

基于多维特征融合的恶意代码分类方法

冯帅, 高见*

(中国人民公安大学信息安全学院, 北京 100038)

摘要: 在计算机安全领域, 恶意代码防护一直是重要的研究课题。随着计算机技术的快速发展, 恶意代码的种类和形式不断演变, 传统特征工程方法在处理复杂恶意样本时特征维度单一, 致使表征能力不足, 无法精准识别各类恶意代码。其他基于特征融合的恶意代码分类方法在特征提取过程中依赖专家经验进行手工特征设计, 而多模态深度学习模型可解释性不足, 计算开销大。针对上述问题, 提出一种新的特征融合方法, 该方法应用于 Windows PE 文件的恶意代码分类, 通过整合行为特征、结构特征及纹理特征, 并采用 LightGBM 作为分类器完成对恶意代码的分类。实验结果表明, 该方法在 Microsoft 恶意软件分类挑战赛数据集上的测试准确率为 99.90%, 对数损失为 0.005 7, 在 Bazaar 数据集上的测试准确率为 98.97%, 对数损失为 0.042 0。所提方法能够全面、准确地表征恶意代码, 其通过融合多维特征, 为恶意代码检测提供了一种有效的解决方案, 具有重要的理论意义和实际的应用价值。

关键词: 恶意代码; 特征融合; 特征工程; 多模态; LightGBM

中图分类号: TP309.5

文献标志码: A

DOI: 10.19678/j.issn.1000-3428.0252241

Malicious Code Classification Method Based on Multidimensional Feature Fusion

FENG Shuai, GAO Jian*

(College of Information Network Security, People's Public Security University of China, Beijing 100038, China)

【Abstract】 Malicious code protection is an important research topic in computer security. With the rapid development of computer technology, the types and forms of malicious codes continue to evolve. Traditional feature engineering methods have a single feature dimension when dealing with complex malicious samples, resulting in insufficient representation ability and inability to accurately identify various types of malicious code. Malicious code classification methods based on feature fusion rely on expert experience in manual feature design during feature extraction, whereas multimodal deep learning models have insufficient interpretability and a high computational overhead. A new feature fusion method is proposed to address the aforementioned issues and applied to the classification of malicious code in Windows PE files. By integrating behavioral, structural, and textural features, LightGBM is used as a classifier to complete the classification of the malicious code. The experimental results show that the testing accuracy of this method on the Microsoft Malicious Software Classification Challenge dataset is 99.90% with a logarithmic loss of 0.005 7. The testing accuracy on the Bazaar dataset is 98.97%, with a logarithmic loss of 0.042 0. The proposed method can comprehensively and accurately characterize malicious code. By integrating multidimensional features, the method provides an effective solution for malicious code detection, which has important theoretical significance and practical application value.

【Key words】 malicious code; feature fusion; feature engineering; multimodal; LightGBM

0 引言

自计算机诞生以来, 恶意代码便成为网络安全防护领域的重大威胁。伴随着计算机应用场景的持续拓展, 恶意代码的种类与形式不断迭代更新。传统的特征工程仅依赖单一维度特征, 对恶意样本的表征能力有限, 致使基于单一特征的恶意代码分类方法准确率难以提升。

在基于融合特征的恶意代码分类方法中, 尽管

部分表现出色的方法在特征提取阶段依赖手工设计的特征, 在特定数据集中提高了分类准确率, 但是在面对新型恶意代码时, 这些方法往往表现欠佳, 并且高度依赖专家经验。此外, 多模态深度学习框架虽在恶意代码分类中有所应用, 但随着输入数据维度的增加, 其网络结构更复杂, 计算开销更大, 且模型可解释性欠佳。

为解决上述问题, 本文提出一种全新的基于融合特征的恶意代码分类方法。在特征提取阶段, 全

收稿日期: 2025-03-18 修回日期: 2025-04-29

基金项目: 中国人民公安大学中央基本科研业务费项目(2024JKF17)。

通信作者 E-mail: * gaojian@ppsuc.edu.cn

面提取恶意样本的行为特征、结构特征以及纹理特征;在特征融合阶段,采用早期融合策略整合多源特征;最后,运用 LightGBM 算法进行分类。本文在 Microsoft 恶意软件分类挑战赛数据集以及 Bazaar 数据集上展开实验,结果表明,该恶意代码检测方法在不依赖手工设计特征的情况下,能用更低的计算开销实现更高的恶意代码家族分类准确率以及更低的对数损失,且在不同数据集上均表现良好。

1 相关工作

在恶意代码检测与对抗的研究领域,众多研究者针对日益复杂且快速演变的恶意软件环境提出了多样化的解决方案。学者们致力于改进特征工程方法,从不同维度精准刻画恶意代码的行为模式和结构特性,同时探索和优化各类机器学习及深度学习模型,以实现更高效准确的分类识别,从而为后续研究提供坚实的理论支撑和实践依据。

当前在恶意代码分析领域,特征提取技术主要围绕两种核心途径展开:一是对恶意代码的原始二进制字节流进行深入挖掘,通过统计分析、序列模式识别、恶意代码可视化等手段捕捉其独特的内在规律和编码特征;二是运用反汇编技术将二进制文件转换成可读性更强的汇编代码,并在此基础上提炼高级抽象特征,如函数调用序列、API 调用行为、控制流图(CFG)结构等。这两种特征提取方法分别从低层构造细节到高层执行逻辑,构建全面且具有鉴别力的特征集,为后续分类模型的训练和恶意代码检测提供了强有力的支持。基于多模态的恶意代码分类方法包括多模态深度学习模型以及在特征提取阶段融合多模态特征等思路,通过深入研究相关文献的前沿成果,该领域的核心发展趋势得以梳理。

1.1 基于二进制字节流的特征工程

针对恶意代码的原始二进制字节流,文献[1]首次提出将恶意代码二进制样本转化为图像,并使用卷积神经网络(CNN)进行恶意代码分类。该方法将图像处理技术引入恶意代码分类领域,为恶意代码分类提供了全新思路。文献[2]提出了基于微调 CNN 的恶意软件分类方法(IMCFN),用于检测和识别不同恶意软件家族及其变种。IMCFN 方法将原始恶意软件二进制文件转换为彩色图像,然后采用经过微调的 CNN 架构进行处理。文献[3]提出了基于马尔可夫图像和深度学习的字节级恶意软件分类方法。传统的基于灰度图像和深度学习的方法,在统一图像尺寸过程中

由于信息丢失和冗余问题导致分类准确性仍有待提升。为了克服这些问题,文献[3]利用马尔可夫转移概率矩阵构建马尔可夫图像,这样可以生成固定大小的恶意软件图像,并有效减少字节信息中的冗余部分,通过将恶意软件二进制文件转换为马尔可夫图像进行分类。

文献[4]提出了一种结合深度迁移学习和机器学习的混合框架,通过将可执行文件(PE)转换为彩色图像,使用微调的深度学习模型提取深度特征,并利用支持向量机(SVM)进行恶意软件检测,实现了比传统方法更高的准确性和效率。文献[5]提出了一种利用一维 CNN(1D CNN)对恶意软件可执行文件的位和字节级序列进行分类的方法,以避免将二进制代码转换为图像导致的结构扭曲问题。实验使用了 2 个基准数据集,结果表明,所提出的 1D CNN 模型相较于现有的二维 CNN(2D CNN)恶意软件分类模型具有更好的性能,通过提供更小的位/字节级序列重采样,减少了计算成本。文献[6]提出了一种基于胶囊网络架构并带有优化卷积层的新型恶意软件检测方法(MalCaps),该方法通过去除池化层并引入胶囊层来克服传统 CNN 的局限性,在 Microsoft Malware Classification Challenge 数据集上实现了 99.34% 的测试准确率。

文献[7]提出了一种新的可视化方法,通过引入彩色标签框(CoLab)来标记 PE 文件的各个部分,从而在转换后的恶意软件图像中强调部分分布信息。此外,该文还构建了一种名为 MalCVS(使用 CoLab 图像、VGG16 和 SVM)的恶意软件分类方法。实验结果表明,在从 VX-Heaven、VirusShare 以及 Microsoft Malware Classification Challenge 数据集中收集的恶意软件上,MalCVS 能够以高精度将恶意软件分类到不同家族中,平均准确率分别为 96.59% 和 98.94%。文献[8]提出了一种基于 Bootstrap Aggregating 技术的集成胶囊网络模型,旨在减少特征工程的工作量并提高模型对数据的鲁棒性。该模型在 2 个广泛使用的高度不平衡数据集(Malimg 和 BIG2015)上进行了测试,并分别达到了 0.982 0 和 0.966 1 的最高 F1 值。此外,该模型使用的可训练参数量比文中最好的对比模型低 99.7%。

1.2 基于反汇编代码的特征工程

在恶意代码的反汇编文件的利用上,文献[9]构建了一个基于深度图卷积神经网络(DGCNN)的系统,将恶意软件程序表示为控制流图(CFG),通过对 CFG 进行属性化处理,并定义一系列与基本块相关

的数值型属性(如常量数、转移指令数、调用指令数等),将 CFG 转化为适用于机器学习的形式。利用 DGCNN 提取结构信息,有效地从控制流图中学习并实现恶意软件分类。文献[10]采用了一种混合方法提取恶意代码的特征并构建分类和聚类模型。对于特征工程部分,首先使用了 TF-IDF 算法来获取特征字段的频率和区分度特征,同时利用 n -gram 算法捕捉特征字段间的上下文关系。此外,研究者还引入了人工特征以增加特征多样性,并利用相关领域的经验信息提升模型性能。文献[11]提出了一种结合 Doc2vec 和 GRU 的恶意代码行为分类方法,通过正则表达式从汇编代码中提取 API 调用序列,将其向量化为 50 维向量,然后输入包含 3 个 GRU 层和 1 个注意力机制的分类模型,实现了高精度的恶意代码分类。文献[12]针对提取出的 API 调用序列长且难以理解的问题,提出了一种对 API 调用序列进行优化的方法。这种行为信息随后用于自动分类任务,并使用 fastText 和 BERT 算法来进行分类,实验结果显示,在分类和检测任务中,fastText 模型的表现优于 BERT 模型。文献[13]利用信息增益绝对特征选择(IGAFS)技术从预处理的数据集中选择与恶意软件最相关的特征,将所选特征输入到循环神经网络(RNN)中进行训练,实验结果表明,这种方法相比以往方法具有更好的性能。

1.3 基于多模态的特征融合方法

在基于多模态的恶意代码分类领域,文献[14]提出了多模态深度学习框架(HYDRA),它融合了 Windows API 函数调用、操作码序列和二进制字节序列这 3 种不同模态的数据特征,通过特定的神经网络架构设计,实现对恶意软件特征的全面捕捉。该文后续又探讨了手工特征与深度特征在恶意代码分类领域的关系^[15],并提出结合深度特征与手工特征的恶意代码分类方法,但是该方法依赖专家经验,且存在数据局限性,在面对新型恶意代码时未经优化的手工特征会出现表现能力不佳的情况。文献[16]提出了一种自动化的恶意代码分类系统,该系统通过静态分析从 PE 文件中提取 55 个静态特征,包括熵值和封装信息,同时将 PE 文件转换为图像数据。系统结合了决策树、随机森林、梯度提升算法用于 PE 特征分类,并使用 AlexNet 对图像数据进行分类,通过集成这些模型的结果,实现了 98.77% 的高准确率。文献[17]提出了一种静态恶意软件检测方法,从 27 920 个 Windows PE 恶意软件样本中提取了导入表、API 函数调用、PE 头以及

PE 区段统计信息作为特征,采用信息增益和主成分分析两种降维技术来进一步提高系统的性能。接着,使用集成学习技术进行分类。最终,该系统实现了 99.5% 的检测率和 0.47% 的错误率。

在安卓恶意代码分类任务中,文献[18]将 API 调用图(ACGs)与字节级图像表示相结合,同样取得了很好的分类效果。文献[19]提出了一种基于行为的恶意软件检测技术,通过在 Cuckoo 沙箱中建立动态分析环境来提取运行时特征,并使用文本挖掘技术处理可打印字符串,应用奇异值分解减少字符串特征维度、计算 Shannon 熵以及考虑文件操作、注册表键修改和网络活动的行为特征,最终使用集成机器学习算法构建恶意软件分类器,实现了 99.54% 的检测准确率。文献[20]提出了一种利用混合深度生成模型同时利用全局和局部特征来有效检测恶意软件变体的方法。该模型通过将恶意软件转换为图像来表示全局特征,通过预定义潜在空间从二进制代码序列提取局部特征。这两种特征被合并后输入恶意软件检测器中,通过使用这两种特征,该模型实现了 97.47% 的准确率,通过类激活映射(CAM)分析了哪部分的恶意代码影响了检测结果,并确认了虚拟恶意软件生成可以改善检测性能。

文献[21]选取图像 GIST 特征、操作码序列、字节统计值、文件结构等 4 种静态特征,采用主成分分析法和随机森林,利用贝叶斯优化策略增强模型,提升了检测性能。然而,这种方法对图像特征的性能利用不足,没能提取到图像的深层特征,也忽略了汇编图像对于融合特征表征能力的提升效果,在分类以及权重分配时效率也低于 LightGBM。文献[22]结合恶意代码的静态和动态特征,将其转换为字节码图像和 API 调用频率灰度图,引入金字塔池化构造双分支密集连接网络模型,实验表明,该方法能有效提升检测准确率。文献[23]以深度神经网络为框架,融合多尺度恶意代码特征与通道注意力机制,使用数据增强技术,实现了高分类准确率和检测时效性。

文献[24]提出一种恶意软件分类方法,融合操作码与可读字符的频率和语义特征、字节序列生成的马尔可夫图像特征,通过 LGBM/RF/CNN 第一层单特征分类、SVM 第二层决策融合的两层框架,提升分类准确率至 99.4%。文献[25]提出一种基于多特征融合的 BiTCN-TAEfficientNet 恶意软件分类方法,融合恶意软件的 API、操作码与操作数序列特征及二进制、操作码、API 信息生成的 RGB 图像特征,通过双向时间卷积网络及量子粒子

群优化算法优化分类,在 BIG2015 和 DataCon 数据集上的分类准确率分别达 99.46% 和 97.92%。这些研究从不同层面展示了特征融合在恶意代码检测中的应用,为该领域的发展提供了重要参考。

尽管上述方法都展现了出色的分类效果,但它们依然存在一些不足:

1) 基于单一维度的特征提取在恶意代码分类任务中会存在特征表示不足的情况,准确率很难进一步提高。

2) 表现较好的融合特征分类大量采用手工设计特征,手工设计特征依赖专家经验,且存在数据局限性,在面对新型恶意代码时表现欠佳。

3) 部分融合方法完全依赖深度学习模型,随着输入数据维度的增加、网络结构的复杂化,其计算开销增大,且模型可解释性欠佳。

本文提出了基于融合特征的恶意代码检测方法,该方法提取并融合恶意代码的行为特征、空间结构特征以及纹理特征,并使用 LightGBM 分类器进行分类。该方法在 Microsoft 恶意软件分类挑战赛

数据集^[26]上的分类准确率为 99.90%,对数损失(Logloss)为 0.005 7,在 Bazaar 数据集^[27]上的分类准确率为 98.97%,对数损失为 0.042 0,在不依赖特定手工设计特征的情况下,在面对新老恶意病毒样本时表现均优异,且模型可解释性更佳,计算开销更小。

2 检测方法概述

本文提出了一种恶意代码分类方法,其核心在于整合多维度特征并采用早期融合策略,最终使用 LightGBM 进行分类。在研究过程中,从行为、结构和纹理 3 个层面全面抽取恶意样本的关键特征,在特征融合阶段采用早期融合技术,并在训练过程中根据特征重要性进行特征选择,有效地整合这些多元特征信息,该方法应用于 Windows PE 文件的恶意代码检测和分类任务,旨在提升恶意软件识别的准确性和效率。本文恶意代码分类方法检测框架如图 1 所示(彩色效果见《计算机工程》官网 HTML 版,下同)。

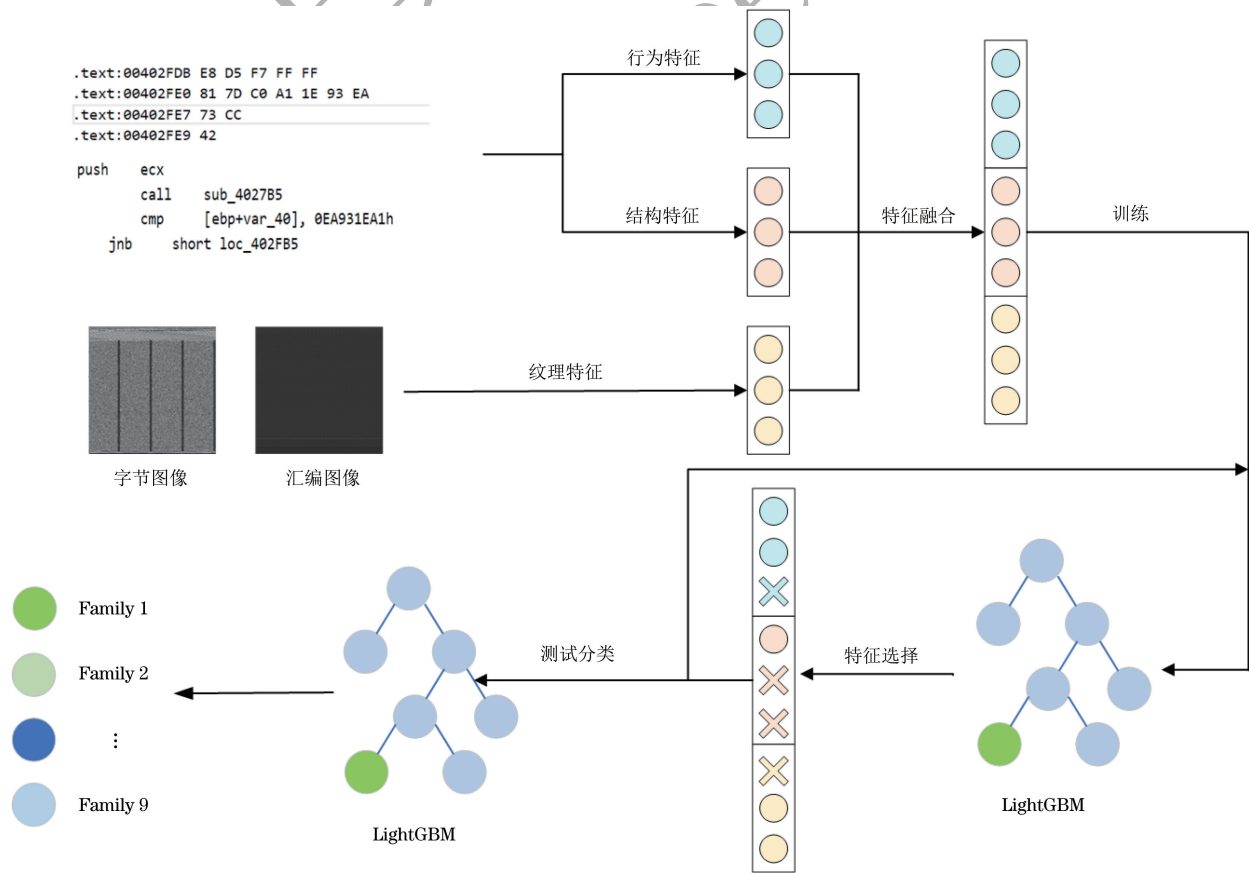


图 1 基于融合特征的恶意代码检测框架

Fig.1 Malicious code detection framework based on fusion features

2.1 特征定义

本文构建了一个多维度的 Windows PE 文件特征分析框架,该框架涵盖了多种有助于揭示 PE 文

件内在特性的特征类别,尽管部分特征可能不直接关联恶意行为,但却能有效描绘应用程序的独特属性。

2.1.1 行为特征

Opcode n -gram 特征:本文选择了若干具有代表性的 n -gram 特征,以增强恶意代码检测的表征能力和鲁棒性。在实验中发现,单一的 Opcode 1-gram 特征(如“add”“call”等)在捕捉复杂行为模式时存在一定的局限性。因此,本文进一步引入了 Opcode 2-gram 和 Opcode 3-gram 特征,并通过逆文档频率(IDF)方法统计这些操作码组合的频率,选取大于第九十分位数的操作码,从而筛选出更具区分度的特征。

本文选取了 93 个 Opcode 1-gram 特征,能够简单反映单一操作的频率分布。为增强上下文信息的捕获能力,统计了 178 个 Opcode 2-gram 特征(如“mov mov”“push push”等)以及 300 个 Opcode 3-gram 特征(如“push mov push”“push call add”等),这些特征能够更全面地刻画代码片段的行为模式和操作逻辑。

2.1.2 结构特征

Asm Section 特征:Asm Section 特征即汇编文件的区段特征,恶意软件通常采用非标准或随机的区段名称,以存储恶意负载或执行特定功能。这些区段通常在正常软件中不存在,因此,通过分析这些区段可以揭示恶意软件的潜在威胁。具体来说,恶意软件开发者可能会使用随机生成或特定的非标准

区段名称,这些名称通常在合法软件中不会出现,例如,.bss,.data,.text 等是标准区段名称,而恶意软件可能会使用诸如. xyz,. malware 等不常见的名称。此外,统计 PE 文件中的区段数量也十分重要,通常合法软件的区段数量是有限且符合规范的。如果一个 PE 文件中包含异常多的区段,这可能是恶意软件的迹象。通过区分已知和未知区段,统计每种区段的数量和比例,可以有效捕捉到异常的结构特征。例如,已知区段(如.text,. data 等)的比例明显偏低,而未知区段的比例异常偏高,则可能存在恶意行为。通过这些统计信息,可以更准确地识别出恶意软件的结构异常,区段特征向量长度为 24。

2.1.3 纹理特征

恶意代码可视化通过将二进制文件的字节数据按顺序映射为像素的灰度或颜色,把不可读的二进制数据转化为图像,其优点在于以直观的视觉形式呈现数据,便于利用人类视觉模式识别能力捕捉结构与模式,还能展示同家族恶意代码的图像关联性以辅助分类,如图 2 所示,为引入计算机视觉等技术搭建桥梁。利用图像处理领域的成熟技术有效提取恶意代码图像的纹理特征,能够在恶意代码分类任务中更全面地表征样本的细微差异与复杂模式,有效应对恶意代码变种带来的挑战。

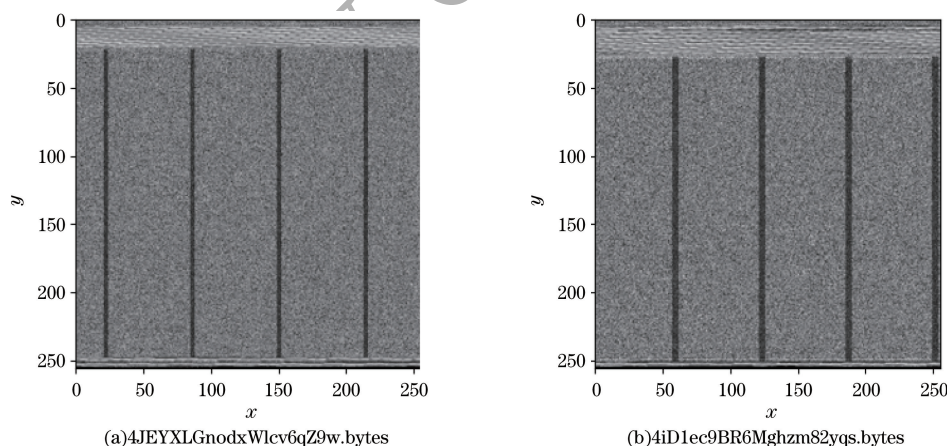


图 2 同家族的恶意样本文件可视化对比

Fig.2 Visual comparison of malicious sample files from the same family

深度纹理特征:深度纹理特征依托残差网络设计,凭借其在图像特征提取领域的显著优势,在本方法中被用于从恶意软件汇编代码转换的灰度图像中提取深度特征。本文采用预训练 ResNet50 模型^[28],鉴于输入为单通道灰度图像,将模型首层卷积层输入通道数调整为 1。在图像预处理阶段,尺寸统一调整为 224×224 ,并实施归一化操作。经预处理的图像输入改造后的 ResNet50 模

型,去除模型分类头,仅保留特征提取部分,为降低特征维度,添加全连接层,将特征维度由 2 048 降至 256。

局部二值模式(LBP)^[29]和 Haralick^[30]特征:恶意软件通常通过修改其代码结构和行为特征来躲避检测,然而,其底层二进制数据及汇编代码的模式和结构特征仍然存在。

运用图像处理领域的 LBP 和 Haralick 纹理特

征提取方法,对恶意代码图像的纹理特征进行深入分析和提取,将其应用于后续的特征融合及分类任务。LBP 是一种用于纹理分析的有效方法,通过比较中心像素与周围像素的灰度值,生成一个二进制模式,对每个像素,若其周围像素的灰度值高于中心像素,则标记为 1,否则标记为 0。最终得到一个二进制数,用于表示该像素的局部纹理特征。

Haralick 特征基于共生矩阵 (Co-occurrence Matrix),可以使用 Mahotas 库中的 Haralick 函数来计算,通过统计图像中像素对的空间关系,提取出一组描述图像纹理的统计特征。

经过提取,其中,LBP 特征向量长度为 108,Haralick 特征向量长度为 52。

Pixel intensity 特征:Pixel intensity 即像素强度特征,以二进制模式读取汇编文件,采用同样的图像转换技术,本文将恶意软件的汇编语言源代码视作灰度图像进行处理,如图 3 所示。该方法将灰度图像的像素强度值作为特征,以量化图像的视觉信息^[31]。对于给定的灰度图像,它会先将图像展平为一维数组,将所有像素按行依次排列。然后,提取该数组中指定数量的像素强度值,将每个值作为一个特征,并使用一个有序字典来存储这些特征。每个特征的键是一个字符串,用于标识该像素的位置,值则是该像素的实际强度值。通过这种方式,可以从图像中获取特定数量的像素强度特征,用于后续的分析或机器学习任务,特征长度为 800。

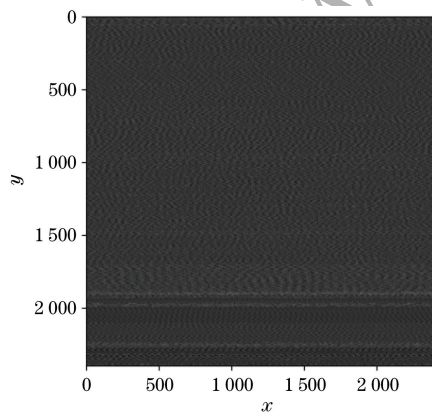


图 3 恶意样本.asm 文件可视化

Fig.3 Malicious sample .asm file visualization

2.2 特征融合

本文采用特征融合技术,旨在集成不同源头或具备不同类型特征的信息,以构建更为详尽且精确的数据表征,进而赋能机器学习模型的训练过程。通过特征融合,可以全方位地捕获多元特征间的关系与模式,提升模型的解释性和预测性能。其中,常

用的融合策略包括特征拼接、特征交叉、特征嵌入、特征选择、特征组合等。

在系统设计中,本文采取了一种早期融合机制以构筑一个多模态特征的综合表示形式。此机制的核心在于将独立的多种特征向量逐一拼接成一个统一的长向量,实现了跨模态特征的有效整合。早期融合的优势在于保留原始特征的完整信息,避免了后期融合(如模型输出结果融合)可能导致的特征语义损失,为分类器提供更丰富的输入维度。

针对特征维度不同而带来的权重分配问题,LightGBM 迭代训练弱决策树,每个树节点的分裂依据是特征减少对损失函数的贡献,特征重要性评分越高,说明该特征在分类中权重越大。权重分配完全由数据驱动,无须专家经验介入。

在特征提取完成后,PE 文件被表示为多个特征向量,即 $\vec{v}, \vec{w}, \dots, \vec{z}$,它们分别对应于不同类型的特征,从而对内容提供抽象视图。最终模型的预测(记为 h)如式(1)所示:

$$p = h([\vec{v}, \vec{w}, \dots, \vec{z}]) = h([v_1, v_2, \dots, v_i, w_1, w_2, \dots, w_j, \dots, z_1, z_2, \dots, z_k]) \quad (1)$$

2.3 特征选择

在特征融合的基础上,本文进一步引入了特征选择技术,旨在从高维度的特征空间中筛选出对分类任务具有关键影响的特征,以降低模型复杂度,提升计算效率,同时避免冗余特征对模型训练的干扰。特征选择不仅能够简化模型的结构,还能增强其泛化能力,从而提高在未知数据上的预测性能。

本研究采用了基于 LightGBM 的嵌入式特征选择方法,通过计算特征的重要性评分,自动评估每个特征对分类任务的贡献大小。在具体实现过程中,本文遵循以下步骤:

1) 初步训练。在融合后的特征集上训练 LightGBM 模型,以计算特征的重要性分数。

2) 重要性排序。依据 LightGBM 输出的特征重要性分数对特征进行排序。

3) 特征筛选。根据设定的特征数量(如前 900 个特征)或重要性阈值,筛选出最具代表性的特征子集。

特征选择后的特征子集不仅保留了多模态特征间的重要信息,还显著减少了输入维度,使后续模型训练更为高效。此外,为验证特征选择的有效性,本文在交叉验证的第一个折次中完成特征选择,并在所有折次中共享筛选后的特征子集,从而确保筛选

过程的一致性和稳定性。

实验结果表明,特征选择策略能够在保证模型预测性能的同时降低计算开销。与未进行特征选择的模型相比,筛选后的模型在分类准确率和 Logloss 等指标上表现更为优异。通过特征选择,本研究进一步优化了多模态特征的表示能力,为高效的恶意代码检测奠定了基础。

2.4 LightGBM 分类器

针对特征维度长、数据规模大、数据分布稀疏的特点,在分类任务上,本文选择了 LightGBM^[32] 作为分类器工具。LightGBM 是梯度提升决策树 (GBDT) 算法的一种高效实现版本,它继承了 XGBoost 的优秀特性,并针对效率和性能进行了深度优化,尤其适用于大规模数据集和需要高精度分类的任务。LightGBM 通过直方图近似法减少了内存使用,通过 GOSS (Gradient-based One-Side Sampling) 和 EFB (Exclusive Feature Bundling) 技术提高了计算效率,支持并行化训练,易于调整参数。此外,在此次任务中融合特征维度较长,不同样本提取出的特征取值差异较大,会出现数据集稀疏的问题,LightGBM 能自动处理缺失值从而解决这一问题。这些技术创新使得 LightGBM 在处理大规模 PE 文件特征数据集时,能够在保持高精度的同时,显著降低内存消耗和计算时间,非常适合本次的分类任务。

梯度提升决策树作为一种集成学习算法,其工作原理在于逐步构建一系列弱学习器——决策树,每个新的决策树都致力于拟合前面模型集合所遗留的残差。具体来说,在每一次迭代中,模型首先计算当前模型预测结果与真实标签之间的梯度(或残差),然后训练一个新的决策树去拟合这些梯度,这样做的目的是逐步改进模型在尚未准确预测部分的效果。

在数学上,假设当前的模型预测为 $F_t(x)$,目标是对样本 x 的真实标签 y 进行预测,损失函数记为 $L(y, F_t(x))$ 。在第 $t+1$ 步迭代时,LightGBM 寻找一个弱学习器 $h_{t+1}(x_i)$ 来最小化加权残差平方和,如式(2)所示:

$$\min_{h_{t+1}} \sum_{i=1}^n L(y_i, F_t(x_i) + h_{t+1}(x_i)) \quad (2)$$

在每次迭代后,更新模型预测为 $F_{t+1}(x)$,如式(3)所示:

$$F_{t+1}(x) = F_t(x) + h_{t+1}(x) \quad (3)$$

这个递归过程持续进行,直至达到预设的最大迭代次数或者满足其他停止条件。通过不断累加新

的决策树,最终形成的强学习器能够综合所有单个决策树的输出,从而实现对复杂非线性关系的建模,并显著提升分类性能。

3 实验验证

3.1 数据集与实验环境

本次实验采用了 2 个数据集作为数据源,分别为 Kaggle 举办的 Microsoft 恶意软件分类挑战赛数据集和自建的 Bazaar 恶意软件数据集,以全面覆盖多样化的恶意软件家族,验证模型的泛化能力。Microsoft 恶意软件分类挑战赛数据集涵盖了 9 个不同恶意软件家族的共计 10 868 个代表性样本,具体包括 Ramnit、Lollipop、Kelihos_ver3、Vundo、Simda、Tracur、Kelihos_ver1、Obfuscator.ACY 以及 Gatak,各家族样本数量详情如图 4 所示,在后续实验中称此数据集为 Microsoft 数据集。

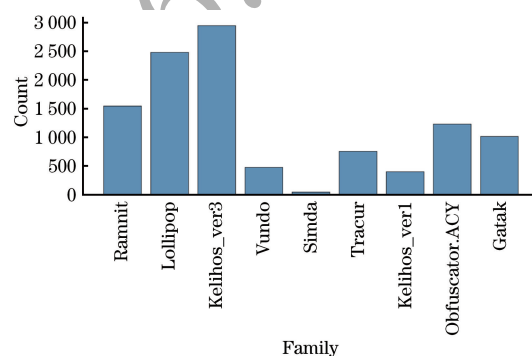


图 4 Microsoft 数据集分布情况

Fig. 4 Distribution of Microsoft dataset

自建的 Bazaar 数据集基于 2024 年及以后 Malware Bazaar 网站收录的最新恶意软件家族样本,包含了 AgentTesla、Amadey、Stealc、DeRat、Socks5Systemz、Healer、Emotet、CobaltStrike 和 TrickBot 等 9 个家族的共计 4 398 个样本,提供了当前威胁趋势下的典型代表性样本,具体分布情况如图 5 所示,在后续实验中称此数据集为 Bazaar 数据集。

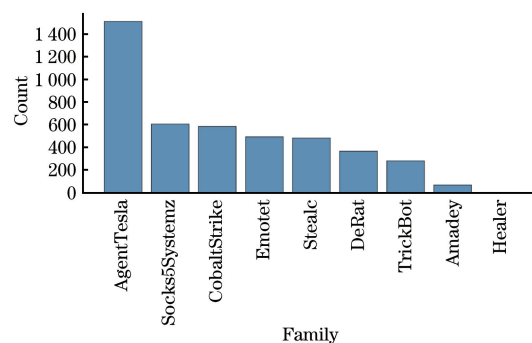


图 5 Bazaar 数据集分布情况

Fig. 5 Distribution of Bazaar dataset

实验是在配备 AMD Ryzen 7 8845H 处理器、NVIDIA GeForce RTX 4070 Laptop 显卡和 32 GB DDR5 内存的硬件环境下进行的,系统运行 Windows 11 家庭版,并配有两块 SSD,总容量约为 2.8 TB。

3.2 评价指标

准确率(Accuracy, A)是评估预测模型效果的常用性能指标之一,它衡量了正确预测的比例。准确率的定义如式(4)所示:

$$A = \frac{N_{TP} + N_{TN}}{N_{TP} + N_{TN} + N_{FP} + N_{FN}} \quad (4)$$

式中: N_{TP} 表示真正例,即模型正确预测为正类的样本数量; N_{TN} 表示真负例,即模型正确预测为负类的样本数量; N_{FP} 表示假正例,即模型错误预测为正类的样本数量; N_{FN} 表示假负例,即模型错误预测为负类的样本数量。

多类别对数损失(Logloss, L_{Logloss})是真实标签分布与预测概率之间的交叉熵,计算方式如式(5)所示:

$$L_{\text{Logloss}} = -\frac{1}{N} \sum_{i=1}^N \sum_{j=1}^M y_{i,j} \log_a(p_{i,j}) \quad (5)$$

式中: N 是样本数量; M 是类别标签数量; $\log$ 表示自然对数; $p_{i,j}$ 是样本 i 属于类别 j 的预测概率。如果样本 i 属于类别 j ,则 $y_{i,j}$ 为 1,否则为 0。

3.3 实验结果分析

3.3.1 分类效果

本节对模型在恶意软件分类任务中的表现进行全面分析,评估其分类效果。在本研究中,使用 LightGBM 框架进行恶意软件分类模型的训练与评估。为了优化模型性能并确保其在多分类任务中的有效性,对模型的参数进行了选择和调整,具体参数设置如表 1 所示。

表 1 LightGBM 模型参数
Table 1 LightGBM model parameters

参数	设置
learning_rate	0.1
min_child_weight	5
num_class	9
nthread	4
objective	multiclass
metric	multi_error, multi_logloss
num_iterations	200

为了更直观地呈现模型的分类准确性,图 6 与图 7 分别展示了模型在 Bazaar 数据集与 Microsoft 数据集上对各类恶意家族的分类效果。

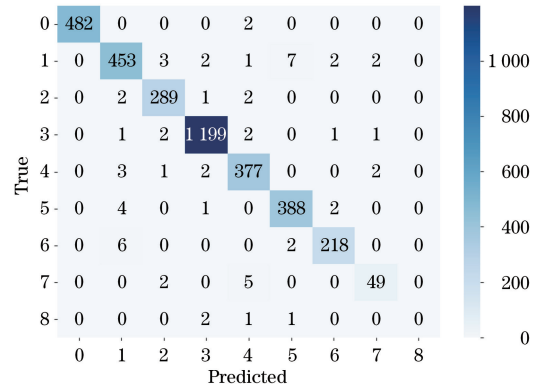


图 6 Bazaar 数据集上分类结果的混淆矩阵

Fig.6 Confusion matrix of classification results on Bazaar dataset

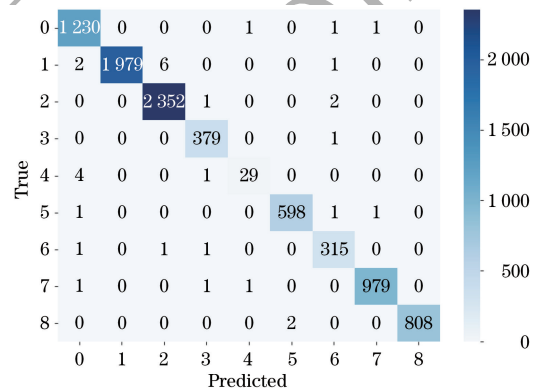


图 7 Microsoft 数据集上分类结果的混淆矩阵

Fig.7 Confusion matrix of classification results on Microsoft dataset

3.3.2 单一特征对比

为了证明特征融合方法的优越性及其增强的泛化能力,分别采用各个单一特征训练模型。每轮测试后,将模型在测试集上的表现以对数损失为指标记录下来,结果如表 2 所示。

表 2 单一特征性能对比

Table 2 Single feature performance comparison

特征	Microsoft		Bazaar	
	Accuracy	Logloss	Accuracy	Logloss
Opcode 1-gram	0.992 4	0.024 2	0.960 2	0.195 7
Opcode 2-gram	0.990 7	1.483 8	0.850 2	0.462 7
Opcode 3-gram	0.978 8	0.037 5	0.851 7	0.443 5
Section	0.989 8	0.035 1	0.935 4	0.277 8
LBP	0.981 4	0.061 2	0.942 9	0.245 7
Haralick	0.978 7	0.085 2	0.949 7	0.216 1
Pixel intensity	0.765 9	9.533 1	0.666 9	1.347 5
deep	0.995 1	0.013 0	0.975 1	0.093 0
Fusing features	0.999 0	0.005 7	0.989 7	0.042 0

3.3.3 与其他分类器的对比

为了选择适合的分类型器,本文对比了随机森林(RF)、 k -近邻(k -NN)、支持向量机(SVM)、

XGBoost、神经网络(NN)这几种方法。实验结果通过十折交叉验证以及测试集上的对数损失和准确率进行评价。从表 3 可以看出,LightGBM 在 Microsoft 测试集上的对数损失为 0.005 7,优于其他所有对比方法,在Bazaar数据集上,其准确率与

表 3 LightGBM 与其他分类器的对比

Table 3 Comparison between LightGBM and other classifiers

分类器	Microsoft		Bazaar	
	Accuracy	Logloss	Accuracy	Logloss
RF	0.998 1	0.016 1	0.964 7	0.212 9
k-NN	0.971 7	0.428 6	0.921 9	0.951 9
SVM	0.903 0	0.436 7	0.695 2	0.872 9
XGBoost	0.998 2	0.008 8	0.963 2	0.173 9
NN	0.996 1	0.025 9	0.956 4	0.216 4
LightGBM	0.999 0	0.005 7	0.989 7	0.042 0

对数损失也都最佳。

3.3.4 特征贡献可视化

在特征内容可视化研究中,本文利用基于LightGBM算法的 Gini 重要性量化方法实现特征重要性排名,通过正则化清洗特征名称,利用交叉验证筛选关键特征(如首折后保留前 900 个重要特征)及可视化呈现(生成特征重要性条形图与保存 JSON 数据),系统揭示模型决策逻辑。相较于深度学习模型的“黑箱”特性,该方法作为树模型原生支持的可解释工具,通过特征分裂增益累加明确量化每个特征对分类结果的贡献度,在透明性、领域适配性及优化导向性上具有显著优势,例如,其能辅助专家验证模型逻辑并为特征工程提供直接指导。在特征重要性分析中,图 8、图 9 这 2 张可视化图呈现出显著差异。

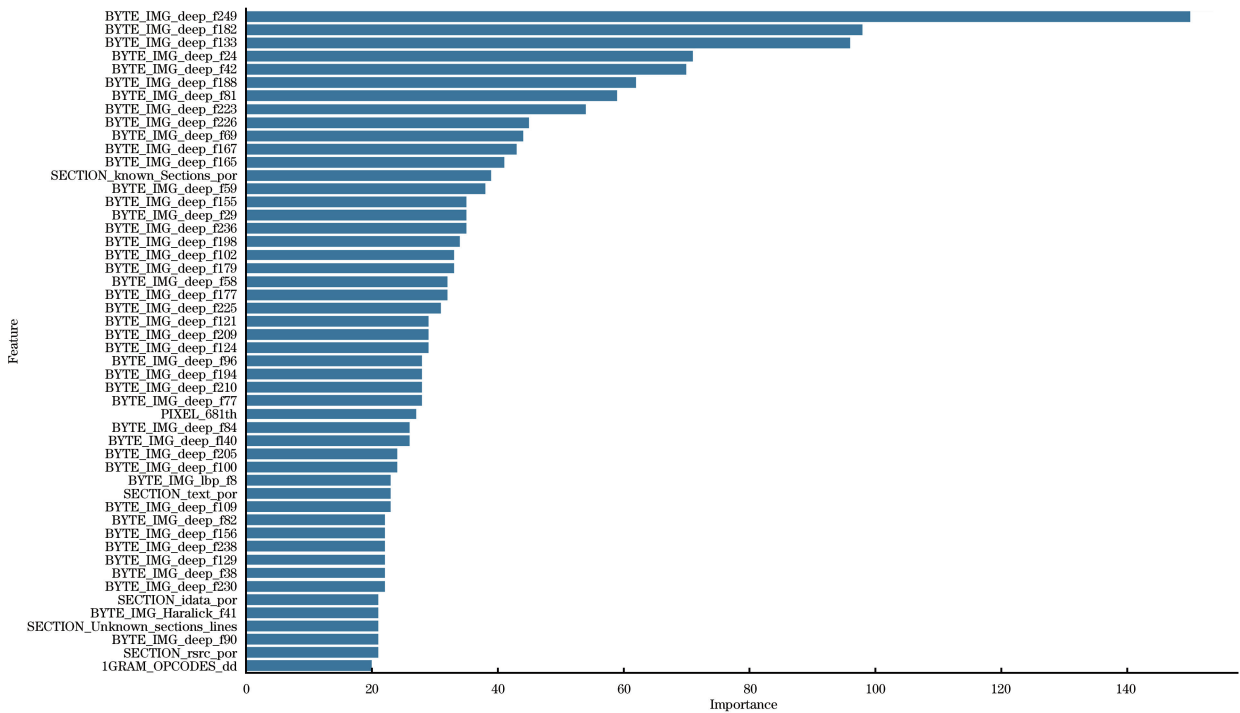


图 8 Microsoft 数据集中的特征重要性排名

Fig.8 Feature importance ranking in the Microsoft dataset

从特征图 8、图 9 中可以看出,Bazaar 数据集包含 2024 年后的新型恶意软件(如 AgentTesla、Emotet),这类样本普遍采用加密、混淆技术(如动态代码生成、多态变异),导致传统 Opcode 序列等浅层特征被破坏。尽管恶意代码通过加密混淆改变指令表现形式,但其核心功能代码的字节分布在图像化后仍可能呈现特定周期性。例如,加密循环体在图像中形成规律的纹理模式,反映恶意软件为维持功能而保留的底层结构特征。此外,正常 PE 文件的区段边界(如代码段与数据段过渡)在图像中表现为平滑的像素变化。而恶意软件注入混淆代码或

加密数据时,会导致区段边界出现像素突变(如灰度值骤变),ResNet50 提取的深层纹理特征能够捕捉恶意代码图像中未被混淆破坏的底层结构,对变异恶意代码具有更强的鲁棒性。对于 Pixel intensity 特征,恶意软件常通过添加签名、打包器痕迹等元数据实现后续功能(如 C2 通信标识、版本管理)。这些信息在图像中表现为特定位置的像素强度异常。例如,某恶意家族在汇编头部嵌入独特灰度模式作为通信签名,当模型检测到对应 PIXEL 强度值偏离正常范围时,即可映射到恶意软件的隐藏通信或自我标识行为。如 PIXEL_750th、PIXEL_668th 等

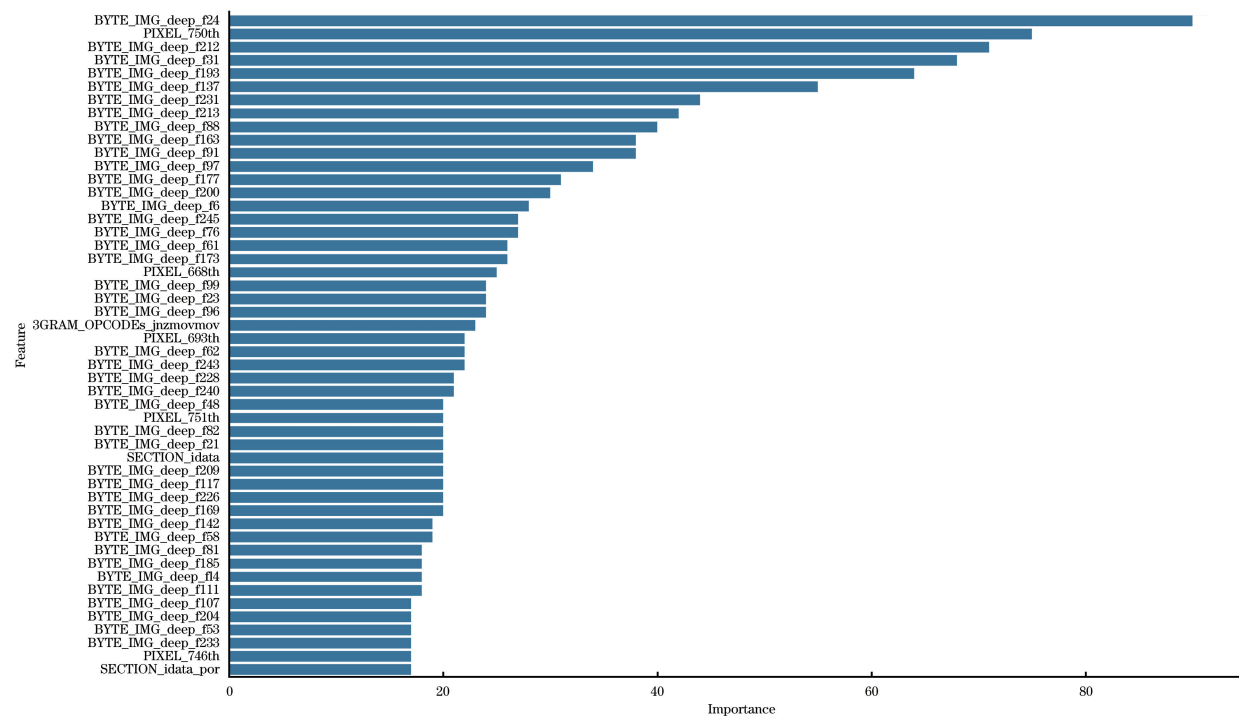


图 9 Bazaar 数据集中的特征重要性排名

Fig.9 Feature importance ranking in the Bazaar dataset

位置的像素强度值,对应图像中特定区域的灰度值,这些位置可能隐含恶意代码的元数据信息(如签名、打包器痕迹)。而 SECTION_idata、SECTION_idata_per 等特征反映区段_idata 的统计特征,.idata 区段通常是可执行文件用于存储导入表相关信息的区段,合法 PE 文件通常包含 1 或 2 个导入表区段,而恶意样本可能通过分割或伪造区段导致区段数量异常,正常文件的_idata 区段占比约为 5%~10%,恶意样本因注入混淆代码或加密数据,可能导致该区段占比显著升高或降低,由此区段特征可以捕捉到这类恶意行为的异常。

相较于 Microsoft 数据集(图 8 中 Opcode 3-gram 和 Section 特征重要性较高),Bazaar 数据集更依赖深层纹理特征,新型恶意代码的加密混淆技术迫使模型从“基于指令序列的行为特征”转向“基于图像纹理的结构特征”。LightGBM 的特征选择机制能自适应数据特性,在不同数据集上动态筛选关键特征,体现了融合特征的优势以及方法的泛化能力。

通过 LightGBM 的 Gini 重要性排序,明确揭示了模型的决策逻辑:深度纹理特征主导分类,表明恶意代码图像的全局纹理模式是区分家族的核心依据;结构特征(区段)和浅层行为特征(Opcode)作为辅助,补充细节信息。这避免了深度学习模型的“黑箱”问题,为安全专家分析新型恶意软件的可视

化特征提供了直接线索(如特定家族图像的纹理共性)。

3.3.5 计算开销

为评估本文方法的效率优势,本节将其计算开销与近年来恶意代码分类领域的主流方法(包括多模态深度学习模型、传统特征融合方法)进行对比,对比指标涵盖训练时间、推理速度,以 Bazaar 数据集上的复现结果为例,具体数据如表 4 所示。在计算开销上,本文方法在 Bazaar 数据集上的训练耗时为 25 min,单个样本推理时间为 0.011 ms,显著低于对比方法,这种高效性得益于采用预训练 ResNet 作为特征提取器,避免了复杂深度学习模型的高计算消耗,同时 LightGBM 分类器凭借高效的训练与推理机制,进一步提升了整体效率。

表 4 计算开销对比

Table 4 Comparison of computational costs

方法	训练时间/min	单个样本推理时间/ms
文献[20]方法	65	5.142
文献[15]方法	55	0.024
文献[24]方法	101	10.318
文献[25]方法	90	7.233
本文方法	25	0.011

3.3.6 与现有方法的比较

将本文方法与现有方法进行对比,结果如表 5 所示。对于已公布的成果,以原文中的实验结果为准,对于 Bazaar 数据集中的表现,则以复现结果为

准。从表 5 可以看出:在 Microsoft 数据集上,本文方法测试准确率达 0.999 0,对数损失低至 0.005 9;在 Bazaar 数据集上,本文方法测试准确率为 0.989 7,对数损失为 0.042 0,两项指标均优于对比方法。这充分表明,通过整合行为特征、结构特征及纹理特征的策略,本文方法有效避免了传统特征工程方法特征维度单一、表征能力不足的缺陷,无须依赖专家经验进行手工特征设计,即可全面捕捉恶意代码的复杂模式。同时,采用 LightGBM 作为分类器,规避了多模态深度学习模型可解释性不足的问题。LightGBM 基于树模型的特性,其特征重要性分析为模型决策提供了清晰的解释路径,且在处理效率上更具优势,实验结果中高精度与低对数损失的表现,验证了模型在准确分类的同时,并未因复杂特征融合而增加不必要的计算负担,实现了性能与效率的平衡。

表 5 本文方法与现有方法的比较

Table 5 Comparison between this method and existing methods

方法	Microsoft		Bazaar	
	Accuracy	Logloss	Accuracy	Logloss
文献[20]方法	0.974 7	—	0.969 0	0.081 0
文献[15]方法	0.998 1	0.007 0	0.980 1	0.066 0
文献[24]方法	0.994 0	—	0.976 9	0.110 0
文献[25]方法	0.994 6	—	0.978 1	0.096 0
本文方法	0.999 0	0.005 9	0.989 7	0.042 0

4 结束语

本文针对传统特征工程方法、现有特征融合方法以及多模态深度学习模型的局限性,提出一种新的特征融合方法,其整合行为特征、结构特征及纹理特征,并采用 LightGBM 完成 Windows PE 文件中的恶意代码分类。实验结果表明,该方法在 Microsoft 与 Bazaar 数据集上分别取得了 0.999 0 与 0.989 7 的准确率,且对数损失分别为 0.005 9 和 0.042 0,能够全面、准确地表征恶意代码。该方法无须手工特征设计,摆脱了对专家经验的依赖,同时借助 LightGBM 的可解释性与低计算开销特性,克服了多模态深度学习模型的固有缺陷。本文研究成果为恶意代码检测提供了一种高效、可解释的解决方案,在网络安全防护领域具有广阔的应用前景。未来可进一步探索更丰富的特征维度或优化模型参数,以应对不断演变的恶意代码威胁,持续提升检测方法的泛化能力与鲁棒性。

参考文献

[1] NARAYANAN B N, DJANEYE-BOUNDJOU O, KEBEDE T M. Performance analysis of machine learning and pattern recognition algorithms for malware classification [C] // Proceedings of the IEEE National Aerospace and Electronics Conference (NAECON) and Ohio Innovation Summit (OIS). Washington D. C., USA: IEEE Press, 2016: 338-342.

[2] VASAN D, ALAZAB M, WASSAN S, et al. IMCFN: image-based malware classification using fine-tuned convolutional neural network architecture [J]. Computer Networks, 2020, 171: 107138.

[3] YUAN B G, WANG J F, LIU D, et al. Byte-level malware classification based on Markov images and deep learning [J]. Computers & Security, 2020, 92: 101740.

[4] SHAUKAT K, LUO S H, VARADHARAJAN V. A novel deep learning-based approach for malware detection [J]. Engineering Applications of Artificial Intelligence, 2023, 122: 106030.

[5] LIN W C, YEH Y R. Efficient malware classification by binary sequences with one-dimensional convolutional neural networks [J]. Mathematics, 2022, 10(4): 608.

[6] ZHANG X L, WU K H, CHEN Z G, et al. MalCaps: a capsule network based model for the malware classification [J]. Processes, 2021, 9(6): 929.

[7] XIAO M, GUO C, SHEN G W, et al. Image-based malware classification using section distribution information [J]. Computers & Security, 2021, 110: 102420.

[8] ÇAYIR A, ÜNAL U, DAĞ H S. Random CapsNet forest model for imbalanced malware type classification task [J]. Computers & Security, 2021, 102: 102133.

[9] YAN J Q, YAN G H, JIN D. Classifying malware represented as control flow graphs using deep graph convolutional neural network [C] // Proceedings of the 49th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN). Washington D. C., USA: IEEE Press, 2019: 52-63.

[10] ASLAN O, YILMAZ A A. A new malware classification framework based on deep learning algorithms [J]. IEEE Access, 2021, 9: 87936-87951.

[11] WANG H M, ZHAO Y T, WANG Z J. Doc2vec-GRU: a behavior classification method for malicious code [J]. International Journal of Network Security, 2024, 26(3): 467-476.

[12] YESIR S, SOGUKPINAR I. Malware detection and classification using fastText and BERT [C] // Proceedings of the 9th International Symposium on Digital Forensics and Security (ISDFS). Washington D. C., USA: IEEE Press, 2021: 1-6.

[13] KUMAR P S, UMI SALMA B, MISHRA I, et al. Malware detection classification using recurrent neural network [C] // Proceedings of the 2nd International Conference on Technological Advancements in Computational Sciences (ICTACS). Washington D. C., USA: IEEE Press, 2022: 876-880.

[14] GIBERT D, MATEU C, PLANES J. HYDRA: a multimodal deep learning framework for malware classification [J]. Computers & Security, 2020, 95: 101873.

[15] GIBERT D, PLANES J, MATEU C, et al. Fusing feature engineering and deep learning: a case study for malware classification [J]. Expert Systems with Applications, 2022, 207: 117957.

[16] SEONGKYU Y, HAENGROK O, DONGIL S, et al. Automatic malicious code classification system through static analysis using machine learning [EB/OL]. [2024-02-05]. <https://pdfs.semanticscholar.org/da97/387be1ae641c0f262e>

- 5ba9f5b011d372c5c0.pdf.
- [17] YOUSUF M I, ANWER I, RIASAT A, et al. Windows malware detection based on static analysis with multiple features[J]. *PeerJ Computer Science*, 2023, 9: e1319.
- [18] ULLAH F, SRIVASTAVA G, ULLAH S. A malware detection system using a hybrid approach of multi-heads attention-based control flow traces and image visualization[J]. *Journal of Cloud Computing*, 2022, 11(1): 75.
- [19] SINGH J, SINGH J. Detection of malicious software by analyzing the behavioral artifacts using machine learning algorithms[J]. *Information and Software Technology*, 2020, 121: 106273.
- [20] KIM J Y, CHO S B. Obfuscated malware detection using deep generative model based on global/local features[J]. *Computers & Security*, 2022, 112: 102501.
- [21] 熊其冰, 郭洋, 王世豪. 基于多特征融合和增强模型的恶意代码检测方法[J]. *通信技术*, 2023, 56(5): 640-646.
- XIONG Q B, GUO Y, WANG S H. Malicious code detection method based on multi-feature fusion and enhanced model[J]. *Communications Technology*, 2023, 56(5): 640-646. (in Chinese)
- [22] 李梦, 刘万平, 黄东. 基于特征融合的恶意代码检测[J]. *计算机工程与设计*, 2024, 45(12): 3568-3574.
- LI M, LIU W P, HUANG D. Malicious code detection based on feature fusion[J]. *Computer Engineering and Design*, 2024, 45(12): 3568-3574. (in Chinese)
- [23] 王硕, 王坚, 王亚男, 等. 一种基于特征融合的恶意代码快速检测方法[J]. *电子学报*, 2023, 51(1): 57-66.
- WANG S, WANG J, WANG Y N, et al. A fast malicious code detection method based on feature fusion[J]. *Acta Electronica Sinica*, 2023, 51(1): 57-66. (in Chinese)
- [24] YAN H, ZHANG J, TANG Z G, et al. Malware classification method based on feature fusion[J]. *International Journal of Information Security*, 2025, 24(2): 97.
- [25] XUAN B N, LI J, SONG Y F. BiTCN-TAEfficientNet malware classification approach based on sequence and RGB fusion[J]. *Computers & Security*, 2024, 139: 103734.
- [26] ALESSANDRO P, MARIAN M, WILL C, et al. Microsoft malware classification challenge (BIG 2015) [EB/OL]. [2024-02-05]. <https://www.kaggle.com/c/malware-classification>.
- [27] MalwareBazaar. Malware samples repository [EB/OL]. [2024-02-05]. <https://bazaar.abuse.ch/>.
- [28] KOONCE B. ResNet50 [C] // *Proceedings of International Conference on Convolutional Neural Networks with Swift for Tensorflow*. Berkeley, CA: Apress, 2021: 63-72.
- [29] OJALA T, PIETIKAINEN M, HARWOOD D. Performance evaluation of texture measures with classification based on Kullback discrimination of distributions[C] // *Proceedings of the 12th International Conference on Pattern Recognition*. Washington D.C., USA: IEEE Press, 1994: 582-585.
- [30] HARALICK R M, SHANMUGAM K, DINSTEN I. Textural features for image classification [J]. *IEEE Transactions on Systems, Man, and Cybernetics*, 1973, 3(6): 610-621.
- [31] ZHANG Y N, HUANG Q J, MA X J, et al. Using multi-features and ensemble learning method for imbalanced malware classification [C] // *Proceedings of the IEEE Trustcom/BigDataSE/ISPA*. Washington D. C., USA: IEEE Press, 2016: 965-973.
- [32] MENG Q. LightGBM: a highly efficient gradient boosting decision tree[EB/OL]. [2024-02-05]. https://proceedings.neurips.cc/paper_files/paper/2017/file/6449f44a102fde848669bdd9eb6b76fa-Paper.pdf.

编辑 吴云芳