

基于特征融合的宽度—深度学习恶意软件检测

赵月爱^{1,2}, 刘美晨^{1,2}, 展翼乐^{1,2}, 王玲³

(1. 太原师范学院 计算机科学与技术学院, 山西 晋中 030602;
2. 山西智能优化计算与区块链技术重点实验室, 山西 晋中 030619; 3. 山西大学 自动化与软件学院, 山西 太原 030006)

摘要: 针对现有恶意软件检测方法存在的图像表现形式不丰富和特征提取不充分问题, 提出一种基于金字塔特征融合的宽度—深度学习恶意软件检测模型 EffNet-IPDBL。首先, 将恶意软件可视化为多通道图像, 以增强样本数据特征; 其次, 在 EfficientNetB3 模块中引入双通道注意力机制, 以加强通道特征选择能力, 获取更多信息强化图像特征; 再次, 使用金字塔特征融合多尺度多层次图像特征, 同时捕捉全局语义和局部细节, 以提升系统鲁棒性; 最后, 采用增量主成分分析优化金字塔深度—宽度学习模块, 使其更适用于处理大规模数据集。在自建数据集和 Maling 公开数据集上进行实验, 结果表明该模型的检测准确率分别达到 88.35% 和 98.69%, 能有效检测恶意软件。

关键词: 恶意软件; 宽度学习系统; 特征融合; EfficientNet; 增量主成分分析

DOI: 10.11907/rjdk.251618

扫描二维码阅读全文:

中图分类号: TP309.5

文献标识码: A

文章编号: 1672-7800(2026)003-0141-09

Wide-Deep Learning Malware Detection Based on Feature Fusion

ZHAO Yueai^{1,2}, LIU Meichen^{1,2}, ZHAN Yile^{1,2}, WANG Ling³

(1. School of Computer Science and Technology, Taiyuan Normal University, Jinzhong 030602, China;
2. Shanxi Key Laboratory of Intelligent Optimization Computing and Blockchain Technology, Jinzhong 030619, China;
3. School of Automation and Software, Shanxi University, Taiyuan 030006, China)

Abstract: In order to solve the problems of insufficient image representation and feature extraction in existing malware detection methods, a width depth learning malware detection model EffNet IPDBL based on pyramid feature fusion is proposed. The model first visualizes malware as multi-channel images to enhance the characteristics of sample data; Secondly, the dual channel attention mechanism is introduced into EfficientNetB3 module to enhance the ability of channel feature selection and obtain more information to enhance image features; Thirdly, pyramid features are used to fuse multi-scale and multi-level image features, while capturing global semantics and local details to improve system robustness; Finally, incremental principal component analysis is used to optimize the pyramid depth width learning module, making it more suitable for processing large-scale data sets. Experiments on self-constructed data sets and Maling public data sets show that the detection accuracy of this model reaches 88.35% and 98.69% respectively, which can effectively detect malware.

Key Words: malware; breadth learning system; feature fusion; EfficientNet; incremental principal component analysis

0 引言

2025年1月,中国互联网络信息中心(China Internet Network Information Center, CNNIC)发布的第55次《中国互联网络发展状况统计报告》显示,截至2024年12月,中国

网民规模已从1997年的约62万人增长至11.08亿人,互联网普及率亦相应提升至78.6%,呈现显著增长态势^[1]。然而,互联网的迅速发展为恶意软件的传播提供了媒介,给互联网安全带来了极大威胁。根据卡巴斯基安全网络报告,仅2025年第一季度,全球超过85 000名用户遭受到勒索软件攻击,近315 000名用户遭遇挖矿攻击^[2]。这些攻

收稿日期: 2025-09-04

基金项目: 山西省自然科学研究面上项目(202303021221173); 山西省科技战略研究专项重点项目(202304031401011)

作者简介: 赵月爱(1974-),女,博士,CCF会员,太原师范学院计算机科学与技术学院教授、硕士生导师,研究方向为机器学习、智能信息处理和网络安全等; 刘美晨(1999-),女,太原师范学院计算机科学与技术学院硕士研究生,研究方向为网络与信息安全; 展翼乐(2001-),男,太原师范学院计算机科学与技术学院硕士研究生,研究方向为网络与信息安全; 王玲(1980-),女,硕士,山西大学自动化与软件学院讲师,研究方向为网络安全,网络入侵检测。本文通讯作者: 赵月爱。

击不仅对个人和企业造成影响,而且波及到政府机构。因此,对恶意软件进行检测具有重要的政治、经济和社会价值。

传统的恶意软件检测技术主要包括静态检测、动态检测和两者结合三大类。静态检测方法主要基于可执行程序及其反汇编结果解析,提取字节序列、汇编指令、节区结构和导入函数等特征进行检测。例如,姚斌荣等^[3]提出一种基于组合特征的安卓恶意软件静态检测方法,在检测效率方面具有显著优势,但对采用代码混淆或加壳技术的恶意软件检测效果存在一定局限性。相比之下,动态检测方法能有效缓解混淆技术带来的干扰^[4-5]。例如,Nikolopoulos等^[6]通过动态污点分析构建系统调用依赖图以检测未知程序的潜在恶意行为,有效提升了检测准确性与鲁棒性。然而,该类方法严重依赖逆向工程,操作复杂度高且计算资源消耗大。赵云程等^[7]提出一种静动态结合的双重恶意代码检测系统,兼具静态检测技术的优势,但对可疑代码的检测速度缓慢,不适合实时检测。

近年来,深度学习检测方法由于具有良好的扩展性和稳定性而受到广泛关注^[8]。例如,Nataraj等^[9]将恶意软件转换成灰度图,提取图像的全局特征,能有效识别恶意软件。然而,依赖手动方式进行特征提取往往效率低下,限制了模型检测性能的提升。丁应等^[10]使用双字节特征编码方式将恶意代码转化为灰度图,通过卷积神经网络(Convolutional Neural Network, CNN)提取特征并检测。由于灰度图像所含像素信息相对单一且缺乏显著的纹理或结构特征,难以充分捕捉和反映恶意软件的关键行为特征^[11]。张晓良等^[12]提出一种基于增量学习的恶意软件检测方法,将良性与恶意样本转化为RGB图像后再提取图片特征进行增量训练。王博等^[13]将恶意软件二进制分割成RGB三通道值,然后输入VGG网络中进行分类。Meng等^[14]提出LensDroid技术,结合深度学习与软件可视化,从多个互补视图中自动捕获并聚合高级特征,揭示了隐藏的恶意行为,提升了恶意软件的检测准确性。He等^[15]基于ResNeXt标记模型开发了即插即用的注意力机制,该机制通过捕获恶意软件图像的通道感知视野来优化模型训练,使其更加专注于恶意软件特征。Yadav等^[16]提出一种基于EfficientNet-B4的模型,通过对Android DEX文件进行图像化表示来实现Android恶意软件的检测。陈非等^[17]使用降噪自编码器(Denoising Autoencoder, DAE)抽取特征信息,将特征向量转化为单通道图像,并在特征维度上拼接生成RGB图像,构建BaggingCNN分类算法对图像进行检测。然而,CNN只能处理固定分辨率的图像,导致特征提取模型无法完全利用恶意软件的丰富信息,进而影响检测性能。李思聪等^[18]提出一种名为LKRepNet(Large Kernel Reparameterisation Network)的恶意软件分类网络,提升了检测准确性和效率,但缺乏充分且高质量的训练数据。Lu等^[19]设计了一种基于片段的表示学习方案,结合元学习和片段训练技术,提升了模型对未知恶意软件家族的泛化能力。

在以上研究的基础上,本文提出一种基于金字塔特征融合的宽度—深度学习恶意软件检测模型EffNet-IPDBL(EfficientNetB3-based Incremental Principal Component Analysis-enhanced Pyramidal Deep-Broad Learning Network)。该模型将恶意软件转化为多通道彩色图像,结合改进的EfficientNetB3提取多尺度多层级图像特征,并优化金字塔深度—宽度学习(Pyramidal Deep-Broad Learning, PDBL)模块以适应大规模数据,实现恶意软件检测。本文主要贡献点为:①提出新的恶意软件可视化方法,将恶意软件二进制文件特征、局部熵、相邻字节差分整合,构建全面的特征矩阵,并将其转化为多通道图像,丰富恶意软件的多层次特征信息;②提出融合宽度与深度学习的特征学习网络模型,提高恶意软件检测精度;③在自建数据集和Malimg数据集上对模型进行训练与测试,其检测准确率相较于基准模型分别提升了3.15%和3.57%。

1 理论基础

1.1 PDBL 模型

2018年,Chen等^[20]提出一个可替代深度学习网络的宽度学习系统(Broad Learning System, BLS),其核心思想为利用输入数据映射的特征作为网络的特征节点,然后将这些特征通过非线性激活函数扩展为增强节点,最后将所有特征和增强节点直接连接到输出端,通过快速的伪逆计算或其他优化方法得到输出权重。这种设计使得BLS在结构上不需要深度层次,而是通过增加网络宽度来提升学习能力。为不增加重新训练的负担,提高分类任务中预训练骨干网络的性能,华南理工大学、香港科技大学等7个机构组成的联合研究团队提出一种新颖的即插即用模块,即PDBL^[21]。结构如图1所示。

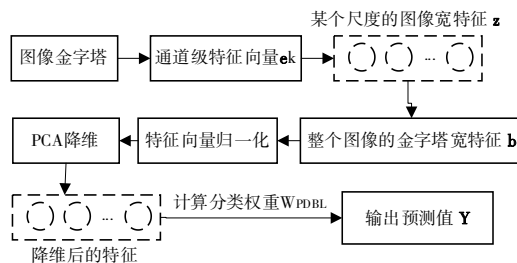


Fig. 1 PDBL module structure

图1 PDBL 模块结构

PDBL模块首先构建图像金字塔,使用双线性插值对图像进行缩放,对输入图像进行多尺度下采样。对于金字塔的每一层图像,通过预训练CNN提取每个阶段 k 的特征图 f_k ,并进行全局平均池化。计算公式为:

$$e_k = \frac{1}{H_k \times W_k} \sum_{i=1}^{H_k} \sum_{j=1}^{W_k} f_k(i, j) \quad (1)$$

式中: H_k 、 W_k 为特征图尺寸; e_k 为压缩后的通道级特征向量。

其次,将所有阶段的特征向量拼接成该尺度图像的宽

特征 $z_i = e_1 \oplus e_2 \oplus \dots \oplus e_n$, 并对所有尺度的宽特征进行拼接, 形成整个图像的金字塔宽特征向量。表示为:

$$b = z_1 \oplus z_2 \oplus \dots \oplus z_s \quad (2)$$

对训练集的所有金字塔宽特征向量进行归一化, 并采用主成分分析(Principal Component Analysis, PCA)降维。基于降维后的特征矩阵和标签, 通过伪逆计算得到分类权重。计算公式为:

$$W_{PDBL} = \lim_{\lambda \rightarrow 0} (\lambda E + A_{train} A_{train}^T)^{-1} A_{train}^T Y_{train} \quad (3)$$

式中: E 为单位矩阵; λ 为正则化系数; A_{train} 为训练集金字塔宽特征进行 PCA 降维后的特征矩阵。最终模型输出的预测值表示为:

$$Y_{test} = A_{test} W_{PDBL} \quad (4)$$

在分类过程中使用 PDBL 模块可以对恶意图像进行多尺度输入, 解决使用深度学习进行图像分类时只能输入固定尺寸图像的问题, 能有效提高模型分类性能。

1.2 EfficientNetB3 模型

谷歌公司在 2019 年国际机器学习大会上推出 EfficientNet 系列模型, 包含从 B0 到 B7 (EfficientNetB0~ EfficientNetB7) 的 8 个模型, 其核心创新在于采用复合系数 (Compound Coefficient) 对 CNN 进行协同缩放, 通过调整网络深度、宽度和分辨率 3 个维度的扩展比例来自动搜索三者间的最优平衡点^[22]。复合系数通过保持参数值平衡来实现, 计算公式为:

$$d = \alpha^\Phi \quad (5)$$

$$w = \beta^\Phi \quad (6)$$

$$r = \gamma^\Phi \quad (7)$$

式中: d 为深度; w 为宽度; r 为分辨率。参数值使用 α, β, γ 来表达, 这 3 个参数需满足的条件为:

$$\alpha \cdot \beta^2 \cdot \gamma^2 \approx 2 \quad (8)$$

本文选择在准确率、计算量与参数量之间达到较优平衡的 EfficientNetB3 变体。模型输入数据首先通过 Stem 层进行快速粗粒度处理; 其次经过由多个 MBConv 模块堆叠而成的骨干网络 (Backbone), 在逐步降低特征图空间分辨率的同时有序增加通道数和 MBConv 模块的重复次数, 实现对图像特征的提取与融合; 最后通过全局平均池化

(Global Average Pooling, GAP) 和全连接层 (Fully Connected Layer, FC) 将提取到的图像特征转化为预测结果。EfficientNetB3 模型各层详细参数如表 1 所示。

Table 1 Detailed parameters of each layer of EfficientNet-B3 model

表 1 EfficientNet-B3 模型各层详细参数

Stage	Operator	Stride	Channels	Repeats
0	Input	—	3	—
1	Conv3×3	2	40	1
2	MBConv1, k3×3	1	24	1
3	MBConv6, k3×3	2	32	2
4	MBConv6, k5×5	2	56	3
5	MBConv6, k3×3	2	112	4
6	MBConv6, k5×5	1	160	4
7	MBConv6, k5×5	2	272	5
8	MBConv6, k3×3	1	448	2
9	Conv1×1 & GAP & FC	—	1 536→1 000	1

2 EffNet-IPDBL 模型构建

EffNet-IPDBL 模型主要由改进的 EfficientNetB3 模块和 IPDBL 模型组成。整体框架如图 2 所示。模型以恶意软件样本生成的 RGB 彩色图像作为输入, 采用多尺度图像生成策略, 以充分捕捉图像的不同层次特征。首先, 利用改进的 EfficientNetB3 模型对多尺度图像进行特征提取, 通过在 MBConv 块中引入双通道注意力机制来增强关键字序列区域的权重, 从而提高对恶意软件样本中重要特征的辨识能力; 其次, 使用金字塔特征融合方法, 对每个尺度的图像分别提取 4 个阶段的特征图 (reduction_2, reduction_3, reduction_4, reduction_5), 经由 GAP 后拼接形成新特征, 这种融合方法能有效结合不同尺度下的局部和全局信息, 增强模型的鲁棒性和适应性; 最后, 将新特征输入 IPDBL 模型中, 对整个图像的金字塔宽特征进行归一化处理, 使用增量主成分分析 (Incremental Principal Component Analysis, IPCA) 进行特征降维, 以降低计算复杂度并避免过拟合。降维后的特征通过伪逆计算得到分类权重, 最终输出预测值。通过层层递进的特征提取、融合和降维步骤不断精炼和优化输入信息, 最终实现高效且精确的恶意软件分类。

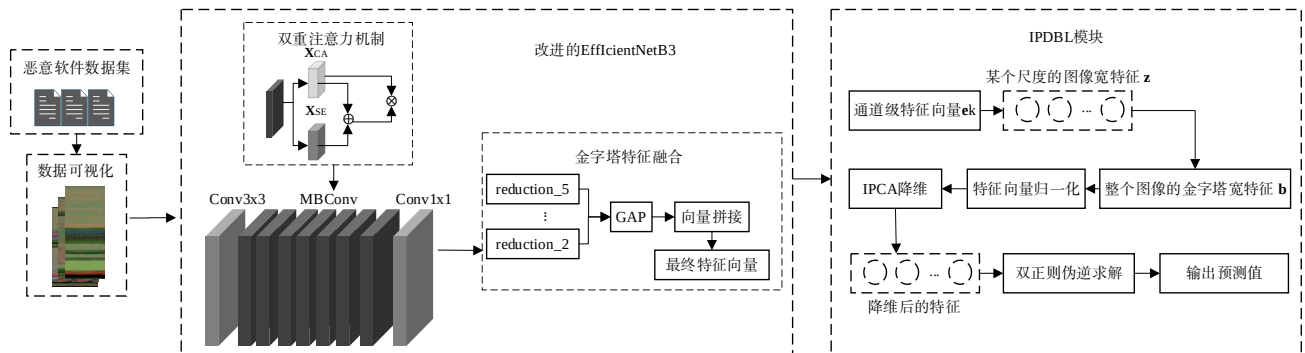


Fig. 2 Overall framework

图 2 整体框架

2.1 恶意软件可视化

本文融合恶意软件二进制文件特征、局部熵与相邻字节差分构建特征矩阵。该特征矩阵按虚拟地址顺序填充，

将恶意软件特征转换为机器易于理解和处理的RGB图像。具体流程如图3所示。

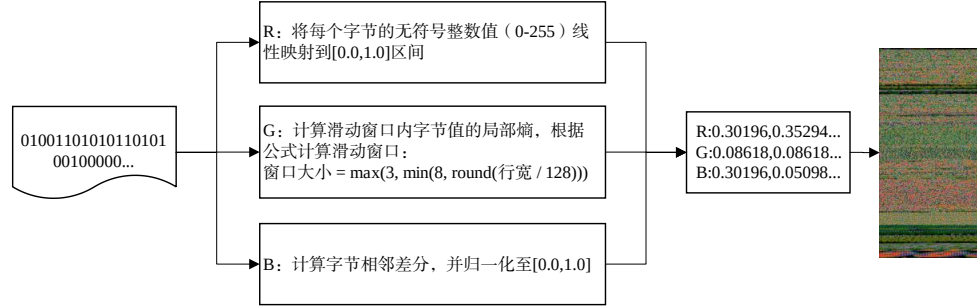


Fig. 3 The process of generating RGB images by malware

图3 恶意软件生成RGB图像的流程

R通道采用直接二进制码转换方法,以8位无符号整数组为单位读取给定文件的二进制文件。由于文件大小存在差异,将其转换为具有相同宽度的二维矩阵时,较小文件在扩展过程中可能会导致其原有结构损失^[23]。本文根据文件大小选取不同行宽进行数据填充来构建特征矩阵。文献[9]中提供了针对不同文件大小的推荐图像宽度,具体如表2所示。将每个字节的无符号整数值[0, 255]线性映射到[0.0, 1.0]区间,并保留原始数据分布特征。

Table 2 Correspondence among line width, entropy window and malicious file size

表2 行宽、熵窗口与恶意文件大小的对应关系

文件大小/KB	宽度	熵窗口
<10	32	3
10~30	64	3
30~60	128	3
60~100	256	3
100~200	384	5
200~500	512	5
500~1 000	768	7
>1 000	1 024	9

恶意软件通常会使用混淆、加密等技术,使用局部熵可以帮助区分恶意软件的不同数据区域,如数据段、代码段、加密段等。因此G通道使用局部熵来突出恶意软件的局部复杂度。使用不同的熵窗口应对不同文件大小,以防止过拟合和欠拟合。熵窗口的计算公式为:

$$\text{窗口大小} = \max(3, \min(8, \text{round}(\text{Width}/128))) \quad (9)$$

式中:round表示四舍五入取整函数。

最终将熵值归一化至[0.0, 1.0]。局部熵的计算公式为:

$$H = -\sum_{i=0}^{255} p_i \log_2 p_i \quad (10)$$

式中: $p_i = \frac{\text{count}(i)}{N}$, N 为窗口内字节数。

B通道使用相邻字节差分处理后的数据。首先计算相邻字节之间的差异,反映二进制文件字节突变位置,将计

算好的一维差分数组归一化至[0.0, 1.0]。字节相邻差分的计算公式为:

$$D(x) = \begin{cases} |Byte(0)|, & x = 0 \\ |Byte(x) - Byte(x-1)|, & x > 0 \end{cases} \quad (11)$$

最终将三通道堆叠为RGB彩色图片。研究表明,对于相同的图像预处理方法,不同的恶意软件图像输入大小会在一定程度上影响模型性能^[24]。本文模型为多尺度输入,输入图片尺寸为112×112、160×160和224×224。

2.2 双通道注意力机制

为了更好地提取转化后图像中局部特征保留的恶意软件信息,本文在EfficientNet B3的MBConv层引入双通道注意力机制,以增加关键字序列区域的权重。改进后的MBConv层首先使用 1×1 卷积对输入特征的通道维度扩展。设输入特征图为 $X_{in} \in \mathbb{R}^{H \times W \times C_{in}}$,通过 $1 \times 1 \times (t \cdot C_{in})$ 卷积核进行卷积操作,将其通道维度扩大至 $t \cdot C_{in}$,其中 $t = 6$,为扩展因子;“ $\cdot$ ”表示标量乘法。扩展卷积的计算公式为:

$$X_{\text{expand}} = \text{Conv}_{1 \times 1}(X_{in}) \quad (12)$$

对扩展后的特征图执行 $k \times k$ 深度可分离卷积,计算公式为:

$$X_{DW} = \text{DepthwiseConv}_{k \times k}(X_{\text{expand}}) \quad (13)$$

引入新的通道注意力机制,将每个通道的特征简化为单个数值。使用两个 1×1 卷积层(降维至 $C/4$ 再恢复)配合Swish激活,生成通道注意力权重。计算公式为:

$$z = \text{GAP}(X_{DW}) \quad (14)$$

$$s_{CA} = \sigma(W_2 \delta_1(W_1 z)) \quad (15)$$

$$X_{CA} = s_{CA} \otimes X_{DW} \quad (16)$$

式中:GAP为全局平均池化; $W_1 \in \mathbb{R}^{C \times (C/4)}$ 和 $W_2 \in \mathbb{R}^{(C/4) \times C}$ 分别表示通道压缩层权重和恢复层权重; δ_1 为Swish激活函数; σ 为Sigmoid激活函数; $\otimes$ 表示通道级乘法。

其次进行压缩和激励网络(Squeeze-and-Excitation Networks, SE)计算,通过GAP将 X_{CA} 压缩为通道描述符 z' ,

使用两个FC(先降维至C/4再恢复至C)学习各通道的重要性权重 s_{SE} ,再利用通道乘法实现特征重校准。SE模块的计算公式为:

$$z' = GAP(X_{CA}) \quad (17)$$

$$s_{SE} = \sigma(W_4 \delta_2(W_3 z')) \quad (18)$$

$$X_{SE} = s_{SE} \otimes X_{CA} \quad (19)$$

式中: $W_3 \in \mathbb{R}^{C \times C/8}$ 和 $W_4 \in \mathbb{R}^{C/8 \times C}$ 分别表示通道压缩层权重和恢复层权重; δ_2 为ReLU激活函数。

融合两个注意力机制,公式为:

$$X_{DUL} = (X_{CA} \oplus s_{SE}) \otimes X_{CA} \quad (20)$$

最后使用 1×1 卷积将通道维度从扩展后的 $t \cdot C_{in}$ 压缩回原始尺寸 C_{out} ,当满足步长为1且输入输出通道数相等的条件时,执行残差连接操作。计算公式为:

$$X_{proj} = Conv_{1 \times 1}(X_{DUL}) \quad (21)$$

$$X_{out} = \begin{cases} X_{in} + DropConnect(X_{proj}), & \text{残差连接} \\ X_{proj}, & \text{其他} \end{cases} \quad (22)$$

2.3 金字塔特征融合方法

恶意图像经过改进后的MBConv层处理后输出深层图像特征,使用金字塔特征融合方法对多尺度多层次特征进行拼接,实现特征互补,从而提高图像特征表达能力。融合流程见图4。

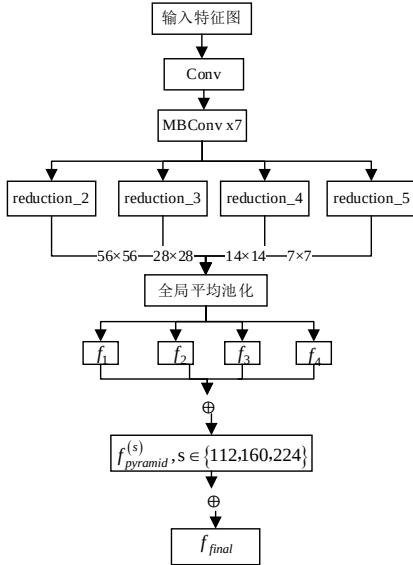


Fig. 4 Flow of pyramid feature fusion method

图4 金字塔特征融合方法流程

针对输入的多尺度恶意软件图像,使用改进的EfficientNetB3模型输出不同层级的特征图。本文提取reduction_2、reduction_3、reduction_4和reduction_5 4个层级特征,记为 $F_i \in \mathbb{R}^{H_i \times W_i \times C_i}$, $i \in \{2, 3, 4, 5\}$,其中 H_i 为特征图的高度, W_i 为特征图的宽度, C_i 为通道数。对每个特征图进行GAP操作,将每个特征图的每个通道压缩为一个标量值,使不同空间尺寸的特征图转换为相同长度的特征向量。计算公式为:

$$f_i = GAP(F_i) = \left[\frac{1}{H_i \times W_i} \sum_{h=1}^{H_i} \sum_{w=1}^{W_i} F_i(h, w, c) \right]_{c=1}^{C_i} \quad (23)$$

将4个特征向量按照从深层到浅层的顺序拼接成一个长特征向量。计算公式为:

$$f_{pyramid} = f_5 \oplus f_4 \oplus f_3 \oplus f_2 \quad (24)$$

式中: $f_{pyramid} \in \mathbb{R}^{C_2 + C_3 + C_4 + C_5}$; $\oplus$ 表示向量拼接操作。

对于不同的尺度 s ,得到该尺度图像的特征向量 $f_{pyramid}^{(s)}$ 。拼接3个尺度的金字塔特征向量,形成最终特征表示,包含同一恶意软件转化后的图像在不同尺度和层级上的特征信息。公式为:

$$f_{final} = f_{pyramid}^{(224)} \oplus f_{pyramid}^{(160)} \oplus f_{pyramid}^{(112)} \quad (25)$$

式中: $f_{final} \in \mathbb{R}^{3 \times (C_2 + C_3 + C_4 + C_5)}$ 。

2.4 IPDBL模型

为使模型适配大规模数据集,本文改进PDBL模型,使用IPCA代替PCA对特征矩阵进行降维,主要步骤为:

(1)IPCA降维。使用IPCA对从金字塔特征融合得到的高维特征进行降维。首先,计算当前批次的均值 μ_b 和协方差 Σ_b ,根据批次大小 n_b 和累积样本量 n_t 更新全局均值 μ_t 和协方差 Σ_t ;其次,对更新后的协方差矩阵进行特征分解 $\Sigma_t = V \Lambda V^T$,根据样本总量动态选择目标维度;最后,将原始特征 X 投影到前 k 个主成分上,得到降维特征 Z 。计算公式为:

$$\mu_t = \mu_{t-1} + \frac{n_b}{n_t} (\mu_b - \mu_{t-1}) \quad (26)$$

$$\Sigma_t = \Sigma_{t-1} + \frac{n_b}{n_t} (\Sigma_b - \Sigma_{t-1}) \quad (27)$$

$$Z = (X - \mu_t) V[:, :k] \quad (28)$$

(2)双正则伪逆求解。构建双重正则化优化问题求解权重,计算公式为:

$$W^* = (Z^T Z + \lambda I + \alpha \Lambda_k)^{-1} Z^T Y \quad (29)$$

式中: $Z^T Z$ 为特征的自相关矩阵; λI 为岭回归正则项($\lambda = 10^{-4}$); $\alpha \Lambda_k$ 为特征值正则项($\alpha = 10^{-3}$); W^* 为通过求解得到的最优权重。

(3)新样本预测。得到最优权重后对新样本进行预测,对于一个新的测试样本 $x_{test} \in \mathbb{R}^D$,使用训练时得到的模型进行降维,计算公式为:

$$z_{test} = (x_{test} - \mu) V_k \quad (30)$$

使用训练好的权重矩阵进行预测,得到预测值,计算公式为:

$$\hat{y}_{test} = z_{test}^T W^* \quad (31)$$

3 实验方法与结果分析

3.1 实验环境

实验使用Windows系统,硬件配置为12th Gen Intel (R) Core (TM) i9-12900K@3.20 GHz*64, 32GB RAM,

NVIDIA GeForce RTX 3090 GPU, Python 3.9 编译环境和 PyTorch 1.12 深度学习框架。

3.2 数据集

本文在 3 个数据集上对模型进行验证。数据集一为 Vision Research Lab 团队在 2011 年收集发布的 Maling 数据

集^[7], 文献[11-13, 21]等均使用该数据集验证模型的有效性。该数据集包含 25 个恶意代码家族的 9 435 个样本数据, 涵盖 Allapple.A、Fakerean、Yuner.A 等恶意软件家族。具体家族数量分布如图 5 所示。

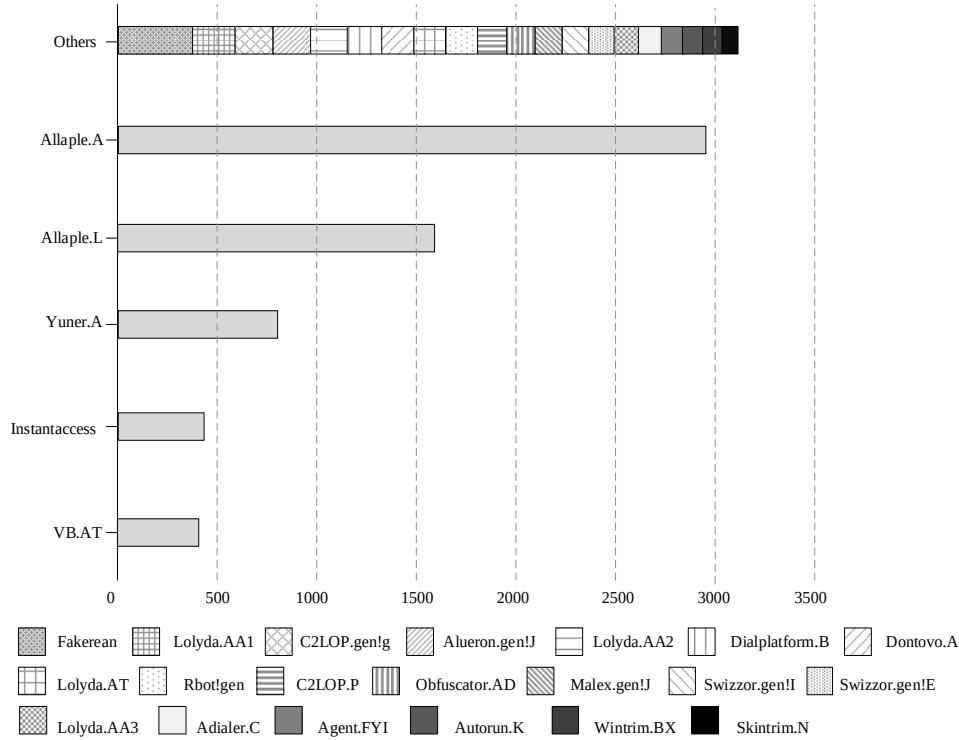


Fig. 5 Maling dataset distribution

图 5 Maling 数据集分布

数据集二为从 VirusTotal 平台、MalwareBazaar 平台中获取的恶意软件数据组成的自建数据集, 包含 30 多个恶意软件家族样本的 917 289 个数据, 包括蠕虫、木马、漏洞、后门和广告软件等。数据集详细信息如表 3 所示。该数据集存在显著的类别不平衡问题, 即某些高频类别的样本数量超过 100 000 条 (如木马 Trojan、病毒 Virus), 而低频类别的样本数量不足 500 条 (如工具 Tool、构造器 Constructor、拒绝服务 DoS 等)。在极端类别之间, 样本量差异最高可达 1 000 倍以上。为缓解类别不平衡对模型性能可能带来的潜在偏差, 本文采用一种基于类别筛选的数据平衡策略。具体而言, 选取样本量相对充足且分布较为均衡的 15 个主要类别, 其样本数量范围为 15 693~365 045 条, 最大样本量 (木马 Trojan, 365 045 条) 与最小样本量 (广告软件 Adware, 15 693 条) 的比值约为 23:1, 其他 13 个类别间的样本量比值均小于 5:1。通过排除样本量极少或适中的边缘类别来缩小不同类别之间的样本数量差异, 确保数据集的代表性, 为模型提供一个更加均衡的学习环境, 从而降低分类器对多数类的倾向性。

数据集三为巴塞特佩大学计算机工程系多媒体信息实验室与 COMODO 公司在 2019 年收集发布的 Malevis 数据集^[25]。使用其中 25 个恶意代码家族的 12 394 个样本数

据进行实验, 涵盖 Adposhel、Agent、Allapple 等恶意软件家族。

3.3 评价标准

使用准确度 (Accuracy)、精确度 (Precision)、召回率 (Recall)、F1 分数 (F1-score)、马修斯相关系数 (Matthews Correlation Coefficient, MCC) 和微软公开数据集的 Private Score 对模型性能进行评估。计算公式分别为:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (32)$$

$$Precision = \frac{TP}{TP + FP} \quad (33)$$

$$Recall = \frac{TP}{TP + FN} \quad (34)$$

$$F1 - score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (35)$$

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \quad (36)$$

$$PrivateScore = \frac{1}{N_{private}} \sum_{i=1}^{N_{private}} (y_i, \hat{y}_i) \quad (37)$$

式中: TP (True Positive) 表示实际为正例, 预测为正例的数量; TN (True Negative) 表示实际为负例, 预测为负例的数量; FP (False Positive) 表示实际为负例, 预测为正例的数

Table 3 Details of dataset II

表 3 数据集二详细信息

样本名称	样本数量
Adware	15 693
Backdoor	28 480
BrowserModifier	57 914
Constructor	7
DDoS	1 266
Dialer	1 421
DoS	37
Exploit	23 593
Flooder	633
HackTool	1 142
Joke	42
Misleading	174
MonitoringTool	239
Program	114
PWS	20 617
Ransom	5 066
Rogue	1 929
SettingsModifier	24
SoftwareBundler	26 663
Spammer	97
Spyware	267
SupportScan	225
Tool	5
Trojan	365 045
TrojanClicker	36 855
TrojanDownloader	76 471
TrojanDropper	47 793
TrojanProxy	304
TrojanSpy	4 352
VirTool	14 164
Virus	141 582
Worm	45 075

量;FN(False Negative)表示实际为正例,预测为负例的数量; $N_{private}$ 为自建测试集的样本数量(占总测试集的 70%); $\mathcal{L}$ 为任务相关的损失函数; y_i 为样本 i 的真实标签; $\hat{y}_i$ 为模型对样本 i 的预测值。

3.4 可视化结果分析

本文将三通道可视化方法应用于自建数据集中,选取 4 类最常见的恶意软件家族,其可视化结果如图 6 所示。可以看出,不同家族在颜色分布、纹理特征等方面有很大差异,同种家族间只有细微差别,因此可以使用可视化方式对恶意软件进行检测。

3.5 比较实验

3.5.1 可视化方法比较

为探究不同图像化表征方法对恶意软件检测性能的影响,本文选取 PDBL 作为基线模型,分别采用 EfficientNetB3 与 shuffleNet 作为分类模型,使用真实数据集分别生成 RGB 图像和灰度图像进行模型训练与评估。以微软公开数据集提供的 Private Score 作为核心评价指标,分析不同图像处理方式对检测性能的影响。为确保实验结果的公平性,PDBL+EfficientNetB3 和 PDBL+ShuffleNetV2 模型

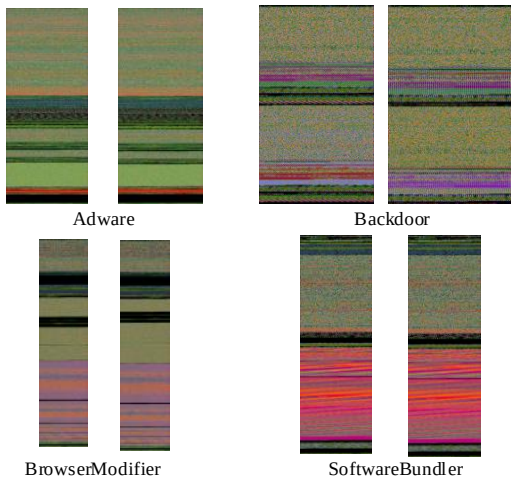


Fig. 6 Dataset visualization results

图 6 数据集可视化结果

采用相同的参数设置:输入分辨率(112×112、160×160、224×224)、批量大小(32)、训练周期(100)。如表 4 所示,本文图像处理方法在所有实验配置下均取得最低损失值,说明其能够提供更具有判别力的恶意软件特征信息,进而提升模型的学习能力与检测性能。

Table 4 Private Score comparison of different processing methods and classification models

表 4 不同处理方式和分类模型的 Private Score 比较

图像处理方法	PDBL+EfficientNetB3	PDBL+ShuffleNet
灰度图 ^[10]	2.367 29	2.673 82
RGB ^[5]	2.264 59	2.574 87
本文方法	1.946 00	2.045 38

3.5.2 不同方法性能比较

为验证 EffNet-IPDBL 模型对恶意软件的检测性能,本文选择目前基于可视化分析的恶意软件检测方法进行比较,分别为 AlexNet^[5]、VGGNet^[13]、3D-VGG-16^[26]。为确保实验结果的公平性与可靠性,所有实验均在统一的软硬件环境下进行,使用相同的超参数设置(学习率 1×10^{-4} 、批量大小 64、SGD 优化器,交叉熵损失函数)进行训练与评估。对于参数量大且计算复杂的 3D-VGG-16 模型,通过调整梯度累积步数等方式来保证训练条件的一致性。

不同方法在 3 个数据集上的表现如表 5、表 6 和表 7 所示。可以看出,在自建数据集上,EffNet-IPDBL 在 Accuracy、Recall 和 F1-score 等指标上均显著优于 VGGNet、AlexNet 和 3D-VGG-16,表明该模型在处理类别不平衡数据时具有更强的鲁棒性,这主要得益于 IPDBL 模块对冗余特征的有效筛选与关键信息增强。在 Malimg 数据集上,EffNet-IPDBL 在 Accuracy 指标上取得最优表现,但在 Precision、Recall 和 F1-score 上略低于 AlexNet。其原因可能是 Malimg 数据集的类别分布相对均衡,使得传统基于反向传播的深度网络在该场景下仍具备较强的竞争力。在 Mal-Vis 数据集上,EffNet-IPDBL 在所有评价指标上均优于对照模型,验证了所引入的双通道注意力机制能更有效地构

建全局空间特征,提升模型的判别能力。

Table 5 Comparison of different methods on self-built datasets

表5 不同方法在自建数据集上的比较 (%)

方法	Accuracy	Precision	Recall	F1-score
VGGNet	87.46	86.74	76.52	77.87
AlexNet	86.59	86.00	77.33	79.01
3D-VGG-16	84.14	78.52	77.89	77.36
EffNet-IPDBL	88.35	85.97	83.07	83.45

Table 6 Comparison of different methods on Maling dataset

表6 不同方法在Maling数据集上的比较 (%)

方法	Accuracy	Precision	Recall	F1-score
VGGNet	96.16	96.00	96.50	96.20
AlexNet	97.80	97.80	98.00	97.80
3D-VGG-16	96.14	97.00	96.00	96.00
EffNet-IPDBL	98.69	96.59	96.31	96.39

Table 7 Comparison of different methods on MalVis dataset

表7 不同方法在MalVis数据集上的比较 (%)

方法	Accuracy	Precision	Recall	F1-score
VGGNet	94.79	94.84	94.69	94.71
AlexNet	94.06	93.89	93.94	93.82
3D-VGG-16	93.68	93.64	93.53	93.51
EffNet-IPDBL	95.82	95.98	95.76	95.83

3.5.3 不同方法效率比较

为验证本文方法的硬件效率,引入每秒浮点运算次数(FLOPs, Floating Point Operations)作为衡量结构复杂度的指标,同时将推理时延作为衡量检测速度的指标,使用统一的软硬件环境。实验在Maling数据集上进行,结果如表8所示。可以看出,与VGGNet、AlexNet和3D-VGG-16相比,本文方法的参数量和FLOPs均为最低。然而,本文方法在推理时延方面略逊于VGGNet和AlexNet,原因在于其采用的EfficientNet网络在推理过程中涉及大量数据读写操作。考虑到本文方法对检测精度的提升效果,该种程度的时延可以接受。

Table 8 Efficiency evaluation results of different methods

表8 不同方法效率评估结果

方法	FLOPs/G	推理时延/ms	参数量/M
VGGNet	15.47	9.08	138.35
AlexNet	0.69	3.00	61.10
3D-VGG-16	81.37	320.02	784.34
EffNet-IPDBL	0.10	22.01	0.09

3.6 消融实验

在3个数据集上开展消融实验,分别验证双重注意力机制、金字塔特征融合和IPCA的作用,结果如表9、表10和表11所示。可以看出,任一关键模块的移除均会对模型整体性能产生不同程度的负面影响,验证了各模块在模型结构中的必要性与有效性。具体而言,去除IPCA模块后,模型在自建、Maling和MalVis数据集上的MCC指标分别下降了3.14%、1.14%和1.94%,同时准确率、精确率和召回率等指标亦呈现出整体下滑趋势,说明IPCA模块在特征降维与关键信息保持方面发挥了重要作用。当移除双

通道注意力机制时,模型在3个数据集上的准确率、精确率和召回率均出现明显下降,其中自建数据集上的F1-score约降低0.44%,表明该模块能有效强化图像特征表达,提升模型对恶意软件的判别能力。移除金字塔特征融合模块后,模型在Maling和MalVis数据集上的精确率分别下降0.89%和1.20%。尽管在自建数据集上的召回率略有提升,但综合性能仍然受到削弱。

Table 9 Ablation experiment results on self-constructed dataset

表9 在自建数据集上的消融实验结果 (%)

IPCA	双注意 力机制	金字塔特 征融合	Accuracy	Precision	Recall	F1- score	MCC
√	√	—	88.06	83.39	85.31	83.06	86.39
√	—	√	87.94	83.31	85.10	83.01	86.26
—	√	√	85.62	78.76	84.52	77.67	83.58
√	√	√	88.35	85.97	83.07	83.45	86.72

Table 10 Ablation experiment results on Maling dataset

表10 在Maling数据集上的消融实验结果 (%)

IPCA	双注意 力机制	金字塔特 征融合	Accuracy	Precision	Recall	F1- score	MCC
√	√	—	98.48	95.70	95.76	95.61	98.21
√	—	√	98.48	95.61	95.70	95.76	98.21
—	√	√	97.72	92.15	92.64	92.37	97.33
√	√	√	98.69	96.59	96.31	96.39	98.47

Table 11 Ablation experiment results on MalVis dataset

表11 在MalVis数据集上的消融实验结果 (%)

IPCA	双注意 力机制	金字塔特 征融合	Accuracy	Precision	Recall	F1- score	MCC
√	√	—	94.56	94.78	94.51	94.57	94.34
√	—	√	94.20	94.39	94.12	94.19	93.97
—	√	√	93.96	94.20	93.87	93.95	93.71
√	√	√	95.82	95.98	95.76	95.83	95.65

4 结语

本文针对恶意软件检测场景下特征表达能力不足与检测性能受限的问题,构建了一种融合多尺度特征的宽度—深度学习检测方法。该方法通过多层级特征融合机制对恶意软件图像特征进行建模,并结合适用于大规模数据处理的特征压缩策略,在保证检测性能的同时提升了模型对不同输入规模的适应能力。实验结果表明,所提方法在多个数据集上具有良好且稳定的检测效果,验证了其在恶意软件识别任务中的有效性与可行性。

尽管本文方法在检测性能和系统鲁棒性方面取得了一定成效,但仍存在进一步完善的空间:一方面,相关实验主要基于离线数据集完成,尚未在真实恶意软件检测系统或实际运行环境中开展系统级验证;另一方面,当前模型在推理阶段仍具有一定计算开销,在资源受限或实时检测场景中的执行效率有待进一步提升。针对上述不足,后续研究将从实验验证与模型优化两个方面展开:①可引入更贴近真实场景的数据来源与测试设置,对模型在复杂运行

环境下的表现进行补充验证,以全面评估其稳定性与泛化能力;②围绕模型结构与计算流程开展优化研究,通过引入更高效的特征表示与推理加速策略来降低计算开销,提升其在实时或资源受限条件下的检测效率。

参考文献:

- [1] China Internet Network Information Center. The 55th statistical report on China's Internet development [EB/OL]. <https://www.cnnic.net.cn/n4/2025/0117/c88-11229.html>.
中国互联网络信息中心. 第55次《中国互联网络发展状况统计报告》[EB/OL]. <https://www.cnnic.net.cn/n4/2025/0117/c88-11229.html>.
- [2] Kaspersky Security Network. IT threat evolution in Q1 2025. Non-mobile statistics [EB/OL]. <https://securelist.com/malware-report-q1-2025-pc-iot-statistics/116686/>.
- [3] YAO B R, ZHANG N. Research on static detection method of Android malware based on combinatorial features[J]. Software Guide, 2024, 23(2): 129-134.
姚斌荣, 张娜. 基于组合特征的安卓恶意软件静态检测方法研究[J]. 软件导刊, 2024, 23(2): 129-134.
- [4] YAN P, TAN S Q, HUANG J W. A survey: malware detection based on deep learning[J]. Journal of Cyber Security, 2025, 10(3): 98-119.
严沛, 谭舜泉, 黄继武. 基于深度学习的恶意代码检测综述[J]. 信息安全学报, 2025, 10(3): 98-119.
- [5] JIANG K L, BAI W, ZHANG L, et al. Malicious code detection based on multi-channel image deep learning[J]. Journal of Computer Applications, 2021, 21(4): 1142-1147.
蒋考林, 白玮, 张磊, 等. 基于多通道图像深度学习的恶意代码检测[J]. 计算机应用, 2021, 21(4): 1142-1147.
- [6] NIKOLOPOULOS S D, POLENAKIS I. A graph-based model for malware detection and classification using system-call groups[J]. Journal of Computer Virology and Hacking Techniques, 2017, 13(1): 29-46.
- [7] ZHAO Y C, MU D J, DAI H. Design and implementation of dual malware detection system[J]. Computer Technology and Development, 2013, 23(10): 111-114.
赵云程, 慕德俊, 戴航. 双重恶意代码检测系统的设计与实现[J]. 计算机技术与发展, 2013, 23(10): 111-114.
- [8] JI T T, FANG B X, CUI X, et al. Research on deep learning-powered malware attack and defense techniques[J]. Chinese Journal of Computers, 2021, 44(4): 669-695.
冀甜甜, 方滨兴, 崔翔, 等. 深度学习赋能的恶意代码攻防研究进展[J]. 计算机学报, 2021, 44(4): 669-695.
- [9] NATARAJ L, KARTHIKEYAN S, JACOB G, et al. Malware images: visualization and automatic classification[C]//Proceedings of the 8th International Symposium on Visualization for Cyber Security, 2011: 1-7.
- [10] DING Y, LI L. A method for detecting malicious code base on feature encoding technology[J]. Computer Technology and Development, 2021, 31(1): 131-136.
丁应, 李琳. 一种基于特征编码技术的恶意代码检测方法[J]. 计算机技术与发展, 2021, 31(1): 131-136.
- [11] SINGH A, HANDA A, KUMAR N, et al. Malware classification using image representation[C]//Cyber Security Cryptography and Machine Learning: Third International Symposium, 2019: 75-92.
- [12] ZHANG X L, CHAI Y Y, WU K H, et al. A malware detection method based on incremental learning[J]. Computer and Digital Engineering, 2024, 52(7): 2141-2145, 2220.
张晓良, 柴艳玉, 吴克河, 等. 一种基于增量学习的恶意代码检测方法[J]. 计算机与数字工程, 2024, 52(7): 2141-2145, 2220.
- [13] WANG B, CAI H H, SU Y. Classification of malicious code variants based on VGGNet[J]. Journal of Computer Applications, 2020, 40(1): 162-167.
王博, 蔡弘昊, 苏旸. 基于VGGNet的恶意代码变种分类[J]. 计算机应用, 2020, 40(1): 162-167.
- [14] MENG Z, ZHANG J, GUO J, et al. Detecting Android malware by visualizing App behaviors from multiple complementary views[J]. IEEE Transactions on Information Forensics and Security, 2025, 20: 2915-2929.
- [15] HE Y, KANG X, YAN Q, et al. ResNeXt+: attention mechanisms based on ResNeXt for malware detection and classification[J]. IEEE Transactions on Information Forensics and Security, 2023, 19: 1142-1155.
- [16] YADAV P, MENON N, RAVI V, et al. EfficientNet convolutional neural networks-based Android malware detection[J]. Computers & Security, 2022, 115: 102622.
- [17] CHEN F, CAO X M, WANG S H. Android malware detection method based on feature image generation[J]. Computer Technology and Development, 2023, 33(6): 125-132.
陈非, 曹晓梅, 王少辉. 基于特征图像生成的Android恶意软件检测方法[J]. 计算机技术与发展, 2023, 33(6): 125-132.
- [18] LI S C, WANG J, SONG Y F, et al. TriCh-LKRepNet: a large kernel convolutional malicious code classification network for structure reparameterisation and triple-channel mapping[J]. Acta Electronica Sinica, 2024, 52(7): 2331-2340.
李思聪, 王坚, 宋亚飞, 等. TriCh-LKRepNet: 融合三通图映射与结构重参数化的大核卷积恶意代码分类网络[J]. 电子学报, 2024, 52(7): 2331-2340.
- [19] LU T, WANG J. DOMR: toward deep open-world malware recognition[J]. IEEE Transactions on Information Forensics and Security, 2023, 19: 1455-1468.
- [20] CHEN C L P, LIU Z L. Broad learning system: an effective and efficient incremental learning system without the need for deep architecture[J]. IEEE Transactions on Neural Networks and Learning Systems, 2018, 29(1): 10-24.
- [21] LIN J, HAN G, PAN X, et al. PDBL: improving histopathological tissue classification with plug-and-play pyramidal deep-broad learning[J]. IEEE Transactions on Medical Imaging, 2022, 41(9): 2252-2262.
- [22] TAN M X, LE Q V. EfficientNet: rethinking model scaling for convolutional neural networks[C]//36th International Conference on Machine Learning, 2019: 6105-6114.
- [23] ZHANG S H, HU C D, WANG L H, et al. Malware detection and classification based on GHM visualization and deep learning[J]. Journal of Information Security Research, 2024, 10(3): 216-222.
张淑慧, 胡长栋, 王连海, 等. 基于GHM可视化和深度学习的恶意代码检测与分类[J]. 信息安全研究, 2024, 10(3): 216-222.
- [24] XUAN B N, LI J, SONG Y F, et al. Malicious code classification method based on improved MobileNetV2[J]. Journal of Computer Applications, 2023, 43(7): 2217-2225.
轩勃娜, 李进, 宋亚飞, 等. 基于改进 MobileNetV2 的恶意代码分类方法[J]. 计算机应用, 2023, 43(7): 2217-2225.
- [25] BOZKIR A S, CANKAYA A O, AYDOS M. Utilization and comparison of convolutional neural networks in malware recognition[C]//2019 27th Signal Processing and Communications Applications Conference, 2019: 1-4.
- [26] AL-KHATER W, AL-MADEED S. Using 3D-VGG-16 and 3D-Resnet-18 deep learning models and FABEMD techniques in the detection of malware[J]. Alexandria Engineering Journal, 2024, 89: 39-52.
(责任编辑: 尹晨茹)