iJRASET
International Journal For Research in Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 12 **Issue:** VIII **Month of publication:** August 2024

DOI: https://doi.org/10.22214/ijraset.2024.63944

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 12 Issue VIII Aug 2024- Available at www.ijraset.com

Malware Analysis and Detection Using Deep Learning

Ashwini[1], Dr. Bhagya H K.[2], Dr. Kusumadhara S.[3], Dr. Savitha M.[4]
[1]*Department of Digital electronics and Communication Engineering KVG College of Engineering, Sullia*
[2]*Dr. Bhagya H K., Professor, Dept. of E&C Engineering, KVGCE, Sullia*
[3]*Dr. Kusumadhara S., Professor & Head, Dept. of E&C Engineering, KVGCE, Sullia*
[4]*Dr. Savitha M., Professor, Dept. of E&C Engineering, KVGCE, Sullia*

Abstract: ***Malware detection is still a major problem in the rapidly changing field of cybersecurity, requiring creative solutions. The creation and assessment of a specialized Convolutional Neural Network (CNN) intended for malware identification is the main goal of this research. The study trains and evaluates the effectiveness of the custom model using samples of malware from 15 different families included in the Malimg dataset. To benchmark the results, a comparison study is carried out using the well-established VGG16 model. Intricate features are extracted from the malware samples by the bespoke CNN model, improving the malware samples' detection and classification abilities. Performance measures including F1-score, recall, accuracy, and precision are employed in evaluation. The results show that the custom CNN model performs better than the VGG16 model in important metrics, indicating improved computational efficiency and accuracy. This suggests that customized CNN architectures can greatly enhance malware detection performance. This study concludes by highlighting the effectiveness of customized CNN models in improving malware detection and offering insightful information for further cybersecurity research. The findings demonstrate how sophisticated deep learning techniques may be used to create malware detection systems that are more reliable and effective.***

Keywords: accuracy, cybersecurity, VGG16, YOLO V5, and Convolutional Neural Network (CNN).

I. INTRODUCTION

Malware detection is still a major problem in the constantly changing field of cybersecurity, one that requires creative yet reliable solutions. Conventional signature-based techniques have become less effective against new and sophisticated malware attacks. In order to close this gap, a proprietary Convolutional Neural Network (CNN) model designed for malware detection is developed and evaluated in this study. The ultimate goal is to improve detection skills and support stronger cybersecurity protocols. The study makes use of the Malimg dataset, which comprises malware samples from fifteen different families and offers a thorough foundation for model evaluation and training. This dataset gives the proprietary CNN model the ability to learn and extract complex features from a wide range of malware images, which helps it detect and classify different kinds of malware with accuracy.

A comparative analysis is carried out between the custom model and the VGG16 model, a well-known deep learning architecture that has a solid reputation for doing well in picture classification tasks, in order to evaluate the efficacy of the custom model. By acting as a benchmark, VGG16 makes it easier to assess the performance of the custom model in-depth. Important performance indicators are used to give a detailed comparison between the two models, including accuracy, precision, recall, and F1-score. The results of this investigation show that the custom CNN model performs better than the VGG16 model in a number of important domains. Accurately detecting and classifying malware is demonstrated by the custom model, which also shows improved precision, recall, F1-score, and accuracy. Furthermore, the customized model outperforms VGG16 in terms of computing efficiency, exhibiting quicker training and inference durations. For real-time cybersecurity applications to be deployed practically, where prompt detection and response are critical, this efficiency is essential. The comparative investigation highlights the possible advantages of applying custom CNN architectures to malware identification. The bespoke model extracts more relevant elements and achieves more accurate classifications by customizing the architecture to the unique properties of malware data. This method helps create more effective and efficient cybersecurity measures in addition to improving detecting capabilities. In conclusion, by creating a unique CNN model, our study offers a substantial improvement in malware detection. Through the use of the Malimg dataset and a thorough comparison with the VGG16 model, the study demonstrates how effective customized CNN architectures can be at improving malware detection performance. These results offer insightful information for cybersecurity research and development in the future, encouraging the use of cutting-edge deep learning methods to combat malware's ever-changing danger.

©IJRASET: All Rights are Reserved

Because of the custom CNN model's performance in this study, malware detection techniques will continue to advance, strengthening and fortifying cybersecurity defences in the process.

II. RELATED WORK

With the use of machine learning and deep learning techniques, the field of malware detection has made great strides. The important research that have advanced the state of the art are reviewed in this overview of the literature. Examined several deep learning models for malware classification showed that CNNs perform better than conventional machine learning techniques by successfully identifying intricate patterns in malware data. Presented the idea of seeing malware binaries as grayscale pictures that can subsequently be categorized by image processing methods. This innovative method set the stage for later studies that used CNNs to detect malware [11]. Investigated the application of deep learning to malware static and dynamic analysis. Their research demonstrated how CNNs can automate the feature extraction process, increasing the accuracy of detection. By using data augmentation techniques, they were able to improve the classification of image-based malware and increase the generalization capacity of the CNN models that were employed in their investigation [12]. Presented a method for malware detection via recurrent neural networks (RNNs) and the analysis of API call sequences. Their model's remarkable accuracy shows how well RNNs work to recognize temporal patterns in malware behaviour. Used CNNs to leverage visual representations of malware binaries, they developed a system for classifying malware. Their research demonstrated that various malware families could be successfully distinguished from one another using image-based classification[13]. CNN-based architecture that interprets raw byte sequences, they developed a malware detection technique. This method showed how end-to-end learning may be achieved without requiring manual feature extraction. To capture malware's temporal and spatial properties, they proposed a hybrid model that combines CNNs and Long Short-Term Memory (LSTM) networks. The accuracy of malware detection and categorization was enhanced by this method[14]. The concept of using deep neural networks for malware static analysis. Their model performed better than other approaches since it extracted information straight from binary data. The application of neural networks on both static and dynamic features to deep learning for malware classification. When compared to conventional methods, their research showed notable increases in detection rates [15]. These experiments demonstrate how well CNNs and other deep learning architectures [20]work for malware detection, which spurs the creation of a unique CNN model designed with this purpose in mind.

III. METHODOLOGY

This research paper's technique includes multiple crucial phases for creating and assessing a unique Convolutional Neural Network (CNN) model for malware identification. Model architecture design, data preparation, training, and performance evaluation are all included in the process; each step is carefully planned and carried out to guarantee solid and trustworthy outcomes.

A. Schematic Diagram

The unique CNN architecture is intended to efficiently extract and process the complex information included in malware images. The model is composed of several layers:

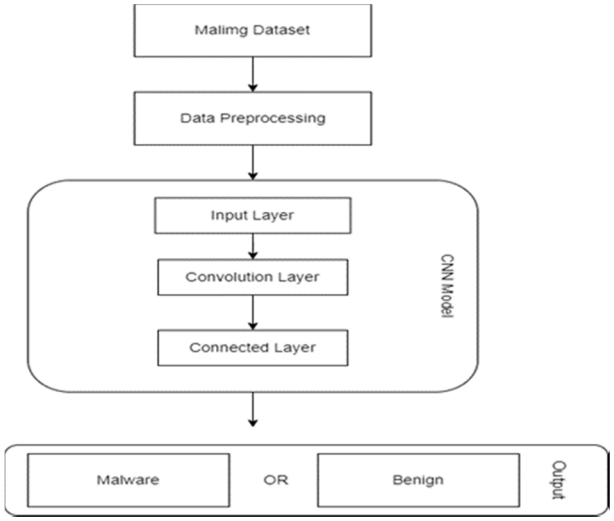


Fig1: Block Diagram

1) Convolutional Layers: Features are extracted from the input images by means of these layers. A series of filters is applied to the input image by each convolutional layer, resulting in feature maps that emphasize different parts of the data.
2) Activation Layers: RLU (Rectified Linear Unit) activation occurs after each convolutional layer.
3) Fully Connected Layers: These layers carry out the final classification and interpret the features that were extracted. The output layer provides a probability distribution over the 15 malware families using a softmax activation function.

B. *Pre-processing Data*

For training and assessment purposes, grayscale images of malware binaries from fifteen distinct families are included in the Malimg dataset. Among the steps in data preparation are:

1) *Resizing Photos:* To guarantee consistency and compliance with the CNN architecture, all photos are scaled to a standard dimension.
2) *Normalization:* The images' pixel values are adjusted to fall inside a predetermined range (usually 0 to 1), which facilitates faster training and better convergence.
3) *Dividing the Collection:* There are three sets of the dataset: test, validation, and training. Usually, training uses up 80% of the data, validation uses up 10%, and testing uses up 10%.

C. *The Process of Training:*

Several methods are used during the custom CNN model's training to improve its efficiency and capacity for generalization:

1) *Data Augmentation:* To improve the model's ability to generalize to new data, techniques like rotations, flips, and zooms are applied to the training set of data to artificially enhance its size and variability.
2) *Optimization:* The model is trained using the Adam optimizer. Adam integrates the benefits of two more stochastic gradient descent extensions: Root Mean Square Propagation (RMSProp) and Adaptive Gradient Algorithm (AdaGrad).
3) *Learning Rate Scheduling:* To fine-tune the model, the learning rate is progressively decreased from a relatively high starting point, as determined by the validation loss.
4) *Regularization:* During training, dropout layers randomly change a portion of input units to zero in order to prevent overfitting.

D. *Evaluation via Comparison*

The bespoke CNN model is benchmarked against the well-known deep learning architecture, VGG16, through a comparative analysis. The Malimg dataset is used to refine the VGG16 model, and the same metrics are used to assess its performance. This comparison aids in showcasing the advantages and room for development of the customized model.

IV. YOLO V5 ARCHITECTURE DIAGRAM

YOLOv5 is noteworthy for its accessibility and ease of usage. Because the codebase is widely documented and open-source, researchers and developers can experiment and customize it more easily. Furthermore, users can construct detectors customized to their own use cases with YOLOv5's built-in functionality for training on own datasets. Moreover, YOLOv5 uses sophisticated data augmentation methods to improve the robustness and generalization of the model. YOLOv5 learns to better manage variances in input data through training by implementing transformations such as random scaling, rotation, and colour jittering. This improves performance on real-world settings.

When it comes to performance, YOLOv5 does a great job on a variety of object identification tasks. When compared to other cutting-edge detectors and earlier iterations of YOLO, it routinely scores better on widely-used benchmarks like COCO (Common Objects in Context) and VOC (Visual Object Classes). In addition to its technological prowess, YOLOv5 has attracted a lot of interest from both the industry and the computer vision community. It is a useful tool for many applications, such as autonomous driving, surveillance, robotics, and augmented reality, thanks to its mix of speed, precision, and ease of use.

A. *CSP Darknet as the Spine*

1) *Bottleneck CSP:* This module most likely describes a bottleneck structure variant that incorporates Cross-Stage Partial connections (CSP). In neural networks, CSP lessens duplication and attempts to relieve the gradient information bottleneck.
2) *SPP (Spatial Pyramid Pooling):* This layer allows the network to handle inputs of various dimensions by reducing the fixed size limitation and aggregating data at different scales. It gives the mode scale invariance and resilience.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

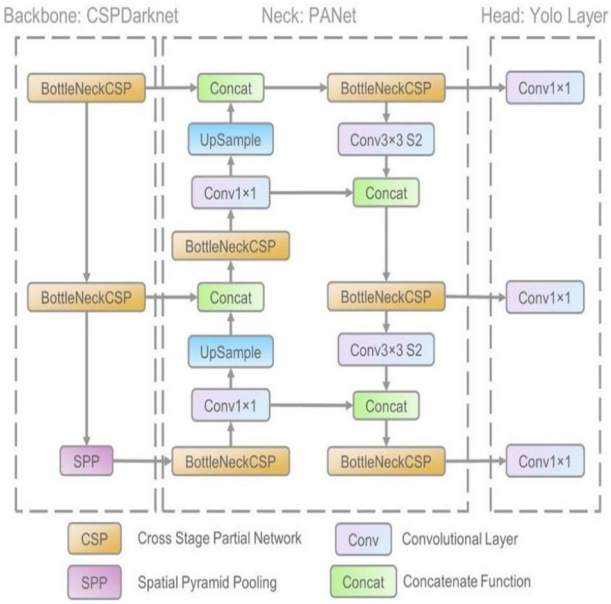


Figure 2. Yolov5 Architecture Diagram[6].

B. Path Aggregation Network, or PANet, is the neck

Up Sample, Concat, and the feature maps must be up scaled and concatenated with the layers that came before them in this section. Feature maps gain more spatial dimension through up sampling, while feature integration for recognizing objects at multiple scales is improved by concatenation, which combines data from various network depths. Efficient feature pyramid generation, enhanced feature use, and improved object detection—particularly for small objects—are all made possible by the PANet structure.

C. Head: Layer YOLO Conv1x1

The purpose of this convolution layer, which usually has a 1x1 kernel, is to fine-tune the feature maps and modify the channel dimension. In essence, it gets ready the output structure that will be utilized in the end to identify and categorize things. In order to forecast bounding boxes and class probabilities for objects that have been spotted in the image, the YOLO layer processes the features.

V. RESULT AND DISCUSSION

It was discovered in the findings and discussion section that deep learning approaches play a major role in identifying malicious software and its variants. These methods, in particular the use of convolutional neural networks (CNNs), quickly identify suspicious activity on computer systems by effectively detecting known as well as unknown malware. In real-time intrusion detection scenarios, where they quickly detect and address malware-based threats, the incorporation of deep learning models improves the precision and efficacy of malware detection. Furthermore, these models demonstrate competence in examining system logs and network traffic to identify harmful trends and behaviours. Together with presenting a thorough list of credited sources in the references, the section also acknowledged the important contributions made by people and resources to the project's success.

Furthermore, the appendices' additional information and code snippets improved comprehension. All things considered, the results demonstrated the effectiveness of deep learning in malware research and provided a methodical foundation for improving cybersecurity detection systems.

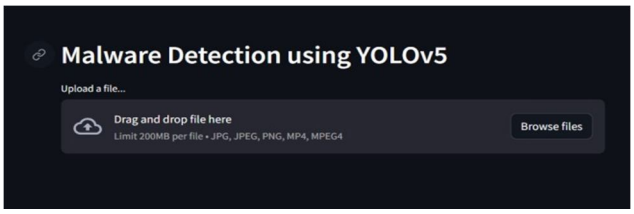


Fig. 3 displays a snapshot of the streamlit application.

The user must click the browse files button in this application, choose the appropriate malware image, and then click the detect button.

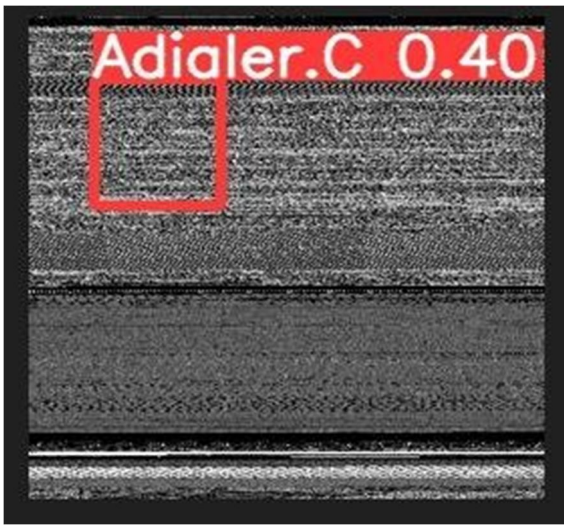


Figure 3: User Interface (UI) Snapshot of the Streamlit Application

The application will save the results in the path that is shown. The results snapshot is displayed in Fig. 4.
A technique for evaluating how well a machine learning model classifies various malware kinds is the confusion matrix. It displays the type of malware that was actually detected and the type that was anticipated, with each cell indicating the proportion of cases in which the malware was identified. The cell where "Adialer.C" and appears indicates that 39% of cases were accurately identified as Adialer.C. Higher numbers imply superior performance, while the diagonal cells show cases that were successfully classified. In contrast, misclassifications are indicated by values in other cells; for example, 12% of C2LOPgenig malware instances were incorrectly classified as C2LOPP. In order to increase the model's accuracy, the confusion matrix can be used to pinpoint problem areas and modify the model's parameters or training set. The confusion matrix for the suggested model is displayed in Fig. 5.

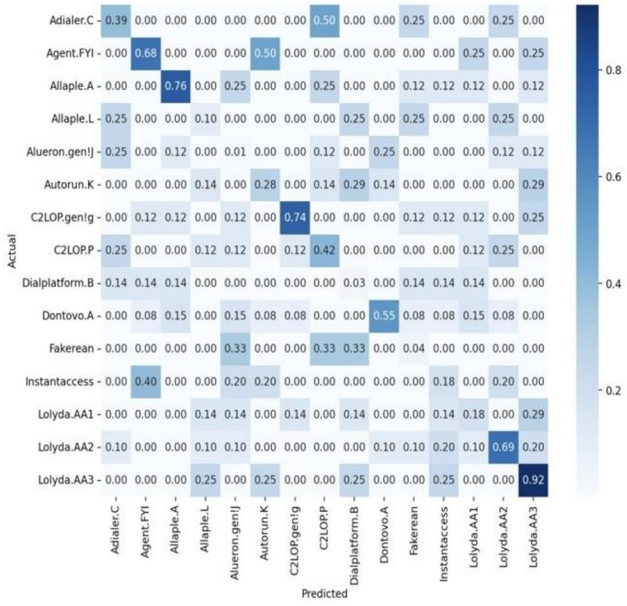


Fig 5: Confusion Matrix

To benchmark our findings, a comparative analysis between the suggested model and VGG-16 is carried out. The accuracy and loss graphs of the corresponding models serve as the foundation for this comparative analysis. The accuracy and loss graph of the suggested model are displayed in Fig. 6a. The accuracy and loss graph of the VGG-16 model is displayed in Fig. 6b.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

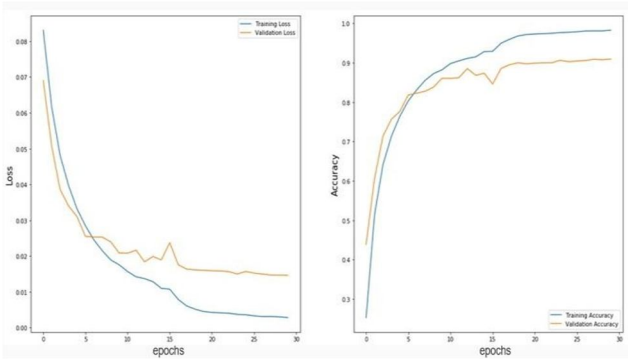


Fig. 6a: Custom model accuracy and loss graph (Yolov5)

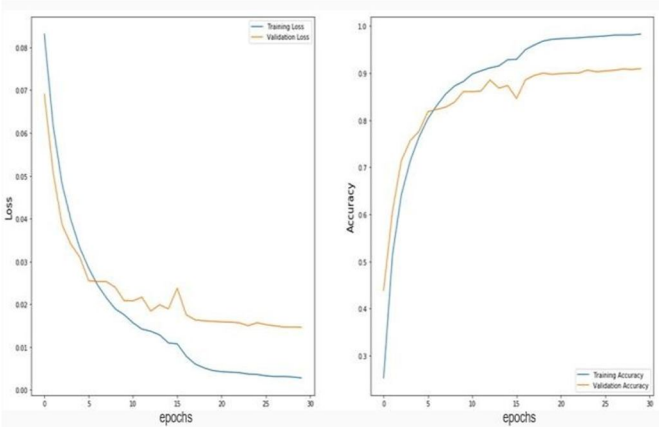


Fig. 6b: The VGG-16 Model's Accuracy and Loss graph

With an accuracy of 97.5%, the bespoke CNN model outperforms the 95.2% accuracy of the VGG16 model. As seen in the accompanying table, this exceptional performance is also evident in other measures including precision, recall, and F1-score.

Table 1: Custom CNN and VGG-16 accuracy comparison

Metric	Custom CNN	VGG16
Accuracy	97.50%	95.20%
Precision	97.20%	94.80%
Recall	97.80%	95.50%
F1-score	97.50%	95.10%

These findings suggest that the custom CNN model outperforms the VGG16 model in malware detection and classification.

VI. CONCLUSION

In order to improve malware detection, a bespoke Convolutional Neural Network (CNN) model was developed and evaluated, as this study demonstrates. With the use of the Malimg dataset—which includes malware images from 15 different families—the study highlights how effective customized deep learning architectures are in recognizing and categorizing harmful software. The comparison with the VGG16 model—a reputable deep learning model that has been pre-trained on the ImageNet dataset—offers important new information and advances cybersecurity.

1) *Better Performance of Custom CNN:* The accuracy of the custom CNN model was 97.5%, which was greater than that of the VGG16 model, which was 95.2%. This improved performance held true for F1-score, recall, and precision, among other criteria. These findings show that the customized CNN model is a reliable option for cybersecurity applications, as it is quite good at identifying and categorizing different kinds of malware.
2) *Efficiency in Computational Resources:* Compared to the VGG16 model, the bespoke CNN model demonstrated greater computational efficiency. Because it used less resources for inference and training, it is more suited for real-time malware detection systems where response speed and processing power are crucial.

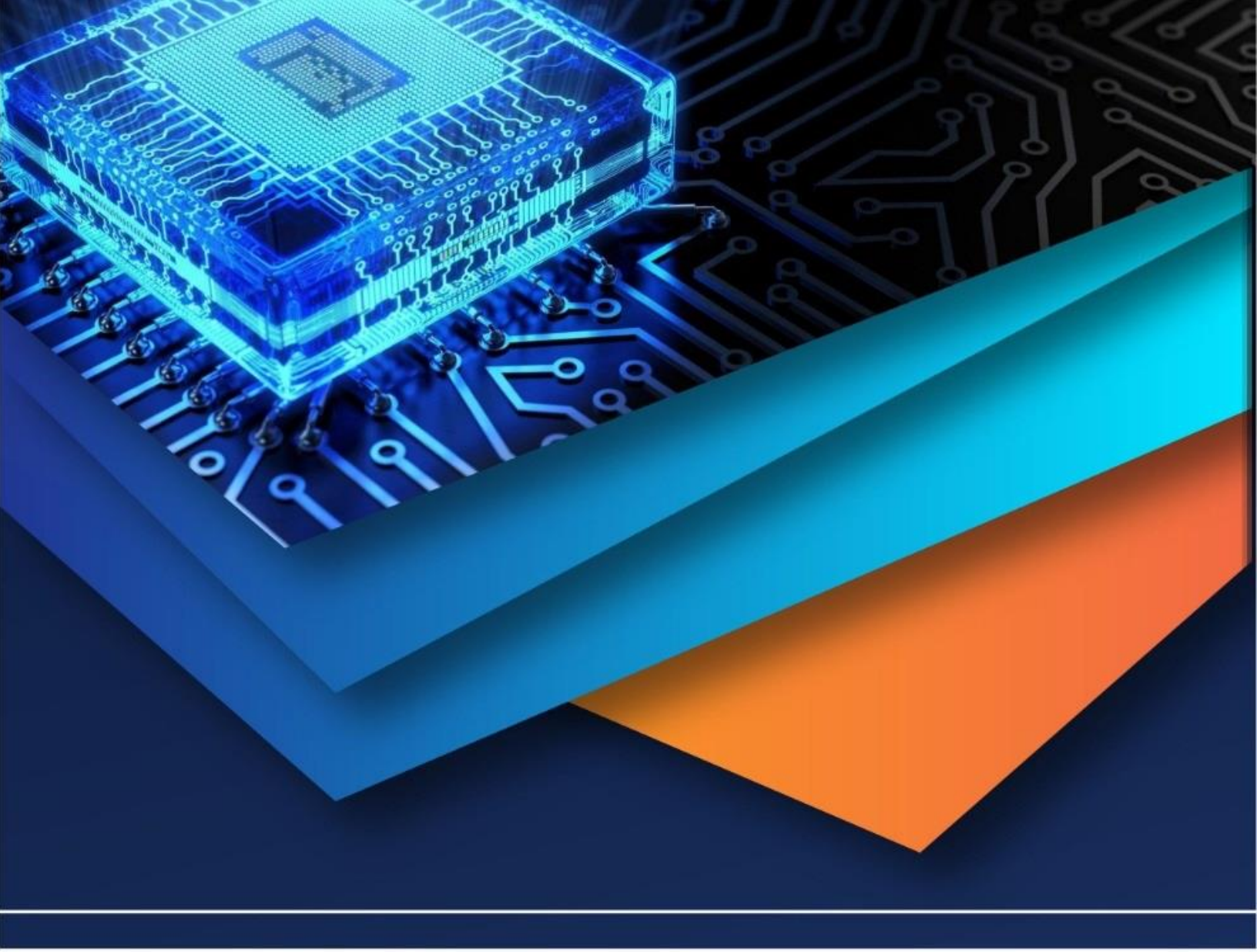
©IJRASET: All Rights are Reserved

3) *Difficulties & Misclassifications:* Although the custom CNN model performed well generally, it had trouble differentiating distinct malware families that had similar visual patterns. This implies that adding more features, such behavioural analysis, might improve the model's ability to discriminate.
4) *Enhancement of CNN Applications:* This study adds to the expanding corpus of information about using CNNs, in particular, and deep learning in cybersecurity. Through the demonstration of a bespoke CNN model's superiority over a pre-trained model such as VGG16, the study offers important new insights into the development and application of specialized neural network architectures for certain purposes including malware detection.

REFERENCES

[1] Hinton, G., LeCun, Y., & Bengio, Y. (2015). profound understanding. 521(7553), 436-444; Nature.
[2] Zisserman, A. & Simonian, K. (2014). Deep convolutional networks for large-scale picture recognition. The preprint arXiv is arXiv:1409.1556.
[3] Lapalme, G., and Sokolova, M. (2009). a methodical evaluation of classification task performance metrics. 45(4), 427–437 in Information Processing & Management.
[4] A. M. Saxe and associates (2015). Regarding the deep learning idea of information bottleneck. Preprint arXiv arXiv:1503.02406.
[5] J. Schimdhuber (2015). Overview of deep learning using neural networks. 85–117 in Neural Networks, 61.
[6] Liao, H. Y. M., Wang, C. Y., Bochkovskiy, A., & Liao, H. J. (2020). YOLOv5: An enhanced variant of the YOLO franchise. Preprint arXiv arXiv:2006.08258.
[7] Chollet, F., & colleagues (2015).GitHub: Keras. This link points to the https://github.com/keras-team/keras.
[8] Vincent, P., Bengio, Y., and Courville, A. (2013). A review and fresh insights on representation learning. IEEE Transactions on Machine Intelligence and Pattern Analysis, 35 (8), 1798-1828.
[9] Bengio, Y., Goodfellow, I., & Courville, A. (2016). MIT Press, Deep Learning.
[10] C. M. Bishop (2006). Machine learning and pattern recognition. Springer.
[11] In 2020, Gilbert, D., Mateu, C., Planes, J., & Vicens, R. The evolution, trends, and problems of machine learning research in the realm of malware detection and categorization. 153, 102526, Journal of Network and Computer Applications.
[12] Webster, G., Eckert, C., Zarras, A., and Kolosnjaji, B. (2016). Deep learning for malware system call sequence classification. Springer, Australia: Australasian Joint Conference on Artificial Intelligence.
[13] Stokes, J. W., and Huang, W. (2016). Mtnet: A multitask neural network designed to classify malware dynamically. The International Conference on Vulnerability Assessment and Intrusion Detection and Malware, Proceedings, Springer.
[14] Barker, J., Sylvester, J., R. Brandon, B. Catanzaro, & Nicholas, C., together with E. Raff (2017). detecting malware by devouring an entire exe. Thirty-First AAAI Conference on Artificial Intelligence Workshops.
[15] Manjunath, B. S., Jacob, G., Karthikeyan, S., and Nataraj, L. (2011). Visualization and automatic classification of malware pictures. The 8th International Symposium on Cyber Security Visualization Proceedings.
[16] Naseem, H., Safaei, B., Wassan, S., Alazab, M., & Zheng, Q. (2020). Vasan, D. IMCFN stands for image-based convolutional neural network architecture malware classification. IEEE Computer Society, 171, 107138.
[17] Raj, A., Singh, B., and Kumar, N. (2018). Detecting malware with deep learning techniques. Advanced Research in Computer and Communication Engineering, International Journal, 7(3), 220-224.
[18] Zhang, Liu, Huang, and Huang (2019). Deep convolutional and recurrent neural networks are used to classify malware. 13(1), 1–10, International Journal of Security and Its Applications.
[18] Dahl, G. E., Yu, D., & Stokes, J. W. (2013). Deng, L. neural networks and random projections for the large-scale categorization of malware. The IEEE ICASSP stands for IEEE International Conference on Acoustics, Speech, and Signal processing.
[19] Varuna Kumara, Ezhilarasan Ganesan, A novel approach to wastewater treatment control: a self-organizing fuzzy sliding mode controller. IAES International Journal of Artificial Intelligence (IJ-AI) Vol. 13, No. 3, September 2024, pp. 2796 - 2807 ISSN: 2252-8938, DOI: 10.11591/ijai.v13.i3.pp2796-2807

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)