



Enhanced Malware Classification Using Convolutional Neural Network with Feature Optimization

¹Mohd Saif, ²Dr. Satyendra Kurariya

M. Tech. Scholar, Department of Computer Science and Engineering, Madhyanchal
Professional University (MPU), Bhopal, India¹

Professor, Department of Computer Science and Engineering, Madhyanchal Professional
University (MPU), Bhopal, India²

Abstract

The rapid evolution of cyber threats and the continuous emergence of new malware variants have created significant challenges for conventional malware detection and classification systems. Traditional signature-based approaches are generally effective for previously identified malware but may fail to detect polymorphic, obfuscated, and previously unseen malware. Deep learning, particularly Convolutional Neural Networks (CNNs), provides an effective solution by automatically learning discriminative patterns from malware representations. However, the performance of CNN-based malware classification can be affected by redundant features, high-dimensional data, class imbalance, and computational complexity. Hence, this research work discloses a novel framework for identifying malware in the Android implemented applications with the help of DL (Deep Learning) methodologies such as enhanced CNN based DenseNET 169 and 201. The proposed approach was tested and trained using Malevis and Malimg Dataset. The results obtained from the experimental procedures demonstrate that the proposed deep learning framework outperforms the various traditional methods with an accuracy of 96.09% and 98.40%.

Keywords: - Malware Classification, Convolution Neural Network (CNN), DenseNET, Accuracy

1. INTRODUCTION

A concise history of the dubious programming gets started in the mid-1970s. Since the origination of the registering period, the presence of malware contamination has additionally denoted its impression. The main reported infection was named the Creeper Worm. It is a trial self-repeating plan that duplicated itself into a couple of far off frameworks with a message showed as "I'm the creeper, get me if possible". Toward the start of the 1980s, there came Elk Cloner, a boot-field infection that chiefly harmed Apply II PCs [1]. With this plain commencement, a monstrous field was sent off and subsequently after that the battle against the malware began and has not finished till now. Because of the seriousness and arising ill-conceived programming, the battle to preserve the organization framework has turned into a ceaseless interaction and a cyclic race for security experts. Specialists' guide to safeguard the organization security with novel innovations by improved malware which distinguishes new contaminations guards the disease from ruining the activity of the framework and overhauls their muddling approaches. With mechanical progression, malware issues grow themselves both in vertical and level ways. Malware extends upward through the expansion in volumes and numbers and develops evenly through its usefulness and types [2, 3].

The innovation development as the web, cell phones, virtual entertainment, IoT machines, etc., improves the chance of aggressors making novel and modern malware. In the current period,

numerous productive sorts of malware like crypto mining and ransomware arise alongside Locky and Cerber which holds the PC frameworks all through the world for recover while the crypto misfortune utilizes the casualty's organization influence to exhume crypto without the information on the client. However most malware assailants target PC frameworks, there is a transcendent ascent in the danger of IoT (Web of Things) and versatile malware. By Symantec, variety in versatile malware got further developed by 54% in the year 2017 though IoT dangers have spread to 600%. A large portion of the potential DDoS dangers were because of the Mirai botnet and its sorts which act as a vehicle for the assault. To deal with the issue of malware, digital safeguard is made by security expert and specialist's continually [4].

One of the critical components in the process is assurance given at the endpoint. The endpoint insurance delivers a gathering of safety programs that incorporates, yet isn't limited to, URL sifting, hostile to spam, firewall, sandboxing and email security. Especially, hostile to malware programming gives the safeguard layer at the last. AV (AntiVirus) motors introduced toward the finish of the gadget are dependable to eliminate, distinguish and forestall dubious programming. Routinely, AV arrangements exclusively rely upon heuristic-based and signature-based strategies. Heuristics are the arrangement of rules laid out by specialists subsequent to finishing the examination of the way of behaving of malware while a mark is a methodology or hash which extraordinarily perceives the specific malware [5, 6].

The two calculations need the malware that should be examined at first prior to characterizing the heuristics and rules. A definitive objective of leading malware investigation is to reveal data in regards to the reason, conduct and qualities of the gave part of the product. The examination is led in two strategies; they are static investigation and dynamic examination. The static examination incorporates exploring an executable strategy without being executed. On other hand, the powerful examination incorporates researching the way of behaving of the implementable through execution. The two techniques include their benefits and inconveniences. The presentation of static examination is viewed as quicker yet when the malware is carried out effectively with the assistance of the code confusion technique, this prompts dodging identification. Interestingly, the code confusion technique and polymorphic malware intensely dodge dynamic examination since they investigate and screen the runtime execution of a program [7, 8].

2. MALWARE DETECTION

Malware detection is the process of identifying malicious software that can compromise computer systems, mobile devices, networks, and other digital platforms. Malware includes different types of threats such as viruses, worms, Trojans, ransomware, spyware, rootkits, and botnets. The primary objective of malware detection is to distinguish malicious programs from legitimate software and prevent unauthorized activities such as data theft, system damage, unauthorized access, and service disruption. With the rapid increase in sophisticated and frequently modified malware variants, developing accurate and reliable automated detection systems has become an important research area in cybersecurity.

Traditional malware detection primarily uses signature-based detection, where a unique signature or pattern is generated from previously identified malware and compared with a new file. Although this approach can efficiently detect known malware, it has difficulty identifying new, polymorphic,

metamorphic, and obfuscated malware. To overcome these limitations, static analysis and dynamic analysis are widely used. Static analysis examines characteristics of a program without executing it, such as byte sequences, opcodes, strings, imported functions, and executable headers. Dynamic analysis, in contrast, observes malware behavior during execution, including API calls, system calls, network communication, file modifications, and memory activities.

3. PROPOSED METHODOLOGY

The proposed malware detection system examines the PE/.apk (Windows/Android) file to see if it is malware or cleanware. The input file is analyzed using any of the malware analysis techniques such as static analysis, dynamic analysis, vision based analysis, and hybrid analysis. If the malware file is obfuscated or packed, it is deobfuscated or unpacked to obtain the original malware sample and then the file is subjected to the analysis phase. The static/ dynamic/ vision-based/ hybrid features are extracted at the end of the malware analysis task. Based on feature representations, machine learning techniques are used to identify binaries as clean or malicious. The malware database is updated regularly with new or modified malware signatures.

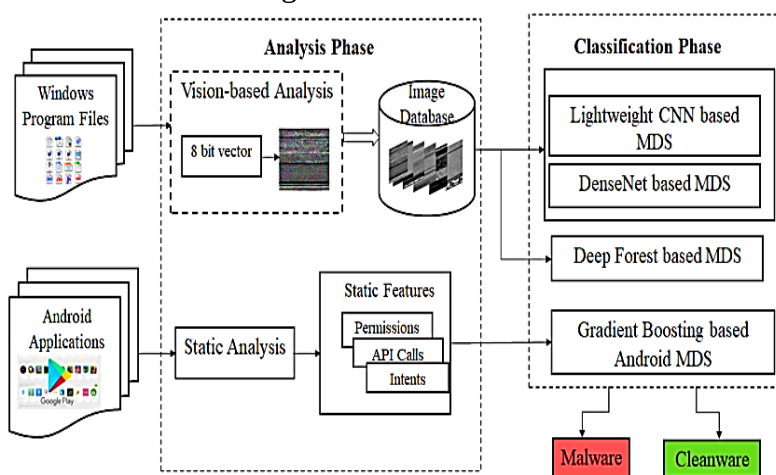


Fig. 1: Proposed Malware Detection

3.1 DenseNET

The DenseNet-based MDS is simpler to prepare since it requires fewer boundaries. This technique embraces include reusability and requires less calculations to accomplish higher exactness. The size of the info executable records is at first factor. CNNs can't acknowledge pictures of changed goals since they are made principally out of completely associated layers that have a predefined number of preparing loads. Accordingly, the info paired pictures of various sizes are resized into equivalent estimated pictures of 64×64 aspects. Utilizing the closest addition approach, the paired pictures are resized. The DenseNet model is utilized to remove and characterize highlights from twofold pictures. A convolutional layer, a maximum pooling layer, 4 Thick Convolution blocks, and 4 progress layers make up the proposed model. By taking care of the info pictures into the primary Convolution layer, the model is prepared. The proposed technique utilizes DenseNet to extricate highlights from double pictures and afterward prepares the DenseNet with those elements. The thick layers get exact data from paired pictures. In the wake of going through these layers, the result highlight maps are sent to the Worldwide Normal Pooling (Hole) layer. A Completely Associated (FC) layer follows Hole, which groups malware and cleanware tests into their particular classes. To manage imbalanced

information issues and get significant execution in malware order, the proposed strategy utilizes a decent re-weighted misfortune capability in the end layer of the DenseNet model.

3.2 Lightweight CNN

The convolutional layer is the most important component of a CNN because it is where the main processing takes place. It usually requires input data, a filter, and a feature map. There are three elements in the input data that represent RGB in an image: height, width, and depth. The feature detector, typically called a kernel or a filter, traverses over the receptive fields of the image, searching for the presence of the feature. The feature detector is a two-dimensional (2D) weighted array that indicates a portion of the image. The filter size, also known as the receptive field size, is commonly a 3×3 matrix. After that, the filter is applied to a portion of the image, and a dot product between the input pixels and the filter is computed. Then, the dot product is placed into an output array. The filter then advances by a stride, and the procedure is repeated till the kernel has passed throughout the image. A feature map, also known as an activation map or a convolved feature, is the final result of a sequence of dot products from the input and the filter.

4. SIMULATION RESULTS

Dataset

In order to classify malware, PE files are converted to byteplot images. For this purpose, the Maling dataset and the Malevis dataset are the most widely used examples of data. In contrast to the Malevis dataset, which has a rather even distribution of classes, the Maling dataset has a skewed class distribution. The two datasets are combined to create a new dataset with 15 classes total (10 from Malevis and 5 from Maling).

Preprocessing

Image size is 64 for height and width, perform interpolation, apply image data generator with operations rotation, height and width shift, zoom, horizontal and vertical flip. Convert all images into numpy array and perform pixel normalization with /255.

Data splitting

Split data into ratio 80:20 total images were 13700 with 31 classes.

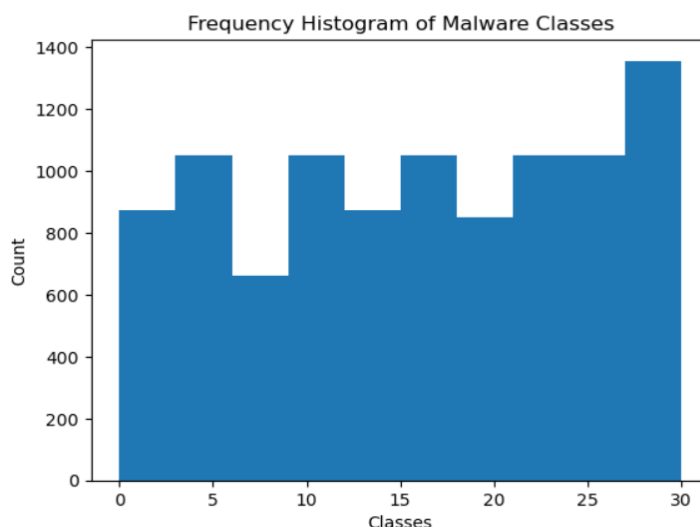


Fig. 2: Malware Classes

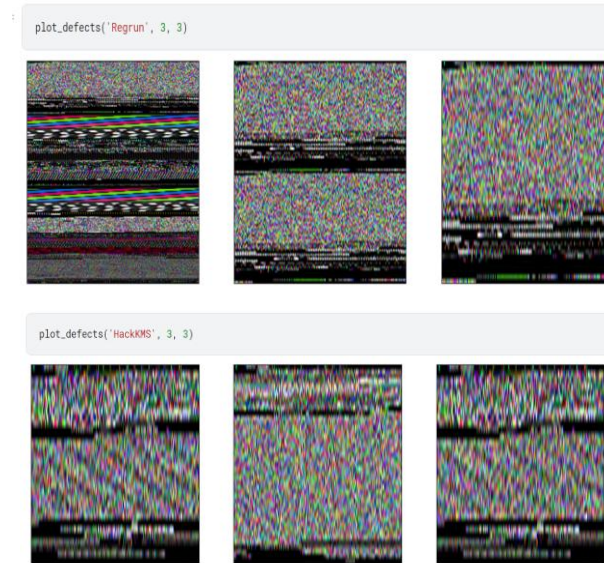


Fig. 3: Malware Classification

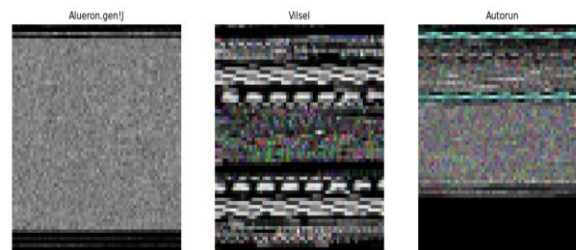


Fig. 4: Defect of Malware

Processed images
 Modeling

In this work used 2 models based on transfer learning like DenseNet169 and 201

Table 1: Parameters

Model	Densenet
Type	Transfer learning
Models	169,201
Layers	2DCNN, Input, Output
Pooling	Global
Normalization	Batch
Dropout	50%
Epochs	100
Activation	Relu, Softmax
Optimizer	adam
Loss	Categorical Cross Entropy
Metrics	Accuracy, Loss

Layer (type)	Output Shape	Param #
input_2 (InputLayer)	[(None, 64, 64, 3)]	0
conv2d (Conv2D)	(None, 64, 64, 3)	84
densenet169 (Functional)	(None, None, None, 1664)	12642880
global_average_pooling2d (GlobalAveragePooling2D)	(None, 1664)	0
batch_normalization (Batch Normalization)	(None, 1664)	6656
dropout (Dropout)	(None, 1664)	0
dense (Dense)	(None, 256)	426240
batch_normalization_1 (Batch Normalization)	(None, 256)	1024
dropout_1 (Dropout)	(None, 256)	0
root (Dense)	(None, 31)	7967

=====
 Total params: 13,084,851
 Trainable params: 12,922,611
 Non-trainable params: 162,240

Fig. 5: Model Summary

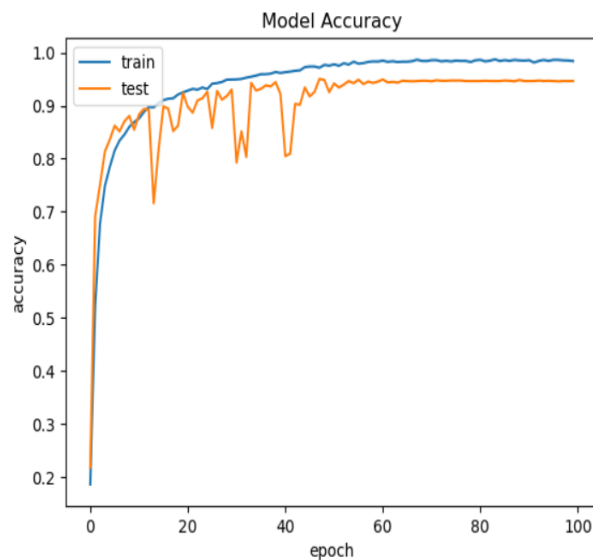


Fig. 6: Accuracy of Train and Test Data

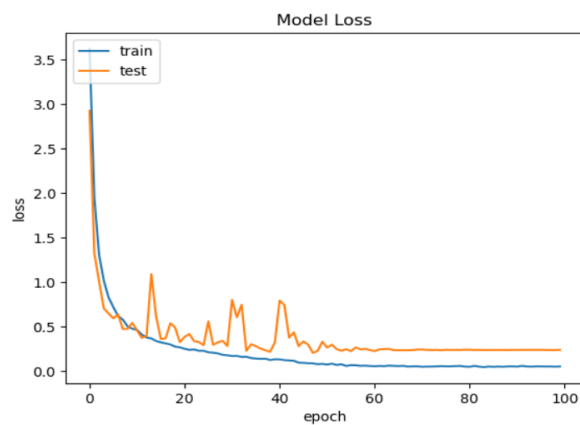


Fig. 7: Loss of Train and Test Data

The performance comparison of the deep learning models shows that DenseNet169 achieved an accuracy of 98.40% with a corresponding loss of 0.05. This indicates that DenseNet169 was able to learn highly discriminative features from the malware dataset and classify the malware samples with high accuracy. The very low loss value further suggests that the model produced predictions that were close to the actual class labels. In comparison, DenseNet201 achieved an accuracy of 96.09% with a loss of 0.13. Although DenseNet201 also demonstrated strong classification performance, its lower accuracy and higher loss indicate that it was comparatively less effective than DenseNet169 for the given malware classification task. Overall, the results indicate that DenseNet169 provided better classification performance, achieving a 2.31 percentage-point improvement in accuracy and a lower loss of 0.08 compared with DenseNet201.

Table II: Results

Models	Accuracy	Loss
DenseNet169	98.40	0.05
DenseNet201	96.09	0.13

5. CONCLUSION

The comparative analysis of DenseNet169 and DenseNet201 demonstrates the effectiveness of deep learning models for malware classification. DenseNet169 achieved an accuracy of 98.40% with a loss of 0.05, whereas DenseNet201 obtained 96.09% accuracy with a loss of 0.13. The higher accuracy and lower loss of DenseNet169 indicate that it learned more effective feature representations for distinguishing malware classes in the evaluated dataset. The results also show that increasing the network depth does not necessarily guarantee improved classification performance, as model architecture and feature representation have a significant influence on the final results.

Overall, the analysis establishes DenseNet169 as the model with the stronger measured performance for the given experimental setup. Further improvement can be investigated through feature optimization, data augmentation, hyperparameter optimization, transfer learning, and hybrid deep learning architectures. Future work can also evaluate the proposed approach on larger and more diverse malware datasets, including newly emerging and obfuscated malware variants, to examine its robustness and generalization capability in real-world cybersecurity environments.

REFERENCES

- [1] M. Alshoulie and A. Mehmood, "Deep Learning Approaches for Malware Detection: A Comprehensive Review of Techniques, Challenges, and Future Directions," *IEEE Access*, vol. 13, pp. 118652–118677, 2025, doi: 10.1109/ACCESS.2025.3582875.
- [2] P. Maniriho, A. N. Mahmood, and M. J. M. Chowdhury, "A Survey of Recent Advances in Deep Learning Models for Detecting Malware in Desktop and Mobile Platforms," *ACM Computing Surveys*, vol. 56, no. 6, Art. 145, pp. 1–41, 2024, doi: 10.1145/3638240.
- [3] H. Wang, B. Cui, Q. Yuan, R. Shi, and M. Huang, "A Review of Deep Learning Based Malware Detection Techniques," *Neurocomputing*, vol. 598, Art. 128010, 2024, doi: 10.1016/j.neucom.2024.128010.
- [4] A. Bensaoud, J. Kalita, and M. Bensaoud, "A Survey of Malware Detection Using Deep Learning," *Machine Learning with Applications*, vol. 16, Art. 100546, 2024, doi: 10.1016/j.mlwa.2024.100546.



- [5] A. R. Redhu, P. Choudhary, K. S. Srinivasan, and T. K. Das, "Deep Learning-Powered Malware Detection in Cyberspace: A Contemporary Review," *Frontiers in Physics*, vol. 12, 2024, Art. 1349463, doi: 10.3389/fphy.2024.1349463.
- [6] Omer Aslan and Abdullah Asim Yilmaz, "A New Malware Classification Framework Based on Deep Learning Algorithms", *IEEE Access* 2021.
- [7] K. Bhargavi N. Vadivelan S. Kodati and M. Nalini "M. Secure internet of battlefield from malicious software using deep eigenspace learning" In *AIP Conference Proceedings* vol. 2405 no. 1 pp. 030002 2022.
- [8] M. D. AlGarni R. AlRoobaea J. Almotiri S. S. Ullah S. Hussain and F. Umar "An efficient convolutional neural network with transfer learning for malware classification" *Wireless Communications and Mobile Computing* pp. 1-8 2022.
- [9] S. Kumar "MCFT-CNN: Malware classification with fine-tune convolution neural networks using traditional and transfer learning in Internet of Things" *Future Generation Computer Systems* vol. 125 pp. 334-351 2021.
- [10] Ö. Aslan and R. Samet, "A comprehensive review on malware detection approaches," *IEEE Access*, vol. 8, pp. 6249–6271, Jan. 2020.
- [11] R. Komatwar and M. Kokare, "A survey on malware detection and classification," *J. Appl. Secur. Res.*, pp. 1–31, Aug. 2020.
- [12] A. A. Yilmaz, M. S. Guzel, E. Bostanci, and I. Askerzade, "A novel action recognition framework based on deep-learning and genetic algorithms," *IEEE Access*, vol. 8, pp. 100631–100644, 2020.
- [13] A. A. Yilmaz, M. S. Guzel, I. Askerbeyli, and E. Bostanci, "A vehicle detection approach using deep learning methodologies," in *Proc. Int. Conf. Theor. Appl. Comput. Sci. Eng.*, Nov. 2018, pp. 64–71.
- [14] A. A. Yilmaz, M. S. Guzel, I. Askerbeyli, and E. Bostanci, "A hybrid facial emotion recognition framework using deep learning methodologies," in *Human-Computer Interaction*, T. Özseven, Eds. Hauppauge, NY, USA: Nova Science Publishers, 2020.
- [15] S. A. Roseline, S. Geetha, S. Kadry, and Y. Nam, "Intelligent vision-based malware detection and classification using deep random forest paradigm," *IEEE Access*, vol. 8, pp. 206303–206324, 2020.