

Malware detection using an ensemble of vision models & LLMs

Afif Ali Saadman
Dhaka, Bangladesh
0009-0000-7719-5094

Abstract—Nowdays, cyber attacks using software are increasing rapidly. Cyber criminals are finding out new ways to make malware and infect networks. Signature based detection is failing. But, image-based analysis & disassembly show signs of new hope. This paper presents an ensemble-based concept which can be used to produce highly-effective security software.

Index Terms—Vision Transformer, Malware, Feature Extraction, Computer Virus, Large Language Model, Ensemble, TF-IDF, XGBoost, LGBBoost, AlexNet, ViT, ResNet, LLaMa, MobileNet , GPT

I. INTRODUCTION

Signature-based malware detection is slowly failing to protect users. Criminals are constantly developing new variants of harmful software which impact negatively in a person's life. With generative AI being "jail-broken" by hackers, they can easily make malware very quickly. Often avoiding the signature-based approach completely. To address the limitations of traditional signature-based methods, image based malware detection methods have been extensively studied since their start in 2011. These things works by converting executable files(exe) to an image representation, which the model identifies with patterns it had seen before. But, current implementation takes too much training time preparing the dataset alone. Some small mistakes result the model performing poorly. To fix these limitations, I have created a concept which can outperform current implementations completely and solve current issues. This model is train on some of the destructive malwares in Microsoft Windows to identify malware more efficiently (even new variants in the test-set)

II. CURRENT IMPLEMENTATIONS & LIMITATIONS

In [1], the generated image is often filled with black padding. Often resulting the image to be 90% black. Resulting in the model to face difficulties. In [2], the authors did a dangerous approach to **directly resizes** the image to fit the model's expectation, which destroys the data plotted in the picture. Resulting in the model to not to be able to match patterns.

III. PROPOSED CONCEPT

This paper proposes an ensemble based approach, this approach consists of the following pipeline The "bin2img.py" utility (can be found at [3]) generates images with the size of 384x384(needs to be defined by the square parameter) , which is fed to the model. And for the LLMs, the "disasm.py"

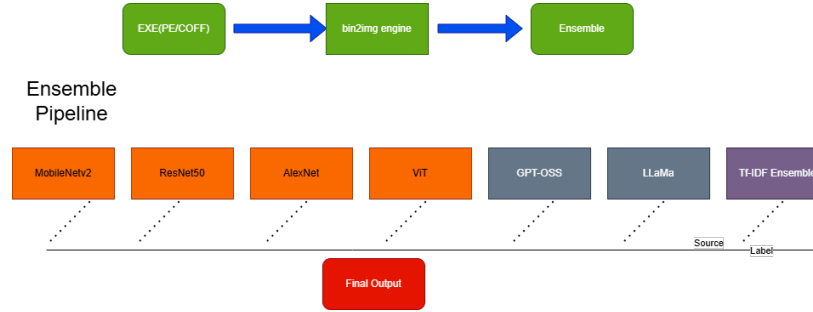


Fig. 1. The Pipeline

utility disassembles the executable and feeds it to the LLM & the TF-IDF ensemble. Then, the final result is decided using the uniform soft-voting method, which averages the model's probabilities. Resulting in an accurate prediction. The image size is chosen due to limited hardware (Only 2 NVIDIA T4 GPU) and small file sizes, which is enough to contain all data the model needs.

IV. THE CUSTOMIZED UTILITIES

The bin2img utility supports both binary-to-image and image-to-binary conversion. It extracts the raw bytes and stores them in a NumPY array. Then, the array is plotted to the image using the tiling method. After plotting, if there are any unnecessary black bytes(NOT THE PROGRAM'S ONES), the utility fills them up with the data that is important to the model for predictions. The disasm utility uses "capstone" library to disassemble the binary into assembly.

V. THE DATASET

The dataset consists of a total of 156 executable files, including 78 dangerous samples and 78 safe samples. The dataset is balanced across both classes to avoid bias during training and evaluation. Due to hardware constraints and limited computational resources, the dataset size is relatively small. Future work will focus on scaling the dataset significantly. This is a proof-of-concept due to dataset constraints. The results are not production grade.

VI. MODEL 1: MOBILENET

MobileNetv2 [4], published by Google, is a lightweight CNN architecture. This model outperforms all other models



Fig. 2. MobileNet Training Plot



Fig. 4. AlexNet Training Plot



Fig. 3. ResNet Training Plot

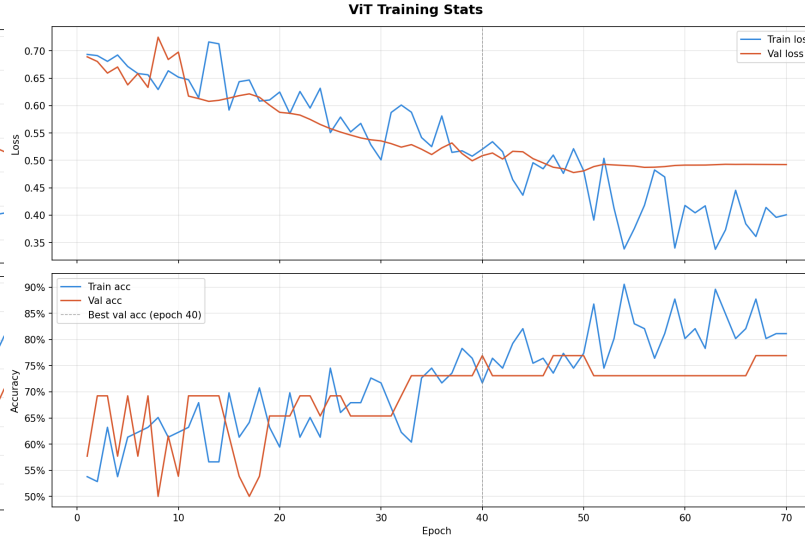


Fig. 5. ViT Training Plot

in the ensemble. This model achieved approximately 83% accuracy in the unseen test-set.

The training instability was caused by the learning rate scheduler disruption, but, it avoided overfitting.

VII. MODEL 2: RESNET50

ResNet [5], developed by Microsoft, is a revolutionary model surpassing human accuracy. This Model is in ties with ViT, maybe for aggressive overfitting. Resulting in a modest 70% accuracy.

VIII. MODEL 3: ALEXNET

This revolutionary CNN [6] have achieved 75% accuracy. Taking 2nd place.

IX. MODEL 4: ViT

Vision Transformer Models are fascinating [7] . But, it needed much more data to perform well. It achieved 70% accuracy.

X. TEXT MODELS & LLMs

After training & hours of fine-tuning using disassembled raw-assembly code(NASM syntax), all of the text models achieved 50% due to random guessing. Even with LLMs, they can code but can do it based on patterns. But, the patterns in the disassembled code are not.

A. Why still include LLMs?

Despite their poor individual performance, LLMs contribute to the uniform soft-voting ensemble by providing a text-based

perspective on assembly semantics. In future work, with fine-tuning on extended malware assembly dataset, they have the potential to significantly improve.

XI. ENSEMBLE METRICS

The ensemble uses a **uniform soft-voting** strategy, which averages the predicted probabilities across all models and after that, it makes a final decision. As shown in Table I, the ensemble achieved 75% accuracy and F1-score, with a notably strong AUC-ROC of 86.8%, indicating good discriminative ability between malware and safe samples despite the modest accuracy.

TABLE I
ENSEMBLE RESULTS

Class	Precision	Recall	F1	Support
Danger	0.75	0.75	0.75	12
Safe	0.75	0.75	0.75	12
Overall	0.75	0.75	0.75	24

As we can notice, the ensemble did not surpass MobileNetV2's individual accuracy of 83%, which suggests that the text-based models (TF-IDF/XGB+LGB, GPT-OSS-20B, LLaMA-3.1-8B) dragged the ensemble down due to their random-guess-level performance. A **weighted soft-voting** approach, assigning higher weights to the stronger vision models, is expected to significantly improve ensemble performance in future work.

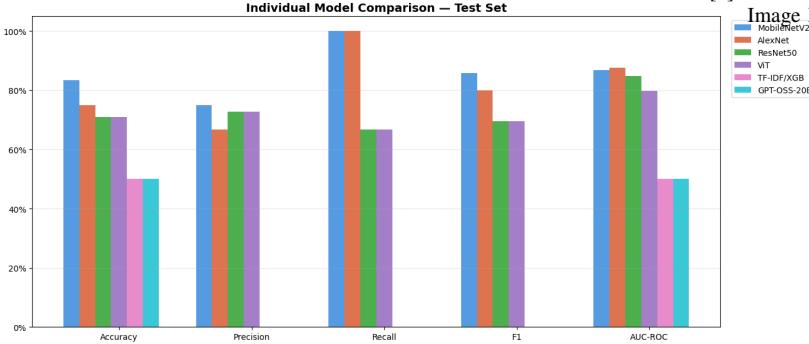


Fig. 6. Individual Model Comparison on Test Set

XII. KEY FINDINGS

- **MobileNetv2** achieved the best individual performance with approx. 83% accuracy and a perfect recall of 1.0, making it the most reliable model for malware detection in this ensemble.
- **AlexNet** secured 2nd place with 75% accuracy, also achieving perfect recall, but lagging behind MobileNetv2 in precision.
- **ResNet50 and ViT** tied at 70% accuracy. ResNet50 suffered from aggressive overfitting, while ViT was limited by the small dataset size, as transformers are data hungry.
- **Text-based models** (TF-IDF/XGB+LGB, GPT-OSS-20B, LLaMA-3.1-8B) all fell to 50% accuracy due to random-

guessing and noise of raw assembly code being outside their learned pattern space.

- The **Uniform soft-voting ensemble** achieved 75% accuracy with an AUC-ROC of 86.8%, suggesting strong probability calibration despite being dragged down by the over-fitted text models.
- **Image-based detection** consistently outperformed text-based approaches, confirming that visual byte patterns are more accurate.

XIII. CONCLUSION

This paper proposed a concept based on ensemble-based malware detection system combining vision models and LLMs, trained on a balanced dataset of 156 Microsoft Windows exe files. MobileNetv2 dominated with 83% accuracy, confirming that image-based detection captures byte-level patterns far more effectively than assembly-based text models. Future work will scale the dataset, adopt weighted soft voting, and fine-tune LLMs on more data close the performance gap.

REFERENCES

- [1] S. Seneviratne et al, Self-Supervised Vision Transformers for Malware Detection
- [2] Han-Ju Lee et al, MalFormer: A Novel Vision Transformer Model for Robust Malware Analysis
- [3] Afif Ali Saadman, <https://github.com/afifafifafifafifali/malware-research>
- [4] Howard et al, MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications
- [5] Kaiming He et al, Deep Residual Learning for Image Recognition
- [6] Alex Krizhevsky et al, ImageNet Classification with Deep Convolutional Neural Networks
- [7] Neil Houlsby et al, An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale