Malware Detection

Final Report

Jan 2020

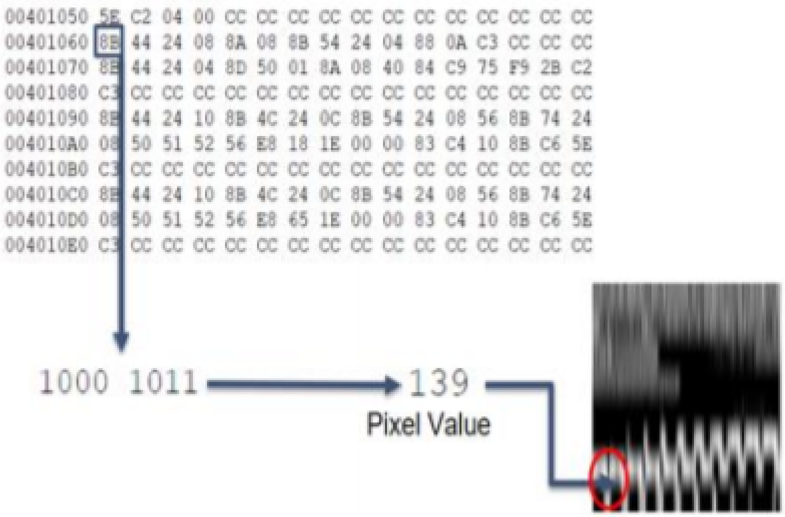
By
Millis Sahar
Segall Tomer
Strahl Zvi

Dataset

We found a couple of intresting datasets, that contains images that represent the byte files. Each file/image is **labeled** as a specific malware.

Conversion of byte files to images:
Byte files are the hexadecimal representation of the portable executable of the malware (see image below). Each hexadecimal pair is treated as a single decimal number which serves as a pixel value of the image. These images are then resized to a standard dimension of 32x32, which is the size of the images in the dataset.

While the article does not specify how the resizing was done, one way of doing so is by passing a low-pass filter (e.g. Gaussian) and then subsampling to the required resolution.



```
00401050 5E C2 04 00 CC CC CC CC CC CC CC CC CC CC CC CC
00401060 8B 44 24 08 8A 08 8B 54 24 04 88 0A C3 CC CC CC
00401070 8B 44 24 04 8D 50 01 8A 08 40 84 C9 75 F9 2B C2
00401080 C3 CC CC CC CC CC CC CC CC CC CC CC CC CC CC CC
00401090 8B 44 24 10 8B 4C 24 0C 8B 54 24 08 56 8B 74 24
004010A0 08 50 51 52 56 E8 18 1E 00 00 83 C4 10 8B C6 5E
004010B0 C3 CC CC CC CC CC CC CC CC CC CC CC CC CC CC CC
004010C0 8B 44 24 10 8B 4C 24 0C 8B 54 24 08 56 8B 74 24
004010D0 08 50 51 52 56 E8 65 1E 00 00 83 C4 10 8B C6 5E
004010E0 C3 CC CC CC CC CC CC CC CC CC CC CC CC CC CC CC
```

Research Question

Deep learning Malware detection using file's image.

```
How Well Can We Classify Malware Files by Applying CNNs(DL) on the Bytes Images?
```

We chose this due to our **common** knowldge in Computer Vision.
Also, by doing malware detection on images, we can avoid doing the analysis in a protected environment, e.g sandbox, virtual machines, etc.

In malware analysis, malware classification is important because categorizing various kinds of malware is important to know how they can contaminate personal computers, the risk level they pose, and how to defend them.
In case malware is detected, it is assigned to the most appropriate malware family through a classification mechanism.
There are numerous approaches for detecting malware in the wild; however, detecting a zero-day malware is still a challenging task.

Article

This is our Article:
Malware Classification using Deep Learning based Feature Extraction and Wrapper based Feature Selection Technique

Abstract:
In the case of malware analysis, categorization of malicious files is an essential part after malware detection.
Numerous static and dynamic techniques have been reported so far for categorizing malware.

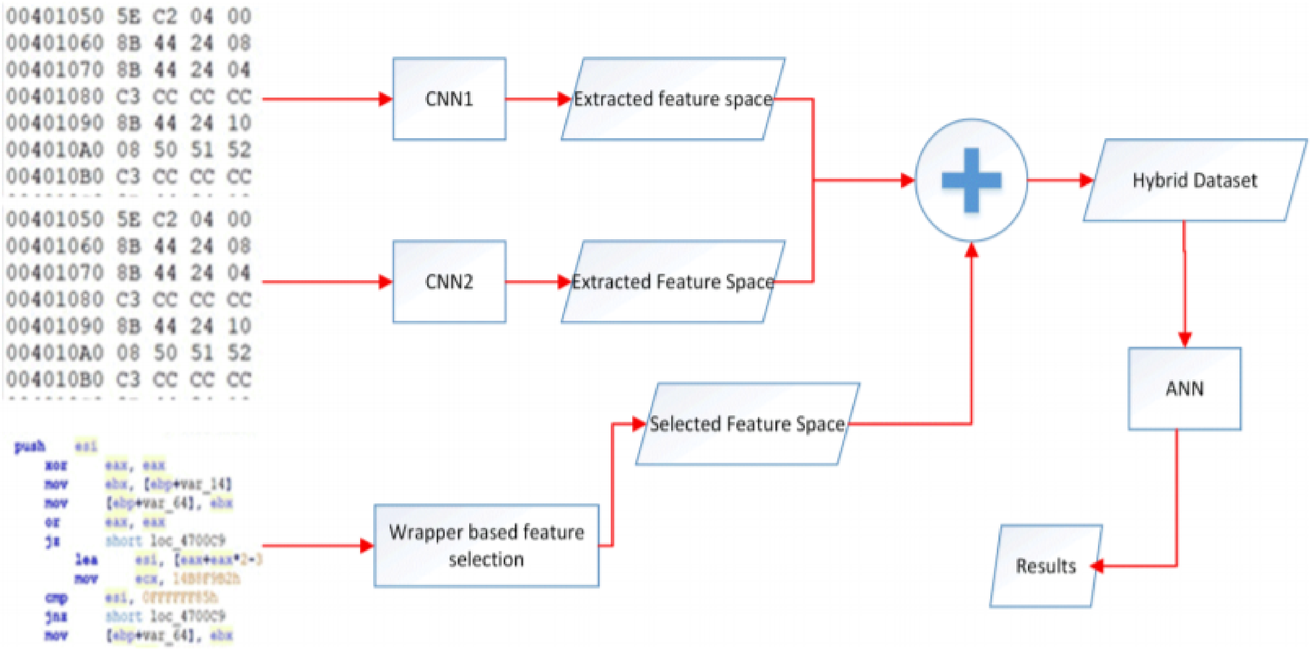
This research presents a deep learning-based malware detection (DLMD) technique based on static methods for classifying different malware families.
The proposed DLMD technique uses both the byte and ASM files for feature engineering, thus classifying malware families.

First, features are extracted from byte files using two different Deep Convolutional Neural Networks (CNN).
After that, essential and discriminative opcode features are selected using a wrapper-based mechanism, where Support Vector Machine (SVM) is used as a classifier.

The idea is to construct a hybrid feature space by combining the different feature spaces to overcome the shortcoming of particular feature space and thus, reduce the chances of missing a malware.

Finally, the hybrid feature space is used to train a Multilayer Perceptron, which classifies all nine different malware families.

Experimental results show that proposed DLMD technique achieves log-loss of 0.09 for ten independent runs.
Moreover, the proposed DLMD technique's performance is compared against different classifiers and shows its effectiveness in categorizing malware.



```
00401050 5E C2 04 00
00401060 8B 44 24 08
00401070 8B 44 24 04
00401080 C3 CC CC CC
00401090 8B 44 24 10
004010A0 08 50 51 52
004010B0 C3 CC CC CC
```

```
00401050 5E C2 04 00
00401060 8B 44 24 08
00401070 8B 44 24 04
00401080 C3 CC CC CC
00401090 8B 44 24 10
004010A0 08 50 51 52
004010B0 C3 CC CC CC
```

```
push    esi
    xor     eax, eax
    mov     ebx, [ebp+var_14]
    mov     [ebp+var_64], ebx
    or      eax, eax
    js      short loc_4700C9
        lea     esi, [eax+eax*2-]
        mov     ecx, 14B0F9B2h
    cmp     esi, 0FFFFFF85h
    jns     short loc_4700C9
    mov     [ebp+var_64], ebx
```

Step A: For presentation and submission by the end of the first semester - 2021.1.24 (33 points)

A. Explain the importance of the research question you have defined (2 points)

If this approach will work, we will be able to add this approach to any operating system & browsers. We could detect any malware files, without using a lot of resources. It will be quick, clean, and will be able to generelize well. For example, if a hacker will change some of the file code. it wouldn't help. we will still manage to detect the file as malware.

Some drawback will be the current resolution, but given more computetinal power we will be able to do better.

B. What is the technique by which the article dealt with the research question you defined, explained in detail (8 Points)

The article displays a variety of techniques, and even uses the images of ASM files. We won't be able to proccess the ASM files due to low computetional power.

We will test a couple of the Deep Learning techniques the article suggests. For example, CNNs and AutoEncoders.

C. On which dataset was the technique proposed in the article tested, describe it briefly (2 points)

Microsoft Challange (500GB)

The dataset contains images labeled to the following kind of malwares:

- Ramnit: It steals user credentials such as password credit card information and halts security software.
- Lollipop: It is an adware that shows ads on the browser; it also allows a hacker to monitor user traffic.

• Kelihos_ver1 and Kelihos_ver3: Trojan types can fully control user pc and spread by sending spam email from user pc to others.

• Vundo: It could be responsible for pop-up ads and installing other malicious content.

• Simda: These type of malware snatch the passwords from user pc and create a backdoor for hackers.

• Traceur: Using this malware attack, Author generates revenue by showing bogus advisement on search engines.

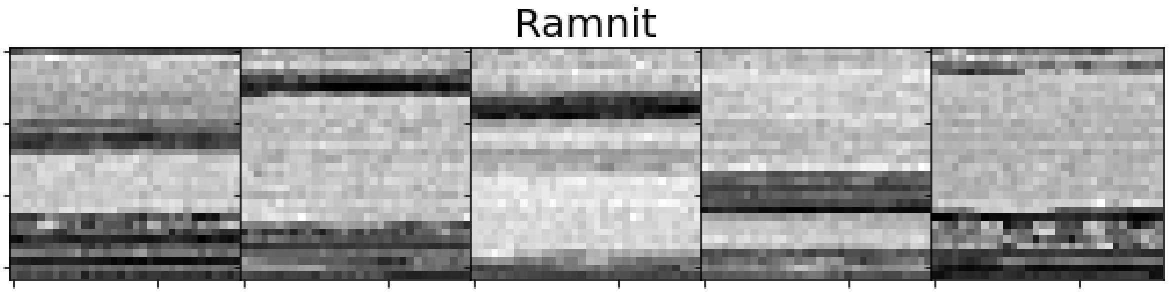
• Obfuscator.ACY: These are considered as obfuscated malware, and their purpose could be any of the malware mentioned above.

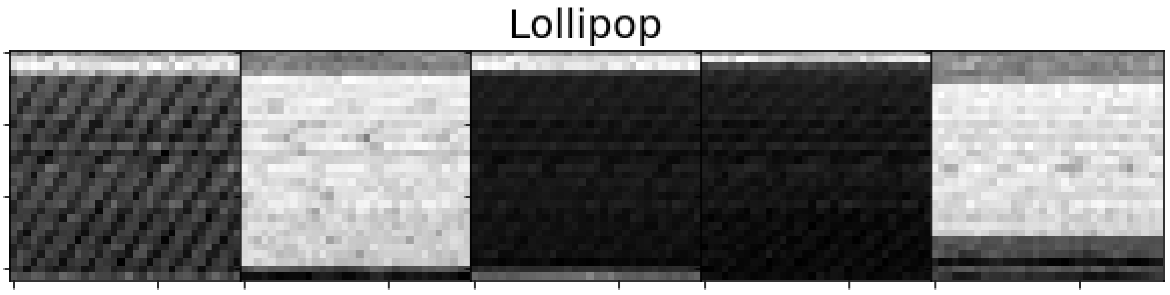
• Gatak: This is also a type of Trojan that seems legitimate but infects a computer with its malicious code.

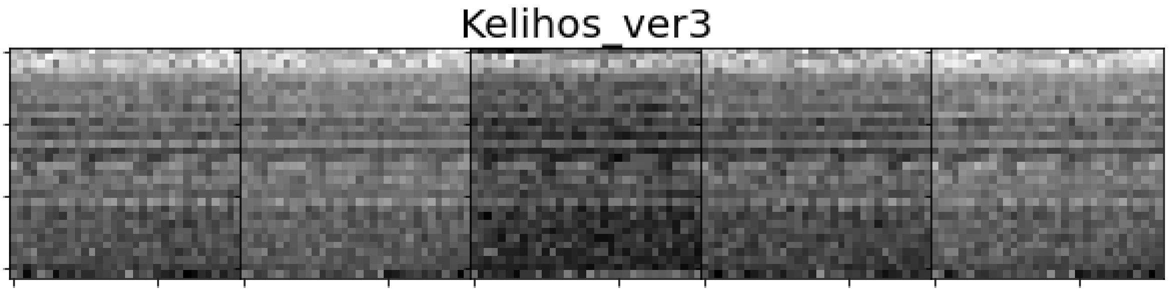
The dataset is divided to three:

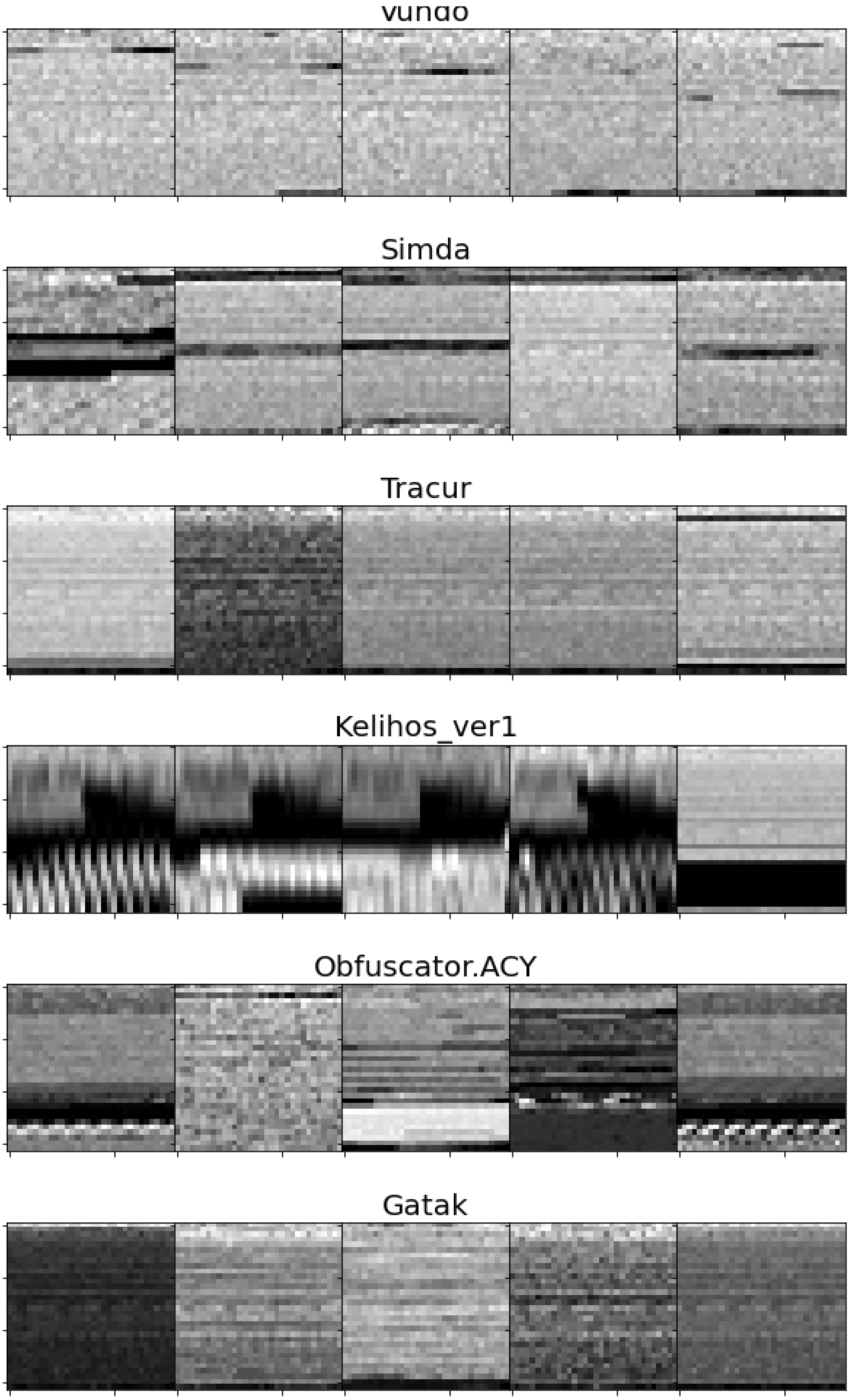
1. Training set - contains 6119 images
2. Validation set - contains 2031 images
3. Test set - contains 2710 images

The following is some images of each label from the dataset:



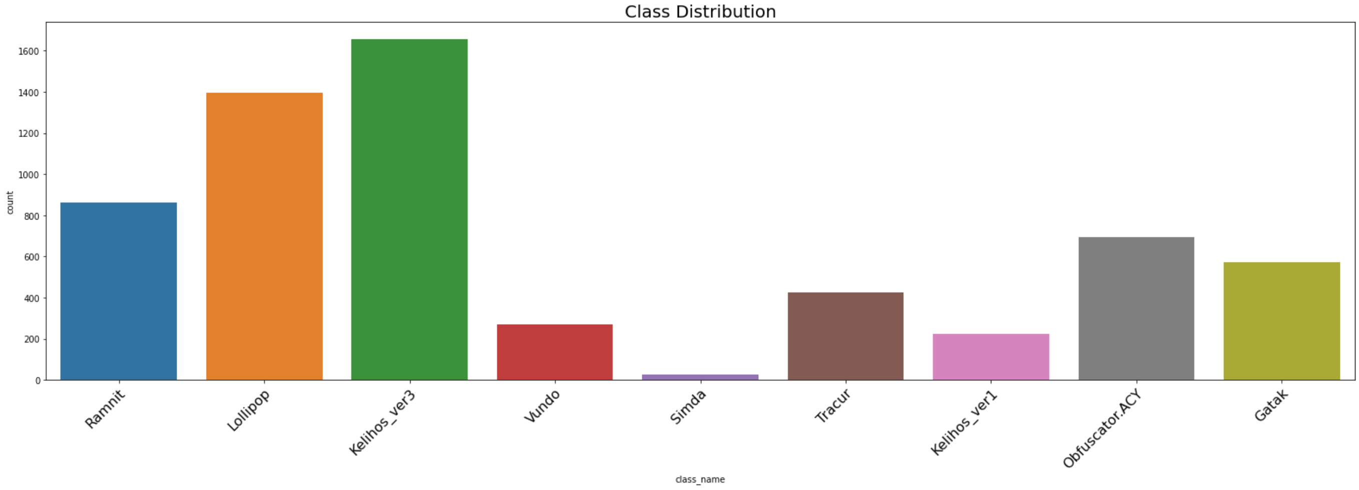




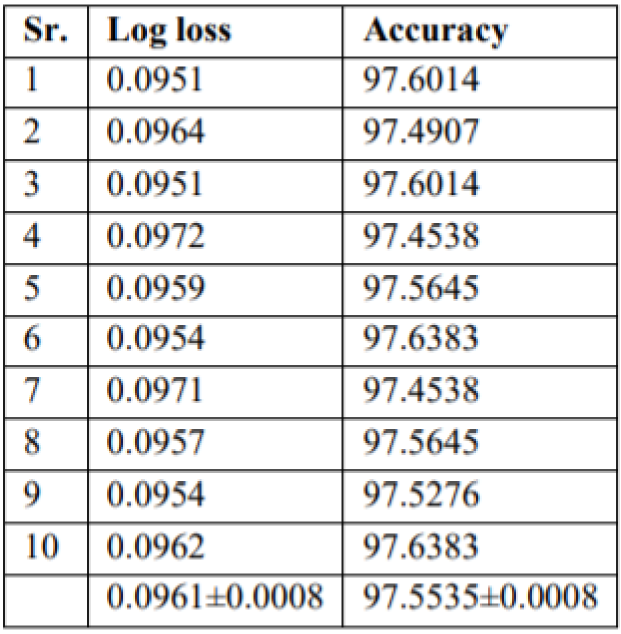


link text

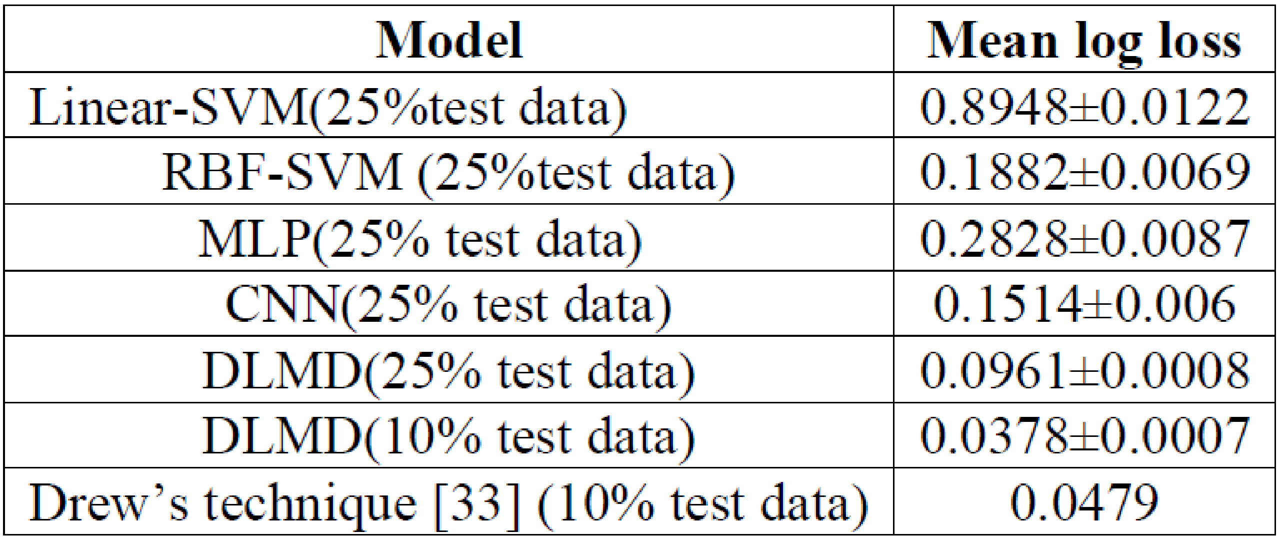
And the sets are divided to Train, Val, and Test - with the same class distribution:



D. What is the accuracy of the application of the technique in the article on the set data of the article (3 points)



Sr.	Log loss	Accuracy
1	0.0951	97.6014
2	0.0964	97.4907
3	0.0951	97.6014
4	0.0972	97.4538
5	0.0959	97.5645
6	0.0954	97.6383
7	0.0971	97.4538
8	0.0957	97.5645
9	0.0954	97.5276
10	0.0962	97.6383
	0.0961±0.0008	97.5535±0.0008

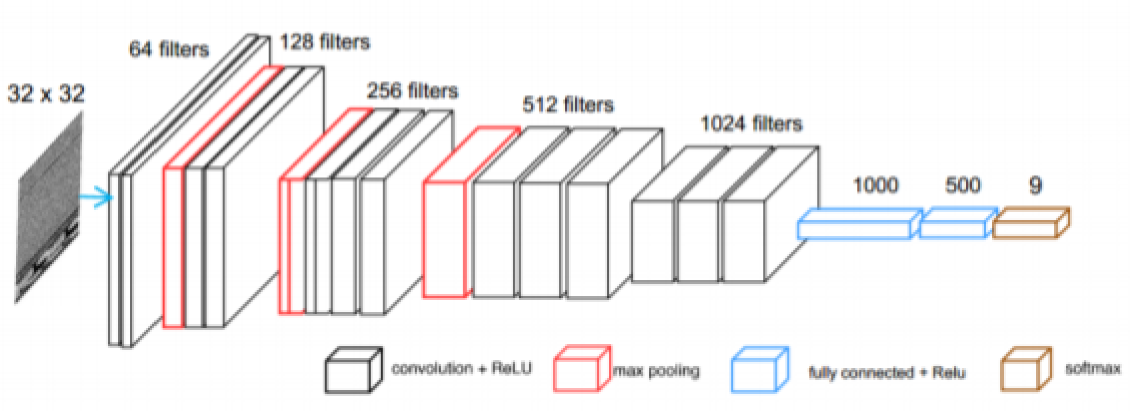


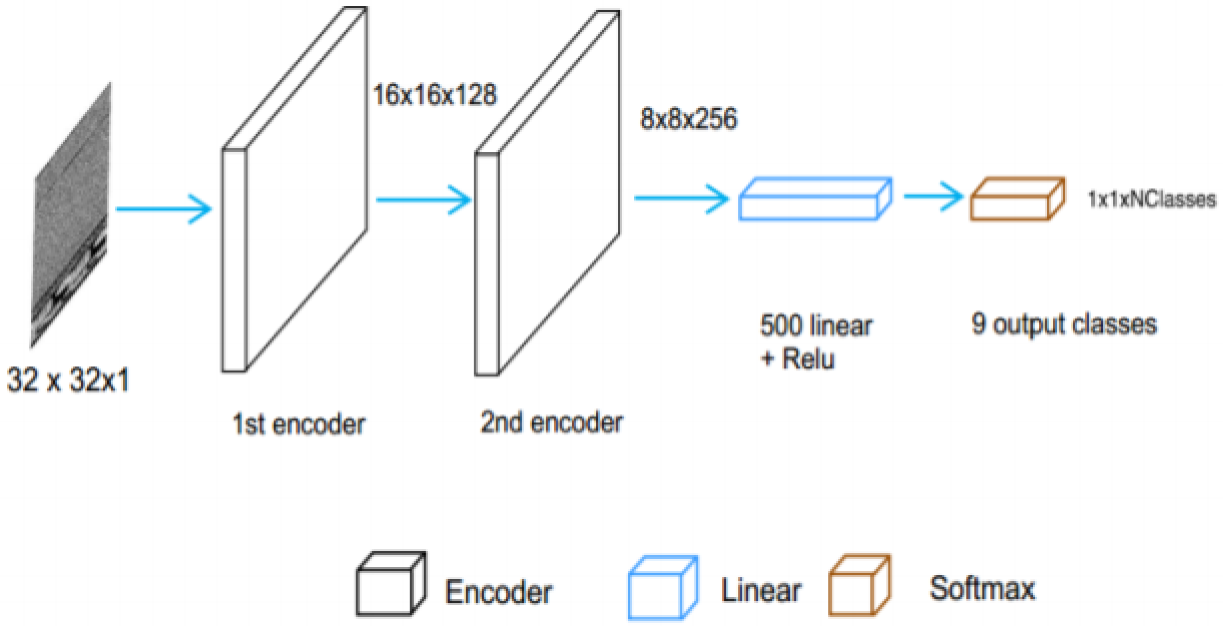
Model	Mean log loss
Linear-SVM(25%test data)	0.8948±0.0122
RBF-SVM (25%test data)	0.1882±0.0069
MLP(25% test data)	0.2828±0.0087
CNN(25% test data)	0.1514±0.006
DLMD(25% test data)	0.0961±0.0008
DLMD(10% test data)	0.0378±0.0007
Drew's technique [33] (10% test data)	0.0479

E. What ML technique do you offer to address the research question you have defined (3 Points)

We will address two techniques:

- CNNs
- Autoencoders





F. Why do you think such a technique is appropriate for the research question you presented (4 points)

CNNs is a long time winner while doing deep learning on images.
We can assume they will understand the context of the pixels and the releshenship between pixels.
We can imagine that each CNN layer will have a "higher" understanding of the files, due to the filters point of view.
Also, the CNNs will condence the images into features, and hopefully with these features a Linear layer will know how to differentiate between the malware files.

AutoEncoders are state of the art archituchtures, who proved time and time again to achieve AMAZING results with images.
They have the ability to reduce the image dimensionality to number of features, while trying to reconstruct the images.
There were a lot of article regarding AEs, and their results on various computer vision tasks.

G. Explain in detail how you intend to apply the research technique you proposed to the dataset You selected (11 points)

a. What are the features you will use

Deep learning alows us to let the network to find the best features by itself. We will just use the **labeled images**, and let the GD algorithms to optimize the network parameters based on the CE-Loss.

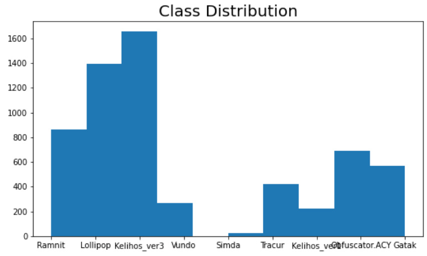
b. Will you run cleaning processes and if so which?

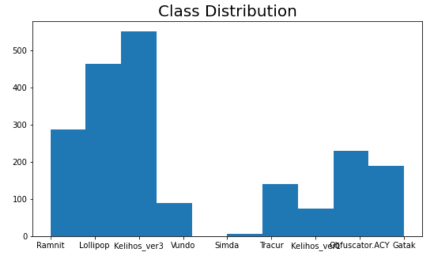
We will apply min-max normalization on the images, and we'll make sure the distibution stay the same.

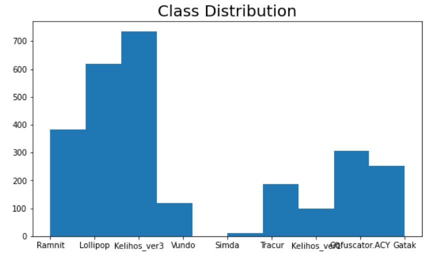
No cleaning wil be needed.
Actualy, in a later stage we will add noise to the images in order to force the network to generelize better.

c. How to divide the set-data

The dataset has no Leakege. Also, the distributions will be as follows:







d. How to test the results

Due to imbalanced classes, we will use Recall, Precision, and F1 score. As the article seggested, we will measure the results individuality.

Step B: To be submitted by the end of February 2021 (67 points)

H. Apply the ML technique shown in the article on the set-data you selected (18 points)

a. Explain in detail your implementation (and attach the code) (14 points)

Notebook 1: Download Dataset

Notebook 2: Explanatory Analysis

Notebook 3: CNNs Implementation - Article & Advanced

Notebook 4: AEs Implementation - Article & Advanced

Notebook 5: GANs Implementation - Generate new Malwares

b. Analyze the complexity of the technique (4 points)

We showed a varity of techniques. Each one inpacted the field of deep learning immensely.

convolutional neural nets - Not complex!
In deep learning, a convolutional neural network is a class of deep neural networks, most commonly applied to analyzing visual imagery. They are also known as shift invariant or space invariant artificial neural networks, based on their shared-weights architecture and translation invariance characteristics.

Generative adversarial networks - Somewhat complex!
A generative adversarial network is a class of machine learning frameworks designed by Ian Goodfellow and his colleagues in 2014. Two neural networks contest with each other in a game. Given a training set, this technique learns to generate new data with the same statistics as the training set.

Autoencoders - Complex!
An autoencoder is a type of artificial neural network used to learn efficient data codings in an

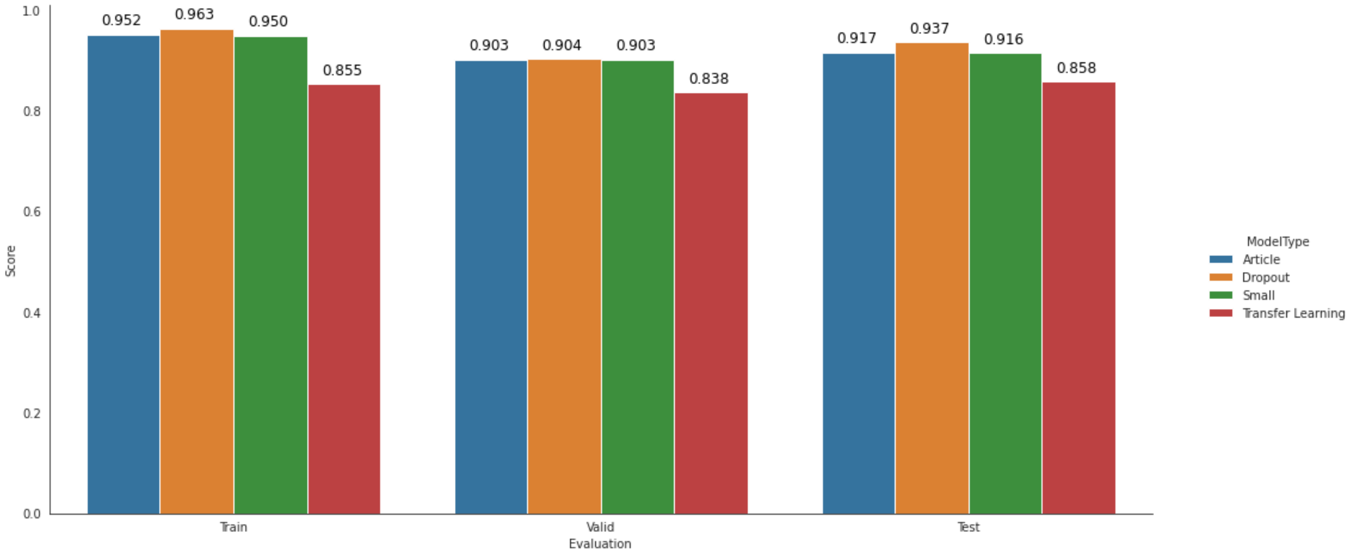
unsupervised manner. The aim of an autoencoder is to learn a representation for a set of data, typically for dimensionality reduction, by training the network to ignore signal “noise”.

Regarding memory complexity, it's about the same, and not spicial compare to any deep learning model.
Regarding inference - all 3 are almost identical. Althogh, Training is different.
CNN - Relatively Quick, Easy and fast to train, Most accurate.
AEs - Relatively Slow, due to 3 different training stages that need to be done. Medium performence.
GANs - Very Slow, Very sensitive to hyperparams, and got really bad performance due to "mode

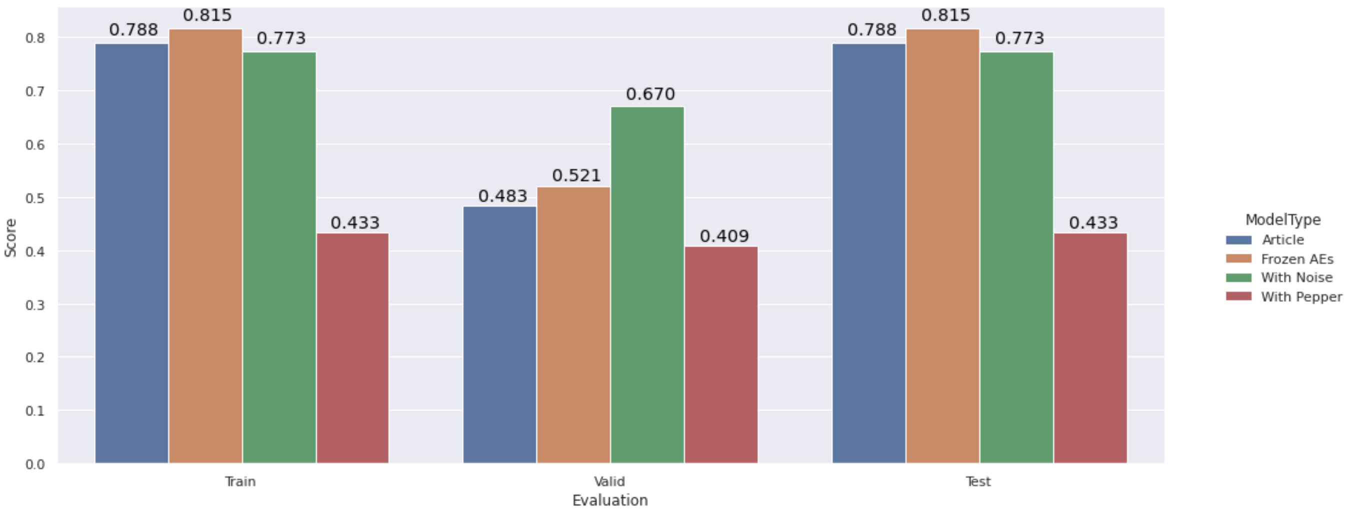
I. Analysis of the results of the technique that appeared in the article (mainly in terms of accuracy) (8 points)

a. What are the results of the technique on the set-data (4 points)

CNN:



AEs:

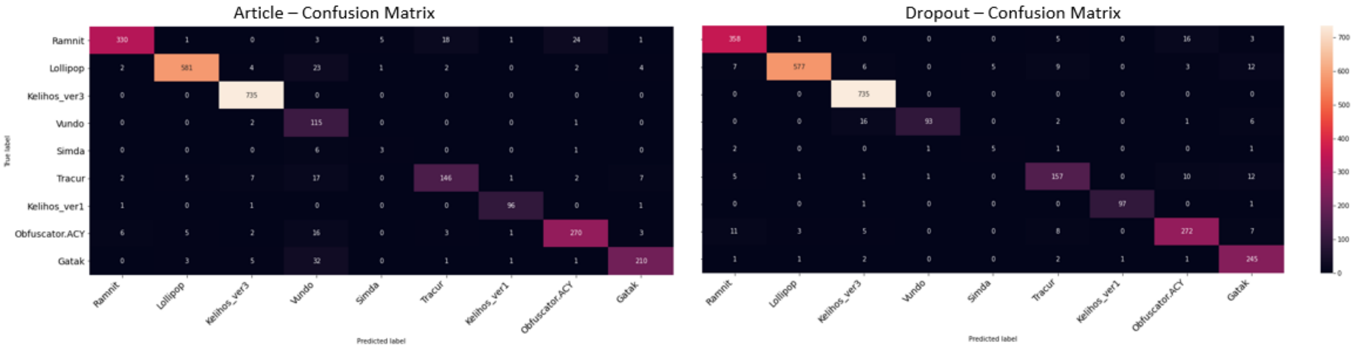


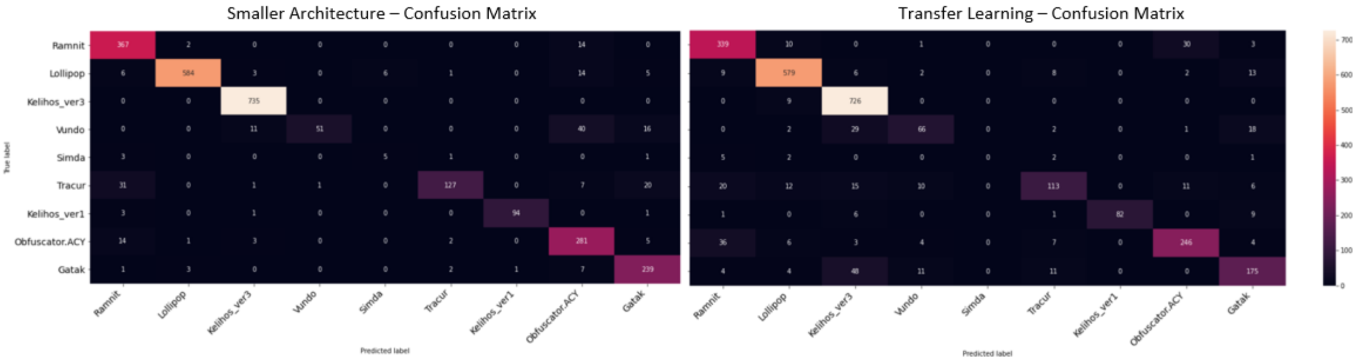
b. Try to explain the identification errors obtained (4 points)

As we can see from the results, the models did nice when dealing with the imbalanced data, specifically the CNNs. But the the inbalanced of the minority class is so high, that the models were not able to recognize it.

We believe the CNN performed better than the AE because it is a much deeper network, which gives its final layers a broader "view" of the input image, thus allowing a greater expressivity of complex relationships between the pixels.

If we had a domain expert we could use a "cost sensetive" approach, which gives a cost value to each of the malwares.
Also, Augmentation, sampeling and more data would have helped.

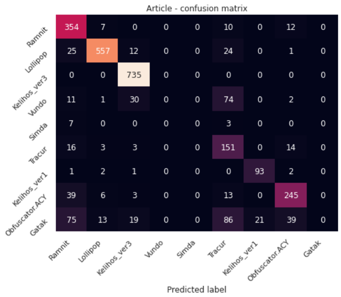




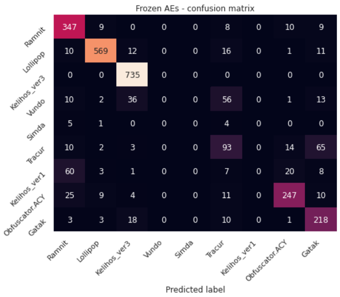
Smaller Architecture – Confusion Matrix

Transfer Learning – Confusion Matrix

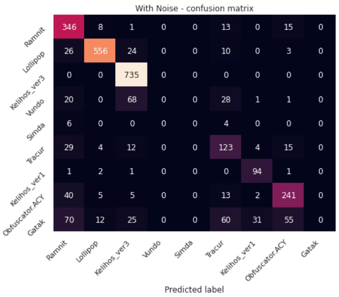
AEs:



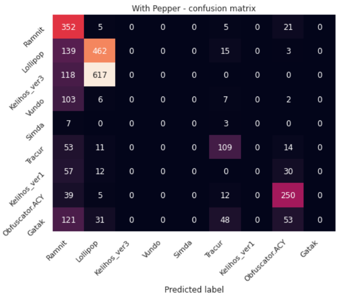
Article - confusion matrix



Frozen AEs - confusion matrix



With Noise - confusion matrix



With Pepper - confusion matrix

J. Apply on the set-data you have chosen the ML technique you propose (18 points)

a. Explain in detail your implementation (and attach the code) (14 points)

CNNs advances include - Dropout, Smaller layers, Transfer learning
AEs advances include - Frozen layer, Noise, Pepper

b. Analyze the complexity of the technique (4 points)

Almost the same complexity as the article seggested.

▾ K. Analysis of the results of the technique you proposed (mainly in terms of accuracy) (8 points)

▾ a. What are the results of the technique on the set-data (4 points)

Answer that above.

▾ b. Try to explain the identification errors obtained (4 points)

Answer that above.

▾ L. Comparing the results of the technique you proposed against the technique in the article (15 points)

▾ a. Compare the accuracy of the two techniques (5 points)

See bar graphs above

▾ b. Compare the mistakes of the two techniques (5 points)

Comparing the article's technique to the dropout technique, which gave the best accuracy, we can observe that the article's technique did a bit better on the two most common malwares while the dropout outperformed it in the rest of the malwares. The dropout model actually "allowed" more mistakes in the two major malwares. This means that the dropout model was able to better generalize the distribution of the data and not overfit to the two major classes as the article's technique did.

▾ c. Try to explain the differences in light of the nature of the techniques you tested (5 points)

Answer above.