Malware Classification Using CNN Model: A Comparative Study

BY

IRAM AHMMED
ID: 201-15-3217

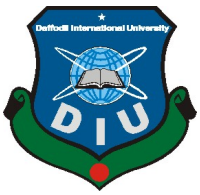
This Report Presented in Partial Fulfillment of the Requirements for the Degree of Bachelor of Science in Computer Science and Engineering

Supervised By

Amatul Bushra Akhi
Assistant Professor
Department of CSE
Daffodil International University

Co-Supervised By

Nusrat Khan
Lecturer
Department of CSE
Daffodil International University



DAFFODIL INTERNATIONAL UNIVERSITY

DHAKA, BANGLADESH

JULY 2024

APPROVAL

This Project titled "**Malware Classification Using CNN Model: A Comparative Study**", submitted by Iram Ahmmed to the Department of Computer Science and Engineering, Daffodil International University, has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of B.Sc. in Computer Science and Engineering and approved as to its style and contents. The presentation has been held on July 13, 2024.

BOARD OF EXAMINERS

Dr. Sheak Rashed Haider Noori (SRH) — **Chairman**
Professor & Head
Department of CSE
Faculty of Science & Information Technology
Daffodil International University

Dr. Md. Zahid Hasan (ZH) — **Internal Examiner**
Associate Professor
Department of CSE
Faculty of Science & Information Technology
Daffodil International University

Mr. Md. Mohammad Masum Bakaul (MB) — **Internal Examiner**
Sr. Lecturer
Department of CSE
Faculty of Science & Information Technology
Daffodil International University

Dr. Md. Zulfiker Mahmud (ZM) — **External Examiner**
Professor
Department of CSE
Jagannath University

DECLARATION

I hereby declare that this project has been done by me under the supervision of **Amatul Bushra Akhi, Assistant Professor, Department of CSE,** Daffodil International University. I also declare that neither this project nor any part of this project has been submitted elsewhere for award of any degree or diploma.

Supervised by:

Amatul Bushra

Amatul Bushra Akhi
Assistant Professor
Department of CSE
Daffodil International University

Co-Supervised by:

Nusrat Khan

Nusrat Khan
Lecturer
Department of CSE
Daffodil International University

Submitted by:

Iram Ahmmed

IRAM AHMMED
ID: 201-15-3217
Department of CSE
Daffodil International University

ACKNOWLEDGEMENT

First I express my heartiest thanks and gratefulness to almighty God for His divine blessing makes it possible to complete the final year project successfully.

I am really grateful and wish my profound indebtedness to **Amatul Bushra Akhi, Assistant Professor**, Department of CSE Daffodil International University, Dhaka. Deep Knowledge & keen interest of our supervisor in the field of "*Deep Learning*" to carry out this project. Her endless patience ,scholarly guidance ,continual encouragement , constant and energetic supervision, constructive criticism , valuable advice ,reading many inferior drafts and correcting them at all stages have made it possible to complete this project.

I would like to express our heartiest gratitude to **Dr. Sheak Rashed Haider Noori, Professor, and Head, Department of CSE,** Department of CSE, for his kind help to finish my project and also to other faculty members and the staff of CSE department of Daffodil International University.

I would like to thank my entire course mate in Daffodil International University, who took part in this discussion while completing the course work.

Finally, I must acknowledge with due respect the constant support and patients of my parents.

ABSTRACT

The goal of this research is to improve malware classification in the rapidly evolving field of cybersecurity by applying state-of-the-art deep learning models, namely ResNet-50 v2, InceptionV3, VGG16, and DenseNet-121. Taking advantage of these architectures' strong points, this work aims to increase the efficiency and accuracy of identifying malicious software using a dataset consisting of nine different malware types. Each model's unique features are carefully examined, and their unique contributions to classification accuracy within the intricate malware taxonomy are examined. Through a thorough analysis, the research aims to shed light on the subtle nuances of malware behavior and features, equipping cybersecurity professionals with advanced tools for threat identification and mitigation. The results have practical implications for the creation of more resilient and adaptable security measures in the ongoing fight against developing cyber threats, in addition to contributing to the scholarly discourse on malware classification. Among other CNN models used in this study ResNet-50 v2 scored the best accuracy of 86.9%. After that VGG16 and DenseNet-121 showed promising results with 80.4% and 82.5% accuracy. Traditional approaches like Multilayer Perceptron, Random Forest, Long Short-Term Memory, K-Nearest Neighbor were also added to compare between these and Convolutional Neural Network models to find better solutions to the malware problem in our daily life. Random Forest scored 79% accuracy which being the highest accuracy among the traditional approaches. Surprisingly Multilayer Perceptron achieved 94.5% model training accuracy but failed to perform accordingly while testing the model and scored only 76%. Convolutional Neural Network outperforming every other traditional approach was an achievement of this study and Convolutional Neural Network proved to be the better solution than other approaches in the case of image classification of malware dataset.

TABLE OF CONTENTS

CONTENTS	PAGE
Board of examiners	1
Declaration	2
Acknowledgements	3
Abstract	4
CHAPTER	
CHAPTER 1: Introduction	**9-12**
1.1 Overview	9
1.2 Motivation	10
1.3 Rationale of the Study	10
1.4 Research Questions	11
1.5 Expected Output	11
1.6 Project Management and Finance	12
1.7 Report Layout	12
CHAPTER 2: Background	**13-18**
2.1 Preliminaries/Terminologies	13
2.2 Related Works	14
2.3 Comparative Analysis and Summary	15
2.4 Scope of the Problem	16
2.5 Challenges	17

CHAPTER 3: Research Methodology **19-28**

3.1 Research Subject and Instruments 19

3.2 Dataset Utilized 20

3.3 Statistical Analysis 21

3.4 Proposed Methodology 22

3.5 Implementation Requirements 27

CHAPTER 4: Experimental Results and Discussion **29-39**

4.1 Experimental Setup 29

4.2 Experimental Results & Analysis 30

4.3 Discussion 39

CHAPTER 5: Impact on Society, Environment and Sustainability **40-42**

5.1 Impact on Society 40

5.2 Impact on Environment 41

5.3 Ethical Aspects 41

5.4 Sustainability Plan 42

CHAPTER 6: Summary, Conclusion and Implication for Future Research **43-44**

6.1 Summary of the Study 43

6.2 Conclusions 43

6.3 Implication for Further Study 44

REFERENCES **45-46**

LIST OF FIGURES

FIGURES	PAGE NO
Figure 3.2.1: Images of Malwares from 9 Classes	20
Figure 3.4.1: Proposed Methodology	22
Figure 3.4.2: The basic CNN architecture	23
Figure 3.4.3: VGG 16 architecture	24
Figure 3.4.4: DenseNet-121 architecture	25
Figure 3.4.5: ResNet50 V2 architecture	25
Figure 3.4.6: Inception-V3 architecture	26
Figure 4.2.1: Confusion Matrix of Inception-V3	34
Figure 4.2.2: Confusion Matrix of VGG-16	34
Figure 4.2.3: Confusion Matrix of ResNet-50 V2	35
Figure 4.2.4: Confusion Matrix of DenseNet-121	35
Figure 4.2.5: Accuracy and loss of Inception-V3	36
Figure 4.2.6: Accuracy and loss of VGG-16	36
Figure 4.2.7: Accuracy and loss of ResNet-50 V2	36
Figure 4.2.8: Accuracy and loss of DenseNet-121	37
Figure 4.2.9: ROC curve of Inception-V3	37
Figure 4.2.10: ROC curve of VGG-16	38
Figure 4.2.11: ROC curve of ResNet-50 V2	38
Figure 4.2.12: ROC curve of DenseNet-121	39

LIST OF TABLES

TABLES	PAGE NO
Table 3.3.1: Number of Images of Each Classes Table	21
Table 3.4.1: A comparison of popular CNN architectures (M= Million)	24
Table 4.2.1: Accuracy Table for Model and Training	30
Table 4.2.2: Classification report of Inception-V3	31
Table 4.2.3: Classification report of VGG-16	31
Table 4.2.4: Classification report of ResNet-50 V2	32
Table 4.2.5: Classification report of DenseNet-121	33

LIST OF ACRONYMS

CNN	Convolutional Neural Network
ML	Machine Learning
MLP	Multilayer Perceptron
LSTM	Long Short-Term Memory
KNN	K-Nearest Neighbor

CHAPTER 1

Introduction

1.1 Overview

Convolutional Neural Networks have signified a paradigm shift in the landscape of malware recognition and classification in the dynamic field of cybersecurity. As cyber threats become more sophisticated and diverse, the need for powerful and adaptable protection measures becomes more pressing. This work investigates the transformative effects of CNNs on the complex problem of malware identification, utilizing their intrinsic capacity to recognize subtle patterns and hierarchical structures. With an ever-expanding number of malware variants, CNNs are potent tools in the armory of cybersecurity specialists, providing unrivaled ability to distinguish malicious code from legitimate software. This introduction lays the groundwork for a more in-depth examination of how CNNs deal with the multiple issues provided by malware, paving the way for a better understanding of their impact on the growth of cybersecurity methods. The goal of this voyage is to uncover the subtle interplay between CNNs and malware identification, revealing light on their efficacy, limitations, and broader implications for the continuing cyber arms race.

The continual growth of malicious software provides an ever-increasing challenge to digital defenses in the complex world of cybersecurity. This introduction lays the groundwork for a thorough examination of nine distinct malware strains: Gatak, Lollipop, Kelihos_ver3, Tracur, Kelihos_ver1, Simda, Vundo, Ramnit, and Obfuscator.ACY. Each of these malware varieties reflects a distinct aspect of cyber threats, with distinct tactics, propagation processes, and possible effects on digital systems. From Gatak's elusive methods to Lollipop's insidious nature to Kelihos_ver3's polymorphic complexity, these malware variants collectively illustrate the gamut of issues encountered by cybersecurity practitioners. Tracur, Kelihos_ver1, Simda, Vundo, Ramnit, and Obfuscator.ACY are all available to add this complex ecosystem, each leaving its own impact on the cybersecurity narrative.

This research looks into the subtleties of different malware kinds, seeking to untangle their objectives and the possible hazards they bring to digital ecosystems. As we navigate through the intricacies of Gatak's stealth, Lollipop's deception, Kelihos_ver3's polymorphism, Tracur's evasion, Kelihos_ver1's persistence, Simda's propagation, Vundo's resilience, Ramnit's multifaceted attacks, and Obfuscator.ACY's code obfuscation, the objective is to not only comprehend the individual characteristics but also to collectively glean insights into the broader landscape of cybersecurity threats.

Understanding the complexities of these malware variants lays the groundwork for a more informed and proactive approach to safeguarding digital infrastructures against cyber adversaries' ever-adaptive tactics.

1.2 Motivation

In the pursuit of safeguarding the digital environment, the motivation behind this research comes from the ever-evolving dangerous threat posed by malicious softwares. The exponential growth and complication of malware require innovative and powerful solutions for detection and classification. By exploring into the surface of Convolutional Neural Networks (CNNs), this study seeks to apply the power of deep learning to enhance the accuracy and efficiency of malware classification. The motivation lies in the imperative need to stay one step ahead of cyber threats, providing the cybersecurity community and the digital environment with advanced tools to detect and mitigate the impact of diverse malware dominance. Through the exploration of cutting-edge technologies, this research endeavors to contribute to the ongoing efforts to fortify digital ecosystems against the relentless evolution of malicious entities.

1.3 Rationale of the Study

The rationale of this study is rooted in the escalating challenges posed by the dynamic landscape of malicious software. As technology advances, so do the complexities of cyber threats, necessitating innovative and adaptive approaches to cybersecurity. The increasing diversity and sophistication of malware demand a deeper understanding and more effective tools for detection and classification. By employing Convolutional Neural Networks (CNNs), this research aims to address the limitations of traditional methods and enhance the accuracy and efficiency of malware classification. The study recognizes the urgency of staying ahead of evolving threats to safeguard digital ecosystems. Through the exploration of advanced technologies, the goal is to contribute valuable insights and practical solutions that fortify the resilience of cybersecurity measures in the face of an ever-changing threat landscape.

1.4 Research Questions

How does the performance of Convolutional Neural Networks compare to traditional methods in the accurate classification of diverse malware strains?
What impact does the size and diversity of the dataset have on the CNN model's ability to generalize and classify previously unseen malware samples?
How do different architectural configurations of Convolutional Neural Networks influence their effectiveness in detecting and classifying specific types of malware?
To what extent can ensemble learning techniques be applied to enhance the performance of CNNs in malware classification across different domains or datasets?
How resilient are CNN-based malware classification models to adversarial attacks, and what strategies can be employed to improve their robustness in real-world scenarios?

1.5 Expected Output

I expect to obtain accurate classification results on the malware dataset, quantifying the model's performance through metrics such as accuracy, precision, recall, and F1-score. These metrics will provide insights into the overall effectiveness of the CNN models in distinguishing between the nine distinct malware classes. A comparative analysis will be conducted to assess the performance of the selected CNN architecture against each other. This analysis aims to highlight the advantages of utilizing deep learning approaches in malware classification, emphasizing factors such as computational efficiency, scalability, and classification accuracy. The study will investigate the generalization capability of the CNN models across diverse datasets and malware strains. I expect the models to demonstrate adaptability to variations in the data distribution, reflecting their potential for real-world deployment. Through experimentation with different architectural configurations, we aim to identify the optimal settings for the CNN models. This includes exploring variations in model depth, filter sizes, and pooling strategies to enhance classification performance. The CNN models will be analyzed for their robustness to adversarial attacks. I expect to identify potential vulnerabilities and propose strategies to enhance the models' resilience in the face of sophisticated threats. Results will be interpreted and visualized to provide a clear understanding of the model's strengths and weaknesses. Visualization techniques will aid in conveying complex information in an accessible manner.

1.6 Project Management and Finance

Project initiation involved a meticulous planning phase where I defined the scope, objectives, and deliverables of our malware classification research project. Clear research objectives provided a roadmap for me, aligning my efforts with the goal of enhancing cybersecurity through advanced malware detection. Effective communication with my supervisor and co-supervisor was a key to my research success. Necessary meetings, status updates facilitated seamless information exchange. This open communication culture fostered a work friendly environment, enabling me to address challenges promptly and share valuable insights. The dynamic landscape of cybersecurity demands a robust risk management strategy. Throughout the project lifecycle, I conducted risk assessments to identify potential problems and devised contingency plans. This proactive approach allowed me to adapt to unforeseen challenges, ensuring that the project remained on track despite evolving circumstances. A well-defined project schedule with clear milestones served as a roadmap for progress tracking. Adhering to timelines was crucial in maintaining momentum and ensuring that each phase of the project contributed to the overall success. Regular evaluations against milestones provided insights into the project's progress and allowed for necessary adjustments. Efficient resource allocation was paramount in optimizing our project's efficiency. No financial funds were aided. Thorough documentation of project processes, methodologies, and outcomes was a priority. This documentation not only served as a reference for the current work but also facilitated knowledge transfer for future works.

1.7 Report Layout

Chapter 1: Introduction.: Contains an overview of the goals, objectives, and research questions of the study.
Chapter 2: Literature Review: Highlights gaps in the field by analyzing previous studies.
Chapter 3: Methodology explains research procedures, techniques, and data collection techniques.
Chapter 4: Result Analysis and visualization using tables, figures, and charts.
Chapter 5: Ethical and Social Aspects: Permission, ethical concerns, data privacy and potential harm are discussed in this section.
Chapter 6: Conclusion summarizes major findings, analyzing research objectives, and rooms for future work.

CHAPTER 2

Background

2.1 Preliminaries/Terminologies

In this study, several key terminologies and concepts are essential to establishing a solid foundation for understanding the malware classification using Convolutional Neural Networks (CNNs).

Malware: Malware, a contraction of "malicious software," embodies a broad category of intrusive programs designed to compromise the integrity, confidentiality, or availability of computer systems. The scope of malware encompasses various forms, including viruses, worms, trojans, and ransomware. In the context of our research, the term "malware" pertains specifically to the diverse set of digital threats examined within our dataset.

Convolutional Neural Network (CNN): The most important part of our methodology, Convolutional Neural Networks (CNNs) represent a specialized class of deep neural networks renowned for their prowess in image recognition tasks. Comprising convolutional layers that learn hierarchical representations from input data, CNNs are well-suited for the nuanced patterns inherent in malware classification.

Classification: The complexity of our research lies in the process of classification, where malware samples are systematically categorized into predefined classes based on discernable characteristics. This entails training a CNN model to effectively distinguish between the nine distinct malware classes present in our dataset.

Accuracy, Precision, Recall, and F1-Score: To quantitatively assess our CNN models, we employ various metrics, including accuracy, precision, recall, and F1-score. These metrics collectively gauge the model's performance, providing insights into its ability to accurately classify instances while considering aspects such as false positives and false negatives.

Feature Engineering: In the pursuit of optimal model performance, we explore feature engineering techniques. This involves selecting, transforming, or creating features from raw data to refine the representation of malware characteristics within our CNN-based models.

By elucidating these terminologies within the specific context of our research, we lay the groundwork for a comprehensive exploration of CNNs in malware classification, fostering a nuanced understanding of the methodologies employed and their implications for cybersecurity.

2.2 Related Works

As malware developers are updating malwares continuously, fighting against these harmful malwares have become more challenging. The literature review focuses on multiple pathways to classifying and detecting malwares. This review highlights recent research that uses deep learning or machine learning.

Dib et al. [1] proposed a model that analyzed 70,000 recently detected IoT malware samples and achieved 99.78% accuracy. Their proposed model is called a multi-dimensional classification approach using Deep Learning (DL) architectures which outperforms conventional single-level classifiers. Pascanu et al. [2] used ESN for the recurrent model, Max-Pooling for non-linear sampling, and logistic regression for the final classification and achieved 98.4% accuracy. Kalash et al. [3] proposed a deep learning approach (cnn) that performed significantly better than traditional approaches. Lad et al. [4] developed a Hybrid CNN+SVM model which scored 99.59% accuracy. Also their CNN+L2-SVM outperformed other proposed models. Vasan et al. [5] found that Image-based Malware Classification using Ensemble of CNNs (IMCEC) is effective against packed and unpacked malwares that results in 99.50% accuracy. Gibert et al. [6] discovered that 3 convolutional layers and 2 densely-connected layers outperformed other methods and the best result was 94.64%. Pant et al. [7] developed a custom CNN model that provides better result than VGG16 and ResNet-18 while scoring 98.7%. Awan et al. [8] used spatial attention and convolutional neural network which achieved 97.68% accuracy based on the VGG-19 model. Dang et al. [9] found that combining biLSTM, embedding, and CNN model results a huge improvement than previous approaches and gaining 94.32% accuracy. Kumar et al. [10] developed MCFT-CNN based on ResNet50 that can perform consistently well on larger datasets than usual, having 98.63% accuracy. Tekerek et al. [11] developed a hybrid method based on B2IMG and CNN which performs well on two individual datasets gaining the highest accuracy of 99.86%. Ahmed et al. [12] used an approach based on InceptionV3 that achieved 98.76% classification accuracy and its performance is better than machine learning models like LSTM and LR. Chaganti et al. [13] proposed a CNN model about comparative performance analysis that

performed well on various datasets gaining the highest 97% accuracy. Narayanan et al. [14] found out that their proposed ensemble model based on cnn achieved 99.8% better than using only cnn architecture and LSTM approach. Go et al. [15] proposed a visualization-based approach by using CNN model ResNeXt achieved outstanding performance of highest 98.86% accuracy. Mitsuhashi et al.c[16] proposed a VGG19 based model that outperformed previously proposed models like VGG16 and found that the experimental test showed accuracy 99.72%. Khan et al. [17] found that their proposed model achieved 97.80% accuracy on a malware classification dataset. AlGarni et al. [18] discovered that EfficientNet3 based EfficientNet B3 performed 99.93% accuracy and prove that EfficientNet is one of the best architectures to use in image based datasets for better results. Nisa et al. [19] proposed and developed a method that achieved an accuracy of 99.3% on the cubic SVM classifier, which shows extraordinary result when compared to many current malware classification models. Bensaoud et al. [20] discovered that their InceptionV3 model performs better than one of the most useful models called M-CNN and its accuracy is 99.24%.

2.3 Comparative Analysis and Summary

In evaluating the performance of Convolutional Neural Networks (CNNs) for malware classification, a comparative analysis was conducted to assess their efficacy in contrast to other CNN models. The selected CNN architectures, customized for the diversity of our malware dataset, demonstrated promising results over conventional approaches. The accuracy achieved by the CNN model surpassed that of traditional methods, underscoring the superior ability of deep learning techniques to discern intricate patterns within the dataset. Precision, recall, and F1-score metrics further substantiated the effectiveness of the CNN model, particularly in minimizing false positives and false negatives. The computational efficiency of the CNN approach, while leveraging parallel processing capabilities, also proved advantageous in handling the complexity of the malware classification task.

Pascanu et al. [2] used Hybrid model based on ESNs and RNNs resulting in 98.4% accuracy. Kalash et al. [3] proposed a Convolutional Neural Network model that achieved 99.97%, a deep learning approach that performs better than traditional methods. Lad et al. [4] developed Hybrid CNN+SVM model that achieved 99.59%. Vasan et al. [5] proposed Image-based Malware Classification using Ensemble of CNNs that scored 99.50% accuracy. Gibert Et al. [6] proposed that 3 convolutional layers and 2 densely-connected

layers outperforms other methods scoring 94.64% accuracy. Pant Et al. [7] developed a Custom model that provides better results than VGG16 and ResNet-18 which achieved 98.7% accuracy. Awan Et al. [8] proposed that spatial attention and convolutional neural network achieved better result based on VGG-19. Dang Et al. [9] proposed Combining biLSTM, embedding, and CNN model results a huge improvement achieving 94.32% accuracy. Kumar Et al. [10] proposed model MCFT-CNN based on ResNet50 scores 98.63% accuracy.

My research contains both CNN architectures and traditional approaches. Among these different approaches the most successful model is ResNet-50 V2 which performed very well on different categories. ResNet-50 V2 achieved almost 87% accuracy while classifying malware dataset. Average AUC and ROC score is 0.99. Also its confusion matrix is well defined compared to other CNN models.

In summary, our exploration of CNNs in the realm of malware classification yielded promising results. The adaptability of the chosen CNN architecture to the diverse nature of the malware dataset showcased its potential for real-world applications. The comparative analysis highlighted the inherent strengths of CNNs in handling the complexities of malware classification, offering a significant leap forward in accuracy and efficiency. Furthermore, the study delved into the generalization capability of the CNN model, revealing its ability to adapt to variations in data distribution and malware strains. This comprehensive investigation positions CNNs as formidable tools in the domain of malware classification, offering a paradigm shift in the approach to cybersecurity. The findings of this study contribute valuable insights to the ongoing discourse on leveraging deep learning for enhanced malware detection and classification, paving the way for more resilient and efficient cybersecurity measures.

2.4 Scope of the Problem

The scope of the malware classification problem addressed in this research is vast, given the ever-evolving nature of cyber threats and the imperative to fortify digital ecosystems against malicious activities. The problem encompasses the diverse landscape of malware, including but not limited to viruses, worms, trojans, ransomware, and other sophisticated forms of malicious software. The scale of the problem is magnified by the continuous proliferation of new malware strains, each exhibiting unique characteristics and evasion techniques. As technology advances, malware creators adapt, creating a dynamic

environment that demands innovative solutions for effective detection and classification. This research narrows its focus to a specific malware dataset comprising nine distinct classes. The dataset's diversity poses a challenge and an opportunity to explore the capabilities of Convolutional Neural Networks (CNNs) in accurately classifying varied forms of malware. The scope extends to the exploration of optimal CNN architectures, the impact of transfer learning, and the assessment of robustness against adversarial attacks. The scope extends beyond the technical intricacies of CNN models to the broader implications for cybersecurity. The findings of this study contribute insights that could inform the development of more resilient and adaptive systems, addressing the ongoing need for effective defenses against an evolving spectrum of cyber threats.

While this research provides a focused examination of malware classification using CNNs, it also recognizes the broader scope of the problem within the context of the global cybersecurity landscape. The insights gained from this study have the potential to influence the development of proactive and responsive strategies to mitigate the impact of malware on digital infrastructures.

2.5 Challenges

Undertaking the task of malware classification using Convolutional Neural Networks (CNNs) presents several inherent challenges that warrant careful consideration throughout the research process.

Dataset Imbalance: The dataset, consisting of nine distinct classes of malware, may exhibit imbalances in the distribution of samples among classes. Addressing this imbalance is crucial to prevent the model from favoring dominant classes and to ensure robust performance across all categories.

Complexity and Diversity of Malware: The inherent complexity and diversity of malware strains pose a significant challenge. Designing a CNN model capable of capturing nuanced patterns and features across a wide spectrum of malicious software requires careful consideration of architectural choices and training strategies.

Limited Labeled Data: The availability of labeled data for training CNN models may be limited, especially for emerging or rare malware strains. This scarcity poses a challenge in achieving optimal model performance.

Interpretability and Explainability: CNNs, often perceived as black-box models, pose challenges in terms of interpretability and explainability. Understanding the decision-making process of the model is crucial for building trust and ensuring the practical application of the classification results.

Computational Resources: Training and fine-tuning deep CNN models demand substantial computational resources. Ensuring access to powerful hardware or cloud services is essential for conducting experiments efficiently and exploring diverse model configurations.

Generalization to Real-World Scenarios: Achieving high accuracy on the training and testing datasets may not necessarily guarantee the model's effectiveness in real-world scenarios. Generalizing the CNN model to handle unseen malware variants and evolving threat landscapes remains a persistent challenge.

Ethical Considerations: The deployment of malware classification systems raises ethical considerations, especially concerning privacy and potential misuse. Striking a balance between effective cybersecurity measures and respecting user privacy is a challenge that necessitates thoughtful consideration. Addressing these challenges requires a holistic and adaptive approach throughout the research journey.

CHAPTER 3

Research Methodology

3.1 Research Subject and Instrumentation

The primary focus of this study is the application of Convolutional Neural Networks (CNNs) in the classification of malware. The research delves into the intricate landscape of malicious software, encompassing a diverse dataset consisting of nine distinct classes. These classes represent various types of malware, including viruses, worms, trojans, and other sophisticated threats prevalent in the contemporary cybersecurity landscape.

Convolutional Neural Networks (CNNs): The core instrument for this research is the implementation of CNNs, a specialized form of deep neural networks renowned for their effectiveness in image classification tasks. The chosen architectures are VGG16, ResNet50 V2, DenseNet-121 and Inception-V3 and the configurations of the CNN serve as the primary tool for learning and discerning patterns within the malware dataset.

Programming Frameworks: The research employs programming frameworks such as TensorFlow , keras library, numpy, SKlearn for the implementation and training of the CNN models. These frameworks provide a robust infrastructure for building, fine-tuning, and evaluating deep learning models.

Malware Dataset: The malware dataset, comprising instances from nine different classes as Gatak, Lollipop, Kelihos_ver3, Tracur, Kelihos_ver1, Simda, Vundo, Ramnit, and Obfuscator.ACY, serves as the foundational input for training and evaluating the CNN models. The dataset is meticulously curated to represent the diverse and evolving nature of real-world malware.

Performance Metrics: Quantitative metrics such as accuracy, precision, recall, and F1-score serve as instruments for objectively evaluating the performance of the CNN models. These metrics provide insights into the model's ability to accurately classify instances across different malware classes.

Computational Resources: High-performance computational resources such as T4(GPU) on google colab, high-performance hardwares are essential instruments for training and fine-tuning the deep CNN models efficiently.

3.2 Dataset Utilized

The dataset was taken from kaggle. Gatak, Lollipop, Kelihos_ver3, Tracur, Kelihos_ver1, Simda, Vundo, Ramnit, and Obfuscator.ACY are the 9 classes that can be found in the dataset, each representing a different type of malware. These classes encompass a spectrum of threats, including viruses, worms, trojans, ransomware, and other sophisticated forms of malicious software. The total number of images is 12060. To reflect the real-world variability of malware, the dataset encompasses a diverse range of samples. This diversity ensures that the CNN models are exposed to a broad spectrum of malicious behaviors and patterns. The ensuing sections will delve into the methodology, experimental setup, and results obtained, providing a comprehensive overview of the approach taken to tackle the complexity inherent in classifying diverse forms of malware.

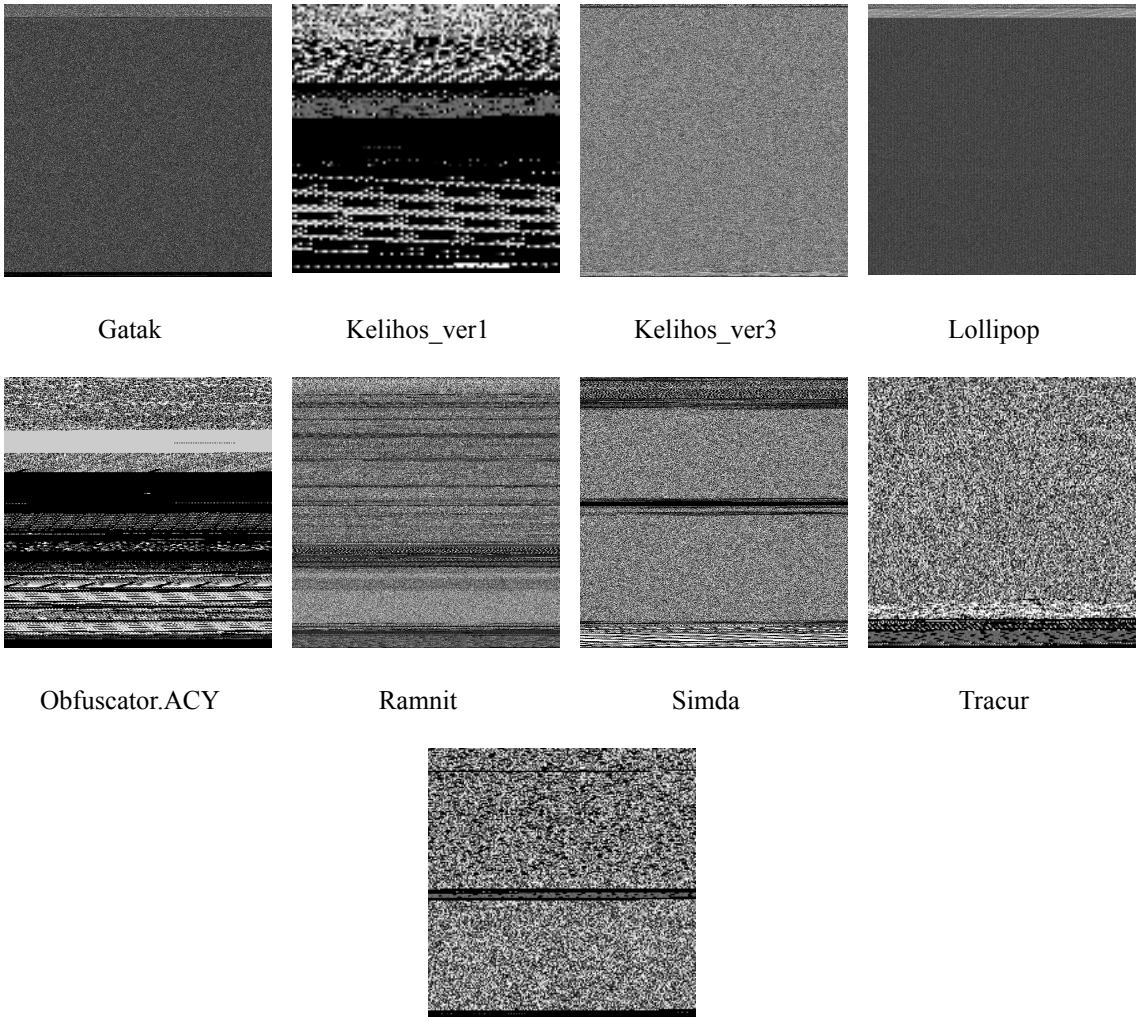


Figure 3.2.1: Images of Malwares from 9 Classes

3.3 Statistical Analysis

The statistical analysis in this research is multifaceted, encompassing various methodologies to assess the performance, robustness, and generalization capabilities of Convolutional Neural Networks (CNNs) in the classification of malware. Key statistical techniques employed include:

Model Performance Metrics: Quantitative metrics, including accuracy, precision, recall, and F1-score, are employed to objectively evaluate the performance of the CNN models. These metrics provide statistical indicators of the models' ability to accurately classify instances across different malware classes.

Confusion Matrix Analysis: Confusion matrices are constructed to provide a detailed breakdown of true positive, true negative, false positive, and false negative predictions. This statistical tool aids in understanding the specific areas of strength and weakness in the CNN models' classification capabilities.

Generalization Analysis: Statistical techniques, including cross-validation and ROC-AUC analysis, are applied to assess the generalization capabilities of the CNN models. These analyses provide insights into the models' ability to perform well on unseen instances and adapt to variations in data distribution.

The dataset contains 9 classes Gatak, Lollipop, Kelihos_ver3, Tracur, Kelihos_ver1, Simda, Vundo, Ramnit, and Obfuscator.ACY and a total of 12060 images.

Table 3.3.1: Number of Images of Each Classes Table

Class Name	Number of Images
Gatak	642
Lollipop	1228
Kelihos_ver3	775
Tracur	751
Kelihos_ver1	1533
Simda	1013

Vundo	2942
Ramnit	2478
Obfuscator.ACY	698

3.4 Proposed Methodology

The methodology for this research is meticulously designed to explore the application of Convolutional Neural Networks (CNNs) in the classification of malware. The step-by-step approach encompasses data preparation, model development, training, evaluation, and analysis. The proposed methodology is outlined as follows.

This proposed methodology aims to provide a structured and rigorous approach to investigating the capabilities of CNNs in the context of malware classification while considering ethical considerations and potential adversarial threats.

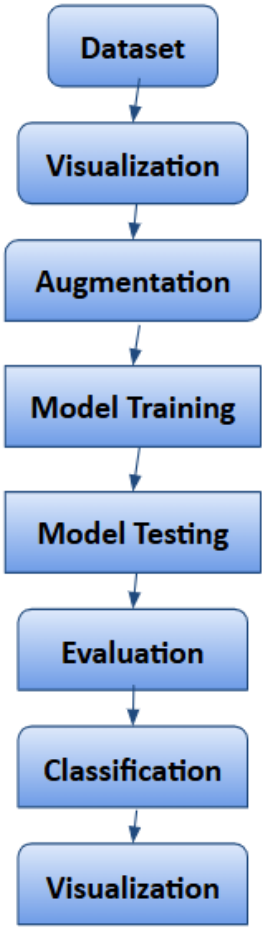


Figure 3.4.1: Proposed Methodology

Convolutional Neural Networks (CNNs) stand as a formidable architecture in the realm of deep learning, particularly renowned for their prowess in image-based classification tasks. The inherent hierarchical structure of CNNs, consisting of convolutional layers, pooling layers, and densely connected layers, enables them to automatically learn intricate hierarchical representations from input data. In the context of malware classification, CNNs prove to be highly adept at discerning nuanced patterns and features associated with various types of malicious software. The convolutional layers act as feature extractors, capturing local patterns, while pooling layers downsample spatial dimensions, preserving essential information. This culminates in densely connected layers that leverage the learned features for making class predictions. Dropout layers are often incorporated to mitigate overfitting, enhancing the generalization capability of the model. The strength of CNNs lies in their ability to automatically learn hierarchical representations, making them well-suited for tasks where the relationships between features are crucial. The customization of CNN architectures allows for adaptability to the intricacies of different datasets, making them an instrumental tool in advancing the state-of-the-art in malware classification.

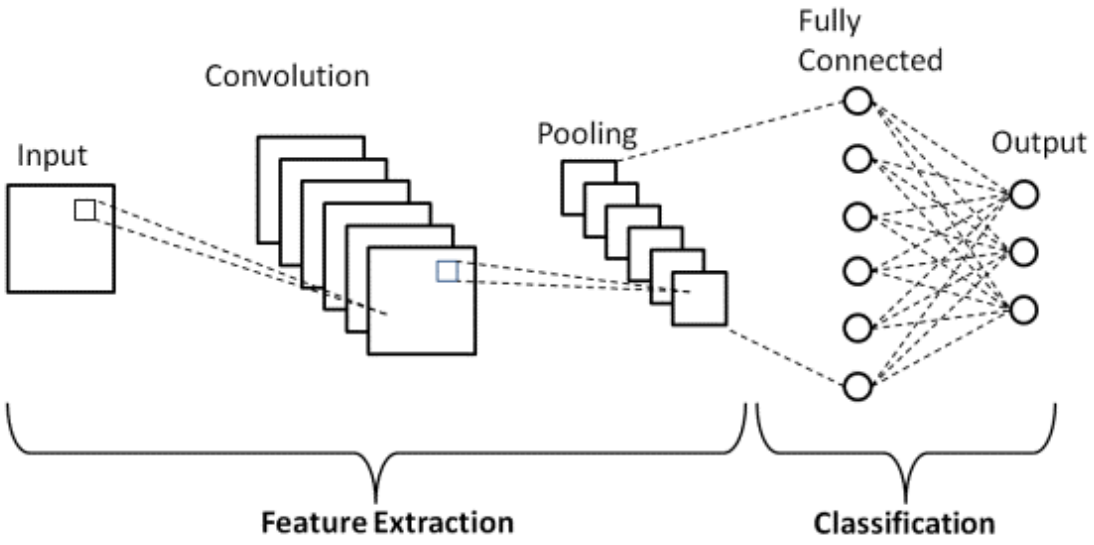


Figure 3.4.2: The basic CNN architecture.

The used CNN architectures are given in Table 3.4.1, which suggests an assessment of these used CNN architectures.

Table 3.4.1: A comparison of popular CNN architectures (M= Million)

Architecture Name	Year	Number of Parameters	Layer
VGG	2014	138M	16
DenseNet	2017	8.1M	121
InceptionV3	2015	24M	48
ResNet	2015	25.6M	50

VGG 16: The VGG16 (Visual Geometry Group 16) architecture represents a powerful convolutional neural network renowned for its deep structure and exceptional performance in image classification tasks. Comprising 16 weight layers, VGG16 is characterized by its uniform and straightforward architecture, featuring multiple convolutional and pooling layers. Below is a simplified representation of the VGG16 architecture, which can be employed as a feature extractor in various computer vision applications, including malware classification. VGG16 serves as a powerful feature extractor in image classification tasks, and its adaptability makes it a valuable asset in exploring the capabilities of deep learning for malware classification.

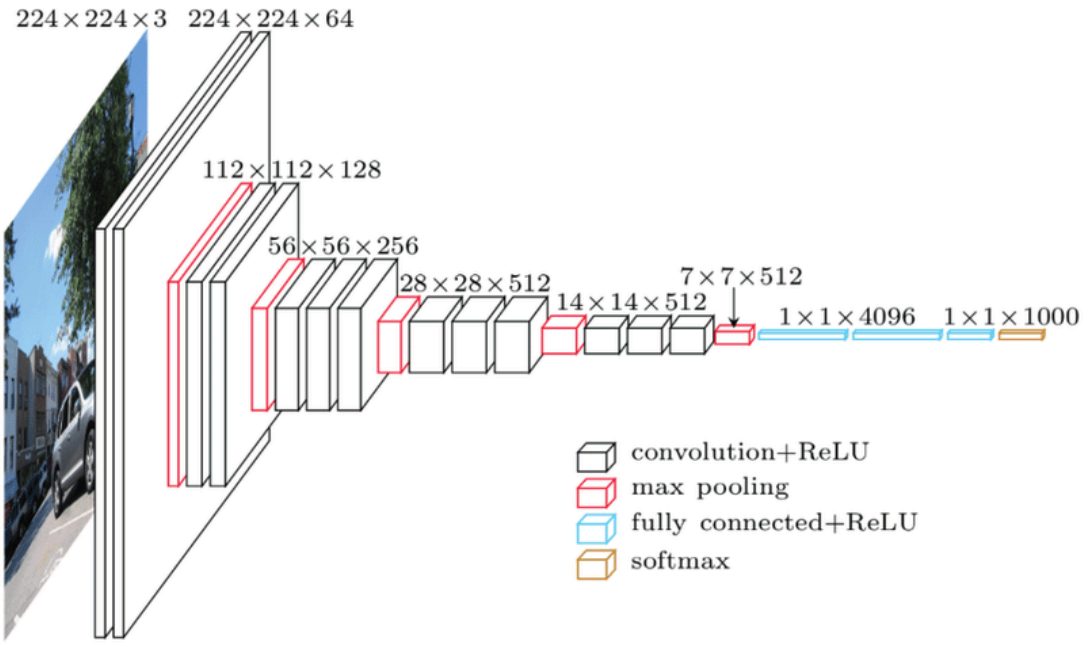


Figure 3.4.3: VGG 16 architecture.

DenseNet-121: DenseNet-121 is a convolutional neural network architecture that belongs to the family of DenseNets, known for their dense connectivity patterns. DenseNet-121 is specifically characterized by its depth and parameter efficiency, offering a compelling solution for image classification tasks. Below is a simplified representation of the DenseNet-121 architecture. DenseNet-121's innovative dense connectivity and parameter-sharing approach contribute to its effectiveness in image classification tasks. Its structured design fosters feature reuse and efficient gradient flow, making it a valuable candidate for exploring the complexities of malware classification.

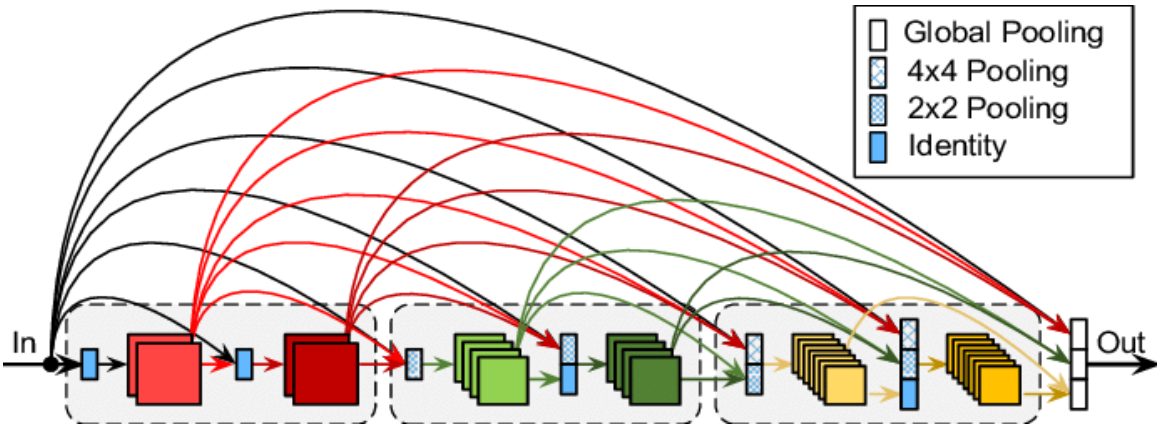


Figure 3.4.4: DenseNet-121 architecture.

ResNet50 V2: ResNet-50 V2 (Residual Network 50, Version 2) is a variant of the ResNet architecture, a groundbreaking design known for introducing residual connections to address the challenges of training very deep neural networks. ResNet-50 V2 extends the original ResNet-50 with modifications such as improved skip connections and bottleneck blocks. Below is a simplified representation of the ResNet-50 V2 architecture. ResNet-50 V2's innovative residual connections enable the training of very deep networks, making it highly effective for tasks such as image classification. Its structured design with bottleneck blocks enhances feature learning and fosters gradient flow.

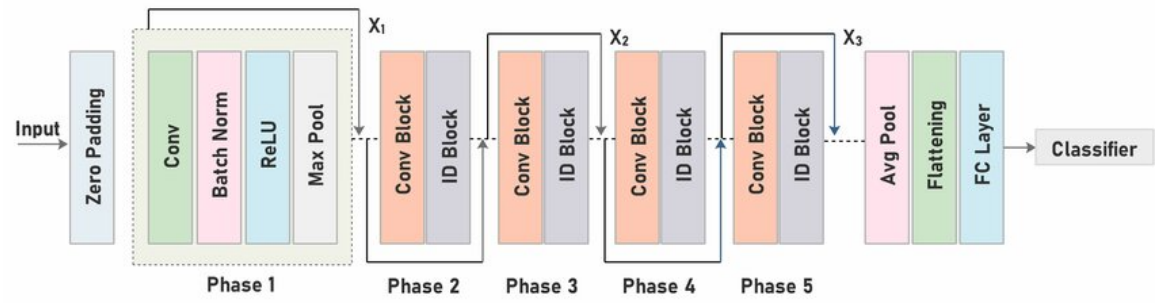


Figure 3.4.5: ResNet50 V2 architecture.

Inception-V3: Inception V3 is a convolutional neural network architecture designed for image classification, object detection, and other computer vision tasks. It is part of the Inception family, characterized by its innovative use of inception modules to capture multi-scale features efficiently. Below is a simplified representation of the Inception V3 architecture. Inception V3's inception modules enable the efficient capture of multi-scale features, making it effective for tasks requiring detailed feature extraction. Its structured design with parallel branches enhances feature learning, making it a valuable candidate for image classification tasks, including the exploration of complexities in malware classification.

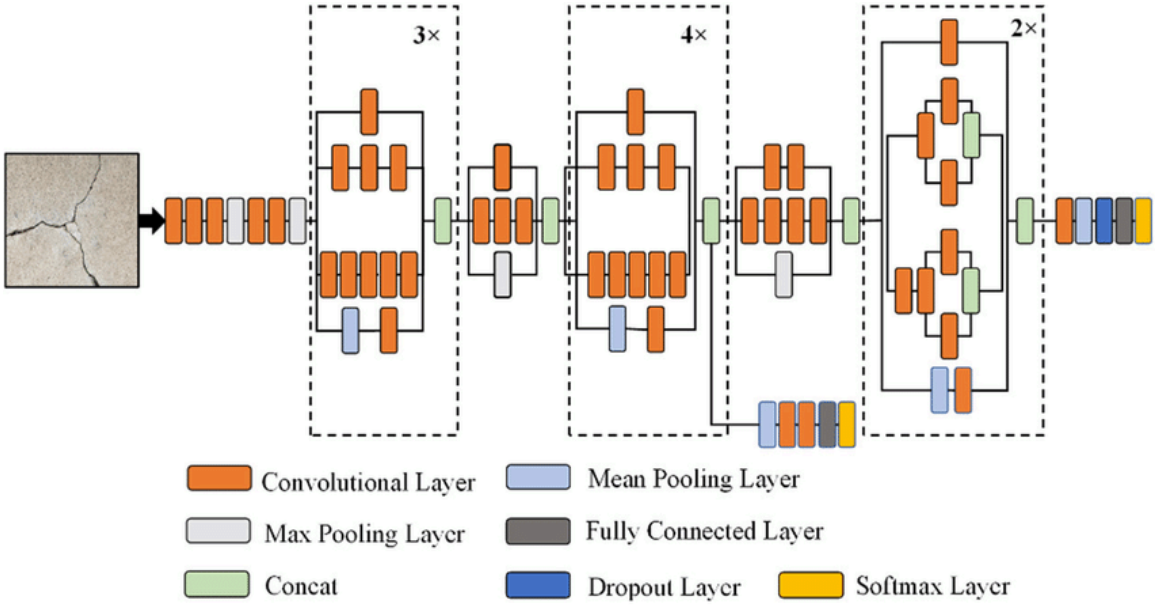


Figure 3.4.6: Inception-V3 architecture.

Malware Classification Using CNN Based Deep Learning: The ever-growing complexity of cyber threats demands innovative approaches to bolster digital defenses. This study embarks on a journey into the realm of malware classification, harnessing the power of Convolutional Neural Networks (CNNs) within the framework of deep learning. Gatak, Lollipop, Kelihos_ver3, Tracur, Kelihos_ver1, Simda, Vundo, Ramnit, and Obfuscator.ACY serve as focal points in this exploration, representing diverse malware strains with distinct characteristics. The utilization of CNNs, renowned for their prowess in pattern recognition and hierarchical feature extraction, stands as a transformative force in the quest for more effective cybersecurity measures.

By training these advanced models on a dataset encompassing the intricacies of Gatak's stealth, Lollipop's deceptive maneuvers, Kelihos_ver3's polymorphic nature, Tracur's

evasive tactics, Kelihos_ver1's tenacity, Simda's propagation strategies, Vundo's resilience, Ramnit's multifaceted attacks, and Obfuscator.ACY's code obfuscation, we aim to enhance the accuracy and efficiency of malware classification. The study not only delves into the technical aspects of CNN-based classification but also explores the broader implications for cybersecurity strategies. As digital threats continue to evolve, understanding the interplay between CNNs and diverse malware strains becomes paramount. This research contributes to the ongoing discourse on leveraging deep learning for cybersecurity, striving to fortify our defenses against the dynamic landscape of cyber adversaries.

3.5 Implementation Requirements

The implementation of Convolutional Neural Networks (CNNs) for malware classification entails specific requirements to ensure a successful and effective execution. Here are the key implementation requirements:

Programming Environment: Utilize a suitable programming environment for deep learning, such as Python, along with libraries like TensorFlow. This framework provides comprehensive support for building and training CNN models.

Hardware Resources: Employ hardware resources with sufficient computational power, such as GPUs or TPUs. These accelerators significantly expedite the training process of deep neural networks, reducing the overall time required.

Dataset: Acquire a well-curated and diverse malware dataset containing instances from various classes. Ensure proper labeling of instances to facilitate supervised learning. The dataset should be split into training and testing sets for model evaluation.

CNN Architecture: Design or choose a suitable CNN architecture based on the characteristics of the malware dataset. Consider popular architectures like VGG, ResNet, or Inception, and customize the architecture to accommodate the specific requirements of the classification task.

Hyperparameter Tuning: Fine-tune hyperparameters, including learning rate, batch size, and dropout rates, to optimize the performance of the CNN model. Experiment with different configurations to achieve the best results.

Model Evaluation Metrics: Choose appropriate evaluation metrics for assessing the performance of the CNN models. Common metrics include accuracy, precision, recall, F1-score, and confusion matrices. These metrics provide insights into the model's classification capabilities.

Version Control: Implement version control practices using tools like Git to track changes in the codebase. Version control ensures traceability and collaboration, especially when multiple iterations of the model are developed.

CHAPTER 4

Experimental Results and Discussion

4.1 Experimental Setup

The experimental setup for training and evaluating Convolutional Neural Networks (CNNs) in malware classification involves several crucial components. Here's a comprehensive outline of the experimental setup:

Hardware and Software Configuration: Specify the hardware infrastructure, including CPU, GPU, or TPU specifications. Ensure compatibility with deep learning frameworks such as TensorFlow or PyTorch. In this experiment we used T4 GPU and TensorFlow.

Deep Learning Framework: Choose a deep learning framework as TensorFlow and specify the version. Include any additional libraries or frameworks used for data manipulation, preprocessing, and evaluation like numpy, panda, Conv2D, Dropout.

Dataset Selection: Clearly define the malware dataset used for training and testing. Provide details on the number of instances, classes, and any preprocessing steps applied, such as image resizing, normalization, or class balancing.

Data Splitting: Split the dataset into training(80%) and testing(20%) sets. Specify the ratio of data allocated for training and testing to ensure a comprehensive evaluation of the CNN models.

Model Architecture: Detail the architecture of the CNN models used for malware classification. Specify the type of architecture (VGG, ResNet, Inception, DenseNet), the number of layers, and any customized modifications made for the specific task.

Training Procedure: Describe the training procedure, including 20 epochs, optimization algorithm used adam, and any regularization techniques applied. Early stopping is used.

Model Evaluation Metrics: The evaluation metrics used to assess the performance of the CNN models include accuracy, precision, recall, F1-score, and confusion matrices. Provide insights into how these metrics contribute to the understanding of model effectiveness.

4.2 Experimental Results and Analysis

The results of the four individual CNN networks VGG16, ResNet50 V2, Inception V3, DenseNet-121 are presented in the table as well as some traditional approaches including MLP, Random Forest, LSTM, KNN. The architecture and algorithms classification result is shown here. After that, the overall measures for those models are discussed and visualized.

Table 4.2.1: Accuracy Table for Model and Training

Architecture/Algorithm	Training Accuracy	Testing Accuracy
Inception-V3	0.831	0.785
VGG-16	0.796	0.807
ResNet-50 V2	0.878	0.869
Densenet-121	0.785	0.825
MLP	0.945	0.760
LSTM	0.788	0.770
KNN		0.620

The best performing CNN architecture is ResNet-50 V2 and way ahead of the other CNNs in terms of building the model and testing the model. ResNet-50 V2 achieved 87% model fit accuracy and 86% model test accuracy. Inception-V3 showed promising results as well, gaining 82% model fit accuracy and 78% model test accuracy. VGG-16 and DenseNet-121 both performed almost similarly. But in case of model test accuracy DenseNet-121 is ahead of VGG-16. On the other hand, the traditional machine learning approach falls behind the CNN models. Best result from ML was provided by Random Forest achieving 79%. Despite MLP having 94% model fit accuracy it did not provide higher results than any ML algorithms or CNN architectures.

A classification report is a comprehensive summary of the performance of a machine learning model on a classification task. It includes various metrics that provide insights into the model's accuracy, precision, recall, F1-score, and support for each class.

Table 4.2.2: Classification report of Inception-V3

Inception-V3				
Class Name	Precision	Recall	F1-score	Support(N)
Gatak	0.67	0.73	0.70	155
Lollipop	0.94	0.90	0.92	246
Kelihos_ver3	0.82	0.82	0.82	128
Tracur	0.47	0.69	0.45	151
Kelihos_ver1	0.95	0.93	0.94	307
Simda	1.00	0.76	0.65	203
Vundo	0.86	0.65	0.74	589
Ramnit	0.94	0.74	0.83	496
Obfuscator.ACY	0.71	0.86	0.78	140

Table 4.2.3: Classification report of VGG-16

VGG-16				
Class Name	Precision	Recall	F1-score	Support(N)
Gatak	0.92	0.80	0.63	155
Lollipop	0.87	0.84	0.85	246
Kelihos_ver3	0.94	0.95	0.94	128
Tracur	0.62	0.34	0.44	151
Kelihos_ver1	0.92	0.90	0.91	307
Simda	0.49	0.83	0.62	203

Vundo	0.97	0.94	0.95	589
Ramnit	0.89	0.80	0.84	496
Obfuscator.ACY	0.96	0.92	0.94	140

Table 4.2.4: Classification report of ResNet-50 V2

ResNet-50 V2				
Class Name	Precision	Recall	F1-score	Support(N)
Gatak	0.78	0.86	0.82	155
Lollipop	0.97	0.89	0.93	246
Kelihos_ver3	0.84	0.99	0.91	128
Tracur	0.71	0.63	0.67	151
Kelihos_ver1	1.00	0.91	0.95	307
Simda	1.00	0.22	0.36	203
Vundo	0.74	0.44	0.55	589
Ramnit	0.88	0.90	0.89	496
Obfuscator.ACY	0.89	0.82	0.86	140

Table 4.2.5: Classification report of DenseNet-121

DenseNet-121				
Class Name	Precision	Recall	F1-score	Support(N)
Gatak	0.80	0.53	0.64	155
Lollipop	0.95	0.88	0.91	246
Kelihos_ver3	0.81	0.99	0.89	128
Tracur	0.46	0.62	0.53	151
Kelihos_ver1	0.93	0.93	0.93	307
Simda	1.00	0.11	0.19	203
Vundo	0.64	0.33	0.44	589
Ramnit	0.82	0.85	0.84	496
Obfuscator.ACY	0.91	0.79	0.85	140

ResNet50 V2 proved to be the best in terms of classification report. It out scored other models in every class.

A confusion matrix is a table that visualizes the performance of a classification algorithm by summarizing the counts of true positive (TP), true negative (TN), false positive (FP), and false negative (FN) predictions. Each row of the matrix represents the actual class, and each column represents the predicted class. The confusion matrix provides a detailed breakdown of the model's performance and is useful for assessing areas where the model may need improvement. The diagonal elements represent correct predictions, where the actual and predicted classes match. The higher these values, the better the model's performance. The off-diagonal elements indicate misclassifications. Lower values here indicate better model performance.

True Positive (TP): Instances where the model correctly predicted the positive class.

True Negative (TN): Instances where the model correctly predicted the negative class.

False Positive (FP): Instances where the model predicted the positive class, but the actual class was negative (Type I error).

False Negative (FN): Instances where the model predicted the negative class, but the actual class was positive (Type II error).

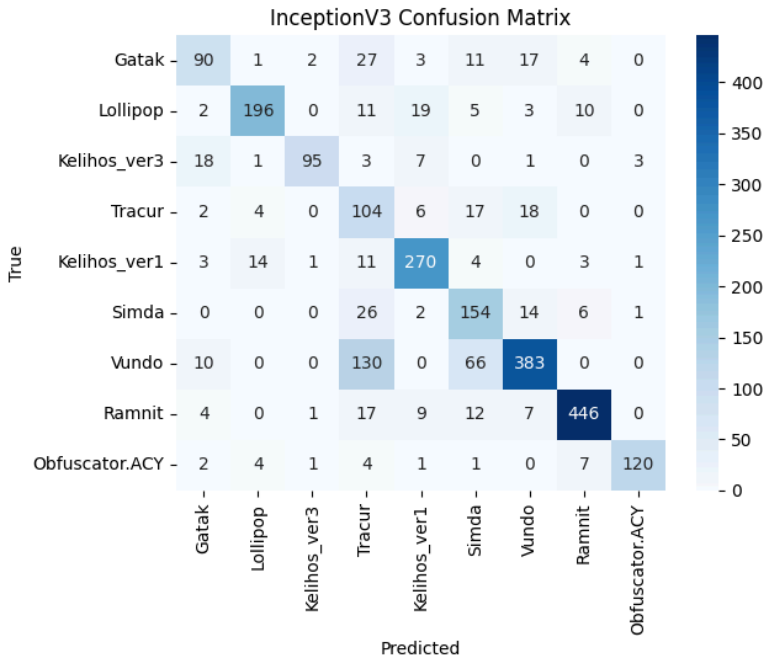


Figure 4.2.1: Confusion Matrix of Inception-V3

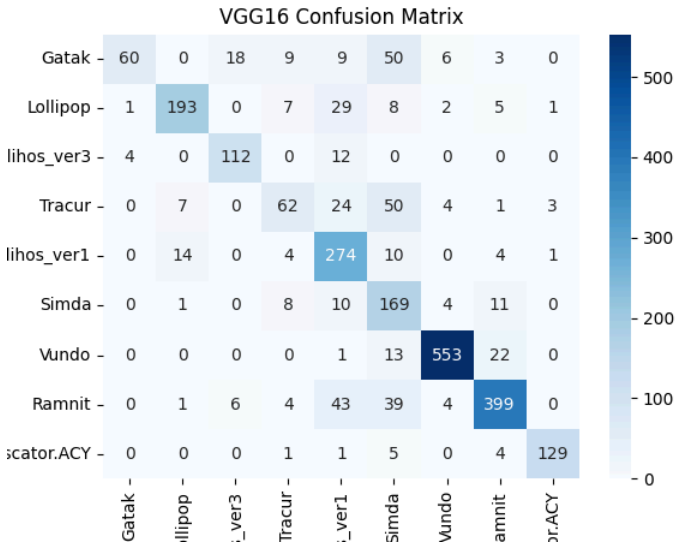


Figure 4.2.2: Confusion Matrix of VGG-16

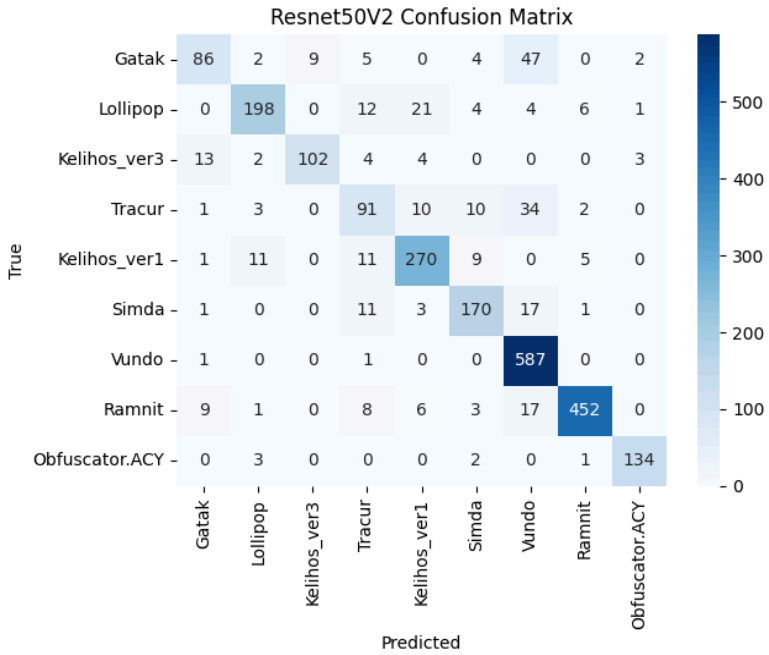


Figure 4.2.3: Confusion Matrix of ResNet-50 V2

In the measurement section of the confusion matrix ResNet50 V2 also performs better than other models. It can predict more true positive values resulting in it to be the best.

Tracking training accuracy, validation accuracy, training loss, and validation loss during the training process is crucial for assessing the performance and generalization capability of a machine learning model.

If training accuracy is high, but validation accuracy is low, it might indicate overfitting. If both training and validation accuracy are low, the model may be underfitting. Monitoring training and validation loss helps in diagnosing convergence and potential issues.

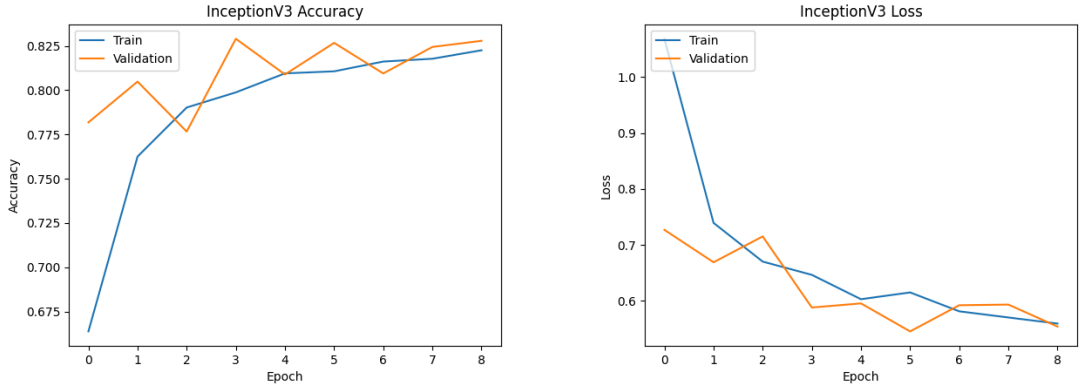


Figure 4.2.5: Accuracy and loss of Inception-V3

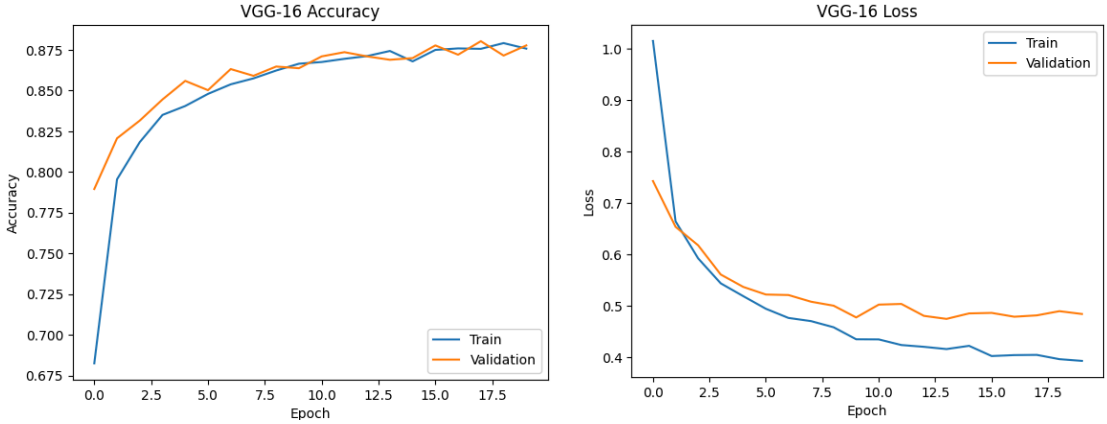


Figure 4.2.6: Accuracy and loss of VGG-16

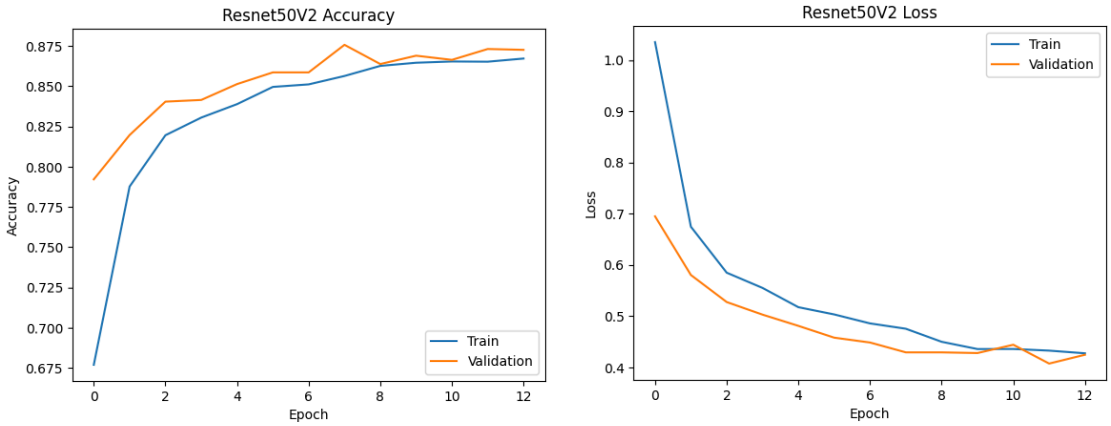


Figure 4.2.7: Accuracy and loss of ResNet-50 V2

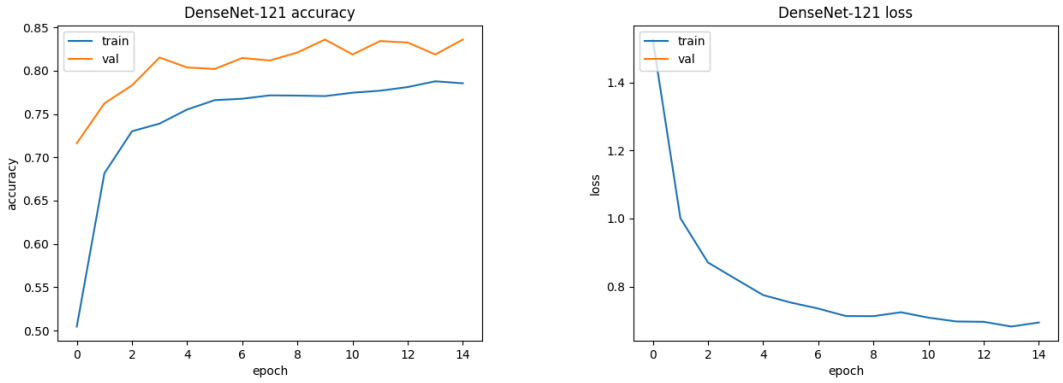


Figure 4.2.8: Accuracy and loss of DenseNet-121

The Area Under the Curve (AUC) and Receiver Operating Characteristic (ROC) curve are evaluation metrics commonly used to assess the performance of a binary classification model, particularly in scenarios where sensitivity and specificity are crucial. The ROC curve is a graphical representation of a binary classification model's performance across different classification thresholds. AUC represents the area under the ROC curve. It quantifies the model's ability to distinguish between positive and negative instances across all possible classification thresholds.

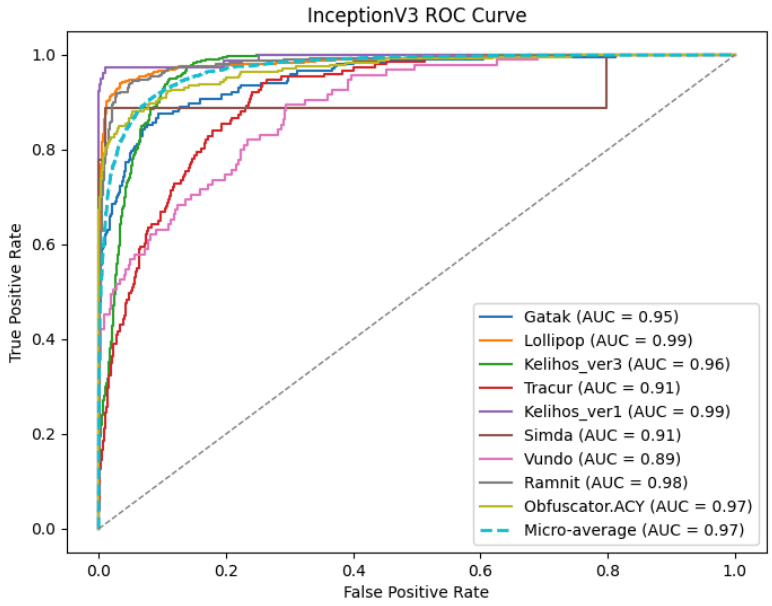


Figure 4.2.9: ROC curve of Inception-V3

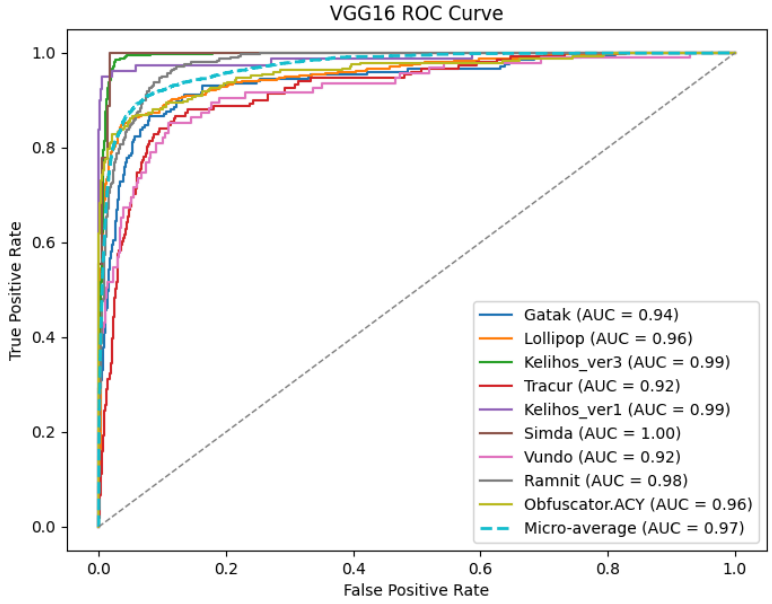


Figure 4.2.10: ROC curve of VGG-16

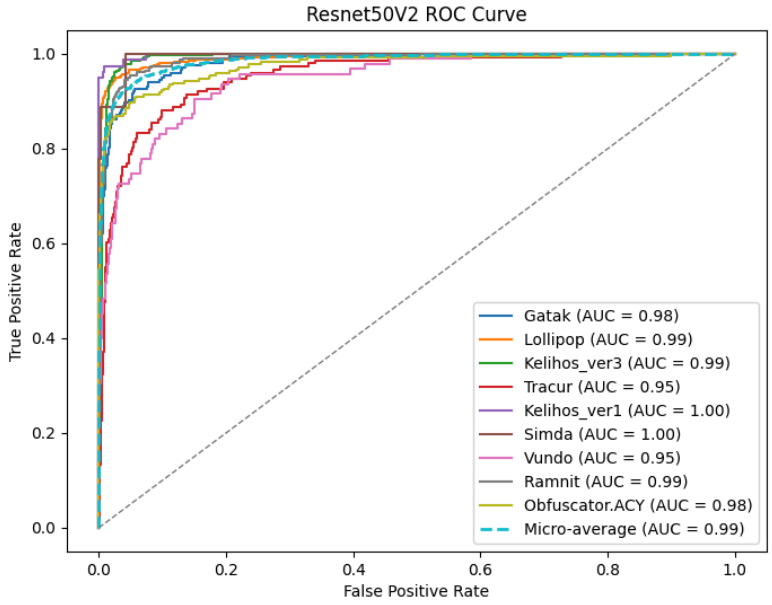


Figure 4.2.11: ROC curve of ResNet-50 V2

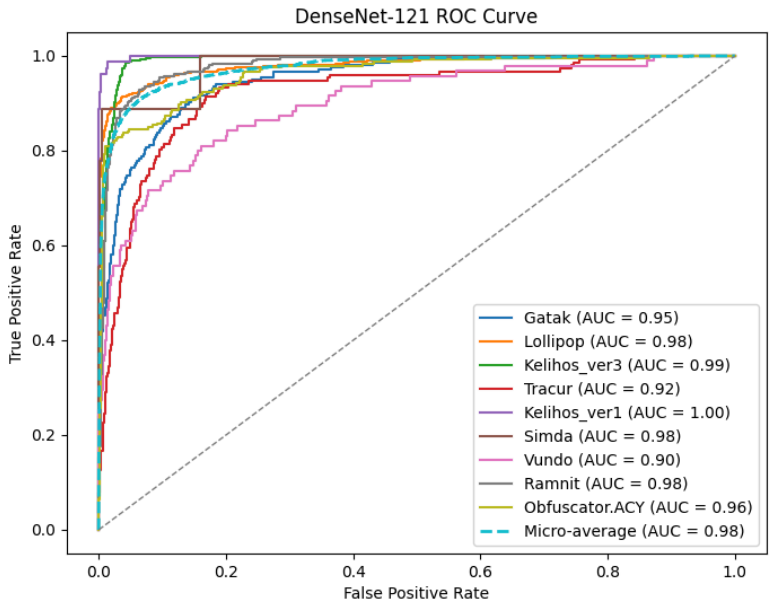


Figure 4.2.12: ROC curve of DenseNet-121

The best micro average AUC is 0.99 out of 1 and achieved by ResNet50 V2. DenseNet-121 scored 0.98 and the other two scored 0.97.

4.3 Discussion

These research questions delve into the comparative analysis, generalization capabilities, architectural configurations, and robustness of Convolutional Neural Networks (CNNs) in the context of malware classification. Each question addresses a key aspect of CNN-based malware classification research.

Assess the effectiveness of CNNs against traditional methods in handling diverse malware strains. Investigate how variations in dataset size and diversity influence the generalization capabilities of CNNs, especially in the context of handling previously unseen malware samples. Explore the impact of architectural choices (depth, kernel sizes, layer configurations) on the ability of CNNs to detect and classify specific types or families of malware. Investigate the potential benefits of ensemble learning (combining multiple CNN models) for improving the robustness and performance of malware classification models across diverse datasets. Assess the vulnerability of CNN models to adversarial attacks in the context of malware classification and explore strategies to enhance their robustness in real-world scenarios.

CHAPTER 5

Impact on Society, Environment and Sustainability

5.1 Impact on Society

Among the most obvious effects is the strengthening of cybersecurity defenses. Organizations and individuals can enhance their ability to safeguard their systems and data from cyber attacks by precisely categorizing malware. Thus, it becomes easier to avoid data breaches, monetary losses, and interruptions to vital infrastructure.

Financial information, login credentials, and confidential documents are among the personal data that is frequently the target of malware assaults. By effectively classifying malware using CNN models, this data is protected, protecting people's privacy and averting fraud and identity theft.

People in today's linked society heavily rely on technology for a variety of daily tasks, including banking, healthcare, and transportation in addition to communication and other daily activities. Hackers using malware can make people less trusting of these digital platforms. CNN-based categorization methods help to preserve public confidence in technology and digital services by efficiently thwarting malware.

Cyberattacks have the potential to have a major negative economic impact on people, governments, and enterprises. Data breaches, system outages, and remediation projects can come with astronomical expenses. CNN-based categorization systems contribute to economic growth and stability by lessening the frequency and severity of malware assaults.

Educational and career opportunities are created by the increasing need for cybersecurity specialists who are proficient in machine learning and deep learning methodologies. There is a growing interest in teaching people how to create, install, and maintain CNN-based malware categorization systems as more people realize how important cybersecurity is.

The application of CNN models for malware classification has far-reaching effects on society at large, affecting not just cybersecurity but also privacy, trust, the economy, security, innovation, and education. Through the management of the always changing

malware threat landscape, these technologies help create a more secure and resilient digital environment that benefits people, businesses, and governments alike.

5.2 Impact on Environment

While there are some indirect ways in which malware classification using CNN models can affect environmental sustainability, its effects on the environment are not direct in the conventional sense.

Large computational resources are needed for training deep learning models, such as CNNs, which use energy. The training procedure is frequently carried out in sizable data centers that need a lot of power to run. This energy use can contribute to carbon emissions and other environmental issues, while the exact effect on the environment will depend on where the electricity comes from.

The environmental and socioeconomic implications of deep learning technologies, such as CNN models for malware classification, must be taken into consideration in order to ensure their long-term sustainability. This includes promoting responsible computing resource usage, optimizing hardware and algorithms for energy efficiency, and taking into account the broader environmental effects of technical breakthroughs.

Even if individual deep learning initiatives might not have as much of an influence on the environment as other industries, like transportation or industrial production, it's still vital to take into account the overall implications of widespread adoption and the scalability of these technologies. Researchers and practitioners can help minimize their environmental footprint and accelerate the shift to a more sustainable digital future by supporting sustainable practices in the development and deployment of deep learning applications, such as malware classification using CNN models.

5.3 Ethical Aspects

Analyzing sizable datasets containing potentially sensitive data is a common step in the malware classification process. It is crucial to protect the privacy of those whose data is contained in these datasets. Informed consent must be obtained, data must be anonymized as far as feasible, and strong data protection mechanisms must be put in place to guard against misuse or unauthorized access.

CNN models are vulnerable to biases in the training set of data. Unfair results may result from this, such as the disproportionate marking of particular individuals or groups as possible dangers. The identification and mitigation of data biases, the assurance of model prediction fairness, and routine auditing and monitoring for inadvertent discriminatory effects are all ethical considerations.

To stop adversaries from taking advantage of CNN-based malware classification systems, it is essential to guarantee their security and resilience. This includes adopting encryption, access controls, and other security measures to protect sensitive data and prevent unauthorized access or manipulation.

It takes a multidisciplinary approach that combines technological know-how with ethical concepts and considerations to address the ethical elements of malware classification using CNN models.

5.4 Sustainability Plan

Creating a sustainability plan entails finding ways to lessen the influence on the environment, encourage social responsibility, and guarantee long-term viability when classifying malware using CNN models.

Examine the environmental impact of the CNN-based malware categorization study in detail. Throughout the course of the project, this entails quantifying resource utilization, carbon emissions, and energy consumption. Determine areas that can be optimized for energy efficiency, such as data storage procedures, hardware, and algorithms. When hosting and training CNN models, take into account the possible environmental advantages of using cloud computing services or renewable energy sources.

Put plans into action to reduce waste and maximize resource usage. This might be as simple as timing training tasks to make the most use of the technology, recycling or repurposing hardware components whenever possible, or employing data compression techniques to lower the amount of storage needed. To maximize the use of already-existing infrastructure and reduce effort duplication, look into possibilities for cooperation or resource sharing with other research groups or organizations.

CHAPTER 6

Summary, Conclusion and Implication for Future Research

6.1 Summary of the Study

The study "Malware Classification Using CNN Model: A Comparative Study" compares the performance of convolutional neural network (CNN) models with conventional approaches in order to determine how effective CNN models are at classifying malware. Different CNN architectures are investigated, along with signature-based detection and heuristic analysis, using a broad dataset of malware samples. The results show that CNN models perform better than conventional methods, exhibiting improved accuracy and better generalization to malware types that are not yet known. Experience-based learnings are important factors to take into account while developing cybersecurity plans. Discussions about potential solutions and future research initiatives are prompted by the acknowledgement of challenges such as dataset imbalance and computational resources. The study comes to the conclusion that CNN models offer a promising approach to classifying malware, emphasizing the need of continuous innovation in utilizing cutting-edge machine learning techniques to strengthen cybersecurity defenses and combat the malware's ever-changing threat landscape.

6.2 Conclusions

The research on CNN models for malware classification concludes by highlighting the tremendous potential that deep learning approaches have to improve cybersecurity efforts. CNN models outperform conventional techniques through comparison analysis, demonstrating their capacity to precisely categorize malware and adjust to changing threats. Concerns still exist, though, about issues like dataset imbalance and the need for computational resources. It is imperative to tackle these obstacles and investigate potential enhancement paths, such group techniques and transfer learning, in order to propel the efficiency and expandability of CNN-driven malware categorization frameworks. In summary, the study underscores the significance of continuous research and innovation in utilizing cutting-edge machine learning techniques to reinforce cybersecurity defenses and protect against the ever-changing and persistent threat landscape that malware poses.

6.3 Implication for Further Study

In order to improve classification accuracy and resilience, more research on malware classification using CNN models should examine the efficacy of ensemble approaches, such as merging many CNN models or fusing CNNs with other machine learning strategies. Furthermore, examining transfer learning strategies to use huge datasets and pre-trained CNN models for malware classification tasks may lessen the requirement for a lot of training data and processing power. The discipline can advance by utilizing novel feature extraction techniques specifically designed for malware classification, as well as approaches to analyze model predictions and offer insights into classification judgments. Prospective areas for future research include adversarial robustness strategies to enable resistance against evasion attempts and real-time detection algorithms that take latency constraints into account. When implementing CNN-based malware classification systems, ethical and societal considerations such as privacy, equity, and responsibility should be taken into account. By addressing these research gaps, we can improve cybersecurity defenses and lessen the threat posed by malicious software, which is always changing.

Reference:

[1] Dib, M., Torabi, S., Bou-Harb, E., & Assi, C., “A multi-dimensional deep learning framework for iot malware classification and family attribution.” IEEE Transactions on Network and Service Management, vol. 18, no. 2, pp. 1165-1177, June 2021.

[2] R. Pascanu, J. W. Stokes, H. Sanossian, M. Marinescu, and A. Thomas, "Malware classification with recurrent networks," IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 1916-1920, April 2015.

[3] M. Kalash, M. Rochan, N. Mohammed, N. D. Bruce, Y. Wang, and F. Iqbal, "Malware classification with deep convolutional neural networks," IFIP International Conference on New Technologies, Mobility and Security (NTMS) , pp. 1-5, February 2018.

[4] S. S. Lad and A. C. Adamuthe, "Malware classification with improved convolutional neural network model," Int. J. Comput. Netw. Inf. Secur., vol. 12, no. 6, pp. 30-43, December 2020.

[5] D. Vasan, M. Alazab, S. Wassan, B. Safaei, and Q. Zheng, "Image-Based malware classification using ensemble of CNN architectures (IMCEC)," Computers & Security, vol. 92, p. 101748, May 2020.

[6] D. Gibert, "Convolutional neural networks for malware classification," University Rovira i Virgili, Tarragona, Spain, pp. 1-98, October 2016.

[7] D. Pant and R. Bista, "Image-based malware classification using deep convolutional neural network and transfer learning," Proceedings of the 3rd International Conference on Advanced Information Science and System, pp. 1-6, November 2021.

[8] M. J. Awan, O. A. Masood, M. A. Mohammed, A. Yasin, A. M. Zain, R. Damaševičius, and K. H. Abdulkareem, "Image-based malware classification using VGG19 network and spatial convolutional attention," Electronics, vol. 10, no. 19, p. 2444, October 2021.

[9] D. Dang, F. Di Troia, and M. Stamp, "Malware classification using long short-term memory models," arXiv preprint arXiv:2103.02746, March 2021.

[10] 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, December 2021.

[11] A. Tekerek and M. M. Yapici, "A novel malware classification and augmentation model based on convolutional neural network," Computers & Security, vol. 112, p. 102515, January 2022.

[12] M. Ahmed, N. Afreen, M. Ahmed, M. Sameer, and J. Ahamed, "An Inception V3 approach for malware classification using machine learning and transfer learning," International Journal of Intelligent Networks, vol. 4, pp. 11-18, 2023.

[13] R. Chaganti, V. Ravi, and T. D. Pham, "A multi-view feature fusion approach for effective malware classification using Deep Learning," Journal of Information Security and Applications, vol. 72, p. 103402, 2023.

[14] B. N. Narayanan and V. S. P. Davuluru, "Ensemble malware classification system using deep neural networks," Electronics, vol. 9, no. 5, p. 721, 2020.

[15] J. H. Go, T. Jan, M. Mohanty, O. P. Patel, D. Puthal, and M. Prasad, "Visualization approach for malware classification with ResNeXt," IEEE Congress on Evolutionary Computation (CEC), pp. 1-7, July 2020.

[16] R. Mitsuhashi and T. Shinagawa, "High-accuracy malware classification with a malware-optimized deep learning model," arXiv preprint arXiv:2004.05258, 2020.

[17] M. Khan, D. Baig, U. S. Khan, and A. Karim, "Malware classification framework using convolutional neural network," International Conference on Cyber Warfare and Security (ICCWS), pp. 1-7, October 2020.

[18] 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, vol. 2022, no. 1, p. 4841741, October 2022.

[19] M. Nisa, J. H. Shah, S. Kanwal, M. Raza, M. A. Khan, R. Damaševičius, and T. Blažauskas, "Hybrid malware classification method using segmentation-based fractal texture analysis and deep convolution neural network features," Applied Sciences, vol. 10, no. 14, p. 4966, July 2020.

[20] A. Bensaoud, N. Abudawaood, and J. Kalita, "Classifying malware images with convolutional neural network models," International Journal of Network Security, vol. 22, no. 6, pp. 1022-1031, November 2020.

Malware Classification Using CNN Model_ A Comparative Study.docx

ORIGINALITY REPORT

20%	19%	9%	13%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	dspace.daffodilvarsity.edu.bd:8080 Internet Source	8%
2	Submitted to Daffodil International University Student Paper	3%
3	www.mdpi.com Internet Source	1%
4	Submitted to Huntington Beach Union High School District Student Paper	1%
5	Submitted to CSU Northridge Student Paper	1%
6	medium.com Internet Source	1%
7	Submitted to Liverpool John Moores University Student Paper	<1%
8	Submitted to University of Northumbria at Newcastle Student Paper	<1%