Visualization of Malware Families for Deep Learning Technology

Kire Jakimoski
Faculty of Informatics
AUE-FON University
Skopje, Republic of N. Macedonia
kire.jakimoski@fon.edu.mk

Tarek Chaalan
School of Engineering, Information Technology and Physical Sciences
Federation University
Melbourne, Australia
tarek.chaalan@students.federation.edu.au

***Abstract*—Visualization of malware is very attractive and challenging topic in the academic society and roots of its progress are in the fast development of the deep learning technologies using the neural networks. In this paper focus is put on developing new methodology for visualization of recent malware families in order to improve the malware analysis process using the actual deep learning techniques. Presented dataset for malware classification and detection in this paper is obtained with PNG generation of PE files. This will help in building better advanced intrusion detection and prevention systems for cyber defence using the deep learning technology with the help of CNN (Convolutional Neural Networks). Presented methodology and experimental results in this paper are helping cyber security researchers to dive into malware detection in an amazingly simple approach and away from the Machine Learning complications or any expertise.**

Keywords—CNN, dataset, deep learning, malware, PE files

I. INTRODUCTION

Detection of malware is one of the most important parts of the cyber security science. Number of malware attacks all around the world is increased, especially in the last few years during and after the covid crisis. That's why it is very important intrusion and detection systems to detect the malware efficiently and accurately. Significant increase of the malware volume requests advanced techniques for faster identification and classification of the malware.

Traditional malware detection techniques and systems are difficult and not very user friendly for analysis of the malware. Detection of the malware using Convolutional Neural Networks (CNN) as deep learning technology is very smart and innovative solution. Visualization of the malware files is very important in this context and research in the past few years is already done in this area.

Authors in [1] propose a framework that includes 8 FT CNN models, where binary fields of various malware families are transformed in 2D images and forwarded to the FT CNN models. Results in this research show high percentage of accuracy in the detection of the malware types when using the proposed FT CNN models. Authors in [2] created a new image dataset to serve as a direct replacement for the original MNIST dataset for benchmarking machine learning algorithms. Authors in [3] give benefit to the development of semantic segmentation algorithms through generating a large, curated dataset representative of highly variable segmentation tasks. In [4] malware detection system is developed from the authors. This system transforms malware files into images and CNN is used to classify the image representation. Authors in [5] propose method for malware classification based on images with a help of CNN. Their results show that a set of CNN architectures enables extracting features that have better quality compared with traditional methods. Deep learning framework that is also useful for malware classification is presented by the authors in [6]. Authors in [7] propose algorithm that converts malware to gray images and identifies the corresponding malware families by using CNN. In [8] authors use method for generation images from malware with combining the static analysis of the malicious code together with CNN and RNN (Recurrent Neural Networks) methods. The efficiency of the malware visualization is researched in [9] to solve the features selection and extraction problems. Experiments are obtained from the authors in [9] on 12 different neural network architectures and proposed approach produced high accuracy. Authors in [10] are using hybrid in-house model for identification of the malware families after previous conversion of the malware binaries into gray scale images. They propose in this paper scalable framework that uses visualization and deep learning for identification of the malware families.

As we can see from the above presented papers there are researchers that are researching visualization of the malware families as well as deep learning technologies mostly related with CNN in order to improve and modernize the detection of the malware. In this research visualization of the malware families is done with presentation as PNG files that are divided into a training and dataset. Dataset is actually an extension of Maldataset 2021 where new malware families and classes are added. Furthermore, the conversion is not just grayscale, but also colourful using RGB.

Section 2 in this paper explains the methodology used for development of the malware dataset that includes 30 classes of malware that could be used to train a CNN model. Section 3 describes the structure relationship between the PE and PNG file structure. In Section 4 results from the experiments done for this research are presented. Finally, Section 5 concludes the paper.

II. METHODOLOGY

The dataset was developed by creating a visualization of Portable Executable (PE) raw bytes of malware families and saved as a PNG file. The dataset we provided in this paper is a representation for PNG files split into a training and testing dataset. This dataset can be used to train a CNN model. Hence CNN models don't require any feature engineering, and usually, CNN models have an extremely high accuracy. We chose to create this dataset to help security researchers to dive into malware detection in an amazingly simple approach and away from the Machine Learning complications or any expertise. This dataset is an extension of Maldataset 2021 provided in [11], since we added new malware families and new classes. In addition, we enhanced the conversion to not just grayscale, but we included RGB. Maldataset 2022 is a malware dataset that consists of 30 classes of malware, in which each class represents a malware family, and each sample gives an RGB 224x224 PNG file. The PNG files are transformed from the original binary malware files. The motivation of image transformation is to identify malware on the raw bytes of entire executable files (i.e., image), so that deep learning technologies such as CNN and SVM can be applied to malware classification since CNN model has been demonstrated with its outstanding capability on image classification. In this view, we provide here a new dataset that contains the latest malware samples. The entire PNG files are split as, 70 percentages for training and the remaining 30 percentages for testing.

The PNGs are generated by extracting the byte values from binary executable files and creating the GreyScaleImage and RGBImage and then saving the extracted binary as images using python Pillow library. We observed that the GreyScale Images can be very noisy and close to each other's even if they belong to two different malware families and classes. While the RGBImage solves this issue and enhances the image and shows the differences in these images for the almost identical ones.

A. Dataset Evaluation and Model Testing

During the data and labels discovery we found that one third of the data's dimension is not 224*224(50,176). To overcome this limitation, we performed padding techniques during data processing to ensure all the data fed to the model have the same size 224*224. Then we used K-fold technique (K=5) while training the model [12]. We used this technique to evaluate the performance of the machine learning model by partitioning the data into k equally sized folds.

K-fold provided us with a more accurate estimate of the model's performance than the single train/test split, and it helped us to identify overfitting. Since we are using a complex model structure, we expected overfitting. It also makes an efficient usage for the available data as each data point is used for both training and validation across the k folds especially our dataset is considered a small dataset with limited number of samples in each class.

Cluster by K-means:

The samples clustered by K-means are distributed in this following criteria:

{the cluster number}_index_{index number}_lable_{lable in dataset}.png

Model:

Since the samples of data are not enough to obtain a high accuracy as it is, we had to perform multiple techniques to use this dataset. We used as showing below the following:

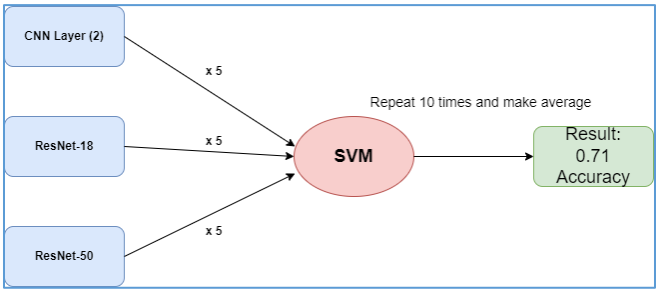

Fig. 1. Model testing method

Our model architecture approach is known as a hybrid CNN-SVM model. The layers consist of a series of complex architectures ResNet50, ResNet18 and convolutional neural network layers (CNN layers) for image classification tasks.

The convolutional and pooling layers in addition to the support vector machine (SVM as classifier) are added on top of the features extracted by the ResNet and CNN model:

- ResNet 50
- Layers refining the features learned.
- ResNet 18
- Convolutional Layer: 5 x 5 size, 256 filters, 2 strides
- Activation Function ReLU
- Pooling Layer: 2 x 2
- Convolutional Layer: 5 x 5 size, 512 filters, 2 strides
- Activation Function ReLU
- Pooling Layer: 2 x 2
- Fully Connected Hidden
- SVM
- Output 30 Classes.

ResNet 50 and ResNet 18 are both CNN architectures that have been designed for image classification tasks and demonstrated high accuracy on large-scale image recognition tasks. These architectures use residual blocks, which help to overcome the issue of vanishing gradients that can occur in very deep networks. ResNet 18 consists of 18 layers, while ResNet 50 consists of 50 layers.

Then we included the convolutional layers to apply a set of filters to the input image, generating a set of feature maps that capture different aspects of the image. The pooling layers used to sample these feature maps, reducing the dimensionality of the input data and making it easier to process.

We then used these architectures in conjunction with an SVM model for image classification, the output of the CNN layers then flattened and fed into the SVM model for classification.

This process was repeated 10 times, with the average accuracy being calculated over the 10 runs.

We generated RGB and replaced the greyscale ones that generated high noise and the ones that were identical with other images from different malware families the RGB images sort out this issue. In the future we will be using RGB images and removing all the grayscale ones.

III. PE AND PNG FILE STRUCTURE RELATIONSHIP

The PE PNG representation is generated based on the structure of executable files as showing in Fig2. All PE files contain at minimum two sections:

- The code section,
- The data section.

The portable execution file has 9 pre-defined sections named:

- .text,
- .bss,
- .rdata,
- .data,
- .rsrc,
- .edata,
- .idata,
- .pdata and
- .debug.

Not every PE file needs all these sections. PE files from different application versions might also differ from each other. These differences include different sections that create executable files. This variation will add complications to the PE file representations as PNG files for human eyes to distinguish and identify the malware family by looking at the PNG representations. This also explains why the representations are different for two PE files that belong to the same malware family. The sections that are most commonly present in PE are:

- Executable Code Section (commonly named .text),
- Data Sections (of which .data, .rdata, .bss are types),
- Resources Section (commonly named .rsrc),
- Export Data Section (commonly named .edata),
- Import Data Section (commonly named .idata),
- Debug Information Section (commonly named .debug).

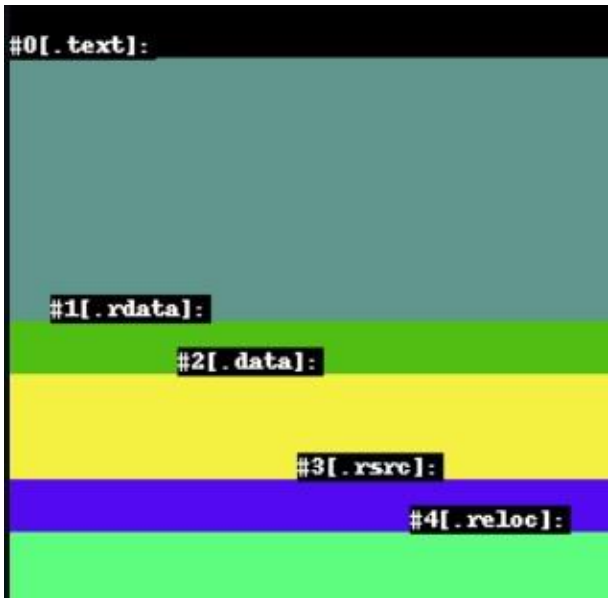

Fig. 2. PE-PNG file structure including PE file sections

Figure 2 shows the PE file structure (text, rdata, data, rsrc, reloc) included in the converted PNG graphical representation. The PNG processed images include PE structures which are the unique features for each malware family. We defined the PNG file structure based on PE file structure, however, we just included (text, rdata, data, rsrc and reloc) into the malware PNG files and we removed for now the other sections.

IV. EXPERIMENTAL RESULTS

Maldataset 2022 contains a set of malware samples of malware classes and the number of samples training set and the testing set. Space limit of this paper doesn't let us to present all of the malware families that are used in the experimental part of the research. That's why we will present just 6 images from them.

In the below Fig.3, Fig.4, Fig.5, Fig.6, Fig.7 and Fig.8 six malware families are presented *Trickbot, Agent Tesla, Androm, Andromeda, Nanocore and Autorun.K.*

Each image represents one malware family and one sample. The one sample was selected from many samples showing the unique representation for a particular malware family. Each PNG graphical representation shows a set of layers for the PE structures that we presented earlier on section III in Fig.2.

The below represents the full list of malware families that were tested and converted to images. Here is explanation of the format of the listing below: (Malware Family Name), (Number of Samples in the Training Set) and (Number of Samples in the Testing Set).

- Agent 350 120
- Agenttesla 85 35
- Androm 350 147
- Andromeda 85 35
- Autorun 350 147
- Autorun.k 80 25
- Azorult 35 10
- Cerber 70 30
- Darkcomet 45 25
- Dridex 30 15
- Dyre 41 18
- Emotet 68 26
- Grandcrab 73 21
- Hawkeye 70 21
- Heyodo 69 30
- IceID 69 20
- Limerat 10 6
- Loki 138 40
- Qaqbot 22 30
- Nanocore 157 42
- Nabulae 32 10
- Neshta 350 147
- Nymaim 73 30
- QuasarRat 92 35
- Regrun 350 147
- Remcosrat 155 70
- Robot!gen 140 18
- Sality 350 147
- Shifu 31 12
- Trickbot 141 69.

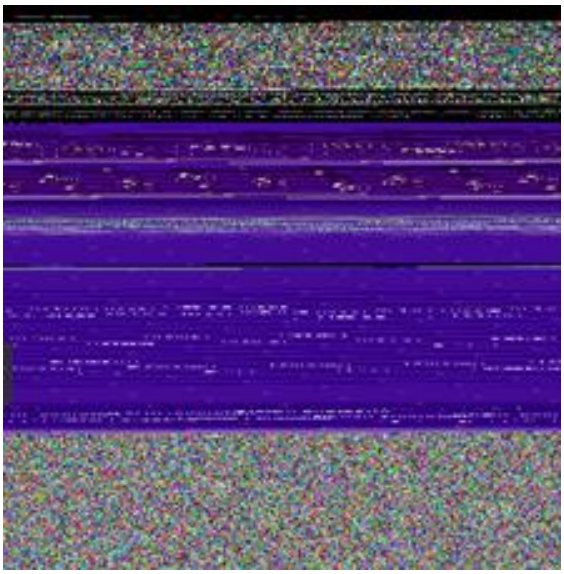

Fig. 3. Trickbot

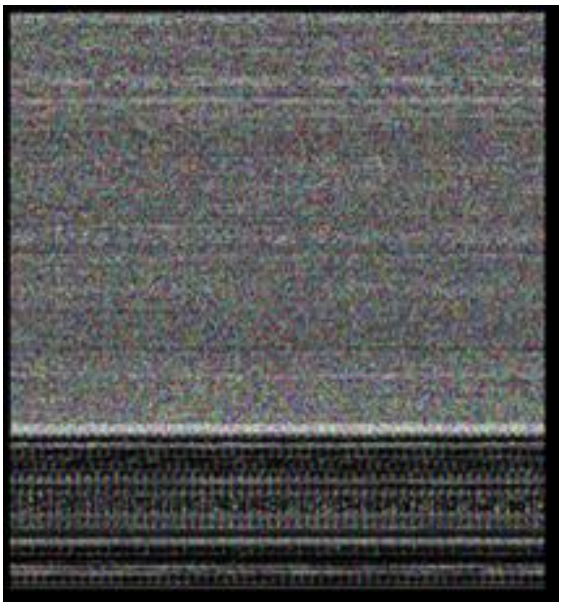

Fig. 4. Agent Tesla

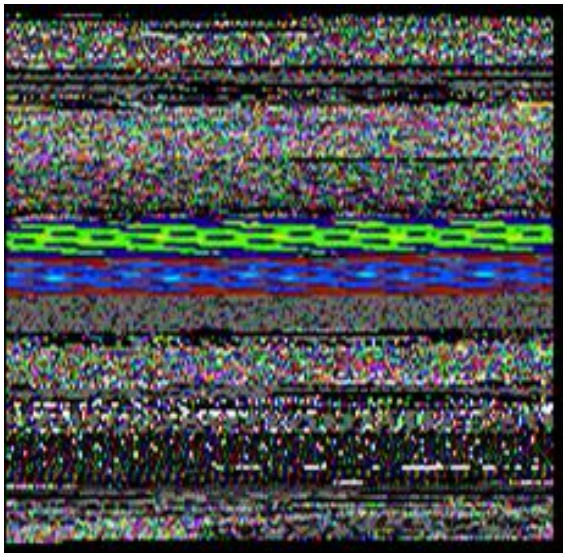

Fig. 5. Androm

Fig. 6. Andromeda

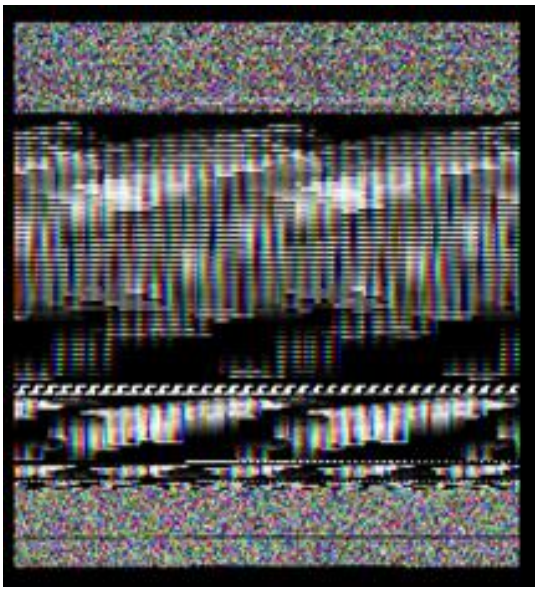

Fig. 7. Nanocore

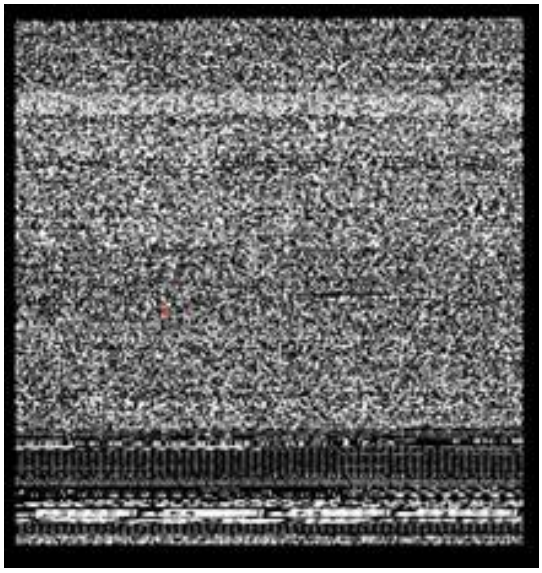

Fig. 8. Autorun.K

V. Conclusion and Future Work

Visualization of the malware is very attractive and hot topic in the recently new era of malware analysis techniques. There is a lot of research that is done in this area of utilizing visual methodology for the malware analysis. Focus of the research in this paper is put on PNG-generated representation from PE files of the new malware dataset from 2022. That was the main motivation for the obtained experiments for this research. Limitation in this dataset is related to the number of samples in each class. It could be higher than the existing one and it is due to not all malwares have many samples and the data gathering takes a lot of time. Therefore, we are continuously adding new samples to each family to address this limitation. However, by using data reply and augmentation techniques we can overcome this limitation.

In this paper, novel methodology for visualization of the malware families is introduced. For that purpose, a new dataset for Malware classification and detection is presented and utilized. The dataset is a PNG-generated representation from PE files. The PNG processed images include PE structures which are the unique features for each malware family. PNG file structure is based on PE file structure, but in this specific experiment we included (text, rdata, data, rsrc and reloc) into the malware PNG files and we removed the other sections.

These are the unique features which we mentioned earlier and related to the PE sections:

- Executable Code Section (commonly named .text),
- Data Sections (of which .data, rdata, bss are types),
- Resources Section (commonly named rsrc),
- Export Data Section (commonly named edata),
- Import Data Section (commonly named idata).

We tested multiple feature selection during the training, and we decided to use the recursive feature elimination (RFE). This method was useful in identifying the most important features in our dataset, by recursively removing features and evaluating model performance at each step. In the case of malware executables, it was very important to identify features that contributed to the classification of malicious files.

The proposed method helps to represent malware families as PNG files. The experiment results showed that the binary PE files can be close and similar to another PE file from a different malware family if the sections of the PE files are identical. However, by using CNN models for classification we can overcome this limitation.

The main motivation is to benefit from the high accuracy and high score predictions that CNN models can provide trained on the image-based dataset. This dataset doesn't require the security practitioners to be experienced in data science techniques and researchers to avoid getting involved in features engineering and extraction during dataset creation which can be a complex process. In addition, this contribution will allow the community to share and develop additional datasets for better malware classification and detection.

In this experiment we limit our selection of the PE file structure into a few sections and sections elements. This selection created a limitation and, in some cases, a very close similarity between different malware families. This will open the research direction into including other PE sections and section elements such as idata (import tables) and edata (export tables).

References

[1] El-Shafai, Walid, Iman Almomani, and Aala AlKhayer. "Visualized malware multi-classification framework using fine-tuned CNN-based transfer learning models." Applied Sciences 11.14 (2021): 6446.

[2] Xiao, Han, Kashif Rasul, and Roland Vollgraf. "Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms." arXiv preprint arXiv:1708.07747 (2017).

[3] Simpson, Amber L., Michela Antonelli, Spyridon Bakas, Michel Bilello, Keyvan Farahani, Bram Van Ginneken, Annette Kopp-Schneider et al. "A large annotated medical image dataset for the development and evaluation of segmentation algorithms." arXiv preprint arXiv:1902.09063 (2019).

[4] He, Ke, and Dong-Seong Kim. "Malware detection with malware images using deep learning techniques." In 2019 18th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/13th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE), pp. 95-102. IEEE, 2019.

[5] Vasan, Danish, Mamoun Alazab, Sobia Wassan, Babak Safaei, and Qin Zheng. "Image-Based malware classification using ensemble of CNN architectures (IMCEC)." Computers Security 92 (2020): 101748.

[6] Akarsh, S., K. Simran, Prabaharan Poornachandran, Vijay Krishna Menon, and K. P. Soman. "Deep learning framework and visualization for malware classification." In 2019 5th International Conference on Advanced Computing Communication Systems (ICACCS), pp. 10591063. IEEE, 2019.

[7] Ni, Sang, Quan Qian, and Rui Zhang. "Malware identification using visualization images and deep learning." Computers Security 77 (2018): 871-885.

[8] Sun, Guosong, and Quan Qian. "Deep learning and visualization for identifying malware families." IEEE Transactions on Dependable and Secure Computing 18, no. 1 (2018): 283-295.

[9] Pinhero, Anson, M. L. Anupama, P. Vinod, Corrado Aaron Visaggio, N. Aneesh, S. Abhijith, and S. AnanthaKrishnan. "Malware detection employed by visualization and deep neural network." Computers Security 105 (2021): 102247.

[10] Akarsh, S., Prabaharan Poornachandran, Vijay Krishna Menon, and K. P. Soman. "A detailed investigation and analysis of deep learning architectures and visualization techniques for malware family identification." In Cybersecurity and Secure Information Systems, pp. 241-286. Springer, Cham, 2019.

[11] Chaalan, T. (n.d.). *Maldataset2021*. https://www.Csmining.org/Cdmc2021/Index.php?id=1.

[12] Anguita, Davide, Luca Ghelardoni, Alessandro Ghio, Luca Oneto, and Sandro Ridella. "The'K'in K-fold Cross Validation." In *ESANN*, pp. 441-446. 2012.