VIT®
Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

Malware classification using Image classification on MalImg dataset

Project Report

Submitted in partial fulfillment of the requirement for the course of

MALWARE ANALYSIS IN DATA SCIENCE
[CSE4063]

Mandava Deepak - 20BAI1162
Lakshmi Sumana N - 20BAI1089
Chinthamani Mohan Krishna - 20BAI1269
Pitchika Vaishnavi - 20BAI1151

To
Dr. Jayasudha M

TABLE OF CONTENTS

CHAPTER I

CHAPTER II

CHAPTER III

CHAPTER IV

CHAPTER V

Abstract

Malware creators change their code to make new malicious software samples that are difficult for traditional antivirus software to detect. Researchers have suggested identifying malware using malware images methods, without having to analyses the code of sample by running it. We have devised a model on such suggestion to find different malware using CNN and ml algorithms, they were ran parallel while predicting the outcome of model just like in random forest but quite the same . We trained our model using a dataset called 'MalImg', which has pictures of 25 well-known malware families. Then, we used our model to identify validation samples from same dataset. In this work we will review the process and procedure we followed while devising such model

Introduction

The main objective of this work is to device and construct a model, based on image classification, that can classify the malware sample to its closely related malware family and to do so, we are using MalImg dataset which consist of images of 25 different malware families.

Malware is a type of software that can cause serious damage to computer and networks. Every day, thousands of new types of malware are created, making it difficult for traditional methods of detecting them to keep up. These traditional methods include looking for specific patterns in the code or behavior of the software.

There are different types of malware, such as viruses, worms, trojans, and backdoors. Each type has a specific purpose, such as damaging files, stealing information, or giving unauthorized access to a system. As more and more new types of malware are created, there is a significant increase in the number of

unique codes (signatures) that antivirus software needs to recognize and protect against each year.

Signature-based detection looks for specific code sequences to identify a known type of malware, but it cannot detect new, unknown malware. Heuristic-based detection scans system behavior to identify abnormal activities and can detect new malware, but it can slow down system performance and require more space. Behavior-based detection analyzes how a program behaves during execution to determine if it's malicious, but it can produce false positives and false negatives.

An Artificial Neural Network (ANN) is trained using pictures of known malware to detect new variations of these malware families. This method is fast and requires less computational power than other methods.

Research has shown that machine learning algorithms are effective and reliable for detecting and preventing malware, which is why they are being increasingly used. So, we have also used the same ideology and have constructed a model based on image classification using Deep learning and machine learning and tested it on MalImg dataset.

The MalImg Dataset contains 9339 malware images, organized in 25 families. Representative images from six malware families: Adialer.C, Agent.FYI, Rbot!gen, Lolyda.AA1, Fakerean and Swizzor.gen!E. It is a collection of 9389 grayscale images that represent 25 different malware families. The images were created by converting the binaries of the malware into 8-bit vectors, which were then converted to grayscale images by assigning each vector to a pixel representing the intensity. This dataset has been widely used as a benchmark in evaluating malware detection methods, including those for IoT environments. The dataset includes various malware families and their variants, and the images were created using a process described by Conti et al

Related work

Malware Detection Based on Code Visualization and Two-Level Classification by Vassilios Moussas and Antonios Andreatos

Author's objective is to devise an ANN classifying visualized malware. The dataset was split into a training set and a testing set using a 70% - 30% ratio. The training set had 6537 samples, and the testing set had 2802 samples. The testing set was further divided into a 15% validation subset and a 15% test subset. A pattern recognition feed-forward ANN was implemented with at least 3 layers: an input layer, an output layer, and one or more hidden layers. The number of nodes in the input layer was equal to the number of features used, and the number of nodes in the output layer was equal to the number of virus categories studied. Different ANN configurations were tested, including one to three hidden layers with hidden layer sizes ranging from 2 to 256 nodes. The best results were obtained for the single hidden layer with 64 nodes, double hidden layers with 64 nodes per layer, and the 3 hidden layers with 128 nodes per layer.The two-level ANN was used to classify a total of 9339 images. The first level of the ANN was able to correctly classify 5339 images with an accuracy of 98.83%. The remaining 4000 images were forwarded to the second level of the ANN for further classification. In the second level, group G1 consisting of 900 images achieved a perfect accuracy of 100%, while group G2 consisting of 3100 images achieved an accuracy of 99.41%.

Image-Based Malware Classification Using VGG19 Network and Spatial Convolutional Attention

by Mazhar Javed Awan,Osama Ahmed,Masood,Mazin Abed Mohammed ,Awais Yasin,Azlan Mohd Zain,Robertas Damaševičius and Karrar Hameed Abdulkareem

The proposed model for malware detection consists of three parts. The first part is a pre-trained VGG19 model which is used as a feature extractor. The layers of the VGG19 model are frozen, and it is only used for extracting features from the input data. The second part is a CNN model enhanced by a simple form of

attention mechanism called dynamic spatial convolution. This attention mechanism helps to capture important regions in the image that are more useful for the task. The attention-enhanced feature maps are fed into a dense layer with 256 units, followed by a dropout layer for regularization and a fully connected layer with 25 units. The SoftMax activation is used in this layer. The third part is a visualization of the proposed architecture. The aim of the model is to show the effectiveness of attention in enhancing the accuracy of the model for malware detection. In this paper we have proposed to use attention enhancement via CNNs to solve the malware recognition problem without the need of a feature engineering technique or handmade feature design. We have aimed to show how the attention mechanism with CNNs can be used and how this type of spatial attention, which demonstrated enormous advantages in computer vision problems, can be applied in images-based malware detection as well.

Malware Images Classification Using Convolutional Neural Network

By Espoir K.Kabanga,Chang Hoon Kim

The author have used the MalImg Dataset which consists of 9458 grayscale images among which 90% of the total data is used for training and 10% is used for testing. He has presented a Convolutional Neural Network model that classified images extracted from malware samples. The result is quite competitive.
Being able to visualize malware as gray-scale images has been a great achievement. Many researchers have been using this technique for the task of malware classification and detection. However, other works have shown that this technique can be easily vulnerable to adversarial attack and produce erroneous results.

Proposed architecture and methodology

The proposed algorithm likely aims to combine the advantages of both methods to create a better and accurate model. In the paper we have experimented with different architectures and hyperparameters to create a random forest-like model with CNN and ML algorithm that can handle complex tasks such as image classification. CNN is a type of neural network used for computer vision tasks, KNN is a machine learning algorithm used for classification and regression based on nearest neighbours, and SVM is a machine learning algorithm used for classification and regression by finding the optimal hyperplane that separates different classes in the dataset.

The decision rule for this process varies depending on the type of model used. For CNN, the decision rule involves obtaining the output probabilities for each class and choosing the class with the highest probability. For KNN, the decision rule involves finding the K nearest neighbors of the new data point and taking the majority vote of their classes or the average value of their outputs. For SVM, the decision rule involves finding the position of the new data point with respect to the hyperplane obtained during training and assigning it to the corresponding class.

In all three cases, the decision rule is a crucial step in the process of making predictions with the model, and its accuracy depends on the quality of the training data and the effectiveness of the model architecture similarly, in the proposed architecture, all the mentioned models and algorithms i.e; CNN, KNN, SVM first be trained on same data set and then using the same models for predicting the outputs but first, output of these models is taken and then all the outputs are converted into integer format, because CNN output is in the form of One-hot encoding, so it is necessary to convert it into integer format as we are planning to combine results of all the models.

It is observed that SVM has high performance than other models, when individual models are trained on same dataset. So, if in case of ambiguity the final output for an image will be the output of SVM. Before that, we will be getting the

predicted results of all models for the image, then we are looking for the “most repeated” output class from the results of all the models for that image. If found then it is predicted as the output for that image else as discussed above, SVM output will become the model’s output for that image

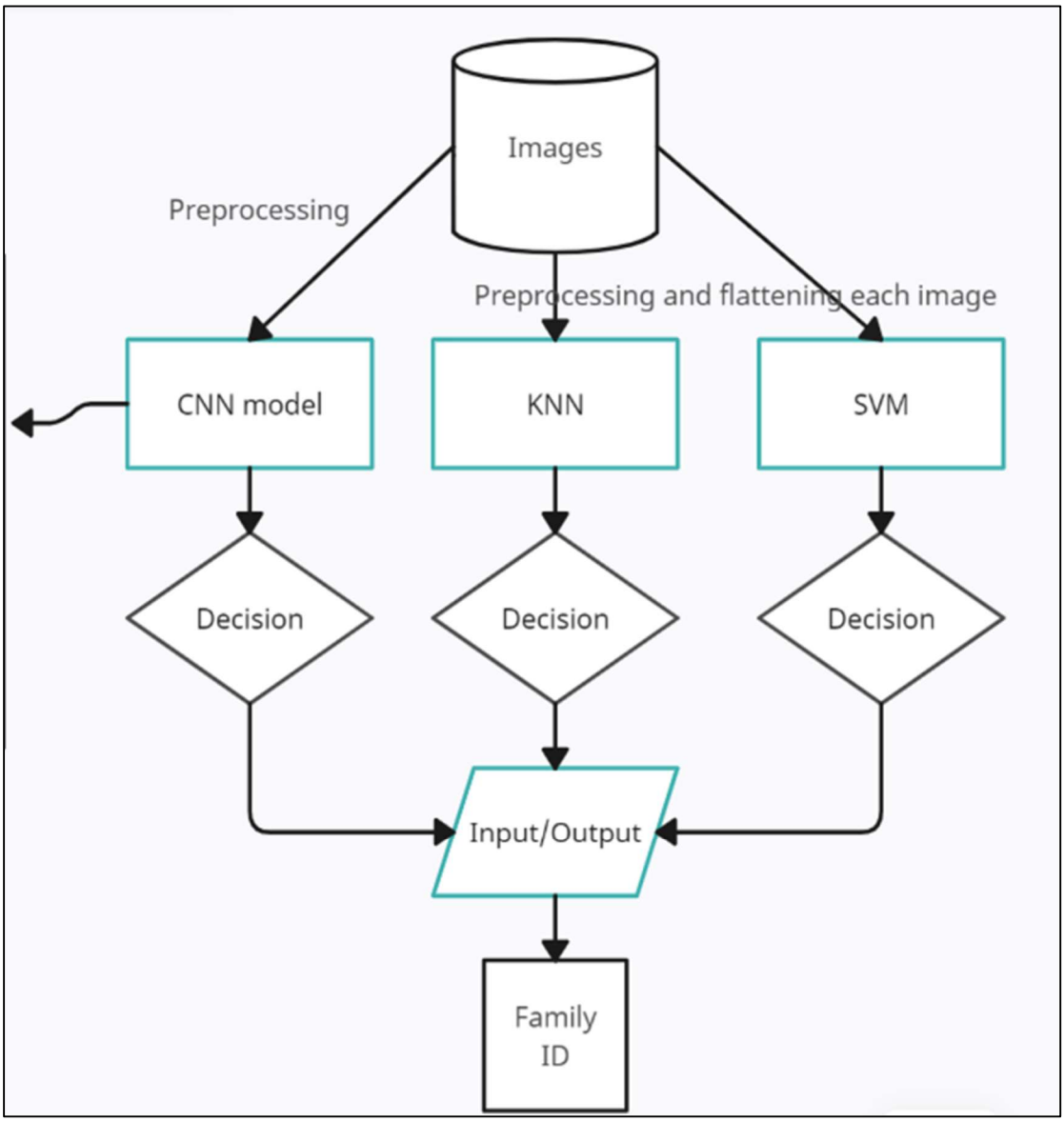

Figure 1: Proposed architecture

Research gap

we are essentially combining multiple classifiers to make a more accurate prediction. One way to do this is to use the accuracy of a Support Vector Machine (SVM) classifier to guide our decision-making in case of ambiguity, We can then use the SVM classifier to make the final decision. If the SVM classifier has a high accuracy for a particular input, we will use its prediction as the final output. However, if the SVM classifier has a low accuracy, we will rely on the predictions of the other classifiers to make a decision.

One way to further improve the accuracy of this hybrid approach is to use techniques such as ensemble learning. Ensemble learning involves combining the predictions of multiple classifiers in a way that results in a more accurate final prediction. For example, we could use a weighted average of the predictions from all of the classifiers, where the weights are determined based on the accuracy of each classifier.

It's important to note that the effectiveness of this approach depends on the quality of the classifiers used. If the classifiers are not accurate, the final prediction will not be accurate either. Therefore, it's crucial to use proper classifiers that are well-suited for the specific problem at hand. Overall, using a proper decision rule with hybrid accuracy can be an effective way to improve the accuracy of the final model. By combining multiple classifiers, we can leverage the strengths of each to produce a more accurate final prediction.

COMPARSION:

MODEL	TIME	ACCURACY
PAPER1(**Malware Detection Based on Code Visualization and Two-Level Classification)(ANN)**	less	96
PAPER2(**Image-Based Malware Classification Using VGG19 Network and Spatial Convolutional Attention)**	less	99
PAPER3(**Malware Images Classification Using Convolutional Neural Network)**	less	98
CNN	less	87
KNN	less	78
SVM	less	94
HYBRID (proposed architecture)	less	92
STATIC	Moderate - high	100
DYNAMIC	Moderate - high	100

Here CNN model, KNN and SVM classifiers are trained to check the basic performance of them on the dataset and to help compare them with proposed hybrid model. Although the accuracy of SVM is higher that hybrid model but hybrid model was able to classify more no. of images correctly per class compared to SVM. By using a better decision-making rules in-place of proposed one could improve the model further

According to the information provided, the proposed models are able to detect malware with relatively high accuracy, but they take less time than the traditional static and dynamic analysis methods. This suggests that the proposed models may be more efficient and practical for detecting malware in real-world scenarios where time is a critical factor.

The proposed models for detecting malware are faster but they may not be as accurate as the traditional analysis methods. It is important to carefully consider the performance and limitations of both methods before deciding which one to

use for detecting malware, depending on the specific goals and requirements of the analysis.

Results

we have trained CNN, KNN, and SVM models for a particular task. Based on your proposed methodology, the best results were obtained using the SVM model, which had a higher accuracy compared to your proposed model. However, we also mentioned that proposed model was able to predict a greater number images correctly per each classes correctly compared to the SVM model. This suggests that while the SVM model may have had better overall accuracy, our proposed model will have better at correctly identifying classes.

Svm classification report:

Classification report:

	precision	recall	f1-score	support
Swizzor.gen!E	0.56	0.70	0.62	20
Yuner.A	1.00	1.00	1.00	20
Rbot!gen	1.00	1.00	1.00	20
Obfuscator.AD	1.00	1.00	1.00	20
VB.AT	1.00	1.00	1.00	20
Swizzor.gen!I	0.53	0.40	0.46	20
Skintrim.N	1.00	1.00	1.00	20
Wintrim.BX	1.00	1.00	1.00	20
Lolyda.AT	1.00	1.00	1.00	20
Fakerean	1.00	1.00	1.00	20
Instantaccess	1.00	1.00	1.00	20
Dontovo.A	1.00	1.00	1.00	20
Lolyda.AA3	1.00	1.00	1.00	20
Malex.gen!J	1.00	1.00	1.00	20
Lolyda.AA1	1.00	1.00	1.00	20
Lolyda.AA2	1.00	1.00	1.00	20
C2LOP.P	0.83	0.75	0.79	20
Dialplatform.B	1.00	1.00	1.00	20
Autorun.K	1.00	1.00	1.00	20
Agent.FYI	1.00	1.00	1.00	20
Alueron.gen!J	1.00	1.00	1.00	20
C2LOP.gen!g	0.83	0.95	0.88	20

...

accuracy			0.95	500
macro avg	0.95	0.95	0.95	500
weighted avg	0.95	0.95	0.95	500

CNN classification report:

Classification report:

	precision	recall	f1-score	support
Swizzor.gen!E	1.00	1.00	1.00	13
Yuner.A	1.00	1.00	1.00	9
Rbot!gen	0.89	1.00	0.94	8
Obfuscator.AD	0.95	1.00	0.97	19
VB.AT	1.00	1.00	1.00	8
Swizzor.gen!I	0.00	0.00	0.00	10
Skintrim.N	0.80	0.53	0.64	15
Wintrim.BX	0.75	0.82	0.78	11
Lolyda.AT	1.00	1.00	1.00	8
Fakerean	1.00	1.00	1.00	10
Instantaccess	0.94	0.94	0.94	18
Dontovo.A	1.00	0.92	0.96	13
Lolyda.AA3	1.00	1.00	1.00	11
Malex.gen!J	1.00	1.00	1.00	17
Lolyda.AA1	1.00	1.00	1.00	15
Lolyda.AA2	1.00	0.85	0.92	13
C2LOP.P	1.00	1.00	1.00	7
Dialplatform.B	1.00	1.00	1.00	12
Autorun.K	0.87	1.00	0.93	13
Agent.FYI	1.00	1.00	1.00	15
...				
accuracy			0.88	300
macro avg	0.86	0.88	0.87	300
weighted avg	0.87	0.88	0.87	300

KNN classification report:

Classification report:

	precision	recall	f1-score	support
Swizzor.gen!E	0.38	0.25	0.30	20
Yuner.A	1.00	1.00	1.00	20
Rbot!gen	1.00	1.00	1.00	20
Obfuscator.AD	1.00	1.00	1.00	20
VB.AT	0.28	0.80	0.42	20
Swizzor.gen!I	0.86	0.90	0.88	20
Skintrim.N	1.00	0.50	0.67	20
Wintrim.BX	0.80	1.00	0.89	20
Lolyda.AT	1.00	0.10	0.18	20
Fakerean	1.00	1.00	1.00	20
Instantaccess	1.00	1.00	1.00	20
Dontovo.A	1.00	1.00	1.00	20
Lolyda.AA3	0.95	1.00	0.98	20
Malex.gen!J	0.61	1.00	0.75	20
Lolyda.AA1	0.57	0.20	0.30	20
Lolyda.AA2	0.80	1.00	0.89	20
C2LOP.P	1.00	0.65	0.79	20
Dialplatform.B	0.74	1.00	0.85	20
Autorun.K	1.00	1.00	1.00	20
Agent.FYI	1.00	0.55	0.71	20
Alueron.gen!J	0.50	0.15	0.23	20
C2LOP.gen!g	0.80	1.00	0.89	20
...				
accuracy			0.78	500
macro avg	0.83	0.78	0.76	500
weighted avg	0.83	0.78	0.76	500

Hybrid classification report:

Classification report:

	precision	recall	f1-score	support
Swizzor.gen!E	0.52	0.75	0.61	20
Yuner.A	1.00	1.00	1.00	20
Rbot!gen	1.00	1.00	1.00	20
Obfuscator.AD	1.00	1.00	1.00	20
VB.AT	1.00	1.00	1.00	20
Swizzor.gen!I	0.80	0.20	0.32	20
Skintrim.N	1.00	1.00	1.00	20
Wintrim.BX	1.00	1.00	1.00	20
Lolyda.AT	1.00	1.00	1.00	20
Fakerean	0.95	1.00	0.98	20
Instantaccess	1.00	1.00	1.00	20
Dontovo.A	1.00	1.00	1.00	20
Lolyda.AA3	0.54	1.00	0.70	20
Malex.gen!J	0.91	1.00	0.95	20
Lolyda.AA1	1.00	1.00	1.00	20
Lolyda.AA2	1.00	1.00	1.00	20
C2LOP.P	0.76	0.65	0.70	20
Dialplatform.B	1.00	1.00	1.00	20
Autorun.K	1.00	1.00	1.00	20
Agent.FYI	1.00	1.00	1.00	20
Alueron.gen!J	1.00	1.00	1.00	20
C2LOP.gen!g	0.92	0.55	0.69	20

...

accuracy			0.92	500
macro avg	0.94	0.92	0.91	500
weighted avg	0.94	0.92	0.91	500

Conclusion

Based on our evaluation of the proposed model and the Support Vector Machine (SVM), we conclude that the proposed model outperforms SVM as SVM has correctly identified many classes but it only predicted very few images correctly in other classes but our proposed model although predicted and classified less no. of classes perfectly than SVM it was also better at predicting other classes (nearly perfect). Also, the decision-making function we used is a basic, by devising and using a better one can help the model to completely outperform SVM and also can greatly boost the performance

Appendix

DATA SET DETAILS

MalImg dataset consists of 9,339 malware samples from 25 different malware families. Nataraj et al. (2011) created the MalImg dataset by reading malware binaries into an 8-bit unsigned integer composing a matrix M ∈ R^{m×n}. The said matrix may be visualized as a grayscale image having values in the range of [0, 255], with 0 representing black and 1 representing white.

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

Figure 2: Families in the MALIMG dataset

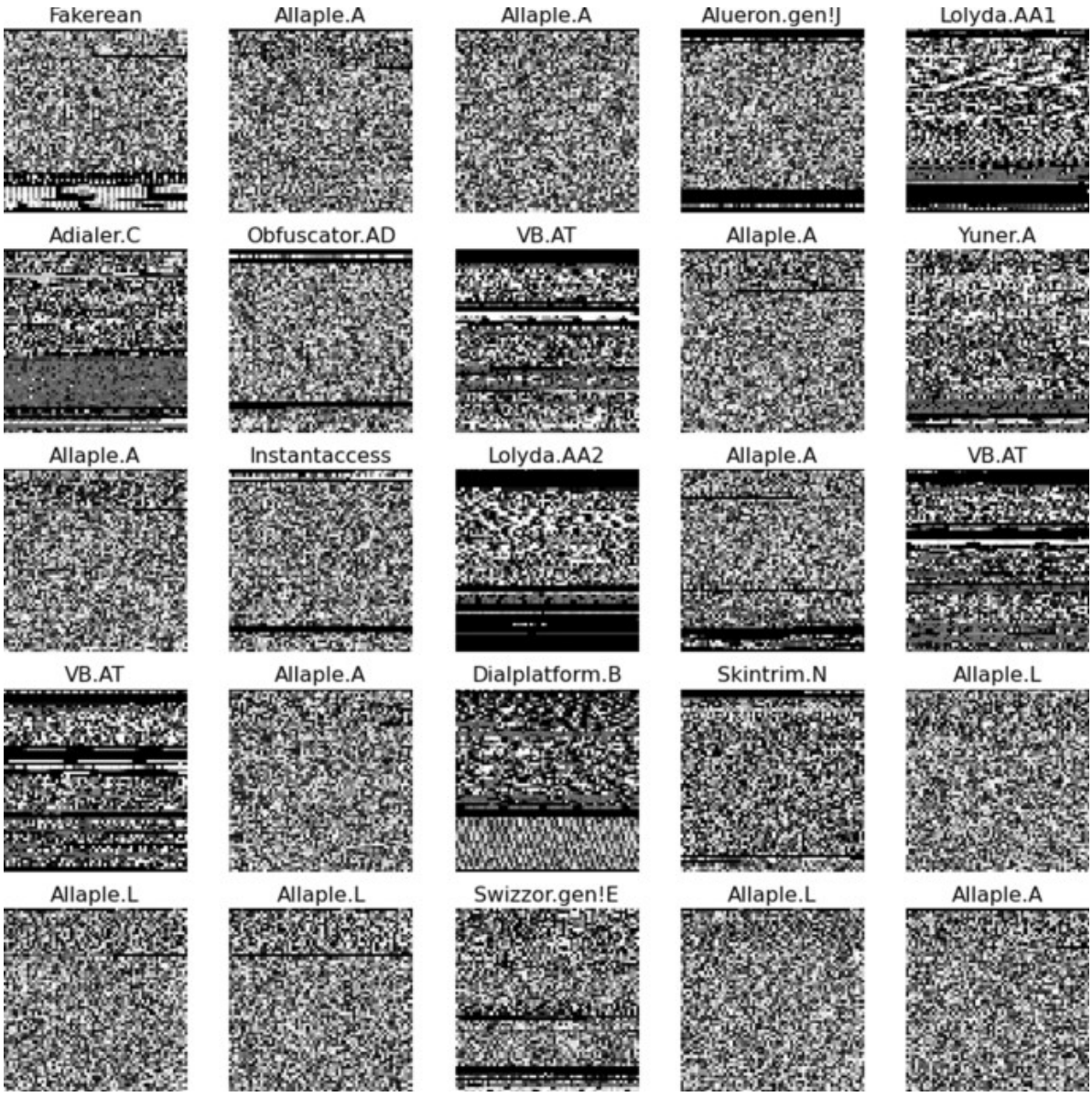

Figure 3: Image of malware samples and their family names from MALIMG dataset

- Complete source code, dataset saved models in one folder named 20BAI1162_Malware_CompleteDATA in GOOGLE DRIVE

https://drive.google.com/drive/folders/1o8d73vv6LXyW3RRqtlMo-baMqsvxEZUA?usp=sharing