Malware Images Visualization and Classification with Parameter Tunned Deep Learning Model

Muhammad Arham Tariq[1], Muhammad Ismaeel Khan[2], Aftab Arif[2], Muhammad Aksam Iftikhar[3], Ali Raza A Khan[4]

[1]Punjab Information Technology Board, Pakistan
[2]Washington university of science and technology, United States
[3]COMSATS University Islamabad, Department of Computer Science, Lahore, Pakistan
[4]Virginia university of science and technology, United States
Email: Arhamtariq99@gmail.com

Abstract: Malwares can be termed as a malicious program that can gain unauthorized access to the computer. This unauthorized access can damage and harm computing world in many capacities. There are many malware detection approaches present in the world. These approaches include static and dynamic analysis, machine learning, semi -supervised and deep learning-based models. These approaches cannot be visualized, thus cyber security experts face difficulty in interpreting underlying patterns. Conversion of malware byte code into images exits. An improved approach that can not only visualize malware, but also predict malware with high accuracy can be beneficial. For this purpose, we have used existing malware visualization technique. A technique which converts malware samples into images and then applies a contrast-limited adaptive histogram equalization algorithm to enhance the similarity between malware image regions in the same family. After conversion into images, we have applied parametrized tunned Convolutional Model to predict malware images. Comparing with existing our approach not only visualizes malware images but also outperforms previous approach by almost 2%, by achieving 98.27% accuracy.

Keywords: Malware Byte Code Conversion, Malware Visualization, Malware detection with Deep Learning.

1. Introduction

Malwares are malicious set of programs that attacks the operating system of the computer. This an-authorized access can by malwares can prove to be harmful. Malware detection has become a focus for many researchers now a days [1]. There are numerous approaches configured for malware detection and its analysis. Mostly used approaches among them consist of Static and Dynamic analysis of malware datasets [2]. After progress in Artificial Intelligence domain, many approaches for malware detection are configured using traditional machine, Deep learning approaches [3].

There were challenges and limitations in the previous approaches that needs to be overcome. Previous approaches were difficult to study and hard to reverse engineered. Lastly, parameters and number of layers in deep learning models required proper tunning during development phase. A new approach that not only converts malwares byte code into images for visualization, and also utilize deep learning models effectively for high accurate prediction can be fruitful. Keeping above factors in mind, our methodology res-uses existing proposed approach (Vismal) [4] to convert malware data points into images. Firstly. malware Converter takes charge of converting a malware

sample into a 2-Dimensional grayscale image, Feature Engineer enhances the recognition of malware via strengthening the local contrast in the malware image regions and resizes the image to a smaller one. Lastly Convolutional Neural Network is developed after testing on different number of layers and different learning rates and other parameter set. After configuring optimized Deep learning model, our approach outperforms previous ones by the margin of almost 2%.

The remainder of this paper is organized as follows. Section 2 describes the background of malware detection and analysis techniques and portrays how previous techniques differ from our proposal. Section 3 Describes the information regarding to the dataset used in our approach. In Section 4 we have elaborated the detailed description of the our proposed method.In Section 5 we have analyzed our expected contribution. In Section 6 we have presented detailed results, it's visualization and final model that is prepared. In Section 6 we have concluded our remarks.

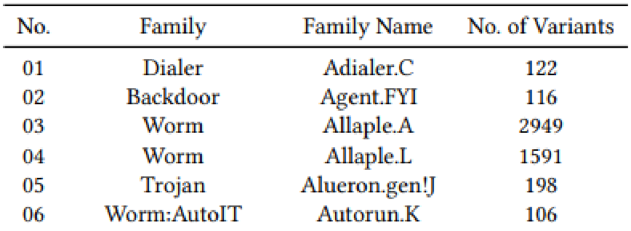
2. Literature Review

Jiang et al. configured a static analysis-based model by using data points carrying codes' semantic information extracted from sensitive opcode sequences, after KNN was applied onto it [5]. Blanc et al [6] configured a model based on ensemble based random forest classifier. Into which external attributes of malicious software's were utilized to classify different malwares of android. In order to develop these malwares, a software was decomposed with ApkTool and then transformed into human readable code named as Smali. These piece of Code, can be then studied by a parser based to check code's quality. After those 10 different attributes was extracted. Jung et al. [7] created a model based on deep learning approach. Into which byte code information of malware samples was used to train convolutional neural network model. The information retrieval procedure was divided into three major steps. Firstly, malware binary code was provided to dissembler, that provides malware code in disassembled form. After, that byte length sequential information was extracted from to the modified malware files. These parameters were used as particular hash function. Lastly, all of these hashed keys were used to develop a hash map. Bensaoud et al. [8] gathered dataset from 100,000 from APK malicious files. In the proposed approach bitmap from malware images was created, then provided as input to deep neural network. Four different Activation functions ReLU, LeakyReLU, PReLU, and ELU was utilized. The model achieved accuracy of 99.87%. Kumar S [9] proposed a model based on two most popular benchmark datasets (Malimg and Microsoft). The model was based on Transfer learning-based model Image Net. After combination of model with Early Stopping the model attained 93.19% and 98.92% accuracy. Bhodia N et al. [10] configured a model based on Malimg Malware benchmark dataset. The model was based on a comparaitative analysis fo KNN and DL based algorithms. The model attained 94.80 % accuracy on multi class classification on Malimg dataset. Fangtian Zhong et al [4] presented a model named as Vismal for malware visualization and detection. Vismal focuses on three main goals, firstly malware visualization, malware classification with high accuracy and lastly reducing classification time. Vismal firstly coverts malware data into images, then applied adaptive histogram to compact malware images. Lastly applied Deep Neural Convolutional Network to predict with high accuracy and low classification time.Vismal attains 96 % accuracy and 4ms classification time on the Malware Malimg dataset. Awan et al. [12] presented a deep learning based spatial attention and convolutional neural network (SACNN). The model was evaluated on the Malimg dataset. The model attained 97% accuracy after application of data balancing algorithm with proposed deep learning algorithm. Agarap AF[13] configured an approach based on comparison between Three Deep Learning methods. The DL methods CNN-SVM, GRU-SVM, and MLP-SVM applied onto Malimg Dataset.Best accuracy attained by DL-SVM 84.92%

3. DATASET

The Dataset used in the approach is famous benchmark dataset for Malware analysis and detection named as Malimg Dataset. It consists of 25 different malware families and 9339 families.

TABLE I Malware Malimg Dataset Description [11]



No.	Family	Family Name	No. of Variants
01	Dialer	Adialer.C	122
02	Backdoor	Agent.FYI	116
03	Worm	Allaple.A	2949
04	Worm	Allaple.L	1591
05	Trojan	Alueron.gen!J	198
06	Worm:AutoIT	Autorun.K	106

07	Trojan	C2Lop.P	146
08	Trojan	C2Lop.gen!G	200
09	Dialer	Dialplatform.B	177
10	Trojan Downloader	Dontovo.A	162
11	Rogue	Fakerean	381
12	Dialer	Instantaccess	431
13	PWS	Lolyda.AA 1	213
14	PWS	Lolyda.AA 2	184
15	PWS	Lolyda.AA 3	123
16	PWS	Lolyda.AT	159
17	Trojan	Malex.gen!J	136
18	Trojan Downloader	Obfuscator.AD	142
19	Backdoor	Rbot!gen	158
20	Trojan	Skintrim.N	80
21	Trojan Downloader	Swizzor.gen!E	128
22	Trojan Downloader	Swizzor.gen!I	132
23	Worm	VB.AT	408
24	Trojan Downloader	Wintrim.BX	97
25	Worm	Yuner.A	800

4. Methodology

This section will define the methodology used in our approach. In the first step of methodology, we have converted malware byte code into gray scale images. After conversion into images, then contrast limited adaptive histogram equalization is applied to improve visualization and decrease classification time. Lastly Convolutional Neural Network is applied with different parameters set and Layers. Different Evaluation Metrics are extracted by our CNN model using 10K-Fold cross validation.

A. Malware Image Conversion

In the first step, malware byte codes are converted into malware images. From to the byte code, a decimal value is extracted within scale of [0-255]. Afterwards for each byte information in the data point, a gray scale value is extracted and then all of these gray scale values are converted into malware images.

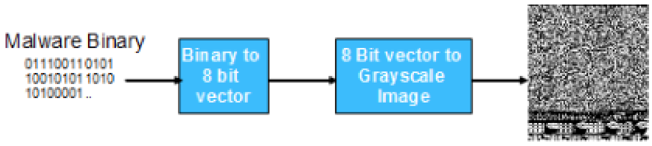


Fig. 1. Malware Byte Code to Image Conversion

B. Feature Enhancement

In the Second step, we have re-used feature enhancement proposed in the previous model named as Vsmal [4]. In order to enhance adaptive histogram equalization algorithm is applied. Similar pixel and contrast values between different malware families is promoted. The equalization algorithm is divided into four different steps. These steps can be termed as Division, Cumulation, Clipping, and Transformation. In the Division step, it divides an image into a × b smaller regions where a and b can be termed as the numbers of pieces split up for the height and width of the malware image. Cumulation, can be defined as a frequency distribution management function. This function can be defined as:

$$cdf(i) = X\ i\ j=0\ nj\ ,\ 0 \leq i < L \qquad (1)$$

In the equation No 1, L can be defined as the total count of gray scales. nj can be represented as the total count of times a particular value of pixel appears into an image. In order to equally distribute the pixels around the image as concept Clipping is applied. In Clipping a random value between [0-255] is assign to a particular value above then clip limit. The concept of clip limit can be visualized in the Fig,2 below.

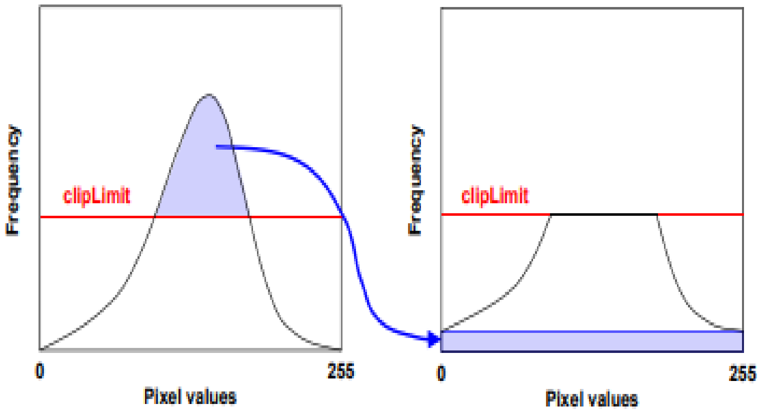


Fig.2 Clipping Limit [4]

Lastly Transformation to malware images is applied. It makes the pixel intensities inter-related, enhances the contrast of the image. Next, the transformed image is resized with a shape (s, s) and then input to Deep Neural Network.

C. Convolutional Neural Network

Convolutional Neural Network is a deep learning based neural network, mostly used for image classification. Main modification in the model is the addition of a operation named as Convolution. It is a operation that utilizes two different operations to produce a third function. The Third function explains how the image is modified by the first two ones. In our model we have developed a 11 layers-based model, evaluated on the basis of the parameter set mention in the Table No II.

TABLE II. Parameters Set for CNN

Parameter Name	Values Set
Epochs	{5,10,15}
Optimizer	'SGD', 'Adam', 'RMSprop'
Learning rate	{0.001,0.1,0.01}

5. Contribution

In this section, we will elaborate our expected contribution in the proposed methodology. We have firstly created a CNN based model by evaluating onto different parameters set. After combination of this model with existing visualization techniques, we claim following things:

- Firstly our model can help cyber security experts in the visualization of malwares effectively
- Our model can predict malwares with less classification time
- Our Model can predict malwares with better accuracy ,than previous models

6. Results

In this section we elaborate the final results with the best parameters that our model has achieved. We have extracted five different evaluation metrics for our proposed approach. The five-evaluation metrics are accuracy, precision, recall, F1-score and classification time. Table No III and IV elaborates the best parameters with most appropriate results and after its complete description of twelve 11 layers of CNN model is presented.

TABLE III. Best Parameters Set for CNN

Parameter Name	Values Set
Epochs	{10}
Optimizer	'Adam'
Learning rate	{0.01}

TABLE IV. CNN Model Description

Layer	Description
Layer No 1	Conv2D, kernel_size=(3,3), activation='relu', input_shape=(64,64,3)
Layer No 2	MaxPooling2D(pool_size=(2, 2))
Layer No 3	Conv2D(15,(3,3), activation='relu')
Layer No 4	MaxPooling2D(pool_size=(2, 2))
Layer No 5	Conv2D(15, (3,3), padding='same', kernel_regularizer=regularizers.l2(0.01))
Layer No 6	Dropout(0.25)
Layer No 7	Flatten()
Layer No 8	Dense(128, activation='relu')
Layer No 9	Dropout(0.5)
Layer No 10	Dense(50, activation='relu')
Layer No 11	Dense(num_classes, activation='softmax')

Table No V below describes the final results that we have achieved from our final model. All the results are achieved by using 10 K Fold cross validation

TABLE V. CNN Results

Evaluation Metrics	Score
Accuracy	98.85%
Precision	97.76%
Recall	98.87%
Classification time	4ms

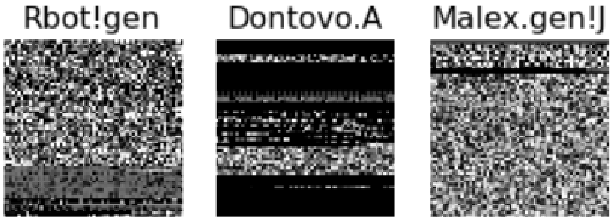
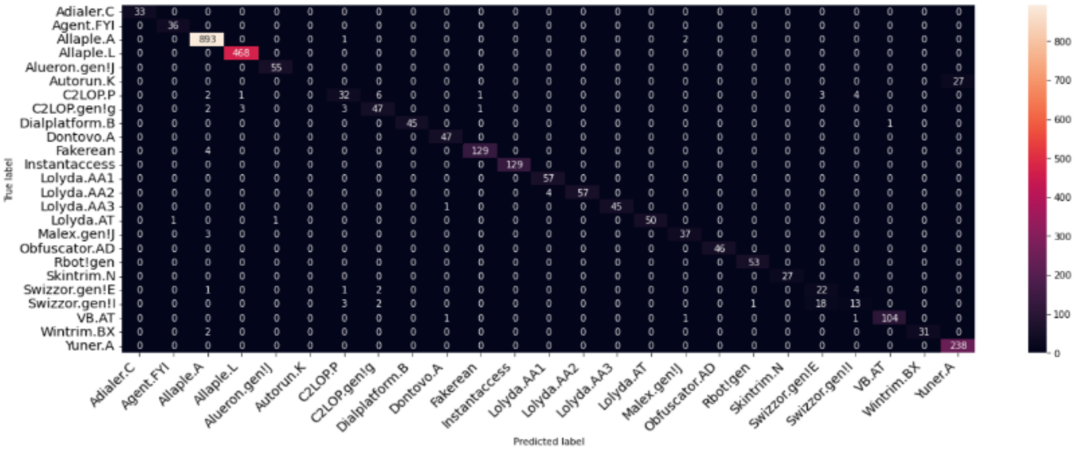


Fig 3 Malware Visualization

Figure No 3 represents some classes visualization examples converted with help of approach re-used in our approach. Fig No 4 represents the confusion matrix generated from the final results model. It includes confusion matrix generation from all the 25 classes used in our approach.



	Adialer.C	Agent.FYI	Allaple.A	Allaple.L	Alueron.gen!J	Autorun.K	C2LOP.P	C2LOP.gen!g	Dialplatform.B	Dontovo.A	Fakerean	Instantaccess	Lolyda.AA1	Lolyda.AA2	Lolyda.AA3	Lolyda.AT	Malex.gen!J	Obfuscator.AD	Rbot!gen	Skintrim.N	Swizzor.gen!E	Swizzor.gen!I	VB.AT	Wintrim.BX	Yuner.A
Adialer.C	33	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Agent.FYI	0	36	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Allaple.A	0	0	893	0	0	0	1	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0
Allaple.L	0	0	0	468	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Alueron.gen!J	0	0	0	0	55	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Autorun.K	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	27
C2LOP.P	0	0	2	1	0	0	32	6	0	0	1	0	0	0	0	0	0	0	0	0	3	4	0	0	0
C2LOP.gen!g	0	0	2	3	0	0	3	47	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Dialplatform.B	0	0	0	0	0	0	0	0	45	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
Dontovo.A	0	0	0	0	0	0	0	0	0	47	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Fakerean	0	0	4	0	0	0	0	0	0	0	129	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Instantaccess	0	0	0	0	0	0	0	0	0	0	0	129	0	0	0	0	0	0	0	0	0	0	0	0	0
Lolyda.AA1	0	0	0	0	0	0	0	0	0	0	0	0	57	0	0	0	0	0	0	0	0	0	0	0	0
Lolyda.AA2	0	0	0	0	0	0	0	0	0	0	0	0	4	57	0	0	0	0	0	0	0	0	0	0	0
Lolyda.AA3	0	0	0	0	0	0	0	0	0	1	0	0	0	0	45	0	0	0	0	0	0	0	0	0	0
Lolyda.AT	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	50	0	0	0	0	0	0	0	0	0
Malex.gen!J	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	37	0	0	0	0	0	0	0	0
Obfuscator.AD	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	46	0	0	0	0	0	0	0
Rbot!gen	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	53	0	0	0	0	0	0
Skintrim.N	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	27	0	0	0	0	0
Swizzor.gen!E	0	0	1	0	0	0	1	2	0	0	0	0	0	0	0	0	0	0	0	0	22	4	0	0	0
Swizzor.gen!I	0	0	0	0	0	0	3	2	0	0	0	0	0	0	0	0	0	0	1	0	18	13	0	0	0
VB.AT	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	1	104	0	0
Wintrim.BX	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	31	0
Yuner.A	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	238

Fig.4 Final Confusion Matrix

7. Comapratative Analysis

In this section, we will compare our configured methodology with previous methodologies in terms of accuracy and classification. Table No V describes the comparison of different approaches.

Table No V Comparison

Methodology	Accuracy	Classification Time
Agarap AF[12]	84.92%	N. A
Fangtian Zhong et al [4]	96%	4ms
Kumar S [9]	93.19%	N. A
Proposed	98.85%	4ms

8. Conclusion

In this approach we have configured by malware classification and visualization model. Previously different approaches for malware classification were presented, but they were difficult to analyze and reverse engineer. Our model by re-using previously presented visualization approach has not only help cyber security experts to visualize malwares. But apart from that with the help of parameter tuning our model has attained more accuracy 98.64 % accuracy then previous ones. In the future we try to incorporate GAN to generate more versions of malwares. Lastly configuring this approach on other benchmark datasets version of malware can be more beneficial.

References

1. Aboaoja FA, Zainal A, Ghaleb FA, Al-rimy BA, Eisa TA, Elnour AA. Malware Detection Issues, Challenges, and Future Directions: A Survey. Applied Sciences. 2022 Aug 25;12(17):8482.
2. da Costa FH, Medeiros I, Menezes T, da Silva JV, da Silva IL, Bonifácio R, Narasimhan K, Ribeiro M. Exploring the use of static and dynamic analysis to improve the performance of the mining sandbox approach for android malware identification. Journal of Systems and Software. 2022 Jan 1;183:111092.
3. Suresh P, Logeswaran K, Keerthika P, Devi RM, Sentamilselvan K, Kamalam GK, Muthukrishnan H. Contemporary survey on effectiveness of machine and deep learning techniques for cyber
4. security. InMachine Learning for Biometrics 2022 Jan 1 (pp. 177-200). Academic Press.
5. Zhong F, Chen Z, Xu M, Zhang G, Yu D, Cheng X. Malware-on-the-Brain: Illuminating Malware Byte Codes with Images for Malware Classification. IEEE Transactions on Computers. 2022 Mar 17.
6. J. Jiang, S. Li, M. Yu, G. Li, C. Liu, K. Chen, H. Liu, and W. Huang, “Android malware family classification based on sensitive opcode sequence,” in 2019 IEEE Symposium on Computers and Communications (ISCC), 2019, pp. 1–7.
7. W. Blanc, L. G. Hashem, K. O. Elish, and M. J. Hussain Almohri, “Identifying android malware families using android-oriented metrics,” in 2019 IEEE International Conference on Big Data (Big Data), 2019, pp. 4708–4713.
8. B. Jung, T. Kim, and E. G. Im, “Malware classification using byte sequence information,” in Proceedings of the 2018 Conference on Research in Adaptive and Convergent Systems (RACS’18), 2018, pp. 143–148
9. Bensaoud A, Kalita J. Deep multi-task learning for malware image classification. Journal of Information Security and Applications. 2022 Feb 1;64:103057.
10. Kumar S, Janet B. DTMIC: Deep transfer learning for malware image classification. Journal of Information Security and Applications. 2022 Feb 1;64:103063.
11. Bhodia N, Prajapati P, Di Troia F, Stamp M. Transfer learning for image-based malware classification. arXiv preprint arXiv:1903.11551. 2019 Jan 21.
12. Awan MJ, Masood OA, Mohammed MA, Yasin A, Zain AM, Damaševičius R, Abdulkareem KH. Image-Based Malware Classification Using VGG19 Network and Spatial Convolutional Attention. Electronics. 2021 Oct 8;10(19):2444.
13. Agarap AF. Towards building an intelligent anti-malware system: a deep learning approach using support vector machine (SVM) for malware classification. arXiv preprint arXiv:1801.00318. 2017 Dec 31.