ENHANCED CONVOLUTIONAL NEURAL NETWORK FOR CLASSIFICATION OF MALWARES (E-CNN)

BY

KETEBU EBIEKINEN KENNEDY

ACE21140006

A Ph.D DISSERTATION SUBMITTED TO THE SCHOOL OF AFRICA CENTER ON EXCELLENCE OF TECHNOLOGICAL ENHANCED LEARNING, NATIONAL OPEN UNIVERSITY OF NIGERIA (NOUN) IN PARTIAL FULFILLMENT OF THE REQUIREMENT FOR THE AWARD OF DOCTOR OF PHILOSOPHY IN ARTIFICIAL INTELLIGENCE (Ph.D)

DECEMBER, 2023

ENHANCED CONVOLUTIONAL NEURAL NETWORK FOR CLASSIFICATION OF MALWARES (E-CNN)

BY

KETEBU EBIEKINEN KENNEDY

ACE21140006

A Ph.D DISSERTATION SUBMITTED TO THE SCHOOL OF AFRICA CENTER ON EXCELLENCE OF TECHNOLOGICAL ENHANCED LEARNING, NATIONAL OPEN UNIVERSITY OF NIGERIA (NOUN) IN PARTIAL FULFILLMENT OF THE REQUIREMENT FOR THE AWARD OF DOCTOR OF PHILOSOPHY IN ARTIFICIAL INTELLIGENCE (Ph.D)

DECEMBER, 2023

DECLARATION

I, **KETEBU EBIEKINEN KENNEDY**, hereby declare that this **Ph.D dissertation** titled Enhanced Convolutional Neural Network for Classification of Malwares has been carried out by me under the supervision Prof. Gregory O. Onwodi and Dr. Kingsley .E Ukhurebor. It has not been presented for award of any degree in any institution. All sources of information are specifically acknowledged by means of reference.

1/12/2023	
Date	Signature

CERTIFICATION PAGE

The research dissertation titled "**Enhanced Convolutional Neural Network for Classification of Malwares (E-CNN)**" meets the requirements governing the award of the Doctor of Philosophy (Ph.D) in Artificial Intelligence and is approved for its contribution to knowledge and literary representation.

DR. GREGORY. O ONWODI	13/12/2023	
Project Supervisor 1	Date	Signature

	09/12/2023	
DR. KINGSLEY. E UKHUREBOR	----------	----------
Project Supervisor 2	Date	Signature

DEDICATION

This research work is dedicated to God Almighty for His guidance and unending love towards me.

My Dad, Late Chief J.T Ketebu, My Mother, Mrs. Theresa N. Ketebu thank you for all your sacrifices in my life. God bless you all. Amen

ACKNOWLEDGMENT

This thesis was accomplished by the assistance of many individuals whom I wish to acknowledge. To my boss Isah J. Abdullahi for his support and guidance, my colleagues Edoh A. Okechalu, Joshua Mamza and Isiyaku B. Boyi

To my supervisors, Dr. Gregory .O Onwodi and Dr. Kingsley .E Ukhurebor thank you for reviewing this work, your understanding, time and energy spent in seeing me through to the final submission of this research work, may God Almighty bless you and your Family.

I want to thank Sydney, Vassal, Monique, Reagan, Comforter and my lovely wife Grace for encouraging and praying with me. Thank you so much!

May God's grace and blessings increase in all aspects of your lives, in Jesus Christ name Amen!

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

In our contemporary society, the widespread use of computer systems has become integral to daily life. However, this increased reliance on technology has also given rise to a surge in cyberattacks, threatening the integrity and security of our computer systems and networks. Malware, in particular, poses a constant and serious threat to people and organizations especially the ones connected to the internet. Various sectors such as universities, schools, hospitals, manufacturing, and healthcare providers, which depend on their information systems to support critical organizational and societal functions are constantly at risk or threat from malware attacks. The escalating frequency and sophistication of malware attacks, driven by the use of automation tools in malware creation, demand continuous efforts to develop efficient and effective means of detecting and classifying malware. Researchers have explored various techniques to counter these threats, with a growing focus on converting malware samples into images, a concept known as malware visualization. This research centers on the application of Convolutional Neural Networks (CNNs) for visual detection and classification of malware. E-CNN, our proposed malware classification model, is developed using a recent malware Malevis dataset. This is critical to address the ever-changing nature of malware, empowering the model to detect new and previously unseen threats. The E-CNN model was evaluated on the widely recognized Malimg dataset, achieving an impressive accuracy score of 98.88%. Furthermore, we compared the performance of the E-CNN model to other recently cited works, also using accuracy scores to assess its effectiveness.

CHAPTER ONE

1.0 INTRODUCTION

1.1 Background of the Study

Modern day computer attacks or cyberattacks are on the increase and becoming more complex, with the increasing number of computers and mobile devices connected to the internet or cyberspace, and users accessing various digital mediums or platforms. However, this has made users, computers, and networks vulnerable to various types of attacks on the internet or in a computer environment. Malicious software, also known as malwares, are harmful programs that expose or steal sensitive data or information, as well as compromising the integrity of a computer system by preventing it from operating securely.

Malware attacks are a major source of concern for individuals and cybersecurity experts, as they have resulted in denial of services, loss of privacy, intellectual property and financial losses for victims and organizations all over the world. Machine learning techniques have been effective in malware detection and classification; however, attackers attempt to disguise malwares as legitimate files using techniques such as packing, encryption, and polymorphism; this may result in the pre-trained model making incorrect predictions (Agrawal & Khan, 2021). Recent malware attacks have become more sophisticated as a result of the use of machine learning; it is estimated that at least 230,000 malware samples are produced every day, and 18 million websites are infected with malwares each week (Ghosh, 2021).

In developing effective malware detection and classification engines, there are two main methods which are namely static and dynamic analysis. Machine learning approaches has been applied in malware analysis, the two main approaches which are static and dynamic approaches differ from each other in the manner in which features are extracted (Gibert et al., 2020) statically or dynamically (runtime execution).

There are various similar variations of malwares samples, this is because malware authors reuse the previous codes only making changes so as to form or develop new malware samples (Moussas & Andreatos, 2021).

Newer methods are being developed towards malware classification and detection, one of such areas is the area of visualization, visualization techniques has been effective in enabling and understanding complex data analytics (Keahey, 2013) or structures.

One of the first researcher to use visualization was (Nataraj et al., 2011) who represented malware samples as grayscale images in order to distinguish similarities and differences among malwares samples. This showed visual similarities of malwares belonging to the same family.

Visualization can greatly aid malware classification and does not necessitate any disassembly (static analysis) or code execution (dynamic analysis) (Moussas & Andreatos, 2021). This is because performing feature extractions of malware samples requires a degree of expert domain knowledge when performing either static or dynamic analysis.

However, Researchers have found that deep learning can produce better performance when compared to existing machine learning approaches or methods (Kalash et al., 2018). Deep learning computational approach has been successful in solving complex computational tasks, this is because of its ability to learn massive amounts of data thereby outperforming other machine learning techniques in different domains such as bioinformatics, language processing, cybersecurity, robotics, control systems and many others (Alzubaidi et al., 2021).

One of the main advantages of deep learning is its ability to execute feature engineering on its own by scanning the dataset for features that correlate with each other so as to enable faster learning.

Due to the increase rate of malware attacks, it has become very critical in how malwares are quickly identified and classified. Traditional machine learning methods are limited by feature engineering and size of data being processed, hence Deep learning has become an effective solution (Yuan, Wang, Liu, Guo, Wu, Bao, et al., 2020).

Recently, researchers have begun the use of deep learning models towards classification of malware (Cakir & Dogdu, 2018) which also has a higher predictive accuracy. Hence deep learning using neural network model is being applied towards classification of malwares. At the current rate deep learning neural network has grown to the state to surpass the limitation of machine learning techniques, this is because of the possibility of using deep learning to develop models with significantly higher number of diverse layers (Kolosnjaji et al., 2016).

1.2 NEURAL NETWORK

Neural networks are a set of algorithms which are modelled after the human brain to recognize patterns and relationship data. Neural network is also called artificial neural network (ANN) in artificial intelligence. Artificial neural network consists of artificial neurons or nodes

interconnected together with each neuron able to receive input signal, process them and send an output signal (Zhang, 2018). ANN are made up of three basic components input, hidden and output layers. The units of a layer relates with other units in the layers by connection with each connection having different numerical weight or strength (Abiodun et al., 2018). The operation of neural network is divided into two main stages namely, training and inference stage. In the training stage, the network learns from a given labelled dataset by adjusting or varying the weights of its connections so as to minimize or reduce the difference between predicted outputs and expected outputs. The network can also undergo a process called backpropagation in which the network calculates the error gradient and then updates the weights using optimizations algorithms, like gradient descent. The second stage which is the inference stage, involves using the already trained model to make predictions on new unseen data. The input data is fed into the input layer which then propagates through the hidden layers and produces and output at the output layer. Neural networks can be applied to various task such as speech recognition, image classification, natural language processing, object detection etc.

The ability of neural networks to automatically learn and extract complex patterns from large datasets has led to their phenomenal success in a variety of fields. They can handle problems that were previously challenging or impossible to solve using conventional programming techniques. Image and speech recognition, autonomous vehicles, medical diagnosis, recommendation systems, and natural language processing are a few notable applications of neural networks.

1.3 DEEP LEARNING

Deep learning is an artificial neural network (ANN) which consists of complex multilayers (Albawi et al., 2017). Deep learning is a sub field of artificial intelligence that enables computer to learn and understand complicated concepts by building hierarchy layers.(Goodfellow et al., 2016). Deep learning accepts inputs and the inputs are then passed through multiple layers which learns and extract complex patterns and representations from data. Deep learning are also called deep neural networks (DNN).

The term "deep" in deep learning refers to the depth of the neural network, which means it has multiple layers of artificial neurons. These networks are often referred to as DNNs or deep learning models. Traditional neural networks typically have only a few layers, whereas deep learning models can have dozens or even hundreds of layers, enabling them to learn hierarchical representations of data (LeCun et al., 2015).

The key advantage of deep learning is its ability to automatically learn and discover features or representations directly from raw data without the need for manual feature engineering. This is in contrast to traditional machine learning approaches that rely on handcrafted features. Deep learning models are capable of learning hierarchical representations by progressively extracting more abstract and complex features at each layer (Goodfellow et al., 2016).

Deep learning models are primarily trained using a technique called backpropagation, which involves iteratively adjusting the weights of the connections between neurons to minimize the difference between the predicted outputs and the desired outputs.

DNN have shown to be very good at complicated machine learning tasks like image classification and speech recognition. However, because of their multilayer nonlinear nature, they are opaque, making it difficult to understand how they arrive at a specific categorization or recognition (Samek et al., 2016) or learning task. Additionally, deep learning models often require a large amount of labelled training data and extensive computational resources for training, making them computationally expensive.

There are different types of deep learning architecture which are namely convolutional neural networks (CNN) for image processing, recurrent neural network (RNN) for sequential data analysis, long short-term memory network (LSTM) for language modelling and speech recognition, generative adversarial networks (GAN) for generating synthetic data Multilayer perceptron and more.

1.3.1 Convolutional Neural Network (CNN)

A CNN is a type of deep learning model specifically designed for analysing visual data, such as images and videos. The research in the area of CNNs have swiftly emerged by achieving state-of-the-art result in various computer vision tasks (Gu et al., 2018) such tasks includes image classification, object detection, segmentation, and other visual recognition tasks. CNN is a type of deep learning neural network which consist of multiple layers which is used to train dataset with very large parameters or features, this is done by converting the datapoints to images as input and combing with filters to produce a desired output (Chauhan et al., 2018). The process of training a CNN involves forward propagation and backpropagation. During forward propagation, input data is passed through the layers of the network, with the weights and biases adjusted to produce an output. The difference between the predicted output and the actual true result is measured using a loss function, such as categorical cross-entropy or mean squared error. CNN which has shown excellent performance in machine learning problems, has multiple layers

which includes convolutional layers, pooling layers, non-linearity layers and fully connected layers (Albawi et al., 2017).

CNN has two main process which are the convolution process and the sampling process. In the convolution process, the input features are applied to matrix filter in each layer to extract meaningful features, next the sampling or pooling process downsizes or compresses the feature maps while maintaining important features of the image and thereafter passed into fully connected layers.

1.4 MALWARES

Malicious software, sometimes known as malware, is damaging to computer systems because of its inherent ability to steal, damage, and interrupt computer networks and resources without the awareness of users (Tahir, 2018). Malwares are designed specifically to exploit vulnerabilities or gain unauthorized access to computer systems or networks. As a result of the rapid growth of different types of malwares, malwares were grouped into first- and second-generation malwares, while first generation malware are grouped on the basis of the manner of infection on the target system. The second generation malwares changes its structure during execution (Sahay et al., 2020).

The some of the different types of first-generation malwares are as follows: -

Worms: this is a type of malware which has the ability exist independently as a standalone program and can replicate itself across computer systems and networks thereby resulting in performance degradation (Tahir, 2018). Worms exploit security vulnerabilities to automatically propagate from one system to another, often causing network congestion and consuming system resources. Worms can also carry payloads, such as other malware or malicious activities.

Viruses: viruses affect computer systems by attaching itself to legitimate programs, executables or any file and replicates itself across a computer network, this affects the performance of the computer systems and also the network. Viruses which are self-replicating in nature causes damage by corrupting or deleting files, disrupting system operation and also stealing of sensitive information

Trojan horse: Trojan horse is a malicious code which hides it true nature of operation so as to perform a wide range of attacks on computer resources or network. Trojan horses are a common means of network attack (Yu et al., 2019). Trojan horses, are sometimes called Trojans these are deceptive programs that masquerade as legitimate software or files. Once executed, they can

perform various malicious activities, such as stealing personal information, providing unauthorized remote access to the system, or downloading and installing additional malware.

Rootkits: these are malicious modules which is loaded into the operating system (OS) kernel, which grants the module elevated privileged to perform other malicious activities such as control of the system, process hiding, information gathering and even spread of malwares.

Bots: this is a type of malware which infects computer or group of systems which enables the attacker to control the computer systems remotely from a central command and control server so as to launch cyber-attacks such as distributed denial of services DDOS, sending of spam mails or distribution of additional malwares. A network of bots-controlled computers is called botnets.

Keylogger: this is a malicious tool which is mostly installed without the knowledge or permission of the user. Keylogger saves all keystroke generated by the user through the machine so as to monitor and steal vital or sensitive information without user's consent.(Wajahat et al., 2019)

Ransomware: this is a type of malware which encrypts or locks data of a victim computer by performing a significant number of file related operation in a short period of time (Bae et al., 2020) and may be released upon payment by victim. It often spreads through phishing emails, malicious downloads, or exploit kits. Ransomware attacks have become increasingly prevalent, targeting individuals, businesses, and even critical infrastructure.

Spyware: Spyware is a type of malicious tool which keep tracks of all the user's activities performed on the computer and the information sent back to hacker or creator. Spyware is designed to covertly gather information about a user's activities and transmit it to a third party without the user's consent. It can track keystrokes, capture screenshots, monitor browsing habits, and collect sensitive data like login credentials or credit card information. Spyware is often used for surveillance, identity theft, or unauthorized advertising purposes.

Adware: Adware, short for advertising-supported software, is typically not as malicious as other forms of malware. It displays unwanted advertisements or redirects users to advertising websites, often bundled with free software downloads. While adware may be more of a nuisance, it can impact system performance and compromise user privacy.

The second generation of malware represents a significant evolution in malicious software, introducing more advanced techniques and capabilities compared to its predecessors. This generation of malware emerged in the late 1990s and early 2000s, building upon the foundation

laid by the first generation of viruses and worms. This second-generation type of malware has the ability to change or hides its structure or conceals by encrypting its true nature of operation so as to evade detection by malware detectors. They can be classified into polymorphic, oligomorphic, encrypted, blended threat malwares and metamorphic malwares.

Encrypted malwares: malware creators use encryption and decryption methods, so as to avoid detection and static code analysis. Recently, Transport layer security (TLS) protocol which is widely used in securing application data, is now being used by malware authors to encrypt malware traffic thereby making malware detection such as deep packet inspection (DPI) ineffective (Liu et al., 2019).

Oligomorphic malwares: This is comparable to encrypted malware, but it differs in that each new infection or attack requires a unique decryptor, which is chosen from a list of decryptors. This makes it difficult for anti-malware engines to detect it; nevertheless, if the anti-malware engine scans all existing decryptors, detection is still feasible.

Polymorphic malwares: The high number of distinct malware samples found each day shows that there is likely a lot of code reuse going on beneath the layers of stealth (Deng & Mirkovic, 2022). These types of malwares can appear to be unique but it functionalities is the same as other malware samples such malwares are polymorphic in nature.

Metamorphic malwares: This is a self-modifying malware that alters the structure of its code while maintaining its functioning so as to evade detection (Mumtaz et al., 2021). Metamorphic malware may or may not need decryptors to appear as a unique malware sample. This is because of its mechanism which changes its syntax after each copy, however its mode of attack or workings does not change.

Blended Threats: Second-generation malware introduced the concept of blended threats, which combined multiple attack vectors or malicious functionalities. For example, a malware program might combine a worm to spread itself, a Trojan to perform unauthorized actions, and a rootkit to hide its activities. Blended threats increased the sophistication and effectiveness of malware attacks, making them more potent and harder to combat.

Advanced persistent threats (APTs) Malwares. These type of malware are designed specifically to infiltrate and compromise specific persons or organizations using multi-steps process over a period of time (Rot & Olszewski, 2017). This advanced type of malware uses

complex tools such as zero-day exploits and social engineering so as to make its attacks more effective.

It's important to note that the field of malware is constantly evolving, with new types and variants emerging regularly. As such, staying informed about the latest threats and implementing appropriate security measures is crucial to protect against these malicious programs.

1.5 PROBLEM STATEMENT

The level of sophistication of malwares is on the increase as well as the rate of malware attacks on computer systems and networks with increase in internet use, malwares pose serious threat to the digital world and its impact severe. Malware developers or attackers are employing different techniques which makes malwares evade detection and classification hence causing serious harm to information, computers and networks. Researchers are developing new methods in detection and classification of malwares with high accuracy which has been effective. One of such technique is the use of CNN in developing models. This works attempts to develop a model which will be an improvement of the existing method CNN, by improving upon existing techniques, this research strives to contribute to the advancement of malware detection and classification by better predictive performance in terms of efficiency and accuracy.

By addressing the limitations of current approaches and leveraging the potential of CNNs, this study aims to provide a valuable contribution to the field of malware detection and classification. The outcomes of this research have the potential to significantly impact the effectiveness and efficiency of combating malware threats in the digital landscape.

1.6 AIM AND OBJECTIVES OF STUDY

The main aim of this desertion is to develop a CNN for accurate malware classification. The proposed model will be trained and feature extraction will be performed on image representations of binaries or executables, resulting in improved predictive performance and results.

The objectives of the research work are outlined as follows:

1. Develop an effective deep CNN model specifically designed for malware detection and classification. By leveraging the power of deep learning techniques, the model will be able to effectively identify and categorize different types of malwares based on their binary or executable representations.

2. Utilize existing pre-trained deep learning architectures and apply transfer learning methods to train the model for malware classification. Transfer learning allows the model to leverage knowledge gained from training on large-scale datasets and adapt it to the task of malware classification. By utilizing pre-trained architectures as a starting point, the model can benefit from the learned features and accelerate the training process.
3. Compare and analyse the results obtained from the newly developed deep CNN model with the models developed using transfer learning methods. By evaluating the performance of different models, this research aims to determine the most efficient and accurate approach for malware classification. The analysis will consider factors such as classification accuracy, computational efficiency, and overall effectiveness.
4. Applying the newly developed model on benchmark dataset and compare its performance with other researchers work to access its effectiveness and benchmark against state-of-the-art approaches.

By achieving these objectives, this research aims to contribute to the advancement of malware classification techniques. The developed deep CNN model, along with the comparison and analysis of different approaches, will provide valuable insights into the most effective methods for accurately identifying and categorizing malware.

1.7 SCOPE OF STUDY

This study focuses on developing a novel CNN model and conduct a comprehensive comparison of its results and performance with existing pre-trained deep learning CNN architectures. The focus will be on applying the models (new and pre trained architecture via transfer learning techniques) to a dataset. The recent dataset utilized in this study comprises of thousands of recent image representation of executable, encompassing both malicious or malware samples and benign programs.

By focusing on the development of a new CNN model and comparing it with an established pre-trained CNN architecture, this study seeks to advance the field of malware classification. The evaluation and comparison of these models' performance will provide valuable insights into the effectiveness and potential improvements in identifying and classifying malware. The dataset, consisting of diverse and up-to-date image representations of executables which will ensure a comprehensive evaluation of the models' capabilities in handling real-world malware samples.

1.8 SIGNIFICANCE OF THE STUDY

This research holds significant importance in the field of malware detection and classification by introducing a newly developed model based on CNN. Through the application and comparison of this model, valuable insights will be gained, aiding malware researchers in identifying the most effective convolutional neural network models for malware detection and classification.

The developed models have practical implications as they can be leveraged by cybersecurity analysts to create robust malware detection and classification tools. By incorporating these models into existing security systems, computer systems can be effectively shielded from the threats posed by malicious programs or codes. This contributes to enhancing information security and reinforces the defence mechanisms of computer networks against evolving malware attacks.

Moreover, the research outcomes have the potential to drive advancements in the field of cybersecurity. By identifying the best performing CNN models for malware detection and classification, future research and development efforts can be directed towards refining and optimizing these models, leading to more accurate and efficient malware identification techniques.

Ultimately, the significance of this study lies in its contribution to the improvement of malware detection and classification methodologies, strengthening the defence against malware and fostering a more secure digital environment.

1.9 OUTLINE OF DISSERTATION

- Chapter 1: Introduction
- Chapter 2: Literature Review
- Chapter 3: Research Methodology
- Chapter 4: Results and Analysis
- Chapter 5: Conclusion

1.10 SUMMARY OF CHAPTER

This chapter delves into the evolving threats and the severe consequences associated with malware attacks on digital data, computer systems, and networks. Despite notable achievements in malware detection and classification, malware creators continuously strive to enhance the

sophistication of their malicious software. They employ a wide range of techniques to evade detection and inflict significant damage on digital information and its associated resources.

To address this ever-evolving landscape of malware, researchers are consistently refining existing methods of malware detection and classification while also exploring new techniques. This research places particular emphasis on leveraging deep learning convolutional neural networks (CNNs) to develop an effective model capable of efficiently analysing, detecting, and classifying malware with substantially higher accuracy than previous approaches.

By employing CNNs, which have demonstrated success in various domains, the research aims to push the boundaries of malware detection and classification. The focus lies on developing a robust model that can adapt to the evolving nature of malware, enabling accurate identification and classification even in the presence of sophisticated evasion techniques.

The outcomes of this research have the potential to significantly enhance the field of malware analysis by providing an advanced model capable of tackling the ever-increasing challenges posed by malware creators. By improving the accuracy and efficiency of malware detection and classification, the research contributes to bolstering the security of digital systems and networks, safeguarding valuable information from malicious attacks.

CHAPTER TWO

2.0 LITERATURE REVIEW

This chapter provide an in-depth examination of researchers' works and techniques for malware classification using deep learning methods. In addition to exploring these approaches, it is also necessary to briefly discuss malware analysis and detection techniques. This is due to the fact that understanding how malware analysis and detection are carried out aids in understanding how well malware classification models perform.

2.1 MALWARE ANALYSIS

Malware analysis is simply a process of analysing malware samples so as to determine its method of operation (functionality, behaviour and impact) on computer systems and network. This is done by extracting information about the malware samples, the information extracted helps in understanding the nature and scope of functionality of the malware sample. Malware analysis helps in categorizing the type of malware sample, i.e. whether the sample is a botnet, virus, ransomware etc. Malware analysis is a vital process towards developing effective detectors this is because useful information (registry keys, filenames, signatures) which are extracted and studied by researchers towards improving and making better future detectors.

TYPES OF MALWARE ANALYSIS

Various techniques are employed to analyse malwares into different categories.

Malware analysis can be broadly classified into three types which are namely

1. Static analysis
2. Dynamic analysis
3. Behaviour analysis

STATIC ANALYSIS: in static analysis, the malware samples are analysed without executing or running the malware, however all necessary information about the malware is extracted. The extracted information can be used to form detection patterns. when static analysis is performed, file information such as string signature, opcode frequency, windows API, control flow graph (CFG), byte sequence n-grams are used as technical indicators for determining whether a file is malicious or not.

DYNAMIC ANALYSIS: in dynamic analysis, the malware sample is executed in a sandbox (safe and controlled) environment so as to analyse its functionality and behaviour during runtime using debugging tools. This is a method of analysis which gives malware researchers deep visibility or insights on the nature of (potential) threat or actions at runtime execution.

BEHAVIOURAL ANALYSIS: this involves analysing and interacting with the suspicious malware samples after execution. It involves the monitoring the processes, registries, memory usage, cpu usage, data transfer and other computer system resources so as to determine its method of operation of the malware sample. Behavioural analysis is a time consuming and complicated process which requires advanced skills.

2.2 MALWARE DETECTION TECHNIQUES

Malware detection refers to the process of identifying and detecting a (suspicious) file or program as malicious or benign on a system or network. Thereby preventing computer systems from incidents such as system compromise, data and information loss. Malware detection techniques can broadly be divided into three categories.

1. Signature based detection
2. Heuristic based detection
3. Specification based detection

SIGNATURE BASED DETECTION

In signature-based detection, the suspected file is disassembled into sequence of bytes which is known as a signature. This signature is then compared with an existing database of known malware signatures to determine if the file is malicious and which family of malware it belongs to. This detection method is usually used by most antivirus programs.

HEURISTICS BASED DETECTION

This is a behaviour-based method of detection in which differentiates between normal and abnormal behaviour of a system. Heuristics based detection process entails a detailed study or observation of the system in an idea condition and in absence of an attack which will be used as a baseline for comparison on the system in the event of a malware attack. This method is effective in detecting unknown malwares or new threats; however, it is a resource intensive method such as use of virtualized environment and usually prone to a high level of false positives.

SPECIFICATION BASED DETECTION

In specification-based detection, rule sets are defined which specifies the valid or intended behaviour exhibited by any program of the system. Specified based detection involves observing and monitoring programs executions so as to determine malicious activities by detecting deviations of their behaviour from previously specified rule sets. It overcomes the limitation usually faced by heuristics-based detection by reducing the level of false positive and increasing the level of false negative.

2.3 MALWARE NORMALIZATION

Malware writers and attacker use often use obfuscation techniques to hide or transform the program codes executables of the malware so as to hide malicious intent and evade detection. Hence the use of malware normalization systems which processes obfuscated program executables and eliminates the obfuscation so as to reveal the true nature of the program codes executables, this helps to improve detection rate. For malwares developed using toolkits (such as UPX, VirtTool, etc) normalization approaches can be employed to improve the detection rate of a malware detector (Dwivedi P & Sharan H).

2.4 DEEP LEARNING METHODS FOR MALWARE CLASSIFICATION

Deep learning, a subset of machine learning, has gained significant attention and shown promise in various domains, including malware classification. Deep learning models, such as CNNs and RNNs, have demonstrated remarkable capabilities in extracting intricate patterns and features from complex data, making them well-suited for malware analysis.

CNNs have been widely employed in malware classification tasks due to their ability to automatically learn hierarchical representations of data, particularly in image-based malware analysis. These models utilize convolutional layers to extract local features from images of malware binaries or executables, followed by pooling layers to capture high-level representations. The extracted features are then fed into fully connected layers for classification.

RNNs, on the other hand, are effective in capturing temporal dependencies and sequential patterns, which are valuable in analysing the behaviour and dynamic aspects of malware. These models, such as LSTM networks, can process sequences of system calls, network traffic, or other time-series data to identify malicious patterns.

2.4.1 CONVOLUTIONAL NEURAL NETWORKS (CNN)

CNN is a type of neural network which is frequently used in the field of computer vision for tasks such as image classification, object recognition and detection. CNN consists of multiple (input, hidden and output) layers of artificial neurons which processes images to identify unique patterns or feature representations. The convolutional layers learn feature representations by extracting local characteristics of from inputs or previous layers so as to obtain a new feature (Guo et al., 2017).

COMPONENTS OF CNN

CNN consist of components called layers. There are broadly three types of CNN layers, namely

1. Convolutional Layers
2. Pooling Layers
3. Fully connected layers

2.5 RELATED WORKS

This section provides a comprehensive analysis of various researchers' approaches in the field of malware classification using deep learning methods. By exploring the works of different researchers, a comprehensive understanding of the advancements and contributions in the application of deep learning for malware classification is gained.

Researchers (Meng et al., 2017) developed a model called malware classification model based on static malware gene sequences (MCSMGS), this model uses genetic theories to analyse malwares in which malware code fragments which carries functional information are referred to as malware gene sequences. The model extract API call sequences from malware gene which are converted into n by k two-dimensional matrix (where n is length of sequence and k is dimensional space) so as to represent intrinsic correlation and similarity. Thereafter CNN model is used for analysing and classification on the malware gene sequences. The result of the experiment achieves an accuracy of 98% on Microsoft challenge dataset.

(Kalash et al., 2018) proposed a deep learning framework for malware classification using deep convolutional neural network (CNN) architecture, which is referred to as M-CNN model. The model processes grayscale images of binaries from two datasets (Malimg and Microsoft malware dataset). The result of the experiments achieved an accuracy score of 98.52% and 99.7% on Malimg and Microsoft malware dataset respectively.

In a research work by (Le et al., 2018), a model was developed which combines a convolutional neural network plus two bi-directional long short term memory architectures (CNN-BiLSTM) for malware classification. A generic image scaling algorithm which interprets the malware file byte code as a one-dimensional image with a fixed target size. The generated images is fed to the CNN-BiLSTM model in which the output of convolutional layers are connected to one forward LSTM layer and one backward layer. The two outputs are then fed to the output layer of the model, the result of model achieved an average accuracy score of 98.8%.

In a research work by (Lo et al., 2019), the researchers performed malware classification using a special CNN architecture Xception model based which its experiment was based on Malimg and Microsoft malware dataset. This approach performs malware classification using two file types (.byte and .asm) in which the predictions are stacked together so as to give a predictive result. This helps to reduce overfitting problem as well as achieved a very high accuracy 99.03%. The Xception model was very effective and less time consuming when compared to other methods such as KNN, SVM and VGG16.

In a research work by (Khan et al., 2019), the researchers based their research work on two pre trained architectures which are GoogleNet and ResNet152. These architectures were applied on the Microsoft malware classification challenge dataset which contains malware binaries which are converted to images. GoogleNet was the fastest among the two models achieving an accuracy score of 74.5% while ResNet152 achieved an accuracy score of 88.36%

A model framework was proposed by researchers (Yuan, Wang, Liu, Guo, Wu, & Bao, 2020) to improve malware classification accuracy using markow images. This model was called byte level malware classification method based on markov images and deep learning (MDMC). This entails converting malware binaries into markov images using markov transfer probability matrix so as to retain global statistics of malware bytes. The generated markov (malware) images has a fixed sized which reduces redundancy of bytes information. The structure of convolutional neural network (CNN) is based on VGG16. The experiments were conducted on two datasets which are Microsoft and Drebin malware dataset. The average accuracy rates where 99.264% (Microsoft dataset) and 97.364% (Drebin dataset).

(Nisa et al., 2020) proposed a hybrid method of malware classification which involves a combination of pre-trained deep convolutional neural network model (Alexnet and Inception-

v3) and scaled feature texture analyser (SFTA) which are used for feature extraction, the results of the feature extraction are then combine into a single feature vector using serial-based feature technique, while using principal component analysis (PCA) for selection the most informative or relevant features. The result of the experiment when applied on the Malimg image dataset achieved an accuracy of 99.3%

A research work by (Bensaoud et al., 2020), the researchers selected six deep learning models for static malware classification in which three models were combine with support vector machines algorithm(SVM) to enhance the neural network models which are MLP-SVM, CNN-SVM, and GRU-SVM. The experiment was performed on the Malimg dataset which contains images of converted malware binaries. The results showed that the pre-trained architecture model Inception-V3 achieved an accuracy score of 99.24%

Researchers (Yoo et al., 2021), proposed a machine learning hybrid model called the Al-Hydra, this model combines random forest (RF) and Multi-layered perceptron (MLP) which are very effective for malware detection. This model which consists of four sub classification models (static RF, Dynamic RF, static MLP and Dynamic MLP) uses a voting scheme in which a rule-based majority vote is used to determine if a sample is malicious or benign. The results of the experiment showed Al-Hydra having an average accuracy of 85.1% using KISA dataset.

A research work by (Kumar, 2021), who developed a model using transfer learning called malware classification with fine-tune convolutional neural networks (MCFT-CNN). This model was developed by altering the last layer with a fully connected dense layer of a pre-trained existing model ResNet50. The MCFT-CNN model when trained with Malimg dataset achieve an accuracy of 99.18% and 98.63% on Microsoft malware challenge dataset.

In another study, (Awan et al., 2021) proposed a model based on deep learning framework called spatial attention and convolutional neural network (SACNN) for malware classification. This model represents a simple solution which does not require generated images from binaries to undergo special preprocesing operations such as data augmentation or feature engineering in order to solve malware classification problems. The model consists of a transfer learning model (VGG19), a dynamic spatial attention mechanism which focuses on only important areas of the generated images for malware classification. The result of experiment when applied to Malimg malware dataset produced an accuracy of 97.68%.

Researchers (Prajapati & Stamp, 2021) conducted CNN experiments in which transfer learning played a vital aspect. The pre trained models used are VGG-19 and ResNet152. The dataset used consist of 20 different malwares families and is a combination of the Malicia dataset and the Microsoft dataset. The malware samples are converted in images and the early portion or layers of the models frozen while the last few parts of the layers are retrained. The result of the experiment achieved and accuracy score of 92.16% for VGG19 while the ResNet152 was 91.50%.

(Asam et al., 2021) proposed a malware classification framework called Deep Feature Space-based malware classification (DFS-MC), the proposed model entails customizing and fine tuning ResNet-18 and DensNet-201 in combination with SVM. The hybrid model learning scheme involves extracting deep ensemble features of customized CNN models and then applying SVM classifier for malware classification on deep ensemble feature space. The proposed model produced an accuracy of 98.61%.

Researchers (Carletti et al., 2021), carried out an evaluation so as to determine the robustness of CNN for malware classification. This was done by specializing existing CNN models (ResNet50, InceptionV3, MobileNet and VGG16) on malware images through transfer learning for malware classification. In accessing the robustness of the models, the malware samples input are perturbed which involves subjecting the original executables through obfuscation methods. A metamorphic technique such as dead code insertion was applied directly on the hexadecimal representation of a binary file, this involves inserting junk codes into the text section of the binary file. The BIG2015 dataset used in the experiment with the experiment being in two folds with malware classification on the original dataset and accessing robustness on obfuscated dataset. The overall best CNN model was MobileNet which a high accuracy score of 99.25% and on obfuscated dataset 96.2% showing the CNN model is very robust for malware classification.

(Schofield et al., 2021) Presented a CNN model malware classification based on Windows system Application Program Interface (API) call. The researchers identified API call sequences as an important feature for malware classification, this is because API calls shows system calls or events on windows operating system occurring during runtime of a malicious file sample. The research work used a database of API call streams. The model uses both one-dimensional (1-D) CNN and term frequency-inverse document frequency (TF-IDF) in mapping API call streams. The result of the experiment showed the 1-D CNN model achieving an accuracy score of 98.17%.

In a recent study, researchers (Marin et al., 2022) developed a model for malware classification, the experiments consisted of two dataset which are the Malimg and the Microsoft challenge dataset. This involves the dataset being converted, processed and resized to a define size (64x64 pixels) so as to enable the model make accurate classification. Experimental results showed that model was efficient and effective with an accuracy score of 98.70% on both datasets.

Researchers (Lin & Yeh, 2022) proposed a bit and byte-level sequence one dimensional (1D) CNN model which extracts vital features from the one dimensional structure of binary executables, instead of converting executables into two dimensional images (2D) which makes it difficult to determine a fixed width with all inherited sequential structures within the byte-level sequence. Resizing and compression methods are applied to fix the length of each byte-level sequence, additionally bit transformation is applied so as to expand the byte-level to bit level sequences. This is because each machine instruction is encoded as 8 bits. The model maintains the contextual information for the machine instructions and also has fewer number of parameters in comparison to 2D CNN models. The model when applied to Microsoft malware challenge dataset achieved an accuracy score of 98.7% for malware classification.

In a research work by (O'Shaughnessy & Sheridan, 2022), one area of concern was malware developers employing obfuscation techniques so as to evade detection. Hence a hybrid framework for malware classification was developed to overcome the challenges faced by other image-based malware classification models. This framework combines the strengths of both static and dynamic analysis to overcome obfuscated malware samples. This is done by converting malware samples into two dimensional images mapped through space filled curve (SFC) traversals. This is important because the data structures of resulting SFC images of original malware samples are maintained after conversion. The result of the experiment when applied to the dataset gave an accuracy score of 97.6%.

Researcher (Alshamrani, 2022) developed a novel approach using deep learning to categorize malwares families and multi classification. This was done by converting malware samples into sequence of pixel values producing two-dimensional matrix grayscale images. The CNN model uses entropy filters to find distinct patterns in the image processed. The performance of the CNN model was evaluated using malware dataset of 10,000 samples with nine classes. The result of the model achieved and accuracy score of 99.7%.

A research work by (Onoja et al., 2022) developed a malware detection and classification model whose goal was not only to enhance effective detection of malware but also to reduce the

prediction time. This was achieved by proposing a hybrid model which integrates XceptionCNN with LightGBM algorithm. The model was applied on the Malimg dataset which contains 9339 gray scale images of malware sample of 25 different classes of malware and 1042 benign samples. The model achieved and accuracy of 99.85% for binary classification and 97.40% for multi-classification.

Researchers (Hammad et al., 2022) developed a model for malware classification, which performed best among the experiment conducted. The proposed model involved using deep feature technique (GoogleNet) to extract features from the Malimg dataset, while KNN is used for classification. This achieved the highest accuracy score of 96.64%.

In a recent study, researchers (Ahmed et al., 2023) formulated malware signatures as 2D image representation in classifying malwares using deep learning techniques on Microsoft malware challenge BIG 2015 dataset which contains malware samples. The model was developed using transfer learning of Inception V3 architecture and its performance produced a classification accuracy score of 98.76%. The research work compares its performance with various machine learning and deep learning technologies towards malware classification such as Logistic Regression (LR), Artificial Neural Network (ANN), Convolutional Neural Network (CNN), transfer learning on CNN and Long Short-Term Memory (LSTM).

Table 2.1: Table showing summary of malware classification models

S\N	Author	Classification Model	Dataset	Accuracy	Limitation
1	(Meng, Shan et al. 2017)	MCSMGS	Microsoft challenge dataset (BIG 2015)	98.0%	The dataset used is limited due to its robustness
2	(Kalash et al., 2018)	M- CNN	Malimg Microsoft challenge dataset (BIG 2015)	98.52% 99.7%	The dataset lacks robustness
3	(Le et al., 2018)	CNN-BiLSTM	Microsoft challenge dataset (BIG 2015)	98.8%	Slow training time, Imbalance dataset
4	(Lo, Yang, & Wang, 2019)	Xception	Malimg Microsoft challenge dataset (BIG 2015)	99.03% 99.97%	The dataset lacks robustness

5	(Khan, Zhang et al. 2019	GoogleNet ResNet152	Microsoft challenge dataset (BIG 2015)	74.5% 88.36%	The dataset used lacks robustness
6	(Yuan, Wang, Liu, Guo, Wu, & Bao, 2020)	MDMC model	Malimg Derbin Dataset	99.264% 97.364%	Processing time The dataset lacks robustness
7	(Bensaoud, Abudawaood et al. 2020)	Inception V3	Malimg	99.24%	Dataset imbalance, high computation time
8	Yoo, Kim, Kim, & Kang, 2021	Al-hydra	KISA Dataset	85.1%	Uses high computation to extract various features, Uses voting mechanism to decide classification, this is suspectiple to high false (FP) rate of begnin samples
9	(Prajapati & Stamp, 2021)	VGG19 ResNet152	Malicia Dataset	92.16% 91.50%	High computation time, dataset imbalance
10	(Kumar, 2021)	(MCFT-CNN).	Microsoft challenge dataset (BIG 2015) Malimg	98.63% 99.18%	Due to the size of complex architecture of resnet50 has high computation overhead
11	(Awan et al., 2021)	SACNN	Malimg	97.68%	Dataset imbalance, lack of the exploration in the data augmentation and the feature engineering domains.
12	Asam, Khan, Jamal, Zahoora, & Khan, 2021	DFS-MC	Malimg	98.61%	Very large processing and computation cost
13	Carletti, Greco, Saggese, & Vento, 2021)	MobileNet Xception XGBoost	Microsoft challenge dataset (BIG 2015)	99.25% 99.07% 99.43%	Accuracy of models drops considerable on obfuscated samples
14	(Lin and Yeh 2022	Byte-level 1D CNN	Microsoft challenge dataset (BIG 2015)	98.7%	Model did not always produce better performance while binary executables were converted and resized to larger images.
15	(O'Shaughnessy & Sheridan, 2022)	SFC KNN-HOG	VirusTotal dataset	97.6%	Long conversion time of malware samples to SFC images
16	Onoja, Jegede et al. 2022	Xception+LightGBM	Malimg	97.40%	The dataset lacks robustness, imbalanced dataset
17	Hammad, Jamil et al. 2022	GoogleNet+KNN	Malimg	96.40%	High computation time
18	(Alshamrani, 2022)	Binary code to pixel vector transformation	Microsoft challenge dataset (BIG 2015)	99.97%	The dataset lacks robustness, model susceptible to overfitting
19	Ahmed, Afreen, Ahmed, Sameer, & Ahamed, 2023	Inception V3	Microsoft challenge dataset (BIG 2015)	98.76%	The dataset lacks robustness

After a comprehensive examination of various researchers' works, it becomes clear that some of these models have been developed by adapting existing architectures through transfer learning, while others are created by combining two or more architectures or crafting entirely novel models from scratch. In Table 2.1, we provide a summarized overview of crucial information, including the dataset used, accuracy scores, and limitations of the models discussed in Section 2.5. These advancements play a pivotal role in enhancing the effectiveness and adaptability of solutions for detecting and classifying malware, ultimately strengthening the security of computer systems and networks.

2.6 RESEARCH GAPS

A notable research gap in the field of malware classification can be identified regarding the utilization of recent and evolving malware datasets in developing models. Many researchers have relied on existing works that employ datasets that may not encompass the most up-to-date malware samples. This limitation arises from the difficulty in obtaining access to publicly standardized or benchmarked datasets containing recent and emerging malware samples. However, there is a clear need to address this gap by developing models that are trained on recent malware datasets, providing a representation of the ever-evolving nature of malware threats.

The absence of up-to-date datasets poses a challenge in evaluating the performance and effectiveness of malware classification models in real-world scenarios. As malware constantly evolves and adapts to evade detection, it is crucial to train models on datasets that encompass the latest malware samples. This ensures that the models are equipped to accurately identify and classify the most recent and sophisticated malware variants.

CHAPTER THREE

3.0 RESEARCH METHODOLOGY

In other to comprehend the proposed research methodology, this chapter discusses concepts, theories and technologies as well as other vital information necessary to understand the suggested approach of malware analysis and classification on malware image dataset using a deep convolutional neural network architecture (CNN). Also, this methodology will analyse different learning rates, as well as optimizing the convolution layers, pooling layers, activation function and other optimal parameters which will be effective for developing a model classification of malware images. The methodology of this study is divided into data gathering, CNN architecture design and classification.

3.1 CONVOLUTIONAL NEURAL NETWORK (CNN)

Convolution is simply featuring transformation on given image which brings out or shows hidden patterns when passed through a filter (kernel). The filter finds or identifies patterns when applied on the image. With the discovery of high-level features, more abstract interpretation of the image dataset can be found. The use of CNN models is mainly applied in solving computer vision problems such as object tracking, object identification, image identification, feature identification etc.

CNN is among the best state of the art neural network architecture used for image classification tasks or problems. CNN consist of various components or layers which is illustrated in Figure 3.1 however, the three main components are namely convolutional layer, pooling layers and fully connected layers.

3.1.1 CONVOLUTIONAL LAYERS

Convolutional layers are usually one of the first layers and also are foundational or core building blocks of the convolutional neural network. Convolution is simply a mathematical procedure that requires two inputs, like an image matrix and a set of filters whose parameters must be learned.

These layers consist of many filters, which are also referred to as kernels (matrix) which extracts local or regional characteristics features from the input images. The filters which usually have a much smaller spatial dimension than the input images are then extended across the input images

or previous feature maps in sequence, this is referred to as a stride, thereafter and then passed through an activation function which forms a feature map.

As mentioned, this layer which consists of a set of filters or kernels produces an output as a result of a linear operation (i.e. linear multiplication of matrices weights and inputs) and then stacking of feature maps of all filters along the depth dimension of the image.

3.1.2 POOLING LAYERS

The primary objective of the pooling layer is to gradually reduce the spatial (parameter) size or dimensions of the input. Consequently, by doing so, the number of parameters to learn and the amount of computation are both reduced. This process is done by down sampling feature maps which simply involves summarizing vital features in patches or regions of the feature maps. The addition of a pooling layer after the convolutional layer is a common technique used for ordering layers within a convolutional neural network that may be performed one or more times in a given model. The pooling layer operates on each feature map separately to generate a new set of the same number of pooled feature maps. There are three main types of pooling functions which are namely max pooling, average pooling and global pooling.

3.1.2.1 MAX POOLING

This is a type of mathematical operation which selects the highest or maximum element value from the region of the feature map covered by the filter. Hence the output of the pooling layer will produce a feature map which contains the most important features of the feature map.

3.1.2.2 AVERAGE POOLING

This is a type of mathematical operation which is achieved by selecting the average element value from the region of the feature map covered by the filter. This involves calculating the average value from a portion of the image using a specific size.

3.1.2.3 GLOBAL POOLING

This is another type of pooling which is used to down sample a feature map into a single value using either global max pooling or global average pooling. In some CNN models global pooling is often used as a replacement for fully connected layer.

3.1.3 FULLY CONNECTED LAYER

This is usually the last layer of the CNN architecture which receives its input as an output from the pooling layer which is a representation of high-level features of the image, are then flattened into a vector so as to perform classification operation based on features extracted.

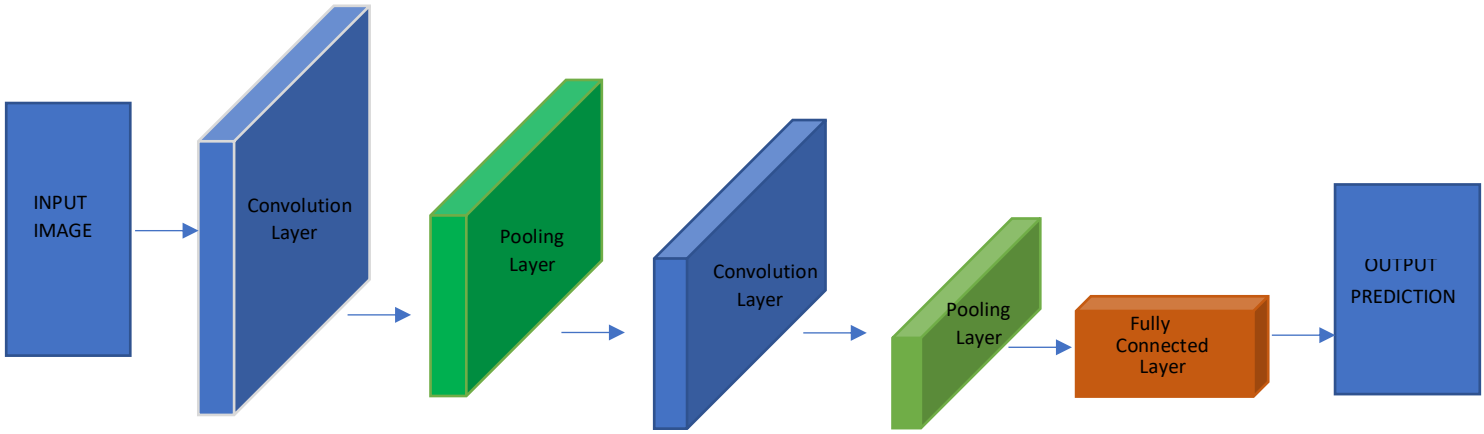

Figure 3.1: Showing basic components of CNN

3.2 TRANSFER LEARNING

This is a deep learning technique which uses an existing trained model on a specific domain problem to develop another model with the aim of solving a different type of domain problem.

Transfer learning is the use of a trained model on a particular task repurposed for a different but similar task (Brownlee, 2019). The use of transfer learning can present enormous benefit as it can shorten the amount of time it takes to train and develop new a model, also does not necessarily require large amount of data as the model is already pre-trained, especially when training data is limited. Transfer learning techniques are used by researchers to solve tasks because of the numerous benefits they provide, the most important of which is time savings when developing a model. This is due to the fact that rather than creating a new model from scratch, which could take time depending on the complexity of the domain problem. As a result, a pre-trained model developed or trained on a similar problem can be used. However, for transfer learning to be effective it must be applied to related or similar problems. That is, old task and new tasks should be similar. Otherwise, it can result in poor performance of the model which is usually referred to as negative transfer. Currently one of the challenges being faced with application of transfer learning technique is that there is no defined, specific standard or algorithms which determines how tasks are similar or related.

Some major pre trained deep learning models for computer vision are

- Inception v3
- VGG- 16
- VGG- 19
- ResNet-50
- Xception

We briefly discuss some of the popular pre trained architecture highlighting some its important features.

3.2.1 VGG-16 ARCHITECTURE

VGG-16 (Visual geometry group 16) is a CNN model used mainly for computer vision tasks such as object detection, image segmentation and image classification. This model was developed by University of Oxford. The vgg-16 architecture which is illustrated in Figure 3.2 is characterized by small 3x3 filters and 16 layers which includes 13 convolutional layers and 3 fully connected layers and 1 softmax layer. This architecture can be broken down into 5 blocks of layers with a different number of convolutional layers.

- The first block consists of 2 convolutional layers, each followed by a max-pooling layer.
- The second block also consists of 2 convolutional layers, each followed by max-pooling layers
- The third block consists of 3 convolutional layers, each followed by max-pooling layer.
- The fourth block consist of 3 convolutional layers, each followed by a max-pooling layer.
- The final block consists of 3 fully connected layers, with the final layer producing the output of the network model.

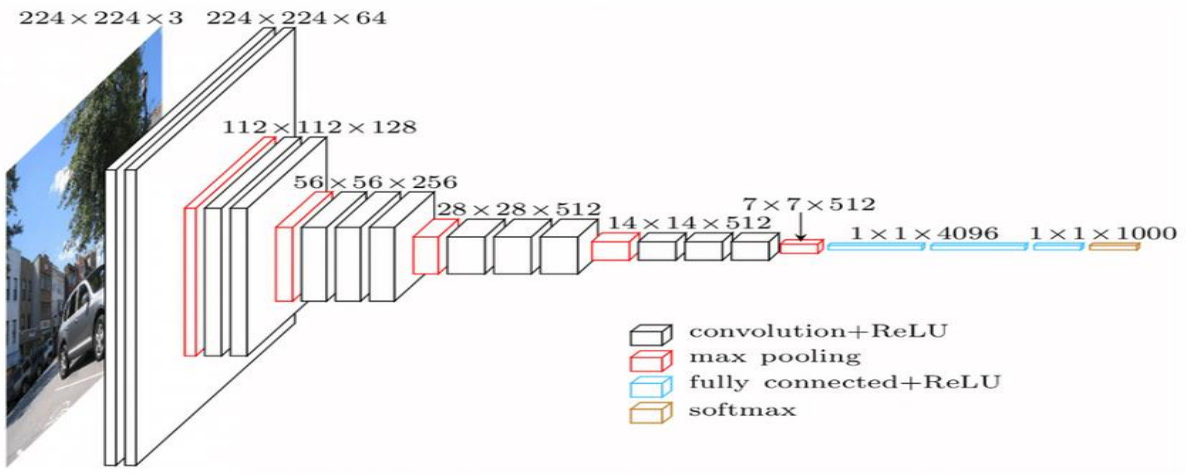

Figure 3.2: showing of basic VGG-16 architecture (Loukadakis et al., 2018)

3.2.2 VGG-19 ARCHITECTURE

VGG-19 is a convolutional neural network (CNN) model which is a variant of visual geometry group (VGG), this model which is illustrated in Figure 3.3 was developed at the University of Oxford 2014, its architecture is characterized by use of small (3x3) filters and 19 layers (16 convolution layers, 3 fully connected layers, 5 MaxPool layers and 1 Softmax layer). The architecture is divided into two parts.

- The first part is made up of multiple convolutional layers, max pooling layers with RELU activation function. The convolutional layers use filter size of 3x3 and a stride and padding of 1, which helps to preserve the spatial dimensions of the given input. The max pooling layers reduce the spatial dimensions of the feature maps as this helps to reduce the computational cost of the network.
- The second part is made up of fully connected layers which accepts outputs of the convolutional layers and perform classification tasks. The fully connected layers use a large number of neurons which helps to capture the high-level features of the input.

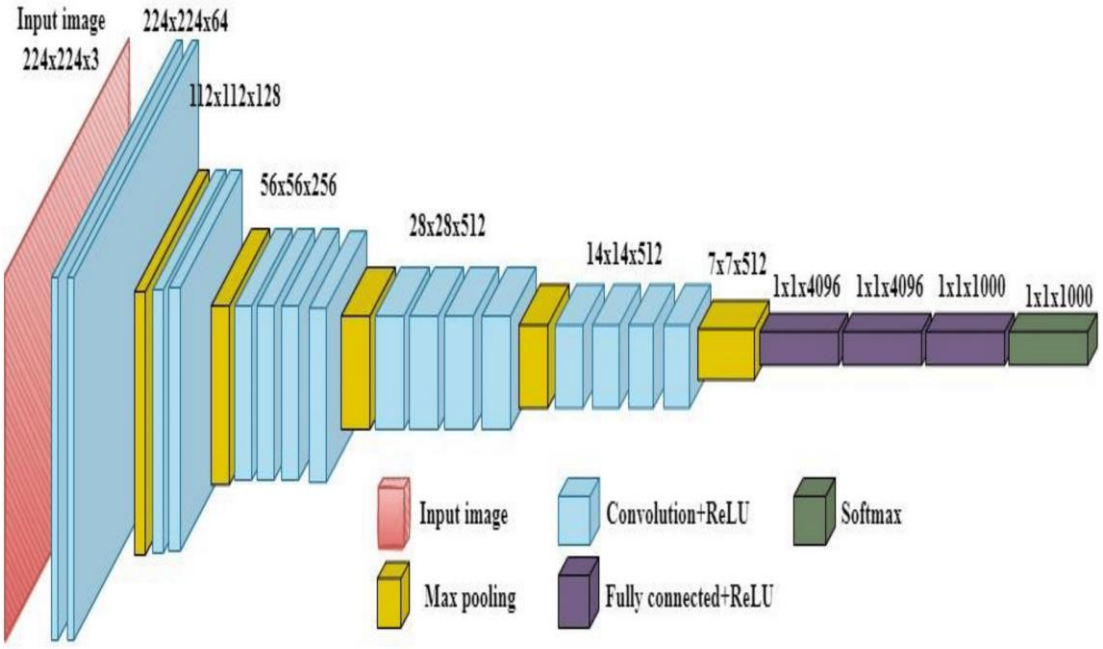

Figure 3.3: showing VGG-19 architecture contain different layers (Nguyen et al., 2022)

3.2.3 RESNET-50 ARCHITECTURE

This model is a type of CNN architecture which belongs to the ResNet (Residual Network) family of model. This CNN architecture is one of the most popular and widely used, which was introduced by Microsoft research in 2015. This architecture illustrated in Figure 3.4 addresses the challenges often associated with deep neural networks which is the problem of

vanishing gradients during training, where the gradients become incredibly small as they propagate backwards through many layers making training difficult. Hence, the architecture utilizes residual connections also referred to as skip connections or shortcut connections. The residual connections allow for the architecture to learn the residual function which simply is the difference the input and the expected output. Instead of attempting to directly learn the complete mapping, the network learns to approximate the residual outcomes. These connections make it possible for the network to effectively pass information from lower layers to higher layers which enable the model to learn more efficiently and effectively perform better. The building block of the ResNet-50 architecture is the residual block which consists of two or more convolutional layers with the addition of skip connections which can bypass these layers. The skip connection has the ability to directly propagate the inputs from one layer to the next layer, thereby allowing the gradient to flow easily during backpropagation. With these shortcuts or skip connections introduced, ResNet-50 is effectively able to train very deep networks with its 50 layers present.

The ResNet-50 is applied to a wide range of computer vision tasks such as object detection, image classification and semantic segmentation. The architecture is characterized by its residual connections which are designed to make it easier for the network to identify functions.

The ResNet -50 architecture consists of 50 layers which includes:

- The input layer: this layer accepts the input image size of 224 x 224 pixels
- The convolutional layers: these layers are responsible for extracting features from the input image. ResNet-50 has several convolutional layers with different filter sizes and strides.
- The pooling layers: these layers are used to reduce the spatial dimensions of the feature maps, which helps to reduce the computational complexity of the network
- The residual layers: these layers are the main feature of the ResNet architecture, these connections are designed to make it easier for the network to learn identity functions.
- The fully connected layer: this layer is used to make the final prediction. It is connected to all the neurons in the final feature map, so it can make use of all the features that were extracted by the convolutional and pooling layers
- The output layer: this layer produces the final prediction of the network.

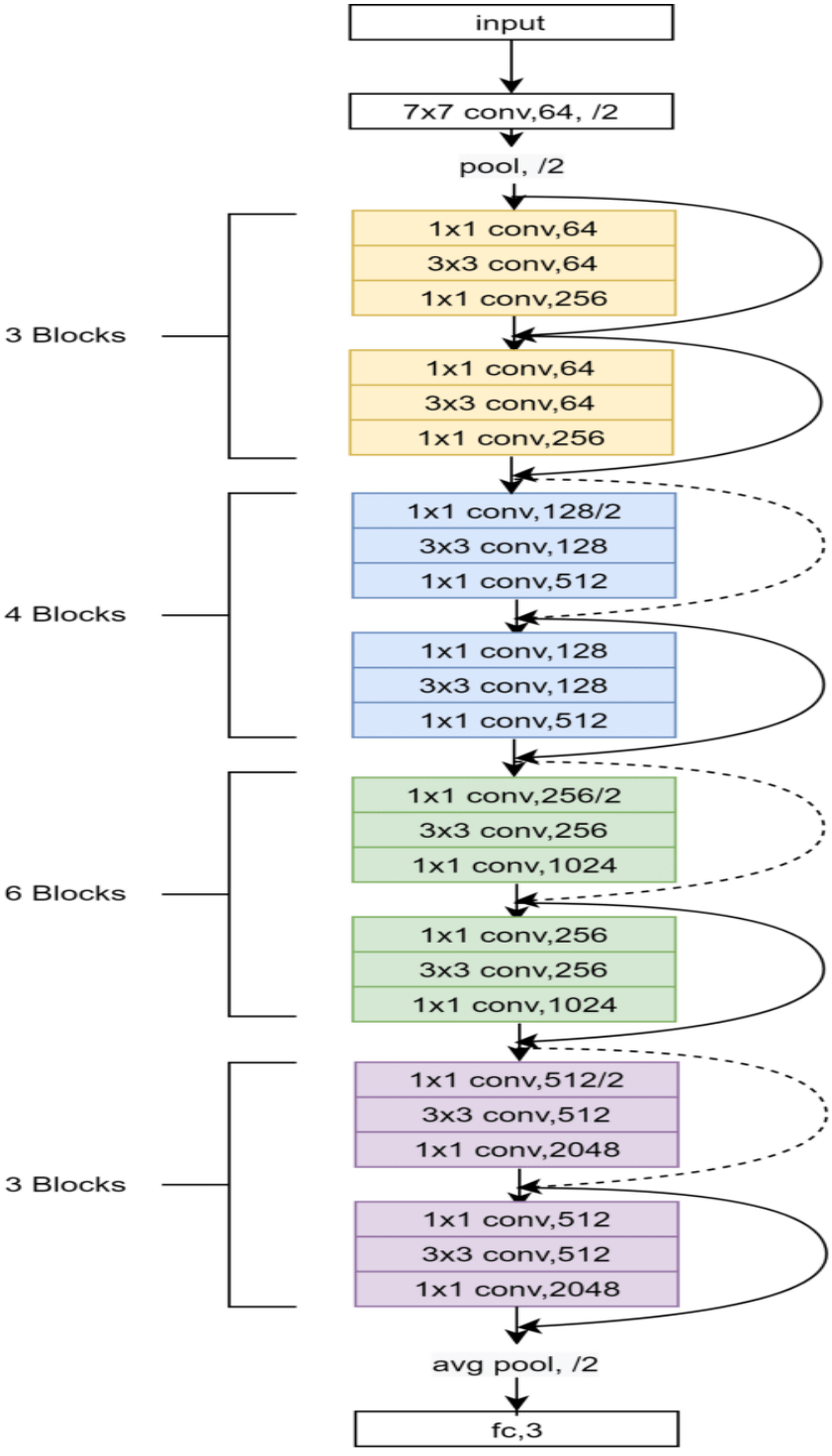

Figure 3.4: Showing the an illustration of the ResNet -50 architecture with skip connections (Al-Humaidan & Prince, 2021)

3.3 MALWARE VISUALIZATION

Due to the increasing rate of malwares attack and the increasing level of sophistication, this is as a result of the fact that most malware authors usually modify small sections of the existing malware codes manually or using automation tools to produce newer malwares. It has become crucial to find effective methods for understanding and combating these threats. One approach gaining traction is malware visualization, which leverages visual similarities properties. This is evident when malware samples are visualized which reveals similarities in structure, composition and other crucial feature information. Hence visualizing malwares can be used to quickly classify malwares into groups. One of the first researcher to use visualization was (Nataraj et al., 2011) who represented malware samples as grayscale images in order to distinguish similarities and differences among malwares samples. This showed visual similarities of malwares belonging to the same family.

Malware visualization is an effective means of representing malware samples in a visual form as this provides the unique opportunity to employ image processing techniques for the detection and classification of malwares. It shows the structure and composition of the malwares as well as other feature information. Visualizing malwares as images unlocks several advantages. Firstly, it allows security analysts to perceive patterns and similarities in the visual representation of malware, which may not be immediately apparent when examining raw code. These visual patterns can serve as valuable indicators for grouping and classifying malware samples efficiently. Moreover, visualizations enable analysts to identify modifications made by malware authors to existing code, whether manually or through automation tools. This insight aids in understanding the evolving nature of malware and devising effective defence strategies.

By representing malware samples as images, the opportunity to use image processing techniques in malwares detection and classification arises. Malware images typically have both local and global features or descriptors. Local features or descriptors are tiny patches or pixels within the image. These localized attributes enable us to discern fine-grained details, such as specific code segments or unique pixel patterns. By examining these localized features, we can identify commonalities among malware variants that share similar code sections or visual characteristics. While global feature or descriptor gives a general or holistic description of the whole malware image. They encompass contour, shape, and texture representations that encapsulate the overall structure and appearance of the malware. Analysing these global features allows us to capture

the overarching characteristics of malware, facilitating higher-level comparisons and classifications.

By combining local and global features, malware visualization empowers security researchers to gain a comprehensive understanding of malicious code. It enables the detection of similarities and patterns that may not be immediately apparent through traditional analysis methods alone. Moreover, visualizing malware facilitates the development of more effective and targeted defence mechanisms, as researchers can leverage these insights to devise advanced detection and prevention strategies.

Malware visualization is a powerful approach for unravelling the complexities of malware attacks. By transforming malware samples into visual representations and leveraging image processing techniques, we can identify shared structures, classify malware into distinct groups, and develop robust defence mechanisms to safeguard against evolving threats.

3.4 E-CNN METHODOLOGY

The concept of using visualizing malwares for classification is not a new technique as it has been done by various researchers. However, there is an imperative or need to improve and develop a better or effective model for detection and classification of malwares using newer methods as well as latest dataset of malware samples. This is achieved by developing a deep enhanced CNN model called E-CNN. Although, existing research has laid the groundwork for malware visualization, our approach aims to push the boundaries further by integrating cutting-edge techniques. The E-CNN model will capitalize on the inherent advantages of CNNs, such as their ability to automatically learn hierarchical features from raw input data. By leveraging this deep learning architecture, our model will gain a deeper understanding of the complex visual patterns within malware samples, thereby enhancing its classification accuracy.

The followings stages are outlined in developing a model namely: Dataset Preparation,

Visualized malware pre-processing, feature Pre-processing and classification (Hammad et al., 2022)

3.4.1 MALWARE IMAGES

In order to capture the inherent characteristics of malware binary files, the binaries are visualized as RGB images so as to extract the texture and coloured features enabling clear distinction between different malware binaries. By representing malwares as images, we

can effectively identify clear feature distinctions (both local and global descriptors) from malware binaries of special byte sequences which consist of Dynamic Link Libraries (DLLs), string constraints, uninitialized data, debug information which are present in the code section and data section as well as other sections. The visual representation of malware binaries offers valuable insights into their structure and composition. By converting the complex binary code into images, we gain a more intuitive understanding of the visual and structural characteristics shared by malwares of the same type or belonging to the same family. This visual similarity becomes evident when observing the resulting images, this is shown in the Figure 3.5 and 3.6.

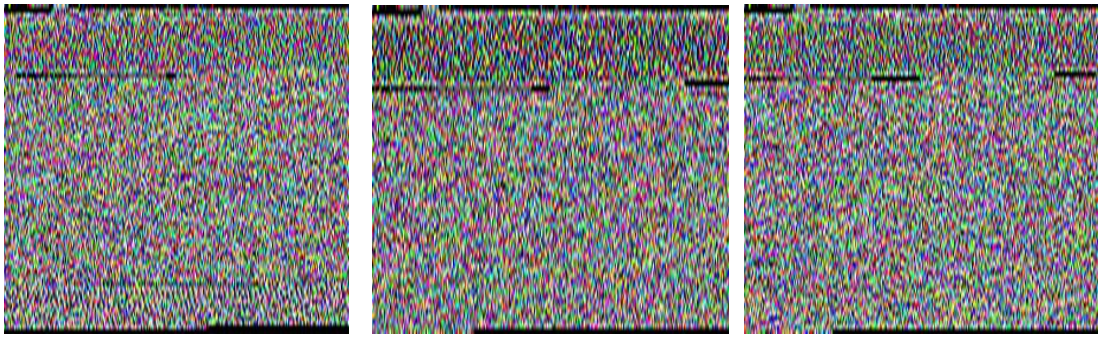

Fig 3.5 Images of malware samples belonging to Allaple Family

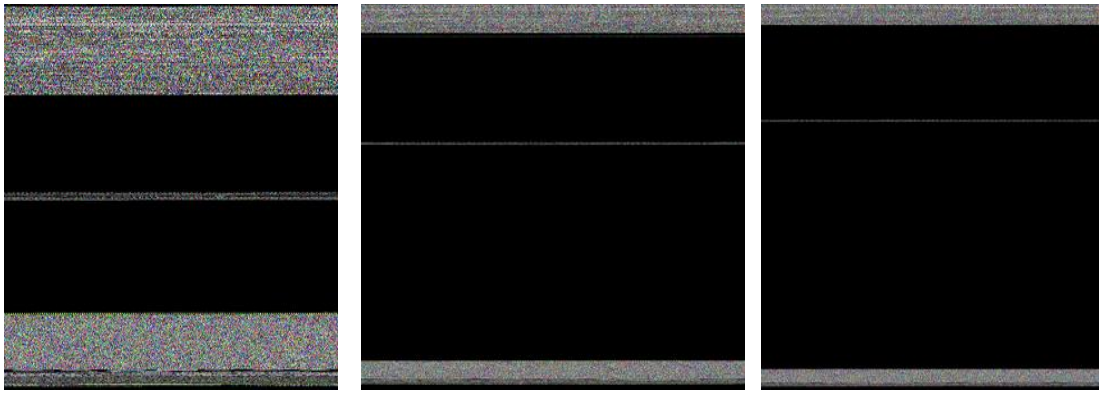

Fig 3.6 Images of malware samples belonging to Autorun Family

When analysing malware images, we can observe common patterns and visual cues that indicate shared traits among related malware samples. These similarities can manifest in the form of recurring pixel arrangements, distinct colour distributions, or recurring shapes and contours. Hence, this relationship can further investigate and quantify these shared characteristics, facilitating the classification and grouping of malwares into distinct families or types.

3.4.2 DATASETS

3.4.2.1 MALEVIS DATASET

The dataset used in the development of the E-CNN model is called the Malevis dataset which is an open set image dataset which consist of 26 classes of byte images (25 malware classes and 1 legitimate class) as shown in the Table 3.1. This dataset was constructed by extracting binary images from malware files in 3 channels RGB form by bin2png script developed by Sultanik. The generated images are then resized into two different squared size resolution (224x224 and 300x300 pixels). The Malevis dataset consist of 9100 training and 5126 validation RGB images.

Table 3.1: Description of the Malevis Dataset.

S\N	Family Name	Family	Total Samples
1	Adposhel	Adware	494
2	Agent	Backdoor	470
3	Allaple	Worm	478
4	Amonetize	Adware	497
5	Androm	Backdoor	490
6	BrowseFox	Adware	493
7	Dinwod	Trojan	499
8	Elex	Adware	500
9	Expiro	Virus	501
10	Fasong	Trojan	500
11	HackKMS	Hacktools	499
12	Hlux	Worm	500
13	Injector	Trojan	495
14	InstallCore	Adware	500
15	MultiPlug	Adware	499
16	Neoreklami	Adware	500
17	Neshta	Virus	497
18	Other (legitimate)	Legitimate	1832
19	Regrun	Trojan	485
20	Sality	Virus	499
21	Snarasite	Trojan	500
22	Stantinko	Trojan	500
23	VBA	Virus	500
24	VBKrypt	Trojan	496
25	Vilsel	Trojan	496
26	Autorun	Worm	496

3.4.2.2 EXPERIMENTAL BENCHMARK DATASET

In other to evaluate the newly developed E-CNN model performance, a public and popular benchmark malware dataset called Malimg is used. The dataset provides a diverse collection of malware images from different families, making it suitable for evaluating the robustness and generalization ability of classification models. Many researchers have used the Malimg dataset as a benchmark to compare the performance of their models with existing state-of-the-art methods. This dataset contains 9435 grayscale image samples from 25 malware families as shown in the Table 3.2. This will further test the suitability of the E-CNN model.

Table 3.2: Data Description of Malimg Dataset

S/N	Family Name	Family	Total Samples
1	Allaple.A	Worm	2949
2	Allaple.L	Worm	1591
3	Adialer.C	Dialer	122
4	Agent.FYI	Backdoor	116
5	Alueron.gen!J	Trojan Horse	198
6	Autorun.K	Worm AutoIT	106
7	C2LOP.gen!g	Trojan Horse	146
8	C2LOP.P	Trojan Horse	200
9	Diaplatform.B	Dialer	177
10	Dontovo.A	Trojan downloader	162
11	Fakerean	Rogue	381
12	Instantaccess	Dialer	431
13	Lolyda.AA1	Password Stealer	213
14	Lolyda.AA2	Password Stealer	184
15	Lolyda.AA3	Password Stealer	123
16	Lolyda.AT	Password Stealer	123
17	Malex.gen!J	Trojan Horse	136
18	Obfuscator.AD	Trojan downloader	142
19	Rbot!gen	Backdoor	158
20	Skintrim.N	Trojan	80
21	Swizzor.gen!E	Trojan downloader	128
22	Swizzor.gen!I	Trojan downloader	132
23	VB.AT	Worm	408
24	Wintrim.BX	Trojan downloader	97
25	Yuner.A	Worm	800

3.4.3 MODEL OVERVIEW

This research work proposes a convolutional neural network CNN for classification of malwares, this developed model is referred to as deep enhanced CNN (E-CNN). This model architecture is developed with the aim of achieving high accuracy for classification of malwares into different classes and to ensure that the model is generic data independent and learns the discriminative feature representation from the image data itself. The parameters which were considered using an optimization library called Keras Tuner library which iteratively fine tunes the CNN layers considering different hyperparameters until best performance values are discovered, such hyperparameters includes the activation function which is Rectified Linear Unit (ReLU) for different layers, this is a non-linear function which is very effective identifying complex relationships within data when analysed. Also, SOFTMAX for the last layer of the model as well as using categorical cross entropy loss function for multi-classification of output classes or labels. Unlike other optimizers, Adam optimizer was selected. This is because of its superior performance and ability to establish adaptive learning rates for each parameters (El-Shafai et al., 2021). Furthermore, ADAM optimizer which tries to minimize the loss during use of training data and with categorical cross entropy to train the model for classification of malwares into different malware classes as well as legitimate class. The Figure 3.7 shows the model diagram which consist of the different layers and Figure 3.8 shows the schematic architecture of the E-CNN model is shown below.

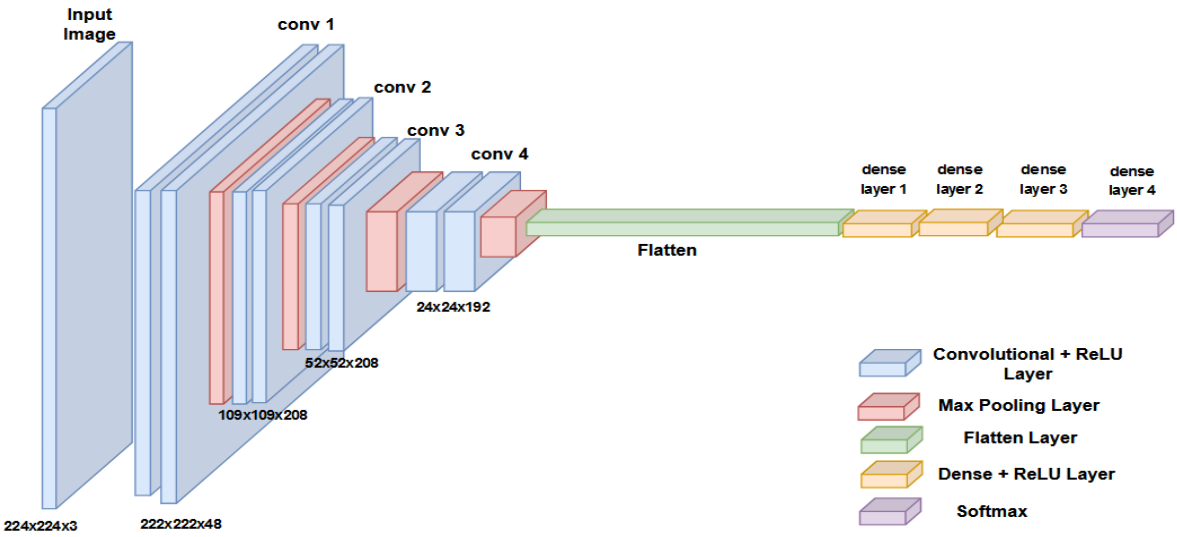

Figure 3.7: Overview of the E-CNN model architecture

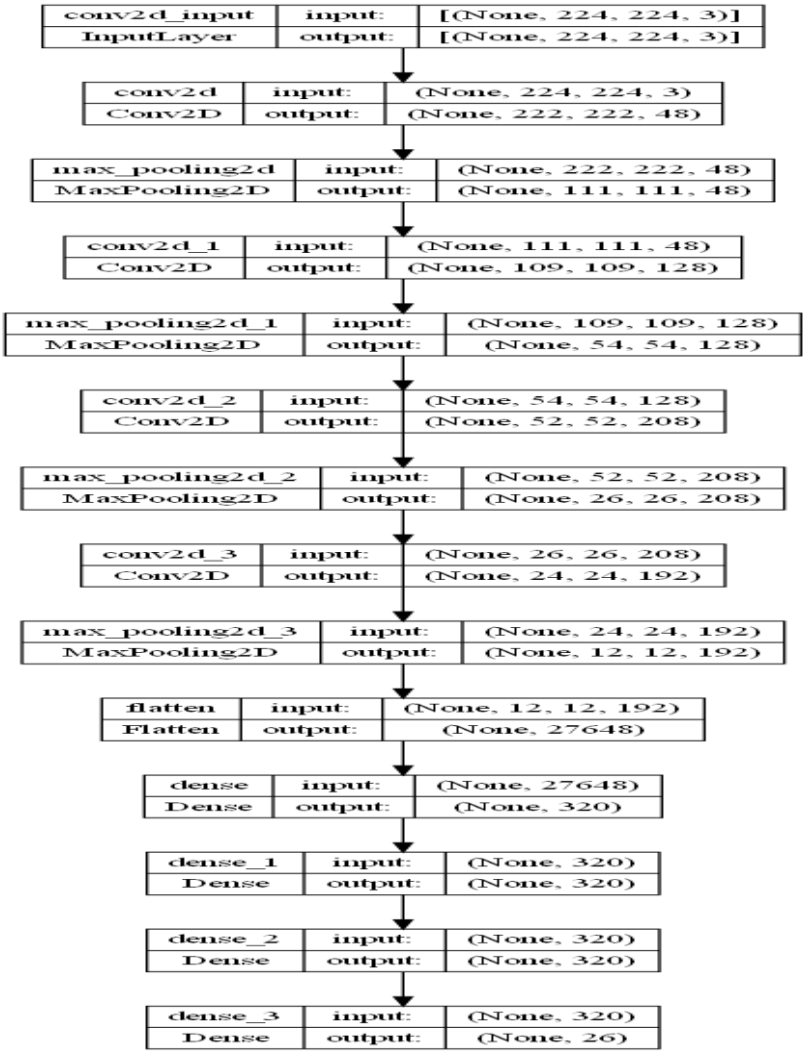

Figure 3.8: Schematic Diagram of the E-CNN architecture

3.5 PERFORMANCE EVALUATION METRICS

Performance metrics is important in evaluating the performance of E_CNN model as this assess how effective the model is in classification of malwares. The performance evaluation metrics includes

1. Accuracy Score
2. Precision score
3. F1- score
4. Recall
5. Confusion Matrix

3.5.1 ACCURACY SCORE

This refers to the ratio of correctly predicted outcomes to the total number of possible outcomes. The accuracy score metrics calculates the percentage of correct predictions made by the classifier. As a result, the overall correctness of the classifier is determined. The accuracy formular is stated equation 3.1.

$$\text{Accuracy} = \frac{True\ Positive + True\ Negative}{Total\ Outcomes} \qquad \ldots\ldots\ldots\ldots\ldots\ldots (3.1)$$

3.5.2 PRECISION SCORE

This is the ratio of correctly predicted positive outcomes to the total predicted positive outcomes. The precision score represents or measures the percentage of correct positive predictions as shown in equation 3.2.

$$\text{Precision score} = \frac{True\ Positve}{True\ positive + False\ positive} \qquad \ldots\ldots\ldots\ldots.. (3.2)$$

3.5.3 RECALL

This refers to the ratio of correctly predicted outcomes to the overall outcomes in the positive class. Recall is the proportion of real positive cases that are correctly predicted positive. This represents the number of true positives divided by the total number of true positives and false negatives as stated in equation 3.3

$$\text{Recall} = \frac{True\ Positve}{True\ positive + False\ negative} \qquad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (3.3)$$

3.5.4 F1 SCORE

F1 score also referred to as F measure, it is the weighted average of the recall and precision. it is the harmonic mean of precision and recall. It represents the harmonic mean of the precision and recall scores, which is used to calculate the F1 score. An F1 score of 1 is assigned to a model that has perfect precision and recall scores. This formular is stated in equation 3.4

$$\text{F1} = 2\ X\ \frac{Precision\ X\ Recall}{Precision + Recall} \qquad \ldots\ldots\ldots\ldots\ldots\ldots (3.4)$$

3.6 STEPS/ PROCEDURE OF PROPOSED RESEARCH MODEL

1. The dataset consists of both legitimate and malware samples represented as images with a total of twenty-six classes with twenty-five (25) malware classes and a legitimate class). It is with the Malevis dataset the proposed E-CNN malware classification model is developed. The image samples give a visual representation of the feature and texture information of the samples. This allows image samples to be classified according to the family they belong as they are often similar visually.
2. The image samples have a dimension of 224x224 pixel as input for the model, then the E-CNN model is then defined using optimization functions which determines the best hyperparameters values such as number of layers (convolutional, pooling, dense layers,), kernel size and filters. So as to determine the best model for classification of malwares.
3. The E-CNN model having being developed is further tested with a test dataset which consist of images samples of different malware families and legitimate files. The model develop in step 2 are evaluated using the performance metrics which includes accuracy score, precision score, recall, and f1 score.
4. The transfer learning technique is applied on the Malevis dataset on the existing pre-trained models which includes (VGG19, RESNET50, VGG16). This process is done until optimal performance values is determined.
5. The pre-trained model's evaluation metrics results are compared with E-CNN model so as to access the performance of newly developed model (E-CNN) for malware classification.
6. The E-CNN model and the pre-trained models are test with public benchmark dataset (MALIMG),

3.6.1 ALGORITHM

Algorithm: To Develop E-CNN model

Input: Using Malevis dataset to develop, malware classification model

Output: Develop the E-CNN model, using optimized parameters values.

1. Malevis dataset is loaded, which contains visual representation of malware samples and used to get the feature vectors
2. The image samples of the dataset were pre-processed, and the resultant size of the image samples was set at 224 x 224, which serves as input to the model.

3. Using optimization function and techniques to determine the best hyperparameter values in developing the model for malware classification
4. The E-CNN model is developed
5. Using pre-trained model (RESNET50, VGG16, VGG19) on Malevis dataset and access its performance with the newly developed E-CNN model
6. E-CNN model is tested with benchmark dataset (MALIMG).

CHAPTER FOUR

4.0 EXPERIMENTS AND RESULTS

This chapter describes and explains the practical environment in which E-CNN model was developed as well as using transfer learning on the pre-trained models (VGG19, VGG16, RESNET50) on the Malevis dataset and comparative analysis of evaluation metrics of malware classification. The newly developed model (E-CNN) was applied on a public benchmark dataset (MALIMG) of the experiments so as to assess its performance.

4.1 HARDWARE SPECIFICATIONS/PROGRAMMING LANGUAGE

The experiment will be carried out using python language version 3.6,

Some of Python library modules used include tensorflow.keras.image, tensorflow.keras.models keras_tuner, sklearn.metrics, tensorflow.keras.layers, tensorflow.keras.callbacks, matplotlib, pandas, seaborn, numpy.

- Tensorflow library: is an open-source library developed by google for both machine learning and deep learning applications. It offers an end-to-end platform which makes handles numerical computation for development of models.
- Keras library: Keras is a Python-based deep learning API that runs on top of the TensorFlow machine learning platform.
- Anaconda navigation studio: this is an open source software which consists of a collection of packages which is used for data visualization and development both machine learning and deep learning tasks (Anaconda, 2023).
- Matplotlib: this is a comprehensive library for creating data visualization in python (Matplotlib, 2023)
- Pandas: it is a powerful and flexible open source tool used for both analysis and processing of numerical data in python programming language (Pandas, 2023)
- Keras_tuner is an hyperparameter optimization framework which uses search algorithms to find best hyperparameter values.
- Numpy: it is a powerful open-source tool used for n-dimensional arrays vectorization, numerical computation
- Scikit-learn: this is a simple and effective tool used for predictive data analysis for machine and deep learning applications.
- Anaconda Navigator studio

Hardware specification

Dell XPS 8930 Core i7 8th generation,

16GB RAM and 256GB SSD, 32GB GPU

Windows 10 professional 20H2

This experiment is divided into two phases

1. Model development using Malevis dataset and transfer learning
2. Evaluation of model using benchmark dataset Malimg

4.2 MODEL DEVELOPMENT

E-CNN model was developed using Malevis dataset, this involves continuous test and optimizing the values of different parameters which makes up the model. These parameters include convolutional layers, kernel size, optimizers, dense layers, epochs, batch sizes. The E-CNN model consist of 4,252,474 parameters. Also, the parameters of VGG19, VGG16 and ResNet 50 is show in the table 4.1.

Table 4.1: Description showing the model's trainable and non-trainable parameters

	Model Name	Trainable parameters	Non trainable parameters	Parameters
1	**E_CNN(Proposed model)**	**4,252,474**	**0**	**4,252,474**
2	VGG16	138,266	1,735,488	1,873,497
3	VGG19	203,802	10,585,152	10,788,954
4	ResNet 50	53,274	23,587,712	23,638,937

MALEVIS DATASET EXPERIMENT

4.2.1 E-CNN PARAMETERS

The Malevis dataset was used in developing E-CNN model for multi classification of malwares without the use of any data augementation technique, this dataset was divided into two folds or parts, with training set is 80% and testing 20%. The use of validation data is necessary so as to help optimizer and fine tune the model hyperparameters. The following are some of the model

hyperparameter values: epochs=23, batch size = 50, adam optimizer as well as learning rate = 0.0001.

The performance analysis metrics of the model used to evaluate the model were confusion matrix, classification report and well as the learning curves (accuracy and loss) of the model. The model achieved and average accuracy of 83.87 % in classification of the 25 different classes of malwares. The Figure 4.1 shows the accuracy graph of the training and validation data curve.

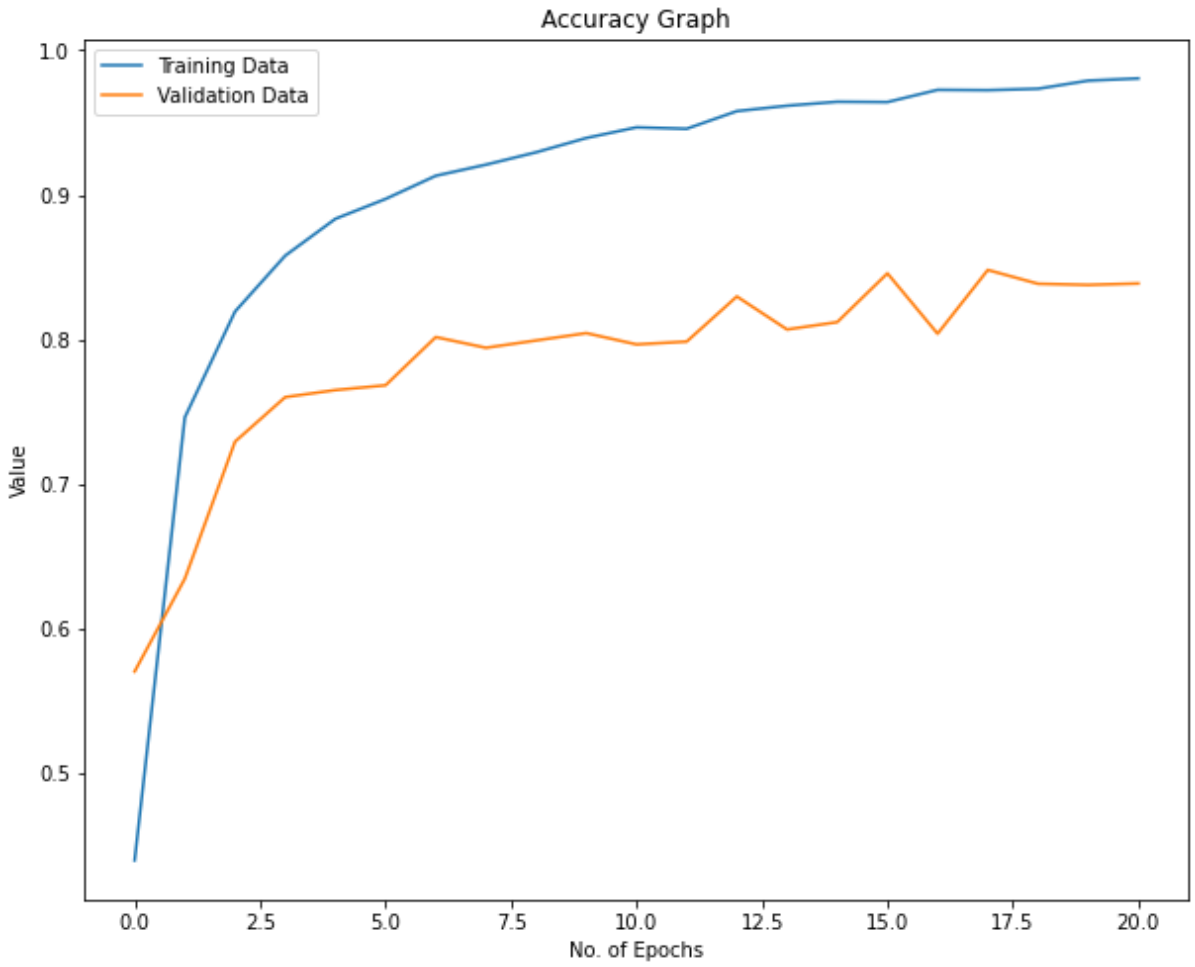

Figure 4.1 Accuracy graph shows Training vs Validation Data curve.

Also, the figure 4.2 shows the loss graph curve of the model for training vs validation data.

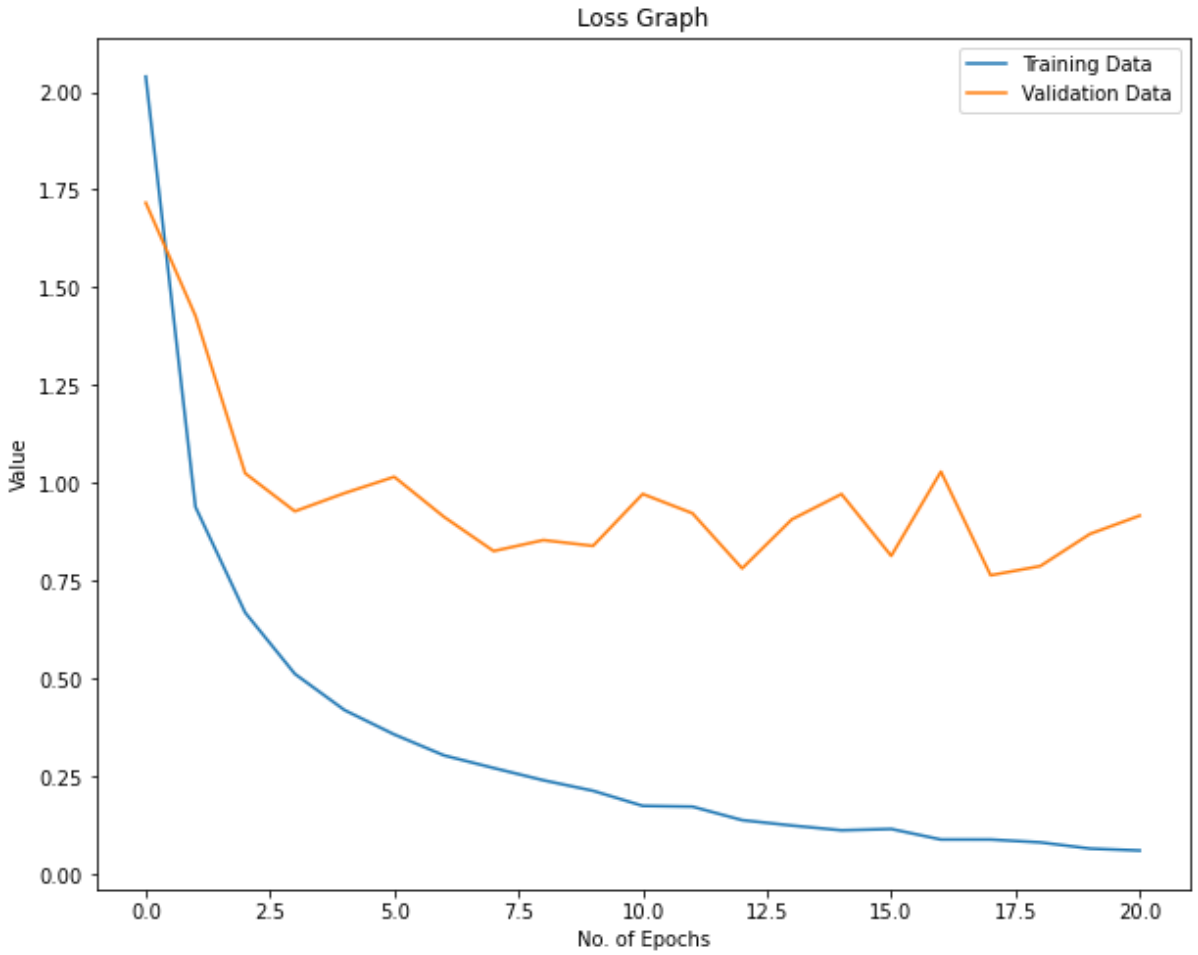

Figure 4.2: Loss Graph showing the training vs validation data curve over a period of 20 epochs

The Table 4.2 below shows the classification report of E-CNN model, accesses the model performance in classifying different malware classes.

Table 4.2: Classification Report of E-CNN on Malevis Dataset

Index	Family Name	Precision	Recall	F1-score	Support
0	Adposhel	0.99	1.00	0.99	144
1	Agent	0.72	0.83	0.77	120
2	Allaple	0.85	0.94	0.89	128
3	Amonetize	0.97	0.97	0.97	147
4	Androm	0.59	0.97	0.74	150
5	Autorun	0.80	0.89	0.84	146

6	BrowseFox	0.89	0.92	0.91	143
7	Dinwod	1.00	0.97	0.98	149
8	Elex	0.79	0.99	0.88	150
9	Expiro	0.79	0.86	0.82	151
10	Fasong	1.00	1.00	1.00	150
11	HackKMS	0.98	0.99	0.98	149
12	Hlux	0.99	1.00	1.00	150
13	Injector	0.72	0.89	0.80	145
14	InstallCore	0.99	0.98	0.98	150
15	MultiPlug	0.92	0.90	0.91	149
16	Neoreklami	0.85	0.99	0.91	150
17	Neshta	0.29	0.62	0.39	147
18	Legitimate	0.96	0.57	0.71	1482
19	Regrun	1.00	0.99	1.00	135
20	Sality	0.42	0.72	0.53	149
21	Snarasite	1.00	1.00	1.00	150
22	Stantinko	0.97	0.97	0.97	150
23	VBA	1.00	1.00	1.00	150
24	VBKrypt	0.60	0.95	0.74	146
25	Vilsel	0.99	1.00	0.99	146
Accuracy				0.83	5126
Macro avg		0.85	0.92	0.87	5126
Weighted avg		0.88	0.83	0.83	5126

The confusion matrix result of the experiment is represented in Figure 4.3 shows E-CNN model performance in predicting different classes of malwares.

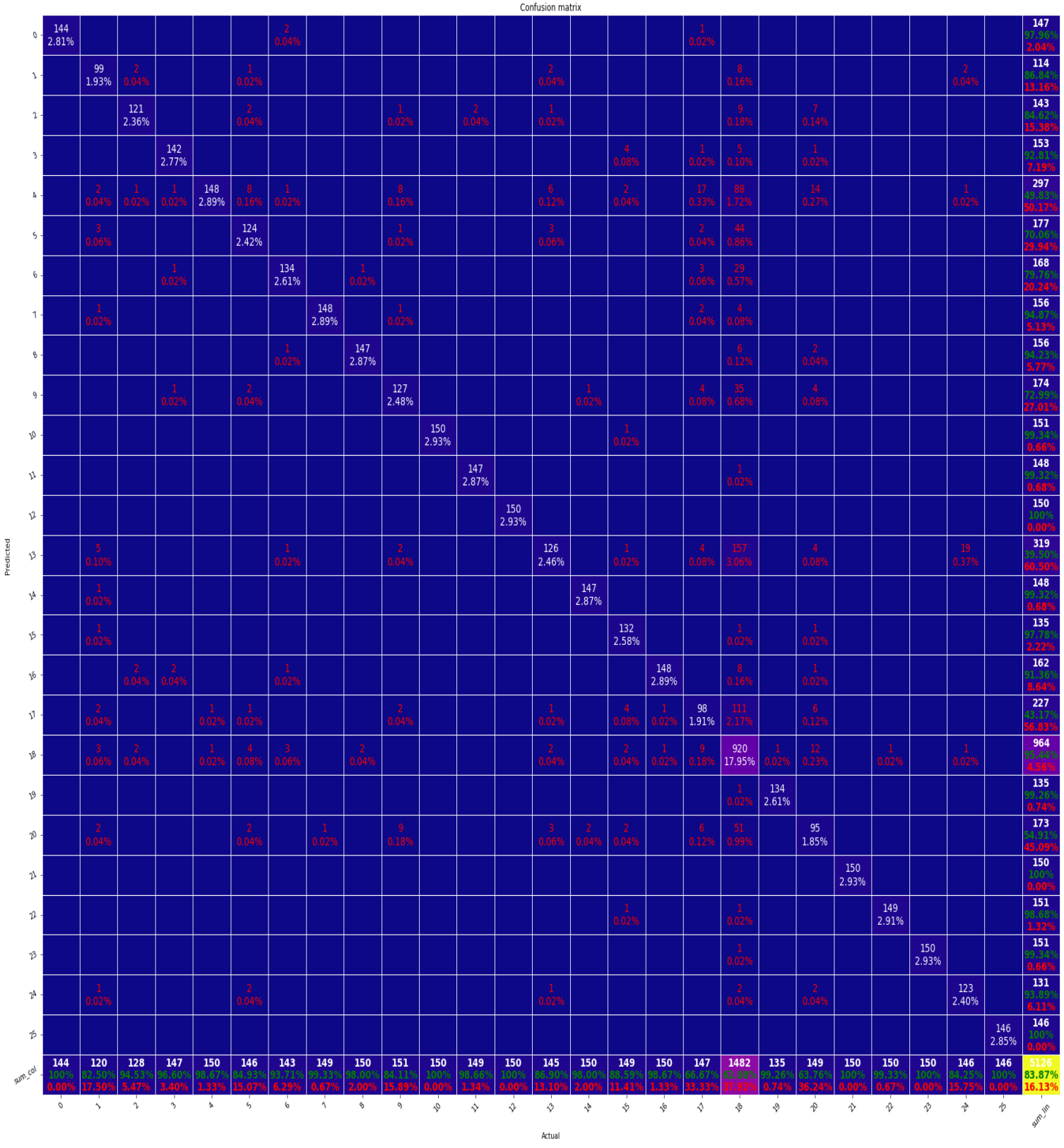

Figure 4.3: Confusion matrix table result of E-CNN model

4.2.2 MALEVIS DATASET TRANSFER LEARNING (VGG16, VGG19, RESNET-50)

Transfer learning was used to train the following CNN architectures: Vgg-16, Vgg19, and Resnet-50. The number of epochs and training time for malware multi classification varies due to differences in architecture size. Applying the Malevis dataset on the resnet-50 architecture yielded an average accuracy score of 49.89% and for loss an average of 2.7 for malware multi classification. The accuracy and loss graph shown in figure 4.4 and figure 4.5 shows the performance of the Resnet-50 architecture when used on training and validation data over 13 epochs (training time).

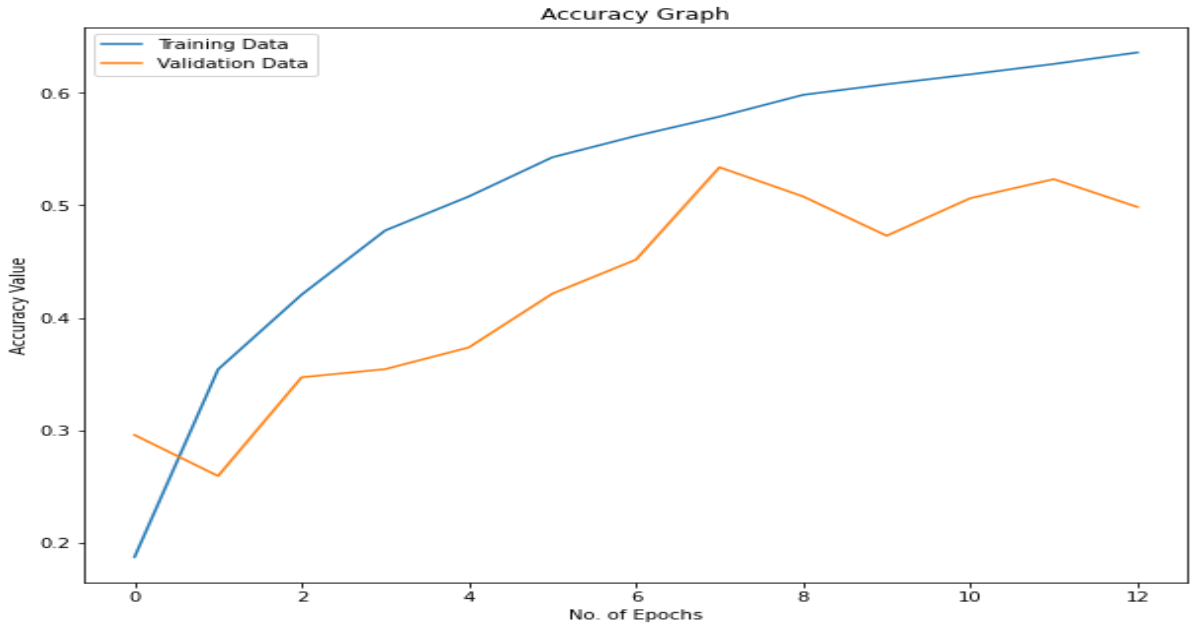

Figure 4.4: Accuracy graph of resnet-50

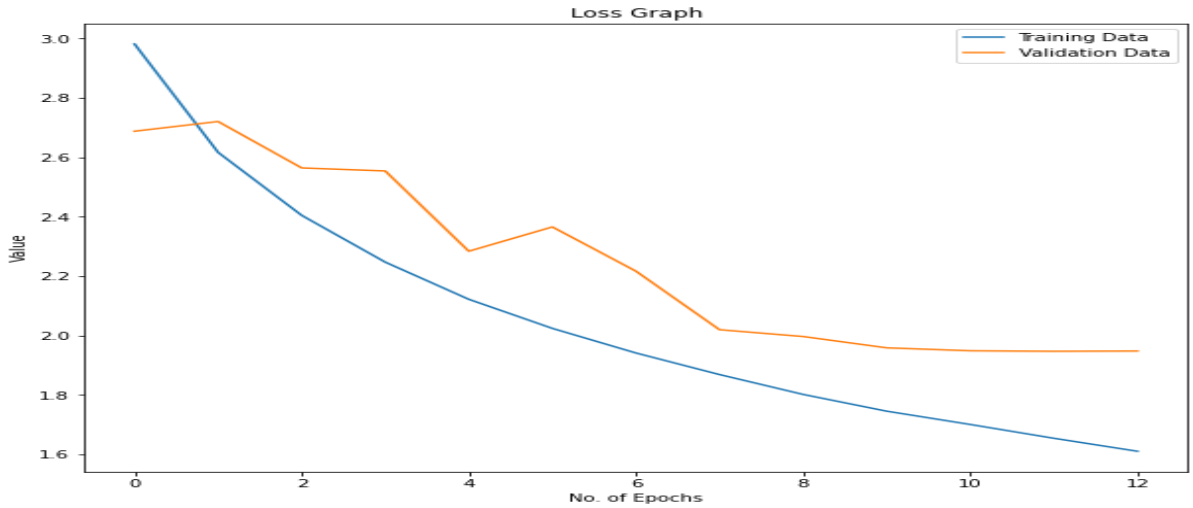

Figure 4.5: Loss graph of resnet-50

The confusion matrix result is shown in Figure 4.6 for Resnet-50 performance

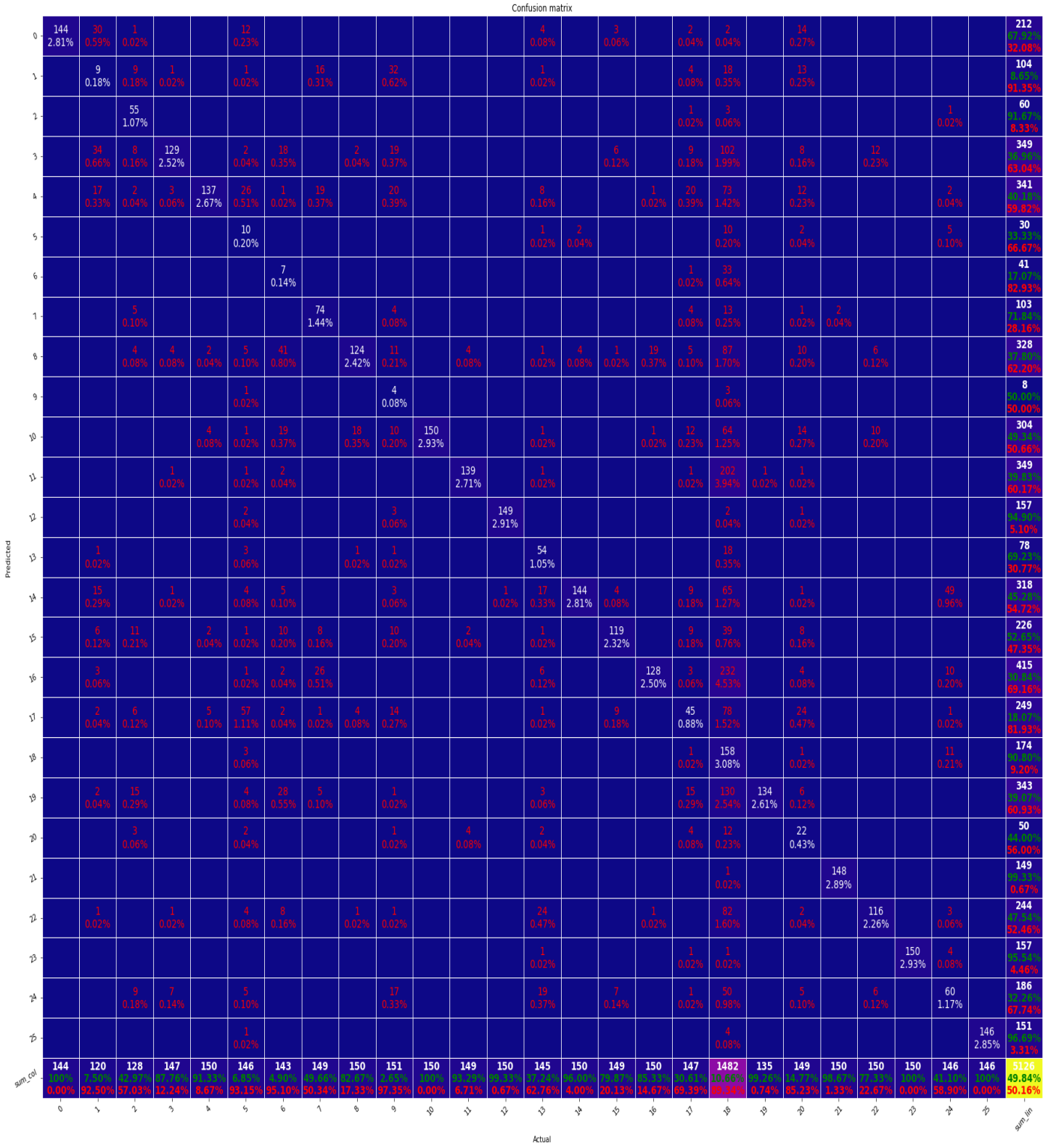

Figure 4.6: showing confusion matrix of malware classification of Resnet-50 architecture

Furthermore, the next CNN architecture VGG16 was trained over 24 epochs. VGG16 architecture achieved an average accuracy score of 82.52% for malware multi classification. The performance of the VGG16 model shows the accuracy and the loss graph in the figure 4.7 and figure 4.8

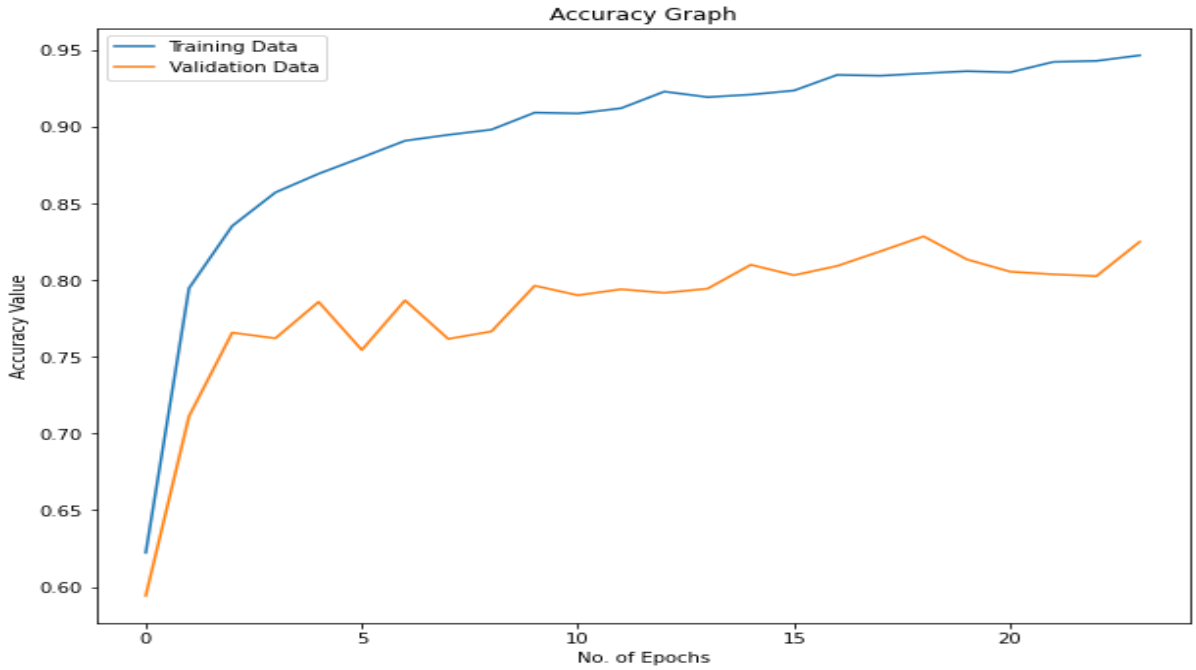

Figure 4.7: Accuracy graph of vgg-16 architecture

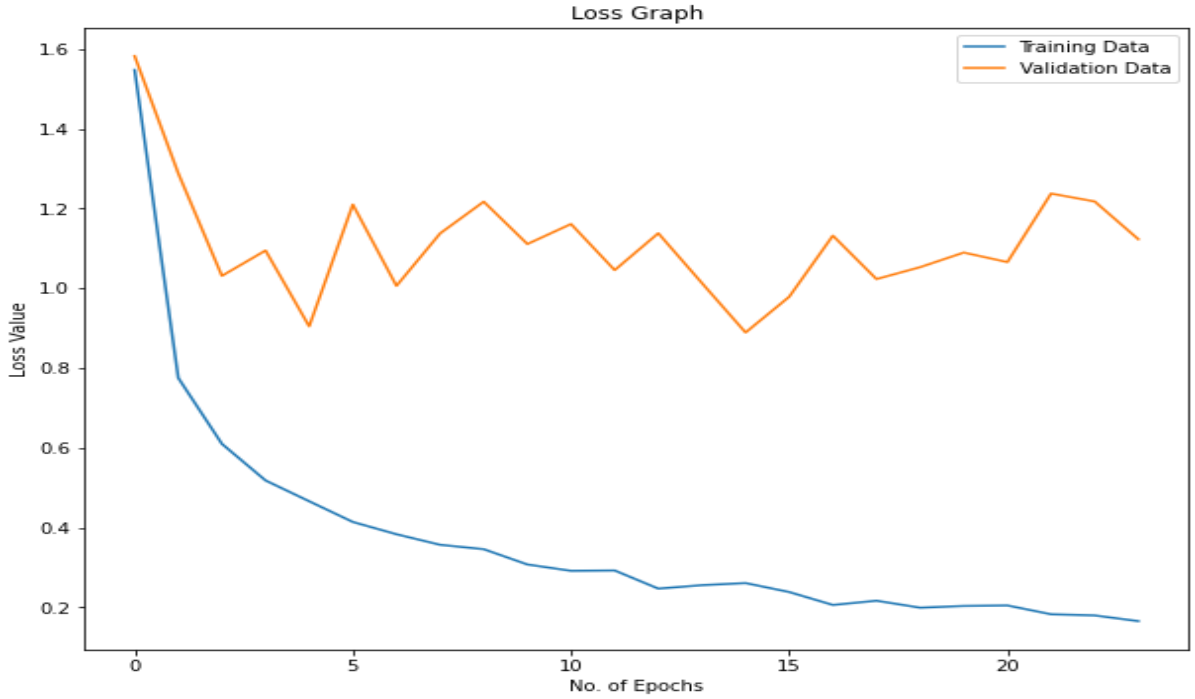

Figure 4.8: Loss graph of vgg-16 architecture

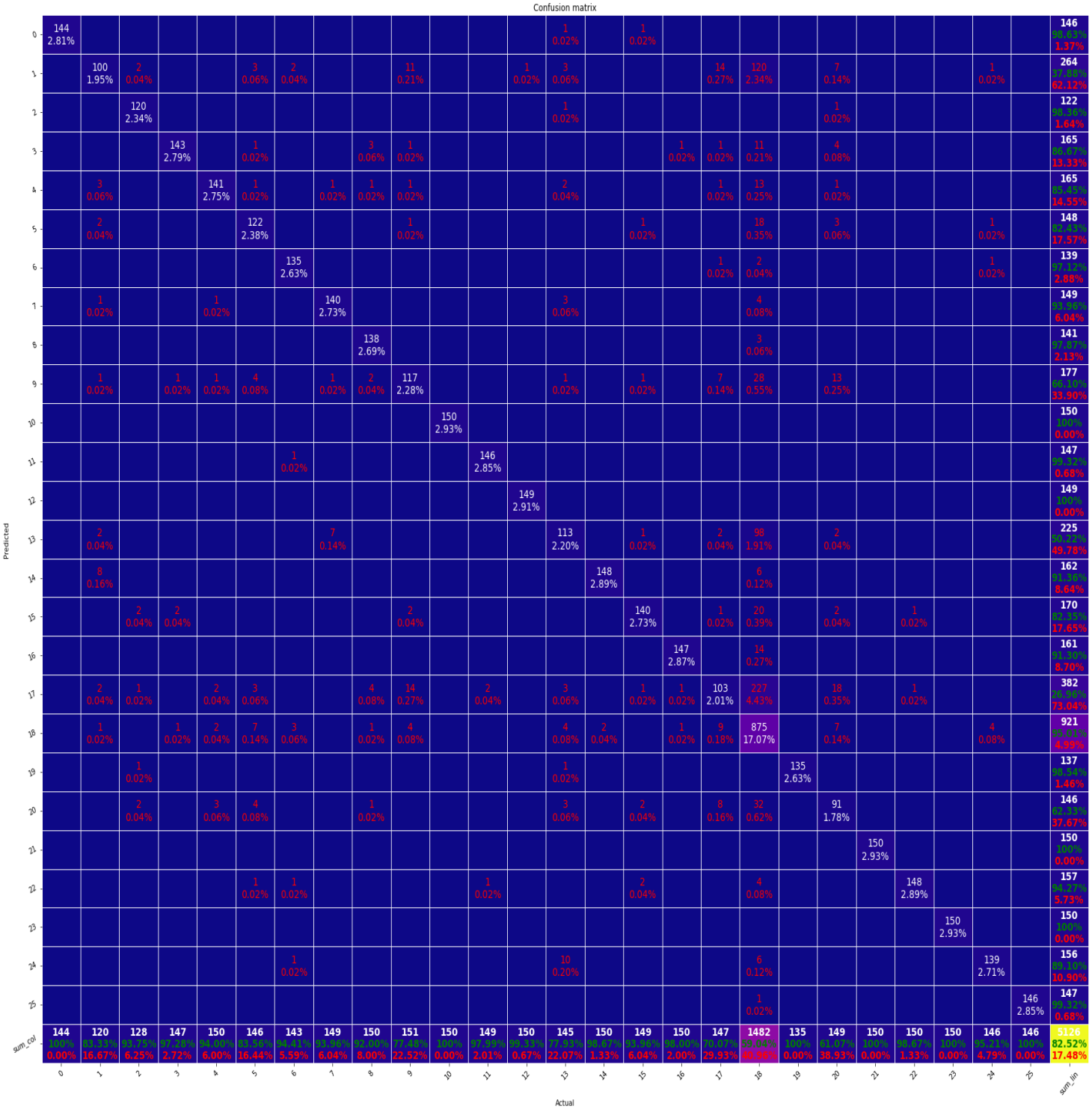

Figure 4.9: showing confusion matrix of malware classification of VGG16 architecture

CNN architecture VGG19 was trained over 23 epochs. VGG19 architecture achieved an average accuracy score of 84.98% for malware multi classification. The performance of the VGG16 model shows the accuracy and the loss graph in the Figure 4.10 and Figure 4.11

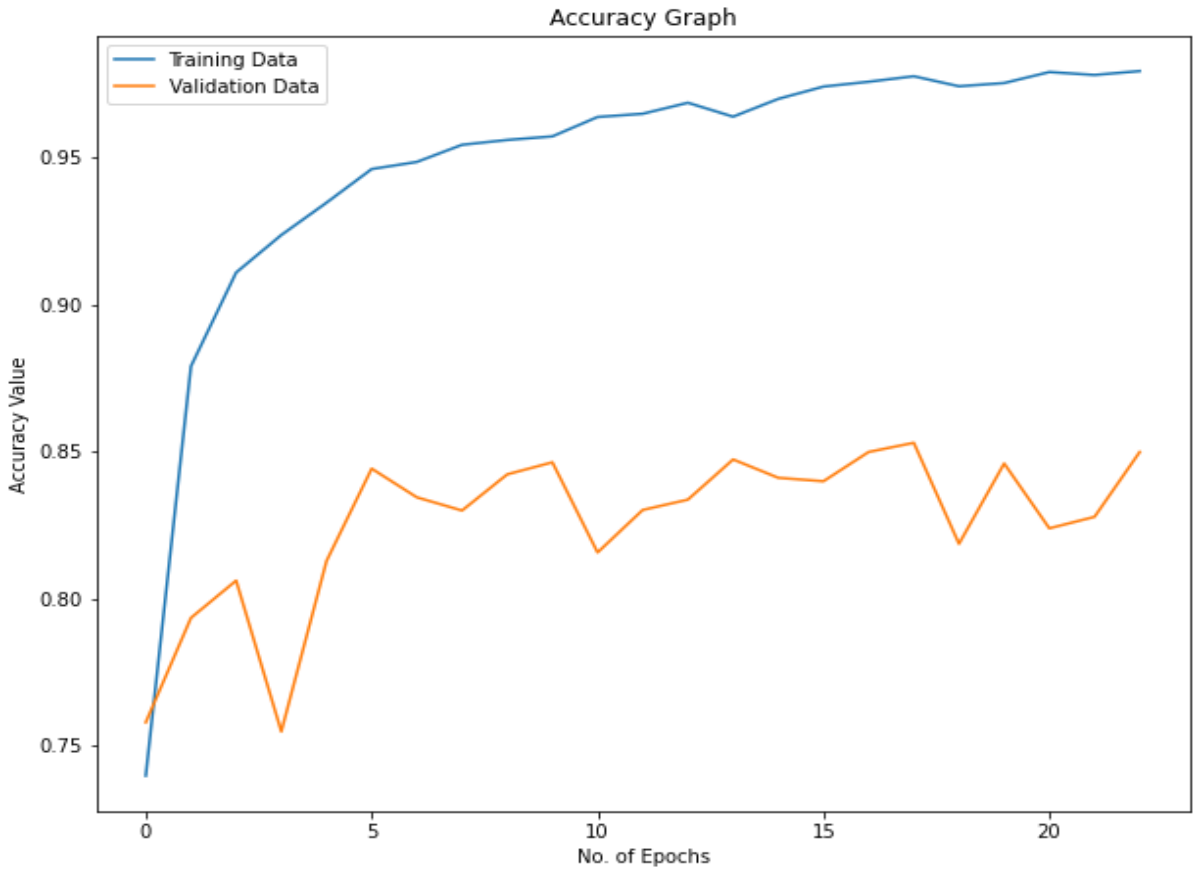

Figure 4.10: Accuracy Graph of malware classification using VGG-19 architecture

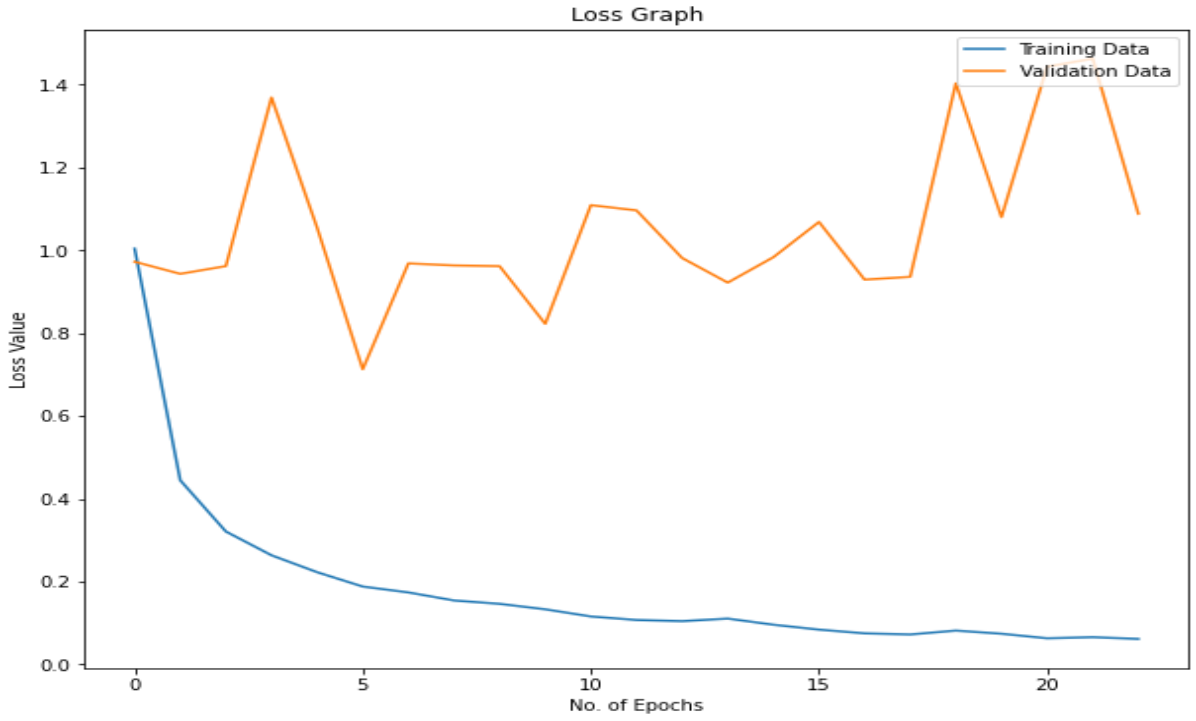

Figure 4.11: Loss Graph for malware classification using VGG-19 architecture

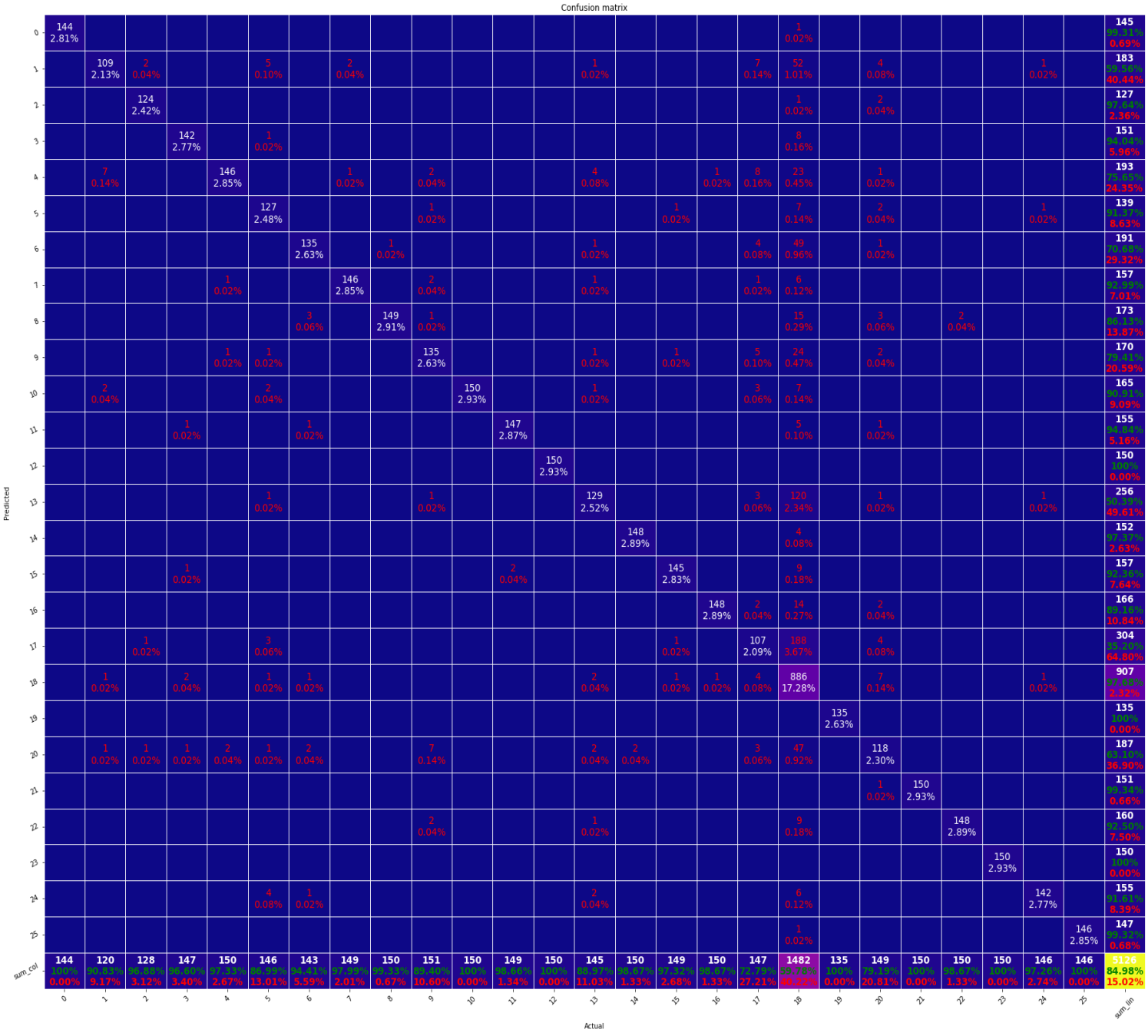

Figure 4.12: showing confusion matrix of malware classification of VGG19 architecture

4.3 RESULT ANAYLSIS

4.3.1 Malevis Dataset Experiment

In the experiment, the E-CNN architecture model for malware classification was developed using optimal hyperparameter values, using the Malevis dataset which is presented in Figure 4.3, fine tuning process was applied to get the best values on both CNN layers and hyperparameters without applying any data augmentation processes with the programming language and hardware

specification explained in section 4.1. The accuracy and loss curves, the confusion matrix and classification report were used as the evaluation metrics. The E-CNN model produced an average accuracy score of 83.87% on the Malevis dataset. Transfer learning techniques were also used to train the following CNN architectures: RESNET-50, VGG-16, and VGG-19, with average accuracy scores of 49.89%, 82.52%, and 84.98%, respectively. Comparing the results shows E-CNN and VGG-19 as the best performing models.

Furthermore, the analysis of the results reveals an interesting observation, in considering the two best performing models E-CNN and VGG19. Despite the fact that VGG19 model having an average accuracy score of 84.98% which is higher than the E-CNN model (83.87%), the E-CNN model performs better in generalization of malware classification based on malware families or types, of the 26 malware families or classes considered the E-CNN model outperformed by 14 malware classes compared to 11 malware classes by the VGG-19 model, while the remaining malware class was the same in both models. Figure 4.13 depicts this illustration

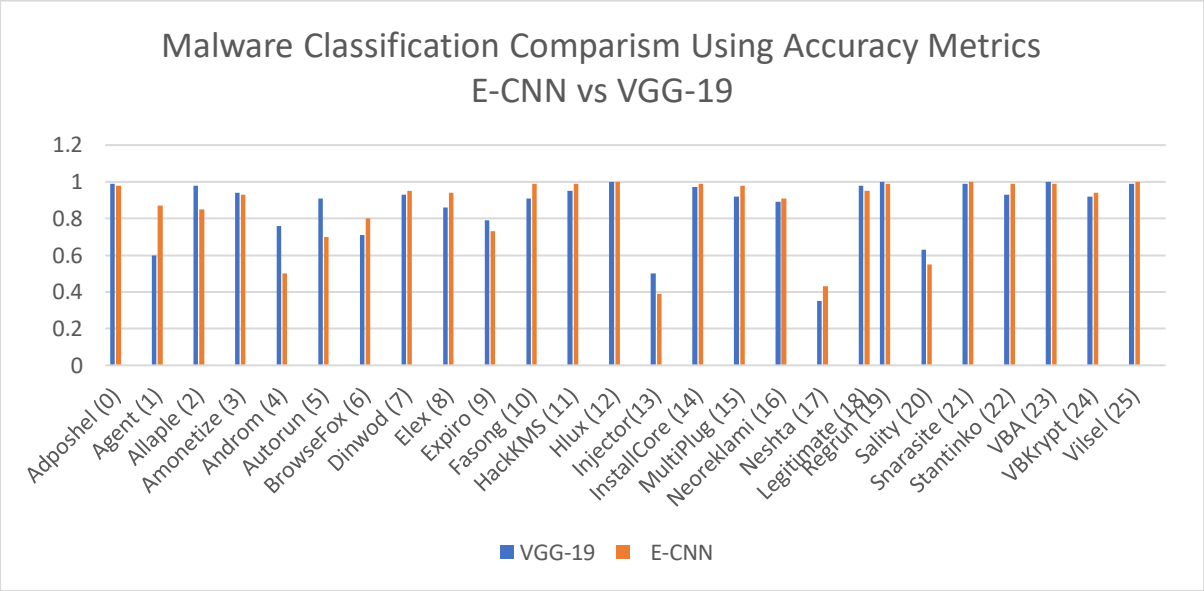

Figure 4.13: Figure showing the comparison between VGG-19 and E-CNN for malware classification

4.3.2 BENCHMARK DATASET EXPERIMENT AND EVALUATION

MALIMG DATASET

The developed E-CNN model was tested further using Malimg, a popular and publicly available malware dataset created in 2013. This is because the dataset is widely used by researchers to assess the performance of both machine and deep learning models for malware detection and classification. The dataset consists of malware samples which are divided into 25 families. As a

result, the Malimg dataset, is used as a benchmark performance measurement for the E-CNN model. Furthermore, the developed E-CNN architecture model is used to compare malware detection and classification models developed by other researchers.

The result of the E-CNN model experiment is evaluated using accuracy metrics, this indicates whether or not malware samples are correctly labelled. The model achieved an accuracy score of 98.88%. Thus, this reveals that this model is highly effective for malware classification of visual samples without using any data augmentation or data balancing method to enhance classification performance. This performance analysis is presented detail in the accuracy and loss graph curve, with the model trained in 16 epochs as shown in Figure 4.14 and Figure 4.15. Also, E-CNN model is evaluated in terms of confusion matrix this is shown in Figure 4.16.

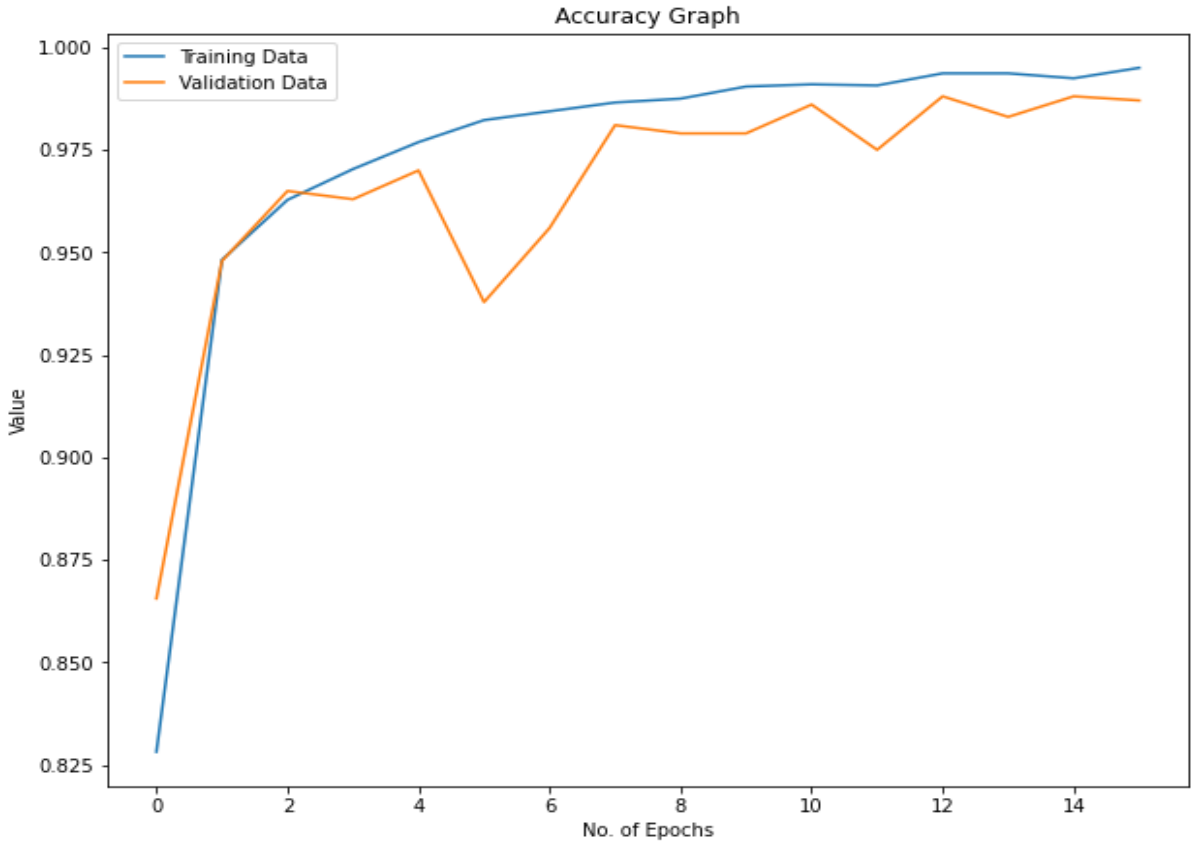

Figure 4.14: Accuracy graph showing the E-CNN model performance on Malimg dataset

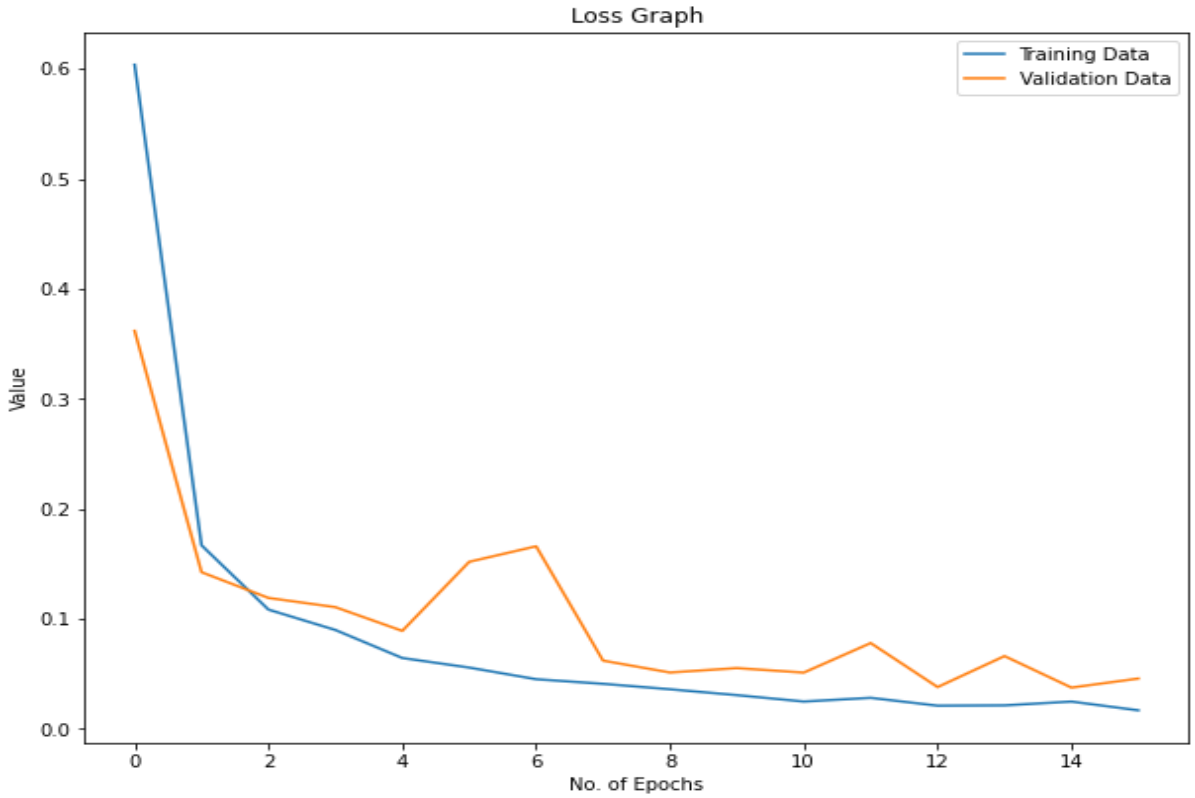

Figure 4.15: Loss graph showing E-CNN model performance on Malimg dataset

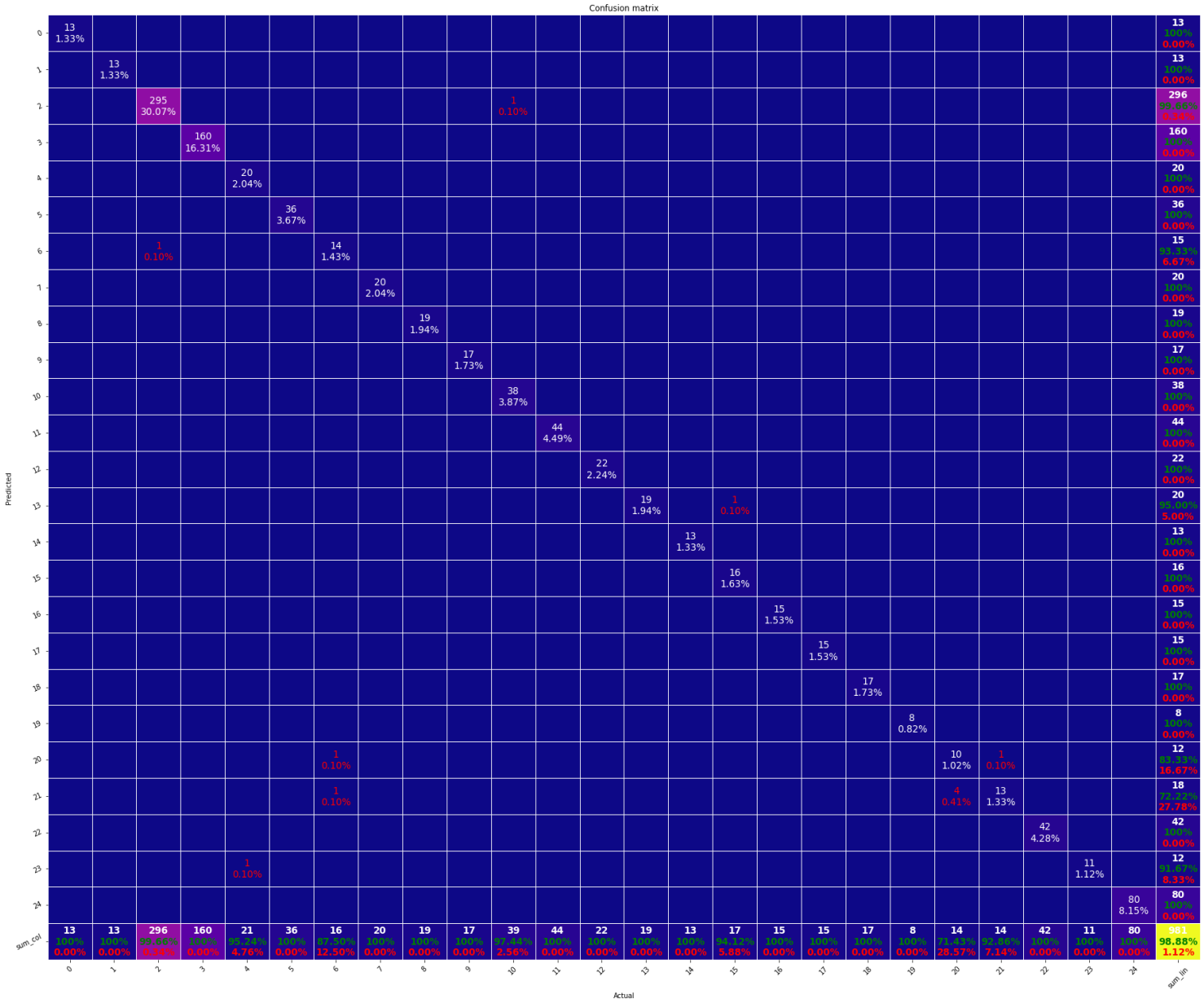

Figure 4.16: showing confusion matrix of malware classification of E-CNN architecture on Malimg dataset.

BENCHMARK COMPARISM WITH OTHER CITED WORKS

In this section, we assess the E-CNN model's performance using the malimg dataset. We further analyze the experiment's outcomes by comparing the model's results with those of other studies that also utilized the malimg dataset, measuring accuracy metrics. The results are presented in the Table 4.3

Table 4.3: Table comparing the proposed model with other models

s\n	Model Name	Authors	Accuracy (%)
1	EEMDS: Efficient and Effective Malware Detection System with Hybrid Model based on XceptionCNN and LightGBM Algorithm	(Onoja et al., 2022)	97.40 %
2	Attention-Based Cross-Modal CNN Using Non-Disassembled Files for Malware Classification	(Kim et al., 2023)	98.72%
3	Robust Malware Family Classification Using Effective Features and Classifiers	(Hammad et al., 2022)	96.64%
4	Malware classification through image processing with a convolutional neural network	(Marin et al., 2022)	98.70%
5	**This study**		**98.88%**

CHAPTER FIVE

5.0 SUMMARY

In this research paper, one of the objective of the research work was to develop a malware detection and classification model using CNN architecture, this was achieved using the malevis dataset which comprises of malware image samples of 25 distinct malware classes. The E-CNN model was able to detect and classify different malware samples into different classes having been able to identify unique structure of malware samples represented as images. Furthermore, the performance of the E-CNN model was shown to be very effect when the newly developed model was further tested by comparing it with estblished pre-trained CNN architectures, namely VGG16, VGG19, ResNet50 using the same malevis dataset. Subseqently, the proposed E-CNN model was applied on the popular Malimg dataset. Where the model achieved high accuracy in multi-classification of different malware image samples. Furthermore, the E-CNN model's result was also compared with results of other cited research work with the Malimg dataset being the benchmark dataset. The result showed E-CNN had a better performance when compared with some previous cited research work. Our findings indicate that the E-CNN model outperformed some of the previously cited research works, showcasing its effectiveness in the field of malware detection and classification.

5.1 CHALLENGES AND ISSUES

From the review of the recent studies, as well as the experiment conducted several noteworthy insights have emerged, shedding light on the challenges inherent in the development of malware classification models. These identified challenges subsequently give rise to a multitude of issues within the realm of image-based malware classification. The crux of achieving successful classification lies in the dual qualities of consistency and effectiveness in the classifier's performance. Constructing such a classifier necessitates a comprehensive consideration of all the intricacies and obstacles that are entailed, which are as follows

Datasets Used

The datasets commonly employed by most researchers for malware classification within the reviewed literature, includes well-known datasets such as "Malimg" and the "Microsoft Malware Dataset 2015" although popular among researchers they could exhibit limitations in their effectiveness when utilized for developing models that can classify newer malwares. This shortcoming arises from the fact that constructing a model based on outdated malware samples

renders it ineffectual against contemporary malware threats. Consequently, the persistently evolving tactics of malware authors, who predominantly utilize modern malware samples, result in the classifiers' inability to accurately categorize these new malicious entities.

Furthermore, these datasets suffer from a limitation of diverse malware samples. This scarcity poses a significant challenge, particularly for CNN models which thrive on substantial data volumes to achieve robust training and model development. Insufficient samples within a dataset can induce overfitting in the models, wherein they become overly specialized to the limited data and consequently fail to effectively identify novel malware instances.

Performance Computation Measures

The computational costs associated with most of these models frequently exhibit a high degree of resource consumption, sometimes with unclear correlations to their performance in malware classification. This ambiguity spans multiple facets, encompassing the definition of performance metrics, the time required for training and testing, and the intricacies of translating malware binaries into colour images. Furthermore, approaches aimed at mitigating data imbalance issues during the classification of malware families, as well as efforts to condense the feature vector's dimensions, assume noteworthy importance. This is particularly significant since the size of the feature vector significantly impacts the overall efficiency attainable by these models.

Data enhancement/Data augmentation

The use of data enhancement and augmentation on datasets although can improve model's performance, however researchers need to exercise measure of caution or balance when employing these methods so as to prevent overfitting hence models memorize only training data and cannot adapt to actual data problems

5.2 CONTRIBUTION TO KNOWLEDGE

This research work highlights a few noteworthy mentions which are as follows

- The E-CNN model demonstrated a high level of cross-dataset generalization, as evident in our experiment. We developed the E-CNN on a recent dataset and assessed its performance using a well-established, albeit outdated, benchmark dataset. Notably, it achieved a remarkable level of accuracy. This adaptability is crucial in the ever-changing field of malware detection, as the model possesses the capability to detect previously unseen malware variants.

- During this research, most researchers applied generalized models through transfer learning to develop malware classification models. However, these models are susceptible to domain mismatches. In contrast, the E-CNN model can serve as a robust foundation for the development of more effective malware classification models

5.3 CONCLUSION AND FUTURE WORKS

The threat of malwares on information systems, computers is continuously evolving and changing hence the need for malware researchers to continually develop new methods or techniques in addressing this problem. In this research, a E-CNN model is proposed which can classify different malware types effectively and efficiently.

This research work aims to contribute to the advancement of developing models for detection and classification of malwares, using CNN. In the domain of malware visualization using CNN architecture, CNN aids analysts in identifying crucial image patterns. Based on the findings, it is clear that pursuing malware visualization in conjunction with the CNN approach can yield a more intelligent framework. This framework promises to enhance accuracy, efficiency, and overall performance, all of which are vital in the ever-evolving landscape of malware threats.

For future studies, several recommendations are worth noting:

1. **Dataset Choice**: Utilize a large, up-to-date malware dataset containing recent malware samples. This is crucial for assessing and validating performance measures effectively.
2. **Image Conversion**: Explore more efficient techniques for converting malware binaries into colour images, considering variations in image sizes across different datasets.
3. **Model Development**: While transfer learning has its merits, consider focusing on the development of novel models. This approach can help mitigate errors stemming from domain mismatch, where a model trained in one domain is applied to a different one.
4. **Feature Vector Dimensionality**: Reduce the dimensionality of the feature vector to enhance model efficiency.
5. **Data Imbalance**: Implement newer methods or techniques to address data imbalance problems effectively.

In the course of this research, significant knowledge gaps have been revealed, major challenges have been identified, also highlighted are open issues that will serve as valuable guides for future research endeavours.

REFERENCES

Abiodun, O. I., Jantan, A., Omolara, A. E., Dada, K. V., Mohamed, N. A., & Arshad, H. J. H. (2018). State-of-the-art in artificial neural network applications: A survey. *4*(11), e00938.

Agrawal, R., & Khan, L. (2021). AN EXPERIENCE IN ENHANCING MACHINE LEARNING CLASSIFIER AGAINST LOW-ENTROPY PACKED MALWARES.

Ahmed, M., Afreen, N., Ahmed, M., Sameer, M., & Ahamed, J. (2023). An inception V3 approach for malware classification using machine learning and transfer learning. *International Journal of Intelligent Networks*, *4*, 11-18.

Al-Humaidan, N. A., & Prince, M. (2021). A classification of Arab ethnicity based on face image using deep learning approach. *IEEE Access*, *9*, 50755-50766.

Albawi, S., Mohammed, T. A., & Al-Zawi, S. (2017). Understanding of a convolutional neural network. 2017 International Conference on Engineering and Technology (ICET),

Alshamrani, S. S. (2022). Digital Forensics for Malware Classification: An Approach for Binary Code to Pixel Vector Transition. *Computational Intelligence and Neuroscience*.

Alzubaidi, L., Zhang, J., Humaidi, A. J., Al-Dujaili, A., Duan, Y., Al-Shamma, O., Santamaría, J., Fadhel, M. A., Al-Amidie, M., & Farhan, L. (2021). Review of deep learning: Concepts, CNN architectures, challenges, applications, future directions. *Journal of big Data*, *8*(1), 1-74.

Anaconda. (2023). *Anaconda* Anaconda.com. Retrieved 12- Jan- 2023 from www.anaconda.com

Asam, M., Khan, S. H., Jamal, T., Zahoora, U., & Khan, A. (2021). Malware Classification Using Deep Boosted Learning. *arXiv preprint arXiv:2107.04008*.

Awan, M. J., Masood, O. A., Mohammed, M. A., Yasin, A., Zain, A. M., Damaševičius, R., & Abdulkareem, K. H. (2021). Image-Based Malware Classification Using VGG19 Network and Spatial Convolutional Attention. *Electronics*, *10*(19), 2444.

Bae, S. I., Lee, G. B., & Im, E. G. (2020). Ransomware detection using machine learning algorithms. *Concurrency and Computation: Practice and Experience*, *32*(18), e5422.

Bensaoud, A., Abudawaood, N., & Kalita, J. (2020). Classifying malware images with convolutional neural network models. *International Journal of Network Security*, *22*(6), 1022-1031.

Brownlee, J. (2019). *Deep learning for Computer Vision*.

Cakir, B., & Dogdu, E. (2018). Malware classification using deep learning methods. Proceedings of the ACMSE 2018 Conference,

Carletti, V., Greco, A., Saggese, A., & Vento, M. (2021). Robustness evaluation of convolutional neural networks for malware classification. *ITASEC 2021*

Italian Conference on Cybersecurity 2021, *2940*, 414-423.

Chauhan, R., Ghanshala, K. K., & Joshi, R. (2018). Convolutional neural network (CNN) for image detection and recognition. 2018 First International Conference on Secure Cyber Computing and Communication (ICSCCC),

Deng, X., & Mirkovic, J. (2022). Polymorphic Malware Behavior Through Network Trace Analysis. 2022 14th International Conference on COMmunication Systems & NETworkS (COMSNETS),

Dwivedi P, & Sharan H. (2021). Analysis and Detection of Evolutionary Malware: A. *International Journal of Computer Applications*, *975*, 8887.

El-Shafai, W., Almomani, I., & AlKhayer, A. (2021). Visualized malware multi-classification framework using fine-tuned CNN-based transfer learning models. *Applied Sciences*, *11*(14), 6446.

Ghosh, A. (2021). An overview article on 600% increase in Cyber Attack in 2021.

Gibert, D., Mateu, C., & Planes, J. (2020). The rise of machine learning for detection and classification of malware: Research developments, trends and challenges. *Journal of Network and Computer Applications*, *153*, 102526.

Goodfellow, I., Bengio, Y., Courville, A., & Bengio, Y. (2016). *Deep learning* (Vol. 1). MIT press Cambridge.

Gu, J., Wang, Z., Kuen, J., Ma, L., Shahroudy, A., Shuai, B., Liu, T., Wang, X., Wang, G., & Cai, J. (2018). Recent advances in convolutional neural networks. *Pattern recognition*, *77*, 354-377.

Guo, T., Dong, J., Li, H., & Gao, Y. (2017). Simple convolutional neural network on image classification. 2017 IEEE 2nd International Conference on Big Data Analysis (ICBDA),

Hammad, B. T., Jamil, N., Ahmed, I. T., Zain, Z. M., & Basheer, S. (2022). Robust Malware Family Classification Using Effective Features and Classifiers. *Applied Sciences*, *12*(15), 7877.

Kalash, M., Rochan, M., Mohammed, N., Bruce, N. D., Wang, Y., & Iqbal, F. (2018). Malware classification with deep convolutional neural networks. 2018 9th IFIP International Conference on New Technologies, Mobility and Security (NTMS),

Keahey, T. A. (2013). Using visualization to understand big data. *IBM Business Analytics Advanced Visualisation*, *16*.

Khan, R. U., Zhang, X., & Kumar, R. (2019). Analysis of ResNet and GoogleNet models for malware detection. *Journal of Computer Virology and Hacking Techniques*, *15*, 29-37.

Kim, J., Paik, J.-Y., & Cho, E.-S. (2023). Attention-Based Cross-Modal CNN Using Non-Disassembled Files for Malware Classification. *IEEE Access*, *11*, 22889-22903.

Kolosnjaji, B., Zarras, A., Webster, G., & Eckert, C. (2016). Deep Learning for Classification of Malware System Call Sequences. In B. H. Kang & Q. Bai, *AI 2016: Advances in Artificial Intelligence* Cham.

Kumar, S. (2021). MCFT-CNN: Malware classification with fine-tune convolution neural networks using traditional and transfer learning in internet of things. *Future Generation Computer Systems*, *125*, 334-351.

Le, Q., Boydell, O., Mac Namee, B., & Scanlon, M. (2018). Deep learning at the shallow end: Malware classification for non-domain experts. *Digital Investigation*, *26*, S118-S126.

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *nature*, *521*(7553), 436-444.

Lin, W.-C., & Yeh, Y.-R. (2022). Efficient Malware Classification by Binary Sequences with One-Dimensional Convolutional Neural Networks. *Mathematics*, *10*(4), 608.

Liu, J., Zeng, Y., Shi, J., Yang, Y., Wang, R., & He, L. (2019). Maldetect: a structure of encrypted malware traffic detection. *CMC-COMPUTERS MATERIALS & CONTINUA*, *60*(2), 721-739.

Lo, W. W., Yang, X., & Wang, Y. (2019). An xception convolutional neural network for malware classification with transfer learning. 2019 10th IFIP International Conference on New Technologies, Mobility and Security (NTMS),

Loukadakis, M., Cano, J., & O'Boyle, M. (2018). Accelerating deep neural networks on low power heterogeneous architectures.

Marin, D., Orozco-Rosas, U., & Picos, K. (2022). Malware classification through image processing with a convolutional neural network. Optics and Photonics for Information Processing XVI,

Matplotlib. (2023). *Matplotlib: Visualization with python*. Matplotlib. Retrieved 5 Jan 2023 from https://matplotlib.org/

Meng, X., Shan, Z., Liu, F., Zhao, B., Han, J., Wang, H., & Wang, J. (2017). MCSMGS: malware classification model based on deep learning. 2017 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC),

Moussas, V., & Andreatos, A. (2021). Malware detection based on code visualization and two-level classification. *Information*, *12*(3), 118.

Mumtaz, Z., Afzal, M., Iqbal, W., Aman, W., & Iltaf, N. (2021). Enhanced Metamorphic Techniques-A Case Study Against Havex Malware. *IEEE Access*, *9*, 112069-112080.

Nataraj, L., Karthikeyan, S., Jacob, G., & Manjunath, B. S. (2011). Malware images: visualization and automatic classification. Proceedings of the 8th international symposium on visualization for cyber security,

Nguyen, T.-H., Nguyen, T.-N., & Ngo, B.-V. (2022). A VGG-19 Model with Transfer Learning and Image Segmentation for Classification of Tomato Leaf Disease. *AgriEngineering*, *4*(4), 871-887.

Nisa, M., Shah, J. H., Kanwal, S., Raza, M., Khan, M. A., Damaševičius, R., & Blažauskas, T. (2020). Hybrid malware classification method using segmentation-based fractal texture analysis and deep convolution neural network features. *Applied Sciences*, *10*(14), 4966.

O'Shaughnessy, S., & Sheridan, S. (2022). Image-based malware classification hybrid framework based on space-filling curves. *Computers & Security*, *116*, 102660.

Onoja, M., Jegede, A., Blamah, N., Abimbola, O. V., & Omotehinwa, T. O. (2022). EEMDS: Efficient and Effective Malware Detection System with Hybrid Model based on XceptionCNN and LightGBM Algorithm. *Journal of Computing and Social Informatics*, *1*(2), 42-57.

Pandas. (2023). *Pandas - Python Data Analysis Library*. Pandas. Retrieved Jan 13 from https://pandas.pydata.org

Prajapati, P., & Stamp, M. (2021). An empirical analysis of image-based learning techniques for malware classification. *Malware analysis using artificial intelligence and deep learning*, 411-435.

Rot, A., & Olszewski, B. (2017). Advanced Persistent Threats Attacks in Cyberspace. Threats, Vulnerabilities, Methods of Protection. FedCSIS (Position Papers),

Sahay, S. K., Sharma, A., & Rathore, H. (2020). Evolution of malware and its detection techniques. In *Information and Communication Technology for Sustainable Development* (pp. 139-150). Springer.

Samek, W., Binder, A., Montavon, G., Lapuschkin, S., & Müller, K.-R. (2016). Evaluating the visualization of what a deep neural network has learned. *IEEE transactions on neural networks and learning systems*, *28*(11), 2660-2673.

Schofield, M., Alicioglu, G., Binaco, R., Turner, P., Thatcher, C., Lam, A., & Sun, B. (2021). Convolutional neural network for malware classification based on API call sequence. Proceedings of the 8th International Conference on Artificial Intelligence and Applications (AIAP 2021),

Tahir, R. (2018). A study on malware and malware detection techniques. *International Journal of Education and Management Engineering*, *8*(2), 20.

Wajahat, A., Imran, A., Latif, J., Nazir, A., & Bilal, A. (2019). A Novel Approach of Unprivileged Keylogger Detection. 2019 2nd International Conference on Computing, Mathematics and Engineering Technologies (iCoMET),

Yoo, S., Kim, S., Kim, S., & Kang, B. B. (2021). AI-HydRa: Advanced hybrid approach using random forest and deep learning for malware classification. *Information Sciences*, *546*, 420-435.

Yu, W., Yalin, Y., & Haodan, R. (2019). Research on the technology of trojan horse detection. 2019 12th International Conference on Intelligent Computation Technology and Automation (ICICTA),

Yuan, B., Wang, J., Liu, D., Guo, W., Wu, P., & Bao, X. (2020). Byte-level malware classification based on markov images and deep learning. *Computers & Security*, *92*, 101740.

Yuan, B., Wang, J., Liu, D., Guo, W., Wu, P., Bao, X. J. C., & Security. (2020). Byte-level malware classification based on markov images and deep learning. *92*, 101740.

Zhang, Z. (2018). Artificial neural network. In *Multivariate time series analysis in climate and environmental research* (pp. 1-35). Springer.