Malware Detection Using Deep Learning

[1]Sneh Gaikwad, [2]Shreya Shah, [3]Kalpesh Tated and [4]Prof. Santosh Tamboli,
[1,2,3,4]Department of Information Technology, Vidyalankar Institute of Technology, Mumbai, India

Abstract: In this study, we have used the Image Similarity technique to detect the unknown or new type of malware using CNN approach. CNN was investigated and tested with three types of datasets i.e. one from Vision Research Lab, which contains 9458 gray-scale images that have been extracted from the same number of malware samples that come from 25 different malware families, and second was benign dataset which contained 3000 different kinds of benign software. Benign dataset and dataset vision research lab were initially executable files which were converted in to binary code and then converted in to image files.

I. INTRODUCTION

A. Background

One of the major challenges in the realm of security threats is malicious software which is also referred as malware. The main focus of malware is, to gather the personal information without the attention of users and to disturb the computer operations which makes problems for users. There are many kinds of malware i.e. Virus, Worm, Trojan-horse, Rootkit, Backdoor, Spyware, Adware etc. Annual reports from antivirus companies show that thousands of new malwares are created every single day. This new malware become more sophisticated that they could no longer be detected by the traditional detection techniques such as signature-based detection, heuristic detection or behaviour-based detection.

Signature-based detection searches for specified bytes sequences into an object so that it can identify exceptionally a particular type of a malware. Its drawback is that it cannot detect zero-day or new malware since these malware signatures are not supposed to be listed into the signature database. Heuristic-based detection was developed to basically overcome the limitation of the signature detection technique, in the way that it scans the system's behaviour in order to identify the activities which seems to be not normal, instead of searching for the malware signature. Heuristic-based detection method can be applied to newly created malware whose signature has not yet been known. The limitation of this technique is that it affects the system's performance and requires more space. Behaviour-based detection technique is more about the behaviour of the program when it is executing. If a program executes normally, then it is marked as benign, otherwise it is marked as a malware. By analysing this definition of the behaviour-based detection, we can directly conclude that the drawback of this technique is the production of many false positives and false negatives, considering the fact that a benign program can crashed and be marked as a virus or virus can execute as if it was a normal program and simply be marked as benign.

B. Motivations

Malware is growing in the huge volume every day, we used image processing technique in order to improve accuracy and performance. Image processing technique analyses malware binaries as gray-scale images. Some of mature image processing techniques are widely used for object recognition e.g. taobao is popular shopping website in china which find's the product using image recognition technique. This method performs high accuracy in practice. In this study, we converted binary code to images for recognizing malware which preserve the similarities variant images. We observed that the image recognition method is helpful to achieve better performance and accuracy.

C. Our approach

Malware classified in different families has multiple characteristics or features. Many authors used machine learning models such as Regression, K-nearest-neighbour, Random Forest etc. Main disadvantage of using machine learning is, features extraction is manual. Gavrilut gave an overview of different machine learning techniques that were previously proposed for malware detection. Unlike Machine Learning, Deep learning skips the manual steps of extracting features. For instance, we can feed directly images and videos to the deep learning algorithm, which can predict the object. In this way deep learning model is more intelligent rather than machine learning model. We used convolutional neural networks because it is reliable and it can be applied to the entire image at a time and then we can assume they are best to use for feature extraction. Recently Constitutional Neural Networks is the new approach to detect malware by using image-based similarity technique. Its automated image comparison helps analysts to visually identify common code portions or specific instruction blocks within a sample. We will use different techniques to prepare datasets for training and testing purposes. We will train and test the CNN model for better understanding of the malware behaviour. Overall, we show that our proposed approach constitutes a valuable asset in the fight against malware. Figure 1 gives a brief overview to the different stages i.e. from data preparation to malware detection.
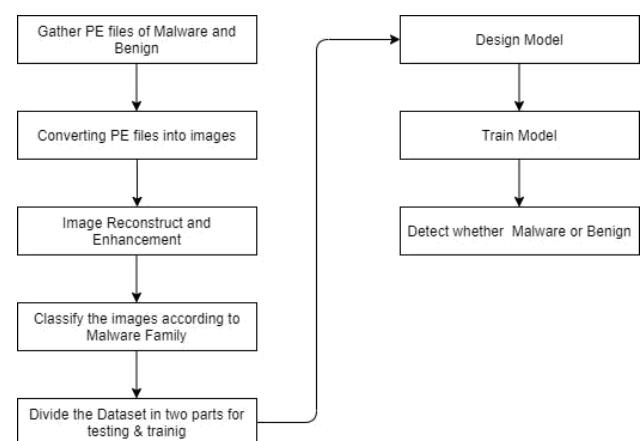

Figure 1. Workflow diagram; from data preparation to malware detection.

D. Contributions

The main contributions of the paper are summarized as follows:

We used the Convolutional Neural Networks for detection of malware, based on image similarity which is further described in Section 3

It can successfully analyse and detect unknown or new type of malware. It can give better results in terms of training / testing accuracy and speed of detection which is further described in section 3.

E. Structure of paper

The Section 1 discusses the background, motivation, approach used in this study and main contributions of this work. Section 2 gives a brief overview to the methodology that how to convert executable files in to images and also setup the python libraries. Section 3 proposes a malware detection technique, discusses optimized CNN model, describes the implementation and experiment results in terms of accuracy. Finally, Section 4 concludes the paper.

II. DATA PREPARATION AND ENVIRON-MENT SETUP

This section is divided into two parts. The first part is, to collect malware and benign datasets from different sources and second part describes the techniques of preparation of the dataset. In second part we used a technique to prepare dataset which is described in Section 2.2.

A. Collection of Dataset

We have collected two datasets from different sources. One of them is a malicious dataset from source i.e. from Vision Research Lab. We also collected 3000 benign file from different sources. All two datasets are discussed briefly in the following discussions. 2.1.1 Vision Research Lab Dataset

First dataset is collected from Vision Research Lab and this dataset is called Malimg Dataset. The dataset comprises 25 malware families while the number of variants is different in each family. Dataset is shown in Table 1 along with class name, family name and number of samples.

Table 1. Malimg Dataset from Vision Research Lab Dataset

| No	Class	Family Name	No of Samples
1	Worm	Allaple.L	1591
2	Worm	Allaple.A	2949
3	Worm	Yuner.A	800
4	PWS	Lolyda.AA 1	231
5	PWS	Lolyda.A A2	184
6	PWS	Lolyda.A A3	123
7	Trojan	C2Lop.P	146
8	Trojan	C2Lop.gen!G	200
9	Dialer	Instantaccess	431
10	Trojan Downloader	Swizzor.ge n!l	132
11	Trojan Downloader	Swizzor.gen!E	128
12	Worm	VB.AT	408
13	Rogue	Fakerean	381
14	Trojan	Aluron.gen!J	198
15	Trojan	Malex.gen!J	136
16	PWS	Lolyda.AT	159
17	Dialer	Adialer.C	125
18	Trojan Downloader	Wintrim.BX	97
19	Dialer	Dialplatform.B	177
20	Trojan Downloader	Dontovo.A	162
21	Trojan Downloader	Obfuscator.AD	142
22	Backdoor	Agent.FYI	116
23	Worm:AutoIT	Autorun.K	106
24	Backdoor	Rbot!gen	158
25	Trojan	Trojan	Trojan
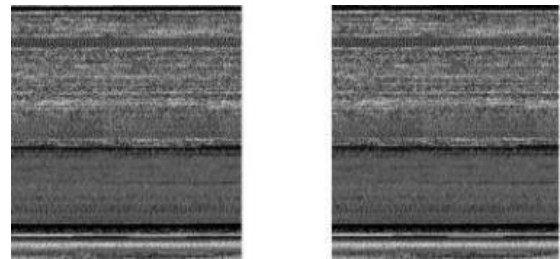
Figure 2. Gray-Scale Images constructed from malware.

Malimg Dataset consists 9,458 gray-scale images of 25 malware families. Ratio of 90-10 was used for model performance evaluation. 90% of the total data will be used for training and 10% will be used for testing.

As Gavrilut et al. explained that a binary code of a given malware can be read as a vector of 8 bits unsigned integers and organized into 2-dimensional array which can be visualized as a gray-scale image in the range of [0,255], where 0 represent black and 255 for white. The size of the image is different depending on their families. We observed in Figure 2, that images which belong to the same family are looking very similar to one another. We collect 3000 benign files from different sources.

B. Data Preparation Techniques

This paper proposes the following technique to process the data.

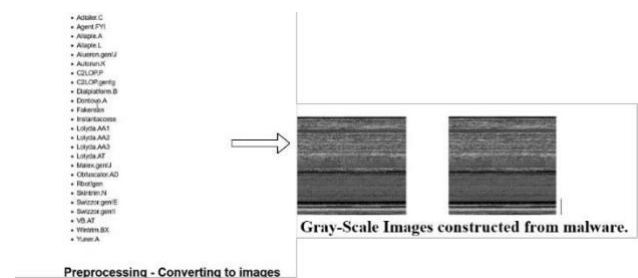### 1. Converting PE (Portable Executable) files to Images:

Figure 3. Overview architecture of Preparation Dataset.

Conversion: The binaries of Portable Executable files are converted into matrix using algorithm forming a grey scale image.

Convert Assembly Code to Image: The process of converting assembly to images is shown in Figure 3.

C. Environment Setup

Operating system with 64bit with 8 GB RAM environment is used to perform tests. We used Python programming language to perform the experiments. Python packages and libraries such

as Tensor Flow, Numpy, will be used which help to detect the malware. The Tensor Flow Library is used for training the model which uses the convolutional natural network (CNN).

III. IMPLEMENTATION AND PERFORMANCE EVALUATION OF THE PROPOSED MODEL

A. Proposed Model

In this design we divided model in two phases i) Training phase and ii) Detection phase. For the training and the detection of malware we will use CNN model, as shown in Figure 5. We prepare the dataset using different techniques shown in data preparation section. The output of the data preparation section is "image files". Images will have binary labels i.e. either benign or malware. We will use supervised learning model in which the features are extracted automatically. The detection phase is shown in Figure 4. The same exe file convert in image and trained classifier detects the malicious code.

B. Training Convolutional Neural Networks Structure

We will use convolutional neural networks because it is reliable and it can be applied to the entire image at a time and then we can assume they are best to use for feature extraction. Convolutional neural network is a feed-forward neural network where the connectivity pattern between neurons is inspired by the structure of an animal visual cortex and that has proven great value in the analysis of visual imagery.
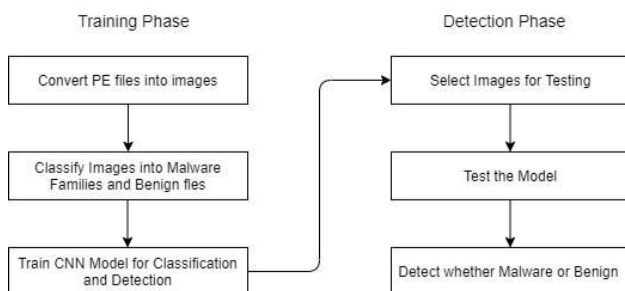

Figure 4. Architecture of proposed Method.

All the images are reshaped into a size of 128 X 128 pixels. Since all the models of deep learning accept data in form of numbers, we will use image library from PIL package of Python to generate vectors of images and further processing are done on these vectors.

We have then designed a three layer deep Constitutional Neural Network for the detection task, which has the following properties: On the Rectified Linear Units (ReLU) layers, we will first apply a two dimensional convolutional layer and after each layer, we apply a nonlinear layer also known as activation layer. In convolutional layer, we have operations like element-wise multiplication and summations. The ReLU adds non-linearity to the system. We will use the ReLU instead of non-linearity function because it is faster than tanh or sigmoid and help in vanishing gradient problem which arises in lower layers of the network.

We will also use max pooling layer instead of other layers. It takes a filter and a stride of the same length then applies it to the input volume and outputs the maximum number in sub region that the filter involves around. The intuition behind this was the fact that our malware image is a gray scale and the layers like average max pooling may not help much because there are a lot of dark space in the image and they don't contribute much in the model.

The output that we want is a single class in which the given malware belongs to. After applying all the layers, we have a three-dimensional vector of arrays. To convert this vector into a class probability, we convert these vectors into a single layer of one dimension, known as fully connected layer. Down-sampling all the vectors to a one-dimensional vector may lead to loss of data. For that reason, we have used two fully connected layers.

Cross entropy loss function that is commonly used for multi class classification will be used for this work as well as Adam optimizer for optimization task. The overall architecture of the model is show in Figure 5.

Initially, all the images were of different sizes and had to be converted into 128 X 128 pixels before they are used as input to the model.
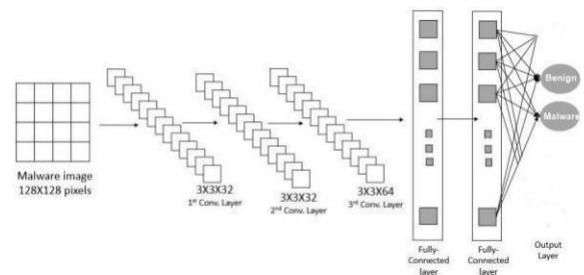

Figure 5. Overview architecture of CNN proposed Method.

C. Implementation

The following tools and techniques are required for experimental setup. For preparation of dataset we used method shown in section 2, tools and algorithm which were used to arrange the dataset for achieving better results is also written in data preparation section. For detecting malware, we will use supervised learning to train the model. CNN algorithm will be used to train and test the model. Figure 5 shows 3 hidden layers, each layer has own parameters (e.g., filter − size1 = 3, numfilters = 32, etc), In this algorithm we use Adam Optimizer and the learning rate of the optimizer is le-4. The size of for all hidden layers for the convolutional neural network are 3*3*32, 3*3*32, 3*3*64, respectively. For the validation, system was trained with 20 epochs.

CONCLUSION

Being able to visualize the malicious code as a gray-scale image will be a great achievement. Many researchers have been using this technique for the task of malware classification and detection. The biggest challenge is to find an efficient way to overcome the vulnerability of Neural Networks. This could be achieved by carefully analysing malware binaries.

References

[1] Papernot, N., McDaniel, P., Goodfellow, I., Jha, S., Celik, Z.B. and Swami, A., 2017, April. Practical black-box attacks against machine learning. In Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security (pp. 506-509 ACM)

[2] Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G.S., Davis, A., Dean, J., Devin, M. and Ghemawat, S., 2016. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. arXiv preprint arXiv:1603.04467.

[3] Adebayo, O.S. and Aziz, N.A., 2015. Static Code Analysis of Permission-based Features for Android

Malware Classification Using Apriori Algorithm with Particle Swarm Optimization. Journal of Information Assurance & Security, 10(4).

[4] Alme, C., Mcafee, Inc., 2012. Systems, apparatus, and methods for detecting malware. U.S. Patent 8,312,546.

[5] Bennasar, H., Bendahmane, A. and Essaaidi, M., 2017, April. An Overview of the State-of-the-Art of Cloud Computing Cyber-Security. In International Conference on Codes, Cryptology, and Information Security (pp. 56-67). Springer, Cham.

[6] Cao, C., Liu, X., Yang, Y., Yu, Y., Wang, J., Wang, Z., Huang, Y., Wang, L., Huang, C., Xu, W. and Ramanan, D., 2015. Look and think twice: Capturing top-down visual attention with feedback convolutional neural networks. In Proceedings of the IEEE International Conference on Computer Vision (pp. 2956-2964).

[7] Gavriluţ, D., Cimpoeşu, M., Anton, D. and Ciortuz, L., 2009, October. Malware detection using machine learning. In Computer Science and Information Technology, 2009. IMCSIT'09. International Multiconference on (pp. 735-741). IEEE.

[8] Gavriluţ, D., Cimpoeşu, M., Anton, D. and Ciortuz, L., 2009, October. Malware detection using machine learning. In Computer Science and Information Technology, 2009. IMCSIT'09. International Multiconference on (pp. 735-741). IEEE.

[9] Goodfellow, I.J., Shlens, J. and Szegedy, C., 2018. Explaining and harnessing adversarial examples. arXiv preprint arXiv:1412.6572.

[10] Nataraj, L., Yegneswaran, V., Porras, P. and Zhang, J., 2015, October. A comparative assessment of malware classification using binary texture analysis and dynamic analysis. In Proceedings of the 4th ACM Workshop on Security and Artificial Intelligence (pp. 21-30). ACM.

[11] Nguyen, A., Yosinski, J. and Clune, J., 2018. Deep Neural Networks Are Easily Fooled: High Confidence Predictions for Unrecognizable Images-Nguyen_Deep_Neural_Networks_2015_CVPR.