A Vision Based Deep Learning Framework for Malware Detection and Classification

Dr. Chaitanya Udatha[1] , Y.S.S.K Keerthija[2] and C. Shashank Reddy[3]

[1,2,3] Information Technology, Mahatma Gandhi Institute of Technology, Hyderabad, Telangana, 500075, India

DOI: https://doi.org/10.51584/IJRIAS.2026.11050042

Received: 30 April 2026; Accepted: 05 May 2026; Published: 27 May 2026

ABSTRACT

Malware detection is a complex task for signature-based anti-virus software, especially for polymorphic malware and zero-day attacks. However, this project proposes a vision-based static malware detection and classification method that represents raw executable file bytes as fixed-size grayscale images called byte plots and attempts to classify malware families based on these images without executing them.

In this project, for the proposed model, the best architecture is Convolutional Neural Networks (CNN) + Random Forest (CNN-RF). Initially, a CNN is trained to learn discriminative feature embeddings for byte plot images. Once this is done, the final softmax classifier is removed, and this CNN is used to generate a 256-dimensional vector for each input.

Then, a class-balanced Random Forest is trained to predict the malware family and confidence scores. In this way, this proposed method is able to achieve better results for two different datasets, and the best results obtained are 98.07% for MalImg and 93.07% for MaleVis.

INTRODUCTION

Malware, which stands for malicious software, refers to programs that are designed to cause harm to computer systems and users by stealing their information, causing system disruption, gaining unauthorized access to computer systems and data, and encrypting files and demanding ransom. Malware has taken different forms in recent years, including viruses, worms, trojan horses, spyware, ransomware, and backdoors. One of the biggest challenges facing the fight against malware is that it is constantly evolving and has become a scalable threat.

This is because malware developers have been constantly developing new versions of malware. Malware has undergone tremendous transformation in recent years. In the past, malware was mostly propagated by removable media and network propagation. However, in recent years, malware has become sophisticated and has adopted new techniques to avoid being detected by antiviral programs. This has made the traditional approach to fighting malware obsolete.

This approach includes using signature-based techniques that depend on known byte patterns and hashes. This approach has become obsolete due to the polymorphism exhibited by modern malware. Signature-based techniques have become the most popular and widely used techniques in fighting malware due to their ease of implementation and execution.

In order to tackle the above-mentioned challenges, this project focuses on the vision-based approach for the static classification of malware. The major reason for choosing this approach is the fact that, according to research, malware tends to have unique textures in the visual representation of the malware, which can be utilized for the efficient classification of the malware by using deep learning algorithms, especially the CNN model, which can automatically learn the patterns from the images of the malware.

ISSN No. 2454-6194 | DOI: 10.51584/IJRIAS |Volume XI Issue V May 2026

The major contributions of the proposed work are the fact that it develops a strong and efficient framework for the classification of malware by using a combination of the CNN model and the Random Forest classifier, which can provide the highest level of accuracy for the efficient classification of the malware, as the proposed framework is tested on two separate datasets, namely MalImg and MaleVis, which can help the proposed framework to perform accurately on different datasets, thereby proving the efficiency of the vision-based approach for the classification of malware.

LITERATURE SURVEY

Ahmed, S. R., et al. proposed a malware detection approach using image processing and machine learning techniques. Malware binaries are converted into images, and features are extracted using CNN, followed by classification using Decision Tree (DT) and Random Forest (RF) models for improved detection accuracy [1].

Aslan, O., and Abdullah Asim Yilmaz introduced a deep learning-based malware classification framework that converts malware binaries into images and applies CNN architectures for automatic feature extraction and classification, achieving high performance compared to traditional methods [2].

Yoo, Suyeon, et al. proposed AI-HydRa, a hybrid malware classification system combining deep learning and Random Forest. The model extracts features using deep neural networks and enhances classification accuracy using ensemble learning techniques [3].

Moawad, Ahmad, Ahmed Ismail Ebada, and Aya M. Al-Zoghby presented a survey on visualization-based malware detection, where malware binaries are transformed into images and analyzed using both traditional feature extraction methods and CNN-based deep learning models for classification [4].

Sewak, M., Sahay, S. K., and Rathore, H. investigated a deep learning-based malware detection system using neural networks for automatic feature learning. The approach focuses on improving detection accuracy by leveraging deep architectures over handcrafted feature-based methods [5].

Buriro, A., et al. proposed MalwD&C, a machine learning-based system that extracts structural features from executable files and applies multiple classifiers such as Random Forest, KNN, and Logistic Regression, with Random Forest achieving the best performance for malware detection and categorization [6].

Tayyab, Umm-e-Hani, et al. presented a survey on recent trends in deep learning-based malware detection, highlighting various approaches including CNN, RNN, and hybrid models, and emphasizing the importance of combining static and dynamic analysis techniques [7].

Vinayakumar, R., et al. proposed a robust malware detection system using deep learning models such as CNN and DNN applied on large-scale datasets. The system focuses on automated feature extraction and high accuracy detection using scalable architectures [8].

Akhtar, M. S., and Tao Feng proposed a real-time malware detection system using a hybrid CNN-LSTM architecture. CNN is used for feature extraction, while LSTM captures sequential patterns in data, improving detection of complex malware behavior [9].

Karat, Gautam, et al. developed a CNN-LSTM hybrid model for enhanced malware analysis, where CNN extracts spatial features and LSTM learns temporal dependencies from behavioral data, leading to improved detection performance and generalization [10].

ISSN No. 2454-6194 | DOI: 10.51584/IJRIAS |Volume XI Issue V May 2026

METHODOLOGY

Data set

To see how effectively the new malware classifier performs, one has to consider the MalImg and MaleVis sets. Images created by program code lines that are harmful are in black and white. This kind of image can be processed by programs that analyze images without any changes . One group lets researchers check how well systems spot odd patterns in varied cases.

Right away, the MalImg set includes 9,339 images linked to 25 types of malware. Visuals emerge from byteplots - raw binary data transformed into pixel tones. Earlier work relied on this dataset often, so scientists chose it again as the core reference. Though certain groups appear more frequently, a division preserving ratios handled distribution: seventy percent fed training phases. A chunk of room - fifteen percent - was set aside just to track how things were moving while learning happened. Behind that, a different slice, also one in six parts, waited untouched until the very end when lessons had already stopped.

Picture 14,226 images sorted across 26 kinds of malware - that's where MaleVis began. Testing model flexibility meant checking shifts in data behavior. Since even distribution counts during evaluation, the setup used 70 percent training, balanced by two equal slices afterward: 15 for checks, another 15 held aside later. Each portion followed identical grouping rules. Still, one collection of data can miss details. When tried on a fresh batch though, the approach proves steady against unfamiliar malware forms. Even as habits change, outcomes remain consistent - just what live environments face every day.

Data pre-processing

The malware samples are first converted to byte-plot images. This is done by taking the first 122,500 bytes of the malware file and transforming them into a 350x350 matrix. If the malware file is smaller than 122,500 bytes, zero padding is done to make it of the required size. This is done to convert the malware file into an image so that it can be analyzed without the need for execution. The malware samples are then divided according to the malware family and are divided further into training, validation, and test sets. For the CNN model, the images are resized to 224x224 pixels and then converted to grayscale and normalized to the range [0, 1].The data augmentation techniques are applied to the training set of the data. This includes flipping, rotation, cropping, and noise addition. The CNN model then converts the malware image into a feature vector of size 256. This feature vector is then given as input to the Random Forest classifier. The data preprocessing step includes the conversion of the malware image and the generation of the feature vector for the Random Forest classifier.

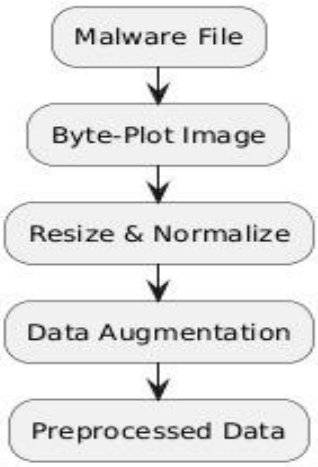


Fig. 1. Data Preprocessing workflow

ISSN No. 2454-6194 | DOI: 10.51584/IJRIAS |Volume XI Issue V May 2026

In the data preprocessing step of this project, the malware is prepared for the CNN-RF model in a similar manner for training and testing. The raw malware is converted to a byte plot image by extracting the first 122,500 bytes of the file and mapping each byte to a pixel on a 350x350 matrix, zero-padding the file if it is shorter than 122,500 bytes to ensure a consistent size for all images. The binary data representing the malware are mapped to grayscale images by considering the byte (0-255) representation of the malware against the pixel intensity value. These grayscale images are classified according to the family of malware and divided into training, validation, and testing datasets using the stratified sampling method, ensuring the equal distribution of samples in all malware classes. Before inputting the images to the CNN model for further operations, the images are first preprocessed by resizing them to 224x224 and normalizing their values between [0,1]. To improve the generalization capability of the CNN model, data augmentation techniques are employed on the training data. The function of the CNN is that of a feature extraction process that maps an image to a low dimensional feature vector of length 256.

Proposed architecture

The proposed architecture will combine the use of the CNN and RF algorithms, striving to enhance the effectiveness of malware classification processes. In particular, the byte-plot images are used for the inputting of malware. The images will be in gray scale, and there is direct correspondence between bytes and pixel intensity values. Then the images will be fed into the CNN as an input to become feature detectors. The features detected will become the input to the RF algorithm, which uses those features for predictions. In particular, the CNN will automatically detect distinct patterns in the images by using the gray scale images of size $224 \times 224 \times 1$. Following feature learning, a Global Average Pooling layer is applied to reduce spatial dimensionality. This is followed by fully connected layers that produce a compact representation of the images. The CNN produces a feature vector of dimensionality 256, which encodes the essential characteristics of each malware. Although the CNN is originally intended for classification and comprises a softmax output layer during training, this layer is discarded after training to make it suitable for feature extraction. The Random Forest classifier uses the feature embeddings obtained by the CNN as input. The Random Forest classifier accepts a feature matrix of size N x 256 as input, where N is the total number of samples. The Random Forest classifier is an ensemble of decision trees that is used for classification. In this work, we use 200 trees in the Random Forest classifier with class balancing to effectively deal with imbalanced malware families. The predicted malware family is obtained as output by the Random Forest classifier.

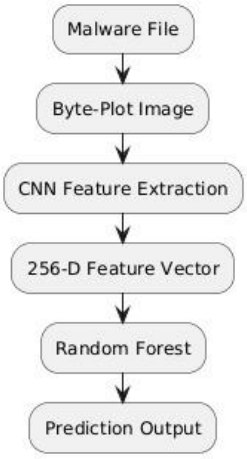


Fig. 2. Proposed architecture of vision based deep learning framework for malware classification using CNN + RF.

The training of the model is done in a two-stage manner. In the first stage, the CNN model is trained in a supervised manner using the labeled malware images, which helps the model to learn the feature representations of the images in a robust manner. In the second stage, the feature embeddings of all the images are obtained by passing the images through the pretrained CNN model, and then the RF classifier

ISSN No. 2454-6194 | DOI: 10.51584/IJRIAS |Volume XI Issue V May 2026

is trained on the feature embeddings obtained from the CNN model. During the testing phase, the same operations are performed on the unseen binary file, and the final classification is done by passing the feature representation of the file obtained from the CNN model to the RF classifier. The proposed CNN-RF model uses the advantages of the CNN model and the RF classifier to improve the accuracy of the malware detection and classification tasks.

Proposed work

The proposed work will develop a static malware classification framework without executing or disassembling binary files, thus ensuring safety and efficiency. The proposed method will convert malware binaries into images called byte plot images, where byte values will be converted to grayscale pixel intensities. The proposed method will utilize the characteristic that different malware families have different visual features and will classify them using image learning techniques. The process begins by transforming malware binaries into grayscale byte-plot images of a fixed size. This is done by extracting the first 350×350 bytes from each file, reshaping them into two-dimensional arrays, and applying zero padding when necessary to maintain consistency. The generated images are then resized and normalized before being provided as input to a Convolutional Neural Network (CNN) used for feature extraction. The CNN captures important visual characteristics from these images and generates a 256-dimensional feature vector representing each sample. These feature vectors are then supplied to a Random Forest (RF) classifier, which is responsible for performing the final classification of malware families. The employment of the class balanced RF classifier ensures that the imbalanced data sets are properly addressed, thereby improving the reliability of the prediction for all malware families. In addition, the classifier provides the class and corresponding probabilities. The training process is performed in a structured manner, where first, the CNN is trained in a supervised manner to learn meaningful representations. Then, feature embeddings are extracted using the trained CNN, and this is used to train the RF classifier. Once this is done, for a new malware sample, the same process is repeated to arrive at the final prediction using the RF classifier. The hybrid model of CNN and RF has the strong feature learning capacity of deep learning and the robustness of ensemble learning, and it is efficient and accurate in malware classification.

RESULTS AND DISCUSSION

To evaluate the different architectures, several architectures were tested (including both a custom built CNN, transfer learning models like ResNet-50 & EfficientNet-B0 and finally the hybrid CNN + Random Forest architecture). For the experiments with just the custom CNN, it consistently produced similar performance, but struggled with the more complex nature of the decision boundaries for those classification tasks. With Transfer Learning models, the deeper network architectures improve the representation of features in comparison to the custom CNN architecture. However, the transfer learning models still had some issues with respect to class imbalance and the neural networks did not all generalize well. In comparison to these other architectures, the hybrid CNN + RF architecture outperformed all other approaches across both datasets. The CNN portion of the model produced useful and discriminative features from the malware images, meanwhile the RF classifier was able to make better decisions for given instances through an ensemble approach. Thus the Two-Part Model architecture was able to better manage complex distributions of features and provide for a lower likelihood of overfitting. Further the hybrid model produced more balanced classifications across classes, demonstrating good performance characteristics. In summary, the hybrid of the CNN + RF architecture was more efficient and reliable compared to all the other standalone and transfer learning-based architectures.

ISSN No. 2454-6194 | DOI: 10.51584/IJRIAS |Volume XI Issue V May 2026

Confusion matrix of CNN + Random Forest (CNN+RF) model

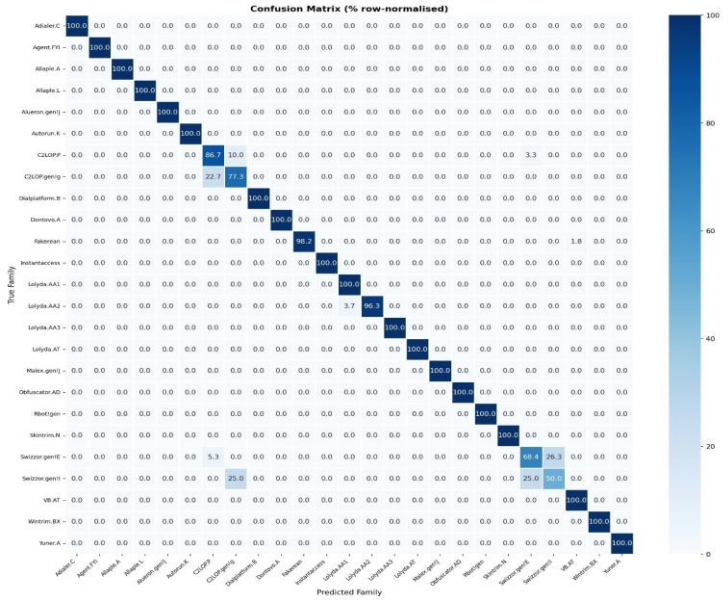


Fig. 3. Confusion matrix for the MalImg dataset.

Figure 3 shows a snapshot of how the malware model performs appears here, displayed as a row-normalized confusion matrix. One looks down rows to find real families assigned during testing. Columns instead reveal what the system guessed for each case. Where these meet, percentages show how often predictions matched - or missed - the truth. High numbers along the diagonal mean correct guesses stood out clearly. Down the middle line, values pile up tightly. Close to all types score extremely high on accuracy. This means the model correctly identifies nearly every harmful file. When results line up so consistently, it points to solid performance overall. Almost every group sees strong precision hold steady.

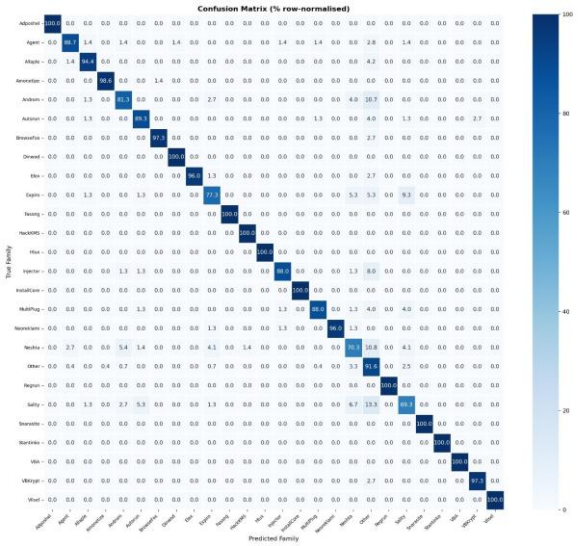
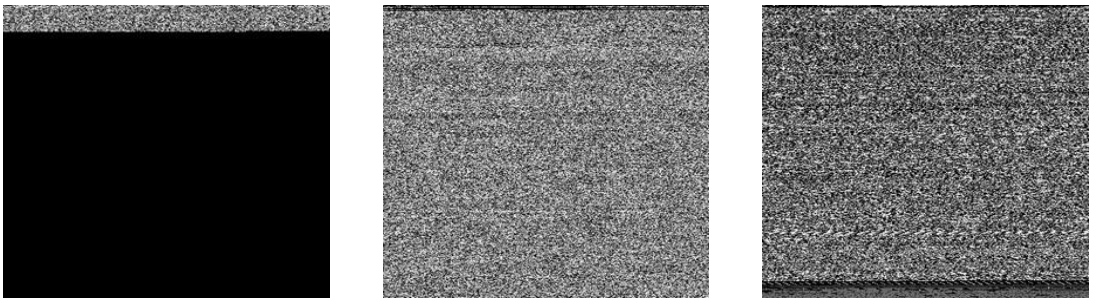


Fig. 4. Confusion matrix for the MaleVis dataset.

Figure 4 : A glance reveals a table split into rows and columns, adjusted per row to reflect proportions. Each horizontal line corresponds to one real malware group. Predictions spread out vertically, marked across the top as column headers. Numbers within cells tell what fraction of samples were assigned there. Mistakes emerge as off-diagonal highlights, revealing which families the system confused. These slips trace links between certain categories, showing where the model blurs boundaries. Where things sit makes it easier to see which groups mix up their names a lot. Every now and then, most marks land along

the diagonal, meaning known threats get spotted nearly always. It keeps steady when tested on different types, separating them without breaking a sweat. The pattern stays clear - familiar names show up where expected. A few off-diagonal points catch the eye - mistakes slip in now and then, particularly around Agent, Androm, InstallCore, that crowd. When malware shares too many structural traits, things blur. Yet clarity holds firm across most of the data precision, even across categories, and it holds up well on new data.

Dataset: MalImg



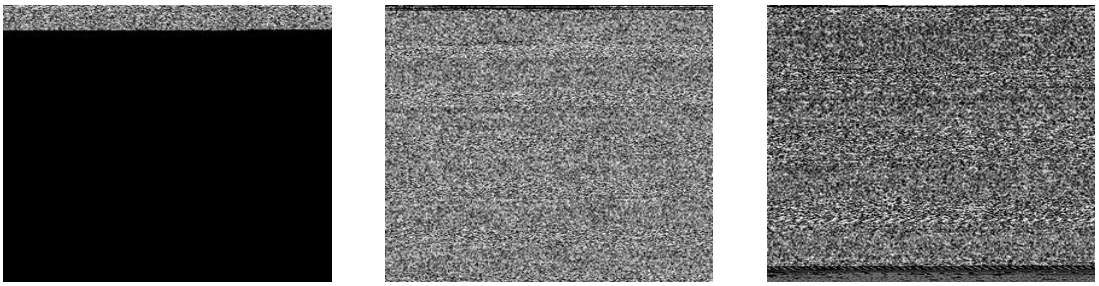
a) Predicted malware family - Lolyda.AA3 Confidence - 57.00%

b) Predicted malware family – Allaple A Confidence – 54.00%

c) Predicted malware family – Rbot!gen Confidence – 28.00%

Fig. 5. Sample predictions on MalImg dataset.

Dataset: MaleVis



a) Predicted malware family - Agent Confidence – 25.50%

b) Predicted malware family – Neoreklami Confidence – 28.50%

c) Predicted malware family – VBKrypt Confidence – 28.00%

Fig. 6. Sample predictions on MaleVis dataset.

Performance comparision analysis

Table 1. Performance comparison on malimg dataset

Model	Accuracy	MacroF1	Top-3	Recall	Precision
Custom CNN	88.37%	65.77%	99.14%	68.11%	71.40%
ResNet-50	91.15%	80.74%	99.50%	82.85%	80.90%
EfficientNet	94.50%	82.48%	99.71%	82.80%	83.50%
CNN+RF(proposed)	**98.07%**	**95.07%**	**99.86%**	**95.08%**	**95.19%**

Table 2. Performance comparison on malevis dataset

Model	Accuracy	MacroF1	Top-3	Recall	Precision
Custom CNN	80.47%	78.66%	90.82%	80.16%	79.73%
ResNet-50	86.00%	86.56%	92.97%	85.78%	88.62%
EfficientNet	80.98%	81.62%	91.43%	80.69%	84.20%
CNN+RF(proposed)	**93.07%**	**93.73%**	**97.85%**	**93.22%**	**94.32%**

CONCLUSION

The aim of this project included creating a system that could detect and classify malware using machine vision techniques, with regards to binary files. The binary files were transformed into grayscale images to be detected by neural networks based on their distinct visual characteristics as malware. Different deep learning models were compared against each other in terms of accuracy on various benchmarking datasets, including MalImg and MaleVis, like Custom CNN, ResNet-50, EfficientNet-B0, and Hybrid CNN combined with Random Forest algorithms. In addition, the Flask server was designed to develop an application that would have functionalities of file upload, prediction, and scoring. The performance of the model was determined using comparative studies and confusion matrices. Amongst all the proposed models, Hybrid CNN + Random Forest generated the most efficiency and effectiveness in differentiating malware families.

Future Enhancements

Never the less, although the model has performed exceptionally well, there are some areas that require improvement. For instance, better results can be attained by employing Vision Transformers in the detection of malware from images. Moreover, the inclusion of visual features along with other contextual attributes like APIs and execution traces might prove helpful in improving the performance of the whole detection process. The integration of the system with a real-time monitoring mechanism will lead to prompt detection and countermeasures against the cyber attack. The problem of class imbalance can be overcome by utilizing artificial data creation methods. Finally, ensuring clarity regarding the decision-making process of the machine learning model is crucial for establishing trust.

REFERENCE

1. Ahmed, S. R., et al. ”A novel approach to malware detection using machine learning and image processing.” Proceedings of the Cognitive Models and Artificial Intelligence Conference, 2024. https://dl.acm.org/doi/abs/10.1145/3660853.3660931
2. Aslan, O., and Abdullah Asim Yilmaz. ”A new malware classification¨ framework based on deep learning algorithms.” IEEE Access, 2021. https://ieeexplore.ieee.org/abstract/document/9455368
3. Yoo, Suyeon, et al. ”AI-HydRa: Advanced hybrid approach using random forest and deep learning for malware classification.” Information Sciences, 546 (2021). https://www.sciencedirect.com/science/article/pii/S0020025520308525
4. Moawad, Ahmad, Ahmed Ismail Ebada, and Aya M. Al-Zoghby. ”A Survey on Visualization-Based Malware Detection.” Journal of Cybersecurity, 4.3 (2022). https://www.academia.edu/104208673/A_Survey_on_Visualization_Based_Malware_Detection
5. Sewak, M., Sahay, S. K., and Rathore, H. ”An investigation of a deep learning based malware detection system.” Proceedings of the 13th International Conference on Availability, Reliability and Security, 2018.

https://dl.acm.org/doi/abs/10.1145/3230833.3230835

6. Buriro, A., et al. ”MalwD&C: a quick and accurate machine learningbased approach for malware detection and categorization.” Applied Sciences, 13.4 (2023). https://www.mdpi.com/2076-3417/13/4/2508
7. Tayyab, Umm-e-Hani, et al. ”A survey of the recent trends in deep learning based malware detection.” Journal of Cybersecurity and Privacy, 2.4 (2022). https://www.mdpi.com/2624-800X/2/4/41
8. Vinayakumar, R., et al. ”Robust intelligent malware detection using deep learning.” IEEE Access, 7 (2019). https://ieeexplore.ieee.org/abstract/document/8681127
9. Akhtar, M. S., and Tao Feng. ”Detection of malware by deep learning as CNN-LSTM machine learning techniques in real time.” Symmetry, 14.11 (2022). https://www.mdpi.com/2073-8994/14/11/2308
10. Karat, Gautam, et al. ”CNN-LSTM hybrid model for enhanced malware analysis and detection.” Procedia Computer Science, 233 (2024). https://www.sciencedirect.com/science/article/pii/S1877050924005982