# National University of Computer & Emerging Sciences
# Karachi Campus


## Project Report
## Information Security
## Section: 7G


## Group Members:
## 19k-0136 Arzaan Ul Mairaj
## 19k-0325 Subhan Saleem
## 19k-0368 Wyle Mustafa

**Abstract**

The abstract should summarize the problem addressed, methodology, results, conclusions, and contributions.

The project is based on the idea of aiding in the detection of anomalies in networks, on the basis of the growing importance of network security within corporations. The project aims to detect a network intrusion. With the use of efficient Deep Learning and Machine Learning techniques, through models like Stochastic Gradient Descent (SGD) and Support Vector Machines (SVMs), it is possible to train a system that effectively detects all such intrusions. In the end, we found out that SGD performs best at detecting these malwares, in which malware such as Agent.FYI, Autorun.K, DialPlatform.B and Lolyda were the easiest to detect and Swizzor family of malware was the hardest to detect.

## I. Introduction (10%)

The project is based on the idea of aiding in the detection of anomalies in networks, on the basis of the growing importance of network security within corporations. With the number, severity, and criticality of professionally deployed attacks on corporate networks increasing, and new threats emerging every passing day, a successfully deployed network attack can cause huge financial losses.

Like any executable file, a malware executable is addressed as a series of zeros and ones. A string is likewise a vector of hexadecimal values; in that capacity, it may be reshaped into a matrix and seen as a picture. Once the malware is changed over into grayscale images, malware discovery can be decreased to an image classification issue.

Different models were tested with rigorous training datasets of each malware family from the dataset “Malimg’ dataset from Kaggle, and then the best-performing model was tuned.

In **[1]**, the authors proposed a method for visualizing and classifying malware using image processing techniques. Malware binaries were visualized as gray-scale images. Images of different malware families appear visually similar and distinct from those of different families. Based on this observation, a classification method using image texture analysis was proposed. Features were extracted from malware images to be used for classification via the K-Nearest Neighbor technique.

## II. Existing System (7%)

This section should describe the existing system, in which you will implement your subsystem.

1. The main components of the existing system and the technologies used for implementing the system components. Make sure to use diagrams and figures to illustrate the main components of the system and their interconnections.
2. The data flow between the components,

3. The ways various stakeholders interact with the system.
4. The adversary model the existing system is intended to resist.

**III. Related Work (5%)**

In **[2]**, the research aimed to detect encrypted malicious content which is commonly distributed through the internet. The paper proposes $D^2PI$, a novel way of identifying network traffic with malware by performing deep packet inspection with a Convolutional Neural Network. $D^2PI$ is a neural network architecture that uses character embeddings followed by deep convolutional networks trained upon the payloads of packets from the data set and functions as an NIDS. In an evaluation that uses a dataset of 127 distinct malwares and a sampling of over 16GB of benign traffic, $D^2PI$ outperformed the popular open source intrusion detection system Snort by more than 17% in F1 score.

In **[3]**, the research aimed to detect DoS of attacks with the justification that cyberattacks are becoming more intelligent and attackers bypass known signatures and pretend to be normal users. A DL-based intrusion detection model, Convolutional Neural Network (CNN), was developed with the focus on denial-of-service (DoS) attacks. Two different data sets were used in order to compare the results and train the models. The performance was evaluated through the comparison with a Recurrent Neural Network (RNN). The experimental results have shown that RGB images in both binary and multi-class classifications have higher accuracy than that of gray-scale images. The comparison of the proposed model with RNN verified that for KDD, CNN model showed 99% or more results in binary and multi-class classifications, the RNN showed 99% accuracy in binary classification and 93% in multi-class classifications. For CSE-CIC-IDS 2018, the CNN model showed 91.5% of accuracy on average while the RNN model showed 65% of accuracy on average. In other words, the CNN model proposed in this paper was able to identify specific DoS attacks with similar characteristics compared to the RNN model.

**IV. Adversary Model (5%)**

The adversary model for my (sub) system is exactly the same as for the existing system.

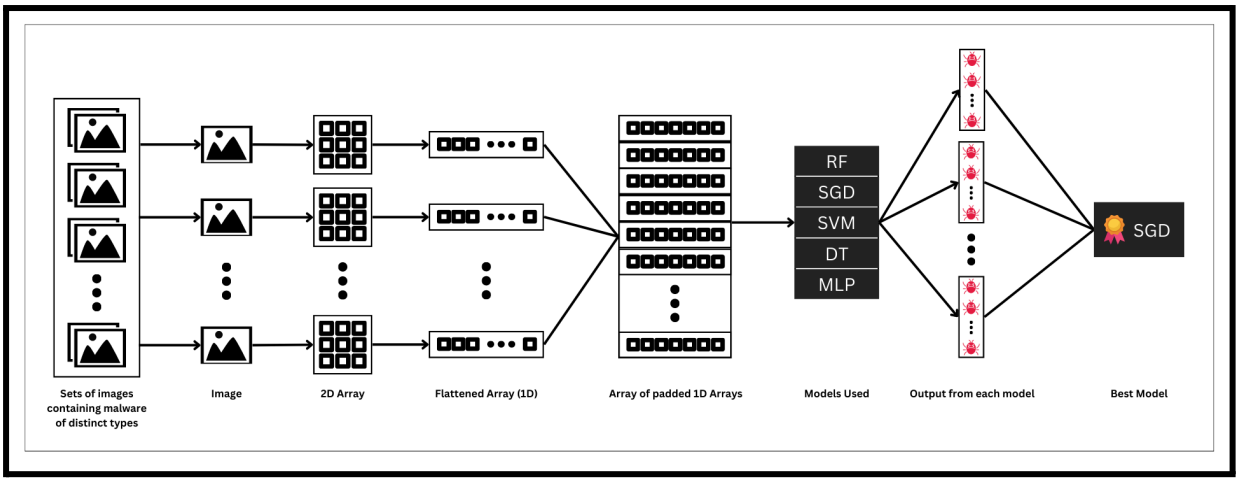
**V. System Design (20%)**

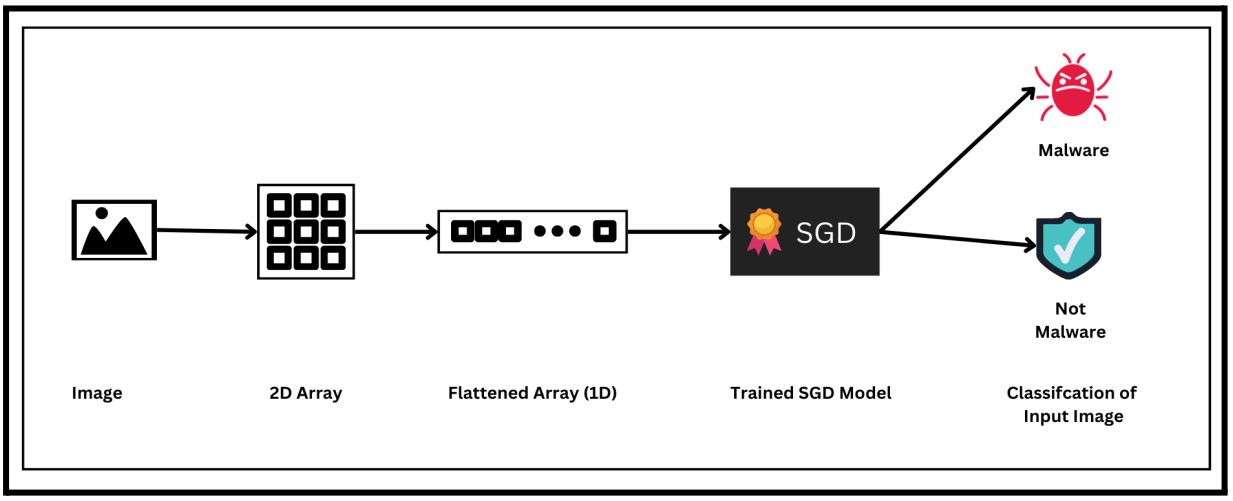
The images would first be reduced to a one-dimensional image and then be scaled equally in size. We would first test multiple models for our malware classification and decide the best one. After that, some model tuning would be required, and then it would be used to determine if an image passed over is malware or not.

**VI. System Implementation (20%)**

During model development, after converting each image to a one-dimensional array, we put to test 5 models: Random Forest, SGD, SVM, Decision Tree, and MLP. The best model from these was SGD. We tuned SGD for better accuracy.

We then decided to use our model for our system. If we pass an image to our model, it must detect if it is malware or not, and if it is then it will identify it. I decided to do this by keeping a threshold, if the class probabilities predicted by the model were all lower than 0.1, then it would be identified as not malware, otherwise, it would predict the malware family.

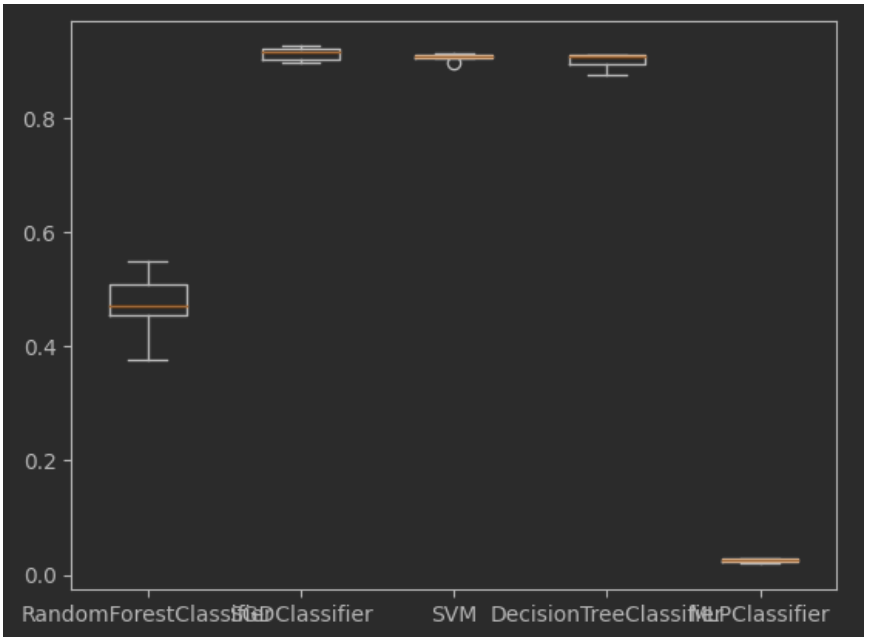



## VII. System Evaluation

The best model was first picked according to best accuracy and after some tuning, a classification report, and a confusion matrix were used.

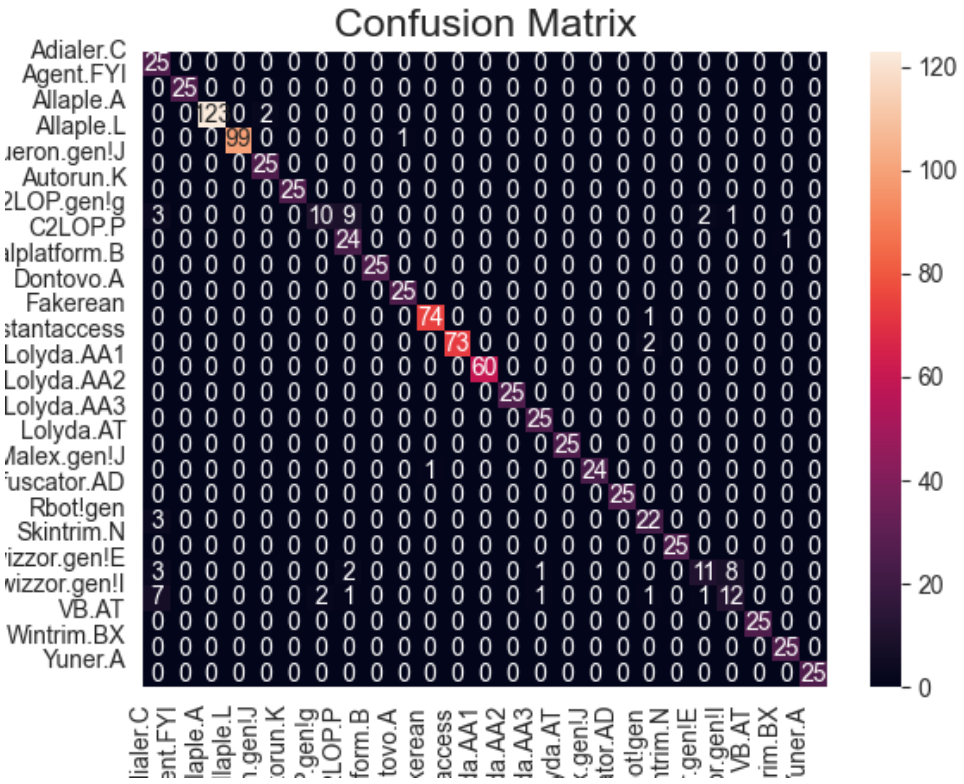
We chose SGD as it has the best accuracy.



Then, after some tuning it provided a 94% accuracy.

| | | | | |
|---|---|---|---|---|
| accuracy | | | 0.94 | 935 |
| macro avg | 0.92 | 0.92 | 0.91 | 935 |
| weighted avg | 0.95 | 0.94 | 0.94 | 935 |

The confusion matrix is as follows:




This shows that the Swizzor family was the hardest to detect.


**VII. Discussion (10%)**

Discuss pros and cons of your implementation, as well as limitations and advantages of it. This discussion should integrate the related work, the motivation for the implementation, the adversary model, the ideas of the implementation, and the results of the evaluation. This section should be about one page.

Also, report here how your implementation has been adopted by the developers of the existing system.

**VIII. Conclusion (5%)**

This section should summarize the report in 1-2 paragraphs. Although a conclusion may review the main points of the report, do not replicate the abstract as the conclusion. A conclusion might elaborate on the importance of the work or suggest

applications and extensions.

**References**

**[1]** L. Nataraj, S. Karthikeyan, G. Jacob, and B. S. Manjunath, “Malware images,” Proceedings of the 8th International Symposium on Visualization for Cyber Security - VizSec ’11, 2011, doi: 10.1145/2016904.2016908.

**[2]** Cheng, Ronald_1801.pdf (umd.edu)

**[3]** J. Kim, J. Kim, H. Kim, M. Shim, and E. Choi, “CNN-Based Network Intrusion Detection against Denial-of-Service Attacks,” *Electronics*, vol. 9, no. 6, p. 916, Jun. 2020, doi: 10.3390/electronics9060916.