used for this proposed work.

4.1 Malware Visualization

There are many ways to convert malware binary into grayscale images [28]. In the present study, malware executable binary file is converted into a grayscale image. The process of conversion of malware binary into a grayscale image is shown in Figure 2.

As indicated in Figure 2 firstly malware binaries are read as unsigned integers. Then these values are converted into binary values afterward these values are converted into pixel values. Consider we have binary values as

p_7, p_6, p_5, p_4, p_3, p_2, p_1, p0 from high significant bit to low significant bit respectively than a transformation of this binary value into pixel value is done using eq (1)[6].

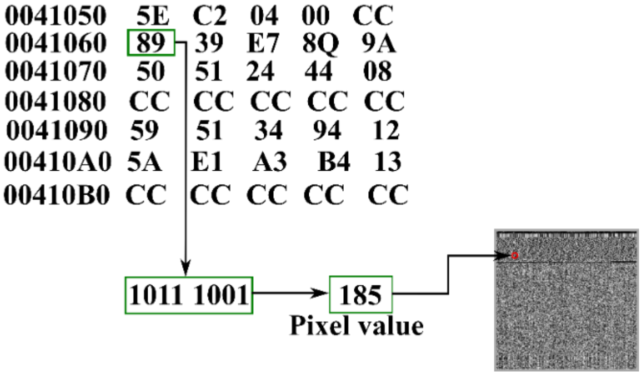

Figure 2: Conversion of binary to image

$$P = p_0 \times 2^0 + p_1 \times 2^1 + p_2 \times 2^2 + p_3 \times 2^3 + p_4 \times 2^4 + p_5 \times 2^5 + p_6 \times 2^6 + p_7 \times 2^7 \qquad (1)$$

[23]

Natraj et al [16] provided the MalImg dataset by using the transformation technique and various malware families generated by using this technique are shown in Figure 3.

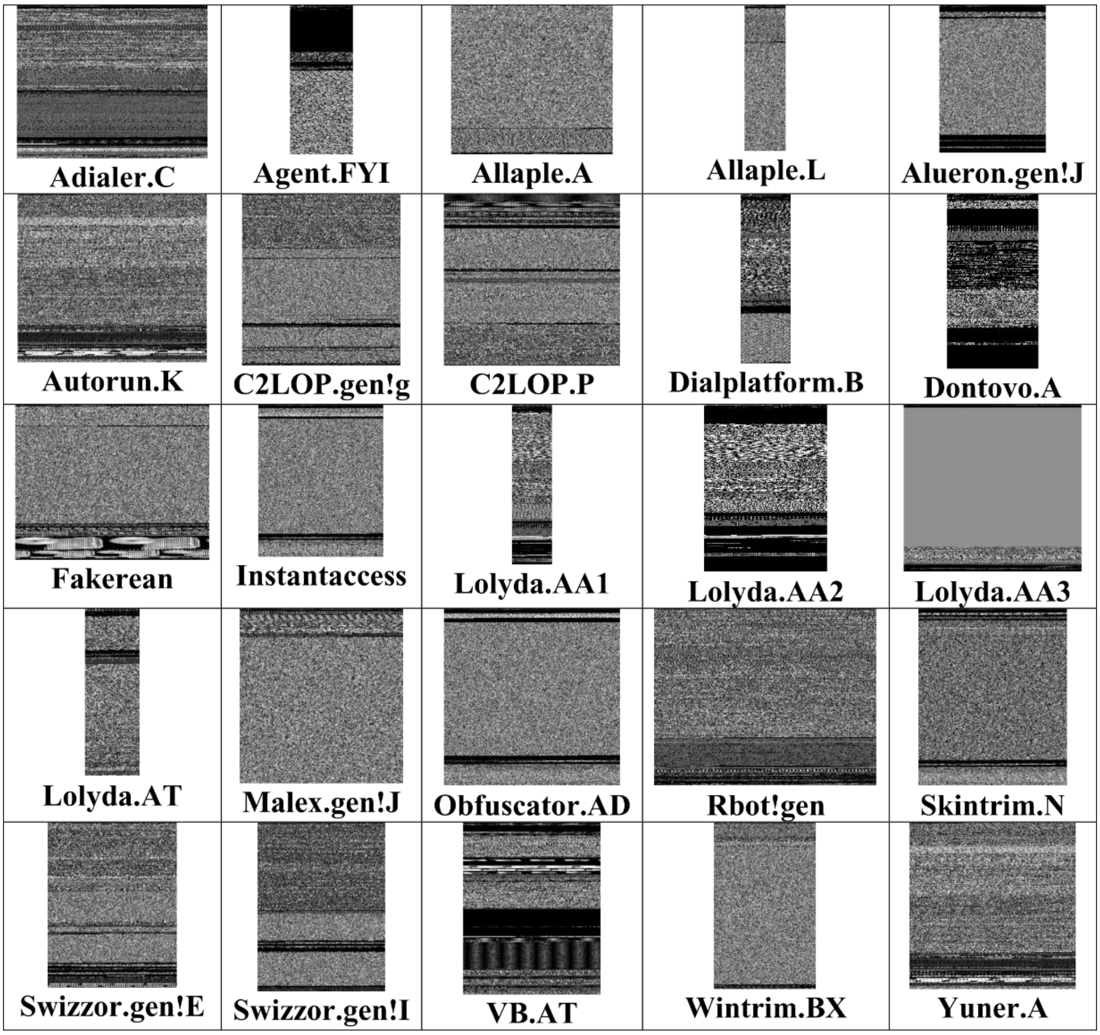

Figure 3: Various Malware families of MalImg Dataset

4.2 Architecture of IVMCT

The classification of malware is a challenging task. We have used a transfer learning-based hybrid model to classify malware. In the aforesaid IVMCT framework, we selected 3 pre-trained models namely ResNet, Alexnet and DenseNet. AlexNet is designed by Alex Krizhevsky et.al. [29] consists of 8 layers. The First 5 layers are convolution layers while the other 3 are dense layers. Dense net -201 connects each layer in a feed-forward manner[26]. The features of previous layers are used as input to the next layer while RestNet -152 was released in 2015[30]. The ResNet models have produced very good results in ImageNet and MS-COCO competition. The ResNet model improves gradient flow and hence provides very deeper training. The architecture of the framework used for IVMCT is shown in Figure 4. We have used transfer learning. Transfer learning is predominantly used these days. As deep learning needs a cluster of computers, a large amount of training data and is a lot of time-consuming transfer learning can solve these issues. As transfer learning uses pre-trained models these models can be extended to other applications also. The pre-trained models are trained in some domains and weights of models are adjusted accordingly these models are extended in other domains as well with fewer number resources and time is also consumed less. We have used 3 pre-trained models namely Alex Net, ResNet-152 and Dense Net-201. These models are chosen wisely and feature-based transfer learning is used to combine features of these models. Afterward, fully connected layers are used and the softmax activation function is used during the final output.

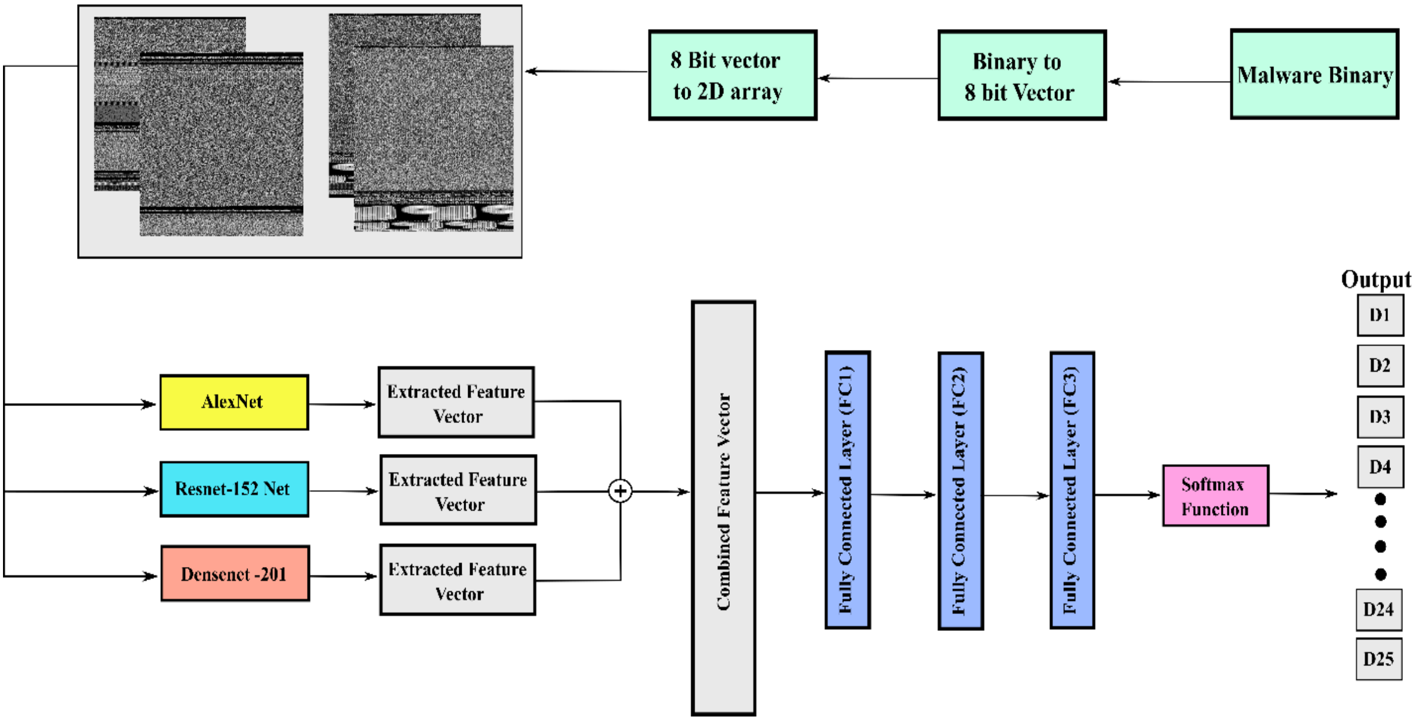

Figure 4: Architecture of IVMCT

5. Results

The implementation of the present work is done in the MalImg dataset. The dataset consists of 9339 malware samples from 25 malware families[16]. The accuracy achieved by various researchers and our work are listed in Table 1.

Table 1: Comparison of MalImg dataset with existing techniques.

Technique name	Accuracy in %age)
OmerAslan et al	97.78
DBFS-MC framework	98.61
IMCFN	98.82
Jeyaprakash et al	98.23
Hemalatha et al	97.55
Roseline et al	98.65
IVMCT	99.12

The graphical representation of Table 1 is provided in Figure 6.

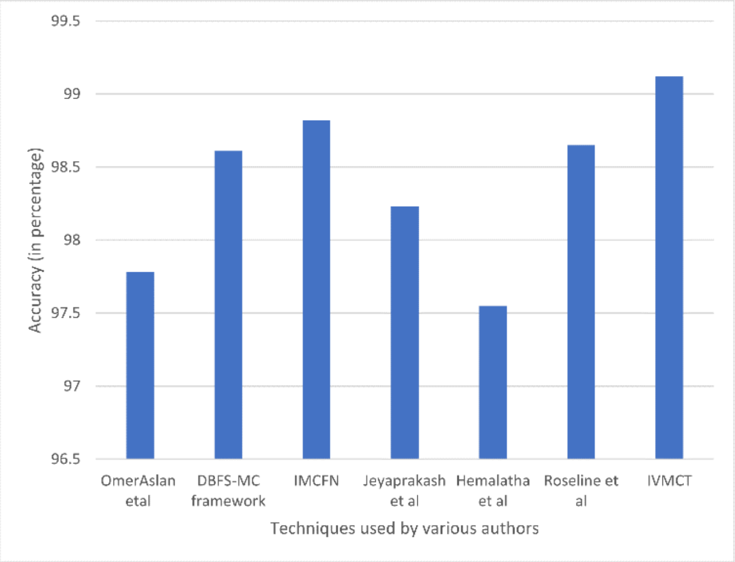

Figure 6: Graphical representation of accuracies of IVMCT with existing techniques.

As clearly indicated from Table 1 and Figure 6 IVMCT obtained the highest accuracy as compared to existing techniques.

6. Conclusion

It is always a race between antimalware software and cyber-attackers. Cyber-attackers must be kept under constant pressure. Malware is an attack vector that can be used to access the internet. We have proposed an IVMCT framework for the multi-class classification of malware using transfer learning on the MalImg dataset. Transfer learning outperforms traditional machine-learning techniques. The proposed solution has some advantages which allow malware to detect in a real-time environment. The first advantage is we have used malware binary which is needed to execute. Secondly, conversion of malware binary into a grayscale image is cost-effective. Thirdly, as malware binary is used without execution which is less time-consuming also the time is saved in using pre-trained models. The fourth advantage is our technique has outperformed previously used techniques.

7. Limitations and Future Scope

Despite so many advantages of classifying malware using IVMCT, there is one major limitation of it. As malware image is created from malware binary so it is incapable of detecting zero-day malware or obfuscated malware. As in case of obfuscation, dead code insertion, reordering of code, adding extra jump statements, etc is done. These changes in malware codes cause a change in malware binary so malware image is also changed. While dynamic malware analysis is slower than static and malware image-based techniques, it can still detect obfuscated malicious software. In the future, we would like to execute samples in an isolated environment and work on dynamic analysis to detect obfuscated malware as well.

Acknowledgment

Manish Goyal would like to thank I.K. Gujral Punjab Technical University for offering the Ph.D. course in Computer Science & Engineering and providing support to access the resources for research

References

[1] "KSB_statistics_2020_en.pdf." Accessed: Mar. 07, 2022. [Online]. Available: https://go.kaspersky.com/rs/802-IJN-240/images/KSB_statistics_2020_en.pdf

[2] M. Goyal and R. Kumar, "The Pipeline Process of Signature-based and Behavior-based Malware Detection," in *2020 IEEE 5th International Conference on Computing Communication and Automation (ICCCA)*, 2020, pp. 497–502.

[3] J. Hemalatha, S. A. Roseline, S. Geetha, S. Kadry, and R. Damaševičius, "An efficient densenet-based deep learning model for malware detection," *Entropy*, vol. 23, no. 3, p. 344, 2021.

[4] Z. Allen-Zhu, Y. Li, and Z. Song, "A convergence theory for deep learning via over-parameterization," in *International Conference on Machine Learning*, 2019, pp. 242–252.

[5] G. Sun and Q. Qian, "Deep learning and visualization for identifying malware families," *IEEE Transactions on Dependable and Secure Computing*, vol. 18, no. 1, pp. 283–295, 2018.

[6] Ö. Aslan and A. A. Yilmaz, "A new malware classification framework based on deep learning algorithms," *Ieee Access*, vol. 9, pp. 87936–87951, 2021.

[7] S. Ni, Q. Qian, and R. Zhang, "Malware identification using visualization images and deep learning," *Computers & Security*, vol. 77, pp. 871–885, 2018.

[8] A. Makandar and A. Patrot, "Malware class recognition using image processing techniques," in *2017 International Conference on Data Management, Analytics and Innovation (ICDMAI)*, 2017, pp. 76–80.

[9] K. Kancherla and S. Mukkamala, "Image visualization based malware detection," in *2013 IEEE Symposium on Computational Intelligence in Cyber Security (CICS)*, 2013, pp. 40–44.

[10] E. Rezende, G. Ruppert, T. Carvalho, F. Ramos, and P. De Geus, "Malicious software classification using transfer learning of resnet-50 deep neural network," in *2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA)*, 2017, pp. 1011–1014.

[11] D. Gibert, C. Mateu, J. Planes, and R. Vicens, "Using convolutional neural networks for classification of malware represented as images," *Journal of Computer Virology and Hacking Techniques*, vol. 15, no. 1, pp. 15–28, 2019.

[12] 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.

[13] H. Naeem *et al.*, "Malware detection in industrial internet of things based on hybrid image visualization and deep learning model," *Ad Hoc Networks*, vol. 105, p. 102154, 2020.

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

[15] V. Moussas and A. Andreatos, “Malware detection based on code visualization and two-level classification,” *Information*, vol. 12, no. 3, p. 118, 2021.

[16] L. Nataraj, S. Karthikeyan, G. Jacob, and B. S. Manjunath, “Malware images: visualization and automatic classification,” in *Proceedings of the 8th international symposium on visualization for cyber security*, 2011, pp. 1–7.

[17] K. Han, B. Kang, and E. G. Im, “Malware analysis using visualized image matrices,” *The Scientific World Journal*, vol. 2014, 2014.

[18] L. Liu and B. Wang, “Malware classification using gray-scale images and ensemble learning,” in *2016 3rd International Conference on Systems and Informatics (ICSAI)*, 2016, pp. 1018–1022.

[19] J. Fu, J. Xue, Y. Wang, Z. Liu, and C. Shan, “Malware visualization for fine-grained classification,” *IEEE Access*, vol. 6, pp. 14510–14523, 2018.

[20] G. Conti *et al.*, “A visual study of primitive binary fragment types,” *White Paper, Black Hat USA*, 2010.

[21] S. Ni, Q. Qian, and R. Zhang, “Malware identification using visualization images and deep learning,” *Computers & Security*, vol. 77, pp. 871–885, 2018.

[22] Z. Cui, F. Xue, X. Cai, Y. Cao, G. Wang, and J. Chen, “Detection of malicious code variants based on deep learning,” *IEEE Transactions on Industrial Informatics*, vol. 14, no. 7, pp. 3187–3196, 2018.

[23] S. Tobiyama, Y. Yamaguchi, H. Shimada, T. Ikuse, and T. Yagi, “Malware detection with deep neural network using process behavior,” in *2016 IEEE 40th annual computer software and applications conference (COMPSAC)*, 2016, vol. 2, pp. 577–582.

[24] Z. Chen and C. Ji, “Spatial-temporal modeling of malware propagation in networks,” *IEEE Transactions on Neural networks*, vol. 16, no. 5, pp. 1291–1303, 2005.

[25] “ImageNet.” https://image-net.org/ (accessed Mar. 07, 2022).

[26] M. Asam *et al.*, “Detection of exceptional malware variants using deep boosted feature spaces and machine learning,” *Applied Sciences*, vol. 11, no. 21, p. 10464, 2021.

[27] D. Vasan, M. Alazab, S. Wassan, H. Naeem, B. Safaei, and Q. Zheng, “IMCFN: Image-based malware classification using fine-tuned convolutional neural network architecture,” *Computer Networks*, vol. 171, p. 107138, 2020.

[28] S. Venkatraman, M. Alazab, and R. Vinayakumar, “A hybrid deep learning image-based analysis for effective malware detection,” *Journal of Information Security and Applications*, vol. 47, pp. 377–389, 2019.

[29] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” *Advances in neural information processing systems*, vol. 25, 2012.

[30] “ResNet-152 - Wolfram Neural Net Repository.” https://resources.wolframcloud.com/NeuralNetRepository/resources/ResNet-152-Trained-on-ImageNet-Competition-Data (accessed Mar. 07, 2022).