

A-HRAB: Adaptive Hybrid Residual Attention for CNN-based Malware Family Classification

Omar Al-Harbi * and Areej AlShorman 

Department of Cyber Security, Faculty of Prince Al-Hussein Bin Abdallah II for IT,
Al al-Bayt University, Mafrq, Jordan

Email: alharbi-omar@aabu.edi.jo (O.A.-H.); areej2017shorman@aabu.edu.jo (A.A.)

*Corresponding author

Abstract—Reliable malware family classification remains a major cybersecurity challenge, particularly for malware families that exhibit high intra-class variability and strong visual similarity to other families. Although Convolutional Neural Networks (CNNs) based on image representations of malware have demonstrated strong performance, they often struggle to capture complex feature dependencies between overlapping and visually similar classes. This paper proposes the Adaptive Hybrid Residual Attention Block (A-HRAB), an attention module integrated into residual convolutional blocks to enhance feature representation. The proposed module computes channel and spatial attention maps and combines them through a learnable adaptive fusion mechanism that dynamically balances their relative contributions based on the input features. Experiments conducted on the Maling and MaleVis datasets demonstrate the effectiveness of the proposed approach. On the Maling dataset, the proposed CNN with A-HRAB achieves 99.4% accuracy and a Macro-F1 score of 98.6%, whereas on the MaleVis dataset, it attains 96.1% accuracy and a Macro-F1 score of 96.0%. Furthermore, the proposed model consistently outperforms the baseline CNN and conventional attention modules, including Squeeze-and-Excitation (SE) and Convolutional Block Attention Module (CBAM). A class-wise analysis further confirms substantial performance improvements for challenging malware families characterized by high intra-class variability and strong inter-class similarity.

Keywords—malware classification, image-based malware analysis, Convolutional Neural Network (CNN), attention mechanisms, deep learning for cybersecurity

I. INTRODUCTION

Malware remains a persistent and rapidly evolving threat, encompassing viruses, worms, trojans, ransomware, spyware, and Advanced Persistent Threats (APTs) that target governments, organizations, and individuals, causing significant financial losses, data breaches, and operational disruptions [1]. The increasing use of obfuscation techniques, polymorphism, and code packing has further complicated detection, enabling malware to evade traditional defenses. With hundreds of thousands of new malware samples reported daily by the

AV-TEST Institute [2] and the rapid proliferation of malware families through automated generation frameworks, there is a critical need for adaptive and intelligent detection mechanisms.

Traditional malware detection methods rely heavily on signature-based approaches that match files against predefined patterns or hashes. While effective against known threats, these techniques struggle to detect novel or obfuscated variants [3]. Polymorphic and metamorphic malware can dynamically alter their structure while preserving their functionality, rendering signature-based systems ineffective and requiring continuous expert updates [4]. Behavioral and heuristic approaches partially address these limitations but often incur higher computational overhead and produce higher false-positive rates [5]. To overcome these challenges, Machine Learning (ML) techniques have been introduced to automatically learn patterns from data, reducing reliance on handcrafted features. Early approaches utilized static features such as opcode distributions, byte n -grams, and control flow graphs, combined with classifiers such as Support Vector Machines (SVM), Random Forests (RF), and k -Nearest Neighbors (KNN) [6, 7]. Although these methods improved detection performance, their dependence on manually engineered features limits their ability to capture complex structural and visual patterns in malware binaries [8].

The introduction of image-based malware analysis in [9] made it possible to transform malware binaries into grayscale images, allowing visual patterns to be learned without disassembly or execution. Building on this representation, Convolutional Neural Networks (CNNs) have become the dominant approach due to their ability to learn hierarchical and spatial features. Architectures such as AlexNet [10], VGGNet [11], and ResNet [12] have demonstrated strong performance in capturing the textural and structural characteristics of malware images. The Squeeze-and-Excitation (SE) block [13] models channel-wise dependencies through global context aggregation but neglects spatial information, thereby overlooking critical structural cues such as texture distribution [14]. In contrast, the Bottleneck Attention

Module (BAM) [15] introduced parallel channel and spatial attention branches to refine intermediate feature representations. However, BAM uses fixed additive fusion, limiting its adaptation to varying channel and spatial importance across inputs. The Convolutional Block Attention Module (CBAM) [16] further integrates channel and spatial attention sequentially; however, its reliance on global average and max pooling for channel attention may oversimplify feature aggregation, leading to the loss of fine-grained patterns essential for distinguishing obfuscated malware [17].

Despite these advances, existing CNN-based and attention-augmented approaches still face key challenges. First, they often struggle to effectively capture complex feature relationships among malware families characterized by high intra-class variability and inter-class similarity [9]. Second, many methods lack robustness under class imbalance, leading to degraded performance on difficult or overlapping classes [18, 19].

To address these limitations, this paper proposes an Adaptive Hybrid Residual Attention Block (A-HRAB) for malware image classification. The proposed module is integrated into a CNN architecture and employs parallel channel and spatial attention branches, following the general paradigm adopted in existing attention mechanisms such as BAM [15], and introduces a learnable adaptive scalar fusion mechanism that dynamically balances their contributions based on the input features. The module is embedded within a residual learning framework to preserve gradient flow and facilitate stable deep feature learning. In addition, a hybrid pooling strategy that combines global average pooling and global max pooling is employed to further enhance feature representation, while the training process is strengthened through class weighting, label smoothing, and cosine decay learning rate scheduling.

The main contributions of this work are as follows: (1) An Adaptive Hybrid Residual Attention Block (A-HRAB) that introduces a learnable adaptive scalar fusion mechanism to dynamically balance parallel channel and spatial attention for input-dependent feature refinement. (2) A CNN-based architecture that enhances discriminative feature learning for malware image classification, particularly for complex and overlapping classes. (3) A robust training strategy that incorporates class weighting, label smoothing, and adaptive learning rate scheduling to address class imbalance and improve generalization.

The remainder of this paper is organized as follows. Section II reviews related work on traditional machine learning, deep learning-based malware classification, and attention mechanisms. Section III presents the proposed methodology, including the A-HRAB framework, data preprocessing, the adaptive hybrid residual attention module, and the classification pipeline. Section IV describes the experimental setup, datasets, evaluation metrics, and experimental protocol. Section V presents and discusses the results, including overall performance, class-wise analysis, and comparisons with state-of-the-art

methods. Finally, Section VI concludes the paper and outlines future research directions.

II. RELATED WORK

Early malware detection approaches relied on traditional machine learning models built on handcrafted features extracted from static or dynamic analysis. For instance, Madamidola *et al.* [20] proposed a lightweight and interpretable RF model using memory-based features from the CIC-MalMem-2022 dataset, achieving an accuracy of 99.84% with low false positive and false negative rates. Similarly, Shatnawi *et al.* [21] utilized permission and API call features for Android malware detection, employing classifiers such as SVM, K-NN, and Naïve Bayes, with the SVM classifier achieving the best performance, attaining an accuracy of 94.36% and an F1-score of 88.75%. Despite their effectiveness, traditional machine learning methods are inherently limited by their reliance on handcrafted features, which may fail to capture complex patterns in malware data. Additionally, these methods often face scalability challenges, increased computational cost in ensemble settings, and susceptibility to overfitting when dealing with high-dimensional feature spaces [22–24].

Recent studies have demonstrated the effectiveness of deep convolutional architectures for malware image classification. For example, El-Shafai *et al.* [25] showed that fine-tuned CNN models, including VGG16 and ResNet-50, achieved classification accuracy exceeding 98% on multi-class malware datasets. Palma Salas and de Geus [26] further improved performance using a fine-tuned MobileNet model with transfer learning, achieving an accuracy of 99.08% on the Malimg dataset. Similarly, Al-Masri *et al.* [27] proposed a Dual Convolutional Malware Network (DCMN) that combines a custom CNN with ResNet-50 to leverage complementary feature representations, achieving an accuracy of 99.89% and demonstrating the effectiveness of multi-branch feature learning. More recently, hybrid CNN–Transformer architectures have been explored to capture long-range contextual dependencies and enhance feature representation. For instance, Ashawa *et al.* [28] proposed an image-based malware classification framework that compares ResNet-152 and Vision Transformer (ViT) architectures, achieving a peak accuracy of 99.62% on the MalNet dataset.

To enhance feature representation, attention mechanisms have been increasingly integrated into CNN-based malware classifiers. Awan *et al.* [29] introduced a VGG19-based architecture augmented with spatial convolutional attention, achieving more than 97% for all accuracy-related evaluation metrics after applying class balancing, demonstrating that lightweight attention can effectively highlight informative regions in malware images while maintaining computational efficiency. Similarly, Fan *et al.* [17] proposed GCSA-ResNet, which integrates a Global Channel-Spatial Attention (GCSA) module into ResNet-50 to jointly model channel and spatial dependencies through channel shuffling and attention. The model achieved more than 98.5% accuracy

on the Malimg and Microsoft BIG 2015 datasets while reducing false positives by up to 50%. Zhang *et al.* [30] introduced IMCMK, a CNN architecture that employs multi-scale convolution kernels alongside an enhanced SE block to improve feature extraction and channel-wise representation. The model achieved an accuracy of 99.25% in malware family classification and demonstrated improved robustness against malware variants and obfuscated malware.

Nevertheless, existing attention-based approaches exhibit several limitations. Many rely on either channel-only or spatial-only attention, and even when both mechanisms are incorporated, they are typically applied sequentially, which may limit their ability to capture complementary feature dependencies effectively. In addition, most methods employ static fusion strategies that do not adapt to the characteristics of the input, reducing their effectiveness in complex scenarios. These limitations are particularly evident in malware datasets characterized by high intra-class variability, high inter-class similarity

and class imbalance, where robust and input-adaptive feature representation is critical for accurate classification.

Motivated by these challenges, this work introduces A-HRAB, which integrates parallel channel and spatial attention with a learnable adaptive fusion mechanism within a residual framework, enabling more flexible and discriminative feature learning for malware classification.

III. METHODS

Fig. 1 illustrates the proposed A-HRAB-based framework for malware classification. The model takes a grayscale malware image as input and processes it through a series of stages, including preprocessing, initial feature extraction, attention-enhanced residual learning, and classification. The architecture integrates adaptive hybrid attention mechanisms within residual blocks to enhance feature representation. Finally, global feature aggregation and SoftMax classification are applied.

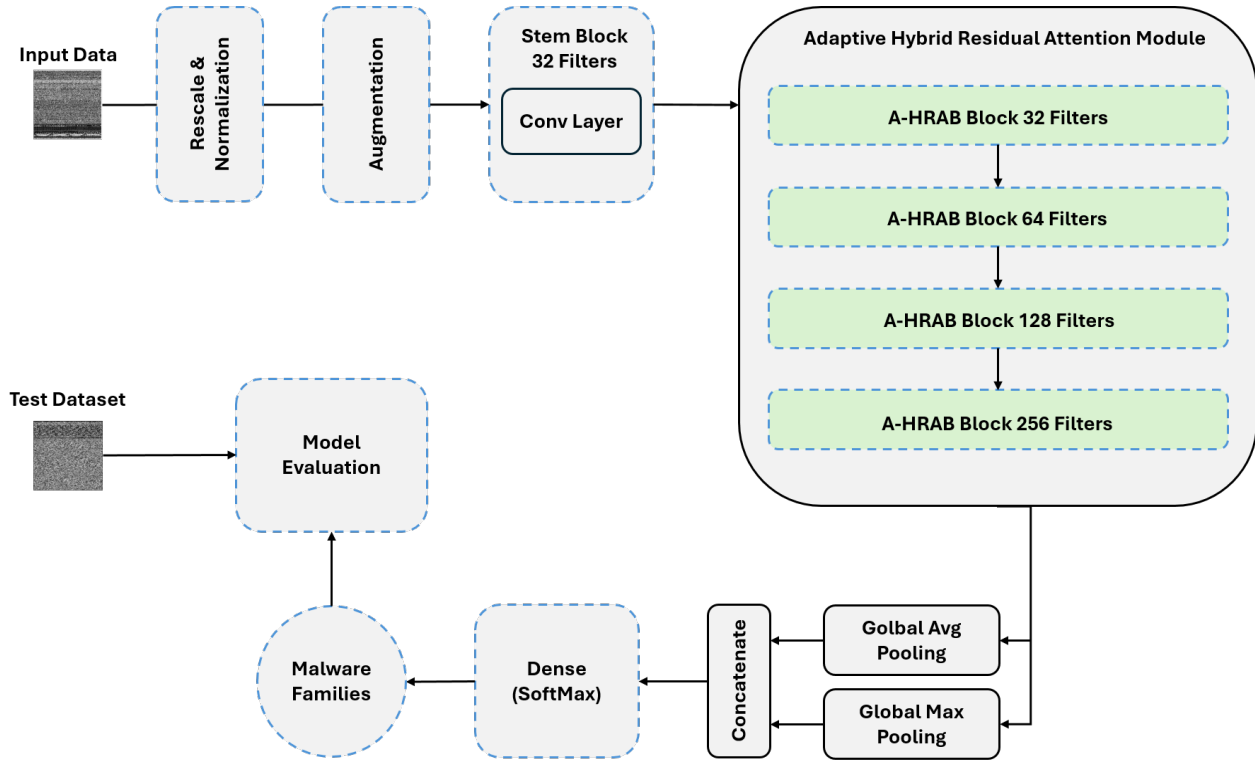


Fig. 1. Architecture of the proposed A-HRAB-based malware classification framework.

A. Data Preprocessing

Each input malware image $X \in \mathbb{R}^{H \times W \times 1}$ undergoes preprocessing to ensure a consistent input distribution and to improve generalization performance. The preprocessing pipeline includes resizing, normalization, and light data augmentation.

1) Image resizing and normalization

Malware images typically exhibit varying spatial dimensions due to differences in binary file sizes. To ensure compatibility with the convolutional neural network, facilitate efficient batch processing, and provide

consistent inputs across all evaluated models, all images are resized to a fixed resolution of 224×224 pixels following the methodology adopted in [16]. Although fixed-size resizing may introduce interpolation artifacts and distort fine-grained spatial structures, it provides a standardized input representation that facilitates fair and consistent model evaluation. Subsequently, pixel intensity values, originally in the range $[0, 255]$, are scaled to the normalized range $[0, 1]$. This normalization improves numerical stability during training and accelerates model convergence. An example of a resized and normalized malware image is illustrated in Fig. 2.

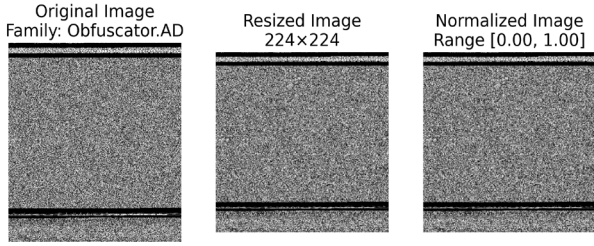


Fig. 2. Example of a resized and normalized Obfuscator.AD malware image.

2) Data augmentation

To improve model generalization and reduce overfitting, light data augmentation is applied during the training phase only. Unlike natural images, malware images are visual representations of binary byte sequences; therefore, aggressive geometric transformations may distort their structural characteristics and produce unrealistic samples. Consequently, the augmentation strategy was intentionally conservative, consisting of random horizontal flipping, slight rotations ($\pm 2^\circ$), and small zoom variations ($\pm 5\%$). These transformations are selected to introduce limited visual variability while preserving the overall structural patterns of the malware images. An example of an augmented malware sample is presented in Fig. 3.

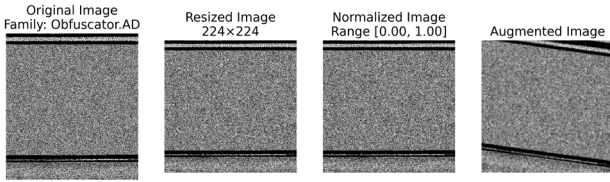


Fig. 3. Example of an augmented Obfuscator.AD malware image.

B. Stem Convolution

Following preprocessing, the normalized input image is passed through a stem convolutional layer to extract low-level spatial and texture features. Let $F \in \mathbb{R}^{H \times W \times C}$ denote a generic feature map with spatial dimensions H, W and C channels:

$$F = \phi(\text{BN}(\text{Conv}_{3 \times 3}^{s=2}(X))) \quad (1)$$

where $\text{Conv}_{3 \times 3}^{s=2}$ denotes a convolutional operation with a 3×3 kernel and stride 2, $\text{BN}(\cdot)$ represents batch normalization, and $\phi(\cdot)$ corresponds to the Rectified Linear Unit (ReLU) activation function.

The use of stride 2 enables early spatial downsampling, reducing computational complexity while preserving essential structural information. The resulting feature map F_0 is then forwarded to the subsequent attention-enhanced residual blocks.

C. Adaptive Hybrid Residual Attention Module

As illustrated in Fig. 1, the proposed framework employs four stacked A-HRAB blocks with progressively increasing numbers of filters (32, 64, 128, and 256) to learn hierarchical and discriminative representations. As shown

in Fig. 4, each block processes the input through two consecutive 3×3 convolutional layers with a stride of 1; the first is followed by batch normalization and activation, while the second is followed by batch normalization and a hybrid attention function $A(\cdot)$, which integrates channel and spatial attention via an adaptive fusion strategy:

$$\hat{F} = \text{BN}(\text{Conv}_{3 \times 3}(\phi(\text{BN}(\text{Conv}_{3 \times 3}(F)))))) \quad (2)$$

To preserve original information and facilitate stable gradient propagation, a residual (skip) connection directly links the input to the attention-refined features via element-wise addition. Finally, the ReLU activation is applied after this fusion, enabling non-linearity in the combined representation. This design facilitates effective feature refinement by capturing both low-level and high-level patterns while adaptively emphasizing the most informative channels and spatial regions.

$$F_{\text{out}} = \phi(A(\hat{F}) + F_{\text{skip}}) \quad (3)$$

Here, F_{skip} denotes either the identity mapping or a 1×1 convolutional projection to align dimensions when necessary.

The attention function $A(\cdot)$ is a critical component of the A-HRAB, as it enhances feature representation by adaptively emphasizing informative channels and salient spatial regions. The following subsections provide a detailed description of its internal components and fusion strategy.

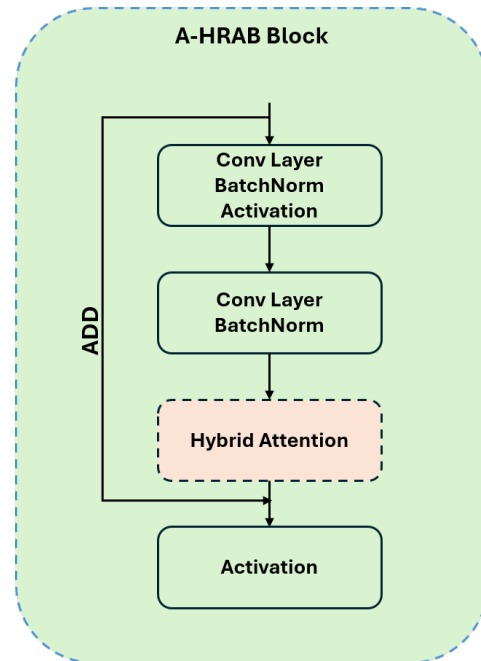


Fig. 4. Architecture of A-HRAB block.

1) Hybrid attention

The proposed A-HRAB builds upon existing attention mechanisms, including CBAM and BAM, while introducing a more flexible attention fusion strategy. Unlike CBAM, which applies channel and spatial attention

sequentially, and BAM, which employs a fixed parallel interaction, A-HRAB computes channel and spatial attention in parallel and integrates them through a learnable adaptive fusion mechanism. Furthermore, the attention module is embedded within a residual convolutional block, allowing attention refinement and residual learning to be optimized jointly. As illustrated in Fig. 5, the proposed hybrid attention layer computes channel and spatial attention in parallel and fuses them through an adaptive, learnable strategy, enabling complementary and input-dependent feature refinement. Moreover, the attention layer includes an internal residual connection, allowing the original features to be preserved alongside the refined representations.

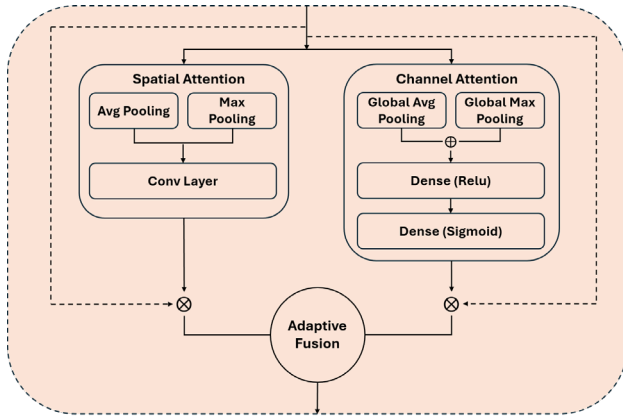


Fig. 5. Architecture of proposed hybrid attention layer.

The attention function $A(\cdot)$ therefore captures both channel-wise and spatial dependencies more effectively, leading to stronger discriminative feature learning.

a) Channel attention

To strengthen the representational power of the network, we design a channel attention mechanism that exploits multiple global descriptors to capture richer feature dependencies. The proposed approach employs Global Average Pooling (GAP) and Global Max Pooling (GMP) to encode complementary statistical information from the feature map. In contrast to CBAM's late fusion strategy, where outputs of parallel MLP branches are summed, A-HRAB aggregates these descriptors at an earlier stage by summing them prior to transformation. This early fusion enables the model to jointly capture global distribution and salient responses in a unified representation. For an intermediate feature tensor $\hat{F} \in \mathbb{R}^{H \times W \times C}$, the aggregated channel descriptor is computed as:

$$z = \text{GAP}(\hat{F}) + \text{GMP}(\hat{F}), z \in \mathbb{R}^{1 \times 1 \times C} \quad (4)$$

The aggregated descriptor z is then passed through a two-layer fully connected network that captures channel-wise dependencies. The first layer performs dimensionality reduction controlled by a reduction ratio r , followed by a nonlinear activation using ReLU, while the

second layer restores the original channel dimension. The resulting channel attention map is defined as:

$$M_c = \sigma(W_2 \cdot \delta(W_1 \cdot z)) \quad (5)$$

where $\delta(\cdot)$ denotes the ReLU activation, $\sigma(\cdot)$ is the sigmoid function, $W_1 \in \mathbb{R}^{\frac{C}{r} \times C}$ and $W_2 \in \mathbb{R}^{C \times \frac{C}{r}}$. The reduction ratio r controls the trade-off between model complexity and representational capacity by compressing channel information prior to reconstruction; in this work, $r = 8$.

Finally, the computed attention weights $M_c \in \mathbb{R}^{1 \times 1 \times C}$ are applied to the input feature map through element-wise multiplication, yielding an enhanced feature representation:

$$F_{\text{channel}} = \hat{F} \odot M_c \quad (6)$$

This mechanism allows the network to prioritize more informative channels while attenuating less relevant ones, thereby improving feature discrimination and overall model performance.

b) Spatial attention

A spatial attention mechanism is employed to identify the most informative regions within the feature map. This component emphasizes the locations of informative features by leveraging channel-wise aggregated information. Given the intermediate feature map $\hat{F}$, two complementary spatial descriptors are first generated using average and max pooling operations across the channel dimension:

$$F_{\text{avg}} = \text{Avg}_c(\hat{F}), F_{\text{max}} = \text{Max}_c(\hat{F}) \quad (7)$$

where Avg_c and Max_c denote average and max pooling across the channel dimension, respectively. These descriptors summarize global contextual information and the most salient responses at each spatial location, respectively. The resulting feature maps are then concatenated along the channel axis to form a unified representation:

$$F_{\text{spatial}} = [F_{\text{avg}}, F_{\text{max}}] \quad (8)$$

To capture broader spatial dependencies, a convolutional layer with a relatively large receptive field is employed. Following the design of CBAM, a 7×7 convolutional layer is applied to generate the spatial attention map:

$$M_s = \sigma(\text{Conv}_{7 \times 7}(F_{\text{spatial}})) \quad (9)$$

where $\sigma(\cdot)$ denotes the sigmoid activation function. The use of a larger kernel size enables the model to incorporate broader contextual information, which is particularly beneficial for capturing structural patterns in malware images.

Finally, the spatial attention map $M_s \in \mathbb{R}^{H \times W \times 1}$ is applied to the input feature map through element-wise multiplication:

$$F_{\text{spatial}}^* = \hat{F} \odot M_s \quad (10)$$

This operation enhances relevant spatial regions while suppressing less informative areas, thereby contributing to more discriminative feature learning.

2) Adaptive attention fusion

To integrate channel-wise and spatial representations, the proposed adaptive fusion mechanism learns a scalar gating coefficient $\alpha \in (0,1)$ from the global context of the feature map, enabling dynamic reweighting between channel-attended and spatial-attended features.

Specifically, the intermediate feature map $\hat{F}$ is first compressed using global average pooling, followed by a fully connected layer and a sigmoid activation to produce the adaptive coefficient:

$$\alpha = \sigma(W_f \cdot \text{GAP}(\hat{F})) \quad (11)$$

where $W_f \in \mathbb{R}^{1 \times C}$ represents a learnable parameter vector. The scalar α controls the balance between channel-focused and spatial-focused feature refinement.

Using this coefficient, the final attention-enhanced feature representation is computed as a weighted combination of channel-attended and spatial-attended features:

$$F_a = \alpha(\hat{F} \odot M_c) + (1 - \alpha)(\hat{F} \odot M_s) \quad (12)$$

This formulation enables the network to adaptively balance channel-wise and spatial information based on the characteristics of the input. By allowing dynamic interaction between the two attention mechanisms, the fusion process enhances flexibility and improves the overall discriminative capability of the model.

D. Feature Aggregation and Classification

Following feature extraction through the stacked A-HRAB blocks, the resulting feature map is transformed into a compact representation using a global feature aggregation strategy. To capture complementary global characteristics, both GAP and GMP are applied to the final feature tensor F :

$$f = [\text{GAP}(F), \text{GMP}(F)] \quad (13)$$

This dual pooling mechanism enables the model to encode both the overall distribution of features and the most prominent activations, thereby providing a more informative global descriptor.

The aggregated feature vector is then regularized using dropout to mitigate overfitting. Finally, classification is performed using a fully connected layer followed by a SoftMax activation function:

$$y = \text{SoftMax}(W_c \cdot f) \quad (14)$$

where W_c denotes the learnable classification weights, and y represents the predicted probability distribution over the malware classes.

IV. EXPERIMENTS

A. Datasets

The experiments were conducted using the Maling [9] and MaleVis [31] datasets, which are two widely adopted benchmark datasets for image-based malware classification. These datasets contain diverse malware families and have been extensively used to evaluate the effectiveness of malware classification models. The following subsections describe the characteristics of each dataset.

1) The maling dataset

The Maling dataset [9] comprises grayscale images of varying sizes generated from malware binaries and contains 9,339 samples representing 25 malware families, including worms, trojans, downloaders, dialers, password stealers, backdoors, and rogue software. The dataset is inherently imbalanced, with certain families (e.g., Allaple.A) being significantly overrepresented. The distribution of the malware families is summarized in Table I. To ensure reproducibility, the experiments were conducted using the Maling dataset obtained from a publicly available Kaggle repository that provides a predefined train/validation/test partition. This split was adopted without modification and consists of 7,459 training samples, 923 validation samples, and 957 test samples. All images were resized to 224×224 pixels and loaded in grayscale format.

TABLE I. DISTRIBUTION OF MALWARE FAMILIES IN THE MALING DATASET

No.	Malware Category	Family Name	Samples
1	Dialer	Adialer.C	122
2		Dialplatform.B	177
3		Instantaccess	431
4	Backdoor	Agent.FYI	116
5		Rbot!gen	158
6		Allaple.A	2949
7	Worm	Allaple.L	1591
8		Autorun.K	106
9		VB.AT	408
10		Yuner.A	800
11	Trojan	Alueron.gen!J	198
12		C2LOP.P	200
13		C2LOP.gen!g	146
14		Malex.gen!J	136
15		Skintrim.N	80
16	Password Stealer (PWS)	Lolyda.AA1	213
17		Lolyda.AA2	184
18		Lolyda.AA3	123
19		Lolyda.AT	159
20	Downloader	Obfuscator.AD	142
21		Swizzor.gen!E	128
22		Swizzor.gen!I	132
23		Wintrim.BX	97
24		Dontovo.A	162
25	Rogue	Fakerean	381
Total			9339

2) The MaleVis dataset

The MaleVis dataset [31] is a publicly available benchmark for malware image classification, comprising images generated from Portable Executable (PE) files belonging to 25 malware families and an additional “Other” class representing benign software. To focus on malware family classification and maintain consistency with the Maling dataset, the “Other” class was excluded from this study. Consequently, the experiments were conducted using the remaining 25 malware families, resulting in 8,750 training images and 3,644 validation images. The malware families include Adposhel, Agent, Allaple, Amonetize, Androm, Autorun, BrowseFox, Dinwod, Elex, Expiro, Fasong, HackKMS, Hlux, Injector, InstallCore, MultiPlug, Neoreklami, Neshta, Regrun, Sality, Snarasite, Stantinko, VBA, VBKrypt, and Vilsel. Table II shows that the MaleVis dataset is relatively balanced across all malware families, with approximately uniform sample sizes per class, thereby minimizing the effects of class imbalance during training. All images were resized to 224×224 pixels and converted to grayscale before training to maintain consistency with the proposed framework and the preprocessing applied to the Maling dataset.

TABLE II. DISTRIBUTION OF MALWARE FAMILIES IN THE MALEVIS DATASET

No.	Family Name	Samples
1	Adposhel	494
2	Agent	470
3	Allaple	478
4	Amonetize	497
5	Androm	500
6	Autorun	496
7	BrowseFox	493
8	Dinwod	499
9	Elex	500
10	Expiro	501
11	Fasong	501
12	HackKMS	499
13	Hlux	500
14	Injector	495
15	InstallCore	500
16	MultiPlug	499
17	Neoreklami	500
18	Neshta	497
19	Regrun	485
20	Sality	499
21	Snarasite	500
22	Stantinko	500
23	VBA	500
24	VBKrypt	496
25	Vilsel	496
Total		12394

B. Experimental Setup and Model Configuration

All experiments were implemented using TensorFlow/Keras and executed on Google Colab with an NVIDIA Tesla T4 GPU. The proposed architecture was trained for 50 epochs with a batch size of 32. Optimization was performed using the Adam optimizer with a learning rate (LR) scheduled according to a cosine decay schedule over the entire training duration. Gradient clipping was

applied to stabilize training. To evaluate the stability of the proposed model, each experiment was independently repeated using five different random seeds (42, 123, 2024, 999, and 7).

To improve generalization and mitigate overconfidence, label smoothing was incorporated into the categorical cross-entropy loss. Class imbalance was addressed using class weights, computed from the distribution of the training data. Regularization was further enhanced by applying dropout in the final classification stage. The training settings and hyperparameter configuration are summarized in Table III.

TABLE III. HYPERPARAMETER CONFIGURATION AND TRAINING SETTINGS

Category	Hyperparameter	Value
Training	Batch Size	32
	Epochs	50
	Shuffle Buffer	1000
Optimization	Optimizer	Adam
	Gradient Clipping	Clipnorm = 1.0
	Loss Function	Categorical Cross-Entropy
	Label Smoothing	0.1
	LR Schedule	Cosine Decay
	initial_learning_rate	1e-4
Regularization	alpha	1e-6
	Dropout Rate	0.5
	Class Weights	Balanced (computed)

C. Evaluation Metrics

The model was evaluated using standard classification metrics, including Accuracy (Acc), Macro-F1 (M-F1), Specificity (Spec), and the False Positive Rate (FPR), to assess overall classification performance, class-wise performance, and the balance between precision and recall.

To account for class imbalance in the Maling dataset, M-F1 is reported. M-F1 averages performance across all classes, making it particularly suitable for assessing performance on minority and difficult malware families.

Additionally, Spec and FPR are used to evaluate classification reliability. Spec measures the correct identification of negative samples, whereas the FPR quantifies the proportion of negative samples incorrectly classified as malware, which is critical for minimizing false alarms in cybersecurity applications.

D. Performance Evaluation Methodology

The effectiveness of the proposed A-HRAB model is evaluated through comparative and class-wise analyses. The proposed model is benchmarked against a baseline CNN and two attention-based variants (CNN + SE and CNN + CBAM). To ensure a fair comparison, all models share the same CNN backbone architecture, preprocessing pipeline, data augmentation strategy, optimizer, learning rate, batch size, and training schedule. In addition to overall metrics, precision, recall, F1-score, and confusion matrices are analyzed for each malware family to assess class-level performance, particularly for complex and overlapping classes.

To further examine classification difficulty, intra-class and inter-class distances are computed using deep features extracted from the final layer before the SoftMax classifier. The intra-class distance is measured as the average pairwise Euclidean distance among samples belonging to the same class:

$$D_{\text{intra}}^{(i)} = \frac{1}{M_i} \sum_{p,q \in C_i} \|f_p - f_q\|_2 \quad (15)$$

where C_i denotes class i , f_p , and f_q represent the feature embeddings of two samples within the same class, and M_i is the total number of pairwise combinations. A higher intra-class distance indicates greater internal variability and increased classification difficulty.

For inter-class analysis, the centroid of each class is computed as the mean feature embedding of all samples within that class. The distances between class centroids provide a global measure of feature-space separability and the overall discriminative capability of the learned representations, although they do not explicitly capture local decision boundaries. The inter-class distance is defined as the minimum Euclidean distance between each class centroid and every other class centroid:

$$D_{\text{inter}}^{(i)} = \min_{k \neq i} \|\mu_i - \mu_k\|_2 \quad (16)$$

where μ_i and μ_k denote the centroids of classes i and k , respectively. A lower inter-class distance indicates

stronger overlap with neighboring malware families and reduced class separability.

V. EXPERIMENTAL RESULTS

This section presents a comprehensive evaluation of the proposed A-HRAB model on both the Maling and MaleVis datasets using overall performance metrics and detailed class-wise analyses. The proposed model is compared against a baseline CNN and attention-based variants (CNN + SE and CNN + CBAM) to assess the effectiveness of the proposed architecture across both datasets.

A. Performance Evaluation on the Maling Dataset

As summarized in Table IV, the experimental results demonstrate that A-HRAB consistently outperforms the baseline and attention-augmented CNN models across all evaluation metrics. While the baseline CNN achieved an accuracy of 0.891, the inclusion of attention mechanisms significantly improved performance, with CNN + CBAM and CNN + SE achieving accuracies of 0.979 and 0.971, respectively. Across five independent runs with different random seeds, the proposed A-HRAB achieved an average accuracy of 0.994 ± 0.0024 , a Macro-F1 score of 0.986 ± 0.0037 , and a weighted F1-score of 0.994 ± 0.0031 , demonstrating stable and reproducible performance with minimal sensitivity to random initialization. In addition, A-HRAB achieved the lowest false positive rate of 0.0002 among all evaluated models.

TABLE IV. PERFORMANCE COMPARISON OF THE EVALUATED MODELS ON THE MALING DATASET

Model	Acc	M-F1	F1	Spec	FPR	Trainable parameters	GFLOPs
CNN	0.891	0.875	0.88	0.995	0.0044	461,977	1.4285
CNN+CBAM	0.979	0.958	0.98	0.999	0.0008	1,263,873	1.562
CNN+SE	0.971	0.936	0.97	0.998	0.0011	1,263,477	1.5572
A-HRAB	0.994	0.986	0.994	0.999	0.0002	1,264,357	1.565

These improvements represent a substantial enhancement in feature representation. Unlike conventional attention modules, A-HRAB integrates channel and spatial attention within a residual learning framework and introduces a learnable adaptive scalar fusion mechanism that dynamically balances the contributions of parallel channel and spatial attention based on the input features. This design enables the model to dynamically balance channel (“what”) and spatial (“where”) information, resulting in more discriminative and context-aware feature learning. The improvement in Macro-F1 is particularly important, as it indicates consistent performance across all classes, including minority and difficult categories, rather than performance gains driven primarily by dominant classes.

The learning curves in Fig. 6 illustrate the training and validation accuracy and loss throughout the training process. The proposed model converges rapidly, achieving near-optimal performance within relatively few epochs. Although minor fluctuations are observed in the validation curve during the early epochs, the model quickly stabilizes, suggesting effective learning. The close alignment between the training and validation accuracy

curves, together with the smooth decline in the loss curves shown in Fig. 7, indicates stable optimization, minimal overfitting, and consistent generalization. This stability can be attributed to the integration of residual connections and batch-normalized feature transformations, which facilitate stable gradient propagation.

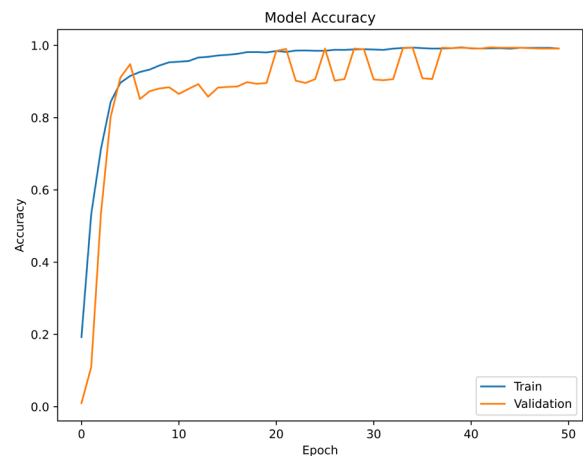


Fig. 6. Training and validation accuracy Maling dataset.

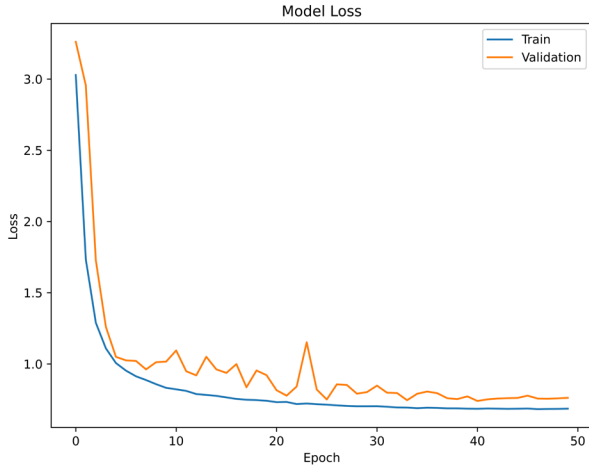


Fig. 7. Training and validation loss on Maling dataset.

B. Performance Evaluation on the MaleVis Dataset

To further validate the proposed A-HRAB, additional experiments were conducted on the MaleVis dataset using the same network architecture and training protocol as those used for the Maling dataset, thereby ensuring a fair comparison across all evaluated models.

TABLE V. PERFORMANCE COMPARISON OF THE EVALUATED MODELS ON THE MALEVIS DATASET

Model	Acc	M-F1	F1	Spec	FPR	Trainable parameters	GFLOPs
CNN	0.91	0.90	0.91	0.996	0.0037	461,977	1.4285
CNN + CBAM	0.943	0.941	0.942	0.997	0.0023	1,263,873	1.562
CNN + SE	0.94	0.939	0.94	0.997	0.0024	1,263,477	1.5572
A-HRAB	0.961	0.96	0.961	0.998	0.0016	1,264,357	1.565

Figs. 8 and 9 present the training and validation accuracy and loss curves, respectively. Similar to the results obtained on the Maling dataset, the proposed model converges rapidly and exhibits stable optimization throughout the training process. Despite minor fluctuations during the early training epochs, the validation performance steadily improves and closely follows the training curves, indicating good generalization and minimal overfitting.

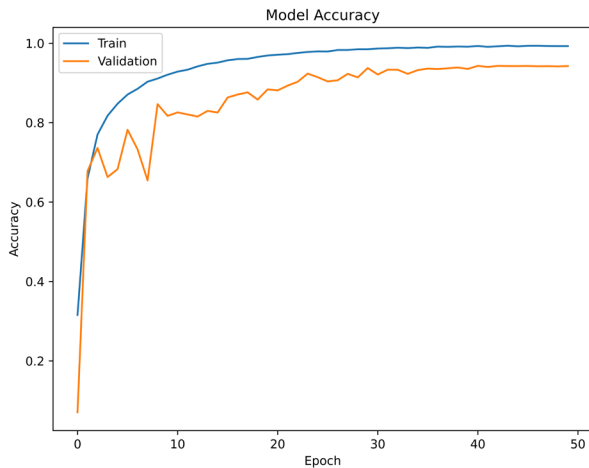


Fig. 8. Training and validation accuracy on MaleVis dataset.

As summarized in Table V, the proposed A-HRAB outperforms the baseline CNN and the attention-based variants (CNN + SE and CNN + CBAM) across all evaluation metrics. The baseline CNN achieved an accuracy of 0.910 and a Macro-F1 score of 0.900, while the incorporation of CNN + SE and CNN + CBAM improved the accuracy to 0.940 and 0.943, respectively. The proposed A-HRAB further improved the classification accuracy to 0.961 ± 0.003 , achieving the highest Macro-F1 score of 0.960 ± 0.0041 , Weighted F1-score of 0.961 ± 0.0044 , and specificity of 0.998, while reducing the false positive rate to 0.0016.

Although the overall performance on the MaleVis dataset is slightly lower than that achieved on the Maling dataset, this result is expected given the greater complexity of the MaleVis dataset. Nevertheless, the consistent improvements over both CNN + SE and CNN + CBAM demonstrate that the proposed learnable adaptive scalar fusion mechanism effectively enhances feature representation and maintains robust discriminative performance under more challenging classification scenarios. These findings indicate that the proposed A-HRAB generalizes well across different malware image datasets and is not limited to the characteristics of a single benchmark.

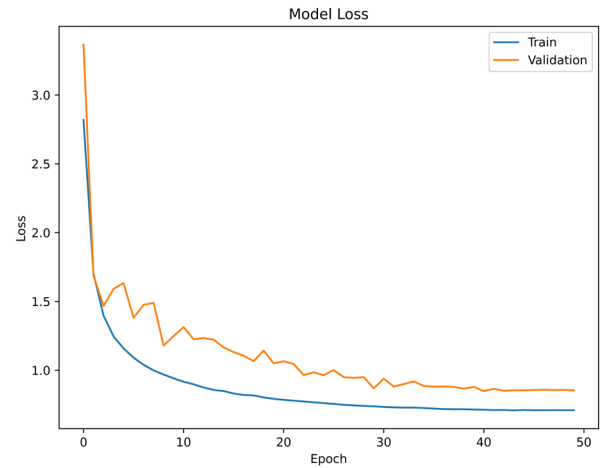


Fig. 9. Training and validation loss on MaleVis dataset.

C. Class-Wise Performance and Effect of Class Imbalance on the Maling Dataset

A deeper class-wise analysis confirms that the superiority of A-HRAB is most evident in challenging classification scenarios. The confusion matrix in Fig. 10 shows strong diagonal dominance, indicating minimal misclassification across most malware families, while Table VI demonstrates consistently high Precision (P), Recall (R), and F1-scores across all 25 classes. Unlike the

baseline CNN, which exhibits significant performance degradation on difficult classes, A-HRAB maintains stable performance even for malware families characterized by high intra-class variability and strong inter-class similarity.

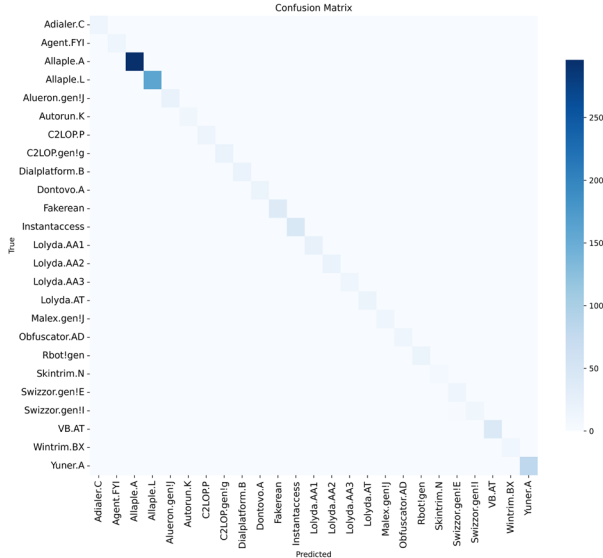


Fig. 10. Confusion matrix of A-HRAB on Maling dataset.

The most challenging classes are C2LOP.P, Swizzor.gen!E, and Swizzor.gen!I, which exhibit the highest intra-class distances (5.2589, 5.0128, and 4.8231, respectively) and the lowest inter-class distances (approximately 1.55 for the Swizzor variants), indicating high internal variability and substantial overlap with neighboring classes. These characteristics make them

inherently difficult to separate. As shown in Table VI, the baseline CNN achieves only an F1-score of 0.90 for C2LOP.P, 0.69 for Swizzor.gen!E, and 0.62 for Swizzor.gen!I. Although CNN + CBAM improves these results slightly, performance remains unstable, while CNN+SE shows severe degradation, particularly for Swizzor.gen!E (F1-score = 0.40). In contrast, A-HRAB achieves F1-scores of 0.97 for C2LOP.P, 0.90 for Swizzor.gen!E, and 0.85 for Swizzor.gen!I, demonstrating stronger discriminative capability in overlapping feature spaces.

A similar observation can be made for Yuner.A, where the baseline CNN completely fails (F1-score = 0.00) despite having a relatively large number of training samples (640). This finding indicates that classification difficulty is not solely determined by class size. Although Yuner.A has moderate intra-class compactness, its limited separability from other malware families causes severe misclassification in conventional CNNs. A-HRAB effectively mitigates this issue, achieving perfect precision, recall, and F1-score, thereby highlighting its ability to enhance feature discrimination beyond simple class balancing.

In contrast, classes such as Obfuscator.AD, Dialplatform.B, Allapple.L, and Rbot!gen exhibit low intra-class distances and high inter-class separability, resulting in compact and well-separated feature distributions. These classes achieve near-perfect performance across all evaluated models, regardless of sample size. For example, Obfuscator.AD and Dialplatform.B achieve an F1-score of 1.00 across the baseline CNN, CNN + CBAM, CNN + SE, and A-HRAB, confirming that well-separated classes are less sensitive to attention design.

TABLE VI. CLASS-WISE PERFORMANCE OF THE EVALUATED MODELS ON THE MALING DATASET

Malware Family	A-HRAB			CNN			CNN + CBAM			CNN + SE			Training Samples
	P	R	F1	P	R	F1	P	R	F1	P	R	F1	
Adialer.C	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.92	0.96	1.00	1.00	1.00	97
Agent.FYI	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	92
Allapple.A	1.00	1.00	1.00	1.00	0.98	0.99	1.00	0.99	0.99	1.00	0.98	0.99	2359
Allapple.L	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1272
Alueron.gen!j	1.00	1.00	1.00	0.84	1.00	0.91	1.00	1.00	1.00	0.95	1.00	0.98	158
Autorun.K	1.00	1.00	1.00	0.13	1.00	0.23	1.00	1.00	1.00	1.00	1.00	1.00	84
C2LOP.P	0.94	1.00	0.97	0.93	0.88	0.90	0.64	1.00	0.78	0.65	0.94	0.77	116
C2LOP.gen!g	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.90	0.95	0.85	0.85	0.85	160
Dialplatform.B	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	141
Dontovo.A	1.00	1.00	1.00	0.94	1.00	0.97	0.94	1.00	0.97	1.00	1.00	1.00	129
Fakerean	1.00	0.97	0.99	1.00	0.97	0.99	1.00	1.00	1.00	1.00	0.97	0.99	304
Instantaccess	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	344
Lolyda.AA1	1.00	1.00	1.00	0.85	1.00	0.92	0.96	1.00	0.98	0.96	1.00	0.98	170
Lolyda.AA2	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	147
Lolyda.AA3	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	98
Lolyda.AT	1.00	1.00	1.00	1.00	0.76	0.87	1.00	0.94	0.97	1.00	0.94	0.97	127
Malex.gen!j	1.00	1.00	1.00	1.00	0.87	0.93	0.83	1.00	0.91	1.00	1.00	1.00	108
Obfuscator.AD	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	113
Rbot!gen	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	126
Skintrim.N	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	64
Swizzor.gen!E	0.87	0.93	0.90	0.67	0.71	0.69	0.85	0.79	0.81	0.67	0.29	0.40	102
Swizzor.gen!I	0.92	0.79	0.85	0.60	0.64	0.62	0.90	0.64	0.75	0.54	0.93	0.68	105
VB.AT	1.00	1.00	1.00	1.00	0.98	0.99	0.98	0.98	0.98	1.00	1.00	1.00	326
Wintrim.BX	0.92	1.00	0.96	0.79	1.00	0.88	1.00	0.82	0.90	0.89	0.73	0.80	77
Yuner.A	1.00	1.00	1.00	0.00	0.00	0.00	1.00	1.00	1.00	1.00	1.00	1.00	640

The results in Table VI further show that A-HRAB consistently improves performance on minority and difficult classes such as Autorun.K, Wintrim.BX, and Lolyda.AT. For instance, Autorun.K improves dramatically from an F1-score of 0.23 in the baseline CNN to 1.00 in A-HRAB, while Wintrim.BX improves from 0.88 to 0.96. These gains are particularly important because they directly influence the Macro-F1 score, which assigns equal weight to all classes regardless of their size.

These findings show that classification performance is driven more by feature distribution than by class imbalance alone. While traditional CNNs perform well on majority or well-separated classes, they struggle with overlapping and highly variable categories. In contrast, A-HRAB remains robust across both easy and difficult classes, as reflected by its higher Macro-F1 score and lower misclassification rate. This improvement is mainly attributed to its parallel attention mechanism and learnable adaptive scalar fusion mechanism, which dynamically balances channel-wise and spatial feature representations, resulting in more discriminative and reliable feature learning.

D. Class-Wise Performance and Effect of Class Imbalance on the MaleVis Dataset

A deeper class-wise analysis further confirms the effectiveness of A-HRAB on the MaleVis dataset. The confusion matrix in Fig. 11 exhibits strong diagonal dominance, indicating minimal misclassification across

most malware families, while Table VII shows consistently high Precision (P), Recall (R), and F1-scores for all 25 classes. Compared with the baseline CNN and the CNN + SE and CNN + CBAM models, A-HRAB demonstrates greater robustness for malware families characterized by high intra-class variability and strong inter-class similarity.

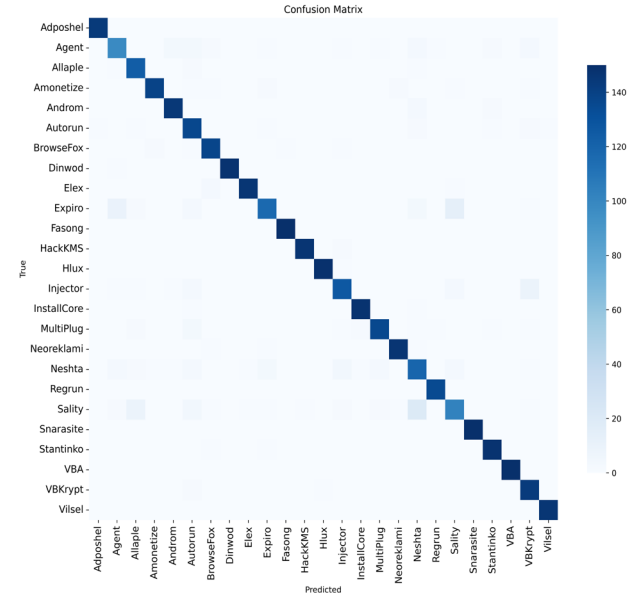


Fig. 11. Confusion matrix of A-HRAB on MaleVis dataset.

TABLE VII. CLASS-WISE PERFORMANCE OF THE EVALUATED MODELS ON THE MALEVIS DATASET

Malware Family	A-HRAB			CNN			CNN + CBAM			CNN + SE			Training Samples
	P	R	F1	P	R	F1	P	R	F1	P	R	F1	
Adposhel	0.99	1.00	1.00	0.99	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	144
Agent	0.87	0.87	0.87	0.65	0.83	0.73	0.82	0.84	0.83	0.86	0.84	0.85	120
Allaple	0.94	0.98	0.96	0.90	0.92	0.91	0.92	0.96	0.94	0.87	0.98	0.92	128
Amonetize	0.98	0.97	0.97	0.96	0.91	0.94	0.99	0.93	0.95	0.98	0.94	0.96	147
Androm	0.94	0.97	0.96	0.95	0.98	0.96	0.92	0.97	0.95	0.95	0.96	0.95	150
Autorun	0.90	0.94	0.92	0.78	0.86	0.82	0.82	0.95	0.88	0.78	0.93	0.85	146
BrowseFox	0.95	0.97	0.96	0.90	0.96	0.93	0.95	0.96	0.95	0.95	0.96	0.95	143
Dinwod	0.98	0.99	0.99	0.99	0.98	0.98	1.00	0.97	0.99	1.00	0.99	0.99	149
Elex	0.98	0.98	0.98	0.98	0.89	0.93	0.98	0.98	0.98	0.98	0.97	0.98	150
Expiro	0.95	0.92	0.93	0.84	0.58	0.68	0.87	0.80	0.83	0.89	0.78	0.83	151
Fasong	0.99	1.00	0.99	1.00	1.00	1.00	0.99	1.00	1.00	0.99	1.00	1.00	150
HackKMS	0.99	0.99	0.99	0.99	0.96	0.97	0.98	0.99	0.98	0.99	0.99	0.99	149
Hlux	1.00	1.00	1.00	0.99	0.99	0.99	0.97	0.99	0.98	0.99	1.00	0.99	150
Injector	0.96	0.88	0.92	0.93	0.77	0.84	0.92	0.86	0.89	0.93	0.86	0.89	145
InstallCore	0.99	0.99	0.99	0.99	0.98	0.98	0.99	0.99	0.99	0.99	0.99	0.99	150
MultiPlug	0.99	0.94	0.97	0.95	0.89	0.92	0.97	0.91	0.94	0.99	0.90	0.94	149
Neoreklami	0.98	0.99	0.98	0.99	0.96	0.97	0.99	0.99	0.99	0.98	0.99	0.98	150
Neshta	0.85	0.86	0.86	0.74	0.77	0.75	0.77	0.78	0.77	0.80	0.78	0.79	147
Regrun	1.00	1.00	1.00	0.99	1.00	1.00	1.00	1.00	1.00	0.99	1.00	0.99	135
Sality	0.88	0.83	0.85	0.56	0.69	0.62	0.83	0.75	0.79	0.82	0.77	0.79	149
Snarasite	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.99	1.00	0.99	150
Stantinko	0.98	0.99	0.99	0.97	0.98	0.97	0.97	0.99	0.98	0.97	0.99	0.98	150
VBA	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	150
VBKrypt	0.92	0.98	0.95	0.82	0.82	0.82	0.91	0.97	0.94	0.86	0.91	0.89	146
Vilsel	0.99	1.00	1.00	0.99	1.00	1.00	0.99	1.00	1.00	0.99	1.00	1.00	146

The most challenging malware families include VBKrypt, Agent, Neshta, Injector, Expiro, and Sality, which exhibit the largest intra-class distances (7.4034, 6.4261, 6.4238, 6.2507, 6.1704, and 6.0879, respectively) together with relatively small inter-class distances, indicating substantial feature variation within each class

and greater similarity to neighboring malware families. These characteristics make them inherently more difficult to classify. As shown in Table VII, the baseline CNN achieved F1-scores of 0.82 for VBKrypt, 0.73 for Agent, 0.75 for Neshta, 0.84 for Injector, 0.68 for Expiro, and 0.62 for Sality. Although CNN + CBAM and CNN + SE

improved performance for several of these classes, their performance remains inconsistent. In contrast, A-HRAB achieved F1-scores of 0.95, 0.87, 0.86, 0.92, 0.93, and 0.85, respectively, demonstrating its superior capability to distinguish highly overlapping malware families.

Conversely, malware families such as VBA, Hlux, Vilsel, Snarasite, InstallCore, and Regrun exhibit compact intra-class distributions and large inter-class separations, resulting in well-defined feature representations. These classes consistently achieve near-perfect precision, recall, and F1-scores across all evaluated models. For example, VBA, Adposhel, and Snarasite achieved an F1-score of 1.00, confirming that visually distinctive malware families can be reliably classified regardless of the attention mechanism employed.

Unlike the Maling dataset, the MaleVis dataset is relatively balanced, with each malware family contributing approximately the same number of training samples. Consequently, the observed performance differences are primarily driven by feature-space characteristics rather than class imbalance. For instance, Agent, despite having only 120 training samples, achieves a higher F1-score than Sality, which has 149 training samples, indicating that classification difficulty is governed more by feature overlap than by sample availability. Similarly, Regrun achieves perfect classification despite having fewer training samples than many other malware families, further demonstrating that the proposed model is robust to moderate variations in class distribution.

Overall, these findings indicate that A-HRAB learns more discriminative feature representations, characterized by lower intra-class variability and greater separation between malware families. This behavior is reflected in its higher Macro-F1 score and lower false positive rate, and can be attributed to the learnable adaptive scalar fusion mechanism, which dynamically balances channel-wise and spatial attention to produce more discriminative and reliable feature representations across both well-separated and highly overlapping malware families.

E. Comparison with State-of-the-Art Methods

To further assess the effectiveness of the proposed model, A-HRAB is compared with several recent malware image classification methods on the Maling and MaleVis datasets, as summarized in Table VIII. Since these studies employ different dataset partitions, preprocessing pipelines, data augmentation strategies, backbone architectures, and evaluation protocols, the comparison is intended to provide a general benchmark of performance rather than a strictly controlled head-to-head evaluation.

For the Maling dataset, existing methods such as SACNNVGG19 [29], GCSA-ResNet [17], IMCMK-CNN [30], RegNetY320 [24], MobileNet fine-tuning [26], and 3D-ResNet-18 [32] have demonstrated excellent performance, with reported accuracies ranging from 0.9762 to 0.996. The proposed A-HRAB achieves an accuracy of 0.994, a Macro-F1 score of 0.986, a Weighted F1-score of 0.994, and the lowest reported false positive rate of 0.0002 among the methods reporting this metric. Although the 3D-ResNet-18 model

reports a slightly higher accuracy, A-HRAB delivers highly competitive performance while demonstrating strong robustness for challenging malware families.

For the MaleVis dataset, direct comparison should be interpreted with additional caution because the published studies predominantly adopt an 80:20 train-test split, whereas the present work uses the official predefined training and validation partitions provided with the dataset. Despite these differences, A-HRAB achieves an accuracy of 0.961, a Macro-F1 score of 0.96, a Weighted F1-score of 0.961, a specificity of 0.998, and a false positive rate of 0.0016, outperforming MobileNet fine-tuning [26], CNN-AutoMIC [33], and the EfficientNetB0 framework [34] in overall classification accuracy and Macro-F1 score. Although the 3D-ResNet-18 model [32] reports a slightly higher accuracy, the comparison should be interpreted with caution because the models were evaluated under different experimental protocols. Nevertheless, A-HRAB demonstrates highly competitive performance while providing consistently stronger class-wise robustness.

TABLE VIII. COMPARISON WITH STATE-OF-THE-ART METHODS

Dataset	Model	Acc	M-F1	F1	Spec	FPR
Maling	[29]	0.9762	0.972	-	0.9768	-
	[17]	0.9908	0.9834	0.9908	0.9996	0.0004
	[30]	0.9925	0.9927	-	-	-
	[24]	0.9930	0.9930	-	-	-
	[26]	0.9907	0.9906	-	-	-
	[32]	0.996	-	0.999	-	-
	Our work	0.994	0.986	0.994	0.999	0.0002
Malevis	[26]	0.96	0.96	-	-	-
	[33]	0.87	0.87	-	-	-
	[32]	99.46	-	0.99	-	-
	[34]	0.943	0.9426	0.95	-	-
	Our work	0.961	0.96	0.961	0.998	0.0016

Overall, the results on both datasets demonstrate that A-HRAB consistently achieves highly competitive performance across different malware image benchmarks. More importantly, the proposed learnable adaptive scalar attention fusion mechanism improves class-wise robustness, particularly for malware families exhibiting high intra-class variability and strong inter-class similarity, thereby confirming its effectiveness beyond improvements in aggregate performance metrics.

VI. CONCLUSION

This paper presented an Adaptive Hybrid Residual Attention Block (A-HRAB) for malware image classification, introducing a learnable adaptive fusion mechanism within a residual attention framework to dynamically balance parallel channel and spatial attention according to the characteristics of the input. By extending existing attention mechanisms with input-dependent adaptive fusion, the proposed architecture enhances feature representation and improves the discrimination of visually similar and structurally complex malware families.

Experimental evaluation on the Maling and MaleVis datasets demonstrated that the proposed A-HRAB outperforms the baseline CNN as well as CNN + SE and

CNN + CBAM across multiple evaluation metrics. In addition, class-wise analysis showed that the proposed model is particularly effective for challenging malware families exhibiting high intra-class variability and strong inter-class similarity.

Nevertheless, several limitations should be acknowledged. Resizing malware images to a fixed resolution of 224×224 pixels may introduce interpolation artifacts and distort fine-grained structural patterns. Additionally, the integration of attention mechanisms introduces a minor computational overhead compared to a standard CNN, although the resulting performance gains yield a favorable trade-off between efficiency and classification accuracy. Future work will investigate aspect-ratio-preserving preprocessing, adaptive spatial pooling, and evaluation on more recent malware datasets.

CONFLICT OF INTEREST

The authors declare that they have no conflicts of interest to this work.

AUTHOR CONTRIBUTIONS

Omar Al-Harbi: Conceptualization, Data curation, Formal Analysis, Investigation, Methodology, Project administration, Software, Visualization, Validation, Writing—original draft, Writing—review & editing. Areej AlShorman: Data curation, Formal Analysis, Investigation, Software, Visualization, Validation, Writing—original draft, Writing—review & editing. Both authors had approved the final version.

REFERENCES

- [1] L. Abrardi, S. Comino, and S. Grassini, "The economics of cyber risk: A survey of the literature," *Journal of Industrial and Business Economics*, 2025. doi: 10.1007/s40812-025-00370-3
- [2] AV-TEST Institute. Malware statistics. [Online]. Available: <https://www.av-test.org/en/statistics/malware/>
- [3] J. Geng, J. Wang, Z. Fang, Y. Zhou, D. Wu, and W. Ge, "A survey of strategy-driven evasion methods for PE malware: Transformation, concealment, and attack," *Comput. Secur.*, vol. 137, 2024. doi: 10.1016/j.cose.2023.103595
- [4] A. A. Al-Hashmi *et al.*, "Deep-Ensemble and Multifaceted Behavioral Malware Variant Detection Model," *IEEE Access*, vol. 10, 2022. doi: 10.1109/ACCESS.2022.3168794
- [5] F. A. Aboaoja, A. Zainal, F. A. Ghaleb, B. A. S. Al-rimy, T. A. E. Eisa, and A. A. H. Elnour, "Malware Detection Issues, Challenges, and Future Directions: A Survey," *Applied Sciences*, vol. 12, no. 17, 8482 2022. doi: 10.3390/app12178482
- [6] M. S. Alam and S. T. Vuong, "Random forest classification for detecting android malware," in *Proc. 2013 IEEE International Conference on Green Computing and Communications and IEEE Internet of Things and IEEE Cyber, Physical and Social Computing, GreenCom-iThings-CPSCOM 2013*, 2013. doi: 10.1109/GreenCom-iThings-CPSCOM.2013.122
- [7] M. M. Abualhaj, A. A. Abu-Shareha, Q. Y. Shambour, A. Alsaaidah, S. N. Al-Khatib, and M. Anbar, "Customized K-nearest neighbors' algorithm for malware detection," *International Journal of Data and Network Science*, vol. 8, no. 1, 2024. doi: 10.5267/j.ijdns.2023.9.012
- [8] J. Ferdous, R. Islam, A. Mahboubi, and M. Z. Islam, "A Survey on ML Techniques for Multi-Platform Malware Detection: Securing PC, Mobile Devices, IoT, and Cloud Environments," *Sensors*, vol. 25, no. 4, 1153 2025. doi: 10.3390/s25041153
- [9] L. Nataraj, S. Karthikeyan, G. Jacob, and B. S. Manjunath, "Malware images: Visualization and automatic classification," in *Proc. the 8th International Symposium on Visualization for Cyber Security*, 2011. doi: 10.1145/2016904.2016908.
- [10] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Commun. ACM*, vol. 60, no. 6, 2017. doi: 10.1145/3065386
- [11] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. 2015 the 3rd International Conference on Learning Representations*, 2015.
- [12] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in *Proc. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778. doi: 10.1109/CVPR.2016.90
- [13] J. Hu, L. Shen, S. Albanie, G. Sun, and E. Wu, "Squeeze-and-Excitation Networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 8, 2020. doi: 10.1109/TPAMI.2019.2913372
- [14] Y. Wang *et al.*, "An attention mechanism module with spatial perception and channel information interaction," *Complex and Intelligent Systems*, vol. 10, no. 4, 2024. doi: 10.1007/s40747-024-01445-9
- [15] J. Park, S. Woo, J. Y. Lee, and I. S. Kweon, "BAM: Bottleneck attention module," in *Proc. British Machine Vision Conference 2018*, 2018.
- [16] S. Woo, J. Park, J. Y. Lee, and I. S. Kweon, "CBAM: Convolutional block attention module," in *Proc. European Conference on Computer Vision*, 2018, pp. 3–19.
- [17] Y. Fan, K. Zhang, B. Zheng, Y. Zhou, J. Zhou, and W. Pan, "GCSA-ResNet: A deep neural network architecture for Malware detection," *Sci. Rep.*, vol. 15, no. 1, 2025. doi: 10.1038/s41598-025-10561-6
- [18] M. Dhingra *et al.*, "Image-Based Malware Classification Using DCGAN-Augmented Data and a CNN–Transformer Hybrid Model," *Future Internet*, vol. 18, no. 2, 2026. doi: 10.3390/fi18020102
- [19] J. Parchami, S. R. Talebiyan, A. A. Dahham, D. D. Husam, and A. Darroudi, "A hybrid deep learning approach for malware detection using generative adversarial network-based augmentation and multilevel feature selection," *Computers and Electrical Engineering*, vol. 132, 110997, Apr. 2026. doi: 10.1016/j.compeleceng.2026.110997
- [20] O. A. Madamidola, F. Ngobigha, and A. Ez-zizi, "Detecting new obfuscated malware variants: A lightweight and interpretable machine learning approach," *Intelligent Systems with Applications*, vol. 25, 2025. doi: 10.1016/j.iswa.2024.200472
- [21] A. S. Shatnawi, Q. Yassen, and A. Yateem, "An android malware detection approach based on static feature analysis using machine learning algorithms," *Procedia Computer Science*, vol. 201, pp. 653–658, 2022. doi: 10.1016/j.procs.2022.03.086
- [22] R. Kharoubi, K. Oualkacha, and A. Mkhadri, "The cluster correlation-network support vector machine for high-dimensional binary classification," *J. Stat. Comput. Simul.*, vol. 89, no. 6, 2019. doi: 10.1080/00949655.2019.1575382
- [23] S. Yoo, S. Kim, S. Kim, and B. B. Kang, "AI-HydRa: Advanced hybrid approach using random forest and deep learning for malware classification," *Information Sciences*, vol. 546, 2021. doi: 10.1016/j.ins.2020.08.082
- [24] K. Shaikat, S. Luo, and V. Varadharajan, "A novel machine learning approach for detecting first-time-appeared malware," *Eng. Appl. Artif. Intell.*, vol. 131, 2024. doi: 10.1016/j.engappai.2023.107801
- [25] W. El-Shafai, I. Almomani, and A. Alkhayer, "Visualized malware multi-classification framework using fine-tuned cnn-based transfer learning models," *Applied Sciences*, vol. 11, no. 14, 2021. doi: 10.3390/app11146446
- [26] M. Palma Salas and P. L. de Geus, "Deep Learning Applied to Imbalanced Malware Datasets Classification," *Journal of Internet Services and Applications*, vol. 15, no. 1, 2024. doi: 10.5753/jisa.2024.3907
- [27] B. Al-Masri, N. Bakir, A. El-Zaart, and K. Samrouth, "Dual Convolutional Malware Network (DCMN): An Image-Based Malware Classification Using Dual Convolutional Neural Networks," *Electronics*, vol. 13, no. 18, 2024. doi: 10.3390/electronics13183607
- [28] M. Ashawa, N. Owah, S. Hosseinzadeh, and J. Osamor, "Enhanced Image-Based Malware Classification Using Transformer-Based Convolutional Neural Networks (CNNs)," *Electronics*, vol. 13, no. 20, 2024. doi: 10.3390/electronics13204081

- [29] M. J. Awan *et al.*, "Image-based malware classification using vgg19 network and spatial convolutional attention," *Electronics*, vol. 10, no. 19, 2021. doi: 10.3390/electronics10192444
- [30] D. Zhang, Y. Song, Q. Xiang, and Y. Wang, "IMCMK-CNN: A lightweight convolutional neural network with Multi-scale Kernels for Image-based Malware Classification," *Alexandria Engineering Journal*, vol. 111, 2025. doi: 10.1016/j.aej.2024.10.055
- [31] A. S. Bozkir, A. O. Cankaya, and M. Aydos, "Utilization and comparison of convolutional neural networks in malware recognition," in *Proc. 27th Signal Processing and Communications Applications Conference*, 2019. doi: 10.1109/SIU.2019.8806511
- [32] W. Al-Khater and S. Al-Madeed, "Using 3D-VGG-16 and 3D-Resnet-18 deep learning models and FABEMD techniques in the detection of malware," *Alexandria Engineering Journal*, vol. 89, 2024. doi: 10.1016/j.aej.2023.12.061
- [33] S. Andriani, S. Galantucci, A. Iannacone, A. Maci, and G. Pirlo, "CNN-AutoMIC: Combining convolutional neural network and autoencoder to learn non-linear features for KNN-based malware image classification," *Comput. Secur.*, vol. 156, 2025. doi: 10.1016/j.cose.2025.104507
- [34] E. Eroglu Demirkan and M. Aydos, "Enhancing Malware Detection via RGB Assembly Visualization and Hybrid Deep Learning Models," *Applied Sciences*, vol. 15, no. 13, 2025. doi: 10.3390/app15137163

Copyright © 2026 by the authors. This is an open access article distributed under the Creative Commons Attribution License which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited ([CC BY 4.0](https://creativecommons.org/licenses/by/4.0/)).