基于图像化方法的恶意软件检测与分类综述

谢丽霞[1] 魏晨阳[1] 杨宏宇[1),2)] 胡 泽[2] 成 翔[3),4)] 张 良[5]

[1)](中国民航大学计算机科学与技术学院 天津 300300)

[2)](中国民航大学安全科学与工程学院 天津 300300)

[3)](扬州大学信息工程学院 江苏 扬州 225127)

[4)](中国民航大学民航飞联网重点实验室 天津 300300)

[5)](亚利桑那大学信息学院 图森 85721 美国)

摘 要 恶意软件的检测与分类是一种发现并消除潜在威胁、识别恶意软件家族的有效方法,在个人隐私保护和系统安全维护等任务中起关键作用。传统检测分类方法在面对使用复杂混淆技术的恶意软件新变种时,存在检测准确率低、误报率高和计算成本高等问题。在此背景下,利用基于深度学习的图像化方法解决恶意软件检测分类问题逐渐成为研究热点。因此,为全面总结分析图像化方法在恶意软件检测分类领域的应用,本文首先概述了恶意软件的定义、发展历程以及常用的混淆规避技术,讨论了基于静态分析、动态分析以及机器学习的检测分类方法存在的局限性,尤其是在应对复杂混淆技术和未知变种方面存在的不足。然后,系统总结了近年来图像化检测方法的最新研究进展,全面评估该方法在检测不同类型、不同平台(Windows、Android、IoT)恶意软件时的应用效果,深入分析该方法在提升检测分类精度、对抗高级混淆技术以及处理恶意软件新变种时的优势。最后,本文介绍并分析了可用于评估实验模型性能的各类数据集,深入讨论了图像化检测分类方法当前面临的各种挑战,并对未来研究方向进行了展望。

关键词 恶意软件;检测与分类;混淆技术;深度学习;图像化方法;数据集

中图法分类号 TP309 **DOI 号** 10.11897/SP.J.1016.2025.00650

Review on Malware Detection and Classification Using Imaging-Based Methods

XIE Li-Xia[1] WEI Chen-Yang[1] YANG Hong-Yu[1),2)] HU Ze[2] CHENG Xiang[3),4)] ZHANG Liang[5]

[1)](School of Computer Science and Technology, Civil Aviation University of China, Tianjin 300300)

[2)](School of Safety Science and Engineering, Civil Aviation University of China, Tianjin 300300)

[3)](School of Information Engineering, Yangzhou University, Yangzhou, Jiangsu 225127)

[4)](Key Laboratory of Civil Aviation Flight Networking, Civil Aviation University of China, Tianjin 300300)

[5)](School of Information, The University of Arizona, Tucson 85721 USA)

Abstract The detection and classification of malware are essential processes for identifying potential threats, recognizing malware families, and mitigating security risks. These tasks are critical in various applications, such as personal privacy protection and system security maintenance. However, traditional malware detection and classification methods face significant challenges, particularly when encountering new malware variants that employ advanced obfuscation techniques. Specifically, these methods often suffer from low detection accuracy, high false positive rates, and substantial computational costs. As a result, the growing complexity of malware has made it increasingly

收稿日期:2023-09-22;在线发布日期:2024-10-23。本课题得到国家自然科学基金民航联合研究基金重点项目(U2433205)、国家自然科学基金项目(62201576,U1833107)、江苏省基础研究计划自然科学基金青年基金项目(BK20230558)、中国民航大学飞联网重点实验室开放基金(MHFLW202304)资助。**谢丽霞**,硕士,教授,中国计算机学会(CCF)高级会员,主要研究领域为网络与系统安全、信息安全。E-mail:lxxie@126.com。**魏晨阳**,硕士研究生,主要研究领域为网络信息安全、恶意软件检测与分类。**杨宏宇**(通信作者),博士,教授,博士生导师,中国计算机学会(CCF)高级会员,主要研究领域为网络与系统安全、软件安全检测、网络安全态势感知。E-mail:yhyxlx@hotmail.com。**胡 泽**,博士,讲师,硕士生导师,主要研究领域为人工智能、自然语言处理、网络信息安全。**成 翔**,博士,实验师,硕士生导师,主要研究领域为网络与系统安全、网络安全态势感知、APT 攻击检测。**张 良**,博士,研究员,主要研究领域为强化学习、基于深度学习的信号处理、网络与系统安全。

difficult for traditional approaches to maintain the effectiveness needed for real-time security applications. In response to these limitations, deep learning-based imaging techniques have emerged as a significant area of research, offering potential solutions to the problems associated with traditional detection methods. This paper aims to provide a comprehensive review and analysis of the application of imaging techniques in malware detection and classification. Initially, the paper presents an overview of malware, defining its characteristics, tracing its evolution, and discussing the commonly used obfuscation and evasion techniques that enable malware to evade traditional detection methods. Furthermore, the limitations of conventional detection techniques, including static analysis, dynamic analysis, and machine learning-based methods, are explored in detail. These traditional approaches often struggle to effectively address the challenges posed by complex obfuscation strategies and previously unknown malware variants, which limit their overall effectiveness in real-world scenarios. The growing sophistication of malware continues to expose the weaknesses in these conventional methods, underscoring the need for innovative solutions. Following this, the paper systematically summarizes the latest research advancements in imaging-based malware detection methods. This involves transforming malware binaries into images, which can then be processed using deep learning models for classification. The effectiveness of these techniques is evaluated across different malware types and platforms, such as Windows, Android, and Internet of Things devices. A detailed evaluation of the imaging techniques highlights their ability to improve detection accuracy, counter advanced obfuscation techniques, and effectively manage new and evolving malware variants. The paper also emphasizes the unique advantages that imaging-based approaches provide, particularly in enhancing the robustness and adaptability of detection models, which is critical when dealing with increasingly sophisticated malware threats. These approaches are proving to be not only more accurate but also more efficient in detecting complex malware. In addition to reviewing detection methods, this paper introduces and analyzes various datasets that are commonly used to evaluate the performance of experimental models. Datasets are crucial for benchmarking model effectiveness, yet issues such as insufficient data volume and class imbalance pose ongoing challenges. The paper discusses various solutions proposed by recent research to address these problems, including techniques for augmenting limited data and strategies for balancing skewed datasets. These solutions are crucial for improving the reliability, scalability, and generalizability of deep learning-based malware detection models across various platforms and contexts. Finally, this paper delves into the current challenges faced by imaging-based malware detection and classification methods. These challenges include handling imbalanced datasets, improving model interpretability, addressing adversarial attacks, and mitigating insufficient feature representation. The paper also provides an outlook on future research directions, emphasizing the importance of continuous advancements to address the evolving threat landscape.

Keywords　malware; detection and classification; obfuscation techniques; deep learning; imaging-based methods; dataset

1　引　言

恶意软件作为一种具有恶意目的的软件程序，旨在非法入侵并危害计算机系统、盗取个人敏感信息、破坏隐私数据，甚至影响计算机和网络的正常运行。恶意软件不仅攻击手段复杂多样，还具备快速衍生变种的能力，致使其逐渐成为影响现代网络信息安全的主要威胁之一[1]。

近年来，恶意软件攻击频率和规模在全球范围

内出现持续增长的趋势①。同时，恶意混淆技术的不断发展和自动化代码编写工具的广泛应用[2]，使得恶意软件新变种在数量上呈指数级增长，给现有的防御机制带来严峻挑战②。2015～2022 年恶意软件总量及其年增长率曲线如图 1 所示③。由图 1 可见，恶意软件总量在此时期内逐年递增，且年增长率整体保持在较高水平。因此，数量快速增长的恶意软件仍然是当前严重影响网络信息安全的一个重要因素[3]，如何高效检测并清除恶意软件新变种，已成为网络安全领域亟需解决的热点问题。

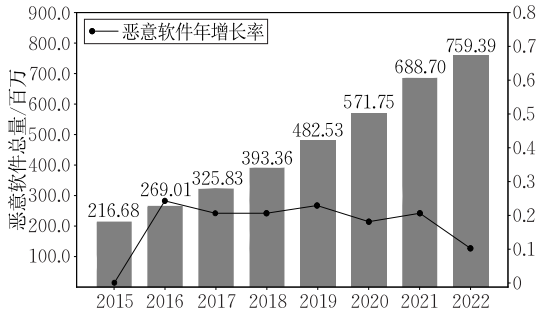

图 1 2015～2022年恶意软件总量及增长率曲线

目前，常见的针对恶意软件的检测分类方法主要包括静态分析方法、动态分析方法、基于机器学习的检测分类方法等[4]。静态分析方法是一种在不运行恶意软件样本的情况下对代码或二进制文件进行分析的检测方法。该方法虽然无法提供关于二进制源文件的所有相关信息，但能够快速获取恶意软件样本中的关键结构属性并完成检测分类[5-6]。但该方法的缺陷在于容易受到例如加密、代码变形和 API 调用等[7]混淆技术的干扰。动态分析方法是一种在受控环境下运行恶意软件样本，并利用预先安装的进程监控行为特征的检测方法。由于提取动态特征环节需要投入大量资源[8]，因此该方法需要较高的时间和人工成本[9]。基于机器学习的检测方法可以通过自动提取恶意软件样本特征信息实现高效检测，但该方法高度依赖人工提取特征和训练数据的质量，对应用复杂混淆和加壳技术的恶意软件检测效果较差[10]。

随着反检测技术和混淆技术的发展，恶意软件新变种不仅可以有效规避常见检测分类方法的识别[11]，还能通过插入冗余代码和改变程序结构等手段掩盖真实意图[12]，通过多个渠道在网络中进行传播并攻击目标用户。在此背景下，随着深度学习技术在图像识别领域取得的突破性进展，将恶意软件的二进制源文件转换为图像，利用深度学习模型对转换后的图像进行特征提取与分类的检测方法，已成为应对恶意软件混淆技术的有效手段之一。

与传统检测方法相比，图像化检测方法在多个方面表现出显著优势。首先，该方法能够以图像的形式直观展示出恶意软件中复杂且隐蔽的代码结构特征，进而通过神经网络模型分析此类特征，更准确地识别分类恶意软件。其次，该方法无需领域专家对特征进行手动定义，极大降低了对时间和人力资源的依赖，从而显著提高检测过程的自动化程度。最后，该方法具有较强的适应性和泛化能力，在面对高度混淆、加密的恶意软件[13]或未知攻击场景时[14]，图像化检测方法仍能保持较高的检测准确率。

在恶意软件检测的发展进程中，已有多篇文献对该领域的研究进行了全面总结分析[15-19]，主要涵盖静态分析、动态分析、基于机器学习和深度学习的恶意软件检测方法。这些研究探讨了各类方法在检测已知恶意软件、应对常规攻击方面的有效性，分析了各类方法如何改进并提升恶意软件检测的精度和效率。但是，现有文献大多侧重于对已有检测方法的简单总结，并未涉及针对恶意软件高级混淆技术的应对方法与技术现状分析。如文献[15，18-19]均未明确指出现有方法是否能够应对混淆类恶意软件的问题。此外，现有文献也未涉及恶意软件检测研究常用数据集的分析、不同场景下的应用效果评估等，对未来研究方向和发展趋势的总结不够全面。

本文聚焦于图像化方法在恶意软件检测分类中的研究，通过梳理图像化方法的研究进展，深入分析该方法在检测应用高级混淆技术的恶意软件变种时存在的优势。同时，本文系统总结了图像化检测研究中常用的数据集，并深入分析了数据规模和数据分布对实验结果造成的影响。此外，本文讨论了在不同平台（如 Windows、Android、IoT）特有的文件结构、数据格式和混淆技术下，图像化检测方法的通用性和有效性。最后，本文在总结目前研究成果的基础上，针对实际应用中面临的挑战和技术瓶颈，提出包括概念漂移、可解释性研究在内的具体的未来研究方向，以期为推进图像化方法在恶意软件检测领域的应用和发展提供一定的参考依据。

本文第 2 节概述恶意软件的类别与定义，分析常见的混淆技术；第 3 节总结传统恶意软件检测与分类方法的研究进展，并分析其局限性；第 4 节详细

① https://www.kaspersky.com.cn/about/press-releases/enterprise/

② https://www.symantec.broadcom.com/hubfs/Symantec_Ransomware_Threat_Landscape_2024.pdf

③ https://portal.av-atlas.org/

总结分析基于图像化方法的恶意软件检测与分类研究进展；第 5 节总结目前研究面临的挑战，并对未来研究做出展望；第 6 节对全文进行总结与回顾。

2　恶意软件分类与常见混淆技术

2.1　恶意软件类别与定义

恶意软件是指运行在用户计算机或其他终端上，在用户不知情或未允许的情况下，利用操作系统或应用程序中存在的漏洞侵害用户合法权益的软件程序。因此，恶意软件对主机完整性、互联网可用性和用户隐私安全性造成严重威胁。

常见的恶意软件包括病毒、蠕虫、木马、后门、Rootkit 和僵尸网络（Botnet）等[20]，各种类型的恶意软件定义如下，不同种类恶意软件之间的区别如表 1 所示。

表 1　各类恶意软件之间的差异

类别	病毒	蠕虫	特洛伊木马	后门	Rootkit	Botnet
存在形式	寄生	独立实体	伪装为其他正常文件	伪装为其他正常文件	伪装为其他正常文件	伪装为其他正常文件
传播模式	通过主机文件或媒体	自我复制	通过欺骗手段	通过欺骗手段	通过欺骗手段	通过欺骗手段
攻击目标	本地文件	网络主机或网络自身	操作系统	操作系统	操作系统	操作系统
主要风险	系统损坏、文件、数据丢失	网络瘫痪、数据丢失	信息泄露	信息泄露	信息泄露	信息泄露、系统损坏
传播速度	快（每小时传播 100～999 台）	很快（每小时传播 >1000 台）	慢（每小时传播 <100 台）	不传播	快（每小时传播 100～999 台）	很快（每小时传播 >1000 台）
检测效率	高效（检测率>99%，误报<0.1%）	低效（检测率<90%，误报>1%）	中等（检测率90%～99%，误报0.1%～1%）	低效（检测率<90%，误报>1%）	中等（检测率90%～99%，误报0.1%～1%）	中等（检测率90%～99%，误报0.1%～1%）
感染传播是否需要人为干预	是（需用户打开邮件或下载软件）	否（可通过网络传播）	是（软件下载时需用户授权）	是（安装时需用户授权）	是（植入时需用户授权）	是（需用户点击恶意链接）

病毒：一种通过硬件或软件进入计算机系统并附着在程序文件上的恶意程序，通过执行程序设定的恶意任务，破坏用户数据、软件内容，发起拒绝服务攻击等。

蠕虫：一种未经用户许可潜入计算机系统，利用用户的电子邮件地址传播感染，对网站和用户计算机造成毁灭性影响的恶意程序。

特洛伊木马：特洛伊木马是伪装为合法软件的有害软件，用户多在未知情况下下载并执行此类软件。木马在激活后会多次攻击用户主机，并通过删除文件、窃取数据、激活和传播其他恶意软件，破坏主机。

后门程序：后门程序是一种远程管理应用程序。一旦安装在计算机上，后门即允许攻击者通过互联网访问和控制主机。后门通常利用系统代码中未记录的进程获得对系统的控制。同时，攻击者也经常利用后门在用户不知情或未经授权的情况下，控制用户机器。

Rootkit：一种特殊类型的恶意软件，将特定文件和进程，以及网络链接隐藏在受感染设备中，并在运行时加载特殊的驱动程序，修改操作系统内核与用户数据，实现破坏目的。

Botnet：一种允许攻击者远程控制受损设备的恶意软件。攻击者常利用其发动分布式拒绝服务（Distributed Denial of Service，DDoS）等大规模网络攻击。

2.2　常见恶意软件混淆技术

为规避现有防御机制的检测，恶意软件开发者不断更新混淆、加壳等反检测技术，这些技术通过隐藏恶意软件的真实意图或功能，增加检测和分析的复杂性，导致安全威胁持续加剧。此外，混淆技术还能够掩盖恶意软件的代码结构，通过动态执行、加密和虚拟化手段，使得传统检测方法逐渐失效[21]。

具体而言，常见的恶意软件混淆技术包括以下几种：

（1）加壳技术：通过对恶意软件的可执行文件进行压缩或加密，以此隐藏其真实代码，从而规避检测的混淆技术。由于加壳技术可以有效地混淆恶意软件代码结构，使其难以被静态分析工具直接识别，因此被广泛应用于各类恶意软件中。

（2）变形技术：也称为多态技术，通过直接对恶意代码体进行混淆操作，以此规避检测软件。变形的恶意代码通常由代码体和变形引擎两个部分组成。

（3）虚拟机技术：通过检查硬件特征、系统行为

等,识别虚拟化环境,从而绕过检测工具的侦查,增加检测难度。恶意软件利用虚拟机检测技术规避安全分析,当其被检测到运行于虚拟机或沙盒中时,会停止或改变恶意行为,以逃避动态分析与沙盒检测。

基于静态分析和动态分析的检测方法通常依赖于分析固定的代码特征和执行路径,无法有效应对恶意软件新变种复杂的加壳、变形以及虚拟机等混淆手段。与传统检测方法相比,基于图像化的恶意软件检测分类方法在应对复杂混淆技术时具有显著的优势,通过将恶意软件二进制源文件转换为图像形式,可以有效避免混淆手段对代码特征的遮蔽作用,从而揭示出传统检测方法难以捕获的独特特征[22]。在应对代码打包和变形等混淆技术时,图像化方法能够通过提取和分析二进制文件的视觉特征,有效识别经过变形处理的恶意软件,从而绕过混淆技术对代码特征的掩盖,显著提高检测分类的准确率[23]。该方法的核心优势在于,即使恶意软件经过多层加密或复杂的变形处理,其底层特征在被转换为图像的过程中仍然能够被检测系统准确识别和捕捉,从而揭示出恶意软件潜在的行为模式[24],这是传统检测方法所难以实现的。

3 传统恶意软件检测与分类方法

随着混淆技术的不断更新,恶意软件逐渐将攻击目标转向操作系统的内核层,极大增强了恶意行为的隐蔽性与复杂性[25]。与早期的恶意软件相比,恶意软件新变种表现出更强的对抗性和逃逸能力[26]。本节将简要讨论基于静态分析、动态分析和机器学习的恶意软件检测分类方法,分析上述方法存在的优势与不足。

3.1 基于静态分析的检测分类方法

静态分析是指在不运行恶意软件样本的情况下,通过反汇编和其他技术提取恶意软件静态特征进行检测的方法。该方法常用的特征包括代码程序结构、可移植的可执行文件(Portable Executable,PE)、指令序列和应用程序编程接口(Application Programming Interface,API)调用。早期的静态分析方法通过提取字节标识符和字符串信息等特征进行检测分类,其中 Tian 等人[27]提出一种基于字符串信息的恶意软件检测分类方法,将恶意软件样本中包含的可打印字符串信息输入到多种经典分类算法中进行实验,结果表明,该方法检测分类准确率最高可达 97%,具有良好的检测效果。

Feng 等人[28]提出一种静态 Android 恶意软件检测方法,通过深入分析 Android 应用程序的数据流和控制流属性,可以有效识别应用程序是否属于已知的恶意软件家族,实验结果表明,该方法能高效完成恶意软件检测分类任务,并且其签名对各种混淆技术具有一定的鲁棒性。然而,文献[27-28]所提方法容易受到模糊处理、加密打包等高级反检测技术的影响,恶意软件开发者可以利用这些方法轻易规避检测。同时,上述方法提出时间较早且检测效率较低,不适用于实时恶意软件检测。

为缓解加密打包等混淆技术对检测工作带来的影响,Tian 等人[29]提出一种基于代码异质性特征的 Android 重打包恶意软件检测方法,通过分析恶意软件样本与原始应用程序之间的代码相似性,利用代码异质性特征识别重打包行为。实验结果表明,该方法可以有效完成针对重打包类恶意软件的检测分类任务。此外,Zou 等人[30]提出一种基于 API 亲密度分析的 Android 恶意软件检测分类方法,通过分析应用程序中不同 API 调用之间的亲密度关系,构建亲密度模型识别隐藏的恶意行为和异常 API 调用模式。然而文献[29-30]所提方法虽然能有效识别加密打包的恶意软件,但无法准确检测出应用高级动态代码生成和反射调用等复杂混淆技术的恶意软件新变种。

3.2 基于动态分析的检测分类方法

动态分析是一种使用自动化检测工具,在隔离和安全的环境中运行恶意软件,并在执行过程中监视和捕获程序执行轨迹的检测方法[31]。尽管恶意软件变体众多,且部分变种的代码结构与原始样本也有所不同,但其核心行为模式仍然具有一定的相似性。因此,动态分析可以通过捕捉恶意软件运行时的行为特征,有效完成检测分类任务[32]。

常用的动态分析方法包括动态行为监测、API 调用分析和网络流量分析等。Kim 等人[33]在受限环境中运行恶意软件样本,通过比较 API 运行过程中时序行为之间的相似程度,实现恶意软件的高效检测。Babu 等人[34]采用 API hooking 捕获恶意软件在虚拟环境中的行为,监控其系统调用、文件修改和网络活动,获取恶意软件的动态行为特征进行有效检测。文献[33-34]所提方法通过分析恶意软件动态行为特征,显著提高了检测分类准确率。但上述方法开发成本较高,分析过程会消耗较多的时间和资源,难以有效应对现有研究普遍使用的大规模数据。

除动态分析常用 API 序列外，使用网络流量进行检测也是一种有效手段。Wang 等人[35]提出通过传输控制协议（Transmission Control Protocol，TCP）进行数据传输，并将收集到的网络流量传输到服务器进行集中数据分析，最后结合机器学习分类算法对处理后的流量数据进行检测分类。实验结果表明，该方法可以有效区分不同类型的恶意流量数据。虽然该检测模型在识别未知类型恶意软件时表现效果良好，但仍有大量包含隐藏恶意行为的网络流量无法被发现。

为解决上述问题，Singh 等人[36]提出一种基于行为异常的恶意软件检测分类方法，提取恶意软件运行时的行为特性，并通过对提取到的动态特征进行分组分类的方式，分析其恶意行为，有效区分良性和恶意软件。尽管动态分析能够有效捕获多数恶意软件运行时的行为特征，但其在面对更复杂的新变种时仍然存在难以准确捕获运行行为特征等局限。例如，沙箱规避技术的出现致使恶意软件新变种能够在受控环境中隐藏其恶意行为以逃避检测[37]，这也是影响动态分析检测准确率的一个重要原因。

动态分析侧重于在安全环境中执行恶意软件并监测其行为，并深入地了解恶意软件内在行为特征。然而，受复杂混淆技术和规避技术的影响，单纯依赖动态分析得到的检测结果可能会导致较高的误报率和漏报率。而图像化方法能够补充动态分析无法获取的，包含加密和混淆代码的视觉模式，通过结合使用动态分析和图像化检测方法的方式，可以进一步提高恶意软件检测分类的全面性和准确性。

表 2 对基于静态分析和动态分析的经典检测分类方法进行了全面总结，包括各种检测分类方法所用分类算法、可检测的恶意软件类型、抗混淆能力、实验所用数据集与数据规模、可完成的检测任务和检测分类准确率等。通过分析表 2 内容可知，传统检测方法普遍应用于恶意检测而非家族分类，分析其原因为此类方法无法有效捕获多类家族间细微的特征差异，导致在多分类任务场景中会出现检测准确率下降等问题。

表 2　传统恶意软件检测方法总结

现有研究工作	文献年份	分析手段	所用检测/分类算法	抗混淆能力	所用数据集和规模	所检测的恶意软件类型	检测任务	检测/分类准确率/%
Tian等人[27]的工作	2009	静态分析	最近邻算法、统计算法和 AdaBoost	较弱	非公开自收集数据集（约 15 000 个数据样本）	PE	家族分类	97.00
Feng等人[28]的工作	2014	静态分析	组间调用图	较弱	非公开数据集，Google Play（约 20 000 个数据样本）	APK	恶意检测	97.60
Tian等人[29]的工作	2017	静态分析	类级依赖图 CDG	较强	Genome，VirusShare，Benign Apps（共计 4572 个 Android 应用）	APK	恶意检测	97.04
Zou等人[30]的工作	2021	静态分析	API 调用敏感度中心分析	较强	IntDroid（共计 8253 个数据样本）	APK	恶意检测	99.10
Kim等人[33]的工作	2019	动态分析	相似性计算，序列比对算法 MSA	一般	malshare.com 和 VXVolt.net（共计 1790 个恶意软件样本）	PE	恶意检测	95.01
Wang等人[35]的工作	2019	动态分析	决策树和支持向量机	一般	Drebin 数据集（共计 5560 个数据样本）	APK	恶意检测	97.89
Singh等人[36]的工作	2020	动态分析	DT算法，随机森林RF，Adaboost 和梯度提升	较强	VirusShare 和 VirusTotal 恶意软件数据集中收集（约 25 000 个数据样本）	PE	恶意检测	99.54
Babu等人[34]的工作	2024	动态分析	基于 Transformer 的 Longformer 分类器	较强	Datacon2019（约 60 000 个数据样本）	PE	恶意检测	99.20

3.3　基于机器学习的检测分类方法

与基于签名规则或特征匹配的静态分析和动态分析方法相比，基于机器学习的恶意软件检测分类方法能并行处理大规模恶意软件数据，自动挖掘有效特征并完成检测分类任务，因此具有较强的适应性和泛化能力[38-39]。Huda 等人[40]提出一种基于半监督学习和无标签数据的检测框架，用于检测网络系统中的未知恶意软件攻击。通过聚类和伪标签生成，分析物理信号和网络流量特征，提高恶意软件检测精度。此外，Martín 等人[41]提出利用马尔可夫链对序列状态之间的转移概率进行建模，构建转移概率特征空间以用于训练机器学习模型，实验结果表明，该方法在面对混淆恶意软件时能表现出较强的适应性。然而，文献[40-41]所提方法在处理大量数据样本时计算复杂度较高，对高质量数据的依赖也较为显著。

考虑到恶意软件运行时存在的序列特征信息，Acarturk 等人[42]提出一种基于长短期记忆网络（Long Short-Term Memory，LSTM）的检测方法，通过分析恶意软件的运行轨迹输出，捕捉其在执行

过程中的复杂时序行为，可以有效提升检测分类准确率。然而，该方法在面对长时间运行轨迹时的计算资源需求较高，增加了训练和推理的时间成本。Kumar 等人[43]提出一种融合图像分析和机器学习的检测方法，将恶意软件样本的图像纹理特征与多个机器学习算法相结合，通过堆叠模型提高恶意软件家族的识别精度，实验结果表明，该方法在处理复杂多类恶意软件样本时能够表现出较强的分类能力。

通过对文献[42-43]进行分析总结可知，尽管基于机器学习的检测分类方法能够捕捉传统检测方法难以发现的规律和特征，但实验模型的检测性能高度依赖于特征提取的质量，并且需要较高的计算成本。表 3 对近年来基于机器学习的经典恶意软件检测分类方法进行了总结，包括分析手段、所用分类算法、抗混淆能力、所用数据集及规模、可检测的恶意软件类型、检测任务及准确率等。

表 3 基于机器学习的恶意软件检测方法对比总结

现有研究工作	文献年份	分析手段	所用分类算法	抗混淆能力	所用数据集及规模	可检测的恶意软件类型	检测任务	检测/分类准确率/%
Huda 等人[40]的工作	2017	结合静态分析以及机器学习	支持向量机 SVM，随机森林 RF，决策树 J48	一般	CA Technologies VET Zoo 实验数据(-)	PE	恶意检测	81.80
Martín 等人[41]的工作	2018	结合动态分析以及机器学习	CANDYMAN，马尔科夫链	很强	Drebin 项目中收集的公开数据集（约 14000 个数据样本）	Script	家族分类	99.90
Wang 等人[35]的工作	2019	结合动态分析以及机器学习	-	较强	Drebin 项目中收集的公开数据集（约 14000 个数据样本）	APK	恶意检测	97.89
Acarturk 等人[42]的工作	2021	结合动态分析以及机器学习	长短期记忆 LSTM	较强	VirusShare 网站收集(-)	PE	恶意检测	99.26
Kumar 等人[43]的工作	2022	机器学习	KNN，支持向量机 SVM，随机森林 RF，朴素贝叶斯	较强	Malimg 数据集以及真实世界收集（9339 个数据样本）	PE	家族分类	98.23
Hossain 等人[25]的工作	2024	结合静态分析以及机器学习	堆叠集成学习框架	一般	CIC-MalMem-2022（共计 58596 个数据样本）	APK	恶意检测和家族分类	99.92/83.53
Madamidola 等人[38]的工作	2024	结合静态分析以及机器学习	随机森林 RF	较强	CIC-MalMem-2022（共计 58596 个数据样本）	APK	恶意检测	99.80

通过分析表 3 内容可知，结合静态分析、动态分析和不同机器学习算法的检测分类方法，在应对复杂、混淆恶意软件时能够展现出独特优势，实验所用数据集不仅涵盖广泛的恶意软件类型，如 APK、PE 和脚本类恶意软件，还从公开数据集和实际应用场景中收集大量真实未标记样本，从而确保实验结果的有效性与广泛适用性，避免针对特定数据的过拟合现象出现。然而，现有的机器学习算法在面对恶意软件新变种时，泛化能力和鲁棒性仍有待提高[44]。在此背景下，图像化方法因其具备较强的抗混淆能力，并且在处理大规模数据样本时具有较高的效率，为提升恶意软件检测性能提供了新的途径。

4 图像化的恶意软件检测与分类方法

4.1 早期的图像化检测技术

早期的图像化方法通常依赖于辅助工具，如文本编辑器和二进制编辑器，以便对二进制数据进行图像化处理。之后，基于图形用户界面(Graphical User Interface，GUI)的工具逐渐发展成型，在一定程度上推动了图像化检测方法的研究与发展。然而，这些技术在该领域的应用仍然较为有限。其中，Kancherla 等人[45]通过将可执行文件转换为字节图，提取低级纹理特征以进行进一步的检测分析工作。Ban 等人[46]通过分析恶意软件二进制图像样本的指纹特征信息进行快速指纹匹配，返回视觉上最为相似的变体。Han 等人[47]通过分析自动化恶意软件检测工具可能会带来的模块重用情况，检测恶意软件变体，将二进制文件转换为灰度图像和熵图以检测变体家族的相似性。然而文献[45-47]所提出的方法只是对图像化方法的初步应用，没有考虑如何分析图像所包含的深层特征。

随着研究的深入，使用 RGB 彩色图像代替灰度图像进行恶意软件检测分类的实验逐渐成为热点。

例如，Han 等人[48]将从恶意软件中提取到的二进制代码段映射为彩色图像，并使用经典的图像分类方法对不同恶意软件家族进行检测分类。然而，该方法仅限于识别原始的二进制片段，未能从更宏观的角度对恶意软件进行全面审查与分析。尽管上述方法存在一定不足，但这些研究仍然为利用图像化方法解决恶意软件检测分类问题提供了坚实的理论基础。

4.2　针对图像化过程的研究

基于图像化方法的恶意软件检测分类研究中，首要环节就是将二进制代码段、PE 文件、.dex 文件、ELF 文件和网络流量数据等转换为图像进行表示[49]。当前的各类恶意软件可以按照其文件结构分为 Windows 操作系统恶意软件、Android 移动操作系统恶意软件、物联网设备恶意软件和 Linux 系统恶意软件。在多平台环境下，恶意软件检测的有效性极大地依赖于对特定文件结构的深入理解与分析。图 2 展示了图像化方法常用的检测文件结构类型，其中主要包括 PE 文件、API 调用序列、.dex 文件和 ELF 文件等。

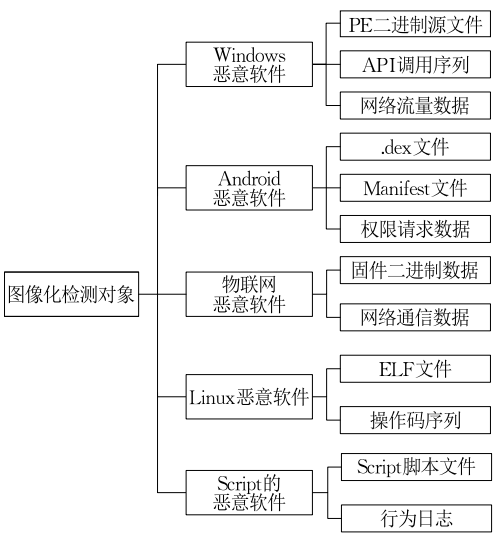

图 2　常用的图像化恶意软件检测文件结构类型

Nataraj 等人[50]首次提出将字节序列映射为灰度图的图像化转换方法，由于恶意软件可执行文件可以表示为由 0 和 1 组成的二进制字符串，因此可以将给定的恶意软件二进制文件读取为 8 位无符号整数向量，并组织成[0,255]的二维数组，根据数值将其映射为灰度像素值。其中，图像的宽度是固定的，高度允许根据文件大小而变化。具体过程如图 3 所示。

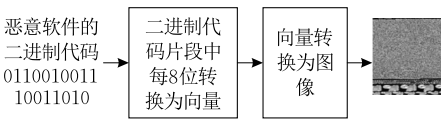

图 3　二进制代码段转换图

4.2.1　针对 Windows 恶意软件的图像化过程研究

针对 Windows 平台下的恶意软件，研究人员常通过将 PE 文件、二进制数据以及 API 指令序列转换为图像的方式，捕捉样本文件结构的特征。其中，Yu 等人[51]提出一种基于奇异频谱变换（SpecView Singular Spectrum Transformation，SST）的光谱可视化方法，从 PE 文件中提取时间序列信号特征转换为一维时间序列频谱数据，之后将其映射在时间序列图中完成可视化。该方法可以解决多数传统图像化检测方法中存在的图像转换效率较低、调整图像尺寸时丢失图像纵横比等共性问题。但是该方法的缺陷在于整个流程对计算资源的消耗较大，且计算成本较高。

在此之后，Moreira 等人[52]提出一种基于便携式可执行文件头信息的勒索类恶意软件检测方法，通过对 PE 文件的标记信息进行处理分析，提取 PE 文件标记序列，并按照其顺序模式生成二维矩阵，使用 Seaborn API 提供的连续光谱反向调色板将生成的矩阵转换为图像。实验结果表明，该方法生成的彩色图像能够更加直观地展示恶意软件的数据特征，在勒索类恶意软件检测任务中能取得较高的检测精度。然而，将所有特征信息映射在单一通道中往往会导致部分特征的丢失，因此，通过多通道技术生成彩色图像逐渐成为近年来研究的热点。

Deng 等人[53]选择对反汇编得到的.asm 文件进行研究，通过提取操作码中字母与数字的唯一转移概率构建马尔科夫状态转移矩阵，并将其映射在图像的三个通道中生成马尔科夫图。实验结果表明，与常用图像化方法相比，该方法在恶意软件家族检测分类任务中能表现出极高的检测准确率和分类能力，尤其是在处理不同类型的恶意软件样本时具有较强的鲁棒性。然而，计算马尔科夫状态转移矩阵增加了处理和分析的复杂性，因此该方法需要较高的计算资源和专业知识，在处理大规模数据集时，可能会出现性能效率下降的问题。与 Deng 等人的方法类似，Tang 等人[54]通过处理反汇编得到的.bytes 文件以及.asm 文件，提取局部特征信息、汇编指令集信息 Opcode 序列和可见字符信息 String 序列，并使用 MinHash 转换算法[55]将提取到的特征序列映射为三通道 RGB 彩色图像。这种方法的创新之处在于，通过结合多个数据维度以提供一种更全面的视角分析和识别恶意软件。然而，文献[53-54]所提方法均高度依赖于数据预处理的精度和特定的数据特征提取处理方法，同时需要保证

训练数据的多样性和大规模性以避免出现过拟合现象。

为解决上述问题，Li 等人[56]提出一种通过大核卷积网络和三通道映射进行恶意软件检测分类的图像化检测方法，将恶意软件的不同数据维度映射到图像的三个通道，通过多通道视觉化和深度学习，使得模型能够自动提取恶意软件的关键特征并进行高效分类，有效地提升了模型的分类精度和推理效率。Xuan 等人[57]提出一种结合序列信息的图像化检测分类方法，将字节序列信息映射为 RGB 图像，保留二进制源文件关键特征信息并增强图像的特征表达能力，可以有效提高恶意软件分类的准确率。然而，文献[56-57]所提方法高度依赖于图像处理和分析技术的精度，如果图像分类算法不够精确，会严重影响实验最终的检测效果。

文献[51-57]所提到的图像化方法均针对 Windows 平台恶意软件的 PE 文件所展开，由于 PE 文件包含了关于可执行文件的关键元数据，并且其结构特征在图像化后更容易被识别，因此多数研究选择分析 PE 文件进行检测。但将 PE 文件的章节分布信息转化为图像会丢失部分重要的特征信息，且该方法易受到打包技术的影响。此外，该方法通常会生成大量的特征图像，容易引发维度灾难，从而导致模型过拟合，且其训练过程需要投入大量的计算资源和时间，整体成本较高。

针对 PE 文件的研究主要关注恶意软件样本的静态特征，无法充分揭示恶意软件在执行过程中的动态行为。为了更全面地理解和检测恶意软件，部分研究开始关注恶意软件 API 调用序列，通过将 API 调用序列可视化，深入分析恶意软件的行为模式。Tang 等人[58]通过在沙箱中动态运行恶意软件，捕获其行为信息，提取 API 调用序列及运行时间信息，并根据制定的颜色映射规则可视化不同类别的 API 指令，生成特征图像。Barona 等人[59]对不同的 API 调用指令分配不同的颜色，通过计算 API 调用的频率和顺序，更改颜色强度并生成像素值矩阵，最终将矩阵映射为彩色调用图。然而文献[58-59]所提出的方法仅仅是对 API 指令信息的简单映射，没有深入挖掘不同指令间的隐含关系。

受 Tang 等人和 Barona 等人工作的启发，Yang 等人[60]提出基于 API 指令序列重组的检测分类方法。在恶意软件运行期间，将相同类型的 API 分组到 API 块中，根据每种 API 类型的第一次调用顺序对每个 API 块进行重组，根据重组后的 API 指令序列提取 API 代码、API 奉献度和 API 顺序索引，生成特征图像。实验结果表明，该方法通过捕捉运行时的 API 调用模式并将这些模式转换为可分析的图像，可揭示被忽视的内在复杂行为特征，显著提高了恶意家族的检测分类准确率。但该方法对原始 API 指令序列存在一定破坏，因此会导致部分调用序列中相关语义信息丢失，影响最终检测效果。

4.2.2 针对 Android 恶意软件的图像化过程研究

随着移动设备的广泛使用，装配 Android 系统的移动通信设备也是攻击的热门目标。Android 恶意软件通常存留在 APK 文件中，其中包含应用程序的全部必需组件，如.dex 文件、资源文件、Manifest 文件等。通过分析此类文件，可以利用图像处理技术挖掘 Android 应用程序的深层特征，从而有效地检测和分类恶意软件。Tang 等人[61]提出一种基于多粒度操作码特征的 Android 恶意软件混淆变体检测方法，通过提取 Android 恶意软件操作码的多粒度特征构建完成操作码频率矩阵，将操作码序列转化为图像表示，实现操作码序列特征到灰度图像的转换。然而，该方法对特征提取和选择的依赖较强，并且特征维度的增加可能导致模型复杂性和计算成本的大幅增加，影响最终的检测效率。

此后，Xiao 等人[62]通过解压缩 Android 恶意软件 APK 包获取.dex 文件，将其中的二进制格式字节码转换为十六进制数并进行连续读取，最后将读取到的十六进制数分别映射在 RGB 图像的三个通道中，生成彩色图像。与 Xiao 等人的方法类似，Zhu 等人[63]同样选择对.dex 文件进行分析，通过对.dex 文件进行切割，保留 index 段并将裁剪后的 index 段转换为由十六进制数组成的数组，按指定的数组读取规则生成像素值，最终映射为 RGB 彩色图像。与文献[61]所提方法相比，文献[62-63]专注于分析.dex 文件内的字符信息并将其映射为 RGB 图像。然而，这种图像化方法容易受加壳混淆技术的影响，致使生成的图像丢失原有的特征信息。

为解决混淆技术带来的影响，Fang 等人[64]在对.dex 文件进行解析时，通过读取文件的字节排列信息，计算熵矩阵和字节类型比例矩阵，并将其映射为图像进行检测。实验结果表明，这种图像化方法对混淆.dex 头干扰技术具有较强的鲁棒性，并且所生成图像包含的特征更为全面。此外，Lachtar 等人[65]使用融合空间填充曲线的图像化方法，通过

从.dex 文件中提取 Android 应用程序原生指令，利用希尔伯特空间曲线，将与应用程序相关的指令映射到二维图像不同像素位置中，生成不同配色方案的图像。这种方法生成的图像依赖于香农熵的计算，有利于反映深层的恶意行为。然而文献[58-59]所提方法会导致图像失去部分原始细粒度信息，最终影响检测分类准确率。

不同于之前大量针对.dex 文件的研究，Tasyurek 等人[66]将目标转向 manifest.xml 文件，通过提取 manifest 文件中的静态特征属性构建特征向量，并将特征向量转换为归一化填充矩阵，最终映射为黑白格式的 QR 码图像。实验结果表明，与传统图像化方法相比，该方法所需内存空间较小，时间成本较低。然而，原始特征的细微差异在此过程中可能无法得到充分表达，导致重要特征信息丢失，最终影响检测分类准确率。

4.2.3　针对物联网及 Linux 恶意软件的图像化过程研究

除 Windows 和 Android 这两个主流平台外，Linux 和物联网(Internet of Things，IoT)设备也同样面临着恶意软件的威胁。这些系统通常在商业和工业环境中扮演着关键角色。鉴于这些平台在架构、操作特性及安全环境方面与 Windows 和 Android 存在显著差异，因此研究针对此类平台恶意软件的特定图像化方法尤为重要。

针对 IoT 恶意软件，Ahmed 等人[67]选择将二进制源文件转换成由矩阵组成的 8 位无符号整数序列，根据矩阵内的数值元素映射为灰度图像，其像素值范围在 0 到 255 之间。此后，Smmarwar 等人[68]提出对传统的图像化方法进行改进。首先，从原始二进制数据中提取特征并将其转换为灰度图像，之后，为提高检测分类的精度，将图像尺寸标准化并进行着色处理。这种方法通过颜色填充强化了图像间的视觉相似度，最终可提高检测分类性能。然而文献[67-68]所提方法主要关注 IoT 恶意软件的静态特征，难以捕捉其动态行为和执行时特性，容易导致高误报率的出现。

为解决此类问题，Alsubai 等人[69]在将 IoT 恶意软件转换为图像后，经过预处理调整尺寸大小，利用 Harris Hawks 进行优化并生成 RGB 图像。采用融合滤波器对图像进行模糊处理以去除噪声，并且调整不均匀的像素值消除色差。实验结果表明，经过处理后的图像适用于 IoT 恶意软件检测，检测效率较传统检测方法有明显提高。但该方法在面对新变种或复杂的恶意软件时泛化能力较差。Ghahramani 等人[70]针对 IoT 设备的在线恶意软件检测问题，将动态恶意软件行为转化为稀疏的二进制图像并进行检测，实验结果表明，该方法可以有效提高检测的准确率和召回率，特别是在处理资源受限的 IoT 设备时提升效果显著。同样，由于 IoT 设备的动态特性，该方法模型更新较为困难，在面对新变种时的泛化能力较差。

针对 Linux 平台下的恶意软件，Landman 等人[71]通过对可执行与链接格式标头(Executable and Linkable Format，ELF)进行切片操作，以减小转换过程中生成的易失性转储文件的尺寸，之后将原始文件表示为八位无符号字节序列并送入缓冲区，这些缓冲区中的数据被转换成 ARGB 数组，进一步映射为可视化恶意图像。实验结果表明，使用该方法生成的图像包含更丰富的特征信息。然而这种图像化方法涉及多个处理步骤，分析的复杂性较高，同时需要较高的计算资源和专业知识。

本节对不同操作系统平台下恶意软件的图像化方法进行了全面的分析和总结。尽管每种平台的恶意软件特性各异，但图像化方法为揭示恶意软件的内在特性提供了一个通用且有效的框架，通过将二进制源文件转换为图像，使得原本隐蔽的代码结构和行为模式得以直观地展示。表 4 对本节提出的图像化方法进行了全面总结，表格中内容包括现有研究研究所用样本文件结构、所用的图像化方法、提取特征类型、生成图像类型及不同研究方案的优缺点等。

通过分析表 4 内容可知，针对不同操作系统平台的恶意软件，现有研究采用了多种图像化方法进行检测分类实验，并且展现出较为优秀的检测效果与显著的性能优势。然而，尽管图像化方法能够在一定程度上揭示隐藏的模式和结构，但其仍然存在二进制文件信息提取不充分、动态行为分析不全面等不足。例如 Yu 等人[51]的研究通过使用基于时间序列频谱数据的图像化方法，简化特征提取的计算成本，显著提高了检测分类效率。但该方法只能处理一维的时序数据，无法完整展示恶意软件的全局行为模式。Li 等人[56]的研究通过多通道图像融合技术，从多角度有效地提取并融合恶意软件图像特征，但该方法忽略了恶意软件运行时的动态行为特征。

表 4 针对不同图像化方法的研究总结

现有研究工作	文献年份	恶意软件所属平台	样本文件结构	研究所用图像化方法	提取特征类型	生成图像类型	研究方案的优点	研究方案存在的局限性
Yu等人[51]的工作	2021	Windows	PE	映射时间序列频谱数据	时间序列特征	灰度时间序列图	仅采用一维时序频谱数据，成本较低	无法直观展现恶意软件全部特征信息
Tang等人[54]的工作	2022	Windows	PE	挖掘映射签名矩阵并转换为对应的图像	局部特征信息、汇编指令集信息	基于特征矩阵的多通道映射图像	从多角度获取恶意代码特征，实现图像融合	所使用的数据集与现实世界的恶意代码不同
Moreira等人[52]的工作	2023	Windows	PE	将PE头文件映射为顺序矢量模式像	PE头中原始结构信息特征	顺序矢量彩色图	简化特征提取与处理负载	仅针对勒索软件一个类
Deng等人[53]的工作	2023	Windows	PE	计算操作码转移概率并映射为马尔科夫图	操作码字母、数字唯一转移概率	三通道马尔科夫彩色图	保留语义和统计属性，提供有关汇编指令更丰富的信息	图像化过程消耗资源量较大
Li等人[56]的工作	2024	Windows	PE	将不同数据维度的信息整合映射为RGB图像	全局和局部特征，汇编指令序列	三通道彩色RGB图像	能够有效捕捉恶意软件中的多尺度特征	模型复杂度较高，计算成本较大
Xuan等人[57]的工作	2024	Windows	PE	将序列信息映射为RGB图像	时序特征与字节特征	三通道彩色RGB图像	使用双分支特征提取网络，融合时序和空间特征	计算成本高，不适用于资源受限的环境
Yang等人[60]的工作	2022	Windows	API	将API调用按种类划分后重组并生成图像	API投入度和API顺序索引	三通道彩色RGB图像	能够更全面地反映恶意软件的行为	重组API指令序列可能丢失调用序列的语义信息
Barona等人[59]的工作	2023	Windows	API	将API调用序列转换为图像	不同API调用的频率和顺序	包含调用频率和顺序的彩色图像	提供更详细的API调用模式	生成图像尺寸调整可能影响结果
Fang等人[64]的工作	2019	Android	.dex	构造字节矩阵计算熵值填充，映射为彩色图	.dex文件文本信息及字节码信息	含有熵信息的彩色图	处理家族分类有更高的分辨率和精度	需要结合分析文本段内容
Tang等人[61]的工作	2022	Android	.dex	将多粒度操作码序列特征映射为灰度图像	不同粒度的操作码特征	灰度操作码图像	能够更有效地抵抗混淆技术的影响	不适用打包的Android应用程序和动态代码加载的应用程序
Xiao等人[62]的工作	2022	Android	.dex	将Dalvik字节码文件转换为RGB图	Dalvik字节码特征	三通道彩色RGB图像	能有效应对数据混淆和原生库问题	不适用于静态分析无法提取特征的情况
Zhu等人[63]的工作	2023	Android	.dex	裁剪.dex文件保留index段，映射为彩色图	Dalvik字节码特征	三通道彩色RGB图像	突出不同部分之间的依赖性或相互作用	不适用于静态分析无法提取特征的情况
Lachtar等人[65]的工作	2023	Android	.dex	使用空间填充曲线将本地指令转换为图像	Android应用程序原生指令	融合空间曲线的熵图	能有效处理本地代码形式的恶意软件	难以检测本地代码中嵌入恶意代码的软件
Tasyurek等人[66]的工作	2023	Android	Manifest	将应用权限转换为类二维码的RGB图像	Android应用的权限数据	三通道彩色RGB图像	图像存储空间需求较低且处理过程高效	不适用于复杂的Android应用
Smmarwar等人[68]的工作	2022	IoT	二进制数据	结合使用离散小波变换转换为图	二进制文件近似系数和细节系数	三通道彩色RGB图像	可以突出视觉相似度并提高分类精度	需要对生成的图像进行二次着色
Ahmed等人[67]的工作	2023	IoT	二进制数据	将恶意软件二进制文件按字节转换成灰度图	二进制文件中的各种模式和结构	灰度图像	能够有效地处理和分析大量的恶意软件样本	在捕捉某些复杂的恶意软件特征方面存在局限性
Alsubai等人[69]的工作	2023	IoT	二进制数据	结合原始字节，光照算法生成图像	二进制文件中的各种模式和结构	三通道彩色RGB图像	突出原文件中复杂模式和结构	包含恶意信息种类单一
Ghahramani等人[70]的工作	2024	IoT	二进制数据	将恶意软件二进制文件按字节转换成灰度图	像素强度和频率特征	灰度图像	可以将时间序列数据转换为结构化图像	模型整体更新较为困难，泛化能力较差
Landman等人[71]的工作	2021	Linux	ELF	将Linux易失性内存转换为图像	内存转储信息特征	内存转储图像	可对整个内存转储深入分析	需要大量的处理资源

此外，表 4 中总结的各项研究还表明，当前图像化方法在应对实时更新的恶意软件时仍存在检测效率低、抗概念漂移能力差的问题。例如，Moreira 等人[52]提出的顺序矢量模式在特征提取效率上表现良好，但由于其仅适用于特定的恶意软件类型，适用范围较为有限。而 Deng 等人[53]的研究虽然提供了更为丰富的统计信息，但由于图像化过程资源消耗较大，因此会影响其在大规模检测任务中的实时性能。Ghahramani 等人[70]的研究虽然兼顾静态图像特征与运行时序特征，但模型的整体更新较为困难，泛化能力与实时检测性能较差。

综上所述，现有的图像化方法在静态特征提取上展现了其高效的通用性和泛化性，但在面对恶意软件的动态行为特征提取和实时性检测准确性问题时，依然存在较大的改进空间。

4.3　针对图像特征提取与增强的研究

通过对恶意软件的二进制代码段进行图像化操作，能够生成对应的灰度和彩色图像，从视觉的角度可以直观判断出某些图像是否具有同一家族的相似性[72]。图 4 展示了 Malimg 收集到的四个不同恶意软件家族的灰度图像样例，可以观察到相同家族的恶意软件在视觉和纹理特征上存在较高的相似性。因此，研究尝试通过提取图像中的纹理信息等特征，进行精准的家族相似性判断。在这个过程中，特征增强技术是一种常用的辅助手段。现阶段研究中，常用的特征增强方法包括调整图片尺寸、提高图像亮度、调转图片角度等[52-53,73]。为追求更深入的特征提取以及更高的检测精度，部分研究尝试使用更复杂精准的方法提取并增强图像特征。

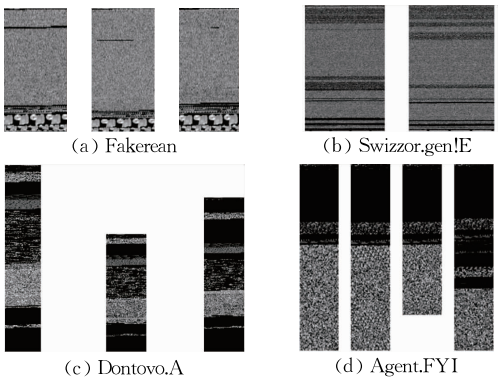

图 4　恶意软件家族灰度图像样例

Zhong 等人[74]将恶意软件样本转换为二维灰度图像，并使用对比有限自适应直方图均衡化算法对不同图像区域进行处理，以扩大所有图像区域的差异。通过调整每个图像区域中的像素强度，改善强度值的分布，从而增加对比度，使分类器更容易捕捉到同一家族中恶意软件样本的相似性。Son 等人[75]发现基于文本分析的灰度图像纹理主要沿垂直方向分布，宽度不是影响图像检测精度的重要因素。因此可以缩小图像的尺寸以减少计算消耗，并通过非长宽固定的矩形切割图片，利用最小的尺寸保留灰度图像垂直方向上的重要特征。然而文献[74-75]所提方法破坏了原始图像比例，可能导致部分特征信息的丢失，从而影响检测分类准确率。

为解决此问题，Xiao 等人[76]通过引入彩色标签框(Colored Label Boxes，CoLab)标记 PE 文件的章节信息，通过调整标签框的厚度突出显示 PE 文件中章节的分布信息。实验结果表明，该方法可以有效地发现恶意软件家族间的纹理相似性，从而实现高效的检测分类任务。此外，Kumar 等人[77]使用局部和全局特征混合的方法处理恶意软件图像样本，使用场景特征描述符提取图像全局特征，通过计算图像的空间分布和颜色分布描述恶意软件图像的整体特征、混合局部特征和全局特征以提高检测能力。Li 等人[78]提出一种轻量级恶意软件检测分类框架，通过像素填充和数据增强技术来应对样本不平衡问题，同时结合多尺度特征融合和通道注意力机制增强特征质量，提高检测分类性能。然而文献[76-78]所提方法在面对复杂恶意软件家族时的泛化能力较差，难以应对大规模、多变、实时更新的恶意软件变种。

通过对特征提取与增强相关研究的分析可以发现，尽可能保留恶意软件图像的原始信息和结构特征，从而充分利用神经网络进行检测，是一种常见的研究方法。然而，不同恶意软件家族之间的代码结构差异可能并不显著，代码框架的重写与复用在恶意软件家族中时常发生。因此，仅仅依赖图像的相似性进行恶意家族分类容易受到恶意混淆技术的干扰，影响最终的检测分类性能。

为解决这类问题，Venkatraman 等人[79]提出使用结合卷积神经网络(Convolutional Neural Networks，CNN)和 LSTM 的混合深度学习方法，通过对 Windows API 调用序列进行统计分析，保留并整合恶意软件的图像特征和时序特征。实验结果表明，该方法可以显著提高模型的鲁棒性和检测分类准确性。然而，模型对序列特征的依赖使其在处理高维特征时计算成本过大，整体结构过于复杂，难以快速适应恶意软件变种的多样化特征。此外，Vasan 等人[80]提出使用多层堆叠神经网络架构提取恶意

软件图像特征进行检测。通过微调预训练的神经网络模型,使其适应恶意软件检测分类任务。实验结果表明,所提出的检测框架模型能够从恶意软件图像中提取出更具代表性和区分度的特征,从而实现更准确的恶意软件家族分类。

通过对文献[79-80]分析可知,目前多数特征提取方案均从单一的灰度图像或彩色图像入手,忽略了其他维度的特征信息。然而,单一的特征表示可能不足以揭示恶意软件家族中隐藏的关键特征。因此,Xiao 等人[81]提出将恶意软件二进制文件转换为结构熵图,并通过堆叠多个小卷积核的方式,在特征提取阶段实现更大的感受野。通过增加网络深度和非线性变换的数量,用较少的网络参数学习更复杂的行为模式,从而提取更深层次的、准确的特征信息。实验结果表明,与现有方法相比,该方法具有更高的检测准确率。Vasan 等人[82]提出使用微调的卷积神经网络架构,结合模型预训练与反向传播技术进行参数微调以提高模型的检测性能,实验结果表明,该方法能有效检测混淆变体类恶意软件。文献[79-82]所提方法为未来的研究提出了新的思路,即使用模型预训练或更深层次的神经网络进行特征提取。尽管这种设计会导致内存占用过载和检测效率降低等问题,但在应对恶意软件威胁时,可以有效提升检测准确率这一核心目标。

表 5 对经典的图像化检测方法进行了全面的总结概括,主要包括实验所用数据集、规模和最后更新年份、可检测的恶意类型、所用方法的抗混淆能力和对传统检测方法的改进等。由表 5 可知,特征提取的质量和分类算法的可靠性对模型的检测分类性能具有显著影响,高质量的特征提取能够更准确地捕获恶意软件的关键行为模式,而可靠的分类算法则可以确保这些特征能够被正确地解析和分类。因此,如何对特征提取和特征增强方案进行深入优化、如何有效提高模型检测分类效率是未来研究中的重点方向。

表 5 基于图像化方法的恶意软件检测研究总结

现有研究工作	文献年份	实验所用数据集、规模和最后更新年份	可检测的恶意类型	抗混淆能力	检测所用图像特征信息	检测分类所用方法	检测任务	是否使用迁移学习	检测/分类准确率/%	对传统检测方法的改进
Nataraj 等人[50]的工作	2011	Malimg,9339 个样本,2023	PE	较弱	原始字节码转换的灰度图	Gabor 滤波器	家族分类	否	—/98.00	无需解析 PE 文件格式、无需动态执行恶意代码
Venkatraman 等人[79]的工作	2019	Malimg,9339 个样本,2023	PE	较强	原始字节码转换的灰度图	基于管道技术的改进 CNN	恶意检测,家族分类	是	99/96.30	可以应对数据密集型环境的规模和复杂性问题
Vasan 等人[80]的工作	2020	Malimg,9339 个样本,2023	PE	较强	原始字节码转换的灰度图	集成堆叠 CNN 的架构 IMCEC	家族分类	是	—/99.50	可以有效抵御恶意软件的混淆攻击技术
Yu 等人[51]的工作	2021	Malimg,Drebin,17863 个样本,2023/2014	PE	一般	包含时间序列的灰度图	集成学习算法 RF 和 VC	家族分类	否	—/99.00	在没有逆向工程的情况下,可以检测到使用规避技术的恶意软件变体
Kumar 等人[77]的工作	2021	Malimg,9339 个样本,2023	PE	较强	局部和全局特征混合提取生成的灰度图	融合早期停止技术的改进 CNN	家族分类	是	—/98.71	可以处理多态代码混淆,以及打包、加密等问题
Awan 等人[83]的工作	2021	Malimg,9339 个样本,2023	PE	一般	原始字节码转换的灰度图	基于空间注意力的卷积神经网络 SACNN	家族分类	是	—/97.62	无需特征工程技术和领域专家知识,检测成本低
Chaganti 等人[14]的工作	2022	MMCC,10 868 个样本,2015	PE	较强	固定宽度参数的灰度图	改进的 CNN 网络 EfficientNet	家族分类	是	—/99.00	可以有效抵御恶意软件的混淆攻击技术
Tang 等人[54]的工作	2022	Malimg,9339 个样本,2023	PE	很强	基于特征矩阵的多通道映射图	微调的 CNN	家族分类	是	—/99.68	通过图像化指令相关信息,获得更全面、更本质的恶意代码特征
Deng 等人[53]的工作	2023	MMCC,10 868 个样本,2015	PE	很强	三通道马尔科夫彩色图	集成网络架构 MCTVD	家族分类	是	—/99.44	该方法不需要裁剪或压缩,不会导致有关提取内容的有用信息丢失
Li 等人[56]的工作	2024	MMCC,DataCon,共 16 737 个样本,2015/2020	PE	很强	不同数据维度映射三通道 RGB 彩色图	大核卷积卷积网络 TriCh-LKRepNet	家族分类	否	—/99.47,97.55	增强了对混淆和变体恶意软件的多尺度特征检测能力
Xuan 等人[57]的工作	2024	MMCC,DataCon,共 16 737 个样本,2015/2020	PE	很强	基于序列信息映射的三通道 RGB 彩色图	双向时序卷积网络 BiTCN-TAEfficientNet	家族分类	是	—/99.46,97.92	通过融合时序和空间特征,能够更有效地分类恶意软件家族

（续　表）

现有研究工作	文献年份	实验所用数据集、规模和最后更新年份	可检测的恶意类型	抗混淆能力	检测所用图像特征信息	检测分类所用方法	检测任务	是否使用迁移学习	检测/分类准确率/%	对传统检测方法的改进
Yang等人[60]的工作	2022	阿里云天池数据集，110 000个样本，2022	API	一般	包含重组API的三通道RGB彩色图	改进的VGG19，MficNN	恶意检测	是	98.66/—	轻量级网络架构降低了对恶意图像进行分类的计算资源和时间成本
Barona等人[59]的工作	2023	沙箱运行收集，50 000个样本，2023	API	较强	包含API调用频率和顺序的彩色图像	FreqSeq图像生产算法，ResNet50	恶意检测	是	98.35/—	结合API调用频率及其在序列中的顺序，可以有效应对恶意混淆技术
Naeem等人[8]的工作	2020	Malimg，Leopard，26 558个样本，2023/2018	.dex	较强	.dex文件字节码生成RGB彩色图	深层次CNN网络，VGG16	家族分类	是	—/97.81，98.47	可以有效应对工业物联网恶意软件的混淆打包技术
Zhu等人[63]的工作	2023	Virusshare收集，4950个样本，2024	.dex	较强	裁剪index段生成的彩色RGB图	改进的CNN MADRF-CNN	恶意检测	是	96.90/—	不依赖于先验知识和手动功能，在处理代码混淆方面比传统检测方法更有效
Lachtar等人[65]的工作	2023	自收集数据集，15 085个样本，2023	.dex	很强	融合空间填充曲线的熵图	与本地指令内容结合的CNN	恶意检测	是	99.70/—	可以有效应对包含恶意内容的应用程序重打包的混淆技术
Vasan等人[82]的工作	2020	IoT_malware，数据集，共13 152个样本，2023	IoT	较强	应用2D矩阵颜色图生成的彩色图像	改进的CNN网络IMCFN	家族分类	是	—/97.35	模型的计算成本更低，精度更高，并且更具有泛化性
Smmarwar等人[68]的工作	2022	IoT_malware，Malimg，13 152个样本，2020/2023	IoT	很强	包含二进制文件近似系数的RGB彩色图	三阶段DMD-DWT-GAN检测方法	恶意检测，家族分类	否	99.99/99.99	能够检测新的恶意软件及其变种，实时性较高
Alsubai等人[69]的工作	2023	IoT_malware，Virusshare，48 000个样本，2023/2024	IoT	较强	包含二进制源文件信息的RGB彩色图	集成YoLoV7以及微调的CNN	家族分类	是	—/98.65，97.30	适用于物联网环境且具有适应性、分布式，所需计算资源较少
Ghahramani等人[70]的工作	2024	IoT_malware，15 085个样本，2023	IoT	较强	包含二进制源文件信息的灰度图	集成集群，概率，深度学习的方法	家族分类	否	—/97.40	通过集成方法提高恶意软件检测的实时性，适用于物联网环境

4.4　针对数据集相关问题的研究

模型训练过程中，数据集的质量直接影响着模型最终的检测效率和准确性。相对而言，使用高质量数据集训练的模型通常具有更强的鲁棒性，有助于提高模型训练的速度和检测效率。然而，如果数据集过时或不符合特定的研究需求，则可能导致模型出现过拟合、泛化性差等问题[84]。因此，针对传统数据集中不可避免的缺陷，需要通过某些特定的技术手段缓解或消除这些影响。同时，恶意软件家族数量与良性软件家族数量差距过大、数据集不平衡以及样本不足的现象时有发生，这些因素都可能对模型的综合性能产生不利影响。

4.4.1　针对少量样本问题的研究

目前研究所用数据集普遍存在部分类别样本数量不足的问题，因此，针对少量样本检测的研究逐渐成为热点。Hsiao等人[85]受利用孪生神经网络（Siamese Neural Network，SNN）进行单次目标图像识别成功的启发，将SNN应用于数据集中仅有少量样本的恶意软件家族检测分类任务，实验结果表明，该方法可以在一定程度上缓解样本数量不足带来的影响，有效提高模型的检测分类性能。Bai等人[86]通过使用基于下采样以及多层感知机（Multi-Layer Perceptron，MLP）的特征提取器，结合SNN检测模型对存在少量样本的严重不平衡数据集进行检测分类实验。实验结果表明，该方法可以通过采样技术缓解数据不平衡对检测结果造成的影响，有效提高模型检测分类准确率。

Zhu等人[87]在总结已有基于少量样本进行检测的研究基础上，提出使用改进的SNN进行少量样本检测，通过使用相似性度量和K临近算法（K-Nearest Neighbor，KNN）对图像进行熵值特征提取，并将其转换为图像以完成检测分类。实验结果表明，该方法能够更准确地捕捉每个恶意软件样本在特征嵌入空间中的独特性，在包含多类家族的样本集上具有较高的分类性能。

通过分析文献[85-87]可知，使用SNN代替传

统模型,可以有效提高对存在少量样本数据的检测分类准确率。然而,这些方法并没有从根本上解决传统深度学习分类器的主要问题,即出现新的未知恶意样本时需要重新训练分类器模型。

为解决此类问题,Conti 等人[88]通过改进图像化手段以及调整神经网络结构以解决此类问题,将恶意软件样本可视化为三通道 RGB 图像后,使用改进的卷积暹罗神经网络(Convolutional Siamese Neural Network,CSNN)进行检测分类实验。通过对比实验结果可知,当只有少量样本可用于训练时,该方法能有效提取特征信息,显著提高模型对未知恶意软件样本的检测分类效果。Jiang 等人[89]通过使用匹配网络和原型网络架构,解决传统检测方法在训练时需要收集大量样本数据的不足,实现对恶意软件新变种样本的快速分类。但该方法忽略了恶意软件在行为层面的动态特征,故在检测具有相似视觉特征但行为模式不同的多态恶意软件时,容易出现误分类的风险,从而影响检测的全面性和准确性。

针对现有模型无法根据样本动态调整模型参数以及没有深入考虑样本之间相关性的问题,Chai 等人[90]提出一种动态卷积方法,可以实现基于样本自适应的动态特征提取。利用文献[50]提到的图像化方法将恶意软件转换为灰度图像后,将类特征定义为支持集中每个类别所有恶意软件样本动态嵌入的平均值,然后使用基于度量的方法计算查询样本与原型之间的距离,实现高效恶意软件检测。

然而,现有的少量样本学习方法仍然存在显著的局限性,其中最突出的一个问题是其可迁移性差,即这些模型在应对不同任务域时,往往无法保持较高的性能。这种方法在训练时通常依赖于特定领域的大量标注数据,当迁移到新的任务或领域时,因样本特征分布的差异,模型难以有效适应新的环境,导致泛化能力不足,最终无法在新任务域中取得理想的效果。这一局限性不仅影响模型在实际应用中的灵活性和普适性,也限制了其在动态、多变的环境下的实用价值。

因此,如何在少量样本条件下进一步提升模型的泛化能力,特别是增强其在未知领域或新任务上的迁移性能,已成为该领域未来研究的关键问题之一。未来的研究可能会集中于开发更具鲁棒性和自适应性的算法,能够在数据稀缺的情况下依然具备强大的跨领域迁移能力。

4.4.2 针对不平衡数据集的缓解手段

除恶意软件新变种样本数量较少的问题外,数据集中的类不平衡现象也是影响恶意软件检测分类准确率的重要因素之一。以 Malimg 数据集为例,图 5 可以直观地展示出该数据集中不同家族间恶意软件样本数量的不平衡现象,样本数最多的 Allaple.A 族样本数可达到 2949 个,而占比最少的 Skintrim.N 族样本仅有 80 个。

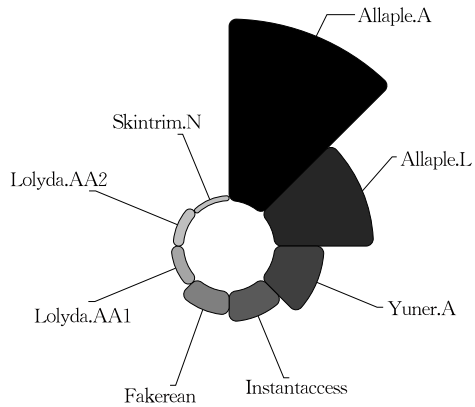

图 5 不同恶意家族恶意样本数量占比

为解决数据不平衡和公开数据集样本短缺的问题,Shaukat 等人[91]通过使用数据增强技术,避免由不平衡数据集引起的过拟合等问题,所用技术包括旋转角度、宽度高度偏移、水平翻转、垂直翻转、亮度调整和重新缩放。通过使用这些技术,不仅可以保留图像的原始尺寸比例,还能保留增强图像中的重要特征,因此,生成的数据与真实数据非常相似。Yang 等人[60]的研究也使用这种方法以应对数据集不平衡的问题。然而,文献[60,91]所提方法均通过处理原始数据集中的图像生成新的图像,没有产生新的可用特征,仍然存在过拟合风险。

为解决此问题,Wang 等人[92]采用生成对抗网络(Generative Adversarial Networks,GAN)生成基于原始数据特征的扩充图像样本,并通过添加噪声模拟恶意软件的变体。这种数据增强方法能够为数据集添加更多可用的特征,使得该方法在处理数据集不平衡时能够取得更好的性能。与该方法类似,Xuan 等人[93]提出一种基于 GAN 的序列特征生成算法。使用条件 Wasserstein 生成对抗网络,在不平衡恶意软件家族数据集上生成扩充样本并将其添加到训练集中,之后在双向时间卷积网络上进行训练。通过分析文献[92-93]的实验结果可知,利用 GAN 缓解数据集不平衡问题虽然表现效果良好,但是模型仍存在过度拟合的风险,即生成的样本可能

过于复杂或接近训练数据，不具有一般性。

为缓解数据集不平衡对实验结果造成的影响，另一种常用方法是通过调整采样方案获得平衡的数据分布，其中包括过采样、欠采样以及合成采样。Liu 等人[94]提出使用合成少数类过采样方法（Synthetic Minority Over-Sampling Technique，SMOTE）改善样本分布，通过生成合成样本平衡少数类和多数类，改善了分类器对少数类的学习能力，显著提升了检测的准确率和 $F1$ 分数。然而，该方法会放大原始数据中的噪声信息，如果少数类数据中含有错误的标签信息，生成的合成样本可能会降低模型的性能。此外，由于该方法没有考虑类边界情况，容易导致类间重叠问题的出现，从而增加错误分类的概率。

受 Liu 等人工作的启发，Onan[95]提出一种基于共识聚类的欠采样方法解决类不平衡的问题，通过共识聚类方法对多数类进行欠采样，得到一个平衡的训练集。使用平衡的训练集训练监督学习算法和集成学习方法以构建分类模型。实验结果表明，异质共识聚类的欠采样方案表现良好，可以有效地处理类别不平衡问题，提高分类模型的性能。然而同文献[92-94]存在的缺陷一样，文献[95]所提方法容易导致模型学习错误标签信息，出现性能下降的问题。

随着研究的深入，传统 SMOTE 方法的局限性逐渐凸显，因此 Chen 等人[96]提出使用单纯性不平衡数据分类方法（Simplex Imbalanced Data Gravitation Classification，S-IDGC），通过引入合成少数过采样技术方法及多种混合算法，在不影响检测分类性能的前提下，尽可能降低检测时间成本。此外，Ma 等人[97]在总结各类不平衡采样算法后，提出一种基于 SMOTE 和最临近欠采样（Edited Nearest Neighbors，ENN）的改进方法，可以缓解数据不平衡带来的影响。该方法使用 SMOTE 生成合成样本并对其进行清洗，从而增加少数类样本的数量，同时减少多数类样本与少数类样本之间的重叠。实验结果表明，通过结合 SMOTE 和 ENN 两种方法，可以有效解决样本不平衡问题，提高模型的检测分类准确性。

4.4.3　常用数据集总结

目前，绝大多数公开数据集内容已经长期没有进行更新，数据老化问题日益凸显，致使现有检测模型难以有效应对数量快速增长的恶意软件新变种。为解决这一问题，部分研究者提出构建实时更新的数据集，或组建非公开自收集数据集以供模型训练。然而，由于数据的非公开性以及样本选取的随机性，难以确定实验数据的真实度和可信度。表 6 对现阶段研究中常用的公开数据集进行了详细总结，包括数据集名称、数据样本量以及获取 URL 地址等。

表 6　常用恶意软件数据集

系统平台	数据集名称	样本数量/个	数据类型	所用研究文献	获取 URL 地址
Windows	Malimg	9339	灰度图像	文献[10,50-51,73,77,90,98]	https://paperswithcode.com/dataset/malimg
Windows	MMCC	10868	反汇编字节码文件	文献[14,53,99-101]	https://www.kaggle.com/c/malware-classification
Windows	Mal-API-2019	7107	API 调用序列	文献[102-105]	https://data.mendeley.com/datasets/w393cchcb7/2
Windows	Ember	1100000	特征集	文献[106-109]	https://github.com/elastic/ember
Android	Derbin	5560	特征集	文献[51,110-111]	https://drebin.mlsec.org/
Android	AndroZoo	17000000	APK 文件	文献[112-113]	https://androzoo.uni.lu
Android	CICAndMal	5491	特征集	文献[114-115]	https://www.unb.ca/cic/datasets/andmal2017.html
Android	RmvDroid	9133	APK文件及其元数据	文献[116-117]	https://zenodo.org/record/2593596
Android	KuafuDet	252900	APK 文件	文献[118-119]	https://nsec.sjtu.edu.cn/kuafuDet/download.html
多平台通用	VirusShare	—	—	文献[37,120-121]	https://virusshare.com

4.5　针对所用检测框架及网络模型的研究

在 Nataraj 等人提出使用图像化方法解决恶意软件的检测分类问题后，相关研究大多选择提取图像复杂纹理特征信息进行分析，并判断待检测样本的类型。随着深度学习技术的快速发展，部分研究者选择结合深度学习方法改进恶意软件检测模型。由于 CNN 可以自动提取恶意软件的图像特征，因此 Cui 等人[73]选用 CNN 完成对灰度图像的识别和分类任务，以改进恶意软件变体检测。实验结果表明，与其他恶意软件检测模型相比，该模型在提高准确率和检测速度方面效果显著。此后，Kumar 等人[98]提出一种基于迁移学习的图像化恶意软件检测分类方法，利用深度卷积神经网络作为特征提取器和图像分类器，结合早期停止技术解决了过拟合问题。该方法还利用迁移学习的优势，将在 ImageNet 数据集上预训练的 VGG16 模型迁移到当前实验中，从恶意软件图像中提取特征并进行分类训练，以提高检测分类性能。但该模型

需要大量的数据以实现有效的训练，这可能导致在数据集较小或样本不足的情况下出现模型性能下降的问题。

此外，Rustam 等人[99]利用迁移学习技术，使用 VGG16 和 ResNet50 等预训练模型从图像样本中提取特征并进行分类，从而绕过常见的恶意混淆技术，以达到提高检测分类准确率的目的。Cui 等人[122]则使用多目标受限玻尔兹曼机模型提取恶意软件的关键特征，优化检测流程和计算效率，提高检测分类准确率。通过分析文献[73，98-99，122]可知，结合使用预训练模型以及迁移学习知识可以高效地完成检测分类任务，但预训练模型需要定期更新，以反映新的恶意软件变种和攻击技巧。

Shen 等人[100]为解决单一特征在面对相似恶意代码族时检测效果差的问题，提出融合多头注意机制和双向长短期记忆（Bidirect-ional Long Short-Term Memory，BiLSTM）的特征以改进恶意软件检测方法。通过对特征映射的通道和空间给予不同的关注，融合局部纹理特征与全局纹理特征并进行检测。实验结果表明，该方法可以有效地区分相似恶意代码家族。Kumar[101]通过微调卷积神经网络构建检测模型，该模型无需特征工程、二进制代码分析或逆向工程等先验知识即可检测未知恶意软件样本。实验结果表明，该方法能够快速完成恶意软件新变种的初步检测。文献[100-101]对比分析了特征工程在恶意软件检测分类中的作用，表明融合多种特征可以提高模型的检测能力。但该方法所用的注意力机制和复杂神经网络模型可能缺乏直观的可解释性。因此，对于深度学习模型以及图像化方法可解释性的研究还需要进一步深入。

表 7 对当前常用的图像化检测所采用的网络模型进行了全面对比和总结。具体内容涵盖了现有研究中使用的图像化方法、生成的图像类型、所使用的网络模型以及对基础网络模型的改进。由表 7 的内容可以看出，网络模型中的一些细微结构参数的调整可能会对整体性能产生显著影响。因此，未来针对模型改进和微调神经网络的研究也必定会受到更广泛的关注。

表 7 图像化恶意软件检测所用网络模型对比总结

现有研究工作	文献年份	实验数据集及规模	图像化方法	图像类型	研究所用网络模型	检测任务	对基础网络模型的改进	检测/分类准确率/%
Cui 等人[73]的工作	2018	Malimg数据集，共9339 个样本	原始字节码转换为灰度图像	尺寸固定的灰度图像	微调的CNN	家族分类	交叉使用卷积层与池化层，减少图像参数量，同时保留主要特征	94.50
Vasan 等人[80]的工作	2020	Malimg 数据集，共9339个样本	原始字节码转换为灰度图像	灰度图像	集成堆叠CNN的架构IMCEC	恶意检测与家族分类	集成VGG16以及ResNet50 卷积层获取特征送入 SVM 分类	99.50
Kumar[101]的工作	2021	Malimg，MMCC数据集，共20207个样本	原始字节码转换为灰度图像	尺寸固定的灰度图像	微调的CNN模型 MCFT-CNN	家族分类	使用全连接层替换ResNet50 的最后一层传入 softmax 层	Malimg：99.18 MMCC：98.63
Cui 等人[122]的工作	2021	Malimg数据集，共9339个样本	原始字节码转换为灰度图像	尺寸固定的灰度图像	多目标RBM模型	家族分类	通过引入 RBM 的对比发散算法将多分类转换为二分类问题	95.83
Ullah 等人[123]的工作	2022	CICInvesAndMal2019，CICMalDroid2020，共22441个样本	将.dex 文件字节流信息映射为灰度图	尺寸固定的灰度图像	微调的CNN	恶意检测与家族分类	使用带有三个卷积层的 CNN	CICInvesAndMal2019：99.00 CICMalDroid2020：97.00
Vasan 等人[82]的工作	2022	IoT_android，Malimg数据集，共13152 个样本	应用 2D 矩阵颜色图生成的彩色图像	Simhash 值为像素的彩色图像	改进的CNN网络 IMCFN	家族分类	添加 dropout 层以及包含 25 类的全连接层，检测时间成本低	Malimg：98.82 IoT-android：97.35
Kumar 等人[98]的工作	2022	Malimg，MMCC 数据集，共20207 个样本	原始字节码转换并复制三次生成RGB图像	尺寸固定的RGB 图像	改进 VGG16 的网络模型 DTMIC	家族分类	添加一个全连接层替换原模型最后两层，具有良好的分类精度	Malimg：98.92 MMCC：93.19
Shen 等人[100]的工作	2022	MMCC 数据集，共10868个样本	原始字节码转换为灰度图像	灰度图像	融合双重注意力机的改进 CNN	家族分类	集成双注意力机制以及 BiLSTM 提取特征并进行融合	97.75
Rustam 等人[99]的工作	2023	Malimg数据集，共9339个样本	原始字节码转换为灰度图像	灰度图像	堆叠 VGG16 和 Resnet50 双模型架构	家族分类	将 VGG16 输出的结果作为输入训练 ResNet50	100

4.6 针对图像化恶意软件检测方法可解释性的研究

图像化方法不仅在提升检测效率方面具有显著优势，还凭借其过程的直观性在增强模型可解释性方面展现出巨大的潜力。通过梯度加权类激活映射(Gradient-Weighted Class Activation Mapping，Grad-CAM)方法，可以在图像上高亮显示对模型分类决策影响最大的区域，从而解释其决策过程，揭示模型的关注点和潜在的判断依据。Marais 等人[124]利用 Grad-CAM 生成热图，直观显示对算法预测影响最大的区域。这种方法有助于理解模型的决策过程，确定二进制文件中需要重点检查的关键区域，从而提高恶意软件检测的可解释性和有效性。与 Marais 等人提出的方法类似，Iadarola 等人[125]提出将深度学习模型与 Grad-CAM 可视化技术结合以检测恶意软件，生成热图并可视化输入图像中对模型决策过程有重大影响的区域，使模型的预测更加易懂和透明。然而文献[124-125]所提方法的有效性高度依赖于深度学习模型检测的准确性以及输入数据特征的质量。

为解决对模型和数据的高度依赖，深度探讨图像化方法的可解释性，Lin 等人[126]提出可解释性集成学习(Interpretable Machine Learning，IML)方法，通过结合基于选择性深度集成学习的检测器和深度泰勒分解技术，实现恶意图像的像素级解释。此外，该研究还通过生成显著性热力图辅助视觉解释，并提出了热力图保真度、鲁棒性和表现力等指标以评估解释的质量。实验结果表明，IML 可以有效平衡可解释性和检测精度之间的关系。但该方法依赖于集成学习的有效性，如果集成的单个模型表现不佳，可能会影响整体的检测准确率和解释能力。

与之前的研究方案不同，Wang 等人[127]提出通过采用层相关性传播方法突出数据中不同特征对模型决策过程的影响。这种方法能直观地展示模型决策过程以及做出决策的原因，从而提高深度学习模型的透明度和可解释性。此外，Ullah 等人[123]提出使用模型无关可解释性(Local Interpretable Model-agnostic Explanations，LIME)方法以及基于 SHAP(SHapley Additive exPlanations)方法解释深度学习模型的决策过程，通过突出输入数据中导致特定预测的特征，以及每个特征对预测的影响，了解决策过程，提高模型输出结果的信任度和可靠性。与 Ullah 等人的研究类似，Card 等人[128]通过集成 SHAP、LIME 和排列重要性等补充方法提供模型决策过程的全面视图，以确保对模型预测提供合理的解释。但文献[127-128]所提方法的可解释性仍然高度依赖于模型的预测性能，难以迁移至实际应用中。

5 面临的挑战与未来展望

近年来，尽管基于图像化的恶意软件检测分类方法取得了显著进展，但其发展过程仍面临诸多挑战，这些挑战在一定程度上限制着当前检测方法性能的进一步提升。恶意软件开发者利用复杂的规避技术，使现有检测模型难以有效应对。本文将在这一节中深入探讨当前研究中所面临的主要挑战，并展望未来可行的研究方向。

5.1 面临的挑战

随着对抗性攻击技术的发展，图像化方法也面临着严峻威胁。恶意软件开发者可以通过修改恶意软件的文件、加密打包等技术逃避深度学习模型的检测，目前主要面临的挑战有以下几个方面。

(1)特征表示不足问题

通过对文献[72-80]进行深入分析可知，恶意软件的源文件通常以二进制形式存在，而现有的图像化方法难以捕捉其动态特征和行为信息，主要依赖于静态特征的提取，因此难以全面有效地反映其深层恶意行为。此外，当前常用的特征增强技术也相对单一，在多数研究中仍然采用调整图片尺寸、亮度、调转图片角度等传统方法。因此，未来的研究应着重于开发更加有效的特征提取和表示方法，特别是关注如何提升恶意软件动态行为的捕捉能力，从而提高模型的检测准确性和泛化能力，构建更具表现力的特征空间。

(2)对抗性攻击问题

在面对混淆打包等技术时，基于 PE 头文件等静态特征的检测方法已表现出明显的局限性。同时，已有研究表明，利用 GAN 生成的对抗样本能够有效混淆现有检测手段，使得检测模型难以分辨恶意软件的真实意图。因此，未来的研究应致力于设计更加稳健的检测模型，在面对对抗性攻击时仍然保持较高的检测精度。此外，还可以结合异常检测技术，识别潜在的恶意规避行为，以减少恶意软件逃逸的可能性。

(3)数据不平衡问题

在实际应用场景中，恶意软件和良性软件的样本分布通常是不平衡的，这种现象会导致模型对多数类样本过度拟合，而对少数类别不敏感。文献[91-97]所提方法并没有从根本上解决数据集不平衡的问

题。因此未来的研究可以从改进数据集的角度入手，根据现有数据及技术手段生成较为平衡的数据集样本，以确保模型在各个类别上均具备良好的性能。

(4)可解释性问题

深度学习模型的可解释性这一问题尚未得到充分解决，与之密切相关的图像化方法同样面临可解释性的难题。由于深度学习的黑盒特性，研究人员难以理解模型做出特定决策的原因，因此无法优化模型的决策过程。未来的研究应着重于提高模型的可解释性，为深入理解数据和模型行为提供基础。

在实际应用中，实时恶意软件检测仍然是一个亟待解决的问题。实时检测要求系统在极短时间内处理大量数据，这意味着传统的图像处理方法在性能优化方面需要面临巨大压力[129]。此外，实时检测对系统资源的占用也提出了更高的要求。如何在保证检测速度的同时，仍然保持检测的高准确率和鲁棒性，是当前图像化检测方法必须面对的关键问题之一[130]。未来的研究应更多关注如何优化图像化模型的计算效率，减少图像转换和模型推理过程中的计算成本，提升其在大规模实时检测任务中的表现，以实现对恶意软件新变种的快速、精准识别。

5.2 未来研究展望

针对本文所指出的目前研究仍需面临的挑战，未来的研究应集中于改进特征表示、抵御对抗性攻击、提高可解释性、处理不平衡数据和集成多模态数据等方面，以进一步提高图像化方法的检测性能和实用性。未来可以重点研究的方向包括以下几点。

(1)多模态数据集成

未来的研究应考虑将多模态数据集成应用于恶意软件检测和分类实验中。除了传统的文件二进制表示外，还可以结合行为日志、网络流量等多种数据源。这种多模态数据融合将为模型提供更全面的上下文信息，有助于挖掘深层次的特征关联，进而提升检测与分类的性能。

(2)对抗性防御

随着对抗性攻击威胁的加剧，未来的研究应着力于设计更具鲁棒性的恶意软件检测模型，以抵御恶意软件开发者采用的对抗性攻击和高级规避技术。为保护检测模型免受攻击的干扰，可以深入研究并应用对抗性训练与防御技术，这将增强模型在面对复杂攻击场景时的稳健性和可靠性。

(3)可解释性研究

未来的研究可以深入探讨如何提高图像化检测模型的可解释性，以帮助网络安全相关人员理解模型的决策过程和输出依据，优化决策处理过程，提高检测分类准确率。

(4)处理大规模数据

随着恶意软件变种数量的快速增长，未来的研究必须能够有效处理大规模且多样化的数据。为扩大检测的覆盖范围，可以探索使用分布式计算技术处理大型数据集，从而实现更高的计算效率和更广泛的检测能力，满足不断增长的数据处理需求。

(5)实时恶意软件检测

当前的恶意软件检测研究大多依赖于公开数据集进行模型训练，然而这些数据集往往存在时效性不足、数据老化以及严重的数据不平衡问题，难以满足实时检测需求。因此，未来的研究应聚焦于开发高效的实时恶意软件检测技术，同时构建能够适应实时场景需求的数据集，以提升检测的准确性和时效性。

6 总 结

本文通过回顾130篇高质量的恶意软件检测与分类相关研究，重点分析了各种基于图像化方法的恶意软件检测技术。根据检测手段、特征提取方法、数据集处理技术和模型优化策略，对这些研究进行系统评述和比较分析。本研究的主要贡献如下。

本文首先总结了常用的恶意软件检测方法，主要包括静态分析、动态分析、基于机器学习和深度学习的检测方法，概要介绍了传统检测方法存在的问题以及面临的挑战。其次，对利用图像化方法进行恶意软件检测与分类的研究进行全面回顾，详细分析了图像化恶意软件检测分类方法的具体流程，内容涵盖以下几个方面：(1)图像化相关手段；(2)特征提取以及特征增强手段；(3)平衡数据集相关方法；(4)优化微调检测模型网络方法，并在后续章节中完成对实验所用数据集相关问题的讨论。最后，本文提出了与恶意软件检测相关的若干新兴研究方向，特别是针对概念漂移和实时恶意软件检测的挑战。

通过对恶意软件检测与分类方法，特别是基于图像化检测方法的全面回顾，本文对自图像化检测方法提出以来的多种具有代表性的优化方案进行了系统的总结与分析。然而，如何有效应对恶意软件带来的威胁依然是当前亟待解决的核心问题。随着恶意软件反检测能力的不断增强，未来如何应对更加隐蔽且难以检测的恶意软件，并在其引发威胁之

前隔离和消除这些威胁，成为亟待研究的新方向。此外，如何加速恶意软件的检测流程，从而迅速识别潜在威胁也是一个重要的挑战。未来的研究还需要探讨如何将先进的检测技术应用于实际环境，实现理论与实践的紧密结合。这些问题都亟待深入研究和探索，以寻求更加有效和可靠的解决方案。

参 考 文 献

[1] Calleja A, Tapiador J, Caballero J. The malsource dataset: Quantifying complexity and code reuse in malware development. IEEE Transactions on Information Forensics and Security, 2018, 14(12): 3175-3190

[2] Gibert D, Mateu C, Planes J. The rise of machine learning for detection and classification of malware: Research developments, trends and challenges. Journal of Network and Computer Applications, 2020, 153(5): 102526

[3] Han Yu, Fang Bin-Xing, Cui Xiang. StealthyFlow: A framework for malware dynamic traffic camouflaging in adversarial environment. Chinese Journal of Computers, 2021, 44(5): 948-962(in Chinese)
(韩宇，方滨兴，崔翔. StealthyFlow：一种对抗条件下恶意代码动态流量伪装框架. 计算机学报，2021，44(5)：948-962)

[4] Chai Y, Qiu J, Yin L, et al. From data and model levels: Improve the performance of few-shot malware classification. IEEE Transactions on Network and Service Management, 2022, 19(4): 4248-4261

[5] Nahmias D, Cohen A, Nissim N, et al. Deep feature transfer learning for trusted and automated malware signature generation in private cloud environments. Neural Networks, 2020, 124(4): 243-257

[6] Venkatraman S, Alazab M. Use of data visualisation for zero-day malware detection. Security and Communication Networks, 2018, 79(12): 1-13

[7] Gao C, Cai M, Yin S, et al. Obfuscation-resilient Android malware analysis based on complementary features. IEEE Transactions on Information Forensics and Security, 2023, 18(1): 5056-5068

[8] Naeem H, Ullah F, Naeem M R, et al. Malware detection in industrial Internet of Things based on hybrid image visualization and deep learning model. Ad Hoc Networks, 2020, 105(8): 102154

[9] Zhang Jian, Chen Bo-Han, Gong Liang-Yi, et al. Research on malware detection technology based on image analysis. Netinfo Security, 2019, 19(10): 24-31(in Chinese)
(张健，陈博翰，宫良一等. 基于图像分析的恶意软件检测技术研究. 信息网络安全，2019，19(10)：24-31)

[10] Moraga L I, Malcó J P R, Zabala-Blanco D, et al. Detection of obfuscated malware by engineering memory functions applying ELM//Proceedings of the 2023 IEEE Colombian Conference on Applications of Computational Intelligence. Bogota, Colombia, 2023: 1-6

[11] Chatzoglou E, Karopoulos G, Kambourakis G, et al. Bypassing antivirus detection: Old-school malware, new tricks//Proceedings of the 18th International Conference on Availability, Reliability and Security. Benevento, Italy, 2023: 1-10

[12] Samociuk D. Antivirus evasion methods in modern operating systems. Applied Sciences, 2023, 13(8): 5083

[13] Lan Q, Du Y, Gao C, et al. Status and outlook of image-based malware detection technology//Proceedings of the 2023 3rd International Symposium on Computer Technology and Information Science. Chengdu, China, 2023: 598-603

[14] Chaganti R, Ravi V, Pham T D. Image-based malware representation approach with EfficientNet convolutional neural networks for effective malware classification. Journal of Information Security and Applications, 2022, 69(9): 103306

[15] Or-Meir O, Nissim N, Elovici Y, et al. Dynamic malware analysis in the modern era—A state of the art survey. ACM Computing Surveys, 2019, 52(5): 1-48

[16] Chakkaravarthy S S, Sangeetha D, Vaidehi V. A survey on malware analysis and mitigation techniques. Computer Science Review, 2019, 32(1): 1-23

[17] Ji Tian-Tian, Fang Bin-Xing, Cui Xiang, et al. Research on deep learning-powered malware attack and defense techniques. Chinese Journal of Computers, 2021, 44(4): 669-695(in Chinese)
(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. 计算机学报，2021，44(4)：669-695)

[18] Guo M H, Xu T X, Liu J J, et al. Attention mechanisms in computer vision: A survey. Computational Visual Media, 2022, 8(3): 331-368

[19] Afianian A, Niksefat S, Sadeghiyan B, et al. Malware dynamic analysis evasion techniques: A survey. ACM Computing Surveys, 2019, 52(6): 1-28

[20] Nair R, Dodiya K R, Lakhalani P. A static approach for malware analysis: A guide to analysis tools and techniques. International Journal for Research in Applied Science and Engineering Technology, 2023, 11(12): 1451-1474

[21] Jin B, Choi J, Hong J B, et al. On the effectiveness of perturbations in generating evasive malware variants. IEEE Access, 2023, 11(1): 31062-31074

[22] Hashemi H, Samie M E, Hamzeh A. IFMD: Image fusion for malware detection. Journal of Computer Virology and Hacking Techniques, 2023, 19(2): 271-286

[23] Quan W, Deng P, Wang K, et al. CGFormer: ViT-based network for identifying computer-generated images with token labeling. IEEE Transactions on Information Forensics and Security, 2024, 19(1): 235-250

[24] Freitas S, Duggal R, Chau D H. MalNet: A large-scale image database of malicious software//Proceedings of the 31st ACM International Conference on Information & Knowledge Management. Atlanta, USA, 2022: 3948-3952

[25] Hossain M A, Islam M S. Enhanced detection of obfuscated malware in memory dumps: A machine learning approach for advanced cybersecurity. Cybersecurity, 2024, 7(1): 1-23

[26] Xu H, Zhou Y, Ming J, et al. Layered obfuscation: A taxonomy of software obfuscation techniques for layered security. Cybersecurity, 2020, 3(9): 1-18

[27] Tian R, Batten L, Islam R, et al. An automated classification system based on the strings of trojan and virus families// Proceedings of the 2009 4th International Conference on Malicious and Unwanted Software. Quebec, Canada, 2009: 23-30

[28] Feng Y, Anand S, Dillig I, et al. Apposcopy: Semantics-based detection of Android malware through static analysis// Proceedings of the 22nd ACM SIGSOFT International Symposium on Foundations of Software Engineering. Hong Kong, China, 2014: 576-587

[29] Tian K, Yao D, Ryder B G, et al. Detection of repackaged Android malware with code-heterogeneity features. IEEE Transactions on Dependable and Secure Computing, 2017, 17(1): 64-77

[30] Zou D, Wu Y, Yang S, et al. IntDroid: Android malware detection based on API intimacy analysis. ACM Transactions on Software Engineering and Methodology, 2021, 30(3): 1-32

[31] Maniriho P, Mahmood A N, Chowdhury M J M. A study on malicious software behaviour analysis and detection techniques: Taxonomy, current trends and challenges. Future Generation Computer Systems, 2022, 130(5): 1-18

[32] Deng L, Yu C, Wen H, et al. Few-shot malware classification via attention-based transductive learning network. Mobile Networks and Applications, 2024, 28(8): 1-15

[33] Kim H, Kim J, Kim Y, et al. Improvement of malware detection and classification using API call sequence alignment and visualization. Cluster Computing, 2019, 22(9): 921-929

[34] Babu S, Singh V. BD-MDLC: Behavior description-based enhanced malware detection for windows environment using longformer classifier. Computers & Security, 2024, 146(11): 104031

[35] Wang S, Chen Z, Yan Q, et al. A mobile malware detection method using behavior features in network traffic. Journal of Network and Computer Applications, 2019, 133(5): 15-25

[36] Singh J, Singh J. Detection of malicious software by analyzing the behavioral artifacts using machine learning algorithms. Information and Software Technology, 2020, 121(5): 106273

[37] Trizna D, Demetrio L, Biggio B, et al. Nebula: Self-attention for dynamic malware analysis. IEEE Transactions on Information Forensics and Security, 2024, 19(1): 6155-6167

[38] Madamidola O A, Ngobigha F, Ez-zizi A. Detecting new obfuscated malware variants: A lightweight and interpretable machine learning approach. arXiv preprint arXiv:2407.07918, 2024: 1-23

[39] Ali S, Abusabha O, Ali F, et al. Effective multitask deep learning for IoT malware detection and identification using behavioral traffic analysis. IEEE Transactions on Network and Service Management, 2023, 20(2): 1199-1209

[40] Huda S, Miah S, Hassan M M, et al. Defending unknown attacks on cyber-physical systems by semi-supervised approach and available unlabeled data. Information Sciences, 2017, 379(10): 211-228

[41] Martín A, Rodríguez-Fernández V, Camacho D. CANDYMAN: Classifying Android malware families by modelling dynamic traces with Markov chains. Engineering Applications of Artificial Intelligence, 2018, 74(1): 121-133

[42] Acarturk C, Sirlanci M, Balikcioglu P G, et al. Malicious code detection: Run trace output analysis by LSTM. IEEE Access, 2021, 9(1): 9625-9635

[43] Kumar S, Janet B, Neelakantan S. Identification of malware families using stacking of textural features and machine learning. Expert Systems with Applications, 2022, 208(12): 118073

[44] Chen X, Li C, Wang D, et al. Android HIV: A study of repackaging malware for evading machine-learning detection. IEEE Transactions on Information Forensics and Security, 2019, 15(1): 987-1001

[45] Kancherla K, Mukkamala S. Image visualization based malware detection//Proceedings of the 2013 IEEE Symposium on Computational Intelligence in Cyber Security. Singapore, 2013: 40-44

[46] Ban X, Chen L, Hu W, et al. Malware variant detection using similarity search over content fingerprint//Proceedings of the 26th Chinese Control and Decision Conference. Changsha, China, 2014: 5334-5339

[47] Han K S, Lim J H, Kang B, et al. Malware analysis using visualized images and entropy graphs. International Journal of Information Security, 2015, 14(2): 1-14

[48] Han K S, Kang B J, Im E G. Malware analysis using visualized image matrices. The Scientific World Journal, 2014, 1(7): 1-15

[49] Li S, Wang J, Song Y, et al. Tri-channel visualised malicious code classification based on improved ResNet. Applied Intelligence, 2024, 54(9): 12453-12475

[50] Nataraj L, Karthikeyan S, Jacob G, et al. Malware images: Visualization and automatic classification//Proceedings of the 8th International Symposium on Visualization for Cyber Security. PA, USA, 2011: 1-7

[51] Yu J, He Y, Yan Q, et al. SpecView: Malware spectrum visualization framework with singular spectrum transformation. IEEE Transactions on Information Forensics and Security, 2021, 16(1): 5093-5107

[52] Moreira C C, Moreira D C, de Sales Jr C S. Improving ransomware detection based on portable executable header using xception convolutional neural network. Computers & Security, 2023, 130(7): 103265

[53] Deng H, Guo C, Shen G, et al. MCTVD: A malware classification method based on three-channel visualization and deep learning. Computers & Security, 2023, 126(3): 103084

[54] Tang C, Zhou C, Hu M, et al. Malicious family identify combining multi-channel mapping feature image and fine-tuned CNN//Proceedings of the 2022 IEEE International Conference on Trust, Security and Privacy in Computing and

Communications. Wuhan, China, 2022: 9-19

[55] Yu Y W, Weber G M. HyperMinHash: MinHash in LogLog space. IEEE Transactions on Knowledge and Data Engineering, 2020, 34(1): 328-339

[56] Li S, Wang J, Song Y, et al. TriCh-LKRepNet: A large kernel convolutional malicious code classification network for structure reparameterisation and triple-channel mapping. Computers & Security, 2024, 144(9): 103937

[57] Xuan B, Li J, Song Y. BiTCN-TAEfficientNet malware classification approach based on sequence and RGB fusion. Computers & Security, 2024, 139(4): 103734

[58] Tang M, Qian Q. Dynamic API call sequence visualisation for malware classification. IET Information Security, 2019, 13(4): 367-377

[59] Barona J P, Alvarez J A, Farfán C J, et al. Malware detection using API calls visualisations and convolutional neural networks //Proceedings of the 2023 IEEE/ACM 23rd International Symposium on Cluster, Cloud and Internet Computing Workshops. Bangalore, India, 2023: 153-159

[60] Yang H, Zhang Y, Zhang L, et al. Malware detection based on visualization of recombined API instruction sequence. Connection Science, 2022, 34(1): 2630-2651

[61] Tang J, Li R, Jiang Y, et al. Android malware obfuscation variants detection method based on multi-granularity opcode features. Future Generation Computer Systems, 2022, 129(4): 141-151

[62] Xiao X, Yang S. An image-inspired and CNN-based Android malware detection approach//Proceedings of the 2019 34th IEEE/ACM International Conference on Automated Software Engineering. San Diego, USA, 2019: 1259-1261

[63] Zhu H, Wei H, Wang L, et al. An effective end-to-end Android malware detection method. Expert Systems with Applications, 2023, 218(15): 119593

[64] Fang Y, Gao Y, Jing F A N, et al. Android malware familial classification based on DEX file section features. IEEE Access, 2020, 8(1): 10614-10627

[65] Lachtar N, Ibdah D, Khan H, et al. RansomShield: A visualization approach to defending mobile systems against ransomware. ACM Transactions on Privacy and Security, 2023, 26(3): 1-30

[66] Tasyurek M, Arslan R S. RT-Droid: A novel approach for real-time Android application analysis with transfer learning-based CNN models. Journal of Real-Time Image Processing, 2023, 20(3): 1-17

[67] Ahmed I, Anisetti M, Ahmad A, et al. A multilayer deep learning approach for malware classification in 5G-enabled IoT. IEEE Transactions on Industrial Informatics, 2022, 19(2): 1495-1503

[68] Smmarwar S K, Gupta G P, Kumar S. Deep malware detection framework for IoT-based smart agriculture. Computers and Electrical Engineering, 2022, 104(1): 108410

[69] Alsubai S, Dutta A K, Alnajim A M, et al. Artificial intelligence-driven malware detection framework for Internet of Things environment. PeerJ Computer Science, 2023, 9(1): 1366-1379

[70] Ghahramani M, Taheri R, Shojafar M, et al. Deep image: A precious image based deep learning method for online malware detection in IoT environment. Internet of Things, 2024, 27(10): 101300

[71] Landman T, Nissim N. Deep-Hook: A trusted deep learning-based framework for unknown malware detection and classification in Linux cloud environments. Neural Networks, 2021, 144(12): 648-685

[72] Chu Q, Liu G, Zhu X. Visualization feature and CNN based homology classification of malicious code. Chinese Journal of Electronics, 2020, 29(1): 154-160

[73] Cui Z, Xue F, Cai X, et al. Detection of malicious code variants based on deep learning. IEEE Transactions on Industrial Informatics, 2018, 14(7): 3187-3196

[74] Zhong F, Chen Z, Xu M, et al. Malware-on-the-brain: Illuminating malware byte codes with images for malware classification. IEEE Transactions on Computers, 2022, 72(2): 438-451

[75] Son T T, Lee C, Le-Minh H, et al. An enhancement for image-based malware classification using machine learning with low dimension normalized input images. Journal of Information Security and Applications, 2022, 69(9): 103308

[76] Xiao M, Guo C, Shen G, et al. Image-based malware classification using section distribution information. Computers & Security, 2021, 110(11): 102420

[77] Kumar S, Janet B. Distinguishing malicious programs based on visualization and hybrid learning algorithms. Computer Networks, 2021, 201(24): 108595

[78] Li S, Wang J, Wang S, et al. PAFE: A lightweight visualization-based fast malware classification method. Heliyon, 2024, 10(16): 35965-35981

[79] Venkatraman S, Alazab M, Vinayakumar R. A hybrid deep learning image-based analysis for effective malware detection. Journal of Information Security and Applications, 2019, 47(8): 377-389

[80] Vasan D, Alazab M, Wassan S, et al. Image-Based malware classification using ensemble of CNN architectures(IMCEC). Computers & Security, 2020, 92(5): 101748

[81] Xiao G, Li J, Chen Y, et al. MalFCS: An effective malware classification framework with automated feature extraction based on deep convolutional neural networks. Journal of Parallel and Distributed Computing, 2020, 141(6): 49-58

[82] Vasan D, Alazab M, Wassan S, et al. IMCFN: Image-based malware classification using fine-tuned convolutional neural network architecture. Computer Networks, 2020, 171(22): 107138

[83] Awan M J, Masood O A, Mohammed M A, et al. Image-based malware classification using VGG19 network and spatial convolutional attention. Electronics, 2021, 10(19): 2444

[84] Miranda T C, Gimenez P F, Lalande J F, et al. Debiasing Android malware datasets: How can i trust your results if your dataset is biased? IEEE Transactions on Information Forensics and Security, 2022, 17(6): 2182-2197

[85] Hsiao S C, Kao D Y, Liu Z Y, et al. Malware image classification using one-shot learning with siamese networks. Procedia Computer Science, 2019, 159(1): 1863-1871

[86] Bai Y, Xing Z, Li X, et al. Unsuccessful story about few shot malware family classification and siamese network to the rescue//Proceedings of the ACM/IEEE 42nd International Conference on Software Engineering. Seoul, Republic of Korea, 2020: 1560-1571

[87] Zhu J, Jang-Jaccard J, Singh A, et al. A few-shot meta-learning based siamese neural network using entropy features for ransomware classification. Computers & Security, 2022, 117(6): 102691

[88] Conti M, Khandhar S, Vinod P. A few-shot malware classification approach for unknown family recognition using malware feature visualization. Computers & Security, 2022, 122(11): 102887

[89] Jiang L, Zhang Y, Shi Y. Visual fileless malware classification via few-shot learning//Proceedings of the International Conference on Cyber Security, Artificial Intelligence, and Digital Economy. Nanjing, China, 2023, 12718: 113-124

[90] Chai Y, Du L, Qiu J, et al. Dynamic prototype network based on sample adaptation for few-shot malware detection. IEEE Transactions on Knowledge and Data Engineering, 2022, 35(5): 4754-4766

[91] Shaukat K, Luo S, Varadharajan V. A novel deep learning-based approach for malware detection. Engineering Applications of Artificial Intelligence, 2023, 122(6): 106030

[92] Wang Z, Wang W, Yang Y, et al. CNN- and GAN- based classification of malicious code families: A code visualization approach. International Journal of Intelligent Systems, 2022, 37(12): 12472-12489

[93] Xuan B, Li J, Song Y. SFCWGAN-BiTCN with sequential features for malware detection. Applied Sciences, 2023, 13(4): 2079

[94] Liu Z, Zhang Y, Chen Y, et al. Detection of algorithmically generated domain names using the recurrent convolutional neural network with spatial pyramid pooling. Entropy, 2020, 22(9): 1058

[95] Onan A. Consensus clustering-based undersampling approach to imbalanced learning. Scientific Programming, 2019, 1(1): 5901087

[96] Chen Z, Yan Q, Han H, et al. Machine learning based mobile malware detection using highly imbalanced network traffic. Information Sciences, 2018, 433(4): 346-364

[97] Ma Y W, Chen J L, Kuo W H, et al. AI@nti-Malware: An intelligent framework for defending against malware attacks. Journal of Information Security and Applications, 2022, 65(3): 103092

[98] Kumar S, Janet B. DTMIC: Deep transfer learning for malware image classification. Journal of Information Security and Applications, 2022, 64(2): 103063

[99] Rustam F, Ashraf I, Jurcut A D, et al. Malware detection using image representation of malware data and transfer learning. Journal of Parallel and Distributed Computing, 2023, 172(2): 32-50

[100] Shen G, Chen Z, Wang H, et al. Feature fusion-based malicious code detection with dual attention mechanism and BiLSTM. Computers & Security, 2022, 119(8): 102761

[101] Kumar S. MCFT-CNN: Malware classification with fine-tune convolution neural networks using traditional and transfer learning in Internet of Things. Future Generation Computer Systems, 2021, 125(12): 334-351

[102] Panda B, Bisoyi S S, Panigrahy S. An ensemble approach for imbalanced multiclass malware classification using 1D-CNN. PeerJ Computer Science, 2023, 9(1): 1677-1699

[103] Almaleh A, Almushabb R, Ogran R. Malware API calls detection using hybrid logistic regression and RNN model. Applied Sciences, 2023, 13(9): 5439

[104] Zhang S, Wu J, Zhang M, et al. Dynamic malware analysis based on API sequence semantic fusion. Applied Sciences, 2023, 13(11): 6526

[105] Odat E, Yaseen Q M. A novel machine learning approach for Android malware detection based on the co-existence of features. IEEE Access, 2023, 11(1): 15471-15484

[106] Eren M E, Bhattarai M, Joyce R J, et al. Semi-supervised classification of malware families under extreme class imbalance via hierarchical non-negative matrix factorization with automatic model selection. ACM Transactions on Privacy and Security, 2023, 26(4): 1-27

[107] Anđelić N, Baressi Šegota S, Car Z. Improvement of malicious software detection accuracy through genetic programming symbolic classifier with application of dataset oversampling techniques. Computers, 2023, 12(12): 242

[108] Luo J, Zhang Z, Luo J, et al. Sequence-based malware detection using a single-bidirectional graph embedding and multi-task learning framework. Journal of Computer Security, 2024, 32(2): 141-163

[109] Copiaco A, El Neel L, Nazzal T, et al. A neural network approach to a grayscale image-based multi-file type malware detection system. Applied Sciences, 2023, 13(23): 12888

[110] Reddy R, Kumara Swamy M, Ajay Kumar D. Feature and sample size selection for malware classification process//Proceedings of the 3rd International Conference on Communications and Cyber Physical Engineering (ICCCE 2020). Singapore, 2020: 217-223

[111] Kim C, Chang S Y, Kim J, et al. Automated, reliable zero-day malware detection based on autoencoding architecture. IEEE Transactions on Network and Service Management, 2023, 20(3): 3900-3914

[112] Darwaish A, Naït-Abdesselam F, Titouna C, et al. Robustness of image-based Android malware detection under adversarial attacks//Proceedings of the ICC 2021-IEEE International Conference on Communications. Quebec, Canada, 2021: 1-6

[113] Jebin Bose S, Kalaiselvi R. An optimal deep learning-based framework for the detection and classification of Android malware. Journal of Intelligent & Fuzzy Systems, 2023, 44(6): 9297-9310

[114] Bayazit E C, Sahingoz O K, Dogan B. A deep learning based Android malware detection system with static analysis //Proceedings of the 2022 International Congress on Human-Computer Interaction, Optimization and Robotic Applications. Ankara, Turkey, 2022: 1-6

[115] Aldehim G, Arasi M A, Khalid M, et al. Gauss-mapping black widow optimization with deep extreme learning machine for Android malware classification model. IEEE Access, 2023, 11(1): 87062-87070

[116] Khan F B, Durad M H, Khan A, et al. Detection of data scarce malware using one-shot learning with relation network. IEEE Access, 2023, 11(1): 74438-74457

[117] Buriro A, Buriro A B, Ahmad T, et al. MalwD&C: A quick and accurate machine learning-based approach for malware detection and categorization. Applied Sciences, 2023, 13(4): 2508

[118] Alomari E S, Nuiaa R R, Alyasseri Z A A, et al. Malware detection using deep learning and correlation-based feature selection. Symmetry, 2023, 15(1): 123

[119] Ravi V, Pham T D, Alazab M. Attention-based multidimensional deep learning approach for cross-architecture IoMT malware detection and classification in healthcare cyber-physical systems. IEEE Transactions on Computational Social Systems, 2022, 10(4): 1597-1606

[120] Li S, Li Y, Wu X, et al. Imbalanced malware family classification using multimodal fusion and weight self-learning. IEEE Transactions on Intelligent Transportation Systems, 2022, 24(7): 7642-7652

[121] Chen Z, Ren X. An efficient boosting-based windows malware family classification system using multi-features fusion. Applied Sciences, 2023, 13(6): 4060

[122] Cui Z, Zhao Y, Cao Y, et al. Malicious code detection under 5G HetNets based on a multi-objective RBM model. IEEE Network, 2021, 35(2): 82-87

[123] Ullah F, Alsirhani A, Alshahrani M M, et al. Explainable malware detection system using transformers-based transfer learning and multi-model visual representation. Sensors, 2022, 22(18): 6766

[124] Marais B, Quertier T, Chesneau C. Malware analysis with artificial intelligence and a particular attention on results interpretability//Proceedings of the 18th International Conference on Distributed Computing and Artificial Intelligence. 2022, 1: 43-55

[125] Iadarola G, Martinelli F, Mercaldo F, et al. Towards an interpretable deep learning model for mobile malware detection and family identification. Computers & Security, 2021, 105(6): 102198

[126] Lin Y, Chang X. Towards interpretable ensemble learning for image-based malware detection. arXiv preprint arXiv: 2101.04889, 2021: 1-14

[127] Wang H, Zhu Z, Tong Z, et al. An effective approach for malware detection and explanation via deep learning analysis //Proceedings of the 2021 International Joint Conference on Neural Networks. Shenzhen, China, 2021: 1-10

[128] Card Q, Simpson D, Aryal K, et al. Explainable deep learning models for dynamic and online malware classification. arXiv preprint arXiv:2404.12473, 2024: 1-15

[129] Weitkamp E, Satani Y, Omundsen A, et al. MalIoT: Scalable and real-time malware traffic detection for IoT networks. arXiv preprint arXiv:2304.00623, 2023: 1-15

[130] Peters T, Farhat H. High-resolution image-based malware classification using multiple instance learning. arXiv preprint arXiv:2311.12760, 2023: 1-16

XIE Li-Xia, M. S., professor. Her research interests include network and system security, information security.

WEI Chen-Yang, M. S. candidate. His research interests include network information security and malicious software detection and classification.

YANG Hong-Yu, Ph. D., professor, Ph. D. supervisor. His research interests include network and system security, software security detection, and network security situation awareness.

HU Ze, Ph. D., lecturer, M. S. supervisor. His research interests include artificial intelligence, natural language processing, and network information security.

CHENG Xiang, Ph. D., researcher, M. S. supervisor. His research interests include network and system security, network security situation awareness, and APT attack detection.

ZHANG Liang, Ph. D., postdoctoral researcher. His research interests include reinforcement learning, deep learning-based signal processing, and network and system security.

Background

Malware has become one of the most significant threats to modern computing systems, compromising data, disrupting services, and causing substantial financial losses. Traditional detection methods, such as static and dynamic analysis, have proven inadequate in the face of increasingly sophisticated evasion techniques used by malware developers, including code obfuscation, polymorphism, and virtualization, which allow malware to hide its true nature, making detection and analysis far more complex.

In recent years, the frequency and complexity of malware attacks have surged globally, fueled by the proliferation of mobile devices, IoT, and cloud-based services. As malware variants evolve at an unprecedented rate, it has become increasingly difficult for conventional detection methods, thus highlighting the urgent need for more advanced techniques. While machine learning and deep learning-based approaches have shown promise in improving detection accuracy, they are often limited by their reliance on known malware patterns, making them less effective when against novel threats.

A promising new approach involves transforming malware binaries into images, allowing for novel feature extraction and classification methods that leverage the strengths of image processing and machine learning. Imaging-based malware detection has the potential to detect previously unseen malware by identifying patterns in the binary data that are difficult to obfuscate. However, comprehensive research on utilizing imaging-based methods for malware detection and classification tasks remains unexplored.

This review aims to provide a comprehensive analysis of imaging-based malware detection techniques, evaluating their effectiveness across different platforms and environments. By synthesizing existing research, we identify current challenges, including handling advanced obfuscation methods and optimizing these techniques for real-time applications. This review also explores future directions for the field, including concept drift, real-time malware detection, and large-scale data adaptation.

This work was supported by the Civil Aviation Joint Research Fund Project of the National Natural Science Foundation of China (No. U2433205), the National Natural Science Foundation of China (No. 62201576, No. U1833107), Youth Fund Project of Jiangsu Provincial Basic Research Program Natural Science Foundation (BK20230558), and the Open Fund of the Key Laboratory of Civil Aviation Flight Networking at Civil Aviation University of China (MHFLW202304).