THE AMERICAN UNIVERSITY IN CAIRO
الجامعة الأمريكية بالقاهرة

Malware Classification

Nada Badawy & Mariam Ahmed

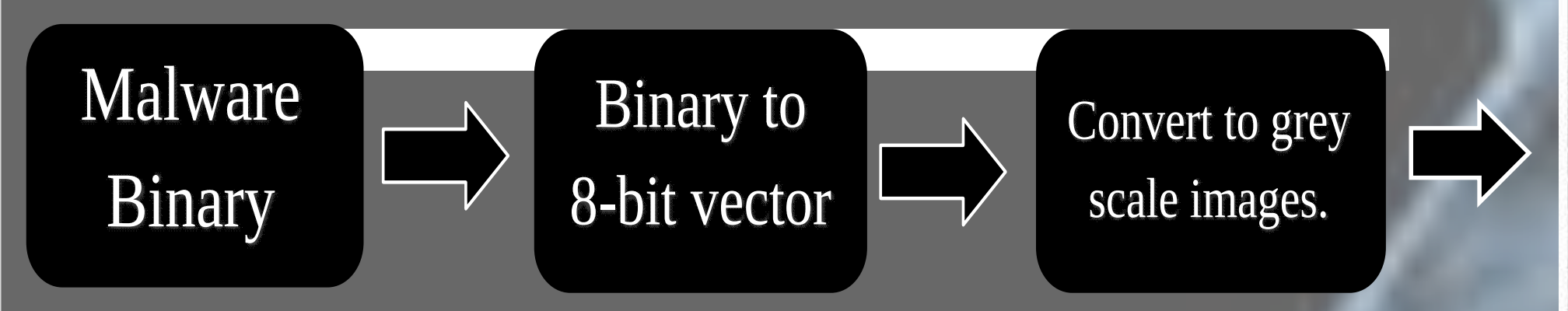
Key Words: pre-hydrolysis process, Kraft process , Causticizing process

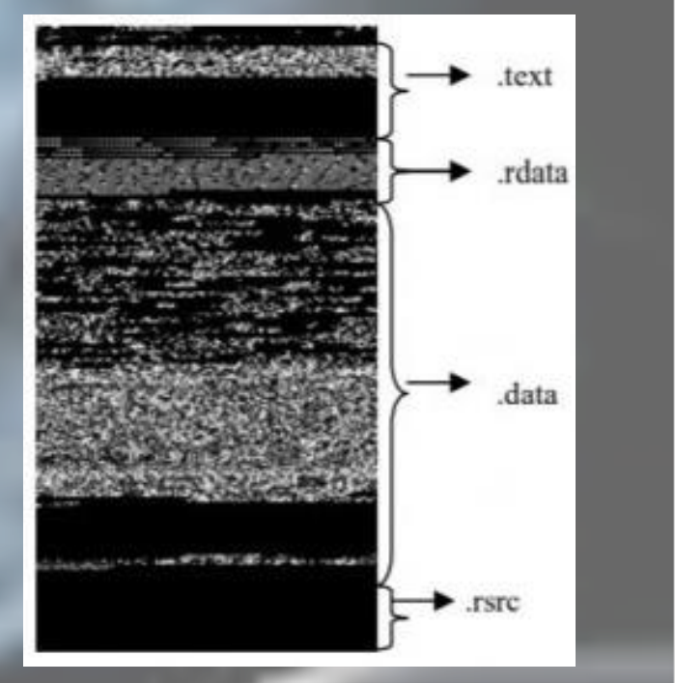
Introduction

Malware, which is short for malicious software, is a blanket term for viruses, trojans and other harmful computer programs. It is one of the most serious security threats on the Internet today. Nowadays, our using to the software is getting bigger, and most people puts their information on different websites, and they might have sensitive information on their e-mails. In fact, most Internet problems such as spam e-mails have malware as their main cause. Computers that are compromised with malware are often networked together to form botnets, and many attacks are launched using these malicious attacker-controlled networks. They exploit target system vulnerabilities, such as a bug in legitimate software; for example, a browser or web application plugin that can be hijacked. This might lead to data theft, extortion, or the crippling of network systems. Unfortunately, detecting the malware is not that trivial. Thus, one way to predict the new attacks is to be able to differentiate between the different Malware families. Predicting the right family enables us to efficiently deal with the threat. So, our goal is to train a deep learning model to classify between 25 different families of Malware.

Dataset

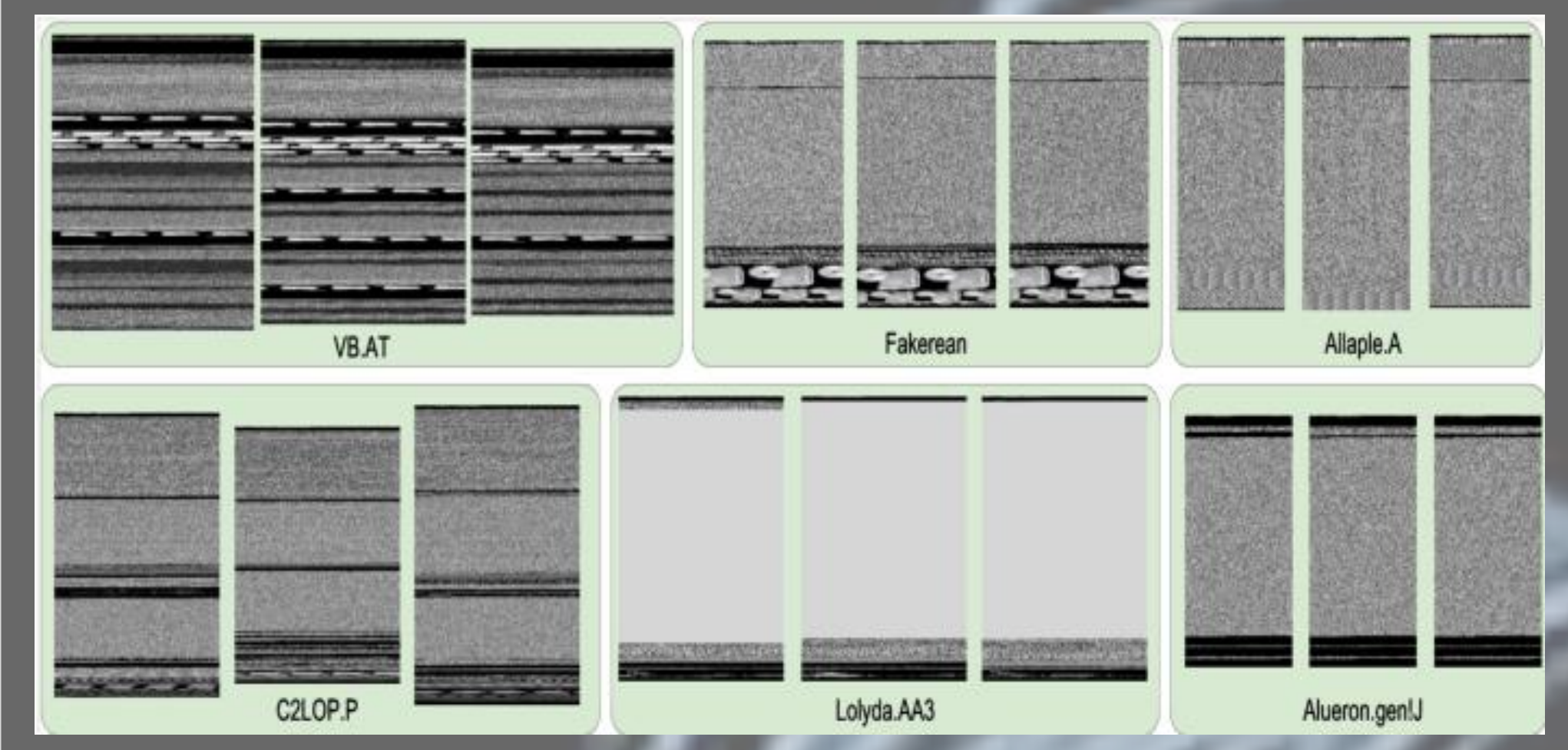
Generally, the malware dataset is in portable Executable (PE) form with file name extensions like .bin, .dll and .exe. The main components of the PE file are text, rdata, data, and rsrc parts. Malicious data binaries can be converted 8 bits at a time to pixels in a grayscale image, consisting of textural patterns.





Fig(1), shows the scenario to convert the PE files to greyscale images

For the sake of this project, we used the Malimg dataset [4] which is a set of grayscale images corresponding to malware binaries saved in .jpg format. In Figure (1), the sections of a malware binary can be shown as different textures patterns and accordingly we classify malware. The Malimg dataset consists of 9,339 malware samples from 25 different malware families. Figure (2) show examples from 6 different class from the Malimg dataset. It is obvious that each family has similar patterns which allow the model to recognize important patterns when extracting the features.



Fig(2), shows some Families from the Malimg dataset

Data pre-processing

To work with the dataset, we first process the data as follow:

1. **Resizing the images:**

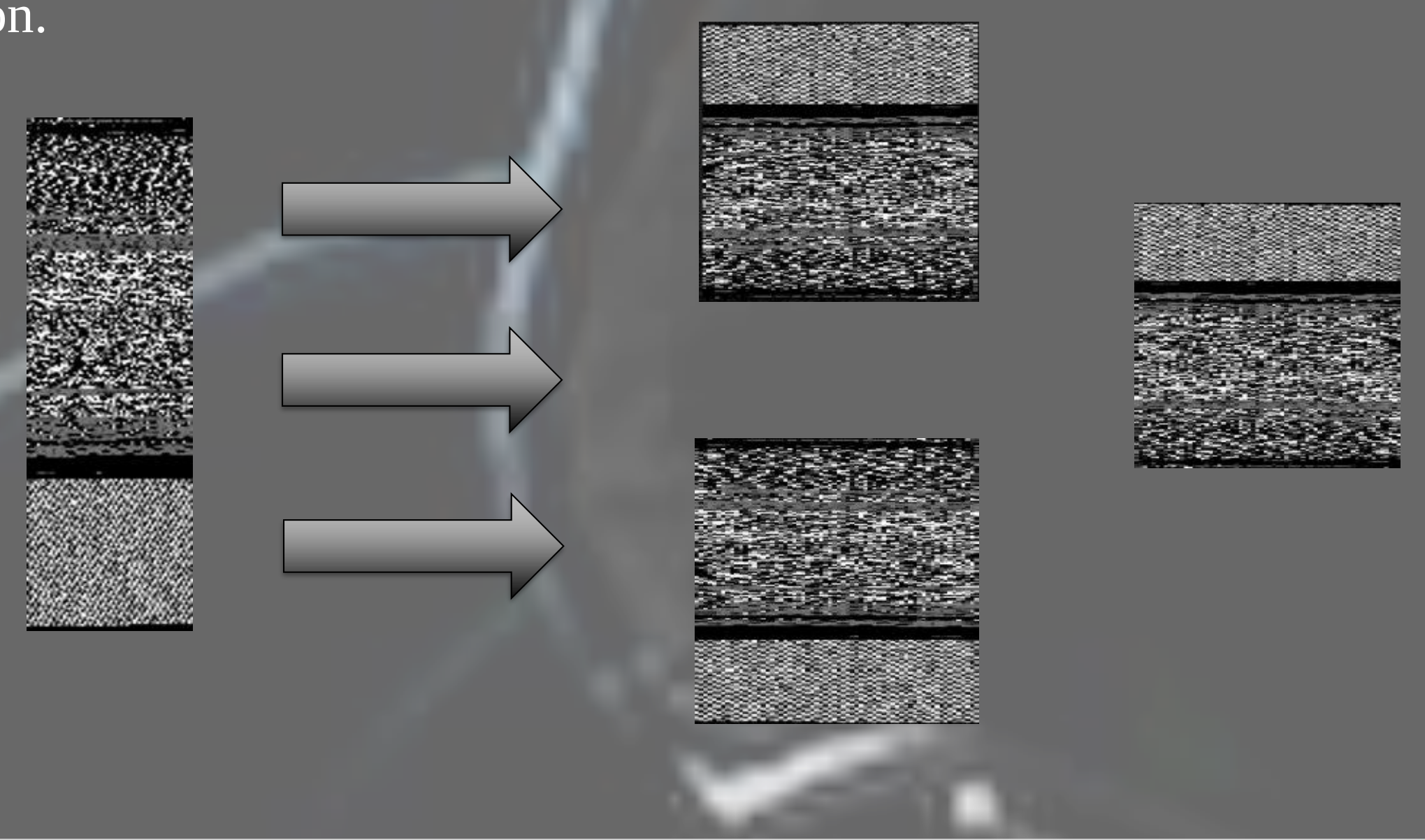
   The images of the Malimg data set varies in size. Thus, we resized all the images to have input shape of 224X224

2. **Data Scaling:**

   We make sure that the data was normalized in which all values are within the range of 0 and 1.

3. **Data augmentation:**

   The Malimg dataset was not balanced, some class dominates others. Thus, to balance the data, we did data augmentation (horizontal flipping, vertical flipping, and adding some noise). We balanced each class to have 600 images with 15000 in total for all classes, in which 70% was used in training the model and the other 30% was for the validation.



State-of-Art

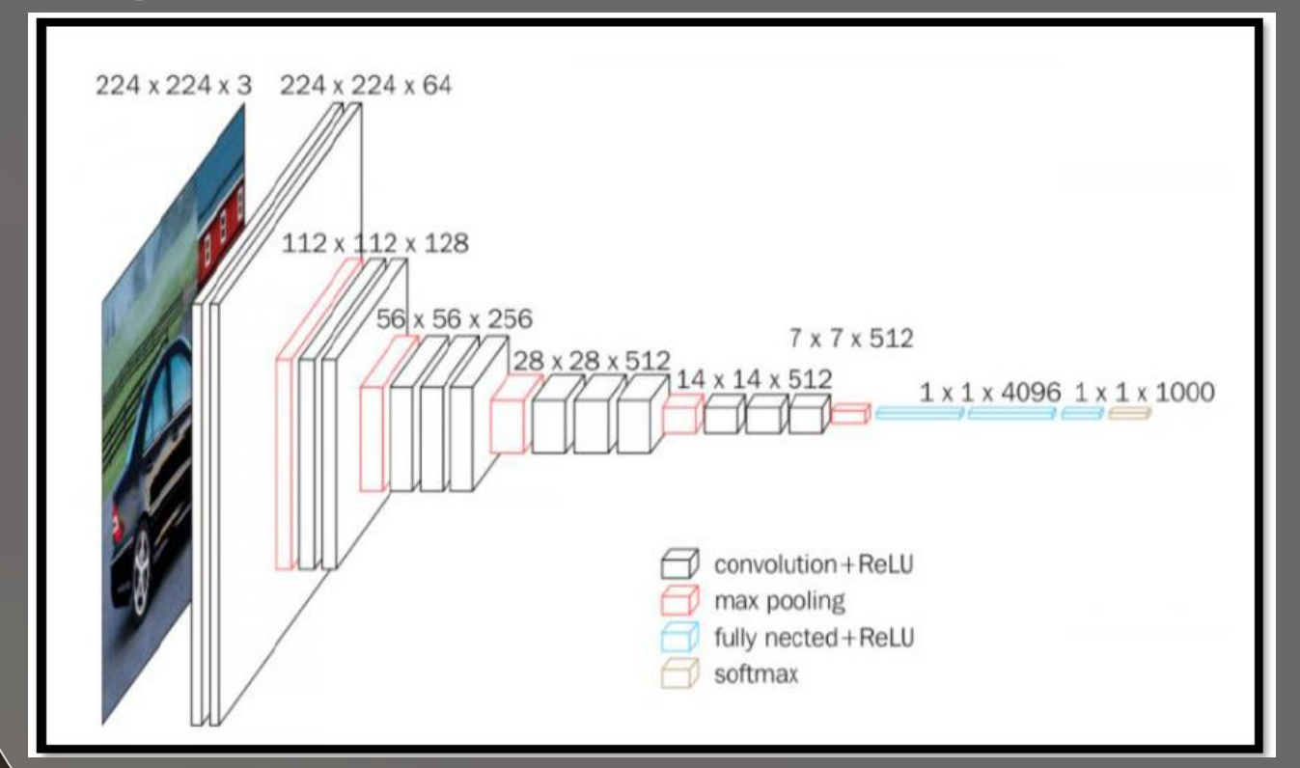
Solving malware classification problem has several deep learning methods. Based on the research we found three models that used for image classification.

First, CNN-SVM the convolutional neural network CNN usually uses the softmax activation function as top layer for prediction and cross-entorpy loss function. But, according to [1] they use linear support vector machine SVM instead of softmax activation function based on the advantage that they found in its usage. SVM objective is to find the optimal hyperplane f (w, x) = w · x + b to separate two classes in a given dataset. Hence, using CNN-SVM will enhance the accuracy of CNN model. This model reach accuracy of 77% using Malimg dataset.

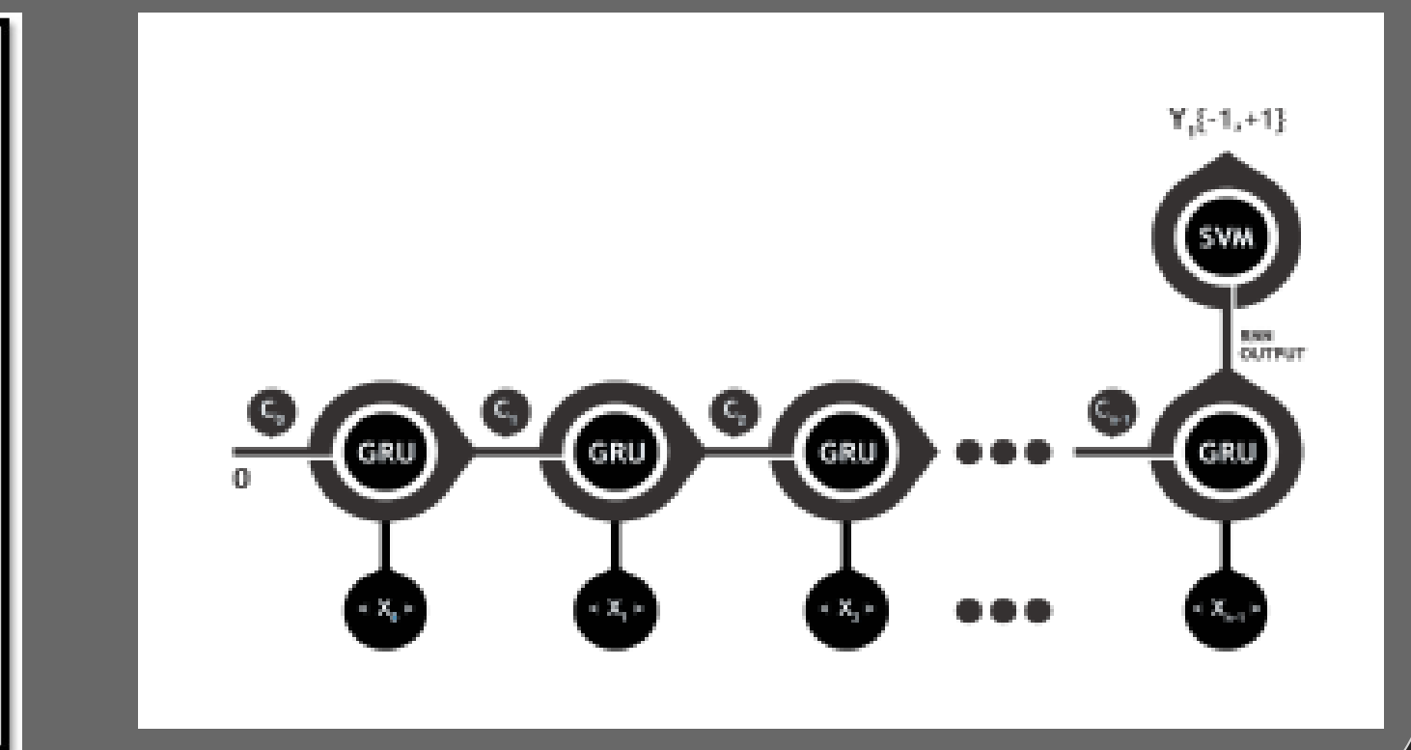
Second, GRU-SVM model. GRU is based on RNN model which is enhanced version of LSTM which using softmax as final output layer and cross-entroby function to compute the losses[2]. Yet GRU-SVM the parameters get learned through GRU and SVM works as classifier in a neural network architecture instead of softmax. This model reached 84% of accuracy using Malimg dataset.

Third, VGG16 model which is a version of CNN model. VGG16 contains 16 deep neural network layers including 3 by 3 convolution filters. 13 of the layers are convolutional layers each three followed by Maxpooling layer. Then two cully connected layers and the output layer. Based on [3] this model achieved 92% accuracy in ImageNet dataset which is a dataset of over 14 million images belonging to 1000 classes.

Fourth: Transformer is a model that uses attention to increase the model speed. Specifically, it uses self-attention. It contains six encoder and six decoders. Each encoder has two layers: self-attention and a feed Forward Neural Network. Also, the decoder has the same layers but between then there is an attention layer. This attention layer helps the decoder focusing on the relevant parts of the input image. Transformers have advantage over the RNN because it solves the problem of parallelization, by utilizing CNN together with attention model. This model archive accuracy of 88.55% in ImageNet dataset.



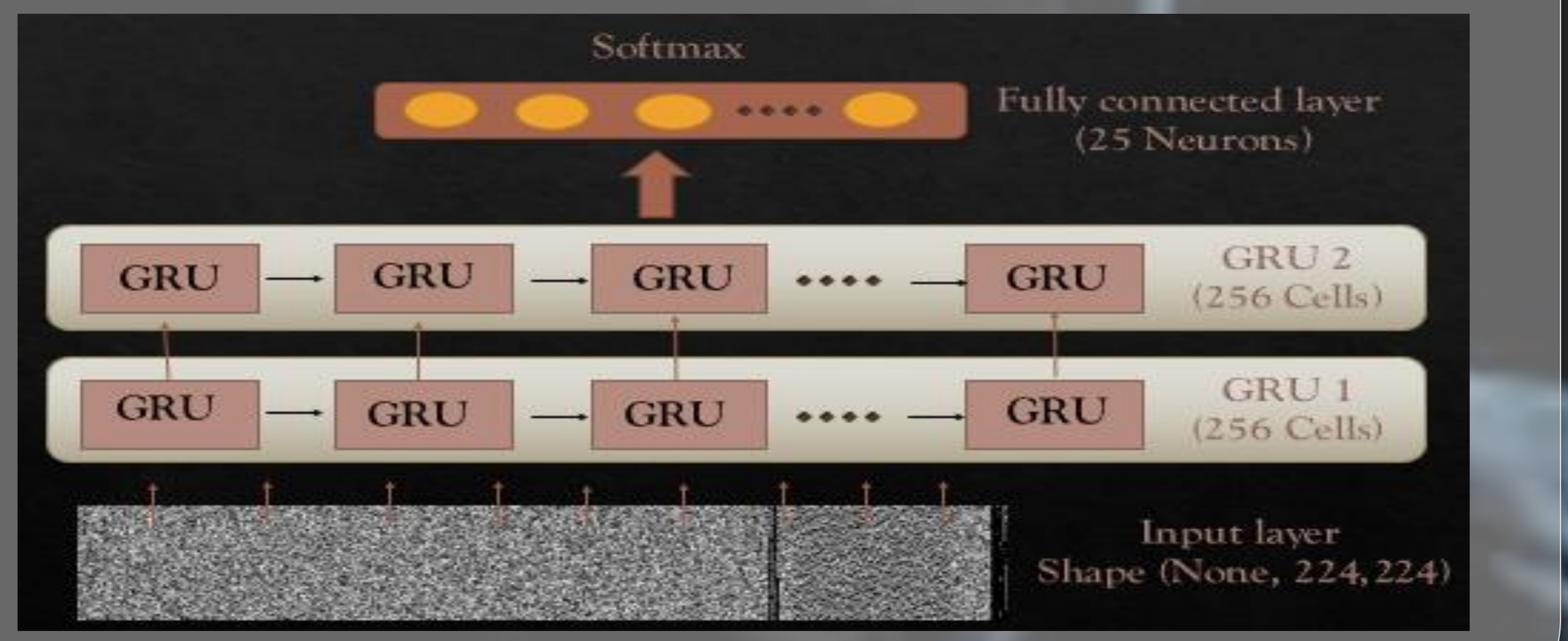
Architecture of VGG16



Architecture of GRU-SVM

Model Description

In our final model we decided to use deep GRU model with two layers in which 256 units. The model takes input image of size (224,224) then it will go through the tow GRU layers to train and finally get to the FC layer with softmax activation finction with output 25 neurons corresponding to 25 classes of our dataset Malimg.


Softmax
Fully connected layer (25 Neurons)
GRU
GRU
GRU
GRU
GRU 2 (256 Cells)
GRU
GRU
GRU
GRU
GRU 1 (256 Cells)
Input layer Shape (None, 224,224)

Final model architecture

Result Analysis

We performed a variety of experiments, here we present a summary of the results for 6 separate experiments, as listed in Table [1]. In the remainder of this section, we discuss the results of our final model in more details .

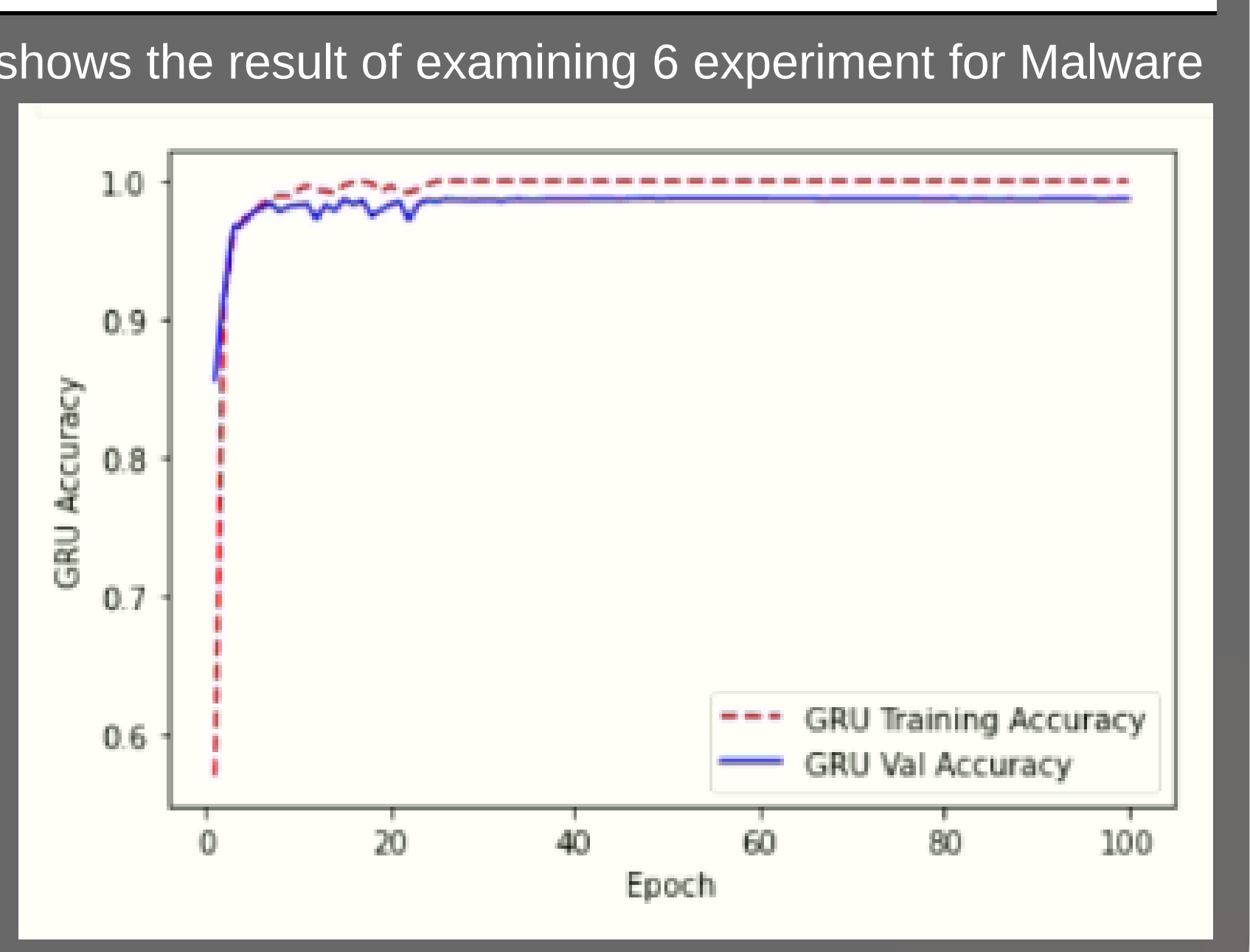
Model name	VGG16-GRU-SVM	GRU-SVM	VGG16	VGG16-GRU	Deep GRU	Transformer
# of training data	11130 samples	11130 samples	11130 samples	11130 samples	11130 samples	11130 samples
Epochs	45	100	45	45	100	100
Batch Size	32	64	32	32	64	64
Dropout Rate	0.2	0.2	0.2	0.2	NA	0.1
Learning Rate	0.01	0.01	0.01	0.01	0.001	0.001
Total Parameters	15,3193,85	786,713	14,844,305	15,3193,85	786,713	21,759,019
Training Time						
Validation accuracy		4%	66%	54%	98.7%	91.68%

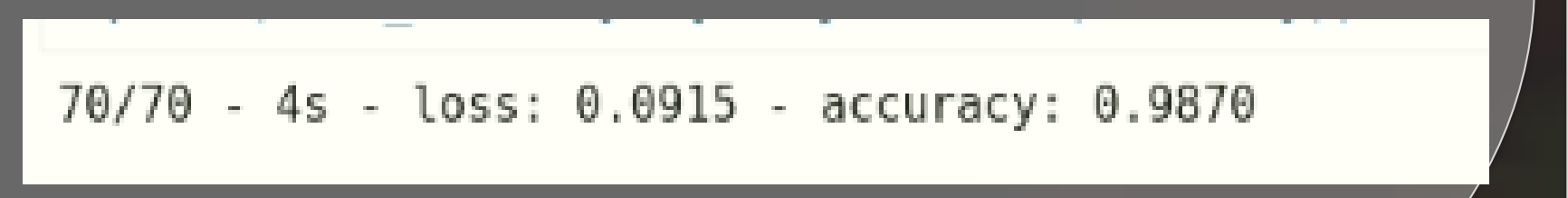
Table [1] that shows the result of examining 6 experiment for Malware Classification

Deep GRU:

We trained our final model for 100 epochs over 11130 Malware grayscale image and tested it with 4470 samples. The model was trained to classify 25 different class of malware.

The validation accuracy of the model reached 98.7% and the validation loss of 0.0915.

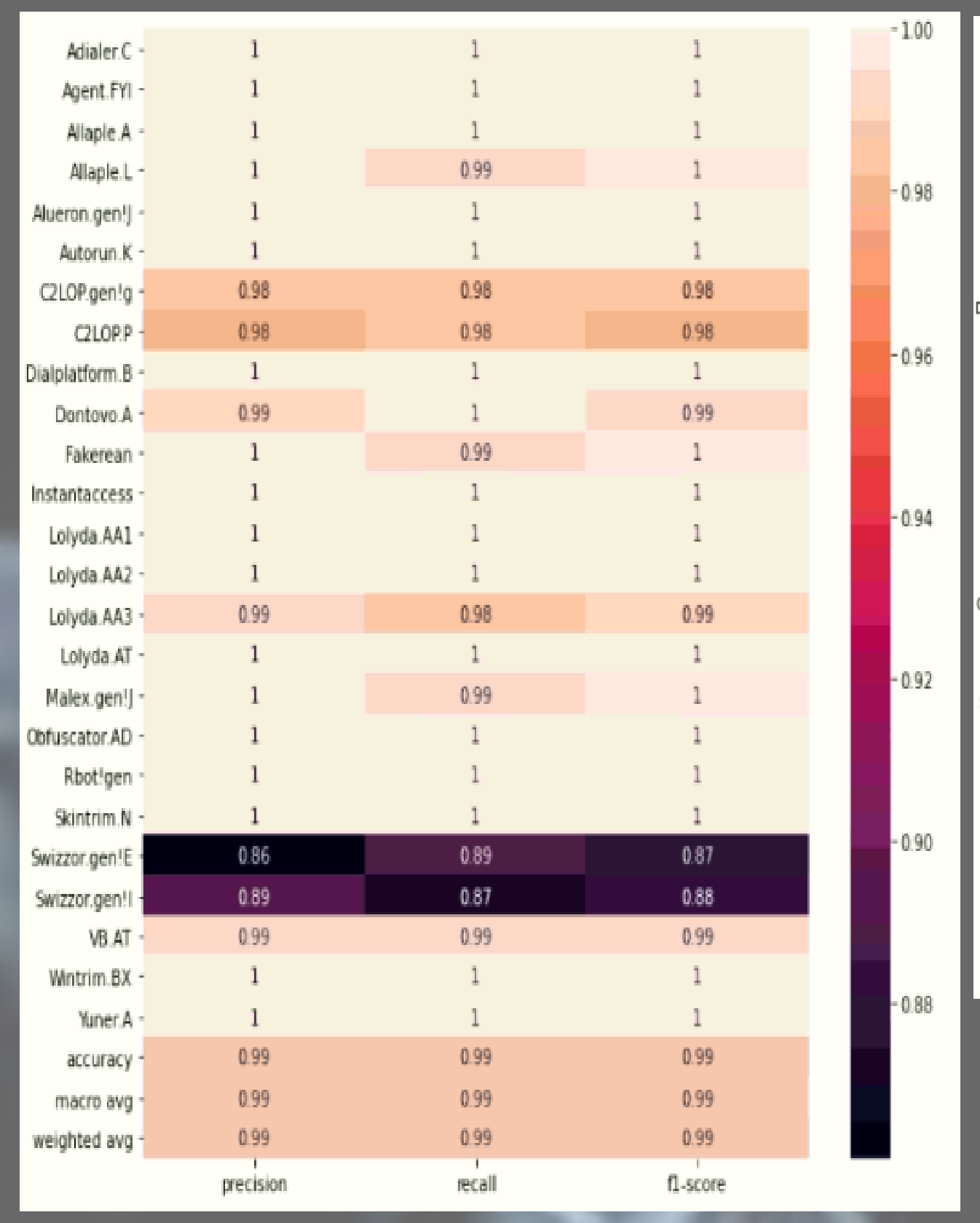




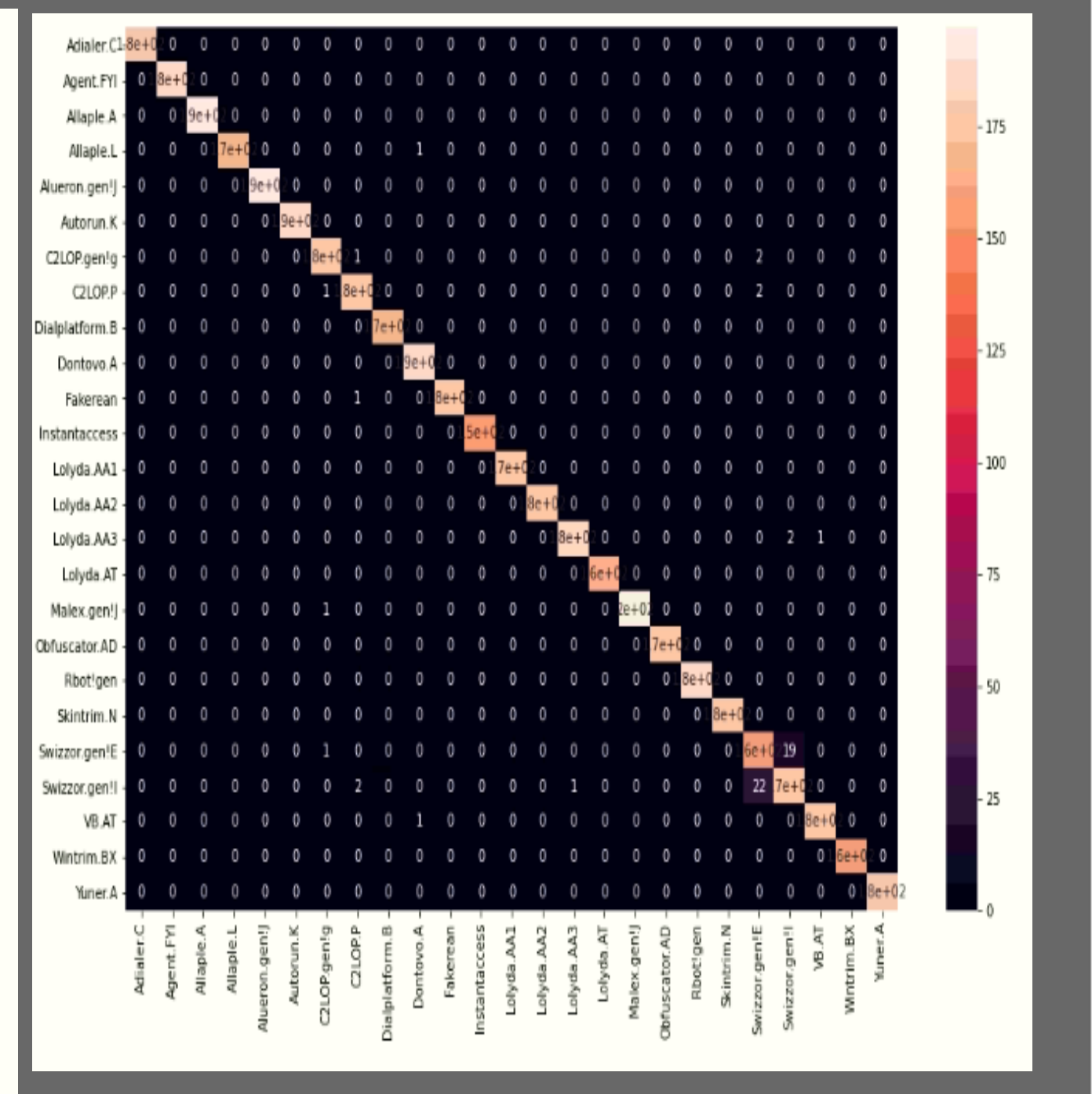
```
70/70 - 4s - loss: 0.0915 - accuracy: 0.9870
```

Representation of the training accuracy virus validation accuracy for Deep GRU model

moreover, the Confusion Matrix shows a good predictive accuracy for each malware family. However, Swizzor.gen!E and Swizzor.gen!I shows relatively low presession, recall and f1 score but in overall, all families shows high low precession, recall and f1 score.



Fig(3) shows precession, recall, and f1-score for each Family.



Fig(4) Plotted using matplotlib. Confusion Matrix for deep GRU testing results, showing its predictive accuracy for each malware family

Conclusions and Recommendations

We used the Malimg dataset prepared by [4], which consists of malware images for the purpose of malware family classification. We trained 6 models on 11130 Malware grayscale image and tested them with 4470 samples. The empirical data shows that the deep GRU model had the highest predictive accuracy among VGG16-GRU-SVM, GRU-SVM, VGG16, VGG16-GRU, and the transformer, having a test accuracy of 98.7%.

Improving the architecture design of the transformer model by adding more layers, adding better nonlinearities, and/or using an optimized dropout, may provide better insights on their application on malware classification. Such insights may reveal an information as to which architecture may serve best in the engineering of an intelligent anti-malware system.

Literature cited

[1] Agarap, A. (2019, February 7). An Architecture Combining Convolutional Neural Network (CNN) and Support Vector Machine (SVM) for Image. doi: https://arxiv.org/pdf/1712.03541v2.pdf

[2] Agarap, A. (2019, September 10). A Neural Network Architecture Combining Gated Recurrent Unit (GRU) and Support Vector Machine (SVM) for Intrusion Detection in Network Traffic Data. Retrieved from: https://paperswithcode.com/paper/a-neural-network-architecture-combining-gated#code

[3] Gopalakrishnan, Kasthurirangan & Khaitan, S.K. & Choudhary, Alok & Agrawal, Ankit. (2017). Deep Convolutional Neural Networks with transfer learning for computer vision-based data-driven pavement distress detection. Construction and Building Materials. 157. 322-330. 10.1016/j.conbuildmat.2017.09.110.

[4] G Giacaglia, G. (2020, October 5). How Transformers Work - Towards Data Science. Medium. https://towardsdatascience.com/transformers-141e32e69591 iacaglia, G. (2020, October 5). How Transformers Work - Towards Data Science. Medium. https://towardsdatascience.com/transformers-141e32e69591

[5] https://sarvamblog.blogspot.com/2014/08/supervised-classification-with-k-fold.html