Malaware Detection based on Binary Visualization

InceptionV3 and Resnet50 Models

Omkar Vijay Bhatkande
College of Engineering and Computer Science University of Michigan - Dearborn
Dearborn, Michigan
omkarb@umich.edu

Emma Loveland
College of Engineering and Computer Science University of Michigan - Dearborn
Dearborn, Michigan
lovemma@umich.edu

Allycia Lemme
College of Engineering and Computer Science University of Michigan - Dearborn
Dearborn, Michigan
allemme@umich.edu

Marissa Banks
College of Engineering and Computer Science University of Michigan - Dearborn
Dearborn, Michigan
bmariss@umich.edu

Theodore Q. Sullivan
College of Engineering and Computer Science University of Michigan - Dearborn
Dearborn, Michigan
theosull@umich.edu

ABSTRACT

This paper aims to leverage various architectural elements to effectively classify files as malware or benign using Convolutional Neural Network (CNN) models. The approach requires files to be processed through a tool that transforms their binary and ASCII values into images. The architecture of a CNN will be explored, optimizing its design and hyperparameters to enhance the accuracy and robustness of the malware detection system. This paper aims to contribute to the advancement of cybersecurity by improving the accuracy and efficiency of malware detection, specifically in EXE files. Our strategy involves harnessing these discernible patterns to train artificial neural networks to distinguish between malicious and safe files. Through this process, our project aims to improve the accuracy and efficiency of malware detection.

CCS CONCEPTS

• **Security and privacy** → Malware and its mitigation;

KEYWORDS

Binvis, computer vision, malware detection, InceptionV3, ResNet50, Convolutional Neural Network (CNN), self-organizing incremental neural network (SOINN)

1 INTRODUCTION

As technology continues to evolve, so do the threats to data security and privacy. There are various approaches to identifying such attacks. Some approaches are old, some new, and some are still being improved upon and/or researched. Benign and malicious data were collected from online sources and their binary visualizations were created using the tool binvis. These visualizations will be added to the dataset. Following data collection, researchers retrain InceptionV3, a pre-trained model, ResNet50, on a dataset consisting of images representing both benign and malicious files. The dataset will contain roughly 8000 images of EXE files, equally divided between benign and malicious types. This training process aims to develop a model that accurately classifies each image. The accuracy of the team's final model will be evaluated and compared to the accuracy of the model referenced in previous research by Baptista (2019) as well as the pre-trained model, ResNet50.

2 APPROACH/DESIGN

The approach taken for this paper was slightly different from the typical approaches used for malware detection. The file type focus, both malicious and benign, was EXE files only. The team had originally planned to use four file types: EXE, PDF, DLL, and DOC. However, due to the time constraint, we found it difficult to find a large amount of other file types. Thus, the team decided to focus on EXE files. There were a total of 8000 files used, 4000 benign and 4000 malware downloaded from GitHub[4] [2].

Next was to convert all of the files, both malicious and benign, to images using the website binvis.io[3] one at a time. Thus this process was tedious and took a lot of time. Thankfully, a script was developed using Selenium and Python in combination with the binvis website. This reduced the time for gathering all the images tremendously.

The models that the team decided to use were InceptionV3 and ResNet50. These models are well known models for malware detection and we made some of our own modifications to the models to improve the accuracy.

The InceptionV3 model is a convolution neural network, developed by GoogleLeNet, that works by reducing the number of layers used to train the model and making it less computationally intensive. Each layer applies a filter to the image that then is given to the next layer for as many layers as specified by the user. This model was pre-trained on the ImageNet dataset with 78.1% accuracy[1]. We altered the model to improve the accuracy as shown in the following sections.

The ResNet50 model is also a convolution neural network, but a specific type of CNN called a residual network, that was pre-trained using the ImageNet dataset. It differs from the InceptionV3 model in that it uses a bottleneck design to reduce the number of parameters and matrix multiplication, which makes the training of each layer faster[5]. The original accuracy of this ResNet50 is 92.2%[7] which we improved upon as shown in the following sections.

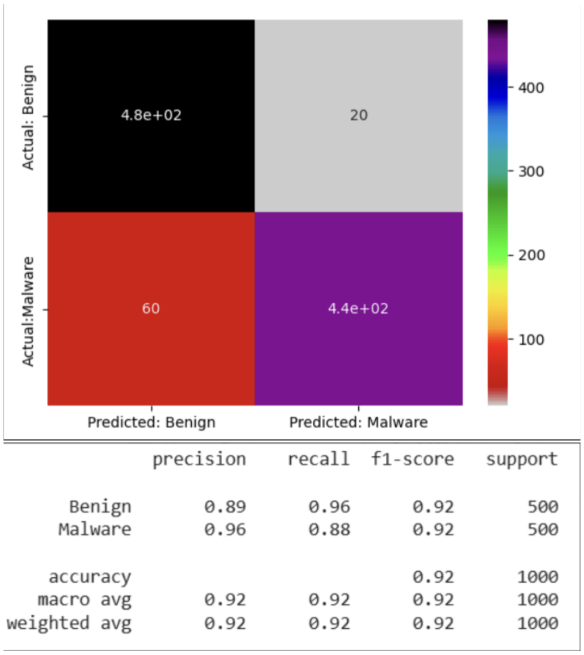
2.1 Implementation

The project initiated with the conversion of executable files into images, followed by resizing to 250 by 250 pixels for enhanced feature extraction. The selected models, InceptionV3 and ResNet, underwent configuration with additional layers, including flattened, dense, dropout, and output layers. The architectures were finalized, accommodating the increased image dimensions. The InceptionV3 model, for example, featured a convolutional base, a flattened layer, a dense layer with 256 units and 'relu' activation, a dropout layer with a 50% rate, and a dense output layer with 'sigmoid' activation. Both models were compiled using the Adam optimizer and binary cross-entropy loss, with accuracy as the monitored metric. To train the models, a dataset was prepared using the image_dataset_from_directory function. The images were organized into training and validation sets, each with a batch size of 32. There were 6000 images used for training, 3000 benign and 3000 malware. Likewise, there were 1000 images used for validation, 500 each. The training process spanned 100 epochs, allowing the models to iteratively learn and adjust their parameters to optimize predictive performance. During training, a ModelCheckpoint callback was employed to save the model with the best performance on the validation set, based on the validation loss. The choice of 100 epochs ensured a comprehensive exploration of the dataset, facilitating the convergence of the models and capturing intricate patterns within the data. This iterative training process, combined with the ModelCheckpoint callback, contributed to the creation of a well-optimized and robust model. Post-training, the models underwent evaluation on a separate test set to distinguish between benign and malignant executable files. There were a total of 1000 images used for testing, 500 malware and 500 benign. Evaluation metrics, including loss and accuracy, were computed and will be discussed further in the Results section below. The InceptionV3 model achieved an accuracy of 92%, while the ResNet model surpassed it with an accuracy of 98% in binary classification.

2.2 Results

Below is the confusion matrix for the first model, InceptionV3, which had an accuracy rate of 92%, a 4% false positive rate, and 12% false negative rate. The model demonstrated an impressive accuracy in correctly classifying files as either malware or safe based on their image representations. Additionally, it exhibited a low 4% false positive rate, indicating that only 4% of the safe files were mistakenly identified as malware. However, there was an 12% false negative rate, meaning that 12% of actual malware files were categorized as benign. While the overall accuracy is high, it's important to keep in mind the false negatives since they are instances where the model failed to detect malicious files. Here are some calculations based on the confusion matrix seen in Figure 1.

Precision: 96%. This is the proportion of true positive predictions among all positive predictions, suggesting that when the InceptionV3 predicts malware, it is correct about 96% of the time. Recall: 88%. This is the proportion of true positive predictions among all actual positives, indicating that InceptionV3 captures 88% of the times malicious files occur. F1 Score: 92%. The F1 suggests a strong balance between precision and recall.



```
              precision    recall  f1-score   support

      Benign       0.89      0.96      0.92       500
     Malware       0.96      0.88      0.92       500

    accuracy                           0.92      1000
   macro avg       0.92      0.92      0.92      1000
weighted avg       0.92      0.92      0.92      1000
```

Figure 1: Confusion Matrix and Classification Report for InceptionV3

In comparison, the self-organizing incremental neural network (SOINN) previously developed by Baptista[6] achieved a 73.7% accuracy rate, a 12.1% false positive rate, and a 14.2% false negative rate for EXE files. In contrast to the SOINN model, our model outperformed significantly with a 92% accuracy in classifying EXE files. Moreover, our model demonstrated a notably lower false positive rate of 4%, indicating a superior ability to discern safe files. These numbers can also be seen in Figure 2, for all file types used for the SOINN model.

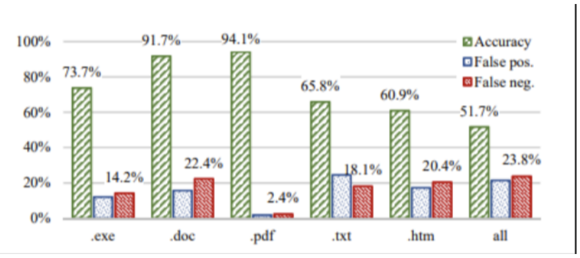
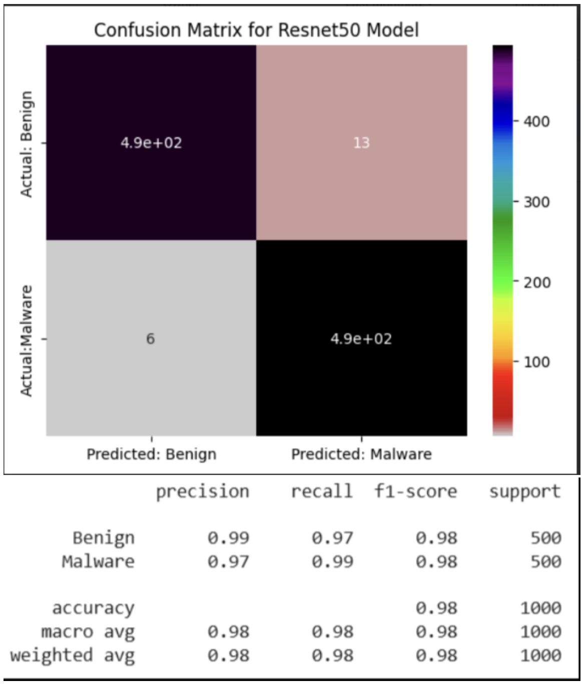


Figure 2: SOINN Accuracy by File Type

The confusion matrix for the second model, ResNet50, which had a 98% accuracy rate, 2.6% false positive rate, and 1.2% false negative rate. This high accuracy, with low false positive and false negative rates, demonstrates Resnet50's performance in distinguishing between malware and safe files. This can be seen alongside its classification report in Figure 3.

Looking at the ROC curve above in Figure 4, we can determine that Resnet50 is the best model as it is closest to the top left corner of the graph. Both models present steep curves as well, suggesting



```
              precision    recall  f1-score   support

      Benign       0.99      0.97      0.98       500
     Malware       0.97      0.99      0.98       500

    accuracy                           0.98      1000
   macro avg       0.98      0.98      0.98      1000
weighted avg       0.98      0.98      0.98      1000
```

Figure 3: Confusion Matrix and Classification Report for ResNet50

better overall performance, as the classifier achieves higher true positive rates with lower false positive rates.

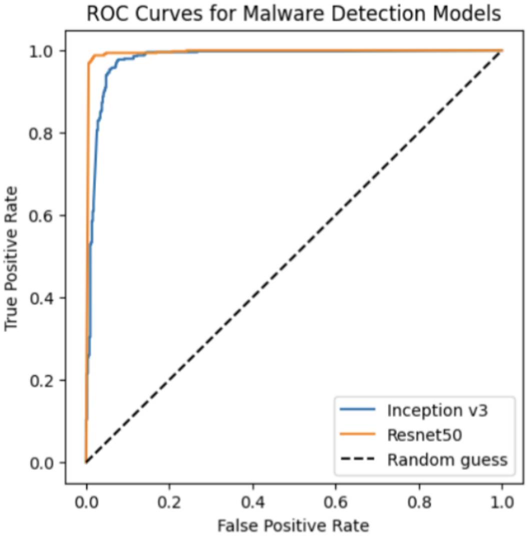


Figure 4: ROC Curve For Malware Detection

In the comparison shown in Figure 5, both the ResNet-50 model and the InceptionV3 model outperform the SOINN model across the board. Both of our models exhibit higher accuracy, lower false positive rates, and lower false negative rates, emphasizing their effectiveness in distinguishing between malware and safe files. The choice between ResNet-50 and InceptionV3 would depend on specific considerations such as computational efficiency, training time, and resource requirements.

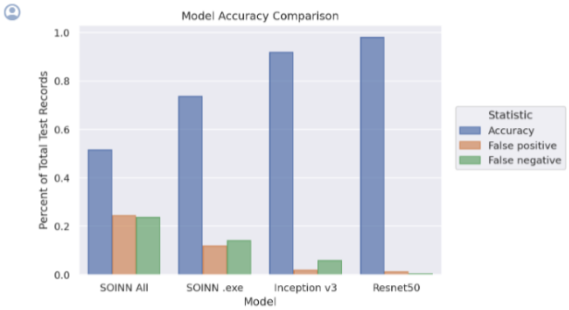


Figure 5: Model Accuracy Comparison For All Models

These improvements in accuracy could be due to the significantly enhanced performance of our malware detection models from the larger dataset we used. This substantial increase in the number of training instances allowed our models to better generalize and learn patterns present in both benign and malicious files. Additionally, employing an input shape of 250 by 250 pixels played a pivotal role in preserving finer details and characteristics from the original images during the training process. This choice in input dimensions helped our convolutional neural network models capture more specific features, contributing to the models' accuracy in classifying malware files from visual representations of EXE files.

3 CONCLUSIONS

In our pursuit of advancing malware detection, we employed Convolutional Neural Networks (CNNs) on binary visualizations of EXE files. Utilizing the Binvis tool, we transformed files into images and trained InceptionV3 and ResNet-50 models on an extensive dataset, achieving significant improvements over the previously developed SOINN model.

Our InceptionV3 model demonstrated 92% accuracy, a 4% false positive rate, and an 12% false negative rate. The ResNet-50 model outperformed with 98% accuracy, a 2.6% false positive rate, and an impressive 1.2% false negative rate. These results underscored the models' efficacy in accurately distinguishing between malicious and safe files, showcasing potential real-world applicability.

Compared to the SOINN model, both InceptionV3 and ResNet-50 exhibited better performance overall, including higher accuracy and lower false positive and false negative rates. The study emphasized the impact of sample size on model performance, with a continuous improvement observed with increased samples.

In conclusion, our research contributes to advanced malware detection by combining innovative data preprocessing and CNN models. The choice between InceptionV3 and ResNet-50 depends on specific considerations. Future work may explore model generalization and the integration of additional features for comprehensive malware detection. Overall, our study marks a significant step forward in enhancing cybersecurity through advanced and reliable malware detection methods.

In the future, since the accuracy of both models was so high, future research could train the same base models on different file types to see if the high-accuracy results would be repeated. The SOINN multiple-file-type classification model had lower accuracy overall than on individual file types. Based on the results from the retrained Inception V3 and ResNet50 models, having dedicated models for respective file types would likely be better than having a common model for all file types. However, further study would be necessary to prove this.

REFERENCES

[1] [n. d.]. Advanced guide to inception V3 | cloud TPU | google cloud. https://cloud.google.com/tpu/docs/inception-v3-advanced. Accessed: 2023-11-29.

[2] [n. d.]. Benign EXE Files Dataset. https://github.com/bormaa/Benign-NET/tree/main. Accessed: 2023-11-29.

[3] [n. d.]. binvis.io visual analysis of binary files. https://Binvis.io. Accessed: 2023-11-29.

[4] [n. d.]. Malware EXE Files Dataset. https://github.com/iosifache/DikeDataset. Accessed: 2023-11-29.

[5] [n. d.]. ResNet-50: Understanding Residual Networks in Deep Learning. https://datagen.tech/guides/computer-vision/resnet-50/. Accessed: 2023-11-29.

[6] Irina Baptista, Stavros Shiaeles, and Nicholas Kolokotronis. 2019. A Novel Malware Detection System Based On Machine Learning and Binary Visualization. *CoRR* abs/1904.00859 (2019). arXiv:1904.00859 http://arxiv.org/abs/1904.00859

[7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep Residual Learning for Image Recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 770–778. https://doi.org/10.1109/CVPR.2016.90