Journal of Harbin Engineering University Vol 44 No. 5
ISSN: 1006-7043 May 2023

Efficient Image based Malware Classification Using a Modified VGG based deep Learning Model

Rohit Salota[1], Inderpal Singh[2]
[1,2]Department of Computer Science and Engineering
[1,2]CT institute of Technology and Research, Jalandhar
rohitrajput797@gmail.com, er.inderpal13@gmail.com

Abstract. The rapid development of internet has led to an increase in attack methods and malware. Every year, anti-virus companies find millions of new variations of malware. Several organizations created novel methods to protect persons from such scams. Malware is increasing in frequency, variety, and sophistication. To stop this rise, novel malware detection approaches should be developed. Recent research has shown that deep learning is very effective at detecting malware in images. For malware detection, DL algorithms like modified VGG are used with an image-based malware dataset. The pre-processed images were used to train DL model first. The dataset is later segmented into training and testing data. For the experimental setting, the proposed model, MalNet successfully identified malware images. MalNet was then used to categorize malware images and was compared to other trained models. The suggested method produced very accurate and precise results.

Keywords: Malware Images Classification, VGG architecture, Cyber Analysis, MalNet

1. Introduction

Malware is purposefully harmful software created to harm computer systems. Recently, there has been a noticeable increase in malware used for illegal and malicious purposes. A highly effective method of malware detection is required given the rising trend in malware attacks. The majority of commercial antivirus programs use signature-based methods, which necessitate local signature databases for the storage of patterns that experts have identified in malicious software **[1].** Since malware authors use code reuse to create new malware and employ code obfuscation approaches, this tactic has significant limitations. As a result, many malware infections can go undetected by detection techniques. Static and dynamic analysis have both been tested in recent years for malware detection **[2]**. Detection and classification of malware currently use a variety of DL and ML techniques. The majority of these methods rely on feature database construction through domain expertise. Researchers have employed visualization techniques to address malware family classification issues to lower feature engineering costs and domain expert knowledge **[3].**

The method suggested here tackles these obstacles by employing DL classifier on classification issues employing pre-trained networks and fine-tuning them for malware pictures **[4].** This is in contrast to traditional ML approaches, that utilize training data to develop one hypothesis. However, in contrast to other benchmark ML classifiers, the framework of DL methods for malware evaluation and categorization is shallower. Additionally, it is difficult to train deep networks with small labeled datasets, whereas DL models require an enormous data set made up of annotated pictures **[5].** The main issue is that the majority of visualization techniques compute the texture resemblance of a grayscale image. These methods address the problem of code obfuscation, but they have significant computational expenses when extracting complex texture features from malware images like LBP and GLCM. When used on large datasets, these feature extraction approaches perform less effectively **[6].** Therefore, the main driving forces behind this study were how to lower the cost of feature extraction, extract pertinent data from raw binary data, and boost the precision of malware detection.

1.1 Categories of Malware Analysis

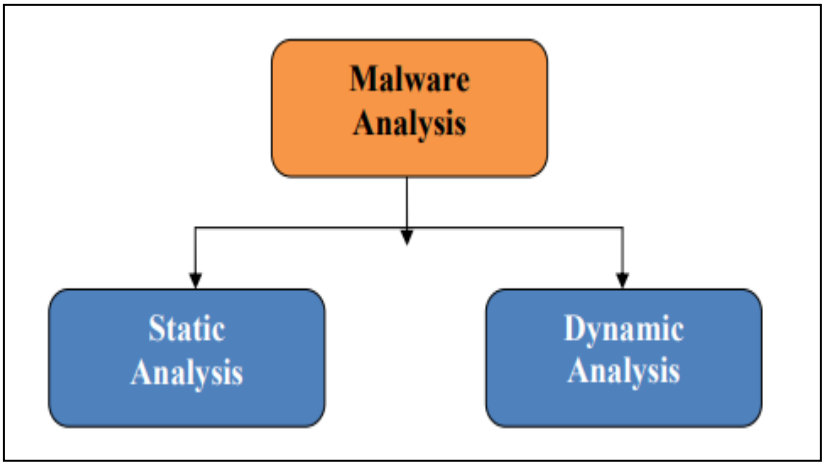

FIGURE 1. Categorization of Malware Analysis

1.1.1 Static Analysis

Static analysis is the practice of examining software without actually running it; this type of methodology can be used on many representations of harmful data binaries. It assists in locating damage to memory and provides numerous examples of how the operating system is proper. The parts of binary files are examined and may be disassembled using tools like IDA. Assembly code, which can be read and understood by people, is possible to generate from machine code. Malware experts are familiar with the instructions for assembling that come with an image of the program to run in order to analyze and learn how to compromise the system. The binary format may be utilized as the foundation for static analysis.

Static malware analysis employs a variety of methods to find dangerous information in binaries. The methods involve obfuscation, binary representation, including file fingerprinting, and operations on the file level, like computation techniques like cryptographic, hashing, and MD5, of the binary in order to differentiate it from similar ones and to confirm that it has not been altered. This software often prints output in the form of status or error messages, which are then included as readable text in the generated binary. Conclusions regarding the internals of the binary under inspection utilized in static analysis may frequently be made by looking at these embedded strings. The greatest and most significant benefit of static malware analysis is that it enables thorough investigation of a given binary, including all potential malware execution routes. Static analysis is safe than dynamic analysis because the source code isn't really run, but it might take a long time & takes a lot of knowledge.

1.1.2 Dynamic Analysis

Dynamic analysis is the process of examining a malware program's interactions with the computer as it runs in a sandbox, virtual machine, or another controlled setting. This is accomplished by seeing and recording the behavior of the malware as it runs on the host computer in virtual machines. Sandboxes are also often utilized for this kind of research. A debugger, like GDB or WinDGB, is used to monitor the behavior of the malware as its instructions are being handled by the processor including their real-time effects on RAM. For the dynamic examination of malware, a number of online automated programs are available, including Anubis and Norman Sandbox. In order to classify malware using similarity measures or feature vectors, the analysis system must have an acceptable description of the malware.

The primary objective of this article is to deal with malware obfuscations to find efficient solutions to malware detection and variant identification problems. Additionally, among different malware-based image families, the quantity of malicious code variants varies considerably **[7].** The biggest obstacle is developing a thorough malware detection classifier that can handle a large number of malicious code variations. The primary goal of

this work is to develop an innovative feature extraction and effective malware image categorization with a minimal running time overhead.

1.2 Malware Images

Malware executables can be compared to a matrix of binary or hexadecimal strings that can be converted into something akin to an image. To create a new variant of malware, malware developers typically add to or update the code in existing malware. As a result, it is much simpler to see tiny additions or modifications to different parts of the file structure when it is presented as an image. Due to the structural similarity of the majority of malware variants, several research studies used digital signal and image processing techniques to categorize malware. They turned the malware codes into grayscale images and found that the structure and texture of malware from the same malware family appear to be quite similar. Image processing techniques are much faster than static and dynamic analysis methods because they do not require disassembly or code execution. The main benefit of this approach is that it is compatible with a variety of malware regardless of the operating system and can handle compressed malware. Additionally, the researchers showed off Search and Retrieval of Malware, an online search and retrieval system that compared binary executables. Additionally, they showed off signal, a malware similarity detection system based on signal processing. Bypassing the time- and resource-consuming unpacking step, it can handle both packed and unpacked samples.

1.3 Malware Detection

Malware's rapid expansion and increasing sophistication present a serious threat to the digital world. Numerous security solutions, such as Anti-Virus (AV) strategies, have been created in order to regulate and reduce the loss brought on by malware. New methods have also been investigated. These AV methods can be divided into two categories: Signature-based methods and Non-Signature-based methods. Scanning is a method used by signature-based antivirus software. It checks for signatures (a certain sequence of bytes) in questionable files. Although this method is quick and provides nearly 100% accuracy for malware that is known to exist, it completely fails to identify "zero-day7" and "unknown8" malware.

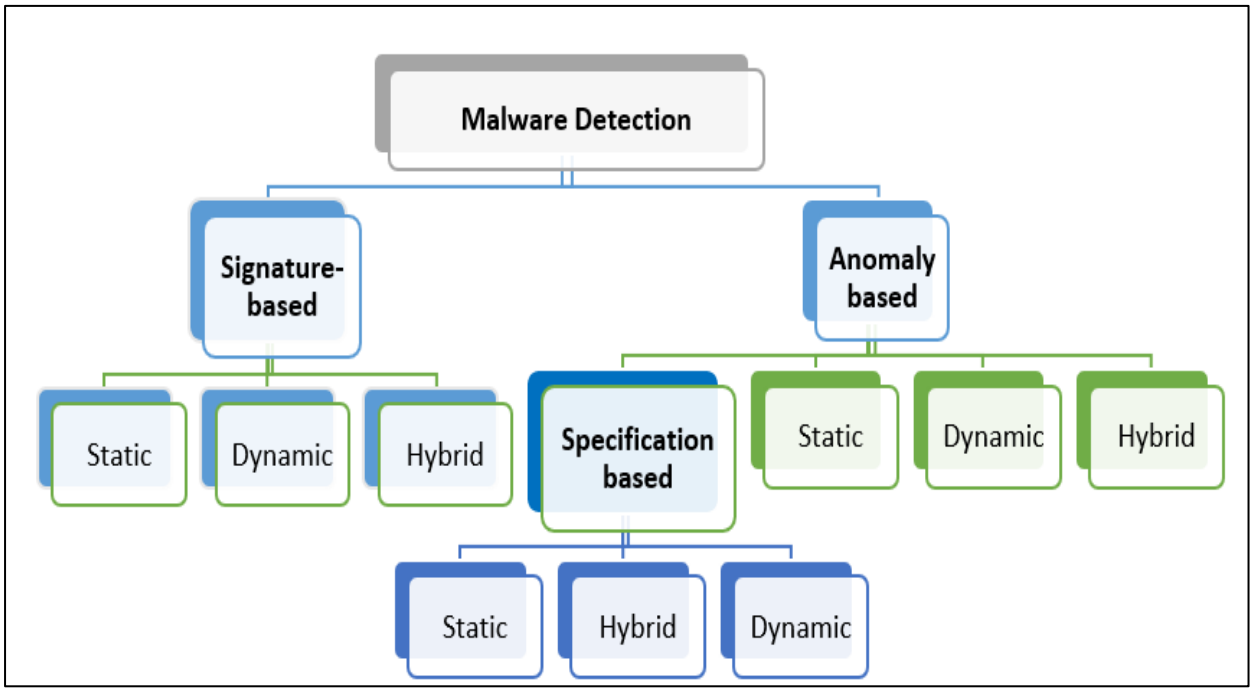

FIGURE 2. Malware Detection

The use of signature-based solutions is constrained by the availability of signature databases. Additionally, the procedure of creating a signature is laborious and complex, which could give an attacker a longer window of opportunity for an assault.

The probability that the attack would be successful for the attacker will be drastically reduced by using established tools and techniques. In order to successfully commit a cybercrime, the attacker must therefore design a new tool, which is why several anti-malware evasion strategies are being created. The most crucial and challenging activity in the anti-malware process is identification, which is one of many duties involved. The anti-virus industry uses many of the malware detection methods that have been covered in literature.

This paper is organized into 5 parts. Section 2 explains the literature review in WSN. Section 3 discusses the proposed methodology. Section 4 gives the particulars of the simulation outcomes. Section 5 finally concludes the overall work done in this research.

1.4 Techniques for Detecting Malware

Anomaly-Based Detection, Signature-Based Detection, and specification-based detection are the two classes in the mobile environment. The way specific techniques collect data to recognize and detect malware dictates a specific study of these techniques**.**

- **Signature-based techniques:**

Malware's destructive actions are detected as signatures. The malware is identified when one of its signatures is detected.

- **Anomaly-based (behavior-based) techniques**

The normal system behaviour is exhibited initially. At that point, the infection is discovered whenever the system's behaviour deviates from the displayed normal behaviour.

- **Specification-based heuristic techniques:**

A method of problem-solving which utilizes practical approach or collection of shortcuts to create answers which are not flawless but are acceptable given a deadline. Artificial intelligence (AI), signature, and anomaly-based techniques are used to improve their proficiency.

1.5 Objectives

- To study various malware classification techniques.
- To develop a lightweight deep learning model for image-based malware classification using a modified VGG architecture.
- To evaluate the performance of the proposed model on benchmark datasets and compare it with state-of-the-art methods.

2. Literature Survey

In this, the Individual malware identification approaches, whether static or dynamic, are insufficient to address this issue **[8].** These two approaches are consequently merged to get around the drawbacks of each methodology used alone. HAAMD provides greater precision and superior results for all of the research when compared with either static or dynamic strategies**.**

It looks into Android's permission-based malware system. The authors present a permission weight strategy, that differs from those used in previous studies **[9].** Every single permission receives a unique score using this technique. After that, a new approach is contrasted with earlier research that made use of K-nearest Neighbor and Naive Bayes methods. KNN's accuracy was 0.96639 and its F-score was 0.96721. Whereas NB had an accuracy value of 0.92437 and 0.92638 F-score**.**

The author used CICA and Mal2017 data records, which include permission and intentions as fundamental features, accessible to everyone **[10].** The authors specifically use a two-layered Mobile malware assessment to address these traits. According to the findings of our study, authors achieved 59.7% accuracy for dynamic categorization at the 2nd layer and 95.3% accuracy for static classification of malware at the first layer**.**

The author describes a novel technique for identifying malware in Android apps that makes use of Gated Recurrent Units, a form of Recurrent Neural Network. The authors extract 2 static aspects from Android applications: API calls and Permissions **[11].** Authors train and test this method on CIC and Mal 2017 datasets.

DL algorithm exceeds numerous methods in terms of accuracy, with a score of 98.2%**.**

In this, the author presented DL approach for Android malware identification. Employing a mobile security framework, they collected permissions, incorrect certificates, and the existence of APK files in the asset folder (MobSF). The five features were then all transformed into vector space **[12].** They used ANN on 600 good and 600 bad apps to gauge the effectiveness of their technique. They achieved a 96.81 percent detection accuracy by using 80% for training and 20% for testing. Similarly, in our situation, we combine different features to create an automated detection system, including the frequency of API calls and permissions**.**

Malware families are detected and identified using refined CNN architecture, which is used by the suggested method, which transforms raw malware binaries into color images. Before fine-tuning, this method handled the imbalanced dataset by using data augmentation to handle the ImageNet dataset (10 million) **[13].** Two datasets: Mailing (9,435 instances) and Android mobile data records (14,733 malware & 2,486 benign instances) were combined to conduct a comprehensive experiment for assessments. IMCFN outshines other DL classifiers, CNNs, according to empirical data, having 98.82% and 97.35% accuracy in Mailing and mobile datasets respectively. Additionally, it shows that colored malware images database outperformed grayscale images in case of accuracy. IMCFN's performance was compared with Google's InceptionV3, ResNet50, and VGG16 because it discovered that the proposed approach is resistant to simple malware obfuscation techniques, like encryption and packing, that are frequently used by hackers**.**

Malware is detected by using Ada boost Technique which is used to increase the efficiency of the hybrid approach which is used to increase the performance of the ML system**[14]**. Using Ada boost technique increases the accuracy by using methods like Decision Tree which gives 98.623%, Naïve Bayes method which gives 79.607% accuracy with the least false positives. Linear SVM gives 96.608% and Random Forest Classifier gives 98.455%.

In this, the author used Convolutional Neural Networks, Caps-Net, and Inception V3 for the classification of Malware Images**[15]**. These are used to save people from deceit. By using these models, we get 90.07% in custom Convolutional Neural Networks, 90% in Caps-Net, 87.10% in the Inception V3 model.

In this, the author detects polymorphic malware and thus, the author cures the malware by using the Decision tree, Convolutional Neural Networks, and Sports Vector Machine**[16].** Thus, we get the result to be as follows: (a) Decision Tree- 99%, Convolutional Neural Networks- 98.76%, Sport Vector Machine- 96.41%.

In this, multiple methods were used for malware detection**[17]**. In method (a), NLP (Natural Language Processing), SVM (Support Vector Machine), CNN (Convolutional Neural Network), MLP (Multilayer Perceptron), XGB (Extreme Gradient Boosting), and RF (Random Forest) were employed, achieving an accuracy of 98.80% and an F-score of 99.00%. Method (b) solely focused on NLP and achieved an accuracy of 99.00%. Method (c) utilized a Fuzzy Pattern Tree and achieved an accuracy of 86.00% and an F-score of 99.00%. These methods demonstrate effective approaches for malware detection and classification.

TABLE 1. Table Literature Survey

Ref. no	Method	Accuracy	F-score
[9]	K nearest neighbors	0.92437	0.92638
[10]	Dynamic categorization of malware Static classification of malware	59.70% 95.30%	-
[11]	CIC and MAL dataset	98.20%	-
[12]	Android malware identification	96.80%	-
[13]	CNN architecture by using (a) Mailing dataset (b) Mobility dataset	98.82% 97.35%	-
[14]	(a) Decision Tree (b) Gaussian Naïve Bayes (c) Linear SBM Accuracy (d) Random Forest Classifier	98.62% 79.60% 96.06% 98.45%	-
[15]	(a) Convolutional Neural Networks (b) Caps-Net (c) Inception V3 model	90.07% 90.00% 87.10%	-
[16]	(a) Decision Tree (b) Convolutional Neural Networks (c) Sport Vector Machine	99.00% 98.76% 96.41%	-
[17]	(a) NLP, SVM, CNN, MLP, XGB, RF (b) NLP (c) Fuzzy Pattern Tree	98.80% 99.00% 86.00%	99.00%
Proposed work	Malnet	99%	90.90%

2.1 Research Gaps

Malware is a growing threat to computer security, and traditional malware detection methods are becoming less effective as malware becomes more sophisticated. Some of the challenges faced by existing network-based malware detection methods are:

The literature review reveals that most of the current methods for malware classification employ complex deep-learning networks, but they still fail to achieve satisfactory accuracy. Deep learning

networks are powerful models that can learn high-level features from large amounts of data, but they also have some drawbacks for malware classification, such as:

- They require a lot of computational resources and training time, which may not be feasible for real-time or resource-constrained applications.
- They are prone to overfitting or underfitting, which may degrade their generalization ability or robustness to new or unknown malware samples.
- They are susceptible to adversarial attacks or perturbations, which may compromise their security or reliability. Therefore, finding simpler and more efficient deep-learning networks for malware classification is an important research goal.
- Image-based malware classification has emerged as a promising approach to detecting malware, but existing methods often require complex architectures and large amounts of computational resources, which limits their practical applications. Some of the limitations of existing image-based malware classification
- According to the literature survey, most of the existing methods for malware classification do not focus on feature extraction, which is an important factor for achieving high performance. Feature extraction is the process of transforming the raw data into a more compact and meaningful representation that can capture the essential characteristics of the data. Feature extraction can enhance the accuracy, efficiency, and robustness of malware classification by reducing the dimensionality, noise, and redundancy of the data. Therefore, developing effective feature extraction techniques for malware classification is a crucial research challenge.

2.2 Proposed Work

Malware is a growing threat approach to detecting malware, but existing methods often require complex architectures for computer security, and traditional malware detection methods are becoming less effective as malware becomes more sophisticated. Image-based malware classification has emerged as a promising and large amount of computational resources, which limits their practical applications. Therefore, there is a need to develop lightweight and efficient deep-learning models for image-based malware classification.

In this proposed research, we aim to address this problem by developing a modified VGG architecture for image-based malware classification. We will evaluate the performance of the proposed model on benchmark datasets and compare it with state-of-the-art methods. Our objective is to improve the accuracy and efficiency of malware detection and enhance the security of computer systems. By developing a lightweight and efficient deep-learning model

we hope to make image-based malware classification more accessible and practical for real-world applications.

2.2.1 Proposed Methodology Workflow

Input: Digital Assets (Image)

Output: Classified Malware Images

Algorithm: Algorithm for Malware Classification	
Input: Sample image of digital assets dataset	
Output: Classified Malware images	
Step 1	Load the testing set image (Sample-image).
Step 2	Set the paths for training and validation datasets based on the Sample-image (Result1).

Step 3	Use VGG16 model to predict the Sample-image (Result2).
Step 4	Use ResNet50 model to predict the Sample-image (Result3).
Step 5	Use Malnet model to predict the Sample-image (Result4).
Step 6	Evaluate the models on the test_generator to obtain the final result (Final-Result).
Step 7	Print the accuracy, precision, recall, and F-score results (Result-print the Accuracy, Precision, Recall, F-Score).

TABLE 2. Algorithm Steps for Malware Classification

3. Research Methodology

Computer security is under increasing threat from malware, and as malware gets more advanced, conventional methods for detecting it are less and less effective. Among the difficulties encountered by current network-based malware detection techniques are**:**

a) The literature review reveals that although most current methods for classifying malware use sophisticated deep-learning networks, they still fall short of achieving acceptable accuracy levels. DL networks are strong models that can extract high-level features from massive amounts of data, but they also have some limitations for identifying malware, including:

- They are prone to overfitting or underfitting, which may reduce their generalization ability or robustness to new or unknown malware samples.
- They demand a lot of computational resources and training time, which may not be practical for real-time or resource-constrained applications.
- They are vulnerable to adversarial assaults or disturbances, which could jeopardize their dependability or security. Therefore, a key research objective is to discover easier and more effective DL networks for malware classification**.**

b) Classification of malware based on images has emerged as a promising method for malware detection, but current approaches frequently need complicated architectures and a lot of computing power, which restricts their usefulness. Some issues are needed to be resolved regarding the current image malware classification. Therefore, to create compact and effective DL models for malware classification using images is necessary.

c) Feature extraction is crucial for achieving high performance and it is not a primary focus of currently used methods for malware classification. The process of turning raw data into a more condensed and meaningful illustration that captures key characteristics of data is known as feature extraction. By lowering the dimensionality, noise, and redundancy of data, feature extraction can improve malware classification's effectiveness and robustness. So, a significant research challenge is to create efficient feature extraction methods for malware classification.

To accomplish the aforementioned goals, the following methodology is used in this work:

In the proposed study, we create a modified VGG architecture for image-based malware classification to solve this issue. We aim to increase the accessibility and applicability of image-based malware classification for real-world applications by creating a compact and effective DL model. MalNet is a modified VGG network that is a proposed architecture for the image-based classification of malware. Pre-trained VGG network serves as the foundation for the first

phase, which extracts deep local attributes from every image. Class-decomposition layer is used to streamline the data distribution's local structure. The model is trained to employ a sophisticated gradient descent optimization in the second phase. The final classification of the image is refined using the class composition layer. By streamlining data distribution and improving the classification process, MalNet architecture is intended to increase the precision and effectiveness of image classification tasks. On several benchmark datasets, we will assess MalNet's effectiveness and contrast it with cutting-edge techniques. MalNet is anticipated to perform better than current approaches in terms of precision and computational effectiveness. The suggested architecture may significantly impact malware detection and boost computer system security.

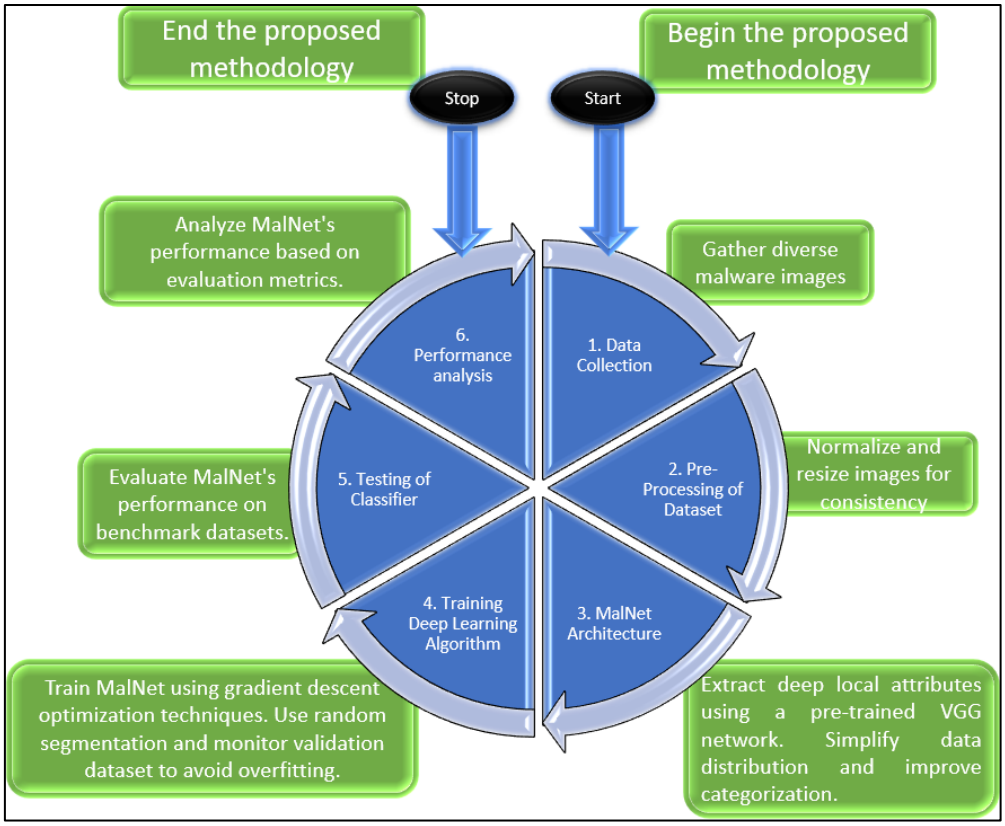

FIGURE 3. Flowchart of the proposed methodology

The steps for the proposed methodology are explained below in detail:

a) Data Collection: To train and test the suggested MalNet architecture, a sizable and varied dataset of malware images must first be gathered. This dataset ought to include different kinds of malware, such as viruses, worms, trojans, etc.

b) Data Preprocessing: To guarantee that images are of the same size and format, the data set gathered will be preprocessed. To guarantee that the values of pixels are within a similar range, the images are normalized. By reducing variation in the data, this step will help MalNet discover underlying patterns more quickly.

c) MalNet Architecture: To implement this, DL frameworks like TensorFlow or PyTorch will be used. Every image's deep local attributes will be extracted using a pre-trained VGG network as a foundation. The local structure of data distribution will be made simpler using the class-decomposition layer, and the final categorization of the image will be improved using the class composition layer.

d) Training: On preprocessed instances, MalNet architecture will be trained using a sophisticated

gradient descent optimization technique, like Adam or RMSprop. The dataset will be randomly segmented into training and validation data & overfitting will be avoided by monitoring the classifier's effectiveness on the validation dataset.

e) Performance Evaluation: On benchmark datasets, trained MalNet architecture will be assessed and contrasted with cutting-edge techniques. Performance assessment will be based on F1 score, recall, accuracy, and precision.

4. Results

a) Accuracy: The no. of true results is divided by the total no. of instances for getting the value of this performance metric. It can be calculated with the help of the equation mentioned below:

$$\boldsymbol{Accuracy} = \frac{TN + TP}{TN + TP + FP + FN}$$

b) F-score: It is a mean value of precision and recall. It acts as a balance between precision and recall.

F-score = $\frac{2*recall*precision}{precision+recall}$

c) Precision: it is used for evaluating performance in text mining like information retrieval. It helps in measuring exactness and completeness.

Precision = $\frac{True\ Positive}{True\ Positive+False\ Positive}$

d) Recall: The ratio of all occurrences accurately identified in the positive class to the total number of real members of the positive class is known as recall. In other words, it tells you how many of the total numbers of positive instances were correctly classified.

Recall= $\frac{True\ Positive}{True\ Positive+False\ Negative}$

TABLE 3. Here is the result of the Performance Evaluation:

Performance Parameter	VGG16	ResNET50	MALNET
ACCURACY	87.10	92.30	98.545455
PRECISION	86.90	92.05	94.736842
RECALL	87.04	92.15	100.000000
F-SCORE	87.09	92.25	90.909091

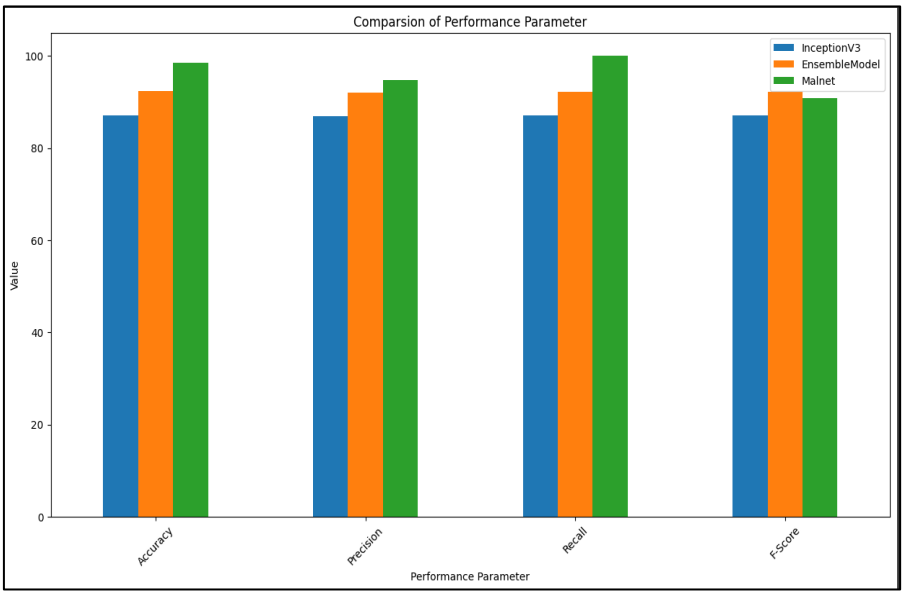

FIGURE 4. Performance Evaluation

In the above graph, it can be seen clearly that Malnet has achieved the highest accuracy (99%), precision, and recall when compared to InceptionV3 and Ensemble models. But the value for F-score is highest in the case of the ensemble model.

TABLE 4. Here is the Comparison of Classifiers:

Performance Parameter	VGG16	ResNET50	MALNET
ACCURACY	80.16	81.20	98.545455
PRECISION	80.00	81.01	94.736842
RECALL	79.90	80.90	100.000000
F-SCORE	80.10	80.16	90.909091

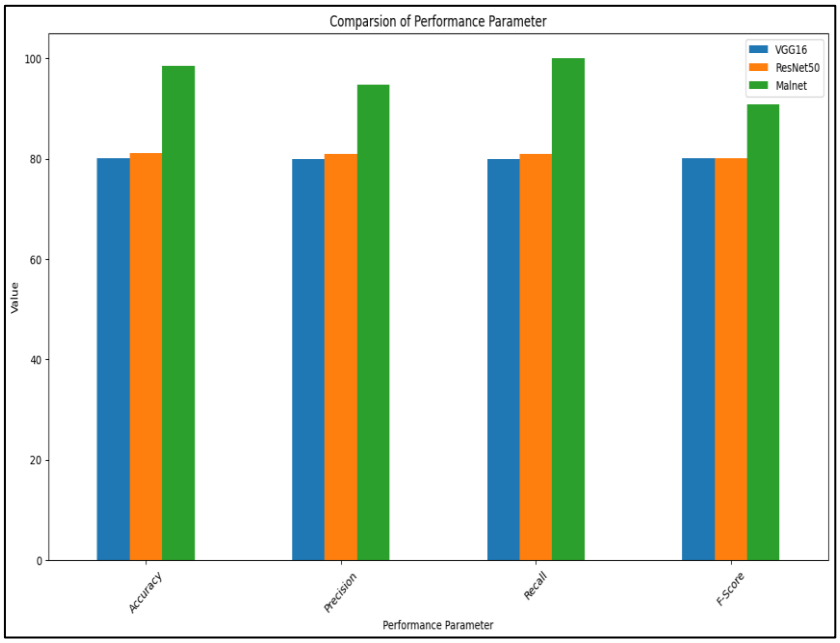

Figure 5. Comparison of Classifiers

From the above figure, we can see that Malnet has achieved the highest accuracy (99%), precision, recall, and F-score when compared to VGG16 and ResNet50. To achieve our objectives, we have applied feature extraction techniques that were not used in earlier studies. In this way, we have achieved better accuracy in every case. From the above results, it is clear that the suggested approach outcomes are outperforming as compared to other approaches.

5. Conclusion

To safeguard digital assets from malware, contemporary anti-malware solutions use ML techniques. While ML-based techniques have shown to be effective at finding new malware, they also have high development costs. It takes a lot of effort from malware analysts to develop a comprehensive set of beneficial characteristics for ML techniques. DL models have proven to be very effective at finding malware. Using a modified VGG architecture, we created and evaluated an innovative malware image classification system. The proposed classifier outperformed existing approaches utilizing comparable benchmarks in its ability to correctly classify the majority of malware samples. While avoiding the manual feature engineering stage, experiments show excellent accuracy rates which are superior to conventional ML approaches. Given that it typically takes very

little time to recognize malware instances, the proposed MalNet is adaptable, practical, and effective.

A further benefit of the suggested model is its 99 % accuracy in correctly diagnosing malware. MalNet is strong enough to recognize malware using image-based techniques, according to a significant accuracy score. To correctly identify complex malware types, future investigations will need to modify the training architecture.

References

[1] Z. Cui, L. Du, P. Wang, X. Cai, W. Zhang, Malicious code detection based on CNNs and multi-objective algorithm, J. Parallel Distrib. Comput. 129 (Jul. 2019) 50–58.

[2] Chaganti, Rajasekhar, Vinayakumar Ravi, and Tuan D. Pham. "Image-based malware representation approach with EfficientNet convolutional neural networks for effective malware classification." *Journal of Information Security and Applications* 69 (2022): 103306.

[3] O'Shaughnessy, Stephen, and Stephen Sheridan. "Image-based malware classification hybrid framework based on space-filling curves." *Computers & Security* 116 (2022): 102660.

[4] Son, Tran The, Chando Lee, Hoa Le-Minh, Nauman Aslam, and Vuong Cong Dat. "An enhancement for image-based malware classification using machine learning with low dimension normalized input images." *Journal of Information Security and Applications* 69 (2022): 103308.

[5] Zou, Binghui, Chunjie Cao, Fangjian Tao, and Longjuan Wang. "IMCLNet: A lightweight deep neural network for Image-based Malware Classification." *Journal of Information Security and Applications* 70 (2022): 103313.

[6] Van Dao, Tuan, Hiroshi Sato, and Masao Kubo. "An Attention Mechanism for Combination of CNN and VAE for Image-Based Malware Classification." *IEEE Access* 10 (2022): 85127-85136.

[7] Paardekooper, Cornelius, Nasimul Noman, Raymond Chiong, and Vijay Varadharajan. "Designing Deep Convolutional Neural Networks using a Genetic Algorithm for Image-based Malware Classification." In *2022 IEEE Congress on Evolutionary Computation (CEC)*, pp. 1-8. IEEE, 2022.

[8] Kishore, B. & Choudhary, M., (2018). HAAMD: Hybrid Analysis for Android Malware Detection. 2018 International Conference on Computer Communication and Informatics (ICCCI).

[9] Kural, O. E., Sahin, D. O., Akleylek, S., & Kilic, E. (2018). New results on permission-based static analysis for Android malware. 2018 6th International Symposium on Digital Forensic and Security (ISDFS).

[10] Kadir, A. F. A., Taheri, L., & Lashkari, A. H. (2019). Extensible Android Malware Detection and Family Classification Using Network Flows and API-Calls. 2019 International Carnahan Conference on Security Technology (ICCST).

[11] Elayan, O. N., & Mustafa, A. M. (2021). Android Malware Detection Using Deep Learning. Procedia Computer Science, 184, 847–852.

[12] Naway, A & Li, Y 2019, 'Android Malware Detection Using Autoencoder', arXiv preprint arXiv:1901.07315.

[13] Vasan, D., Alazab, M., Wassan, S., Naeem, H., Safaei, B., & Zheng, Q. (2020). IMCFN: Image-based malware classification using fine-tuned convolutional neural network architecture. Computer Networks, 171 (2020); 107138.

[14] Bahirithi Karampudi, D. Meher Phanideep, V. Mani Kumar Reddy, N. Subhashini and S. Muthulakshmi, (2023). Malware Analysis Using Machine Learning.

[15] Md Haris Uddin Sharif, Nasmin Jiwani, Ketan Gupta, Mehmood Ali Mohammed, (2023). a Deep Learning Based Technique for the Classification of Malware Images.

[16] Fahd Alhaidari, Nouran Abu Shaib, Maram Alsafi, Haneen Alharbi, Majd Alawami, Reem Aljindan, Atta-ur Rahman, and Rachid Zagrouba, (2022). ZeVigilante: Detecting Zero-Day Malware Using Machine Learning and Sandboxing Analysis Techniques.

[17] Hend Faisal, Hanan Hindy, Samir Gaber, Abdel-Badeeh Salem, (2021). Artificial

Intelligence Techniques for Malware Detection.

[18] Kalash, M., Rochan, M., Mohammed, N., Bruce, N. D., Wang, Y., & Iqbal, F. (2018, February). Malware classification with deep convolutional neural networks. In 2018 9th IFIP International Conference on New Technologies, Mobility and Security (NTMS) (pp. 1-5). IEEE.

[19] Gibert, D., Mateu, C., & Planes, J. (2019, July). A hierarchical convolutional neural network for malware classification. In 2019 International Joint Conference on Neural Networks (IJCNN) (pp. 1-8). IEEE.

[20] Ahmadi, M., Ulyanov, D., Semenov, S., Trofimov, M., & Giacinto, G. (2016, March). Novel feature extraction, selection, and fusion for effective malware family classification. In Proceedings of the sixth ACM conference on data and application security and privacy (pp. 183-194).

[21] Gandotra, E., Bansal, D., & Sofat, S. (2014). Malware analysis and classification: A survey. Journal of Information Security, 2014.

[22] Ni, S., Qian, Q., & Zhang, R. (2018). Malware identification using visualization images and deep learning. Computers & Security, 77, 871-885.

[23] Abusitta, A., Li, M. Q., & Fung, B. C. (2021). Malware classification and composition analysis: A survey of recent developments. Journal of Information Security and Applications, 59, 102828.

[24] Lad, S. S., & Adamuthe, A. C. (2020). Malware Classification with Improved Convolutional Neural Network Model. International Journal of Computer Network & Information Security.

[25] Asam, M., Khan, S. H., Jamal, T., Zahoora, U., & Khan, A. (2021). Malware Classification Using Deep Boosted Learning.

[26] He, K., Zhang, X., Ren, S. and Sun, J., 2016. Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition.

[27] Pehlivan, U.; Baltaci, N.; Acartürk, C. & Baykal, N. The analysis of feature selection methods and classification algorithms in permission-based Android malware detection Computational Intelligence in Cyber Security (CICS), 2014 IEEE Symposium on, pp. 1-8,2014.

[28] Mohata, V. B.; Dakhane, D. M. & Pardhi, R. L. Mobile Malware Detection Techniques International Journal of Computer Science & Engineering Technology (IJCSET), 4, 2229-3345,2013.

[29] Kaur, R.; Kumar, G. & Kumar, K. A comparative study of feature selection techniques for intrusion detection Computing for Sustainable Global Development (INDIACom), 2015 2nd International Conference on, pp. 2120-2124,2015.

[30] Feizollah A, Anuar NB, Salleh R, Amalina F, Maarof RR, Shamshirband S. A study of machine learning classifiers for anomaly-based mobile botnet detection. Malays J Comput Sci 4,2013.

[31] Wu, D.-J., et al. Droidmat: Android malware detection through manifest and api calls tracing. In Information Security (Asia JCIS), Seventh Asia Joint Conference