Visualization Approach for Malware Classification with ResNeXt

Jin Ho Go[1], Tony Jan[2], Manoranjan Mohanty[3], Om Prakash Patel[4], Deepak Puthal[5], Mukesh Prasad[1]

[1]Centre for Artificial Intelligence, School of Computer Science, FEIT, University of Technology Sydney, Australia
[2]School of Information Technology and Engineering, Melbourne Institute of Technology, Australia
[3]Center for Forensic Science, Faculty of Science, University of Technology Sydney, Australia
[4]Department of Computer Science and Engineering, Mahindra Ecole Centrale, Hyderabad, India
[5]School of Computing, Newcastle University, Newcastle upon Tyne, United Kingdom

***Abstract*: The Internet has resulted in cyber-threats and cyber-crimes, which can occur anywhere at any time. Among various cyber threats, modern malware with applied metamorphosis and polymorphic technology is a concern as it can proliferate to advanced variants from its original shape. The typical malware analysis methods, including signature-based approach, remain vulnerable to such advanced variants. This paper proposes a visualization-based approach for malware analysis using the state-of-the-art Convolution Neural Network (CNN) model such as ResNeXt, which had achieved outstanding performance in image classifications with competitive computational complexity. The proposed method transforms the attributes of raw malware binary executable files to greyscale images for further analysis by well-established deep learning models. The greyscale images, which result of data transformation for visualization, are classified using ResNeXt. The experiment results show that the proposed solution achieves 98.32% and 98.86% of accuracy in malware classification on Malimg dataset and modified Malimg dataset, respectively. The proposed method outperforms other comparable methods in terms of classification accuracy and requires similar level of computational power.**

***Keywords:* Malware, cybercrime, cyber threat, cybersecurity, intrusion detection system, convolutional neural network, visualization**

I. INTRODUCTION

Cybercrime has become one of the most severe problems in modern society [2]. Although the Internet has improved lifestyle, the risk of cyber threats to individuals and business houses has increased multi-fold. Some dangerous cyber-threats, such as malware attacks, are proven to be very critical for the organizations. The 2019 ACR reports by Cybersecurity Ventures state that cybercrimes will cause $6 trillion loss to the corporations in 2021 [3]. Some corporations are already wary of such threats, and using Intrusion Detection Systems (IDS) to monitor, detect, and prevent their computer network from cyber-threats. The modern cyber-threats, however, include advanced polymorphic malware attacks (which can change its attribute), which are mostly undetectable by traditional IDS.

The IDS are generally categorized as signature-based or anomaly-based approaches. The signature-based approach is further divided into signature or rule-based approaches. Signature-based IDS checks the signature database for known patterns of each specific intrusion threat. This type of IDS cannot detect an unknown or novel intrusion threat because the signature database is yet to be updated [4].

On the other hand, the rule-based approach follows 'if-then-else' rules that are formed based on the expert knowledge of cybersecurity personnel. The expert knowledge accumulates information from numerous past intrusion events. In a rule-based approach, an intrusion is typically monitored, and its attributes transformed into facts that are inferred to create rules [4]. On the downside, if the deduced rules are not appropriate, it can cause high false alarms. Besides, the expert rule-based model cannot respond appropriately to a new form of threat. Based on the above findings, both approaches are vulnerable to innovative malware that can cause intrusion. For example, attackers can use metamorphosis and polymorphic technology to generate malware that proliferates, leading to the increased variants of malware [6], as depicted in Fig. 1.

In these cases, we require new approaches in malware analysis that can detect malware with high accuracy but with controlled burden on system resource usage, this paper proposes one such approach. In the proposed method, a malware executable is represented as a binary value and transformed into a greyscale image [7]. To detect malicious threats, Convolutional Neural Network (CNN) is used to classify the greyscale images. CNN has achieved outstanding results in image classifications. This paper shows that CNN can be effectively used in detecting malicious threats from a set of grey scaled images that represents malware attributes. This paper uses the ResNeXt CNN model, which combines the feature of the ResNet and the Inception Net architecture for malware classification. In the ImageNet Large Scale Visual Recognition Challenge (ILSVRC), ResNeXt showed improved accuracy than ResNet and Inception Net [8].

The rest of the paper is organized as follows. Section II describes related work in texture-based, and visualization approaches in malware analysis, and also the feature of ResNet and Cardinality, which is a core concept in the ResNeXt. The proposed approach is presented in Section III. The detailed results are presented and discussed in Section IV. Finally, Section V concludes the paper.

978-1-7281-6929-3/20/$31.00 ©2020 IEEE

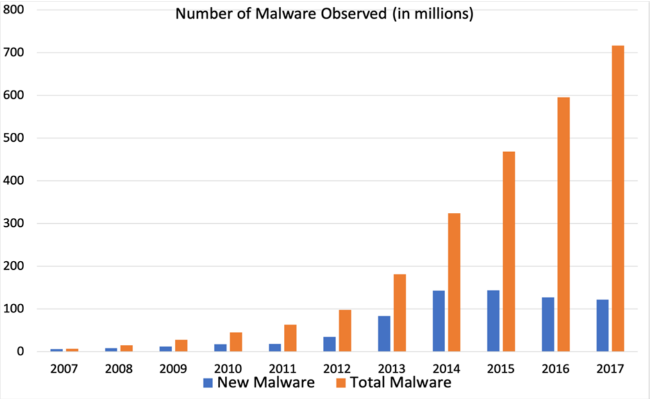


Fig. 1. Increasing the amount of malware (AV-test 2018) [1]

II. RELATED WORK

A) *Texted-based approach on malware analysis*

Malware analysis can be either static or dynamic [9]. In the Static approach, analyses of binary executable files is done without running them. The disadvantages of these static methods are in long-time consumption and inefficiency in carrying out the analysis [10]. The static analysis does not work in the cases of file obfuscating, encrypting or compression. Thereby, the static analysis is vulnerable to modern malware, which is generated by applying polymorphism and metamorphism [11]. Dynamic malware analysis, on the other hand, checks the binary files in execution. The main limitation of dynamic analysis is that it can usually monitor a single execution path; therefore, providing only partial coverage. If the dynamic analysis can be implemented on virtual machines, the virtual machine's settings may change, causing the malware to behave differently from its previous states. Also, bugs in the virtual machine environment can cause malware to infect the host device or other devices located in the network. The main limitation with texture-based image analysis of malware is that certain malware obfuscations cannot be easily overcome [12].

B) *Visualization approach in malware analysis*

Helfman suggested that visualization can assist in getting the comprehensive view of software by applying dotplot data visualization [13]. Fig. 2 shows that basic form of dotplot. As a visualization technique, when dotplot is implemented in the software field, it is useful to identify the patterns of overall uniformity as well as programming language syntax. Due to this feature, dotplot is helpful in software design through successive abstraction that reduces redundancy as a design method.

Automatic binary mapping through byte plot is suggested by *Conti et al.* [14]. Byte plot visualization is used to identify the internal structure of the binary files. Each byte in the target binary file is transformed into a pixel value, ranging from *0* to *FF*. *0* value represents white and *FF* value represents black. By implementing Byte plot, *Conti et al* automated process of detecting distinct sections within the binary file. Also, based on the result of Byte plot visualization, they predicted primitive format types [14]. A primitive type is defined as a homogeneous section of a binary file that has a related binary structure like random number sequences. The byte plot visualization helped in finding distinctive patterns, even when some encoding or encryption had been applied. Fig. 3 displays diverse grey scale sections within binary file through Byte plot technology.

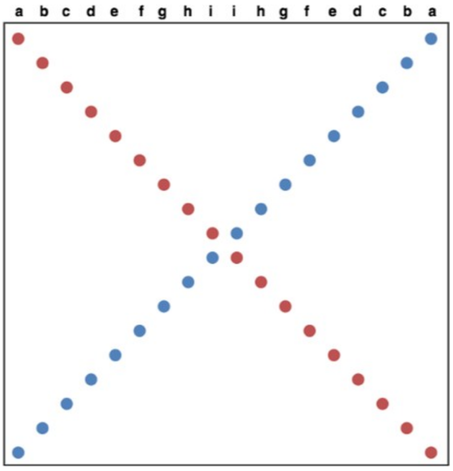


Fig. 2. Example of Dotplot [13]

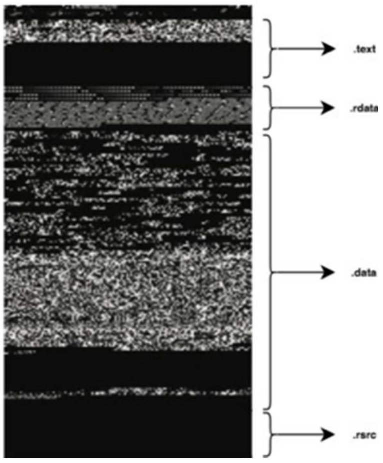


Fig. 3. Various sections within binary file with Byte plot [14]

C) *Residual network*

*He et a*l. suggested a residual network to solve the degradation of network performance. Degradation is defined as a deep networks showing lower performance than a system with fewer layers in training and testing [15]. They regard this degradation problem as a result of a failure in deep network optimization. To solve this problem, *He et al.* proposed residual mapping instead of original mapping. While the initial mapping learns all the inputs *x*, the residual mappings only learn the difference between the inputs and the deserved outputs [16]. In other words, the concept is to learn only the differences between the target and the input. To achieve this aim, they added '*identity shortcut connection'* on a native Convolution Neural Network (CNN). In the residual network, shortcut connection can skip one or more layers without learning undertaken. The experiments comparing the performance of residual network with the base CNN model proved that the residual network reduced degradation in deep learning.

D) Cardinality

Residual network enables the models with hundreds of layers to maintain competitive performance on diverse image processing tasks [17]. Residual network has attracted the attention of the research community and there have been many works with several variants of residual network derived. *Xie et a*l. proposed 'ResNext', which modified the residual network with 'Cardinality' concept. As a hyper-parameter, Cardinality represents the amount of self-standing paths, providing a new way of fine-tuning the model capacity [9]. The experiments have shown that increasing cardinality can be more effective than increasing the depth in terms of accuracy. The authors suggest that the ResNeXt model can more easily adapt to a new dataset in comparison to the other models, as *ResNeXt* requires only one hyper-parameter to tune [18], whereas the other models require optimization of multiple hyper-parameters.

III. METHODOLOGY

A) Transforming binary file to greyscale image

Fig. 4 presents the transforming Process from Malware binary file to GreyScale image. For the visualization of malware, the suspected executable files are converted into a sequence of binary values (ones and zeros) [7]. Then, the result of the conversion is split into 8 bits segments. These units represent the brightness of a pixel between 0 (black), minimum grey level, and 255 (white), maximum grey level in scale, and rearranged into a 2-dimensional array. The greyscale images which have a constant width and different height depending on the file size are generated as result of the process.

Fig. 5 displays the images of two different families of malware generated by malware transforming the process shown in Fig. 4. From these sample images, it can seen the structural similarities between malicious codes in the same family and some discrepancies between different classes of malware. The empirical evidence of clear distinction in visual patterns motivates to apply CNN in threat analysis, as CNN has been widely used in image processing with remarkable success.

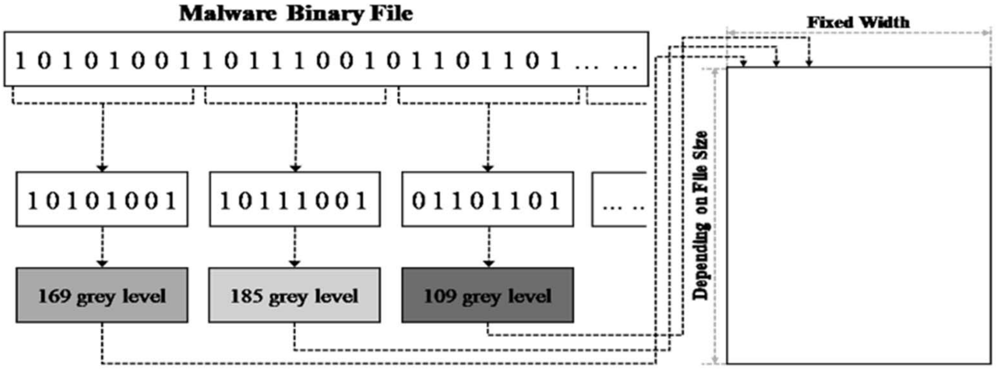
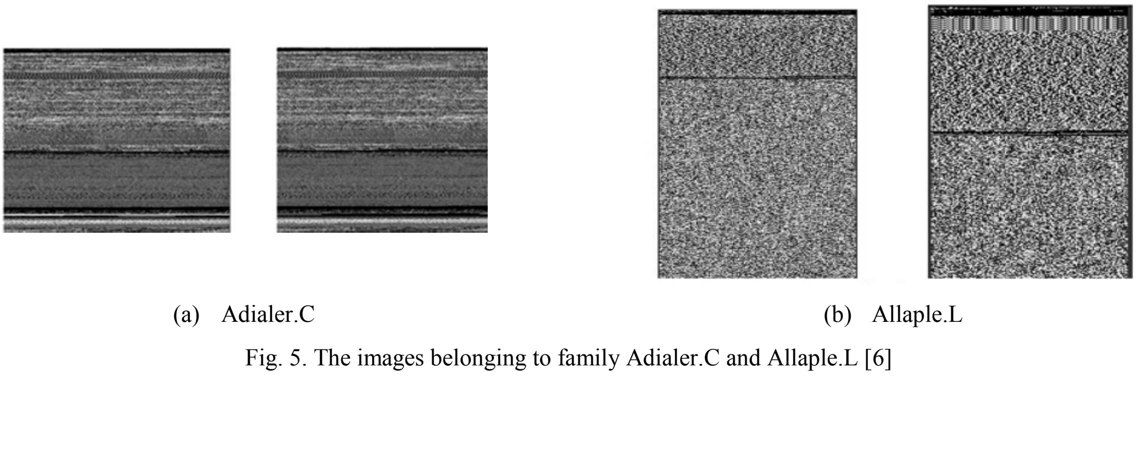


Fig. 4. Transforming Process from Malware Binary File to GreyScale Image



(a) Adialer.C

(b) Allaple.L

Fig. 5. The images belonging to family Adialer.C and Allaple.L [6]

B) *Proposed model*

For malware classification, this paper adopts ResNeXt proposed by Xie *at el.* [9]. As a variant of the deep residual network, ResNeXt replaces the basic residual block with aggregated residual block, in which the so-called '*split-transform-merge*' strategy is implemented in Inception architecture [9].

In an artificial neural network, the primary neurons produce an inner product, which is the weighted sum in each layer. The inner product is regarded as a form of aggregating transformation, as shown in Eq. (1).

$$\sum_{i=1}^{D} w_i\, x_i \tag{1}$$

where x_i is the *D*-channel input vector for the neuron, and w_i is the weight of the filter for the *i-th* channel. Xie *at el.* replaced basic aggregating transformation with a more inclusive function that can perform as a network itself [9]. They presented aggregated transformations, as shown in Eq. (2).

$$F(x) = \sum_{i=1}^{C} \mathcal{T}_i(x) \tag{2}$$

where $\mathcal{T}_i(x)$ is an arbitrary function. Analogous to a simple neuron, here $\mathcal{T}_i$ projects x into an (optionally low -dimensional) embedding and then transforms it. C represents the size of the set of transformations to be aggregated. Xie at el. refers C as cardinality. They suggest that the dimension of cardinality controls the number of more complex transformations. A block of ResNeXt with Cardinality equal to 32 is shown in Fig. 6. The aggregated transformation in Eq. 2 serves as the residual function, as shown in Eq. (3).

$$y = x + \sum_{i=1}^{C} \mathcal{T}_i(x) \tag{3}$$

where y is the output.

ResNeXt is designed by applying the skip concept from ResNet, which is one of basic CNN model, and cardinality with improved accuracy than a wide and deep network. ResNeXt requires low flops and showed high accuracy performance compared to existing models in experiments conducted using Imagenet datasets. Due to this advantages, this paper adopts ResNeXt50 for malware image classification, which is the field for demanding swift and explicit classification. The architecture of ResNeXt 50 is shown in Fig.7.

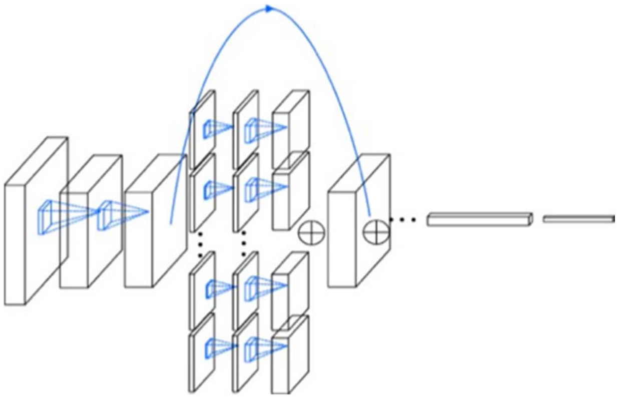


Fig. 6. A block of ResNeXt with Cardinality : 32 [18]

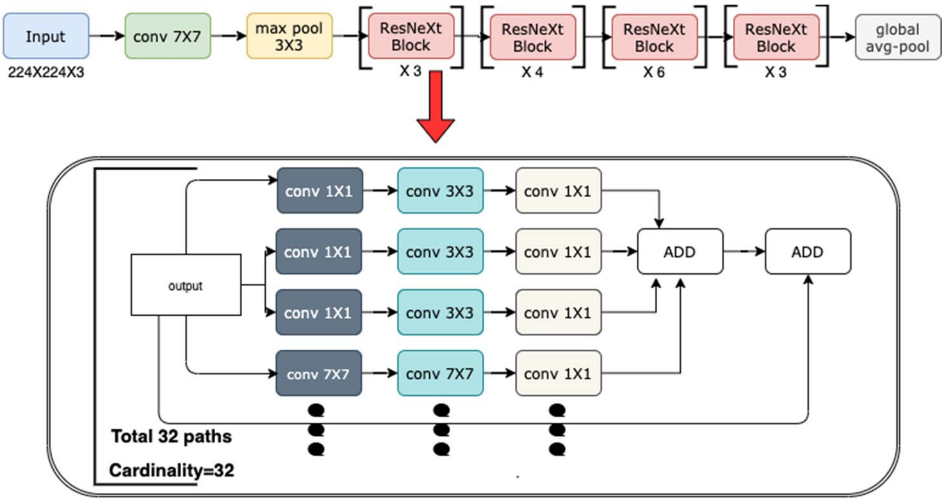


Fig.7. The architecture of ResNeXt50

IV. EXPERIMENTS

This paper focuses on two things in the experimental analysis. Firstly, we want to verify that the state-of-the-art technology yields a better result than previous other works in malware analysis (as per the performance outcomes in an image-based approach). Secondly, we attempted to verify and consolidate the experimental comparison using the most popular dataset, ImageNet.

A) Dataset

This paper uses the Malimg dataset, which was generated by Natataj *et al*. The Original dataset consists of 9,339 numbers of malware over 25 family classes [7]. This is an unbalanced dataset with 2,949 and 80 samples of Allaple.A and Skintrim.N, respectively. To account for the unbalance in the dataset, we added some samples gathered from VirusTotal [20], Malshare [21], and VirusShare [22] and the family of modified dataset describes in Table I. Fig. 8 shows the distribution of datset. Fig. 9 shows a comparison between the original dataset and Skintirm.N family class with added samples. Both feature selection and classification performance of the models are evaluated with 10-fold cross-validation mechanism and average resuts are presented.

Table I : Modified Malimg dataset

	Family	Number of samples
1	Adialer.C	160
2	Agent.FYI	169
3	Allaple.A	2,949
4	Allaple.L	1,591
5	Alueron.gen!J	198
6	Autorun.K	130
7	C2LOP.gen!g	200
8	C2LOP.P	162
9	Dialplatform.B	177
10	Dontovo.A	162
11	Fakerean	380
12	Instantaccess	431
13	Lolyda.AA1	213
14	Lolyda.AA2	184
15	Lolyda.AA3	154
16	Lolyda.AT	159
17	Malex.gen!J	164
18	Obfuscator.AD	165
19	Rbot!gen	158
20	Skintrim.N	124
21	Swizzor.gen!E	158
22	Swizzor.gen!I	172
23	VB.AT	408
24	Wintrim.BX	137
25	Yuner.A	800

B) Experiment result for ResNeXt

This paper applies ResNeXt to classify greyscale images generated by malware binary files. This paper uses ResNeXt 50, which consisted of 50 layers. Among 50 layers, 48 layers except for the input layer and full connected layer are divided into 16 blocks. Each block is composed of 3 layers and has a total of 32 cardinalities, which means the path to the next layer. The proposed model gave 98.8 6% accuracy on the malware classification task. Table II displays the precision, recall, and F1-score for each family. The confusion matrix helps to understand the result as shown in Fig. 10. Each row of the matrix represents a true class with each column of the matrix representing a predicted class, and the number of elements represents the correctly classified number of images.

Table II : Precision, Recall and F1-score of RESNEXT50

	Family	Precision	Recall	F1-score
1	Adialer.C	1	1	1
2	Agent.FYI	1	1	1
3	Allaple.A	1	0.996	0.997
4	Allaple.L	1	0.996	0.997
5	Alueron.gen!J	1	0.975	0.987
6	Autorun.K	1	0.909	0.952
7	C2LOP.gen!g	0.99	0.9	0.942
8	C2LOP.P	0.88	1	0.936
9	Dialplatform.B	1	1	1
10	Dontovo.A	1	0.987	0.993
11	Fakerean	0.97	1	0.984
12	Instantaccess	1	1	1
13	Lolyda.AA1	0.97	1	0.984
14	Lolyda.AA2	0.98	0.962	0.97
15	Lolyda.AA3	0.97	1	0.984
16	Lolyda.AT	1	1	1
17	Malex.gen!J	0.98	1	0.989
18	Obfuscator.AD	1	1	1
19	Rbot!gen	1	1	1
20	Skintrim.N	0.98	1	0.989
21	Swizzor.gen!E	0.84	0.904	0.87
22	Swizzor.gen!I	0.88	0.877	0.878
23	VB.AT	0.97	1	0.984
24	Wintrim.BX	1	0.978	0.988
25	Yuner.A	1	1	1

C) Comparison with previous study

To validate the proposed model, it compared with the other models including Nataraj et al. They first explored the application of byte plot visualization for automatic malware classification and used an abstract representation technique, GIST, for computing texture features from images [7]. They created a Malimg dataset based on data obtained from the Aubis system. Malimg dataset contains 9,458 malware images over 25 classes and achieved an accuracy of 97.18%. In order to get the results in a similar environment, this paper conductes the experiment using the same dataset and achieved an accuracy of 98.32%.

D) Comparison with other CNN models

The proposed model compared with other CNN models such as ResNet and Inception Net. In order to establish a comparable testing environment, we used the dataset that had added samples to the malimg dataset and tried to get as much information on *FLOPs* as an indicator of complexity. The malimg dataset with attached samples contained 9,713 malware images in 25 classes. The proposed ResNeXt50 performs better than ResNet and Inception v4 as shown in Table III.

Table III : FLOPs and Accuracy of ResNeXt50, ResNet50, and Inception-v4 on Modified dataset

	ResNeXt 50	ResNet 50	Inception v4
FLOPs	$4.2 \text{x} 10^9$	$4.1 \text{x} 10^9$	$4.5 \text{x} 10^9$
Accuracy	98.86%	97.97%	97.54%

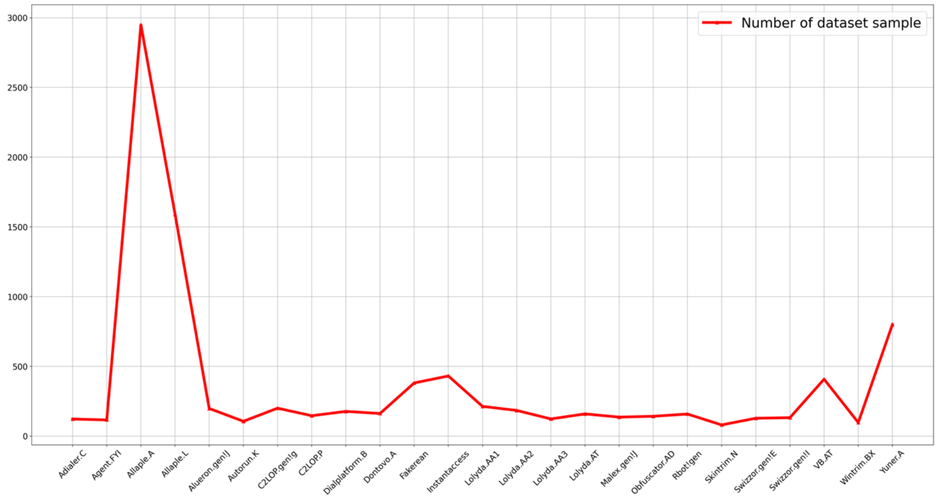


Fig. 8. Distribution of Malimg dataset

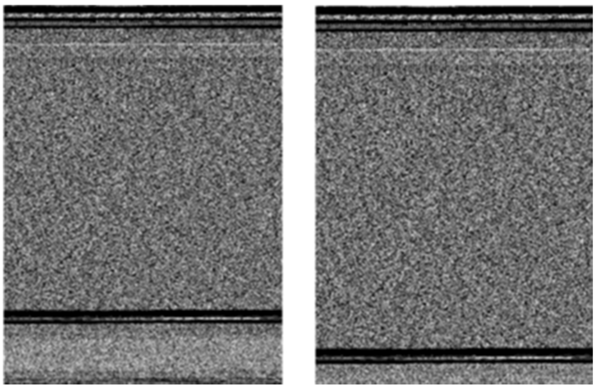


Fig. 9. Sample of Malimg dataset (Left) and added sample (Right)

V. CONCLUSIONS

This paper presented an innovative approach to malware detection by data transformation and utilization of well- established deep learning framework. The accuracy of malware detection shows improvement with the ResNeXt model, which includes the features of CNN's basic models, ResNet and InceptionNet. Furthermore, we can recognize that the CNN models show good performance in malware detection, proving its usefulness regardless of the dataset type. The advanced variants of CNN have demonstrated great success in image classifications, and its great attributes are poised to return great success in malware detection for cybersecurity.

REFERENCES

[1] "AV-test 2018. Malware Statistic" [Online]. Available: https://www.av-test.org/

[2] J. Chen, C. Su, K. H. Yeh, and M. Yung, "Special Issue on Advanced Persistent Threat," 2018.

[3] M. Powell, "11 Eye Opening Cyber Security Statistics for 2019," *CPO Magazine*, 2019.

[4] M. Kadivar, "CyberAttack Attributes," Technology Innovation Management Review, vol. 4, no. 11, 2014.

[5] K. Kendall and C. McMillan, "Practical Malware Analysis," in Black Hat Conference, USA, P-10 2007.

[6] M. Godlewski, G. House, T. Winnie, R. Mutter, B. L. Feore, T. Shipman, A. Scherba, L. McDole, A. L. Kremer, J. Mar-Spinola et al., "Malware Warning," 2018.

[7] 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, p-4, 2011.

[8] J. Brownlee, Deep Learning for Computer Vision: Image Classification, Object Detection, and Face Recognition in Python. Machine Learning Mastery, 2019.

[9] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, "Aggregated Residual Transformations for Deep Neural Networks," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1492–1500, 2017.

[10] M. D. McDougal, B. T. Ford, and W. E. Sterns, "Providing a Network Accessible Malware Analysis," 2015.

[11] H. Kettani and P. Wainwright, "On the Top Threats to Cyber Systems," in 2019 IEEE 2nd International Conference on Information and Computer Technologies (ICICT), pp. 175–179, 2019.

[12] J. Payne, M. Fenner, and R. Mills, "Detection of Malicious Code Insertion in Trusted Environments," 2016.

[13] J. Helfman, "Dotplot Patterns: a Literal Look at Pattern Languages," TAPOS, vol. 2, no. 1, pp. 31–41, 1996.

[14] G. Conti, S. Bratus, A. Shubina, B. Sangster, R. Ragsdale, M. Supan, A. Lichtenberg, and R. Perez-Alemany, "Automated Mapping of Large Binary Objects Using Primitive Fragment Type Classification," Digital Investigation, vol. 7, pp. 3–12, 2010.

[15] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778, 2016.

[16] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, "Inception-v4, Inception Resnet and the Impact of Residual Connections on Learning," in Thirty-First AAAI Conference on Artificial Intelligence, 2017.

[17] S. Targ, D. Almeida, and K. Lyman, "Resnet in Resnet: Generalizing Residual Architectures," arXiv preprint arXiv:1603.08029, 2016.

[18] S. Hitawala, “Evaluating ResNeXt Model Architecture for Image Classification,” arXiv preprint arXiv:1805.08700, 2018.

[19] L. Xiong, J. Karlekar, J. Zhao, Y. Cheng, Y. Xu, J. Feng, S. Pranata, and S. Shen, “A Good Practice Towards Top Performance of Face Recognition: Transferred Deep Feature Fusion,” arXiv preprint arXiv:1704.00438, 2017.

[20] “VirusTotal 2019. Online Malware Report Generator.” [Online]. Available: https://www.virustotal.com/

[21] “Malshare 2019. Malware Repository.” [Online]. Available: http://malshare.com/

[22] “VirusShare 2019. Malware Repository.” [Online]. Available: https://virusshare.com/

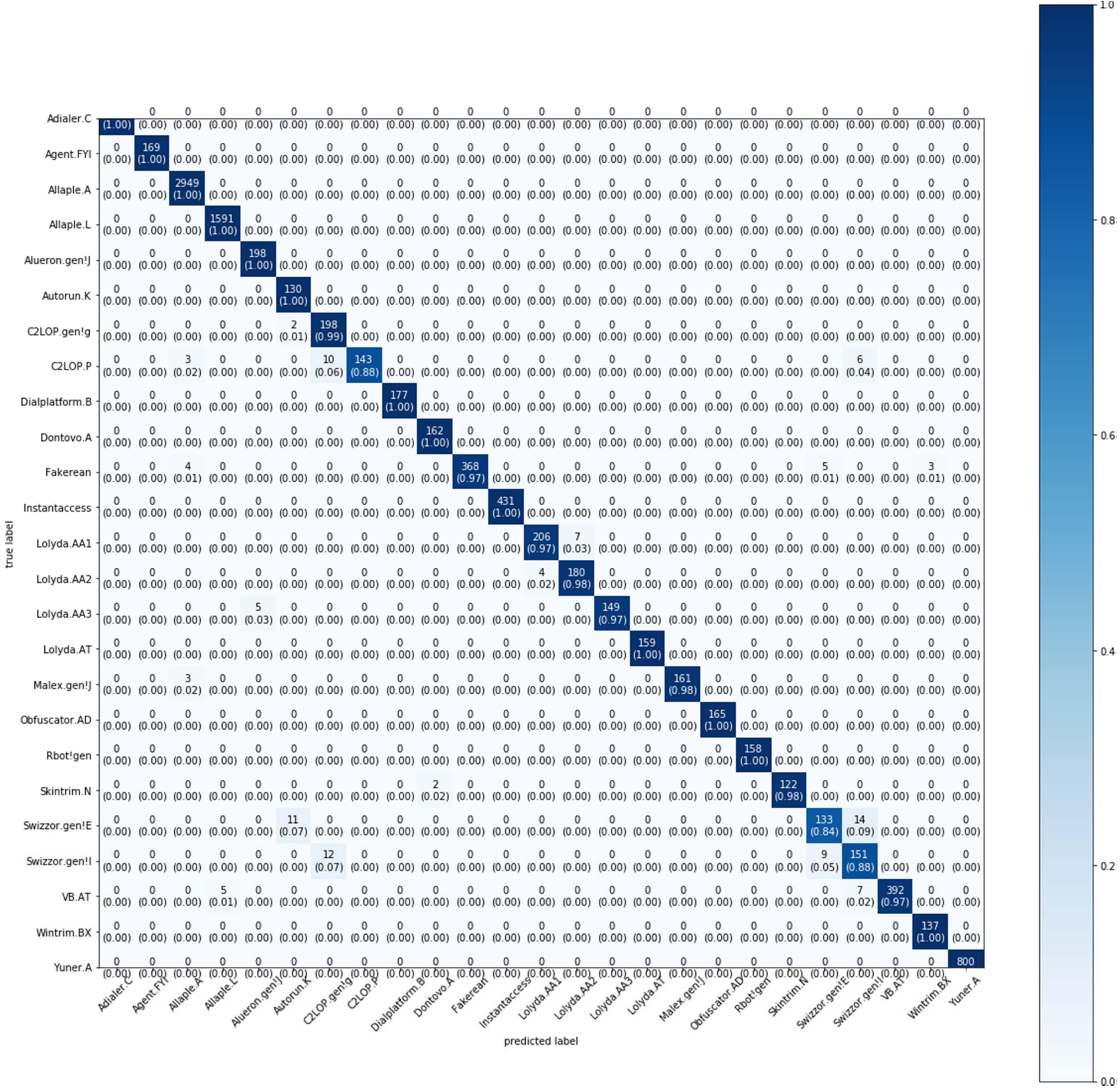


Fig. 10. ResNeXt 50 confusion matrix