

شبكة الكشف عن البرمجيات الخبيثة بالاعتماد على الشبكات Res-Former

Transformers ونماذج المحول ResNet-50 الالتفافية العميقة

علي عباس حافظ ديراوي

شركة نقط البصرة - العراق

حيدر عدنان محمد الجاسم

وزارة التربية - العراق

تقى سرحان عبد المحسن العماري

جامعة البصرة للنفط والغاز - العراق

المقدمة:

يمثل كشف البرمجيات الخبيثة تحديًا رئيسيًا في مجال الأمن السيبراني نتيجة التطور المستمر في تقنيات الإخفاء والتعقيد المتزايد للهجمات. وقد أثبتت أنظمة الكشف التقليدية محدوديتها أمام هذه التهديدات، مما يبرز الحاجة إلى حلول ذكية ومرنة تعتمد على تقنيات التعلم العميق. وعلى الرغم من نجاح الشبكات العصبية الالتفافية (CNNs) في استخراج السمات المحلية والهيكلية، إلا أنها تعاني من قصور في تمثيل العلاقات البعيدة بين السمات، مما يحد من فعاليتها في التمييز بين العائلات المتشابهة من البرمجيات الخبيثة. لمعالجة هذه القيود، يقترح هذا البحث نموذجًا هجينًا جديدًا باسم Res-Former يجمع بين قوة ResNet-50 في استخراج السمات المحلية وصلابة الأداء، وبين قدرة نماذج المحولات (Transformers) على تمثيل الاعتماديات البعيدة عبر آلية الانتباه الذاتي، بهدف تحقيق كشف أكثر دقة ومرونة للبرمجيات الخبيثة وتعزيز استقرار القرارات باستعمال تقنيات التعلم

التجميعي

مشكلة البحث:

1. ضعف أداء بعض النماذج التقليدية في كشف البرمجيات الخبيثة، ولا سيما مع البيانات غير المتوازنة والضوضاء.
2. الحاجة إلى نموذج أكثر استقرارًا وموثوقية يحقق دقة عالية ويقلل من الإنذارات

الكاذبة

أسئلة البحث:

1. كيف يمكن دمج ResNet-50 مع المحولات لتحقيق كشف أكثر كفاءة؟

2. هل يساهم إدخال أسلوب التعلم التجميعي (Ensemble Learning) في تحسين الاستقرار وتقليل التباين؟

3. هل يتفوق النموذج المقترح على النماذج الحديثة الأخرى مثل EfficientNetB0 وDNN؟

4. هل يمكن تطوير نسخة خفيفة من النموذج للاستعمال في الزمن الحقيقي؟

اهداف البحث:

1. تطوير أنموذج هجين (Res-Former) يجمع بين الشبكات الالتفافية ولمحولات.

2. تحسين دقة الكشف وتقليل معدل الأخطاء والإنذارات الكاذبة.

3. تعزيز الاستقرار والقدرة على التعميم باستعمال أسلوب التجميع (Bagging & Majority Voting).

4. اختبار النموذج على بيانات Maling ومقارنته بالنماذج الأخرى.

5. استكشاف إمكانية تطوير نسخة خفيفة للاستعمال العملي في الزمن الحقيقي.

اهمية البحث:

1. يقدم إطاراً حديثاً وفعالاً لكشف البرمجيات الخبيثة.

2. يساهم في تعزيز أمن المعلومات والأنظمة الإلكترونية.

3. يفتح المجال لتطبيقات عملية في بيئات الأمن السيبراني.

منهج البحث:

1. تصميم أنموذج هجين يجمع بين ResNet-50 والمحولات.

2. إدخال مصنف تجميعي (Ensemble Classifier) باستعمال Bagging و Voting .

3. تدريب الأنموذج على بيانات Maling وتقييمه بمقاييس الدقة، الاستدعاء وال F1-score

4. مقارنة النتائج مع نماذج حديثة أخرى .

الكلمات المفتاحية: - Res-Former الشبكات العصبية الالتفافية العميقة أنماذج المحولات، التعلم التجميعي، كشف البرمجيات الخبيثة.

Res-Former: An Efficient Malware Detection Network Based on ResNet-50 Deep Convolutional Network and Transformer Models

Tuqa Sarhan³ Hayder Al Jasim² Ali Dirawi¹

Abstract

Malware detection remains a critical challenge in cybersecurity due to the continuous evolution and obfuscation of malicious code. This paper proposes Res-Former, a novel hybrid deep learning architecture that integrates the strengths of ResNet-50 and Transformer models for efficient and accurate malware classification. The proposed model leverages the local feature extraction and noise robustness capabilities of convolutional neural networks while incorporating the global dependency modeling and self-attention mechanisms of Transformer architectures. In the preprocessing stage, malware executables are converted into grayscale images and enhanced using histogram equalization to improve feature discriminability. The ResNet backbone extracts hierarchical spatial and structural features, which are subsequently refined through the Transformer branch to capture long-range inter-feature relationships. Finally, an ensemble learning classifier employing bagging and majority voting ensures stable and generalized decision-making. Experimental evaluations conducted on the Maling dataset demonstrate that Res-Former achieves a superior accuracy of 99.46%, outperforming several state-of-the-art methods including EfficientNetB0, DNN, and IMCBL.

Keywords :Res-Former, Ensemble Learning, Malware Detection, Transformer, ResNet-50.

¹ Basra Oil Company, Rumaila Operating Organization, Basrah, Iraq

² Ministry of Education, General Directorate of Education in Basrah, Basrah, Iraq

³ University of Basrah for Oil and Gas, Chemical and Petroleum Refining Engineering Department, Basrah, Iraq.

Introduction

The accelerated growth of the malicious software or malware has become one of the greatest dangers to current information systems, networks, and infrastructures. With the digital transformation accelerating and the proliferation of interconnected devices in nearly every place, cyberattacks have become increasingly intricate, bigger, and more advanced [1]. To circumvent the old detection systems that have been in use, malware writers are increasingly adopting code obfuscation, polymorphism and metamorphic transformations, making the forward-looking signature-based and rule-based techniques largely useless. This has led to an increased demand of smart, dynamic, and data-driven detection systems that can be able to detect known and new malware forms with high precision and resilience [2,3].

Over the last few years, deep learning (DL) has transformed the malware analysis domain by allowing automatic feature extraction and end-to-end learning based on raw data representations [3]. Convolutional Neural Networks (CNNs) represent one of the most effective deep learning (DL) methods for recognizing spatial patterns in malware imagery, enabling the conversion of binary executable files into visual formats that reveal distinctive structural and behavioral characteristics. ResNet, VGG and EfficientNet are some of the models where strong discriminative performance has been observed because they are able to represent hierarchical and local textual features. Nonetheless, CNN-based techniques have a fundamental disadvantage in the fact that they have local receptive fields, which limit their ability to detect long-range interactions and global links across image patches, a factor that is especially important in separating between malware families sharing subtle visual similarities [4,5].

In response to these constraints, modern research has resorted to Transformer models, which were initially created to perform sequence modeling in natural language processing [6]. Transformers has self-attention which means the model can selectively attend to the most informative parts of the input, therefore, learning non-local dependencies and improving contextual knowledge. Such architectures, including the Vision Transformer (ViT) and its variants, have demonstrated state-of-the-art results in different tasks when applied to vision [7,8]. However, transformer-based-only models often demand high computational

resources and large-scale data sets to reach optimal generalization - which can be problematic in terms of their application in practice in the context of cybersecurity [9,10].

Based on these observations, this paper presents a new hybrid deep learning framework, Res-Former, to leverage the complementary capabilities of ResNet and Transformer networks to detect malware efficiently and with great accuracy. ResNet-50 backbone is a powerful local feature extractor that is useful to extract structural and hierarchical visual feature of malware images. At the same time, Transformer branch further refines these representations with self-attention to allow the model to learn global dependencies and inter-feature correlations that CNNs do not have the ability to capture. This two-branch structure improves the discriminatory ability and the power of generalization of the entire system.

In order to provide sound classification, the last decision step takes the form of an ensemble learning model that combines various base learners using bagging and majority voting techniques. This combination mechanism greatly increases stability of the decisions, reduces the overfitting and increases the strength of the model against noisy or imbalanced distribution of data. The offered Res-Former model was thoroughly tested on the publicly available Malimg dataset, consisting of a wide range of malware families turned into grayscale images. The experimental outcomes prove that the suggested approach has a better accuracy of 99.46, which is superior to a variety of state-of-the-art CNN- and Transformer-based malware classifiers in all key performance measurements.

The key contributions of this work are outlined below:

- Proposing a Hybrid Architecture Res-Former Based on ResNet and Transformer Models:

In this study, we introduce a novel deep neural network architecture, termed Res-Former, which integrates the strengths of ResNet and Transformer models. By combining the local, hierarchical, and textural feature extraction capabilities of convolutional networks with the ability of Transformers to model long-range dependencies and non-local relationships, Res-Former enhances the representation power of the model. This hybrid design significantly improves the network's performance in recognizing complex malware patterns.

- Feature Representation Enrichment through Self-Attention Mechanism:

In the Transformer component, the self-attention mechanism is employed to model interactions among all spatial positions within the feature space, thereby capturing global dependencies. This process enriches the discriminative capacity of feature representations and enhances the model's effectiveness in detecting both structurally similar and previously unseen malware samples.

- **Stable Decision-Making through Ensemble Learning Classifier:**

In the final stage, an ensemble classifier comprising multiple base learners is utilized to ensure robust and stable decision-making. This ensemble structure effectively reduces prediction variance, improves generalization capability, and increases the overall accuracy of the malware detection system.

The remainder of this paper is organized as follows: Section 2 reviews related works in deep learning-based malware detection. Section 3 details the architecture and methodology of the proposed Res-Former model. Section 4 presents experimental results and performance analysis, followed by the conclusion in Section 5.

Related Works

Authors in [11] discussed the incessant development of malware and the drawbacks of the traditional signature-based method of detecting them by discussing the concept of visual malware classification where binary files are transformed into grayscale images and then analyzed to identify patterns. They suggested a hybrid deep learning model that combines ConvNeXt-Tiny in extracting local features and Swin Transformer in global context perception. Benchmark dataset (Maling, MaleVis, VirusMNIST) with 61 malware classes showed better results, with an accuracy of 94.04% - outperforming baselines of individual models. The interpretability of Grad-CAM was used in the study to demonstrate the complementing nature of CNN and Transformer elements.

In [12], the authors addressed the growing complexity of malware and the shortcomings of the traditional malware detection methods by investigating the image-based malware classification with deep learning. They designed a superior model that used both ResNet-152, a CNN model, and the Vision Transformer (ViT) to assess their respective capabilities in the image classification of malware images based on executable files. There were 6,137 benign and 9,861 malicious samples, which were converted to grayscale images and transformed to hexadecimal. ResNet-152 took

floating-point pixel values, and ViT took unsigned integer pixel representations. The 10-fold cross-validation used in experimental evaluation revealed the high accuracy of the system, which proves the efficiency and strength of the system.

In [13], the authors introduced a new framework of malware visualization and classification based on the structured format of the executable files. It was noted that executables were made out of separate sections each with distinguishing structural and semantic characteristics hence the study suggested splitting images of grayscale malware into sections based on their categories and creating several sub-images that denoted different functional parts. Multi-channel inputs of deep convolutional neural network were then pooled with these sub-images to improve the accuracy of classification. The experimental findings showed that the various section-wise image classes had high levels of discriminative features that were favorable to malware detection.

In [14], the authors proposed a new family of malware malicious classifier based on high-resolution grayscale images and multiple instance learning (MIL) as a solution to the issue of adversarial binary enlargement. In contrast to traditional visualization-based techniques that are based on lossy resizing operations, they maintain important structural information by dividing malware images into smaller patches. Every patch is introduced and learned with the help of a convolutional neural network where an attention-based aggregation mechanism is applied to improve the accuracy of the classification. The experimental assessment of the Microsoft Malware Classification Dataset showed that there was better performance of the adversarially enlarged samples as compared to the baseline models.

In [15], the authors considered the growing risk of malware propagation and the necessity of quick and precise detection by using deep learning methods. Although convolutional neural networks (CNNs) have been developed through ensemble and transfer learning, limited information exists on systematic comparisons of their outcomes in malware family recognition. In order to narrow this divide, the authors created an ensemble CNN-based model that can classify 34 malware families using a combination of the Microsoft Malware and Maling datasets. The Least Absolute Shrinkage and Selection Operator (LASSO) was also used to optimize feature selection in order to minimize the computational cost and

training time. The results imply that the technique is a viable and scalable tool when it comes to real-life malware detection and classification.

In [16], the authors have provided a new multi-task learning model to classify malware image in order to have a fast and precise detection of various types of malware. The method transforms malware properties into BMP and PNG images, which a deep learning-based classifier is used to process. The model was tested on a new curated dataset of about 100,000 benign and malicious samples that consisted of PE, APK, Mach-O, as well as ELF executables. The experimental results of seven tasks and four activation functions (ReLU, LeakyReLU, PReLU, ELU) had the highest accuracy in the PReLU-based configuration.

Motivated by the escalating need to classify malware correctly and efficiently, in [17] the authors presented a problem statement of the necessity in the enhancement of the current condition of the malware classification process and the development of new and more sophisticated methods to conduct it. In the quest to improve the detection performance, they proposed LeViT-MC, a new architecture, a combination of the principles of the vision transformer with malware visualization based on images and advanced transfer learning. The model was tested on the MaleVis dataset to classify multi-class malware and showed better results in comparison with the current strategies. The accuracy of LeViT-MC was state-of-the-art with a higher computational efficiency; hence it could be applied to real-time detection settings. The paper shows that the integration of transformer-based architectures with image-based learning can significantly enhance the modern malware analysis and classification systems.

In [18], the authors examined the solution of the problem on hardware malware detection as a thankful step towards enhancing the security of the system with the aid of microarchitectural data. Specifically, Hardware Performance Counter (HPC) data was used to design the experiment and machine learning and deep learning models were used to determine and classify malware. In order to minimize the false positives and maximize the detection rates, a more adaptable detection system involving Decision Tree (DT), Random Forest (RF), Logistic Regression (LR), Support Vector Machine (SVM), and Few-Shot Learning (FSL) techniques were proposed. The model was highly accurate (97 percent) on two different datasets of the internet which is superior to the conventional techniques.

In [19] authors have studied the use of the convolutional neural networks (CNNs) to classify malware because of the increased complexity and obfuscation of the current malware, such as polymorphism, metamorphism, and encryption. The paper utilized malware binaries as RGB and grayscale pictures to capitalize on the features of CNNs to acquire global structural patterns. The authors employed fine-tuning methods including bicubic interpolation, ReduceLROnPlateau, estimation of class weight, with the assistance of the transfer learning method based on MobileNet to maximize the generalization and minimize the overfitting. The model was experimented on 4 datasets that were; Microsoft Big 2015, Malimg, MaleVis and a Fusion dataset, and high accuracies of 96.04-99.08 were achieved showing the strength of the model.

In [20], authors have touched the new security issues in Internet of Things (IoT) network whereby the various and dynamic networks lead to the large attack surface on malware. In order to enhance real-time malware detection, they introduced a two-dimensional (2D) CNN architecture that is specialized on the Malimg dataset with data augmentation methods to increase training data and generalization. The model automatically derives features directly out of malware images unlike the previous methods that used manually designed features, which minimizes overfitting and the model trains effectively within ten epochs. The performance was also evaluated to be better than the existing models like DenseNet, VGG16, and VBDN, with an accuracy of 98.86. The research highlights the relevance of dataset-optimized solutions and advanced CNN models in addressing the complex IoT malware detection problems and outlines the necessity of changing approaches to protect the new networked spaces.

In [21] authors suggested Image-based Malware Classification using Broad Learning (IMCBL) as a non-GPU based method to solve the problem of detecting new and unknown malware, such as encryption, packing and biased samples. The technique transforms malware binaries into images, uses truncated Singular Value Decomposition (SVD) to reduce features, and uses a Broad Learning (BL) architecture, which projects input features onto enhancement nodes in a flat, incremental architecture. Such design allows effective training, reduces overfitting, and removes retraining. The high accuracies of 93.04% to 97.64% achieved on the experiments of five benchmark datasets, such as Microsoft Windows, Malimg, ionet-Android, Big windows and obfuscated windows malware, are robust against even

obfuscation. IMCBL also performed well compared to the traditional and deep learning models including VGG16, ResNet50, and InceptionV3, in computational efficiency, which demonstrates its feasibility in malware classification in the real world.

In [22], the authors suggested VBDN, a malware detection model, which is founded on convolutional neural networks (CNNs), to overcome the limitation related to unbalanced datasets, lack of generalization of the algorithm, and time-consuming processing. The model combines data visualization, balanced sampling, and data augmentation to improve classification and reduce overfitting. Test on four open malware datasets indicated that VBDN attained more than 90 percent accuracy on all datasets. Another aspect that was also highlighted in the study was the per-class recognition analysis, through confusion matrices, which offers actionable information to cybersecurity researchers. Moreover, balanced sampling methodology minimizes hardware use, does not require building new datasets, and automates weight testing, which indicates the efficiency and feasibility of the framework in comparison with the state-of-the-art malware detectors.

In [23], the authors responded to the increasing security challenges in Internet of Things (IoT) settings where metamorphic malware presents a major threat to privacy, monetary wealth and organizational image. In order to increase the detection, they utilized a deep learning-based model with six architectures of VGG16, InceptionV3, CNN, ResNet50, MobileNet, and EfficientNetB0 on publicly available Malimg data. The paper has focused on dynamic analysis to obtain executable behaviour and used this data to classify, dwarfing the shortcomings of such detection techniques, which were based on signatures. As experimental results proved, the given approach obtained a 99 percent accuracy and 97 percent F1-score, which outperforms the existing machine and deep learning models. These results indicate the possibility of deep learning methods to detect malware of powerful metamorphic malware in IoT systems.

Proposed Method

In this paper, we propose an effective and innovative approach named Transformer Branch ResNet (Res-Former) for malware detection. The overall structure of the proposed method is designed based on the integration of the deep convolutional neural network ResNet with an additional branch that incorporates a Transformer module. The objective

of this integration is to leverage the complementary advantages of both architectures—namely, the ability of ResNet to extract local and hierarchical features, and the capability of the Transformer to model global dependencies and non-local relationships within the data. In the first stage, malware images—converted from executable files—are fed into the ResNet network. Through its residual block architecture, ResNet effectively captures local, textural, and structural features of the images across multiple hierarchical levels. By learning deeply from the data, the network identifies repetitive patterns, distinctive textures, and subtle structural characteristics associated with malware behavior, transforming them into rich and compact feature vectors. The output of this stage is a set of high-level feature representations that encapsulate comprehensive visual information about the image, including its spatial patterns and internal structures. Subsequently, the extracted features are passed into the Transformer branch. The Transformer module, utilizing the self-attention mechanism, models the interrelationships among all spatial positions within the feature vectors, thereby capturing long-range dependencies and non-local correlations among different regions of the image. By assigning adaptive importance weights to each feature relative to others, this module refines the final data representation, enabling a more precise depiction of complex inter-component interactions within the feature space. In the final step, the extracted and refined features are forwarded to an ensemble learning classifier consisting of multiple base learners. This ensemble framework ensures robust and stable decision-making in the final malware classification stage. It effectively reduces prediction variance, enhances generalization ability, and improves the overall accuracy of the detection system. This hybrid design enables the proposed Res-Former model to combine the local feature extraction and noise robustness capabilities of convolutional networks with the contextual understanding and attention-based relational modeling strengths of Transformer architectures. As a result, Res-Former achieves more accurate, stable, and generalizable performance in detecting unknown and sophisticated malware. Fig. 1. illustrates the overall architecture of the proposed network.

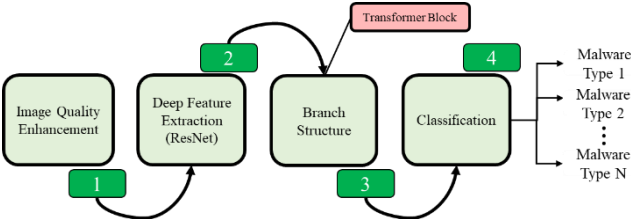


Fig. 1. The Overall Diagram of Proposed Method.

Image Preprocessing

At the preprocessing phase, malware images are initially resized to a standard set of dimensions that are compatible with the input layer of the ResNet network, which makes them fully conform to the structural requirements of the architecture. After resizing, Histogram Equalization is then used to enhance the visual quality of the data and the discriminative power of the extracted features. This operation redistributes the intensity values of the image pixels and also enhances contrast, thus making fine details more visible producing a more consistent overall brightness. As a result, the textural and structural details within the image become more visible and salient, and vital regions can be more easily identified by the model. The resulting preprocessed images therefore have higher levels of clarity, informativeness, and feature discriminability, which provide a better basis for the subsequent stages of feature extraction and malware classification.

Architecture of the Res-Former Model

A new neural network model is used in the suggested approach, called Res-Former to extract the features of malware images. In this context, the malware images are first sent to the ResNet backbone, where the discriminative features are obtained by the intermediate and deep convolutional layers. The obtained feature maps are then passed to the Transformer module to have the feature representation improved. This architecture allows the attention mechanism in the Transformer to extract salient features, and the convolutional modules to simultaneously analyze and extract important spatial patterns of the images. The integration of these complementary processes makes the model effective in creating strong feature representations of malware. The Res-Former network therefore benefits from both the local feature identifiability and noise

resistance properties of convolutional networks as well as the relational modeling capabilities of Transformer networks. Consequently, the suggested model has high accuracy, stability, and generalization during malware detection. Fig. 2 represents the architecture of the proposed Res-Former network.

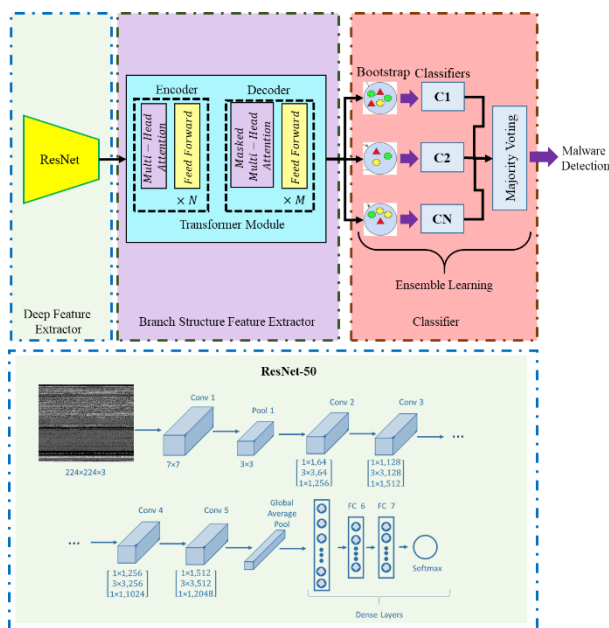


Fig. 2. The Proposed Res-Former Architecture.

References

ResNet-50 architecture is one of the most notable designs in the deep convolutional neural network, in which vanishing gradient problem is successfully addressed. The essence of this architecture is the fact that it uses skip connection or residual connections, whereby information can pass through the intermediate layers without alteration. These links create direct gradient flows, therefore, deep networks can be trained more easily and faster, with more stable convergence. A basic block of ResNet, called Residual Blocks, is used in more sophisticated ones (like ResNet-50), which are implemented as Bottleneck Blocks. In these blocks there are layers with the same number of filters that have the same feature map dimensions. The number of filters is doubled whenever the size of the

feature map is reduced by a half to ensure a balanced trade-off between the spatial resolution and the feature depth. Down-sampling is achieved with 2-stride convolutional layers and Batch normalization, which is applied after the convolution and before the ReLU activation function, is used to increase training stability and reduce the time required to reach convergence. Identity Shortcuts are applied in the ResNet architecture in cases when the input and output dimensions are the same, which guarantees the dimensional consistency in each block. When dimensions differ, 1x1 convolution Projection Shortcuts are used to align them. This mechanism allows dimensional paths transmitted to learn more complex and varied forms of representations. The network is able to extract a number of features (low-level features, e.g., edges, textures) and complex features (e.g., complex structures and semantic patterns) hierarchically after passing through several residual blocks. A fully connected layer with 1000 units and a Softmax activation function are used to project the extracted features into the label (class) space. In the suggested model, this architecture can be served as an effective and noise-insensitive feature extractor, which offers strong and discriminatory representation of malware images, which would be the basis of further Transformer-based analysis.

Transformer Module Branch

Transformer module is one of the components that are used in the proposed model after the ResNet backbone to analyze and model the dependence of high-level features on a global scale. The aim of incorporating this module is to improve the interpretation of the inter feature correlations, detect deep hierarchical patterns and assimilate the contextual details of the obtained representations. Transformer architecture comprises several layers of the encoders and decoders, in a stacked and sequential order. At every level, the Self-Attention mechanism is used to encode the relationships between the input elements and dynamically focus on the most important and specific features. This hierarchy allows the model to learn simpler and more general representations by initially learning simple and low-level patterns in the early layers and gradually incorporating them with the contextual information in deeper layers. Through this process, the model efficiently learns both the short-range and the long-range dependencies and at the same time processes the entire sequence of features in parallel, resulting in a more comprehensive

representation. Fig. 3 shows the general framework of the Transformer module used in the proposed Res-Former model.

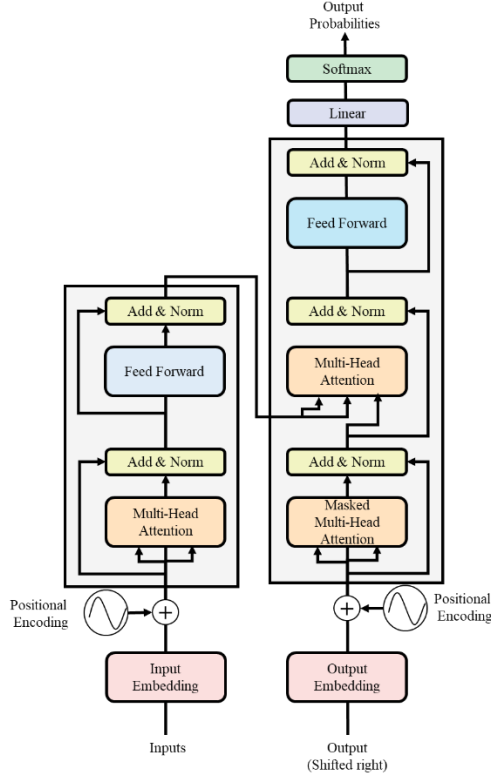


Fig. 3. Structure of The Transformer Module.

The input data of a Transformer network replace this sequence of feature vectors ($x_1, x_2, \dots, x_T$) of sequence length (T), where $x_t \in \mathbb{R}^d$ represents the features at time t . To encode the positional information of every element in the order, a positional encoding vector (PE) is appended to each input vector as follows:

$$z_t = x_t + PE_t \quad \text{for } t = 1, 2, \dots, T \quad (1)$$

In this case, (PE_t) is the positional information which is associated with the position (t).

$$PE_{(t, 2i)} = \sin\left(\frac{t}{10000^{\frac{2i}{d}}}\right) \quad (2)$$

$$PE_{(t,2i+1)} = \cos\left(\frac{t}{10000^{2i/d}}\right) \quad (3)$$

here, (i) denotes the component index and (d) denotes the feature vector dimension. Scaled Dot-Product Attention is the fundamental workflow of the Transformer that performs the calculations of the relations between all the elements of the sequence in parallel. Given every input, three vectors, Query (Q), Key (K), and Value (V) are acquired by multiplying the input matrix with learnable weight matrices:

$$Q = ZW^Q. \quad K = ZW^K. \quad V = ZW^V \quad (4)$$

Where $Z = [z_1, z_2, \dots, z_T]^T$ represents the input matrix. The attention output is determined as:

$$Attention(Q, K, V) = Softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V \quad (5)$$

The scaling factor $(\sqrt{d_k})$ stabilizes the gradients during training. To make the model more effective in representing the idea of complex dependencies, the Transformer uses Multi-Head Attention (MHA), so that multiple attention heads are active in parallel, and their outputs are concatenated:

$$MHA(Q, K, V) = cat(head_1, \dots, head_h)W^O \quad (6)$$

$$head_i = Attention(QW_i^Q, KW_i^K, VW_i^V) \quad (7)$$

Here, W^O is the output projection matrix. Transformer blocks have two principal elements, which are the Multi-Head Attention (MHA) and the Feed-Forward Network (FFN). The FFN leverages two linear layers and a ReLU between them, which is as follows:

$$FFN(x) = max(0, xW_1 + b_1)W_2 + b_2 \quad (8)$$

The Transformer includes a residual connection and then layer normalization in every sub-layer that enhances training stability and converges faster. Normalization is done in the following way:

$$H' = Norm(Z + MHA(Z)) \quad (9)$$

$$H = Norm(H' + FFN(H')) \quad (11)$$

where H' and H are the normalized outputs of the Multi-Head Attention and Feed-Forward layers. Transformer block results in a matrix of the representations of the vectors of all time steps. The temporal Avenue Pooling is used to extract the final representation of features in the time dimension:

$$f = \frac{1}{T} \sum_{t=1}^T H_t \quad (11)$$

where (f) is the extracted final feature vector. Overall, the inclusion of the Transformer module allows the model to learn long-range dependencies efficiently with parallel learning, target important regions in the data with attention, and learn through parallelization. The combination of these properties makes the model more accurate, robust and also more effective in the overall performance of the model, in identifying complex and sophisticated malware patterns.

Classification Using Ensemble Learning

A further Ensemble Learning strategy is used to refine the results of the classification in the final phase of the proposed approach. Within the proposed model, the decision is made based on the sum of the results from several independent classifiers, which enhances the accuracy and strength of the system [24]. To this end, the training set is segmented into specific subsets, and a specific base classifier is used to process each subset. Within this context, a Decision Tree is utilized as the base learner, due to its strong capability of modeling nonlinear relationships and capturing complex feature patterns, which contributes to improved classification accuracy. Furthermore, the use of slightly overlapping sets of data also increases the chances of the accurate recognition of hard to classify samples, and, consequently, improves the overall accuracy of the system. To implement the ensemble strategy, two methods—Bagging (Bootstrap Aggregating) and Majority Voting—are applied. The Bagging process creates several training subsets through sampling with replacement, and each sub-sample is assigned to an independent classifier. These classifiers are then aggregated based on their outputs, which minimizes model variance and enhances the model's stability when handling noisy or imbalanced data. After that, the Majority Voting strategy combines the predictions of all base classifiers, and the final result is determined by the class receiving the highest number of votes. The combination of these two methods contributes to improved classification accuracy, reduced prediction error, and enhanced ability to identify complex malware patterns. Therefore, applying the ensemble learning framework provides an effective and scalable solution for dependable and accurate malware detection. The

general framework of the ensemble-based classification process is presented in Fig. 4.

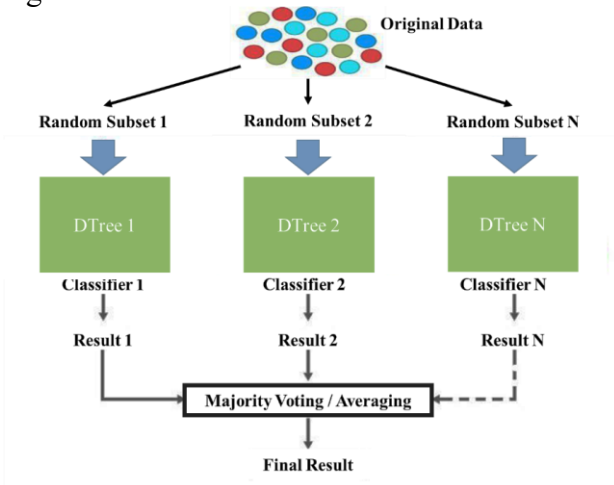


Fig. 4. the classification procedure using ensemble learning.

Results

This part of the paper is devoted to the analysis and comparison of the outcomes acquired during the process of simulation of the proposed model. The MATLAB 2024 was used to simulate the experiments. Moreover, this section presents the evaluation measures used, a description of the dataset considered, and an analysis of the simulation results.

Dataset

In this study, publicly available Maling dataset in the Kaggle repository [25] was used. The Maling dataset is an open-source dataset that is commonly used to evaluated malware classification techniques. It is composed of 9,435 malware executable files which belong to 25 different classes. The malware samples are converted into 32x32- pixel images with the nearest-neighbor interpolation. Table 1 presents a detailed breakdown of the malware types, and Fig. 5 shows the examples of three malware classes.

Evaluation Metrics

In order to assess the performance of the proposed system, some statistical parameters were examined, such as Accuracy, Precision, Recall and F1-score. Accuracy metric is defined in (12).

$$Accuracy = \frac{(TP+TN)}{(TP+FP+TN+FN)} \quad (12)$$

Precision (P) is the percentage of the instances that are correctly identified among all the predictions and is presented in (13).

$$Precision = \frac{TP}{TP+FP} \quad (13)$$

Recall (R) represents the percentage of correctly recognized relevant instances within the complete number of relevant instances in the test set. It is defined in (14).

$$Recall = \frac{TP}{TP + FN} \quad (14)$$

Lastly, F1-score is a harmonic mean of Precision and Recall, and it is calculated in terms of the following (15).

$$F_1score = \frac{2*Percision*Recall}{(Percision+Recal)} \quad (15)$$

TABLE I.
DISTRIBUTION OF SAMPLES IN THE MALIMG DATASET

#	Class	Family	#	Class	Family
1	Worm	Allapple L	14	Trojan	Aluero n.gen!J
2	Worm	Allapple A	15	Trojan	Malex. gen!J
3	Worm	Yuner A	16	PWS	Lolyda AT
4	PWS	Lolyda AA 1	17	Dialer	Adialer .C
5	PWS	Lolyda AA 2	18	TDow nloader	Wintri m BX
6	PWS	Lolyda AA 3	19	Dialer	Dialpla tform B

7	Trojan	C2Lop .P	20	TDow nloader	Dontov o A
8	Trojan	C2Lop .gen!g	21	TDow nloader	Obfusc ator.A D
9	Dialer	Instant access	22	Backd oor	Agent. FYI
10	TDow nloader	Swizzo t.gen!I	23	Worm AutoIT	Autoru n K
11	TDow nloader	Swizzo err.gen!E	24	Backd oor	Rbot!g en
12	Worm	VB.AT	25	Trojan	Skintri m N
13	Rogue	Fakere an			

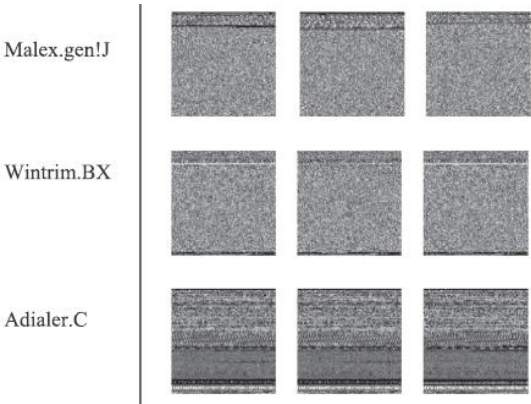


Fig. 5. Different Types of Malware Class.

Equations (12–15) represent the notations True Positive (TP), False Positive (FP), False Negative (FN), and True Negative (TN):

- True Positive (TP): Instances that are correctly classified in the positive category.
- False Positive (FP): Cases that are incorrectly identified as positive when they should be negative.
- False Negative (FN): Cases that are incorrectly classified as negative although they are actually positive.

- True Negative (TN): Predictions that are correctly identified as belonging to the negative class.

Results Evaluation

This section examines the performance of the proposed approach and the simulation results of the approach and compares it to other approaches. The performance measures, such as Precision, Accuracy, Recall, and F-score are summarized and presented in a series of figures and tables. To achieve this, the dataset was split into two subsets (training and testing sets) for experimentation. The training stage involved the use of training subset to adjust the weights of the neural network and optimize the learning process. The testing set, was used to assess the ability of the model to generalize and accuracy in unknown situations. In this research, 70% of the samples were allocated to the training set and the remaining 30% were reserved as the independent test set. To enhance the training procedure and assess the stability of the model, 10-fold cross-validation was applied to the training subset. In this process, the training data were partitioned into ten folds, and in each iteration, nine folds were used for model training while the remaining fold served as the validation subset. This procedure was repeated ten times to ensure that every training sample contributed at least once to the validation phase. After completing the cross-validation process and finalizing the model configuration, the overall performance of the proposed method was evaluated on the independent test set (30% of the data), and the corresponding results were reported. The simulation settings used to ensure the reproducibility of the experimental results are presented in Table II. In this configuration, the Adam optimizer with an initial learning rate of 0.001 is employed to provide stable and well-controlled convergence during training. To prevent overfitting, an early-stopping mechanism with patience = 10 epochs is activated, and the maximum number of training epochs is set to 40, thereby avoiding unnecessary training once the model reaches performance saturation. A batch size of 200 is selected to achieve an effective balance between gradient stability and computational efficiency. Within the Transformer component, four attention heads are utilized to enable simultaneous modeling of diverse relational patterns among feature representations, while the Embedding maxPosition = 256 parameter is applied for positional encoding to ensure accurate processing of spatial and sequential dependencies in the attention

space. Collectively, these settings establish a coherent and reproducible experimental framework for evaluating the performance of the proposed model.

TABLE II.
EXPERIMENTAL SETUP FOR PROPOSED METHOD

Item	Setting / Description
Optimizer	Adam
Initial learning rate	0.001
Data split	70 -30 (Train - Test)
Early stopping	Patience = 10 epochs
Maximum epochs	40
Batch size	200
Number of attention heads	4
Embedding maxPosition	256

Evaluation of the Training Process for Ensemble Learning

The learning curve of the proposed model in the training process is shown in Fig. 6. As illustrated, the top graph (in blue) is the change in accuracy with each training iteration, whereas the bottom graph (in orange) is the loss function trend. The horizontal axis in both plots represents the number of iterations (epochs) and the vertical axes represent the accuracy (in percentage) and the loss value, respectively. At the early stages of training (Epoch 1), there is an increase in accuracy, where the model rapidly improves from an initial low accuracy (below 40 percent) to above 80 percent. The accuracy also improves gradually as the number of iterations increases, and it reaches about 97, which means that the model converges and learns the pattern within the data. In the meantime, the curve of loss in the bottom figure shows an even decreasing trend. Initially, the

loss value is quite high (above 3); however, it quickly drops as training progresses and eventually converges to near-zero values after a few epochs. This trend shows that there is a constant improvement in the performance of the model and an ever-smaller prediction error through time. Overall, the static nature of the two curves demonstrates that the given model can learn effectively. The fact that the loss is steadily decreasing and the accuracy, at the same time, is growing is evidence that the model is well fitted to the training data and not in danger of overfitting or oscillating learning is present. The high accuracy and low values of loss obtained throughout the process of the study attest to the appropriate convergence and strong training behavior of the proposed ensemble learning model.

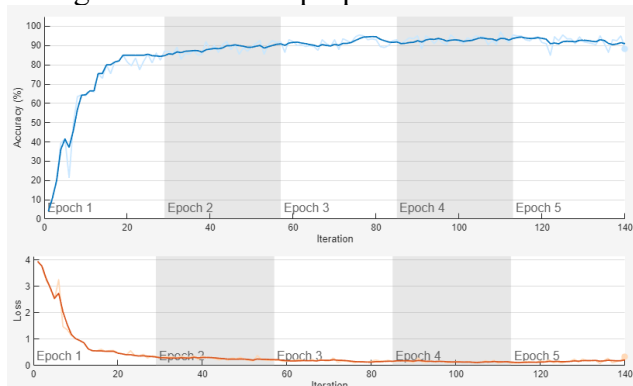


Fig. 6. Learning curve of the proposed model.

Evaluation of the Test Process for Ensemble Learning

Fig. 7 presents the confusion matrix of the results of the malware dataset classification. The confusion matrix gives a list of the individual performance of the classifier in each individual class. The accuracy of each malware class is shown by the column that represents the same in the right side of the matrix. The model proposed is classified with 100 percent accuracies of all classes except classes 6, 21, and 25 that have classification accuracies of 93.7, 99.0 and 95.0 respectively. Such findings depict the great efficiency and reliability of the suggested ensemble-based malware image classification system. It should be mentioned that the results of this confusion matrix are the results of one run and slight deviations could be observed in other runs. The end reported results were taken to be the

average results of the 50 independent runs so that the final results could be fairly and consistently compared with other methods.

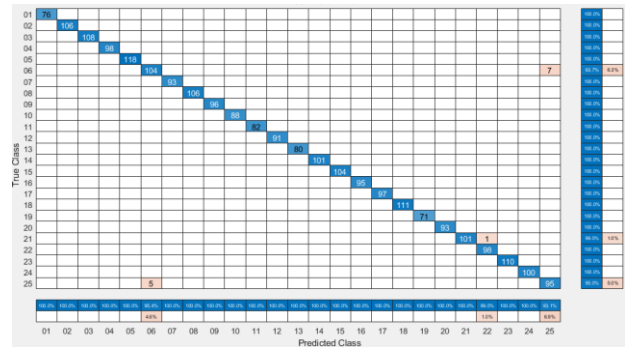


Fig. 7. Confusion matrix of the proposed ensemble classifier.

Receiver operating characteristic (ROC) Analysis

The Receiver Operating Characteristic (ROC) curve is one of the evaluation metrics that provides a graphical explanation of the effectiveness of the classifier. It illustrates the trade-off between the True Positive Rate (TPR) and the False Positive Rate (FPR). The closer the curve lies to the upper-left corner of the graph, the higher the detection capability, whereas the diagonal line represents random classification performance. As can be seen in this figure, the AUC of the proposed method is reported to be 0.946789. Fig. 8 shows the improvement in classifier accuracy in the proposed system as demonstrated by the ROC analysis.

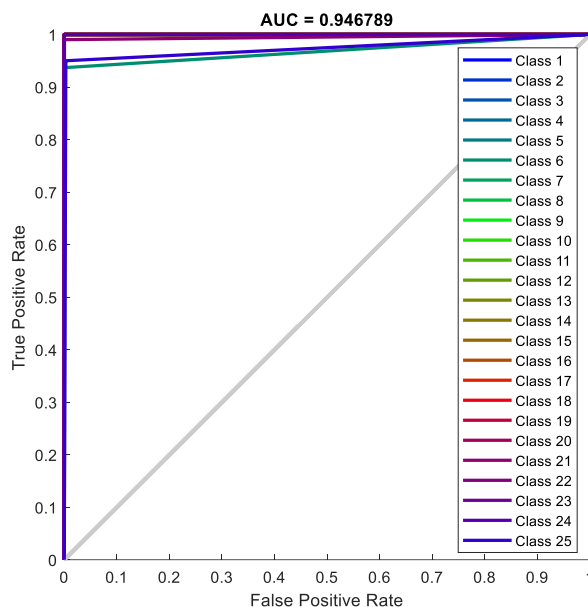


Fig. 8. ROC curve.

Results Comparison and Discussion

A comparison of the suggested Res-Former ensemble framework with recent malware classification techniques, in terms of Accuracy, is presented in Table III. As shown, the proposed approach yields 99.46% accuracy, which is the highest compared to the competing models. The EfficientNetB0 model [23] ranks second with an accuracy of 99.00 and Deep Neural Network (DNN) model [23] is third with an accuracy of 98.00. The IMCBL framework [21] has reached 97.85% and the less powerful convolutional architectures like ConvNet, ResNeXt-50, and MobileNetV2 have reached accuracy in the 93.7% -94.2. These findings evidently prove that the Res-Former model is superior in malware image classification. The impressive improvement in performance is explained by the hybrid implementation of the ResNet backbone (to extract local features in hierarchies) and the Transformer branch (to model long-range dependencies) that, combined with the ensemble learning scheme,

improves the generalization and resistance. Overall, the proposed solution achieves an accuracy of 99.46% and exhibits performance that is both superior and more stable than recent state-of-the-art methods. This confirms that the proposed architecture and preprocessing approach are highly effective in improving malware detection accuracy. The findings of the classification of the malware dataset are presented It should be noted that the presented results of the proposed method are obtained by averaging over 50 independent executions of the program. As can be seen in the table, the average accuracy of the proposed method is 99.46 percent, which is the highest accuracy among the compared methods. Also, the standard deviation of the proposed method for these 50 iterations is 0.67 percent, which shows the stability of the proposed method. The results obtained from 50 independent executions of the proposed method show an average Accuracy of 99.46% with a standard deviation of 0.67%, indicating a high degree of statistical stability. Achieving such a high mean accuracy with a relatively low standard deviation demonstrates that the model maintains consistent and predictable performance across repeated executions rather than relying on the outcome of a single run. This stability is an important consideration for the scalability of malware detection systems, as models deployed in larger-scale or heterogeneous operational environments should maintain reliable performance under varying data volumes, distributions, and computational conditions. Therefore, the results across 50 independent executions indicate that the proposed method exhibits strong stability and robustness to random variations, suggesting its potential for deployment in larger and more complex malware detection environments while maintaining reliable detection performance.

TABLE III.
ACCURACY COMPARISON BETWEEN THE PROPOSED METHOD AND RECENT STUDIES

Author	Method	Accuracy
Vasan et al. [21]	IMCBL ¹	97.85
Liu et al. [22]	ConvNet	94.22
Liu et al. [22]	ResNeXt-50	94.03
Liu et al. [22]	MobileNetV2	93.71

Habib et al. [23]	DNN	98.00
Habib et al. [23]	EfficientNetB0	99.00
-	Proposed	99.46

Fig. 9 also compares the performance of the proposed model with the recent approaches in terms of Precision, Recall, and F-score. As it can be seen, the proposed Res-Former is superior to all the other approaches in all the evaluation measures. In particular, the Precision of the proposed model is 99.46, which means that the model is highly effective to detect malware samples correctly with the low amount of false positive rate. In the same way, recall value of 99.46% proves the ability of the model to retrieve virtually all instances of malware, reducing false negatives. The F-score, a harmonic mean of Precision and Recall also reaches 99.46, which proves the balanced and trustworthy classification efficiency of the model. The compared methods have the closest performance and IMCBL with Precision = 97.97, Recall = 97.85, and F-score = 97.87. Nonetheless, it not yet meets the suggested approach in the three metrics. ConvNet, ResNeXt-50, MobileNetV2, DNN and EfficientNetB0 are other models that have lower scores on all the indicators. All in all, these findings emphasize the strength, accuracy, and high-quality discriminative ability of the proposed Res-Former ensemble model that is quite effective in capturing both local structural trends and global contextual interdependence of malware images. The effectiveness of the suggested hybrid architecture in the process of obtaining accurate and reliable detection of malware is confirmed by the balanced high values of Precision, Recall, and F-score.

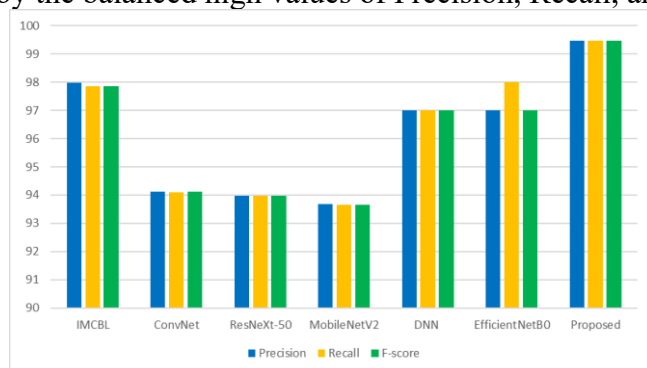


Fig. 9. Comparison of the proposed method and recent studies with respect to Precision, Recall, and F-score.

Discussion

The results reported in Table III and Fig. 9 demonstrate the competitive and superior performance of the proposed method in malware detection compared with the considered baseline approaches. As shown in Table III, the proposed method achieves an Accuracy of 99.46%, which is the highest value among the compared methods reported in the literature. Also, the results presented in Figure 9 compare the values of the three key metrics Precision, Recall, and F score of the proposed method, all of which have values greater than 99 percent, with other methods. The proposed method consistently achieves the highest values across all three metrics. Such simultaneous performance across complementary evaluation measures indicates a high capability for correctly identifying malware samples while effectively controlling both false-positive and false-negative errors. Overall, the results confirm the effectiveness and reliability of the proposed method for accurate malware detection under the evaluated experimental conditions.

Ablation Study

The results of the ablation study are presented with the aim of evaluating the individual contribution of each component within the proposed method, in table IV. This analysis primarily focuses on assessing the impact of different feature-extraction architectures as well as the effectiveness of Ensemble Learning as the final classifier. The results indicate that integrating deep architectures with Ensemble Learning leads to a substantial improvement in classification accuracy. In the ResNet50-Ensemble Learning configuration, the aggregation of multiple models reduces error variance and enhances decision stability, yielding an accuracy of approximately 97.80%. This value suggests that although convolutional networks perform effectively in extracting local features, they exhibit limitations in modeling long-range dependencies—limitations that are partially mitigated through Ensemble Learning. For the Transformer-Ensemble Learning architecture, the use of multi-head attention and the ability to capture global relationships among features further improves performance, resulting in an accuracy close to 98.90%.

This outcome demonstrates that attention-based architectures, when combined with Ensemble Learning, possess a stronger capability to discriminate complex patterns.

Finally, the ResFormer-Ensemble Learning architecture—which integrates the strengths of ResNet and Transformer—achieves the highest accuracy at 99.46%. This finding highlights that combining local and global feature-representation mechanisms, together with Ensemble Learning, can significantly enhance model effectiveness and make the approach suitable for high-precision and safety-critical applications.

Overall, the results in Table IV emphasize that each component of the proposed method contributes meaningfully to performance improvement, and that the use of Ensemble Learning plays a pivotal role in achieving the final accuracy.

TABLE IV.
ABLATION STUDY

Method	Accuracy
ResNet50-Ensemble Learning	97.80
Transformer-Ensemble Learning	98.90
ResFormer- Ensemble Learning	99.46

Conclusion

Res-Former, a state-of-the-art hybrid deep learning model, is the suggested research in this paper, which combines the ResNet-50 convolutional backbone and a Transformer-based attention module to improve the performance of malware detectors. The suggested model is a successful integration of the local feature extraction and hierarchical representation learning abilities of convolutional networks and the global dependency modelling and contextual reasoning advantages of Transformers. In order to enhance stability and generalization, an ensemble learning classifier using bagging and majority voting was introduced in the decision process. This approach minimizes the prediction variance, makes it more robust to noisy and imbalanced data, and ensures more consistent results classification. The experiments using the Maling dataset have shown that the proposed Res-Former ensemble had impressive results because it achieved an accuracy rate of 99.46, which is better than more sophisticated deep learning models, such as EfficientNetB0, DNN, and IMCBL. It was also a sign of a high discriminative capability and reliability because the model has a high precision, recall, and F1-scores. Overall, the findings reveal that an ensemble-based system with a mixture of both convolutional and Transformer-based representations is a promising and scalable system that can detect malware. Future directions will consider lightweight Res-Former to be used in real-time, and multi-modal malware identification to further increase its use in cybersecurity environments.

References

- [1] Dutta, N., Jadav, N., Tanwar, S., Sarma, H. K. D., & Pricop, E. (2022). Introduction to malware analysis. *Cyber Security: Issues and Current Trends*, 129–141.
- [2] Rodríguez, R. J., Ugarte-Pedrero, X., & Tapiador, J. (2022). Introduction to the special issue on challenges and trends in malware analysis. *Digital Threats: Research and Practice*, 3(2), 1–2.
- [3] Gorment, N. Z., Selamat, A., Cheng, L. K., & Krejcar, O. (2023). Machine learning algorithm for malware detection: Taxonomy, current challenges, and future directions. *IEEE Access*, 11, 141045–141089.
- [4] Tayyab, U. E. H., Khan, F. B., Durad, M. H., Khan, A., & Lee, Y. S. (2022). A survey of the recent trends in deep learning-based malware detection. *Journal of Cybersecurity and Privacy*, 2(4), 800–829.
- [5] Bensaoud, A., Kalita, J., & Bensaoud, M. (2024). A survey of malware detection using deep learning. *Machine Learning with Applications*, 16, 100546.
- [6] Alshomrani, M., Albeshri, A., Alturki, B., Alallah, F. S., & Alsulami, A. A. (2024). Survey of transformer-based malicious software detection systems. *Electronics*, 13(23), 4677.
- [7] Nair, S. J., & Syam, S. R. (2024, June). Comparing transformers and CNN approaches for malware detection: A comprehensive analysis. In *Proceedings of the 15th International Conference on Computing Communication and Networking Technologies (ICCCNT)* (pp. 1–6). IEEE.
- [8] Khan, I. U., Zeb, A., Rahman, T., Khan, F. M., Haider, Z. A., & Bilal, H. (2025). ViTDroid and hybrid models for effective Android and IoT malware detection. *ICCK Transactions on Advanced Computing and Systems*, 1(1), 32–47.
- [9] Qian, L., & Cong, L. (2024). Channel features and API frequency-based transformer model for malware identification. *Sensors*, 24(2), 580.
- [10] Çıplak, Z., Yıldız, K., & Altınkaya, Ş. (2025). FEDetect: A federated learning-based malware detection and classification using deep neural network algorithms. *Arabian Journal for Science and Engineering*, 1–28.

- [11] Alshomrani, M., Albeshri, A., Alsulami, A. A., & Alturki, B. (2025). An explainable hybrid CNN–Transformer architecture for visual malware classification. *Sensors*, 25(15), 4581.
- [12] Liu, Y., Wang, R., & Zhang, Q. (2024). Enhanced image-based malware classification using transformer-based convolutional neural networks (CNNs). *Electronics*, 13(20), 4081.
- [13] Ahmed, H., & Zhao, L. (2024). Malware classification based on image segmentation. *arXiv preprint arXiv:2406.03831*.
- [14] Kim, J., & Patel, R. (2023). High-resolution image-based malware classification using multiple instance learning. *arXiv preprint arXiv:2312.09425*.
- [15] Tursunov, B., & Ibragimov, N. (2024). Enhanced image-based malware multiclass classification method with the ensemble model and SVM. *Open Information Science*, 8(1), 1–12.
- [16] Chen, X., Li, K., & Xu, Z. (2024). Deep multi-task learning for malware image classification. *arXiv preprint arXiv:2405.05906*.
- [17] Dutta, S., & Banerjee, A. (2024). Accelerating malware classification: A vision transformer solution (LeViT-MC). *arXiv preprint arXiv:2409.19461*.
- [18] Raza, M., & Ullah, S. (2024). Malware classification using few-shot learning approach. *Information*, 15(11), 722.
- [19] de Souza, L., & Carvalho, T. (2024). Deep learning applied to imbalanced malware datasets classification. *Journal of Internet Services and Applications*, 15(1), 1–17.
- [20] Saleh, M. A., & Hamza, N. H. (2025). Enhanced malware detection for IoT networks utilizing a 2D-CNN with data augmentation on the Maling dataset. *Journal of Quantitative and Computational Science Methods*, 6(2), 1973.
- [21] Vasan, D., Hammoudeh, M., & Alazab, M. (2024). Broad learning: A GPU-free image-based malware classification. *Applied Soft Computing*, 154, 111401.
- [22] Liu, Y., et al. (2024). Efficient and generalized image-based CNN algorithm for multi-class malware detection. *IEEE Access*.
- [23] Habib, F., et al. (2024). Deep neural networks for enhanced security: Detecting metamorphic malware in IoT devices. *IEEE Access*, 12, 48570–48582.

- [24] Obaid, Z. H., Mirzaei, B., & Darroudi, A. (2024). An efficient automatic modulation recognition using time–frequency information based on hybrid deep learning and bagging approach. *Knowledge and Information Systems*, 66(4), 2607–2624.
- [25] Malimg Malware Dataset. (2025). *Kaggle*.
<https://www.kaggle.com/datasets/manmandes/malimg>