

恶意软件检测和分类可视化技术综述

王金伟^{1,2,3}, 陈正嘉^{1,2}, 谢雪^{4,5}, 罗向阳⁶, 马宾⁷

(1. 南京信息工程大学数字取证教育部工程研究中心, 江苏 南京 210044; 2. 南京信息工程大学计算机学院, 江苏 南京 210044;
3. 数学工程与先进计算国家重点实验室, 河南 郑州 450001; 4. 中国科学技术大学网络空间安全学院, 安徽 合肥 230031;
5. 中国航天系统科学与工程研究院, 北京 100048; 6. 信息工程大学, 河南 郑州 450001;
7. 齐鲁工业大学网络空间安全学院, 山东 济南 250353)

摘要: 随着科技的不断发展, 恶意软件及其变种的种类不断增多, 已经成为网络安全的一个巨大挑战。这些恶意软件采用了多样的技术手段, 以欺骗或规避传统检测方法, 从而使得传统非可视化检测技术不再适用。近年来, 数据可视化作为一种强有力的恶意软件检测和分类手段引起了学术界的广泛关注。通过以图像的方式呈现恶意软件的核心特征, 这类方法能够显著提高恶意软件的检测和分类准确率, 从而在网络安全领域具有广阔的研究前景。综述恶意软件检测领域的传统非可视化检测技术与可视化检测技术。介绍了恶意软件检测的传统非可视化方式, 包括静态检测、动态检测、混合检测; 重点对一些常见的恶意软件可视化方法进行了调研和综合评述, 主要包括可视化结合机器学习与可视化结合深度学习两大方向, 这两种方法在恶意软件检测和分类中各有优势和特点, 因此在选择检测和分类方法时, 需要综合考虑多个因素, 包括数据集大小、计算资源和时间限制、模型准确度以及实现复杂度等; 对目前检测技术所面临的问题进行了总结, 并对未来的发展方向进一步展望。

关键词: 机器学习; 深度学习; 数据可视化; 恶意软件检测和分类

中图分类号: TP309

文献标志码: A

DOI: 10.11959/j.issn.2096-109x.2023064

收稿日期: 2022-03-13; 修回日期: 2023-08-10

通信作者: 谢雪, xuexie2008@163.com

基金项目: 国家自然科学基金(62072250, 62172435, U1804263, U20B2065, 61872203, 71802110, 61802212); 中原科技创新领军人才项目(214200510019); 江苏省自然科学基金(BK20200750); 河南省网络空间态势感知重点实验室开放基金(HNTS2022002); 江苏省研究生研究与实践创新项目(KYCX200974); 广东省信息安全技术重点实验室开放项目(2020B1212060078); 山东省计算机网络重点实验室开放课题基金(SDKLCN-2022-05)

Foundation Items: The National Natural Science Foundation of China (62072250, 62172435, U1804263, U20B2065, 61872203, 71802110, 61802212), The Leading Talents Program of Scientific and Technological Innovation in Henan Province (214200510019), The Jiangsu Natural Science Foundation (BK20200750), The Open Fund of the Key Laboratory of Network Space Situation Awareness (HNTS2022002), The Graduate Student Research and Practice Innovation Program of Jiangsu Province (KYCX200974), The Open Project of Guangdong Provincial Key Laboratory of Information Security Technology (2020B1212060078), The Open Research Fund of Shandong Provincial Key Laboratory of Computer Networks (SDKLCN-2022-05)

引用格式: 王金伟, 陈正嘉, 谢雪, 等. 恶意软件检测和分类可视化技术综述[J]. 网络与信息安全学报, 2023, 9(5): 1-20.

Citation Format: WANG J W, CHEN Z J, XIE X, et al. Review of malware detection and classification visualization techniques[J]. Chinese Journal of Network and Information Security, 2023, 9(5): 1-20.

Review of malware detection and classification visualization techniques

WANG Jinwei^{1,2,3}, CHEN Zhengjia^{1,2}, XIE Xue^{4,5}, LUO Xiangyang⁶, MA Bin⁷

1. Engineering Research Center of Digital Forensics, Ministry of Education, Nanjing University of Information Science and Technology, Nanjing 210044, China

2. Department of Computer, Nanjing University of Information Science and Technology, Nanjing 210044, China

3. State Key Laboratory of Mathematical Engineering and Advanced Computing, Zhengzhou 450001, China

4. University of Science and Technology of China, Hefei 230031, China

5. China Aerospace Academy of Systems Science and Engineering, Beijing 100048, China

6. Information Engineering University, Zhengzhou 450001, China

7. School of Cyberspace Security, Qilu University of Technology, Jinan 250353, China)

Abstract: With the rapid advancement of technology, network security faces a significant challenge due to the proliferation of malicious software and its variants. These malicious software use various technical tactics to deceive or bypass traditional detection methods, rendering conventional non-visual detection techniques inadequate. In recent years, data visualization has gained considerable attention in the academic community as a powerful approach for detecting and classifying malicious software. By visually representing the key features of malicious software, these methods greatly enhance the accuracy of malware detection and classification, opening up extensive research opportunities in the field of cyber security. An overview of traditional non-visual detection techniques and visualization-based methods were provided in the realm of malicious software detection. Traditional non-visual approaches for malicious software detection, including static analysis, dynamic analysis, and hybrid techniques, were introduced. Subsequently, a comprehensive survey and evaluation of prominent contemporary visualization-based methods for detecting malicious software were undertaken. This primarily encompasses the integration of visualization with machine learning and visualization combined with deep learning, each of which exhibits distinct advantages and characteristics within the domain of malware detection and classification. Consequently, the holistic consideration of several factors, such as dataset size, computational resources, time constraints, model accuracy, and implementation complexity, is necessary for the selection of detection and classification methods. In conclusion, the challenges currently faced by detection technologies are summarized, and a forward-looking perspective on future research directions in the field is provided.

Keywords: machine learning, deep learning, data visualization, malware detection and classification

0 引言

恶意软件是指以某种方式对用户或计算机造成破坏的任何软件，又被称为恶意代码。作为一种有害的可执行程序，它们严重影响着网络系统的安全，并引起了广泛关注。根据功能，恶意软件可以分为木马、病毒、蠕虫、间谍软件以及广告软件等多种类型；根据平台，恶意软件可以分为 Windows、Linux 或安卓（Android）系统软件，本文主要介绍 Windows 恶意软件。除了功能和平台之外，恶意软件及其变体还可以根据家族进行分类。同一家族的恶意软件及其变体必须具有类似的二进制模式，这些模式在检测恶意软件变体

和分类家族方面具有非常大的帮助。

随着自动化生成工具和恶意软件混淆技术的广泛使用，恶意软件及其变体的数量与种类不断增加。恶意软件混淆技术主要包括包装、变形和虚拟技术，被广泛用于逃避杀毒软件的检测。由此大量新的恶意代码生成，它们的入侵方式以及传播方式不断变化，对网络环境产生了巨大威胁，受影响的用户数量和检测到的恶意软件的数量都呈指数级增长，恶意软件分析师面临巨大挑战。根据 2020 年国家互联网应急中心《网络安全信息与动态周报》第 16 期，一周内我国境内感染网络病毒的主机数量约为 54.8 万台^[1]。

因此，如何从恶意代码中提取更加有效且稳

定的特征, 从而提高恶意代码检测和分类准确率与效率引起了人们深入的思考和研究。为应对不断增长的恶意代码数据集, 越来越多的自动检测和分类方法被应用, 包括随机森林、决策树、支持向量机等机器学习方法以及卷积神经网络(CNN, convolutional neural network)、循环神经网络(RNN, recurrent neural network)等深度学习方法。这些方法的应用大大提高了恶意代码的检测效率和准确率, 成为恶意软件分析师的重要工具。

传统非可视化恶意软件检测和分类方法主要可以分为静态代码分析和动态代码分析两种技术。静态代码分析技术不需要执行代码, 能够快速获取恶意软件的语法和语义信息。通常使用专业交互式反汇编器(IDA Pro, interactive disassembler professional)等反汇编工具进行逆向分析, 提取的静态特征包括字节序列、控制流图和字符串签名等。相比之下, 动态代码分析技术通过在虚拟环境中执行代码来获取恶意软件的行为报告, 包括函数调用监测、功能参数分析、信息流跟踪、指令跟踪和动态可视化分析等, 需要使用自动化工具如CW沙箱(cyber warfare sandbox)、Anubis、TTAnalyzer、Ether和ThreatExpert等来实现。两种技术都有其优缺点。静态代码分析技术覆盖范围较大, 但在面对混淆和加壳代码时可能会漏检。相比之下, 动态代码分析技术更加有效且准确, 但需要付出时间和空间成本。

近年来, 可视化方法(将恶意代码转化为图像形式展示), 作为一种新兴的恶意软件检测和分类技术备受关注。通过可视化方法, 可以发现恶意软件图像中包含着丰富的信息。同一类别的恶意家族的可视化图像通常具有相似性, 而不同家族的可视化图像之间则有较大的差异。此外, 大多数恶意代码变体是使用自动化技术或重用一些重要的模块来生成的, 因此它们在二进制代码中具有一些相似之处。相对于传统的恶意软件分类方法, 可视化方法可以加速恶意软件分类的过程, 满足大数据计算、专家系统反馈和认知复杂性等方面的需求, 从而能够更加有效地检测和分类恶意软件。

本文对恶意软件检测和分类可视化技术进行全方位的综述, 简述了传统非可视化恶意软件检测技术, 包括静态检测、动态检测和混合检测3个

方面; 介绍了可视化技术在恶意软件检测与分类领域的应用, 主要从机器学习、深度学习和其他可视化方法3个角度进行阐述, 这一新兴领域的研究为恶意软件的检测和分类提供了新的思路和方法, 有望成为未来恶意软件检测技术中不可或缺的一部分; 对当前检测技术问题进行了总结与展望。

1 传统非可视化恶意软件检测技术

本节将对传统非可视化恶意软件检测技术进行综述。针对恶意软件程序实际运行, 本文将传统恶意软件检测技术分为静态检测技术、动态检测技术和混合检测技术3类。

1.1 静态检测技术

静态检测技术是在对二进制代码不运行的情况下对其分析, 从而检测出代码特性进行分类的方法, 如何提取准确有效的静态特征是这种检测技术的重点。该技术不需要对代码运行, 因此静态检测时间和空间成本较小, 效率较高, 可以对代码特性进行全面的分析。

常用的静态特征包括API调用序列、N-Gram提取字节序列特征、文件结构信息、操作码频率分布、调用函数、字符串签名和控制流图等。此外, 静态检测技术可以使用许多工具, 如IDA Pro等反汇编工具可用于逆向分析恶意可执行文件, 提供更有效的信息; 而LordPE内存转储工具可以在系统内存中获取受保护的代码, 对于分析有更大的帮助。

(1) 基于API调用序列进行静态分析

在恶意软件检测和分类中, API调用序列常作为一个重要的静态特征被使用, 它能够反映出软件特定的行为顺序。通过分析程序所有的系统API调用, 能够初步了解程序的作用, 或者至少能够判断程序是否为恶意软件。

Iwamoto等^[2]在其研究中提取了API序列, 并使用基于特征提取的层次聚类分析方法对其处理。作者将API序列转化为函数调用图, 并对其进行缩放。该方法考虑了API之间的调用关系, 使得它们更具有区别性。

Imran等^[3]提出了一种基于相似度的分类方法。作者提取API序列来训练隐马尔可夫模型(HMM, hidden Markov model), 进而评估训练序列和测试序列之间的相似性, 并通过计算相关性得分对恶意软

件进行分类。该方法可以有效地利用 API 序列信息，但训练 HMM 模型需要大量的计算开销。

Hardy 等^[4]则从可执行文件中提取 API 调用序列，并将其加入堆叠式自动编码器（SAE, stacked auto encoder）深度学习架构模型中，该模型采用了无监督预训练和监督反向传播算法进行恶意软件检测和分类。

（2）基于 N-Gram 提取字节序列特征进行静态分析

除了 API 调用序列，使用 N-Gram 方法提取恶意代码的字节序列特征也是一种重要的静态分析，该方法的具体步骤为将原始字节序列使用一个长度为 n 的滑动窗口提取子序列的特征。需要注意的是，如果 n 值过小，可能会影响恶意软件的全局性；如果 n 值过大，可能会影响特征之间的相关性。因此，在选择 n 值时需要进行权衡和调整，以获得最佳的检测效果。

Schultz 等^[5]提出了一种在 Windows 平台下使用 N-Gram 算法提取静态特征的方法，并选取了 3 种不同类型的特征进行分析。第一种是从 PE 文件的动态链接库中提取的特征，第二种是提取可执行文件中的可打印文本字符串，第三种是从可执行文件中提取的 N-Gram 字节序列。这些特征可以通过机器学习算法进行训练和分类。该方法是一种有效的恶意软件检测方法。

2004 年，Kolter 等^[6]使用 N-Gram 算法提取收集到的恶意软件特征，并评估了朴素贝叶斯、决策树、支持向量机等归纳方法。结果显示，增强决策树的表现优于其他方法。在此基础上，Kolter 等^[7]在 2006 年进行了进一步实验，通过选取不同

N-Gram 中的 n 值来确定最佳 n 值以实现最高的检测率，从而对收集到的恶意软件进行检测和分类。

Kang 等^[8]从分解的文件中提取了操作码，并利用 N-gram 方法将其组织为特征向量，以此来进行恶意软件分类。此外，他们比较了不同操作码长度对分类精度的影响。实验结果表明，操作码是恶意软件分类的有效特征，并且短操作码长度具有更好的分类性能。

在某些情况下，N-Gram 提取的特征过于单一，可能会导致检测效率较低。为了提高检测精度，考虑将 N-Gram 特征与其他特征进行结合，或使用不同的 n 值提取特征来进行学习。之后，使用一种集成学习将多个分类器的结果进行组合，从而获得更准确的最终结果。这种方法可以在恶意软件检测任务中获得更高的精度和更好的性能。

（3）其他静态分析方法

除此之外，动态链接库、函数长度、PE 文件头、可变长指令序列、函数调用图细粒度分类、可阅读字符串和熵等信息都能够当作静态特征进行使用。

Tian 等^[9]使用函数长度作为静态特征，通过测量代码中函数的字节数来衡量。作者基于此提出了一种用于木马分类的方法。实验结果表明，在识别恶意软件家族的静态特征方面，函数长度及其频率具有重要作用。如果与其他特征相结合，可以更有利于恶意软件的分类。

Siddiqui 等^[10]选取可变长度指令序列作为特征，通过可变长度指令序列中包含的信息进行分析，使用机器学习中的决策树和随机森林机器来检测蠕虫并且进行分类，其中数据预处理过程如图 1 所示。

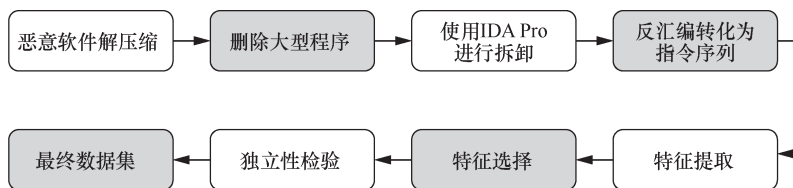


图 1 数据预处理过程
Figure 1 Data preprocessing process

Kong 等^[11]提出了一个用于自动分类恶意软件的通用框架，该框架基于恶意软件的丰富结构信息。该框架分为 3 个步骤：第一步，利用函数调用图的细粒度特征提取恶意软件样本的特征；第二

步，基于距离度量评估样本之间的相似度；第三步，对样本进行聚类。在完成这 3 个步骤之后，利用分类器从样本之间的距离中进行训练和分类。

Baysa 等^[12]采用二维矩阵进行恶意软件比

较, 但该方法的计算成本较高。该方法将结构熵技术应用于变异检测问题, 即通过分析文件的结构熵来实现相似性比较。该方法包括文件分割和序列比较两个阶段。在文件分割阶段, 作者使用熵测量和小波分析对文件进行分割; 在序列比较阶段, 作者通过计算在第一阶段获得的片段序列之间的距离来衡量文件对的相似性。

Li 等^[13]提出了一个纯粹基于静态特征的分类器, 选取的样本特征为文件头、熵、动态链接库(DLL, dynamic link library)等信息, 经过实验发现, 针对自动规避攻击, 该分类器具有很好的鲁棒性。

Kumar 等^[14]对 PE 头的原始值与派生值进行了组合, 还比较了所提出的集成特征集与原始特征集, 通过实验证明了基于前者训练出的分类器能够凭借较高的准确度与较少的成本对恶意软件进行检测和分类。

在恶意代码检测领域, 静态分析已经得到广泛的应用和发展。它可以全面地对恶意代码进行分析, 捕获语法和语义信息。相对于动态分析, 静态分析的时间复杂度和空间复杂度较低, 速度快, 效率高, 适用于大规模数据集。然而, 随着恶意代码混淆技术和反检测技术的不断发展, 基于静态分析的检测方法已经不能完全检测出恶意代码, 易受到加密技术的干扰。因此, 动态检测和混合检测方法逐渐受到重视和研究。

1.2 动态检测技术

恶意软件动态分析是指让恶意软件样本在受控环境(如模拟器、仿真器、虚拟机和沙箱)中执行, 并对其运行时的行为进行监控和分析。目前, 自动化工具如 Norman Sandbox、CWSandbox、Anubis、TTAnalyzer、Ether 和 ThreatExpert 等已经在动态分析领域广泛使用。通过使用这些工具进行分析, 可以获得恶意代码样本的分析报告, 并提供更有效的特征以用于恶意代码分类。

Firdausi 等^[15]首先使用了 Anubis 自动化工具在沙箱中对恶意代码进行动态行为分析, 并将得到的分析报告转化为稀疏向量模型; 然后选用了 K 近邻、朴素贝叶斯、决策树和支持向量机这 4 种分类器算法对恶意样本进行分类。实验结果表明, J48 决策树算法的分类效果优于其他分类器算法。

Zolkipli 等^[16]提出了一种基于动态方法的恶

意软件分类框架, 该框架包括行为识别和恶意软件分类两个阶段, 作者在 CWSandbox 和 Anubis 自动化工具上运行每个恶意软件样本, 并收集与行为相关的报告。该方法的缺点在于需要人工参与动态分析恶意软件的行为, 导致效率低。

Rieck 等^[17]提出了一个用于恶意软件行为监控的框架, 在沙箱环境中监控每个恶意软件样本的行为, 并将这些行为映射到向量空间中, 然后对具有相似行为的新型恶意样本进行了聚类。

Anderson 等^[18]提出了一种基于指令执行轨迹的恶意软件检测方法, 该方法使用动态分析技术收集目标可执行文件的指令轨迹, 并将其构建为图; 利用图核机制构造计算跟踪图之间的相似性, 再使用支持向量机进行分类。然而, 该方法在实际应用中的计算复杂度较高。

Kim 等^[19]提出了一种快速识别和检测恶意代码的算法, 同时能够快速响应网络入侵。该算法的核心思想是采用序列对比算法对 API 调用序列进行动态分析。

陈佳捷等^[20]提出了一种结合恶意代码动态行为和双向门循环单元的恶意软件检测方法。该方法使用 Cuckoo 沙箱对恶意代码进行行为分析, 从 API 调用序列和行为信息两个角度提取和融合特征, 并将双向门循环单元应用于恶意软件检测。实验结果表明, 该方法能够获取更加丰富的行为信息, 且相比双向长短时记忆(BiLSTM, bidirectional long short-term memory)网络等, 性能更为优越。

相比静态分析, 动态分析在恶意代码检测中更加有效, 拥有更高的准确率, 而且无须对恶意代码进行解包和反汇编操作。对于加壳以及混淆的恶意代码样本漏检率会进一步降低。然而, 动态分析的时间和空间成本会增加。此外, 动态分析无法对恶意代码进行全面的监控和分析, 缺乏全局认识, 只能监控和分析恶意代码程序的一条执行路径, 需要一定的跟踪时长才能取得有效结果。

1.3 混合检测技术

针对进行了加壳操作的恶意代码, 直接对其应用静态特征检测不太适用, 因为难以对其进行反汇编操作。虽然将这些加壳的恶意代码放在虚拟环境中检测可以一定程度上观察到它们的隐藏

行为,但是这种动态检测方式实用性不高,耗时长。因此,针对加壳的恶意软件,可以结合动态分析和静态分析两种方式进行检测。首先,使用动态分析对加壳的恶意软件进行去壳操作,然后通过静态分析检测去壳后的恶意软件,从而可以获得更加丰富的恶意代码特性。这种方式不仅可以提高检测的效率,还可以提高检测的准确率。

Islam 等^[21]在恶意代码分类中,将 3 个特征向量进行了集成。这 3 个特征向量分别为函数长度频率、可打印字符串信息矢量化和从日志文件中提取的 API 特性。其中,前两个为静态特征,后一个为动态特征。将这 3 个特征向量进行组合后,使用分类器进行训练,以实现恶意代码的分类。

Santos 等^[22]提出了一种结合静态特征和动态特征的恶意代码检测器。该检测器使用操作码频率作为静态特征,并选择执行程序的轨迹信息作为动态特征,通过将这两种特征进行结合,提高了恶意代码检测器的性能。

金炳初等^[23]提出了一种恶意软件检测和分类方法,该方法包含静态签名、样本灰度图等静态特征以及行为路径树动态特征。该方法首先通过爬虫抓取大量恶意样本的签名,并使用

ClamAV 进行检测;其次,对于未检测出的恶意样本,将其转化为灰度图进行分类;最后,利用行为路径树动态特征对于之前仍检测失败的样本进行恶意软件分类。

陈志锋等^[24]提出了一种内核恶意代码检测方法,该方法基于数据特征创建了内核数据对象访问模型,并采用静态分析和动态分析相结合的方法对恶意代码进行识别。在静态分析阶段,该方法通过对恶意代码进行反汇编获取代码特征,以构建内核数据对象访问模型。在动态分析阶段,该方法运行恶意代码并监测其内核数据对象的访问行为,与模型进行匹配以实现内核恶意代码的检测。该方法可以有效地识别和阻止内核恶意代码的攻击行为。

混合检测作为一种恶意软件检测方法,结合了静态检测的快速和高效以及动态检测的准确性,但其需要大量的资源,对于规模较大的数据集,其复杂度和工作量会相应增加。对于混合检测方法而言,何时对加壳的恶意软件进行去壳操作是一个关键问题,值得进一步深入研究。

传统恶意软件检测的相关文献如表 1 所示,其中“—”表示文献中没有明确提及。考虑到本文的研究重点以及非可视化相关文献的发表时间

表 1 传统恶意软件检测的相关文献
Table 1 Literature of traditional malware detection

文献与年份	静态/动态/混合	特征	机器学习算法	分类	数据集
2001 年 ^[1]	静态	DLL、字符串序列、字节序列	NB	机器学习	自定义数据集
2004 年 ^[6]	静态	字节序列	NB、DT、SVM	机器学习	SourceForge、VX Heavens
2006 年 ^[7]	静态	字节序列	NB、DT、SVM	机器学习	SourceForge、VX Heavens
2009 年 ^[10]	静态	可变长指令序列	DT、RF	机器学习	VX Heavens
2010 年 ^[15]	动态	稀疏向量模型	KNN、NB、J48 DT、SVM	机器学习	自定义数据集
2011 年 ^[16]	动态	行为信息	—	—	自定义数据集
2011 年 ^[18]	动态	指令轨迹	SVM	机器学习	自定义数据集
2013 年 ^[11]	静态	函数调用图细粒度	KNN、SVM	机器学习	Offensive Computing
2013 年 ^[21]	混合	函数长度频率、字符串信息矢量化、API	SVM、IB1、DT、RF	机器学习	自定义数据集
2016 年 ^[4]	静态	API	SAE	深度学习	Comodo Cloud、Security Center
2017 年 ^[13]	静态	PE 头、熵、DLL	SVM	机器学习	WINE
2019 年 ^[14]	静态	PE 头	DT、RF、KNN、LR、NB 等	机器学习	VirusShare
2020 年 ^[23]	混合	静态签名、样本灰度图、行为路径树	KNN、LMNN	机器学习	VirusSign、MalShare 等
2021 年 ^[20]	动态	行为信息、API	BGRU	深度学习	VirusShare

较早,很多文献使用的数据集已经过时并已被市面上新的数据集所取代,表1中未呈现非可视化相关文献的分类准确率。后文会展示可视化相关文献的分类准确率,这样做的目的是聚焦于介绍新的数据集和研究成果,为读者提供更加实用和现代化的信息。

在现有的分析方法中,仍以静态分析和动态分析使用居多。如上所述,静态分析是指对程序可执行文件进行分析而不执行程序。在静态分析中,通常使用的特征包括操作码频率分布、字节序列 N-Gram、控制流程图等。静态分析比动态分析更快、更高效,因为它具有结构属性相关的信息优势,如字节签名序列和文件内容中的异常。

而动态分析是指在一个安全和受控的环境中观察程序的运行状态,通常是分析行为信息,如网络活动、系统调用、文件操作和注册表修改记录。这种方法能够准确反映程序的行为特征,不受加密、压缩、变形等方面的影响。但是,这种方法需要花费时间调试程序和跟踪与记录程序的运行过程。因此,动态分析通常比静态分析效率低得多。此外,动态分析在运行环境中也受到一些限制。静态分析和动态分析有各自的优缺点,但考虑到动态分析所需的时间和资源,静态分析更适合于分析大量的恶意软件。

为了方便读者理解,对传统非可视化恶意软件检测进行文献逻辑图整理,如图2所示。

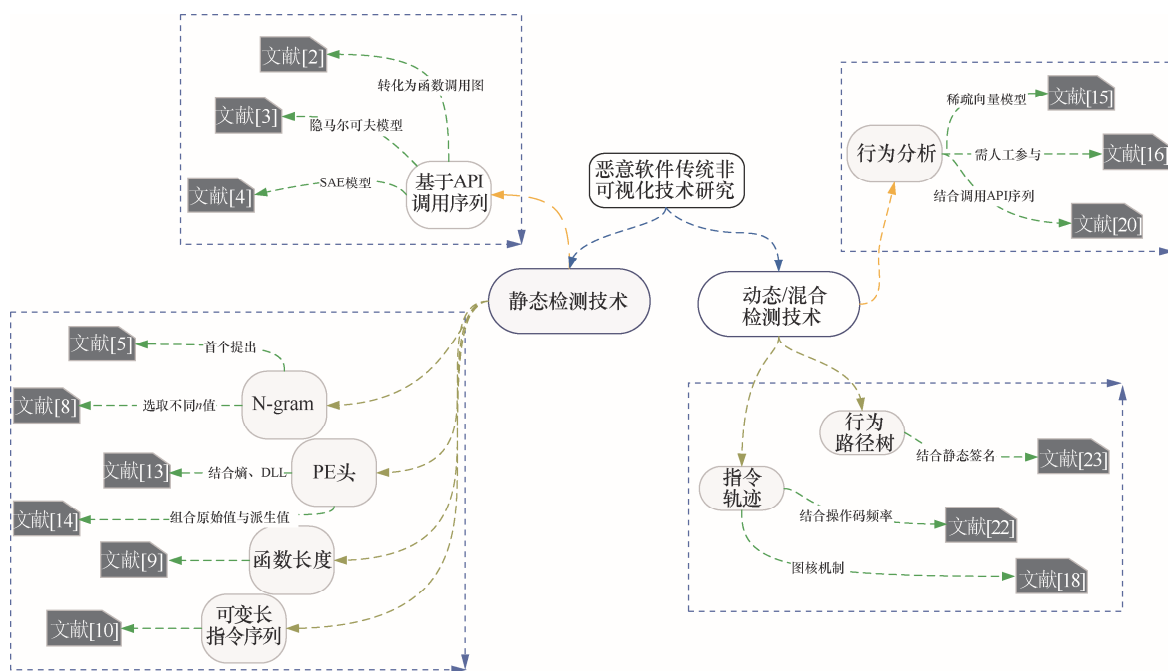


图2 对传统非可视化恶意软件检测进行文献逻辑图整理
Figure 2 Traditional non-visual malware detection for literature logic diagram arrangement

2 恶意软件可视化技术研究

尽管恶意软件不断演变和变化,但是将其转化为可视化图像后,相较于传统非可视化方法,具有更优越的检测和分类效果。恶意软件图像中蕴含了丰富的信息,同类恶意家族的可视化图像具有相似性,而不同类家族的可视化图像之间则有明显差异。此外,大部分恶意代码变体是使用自动化技术或重用重要模块生成的,因此它们在二进制代码中存在一些相似之处。基于以上特

点,研究学者采用两种主流方式开展恶意软件可视化研究:一种主流方式是从图像的纹理或结构等方面入手,提取特征并结合机器学习进行恶意软件的检测和分类;另一种主流方式是使用深度学习自动提取特征进行学习,可以有效对抗混淆技术。除了常见的灰度图像,为了获取更加丰富的特征,许多学者将恶意软件可视化为彩色图像。此外,还有一些基于可视化的方法,虽未使用机器学习或深度学习,但同样可对恶意软件进行有效检测。

2.1 基于机器学习的恶意软件可视化检测

当前, 针对基于机器学习的恶意软件可视化检测方法展开了广泛而深入的研究, 该研究方向的重心在于提取出具有良好分类效果且不易受到干扰的可视化特征。在特征提取的基础上, 利用多种分类器进行图像分类, 进而实现恶意软件的分类。通常采用的方法是利用机器学习方法对提取出的图片全局图像结构张量(GIST, global image structure tensor)特征进行学习和分类, 或者将 GIST 特征作为全局特征与 LBP、SIFT 或者 DSIFT 等局部特征相结合, 再运用机器学习方法进行分类。通过将分类效果最好的特征与分类器结合, 可以实现对恶意软件更为准确和高效的分类。

Nataraj 等^[25]的研究开启了结合可视化技术的恶意软件检测和分类的新兴领域。该研究首先将恶意软件 .text 区块的二进制数据通过 Nataraj 矢量化技术转化为灰度图像, 并基于 GIST^[26-28]对转化后的灰度图特征进行提取, 然后运用 KNN (K-nearest neighbor) 算法对提取的特征进行分类。此外, 在文献[29]中, 研究者表明使用图像处理中的二进制纹理分析技术可以更快速地对恶意软件进行分类。然而, 纹理分析方法具有较大的计算开销, 因此在处理大量的恶意软件时存在问题。

Nataraj 提出了一种针对恶意代码的矢量化方法, 该方法选取的编码量一般为全部 PE 文件或其中包含可执行代码的 .text 节。具体而言, 对于给定的二进制文件, 可以将其切割成长度为 8 bit 的子序列, 每个子序列可以通过二进制转换为 [0,255]范围内的数值。基于这种思想, 恶意代码文件可以被视为在[0,255]范围内的灰度图像。为了处理不同大小的恶意代码文件, 研究人员提出将图像的宽度设置为固定值, 高度根据文件大小

进行调整, 从而将恶意代码转换为长条状的灰度图像, Nataraj 的矢量化如图 3 所示。

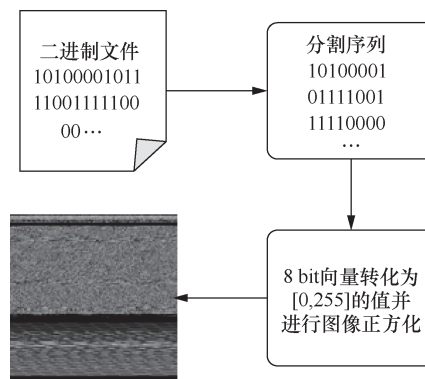


图 3 Nataraj 的矢量化
Figure 3 Vectorization of Nataraj

刘亚妹等^[30]解决了文献[25]在某些相似度比较高的恶意代码家族上分类精确度不高的问题。作者首先将恶意代码根据文献[25]中 Nataraj 矢量化方法转化为灰度图, 之后选取 GIST 作为全局特征, LBP 或者 DSIFT 作为局部特征进行融合。实验结果表明, 该方法更具稳定性和适用性, 并且分类准确率得到了显著提高。

刘亮等^[31]针对现有技术提取特征单一的缺点, 提出了一种基于多特征的恶意软件分类方法。该方法首先对字节码图像进行处理, 提取 GIST 作为全局特征和 SIFT 作为局部特征, 再将这两种特征融合并使用随机森林进行训练; 其次, 对操作码图像提取像素特征和文本特征进行融合, 并使用 XGBoost 进行训练, 此外, 还提取了灰度共生矩阵并使用支持向量机进行训练; 最后, 将这 3 部分分类器的输出通过联合层进行加权投票得出联合输出。3 层多分类器联合框架示意如图 4 所示。实验结果表明, 该方法的分类准确率明显优于单一特征提取的方法。

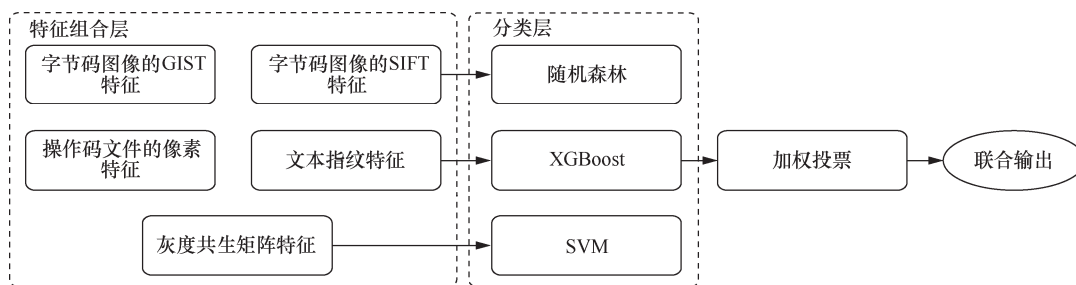


图 4 3 层多分类器联合框架示意
Figure 4 Three-layer multi-classifier joint framework

Naeem 等^[32]提出了一种名为 LGMP 的特征提取方法,该方法包括 3 个步骤:首先,采用 DSIFT 方法提取局部特征并进行选择;其次,提取 GIST 作为全局特征;最后,通过高斯权重集成局部和全局特征得到 LGMP 特征。

除了 GIST 特征之外,基于纹理以及操作码 N-Gram 提取特征等,也是恶意软件分类中常用的特征提取方法。这些特征可以结合多种其他特征进行组合,以进一步提高检测和分类的准确性。

Liu 等^[33]研发了一种基于机器学习的恶意软件分析系统,该系统由 3 个主要组成部分构成:首先是数据处理模块,该模块利用操作码 N-Gram 和导入函数对灰度图像进行特征提取;其次是决

策模块,负责对恶意软件进行分类和识别可疑恶意软件;最后是检测模块,该模块使用 SNN 聚类算法来发现新的恶意软件家族。

Fu 等^[34]提出了一种将恶意软件可视化为 RGB 彩色图像的方法,并利用熵、字节值和相对大小来表征恶意软件。作者将这 3 个特征分别转化为 RGB 三通道表示,基于熵、字节值和相对大小的彩色图片可视化方法如图 5 所示。在特征提取方面,作者选择了 GLCM 和颜色矩作为全局特征,同时为了增强模型的鲁棒性,提取了部分字节码序列作为局部特征,并使用随机森林、 K 近邻和支持向量机对恶意软件进行分类。

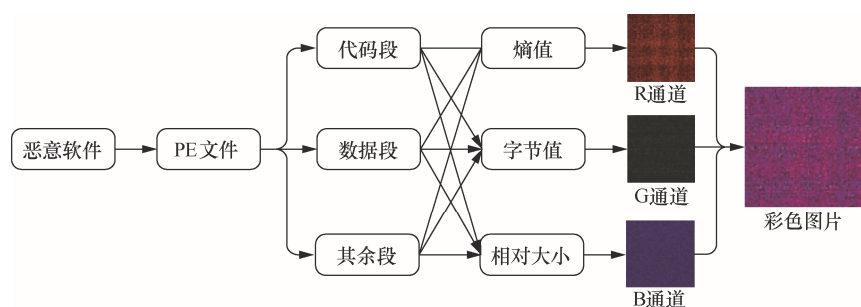


图 5 基于熵、字节值和相对大小的彩色图片可视化方法

Figure 5 Color image visualization method based on entropy, byte value and relative size

郎大鹏等^[35]提出了一种融合三组特征的恶意软件检测和分类算法。该算法首先将恶意代码源文件和反汇编文件使用文献[25]中 Nataraj 矢量化方法转化为灰度图;然后提取灰度图的 GIST 特征和 GLCM 特征,此外,使用 3-Gram 算法提取操作码序列,并采用改进型增益算法提取操作码特征;最后,将这 3 组特征进行不同的组合与分类,并利用随机森林算法进行学习。多特征融合的恶意代码分类算法如图 6 所示。

李劲杰等^[36]提出了一种基于多特征并结合

随机森林的恶意代码检测方法。该方法选取的特征包括 N-Gram 算法提取的文本特征、从灰度图中提取的纹理特征 GLCM 以及灰度直方图提取的颜色特征作为全局特征。3 种特征相结合后,利用随机森林算法对代码进行检测。

除了前文提到的方法,结合机器学习进行恶意代码检测的可视化技术还可以应用其他的特征提取方法,如热图、小波、Gabor 特征、方向梯度直方图 HOG 特征以及区段特征等。

Anderson 等^[37]提出了一种利用热图展示恶

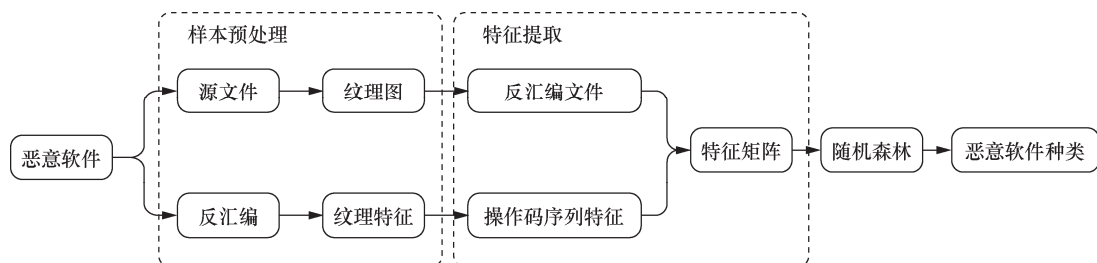


图 6 多特征融合的恶意代码分类算法

Figure 6 Malicious code classification algorithm based on multi-feature fusion

意软件内核使用情况的方法。该方法在每个不同视图上使用相似性度量来表示内核，然后使用多个内核进行学习，找到数据源的加权组合，从而在支持向量机分类器中实现最佳的分类精度。

卢喜东等^[38]提出了一种基于方向梯度直方图的恶意代码分类方法：首先，将恶意代码转化为灰度图并对其进行大小归一化处理；然后，使用方向梯度直方图提取灰度图的特征，这些特征被认为是恶意代码的本质特征；最后，使用深度

森林算法对恶意代码进行分类。

Kancherla 等^[39]提出了基于可视化的方法进行恶意软件检测。该方法首先将可执行文件转化为灰度图像，将其命名为“字节图”；随后，从这一图像中提取低层次的特征，包括基于强度和纹理的特征；最后，采用支持向量机作为恶意软件检测的工具，并利用上一步提取的特征进行分类和检测。

恶意软件可视化结合机器学习的相关文献如表 2 所示，其中“-”表示文献中没有明确提及。

表 2 恶意软件可视化结合机器学习的相关文献
Table 2 Literature of malware visualization combined with machine learning

文献与年份	图像类型	特征	机器学习算法	分类准确率/检测准确率	数据集
2011 年 ^[25]	灰度	GIST	KNN	98%/-	自定义数据集
2012 年 ^[37]	热图	内核	SVM	-/98.07%	Offensive Computing
2013 年 ^[39]	灰度	图像强度、小波与 Gabor 特征	SVM	-	Offensive Computing
2017 年 ^[33]	灰度	操作码序列、函数频率	DT、LR、NB、KNN、RF	98.9%/-	ESET NOD32/VX Heavens 收集
2018 年 ^[34]	彩色	纹理特征、颜色特征、字节序列	RF、KNN、SVM	97.47%/-	自定义数据集
2018 年 ^[30]	灰度	GIST、DSIFT 或 LBP	KNN、RF	NDB: 95.3%/- Antiy: 94.98%/-	Natarj/Antiy Databases (NDB 和 Antiy)
2018 年 ^[31]	灰度	GIST、SIFT、GLCM 等	RF、XGBoost、SVM	-	BIG2015
2019 年 ^[32]	灰度	GIST 和 DSIFT	KNN、SVM、NB	98%/-	Maling
2019 年 ^[35]	灰度	操作码序列和纹理特征	MLP、LR、RF	85%/-	malwaredb
2020 年 ^[38]	灰度	HOG	深度森林	96%/-	BIG2015
2020 年 ^[36]	灰度	GLCM、操作码序列、灰度直方图	RF	97.04%/-	BIG2015

根据表 2 的结果，可以探究不同特征和模型对机器学习方法分类准确率的影响。从表 2 可以看出，在特征选择方面，操作码和 GIST 相关的特征显示出相对较高的分类准确率；在模型选择方面，随机森林和支持向量机是在恶意软件检测和分类问题中较为高效的机器学习模型；此外，多方面多角度的特征合理结合比单一特征产生更好的分类效果。这进一步强调了特征和模型的选择对于分类准确率的重要性。在恶意软件检测领域，研究人员通常会测试多种特征提取方法，包括基于静态分析和动态分析的特征提取方法，并测试不同的机器学习模型，以获得更高的分类准确率。在进行特征和模型的选择时，必须考虑数据集大小、特征质量和可解释性、模型准确率和

计算效率等多个因素。因此，在选择特征和模型时，需要进行全面的分析和评估，以获得最佳的恶意软件检测和分类方法。

近年来，机器学习结合可视化的方法在恶意软件检测领域得到了快速发展，但仍有多个方面可以进一步完善。第一，目前的方法通常将整个恶意代码图像作为处理对象，而未来可以考虑将恶意代码图像分成多个区段进行处理。第二，在提取多特征时，应确保特征相互独立，尽量消除重叠的部分。第三，目前在特征融合方面，静态特征融合居多，未来可探索融合行为特征等动态特征。第四，对于加密或打包的恶意软件，需要开发有效的解压和解密方法以进行可视化研究。第五，可视化技术用于 PE 文件的研究更为广泛，

因为 PE 文件结构清晰统一、含有丰富信息,而一些非 PE 文件结构较为复杂,无法有效提取本地特征,因此未来对于非 PE 文件的可视化研究需要更多探索。第六,目前的方法在提取复杂的恶意软件图像特征时需要高计算成本,如 GIST、SURF、DSIFT、LBP 和 GLCM 等,而且这些特征提取技术在处理大规模数据集时效率较低。因此,未来需要研究如何降低特征提取成本,从原始二进制数据中提取适当的信息,提高恶意软件检测的准确率。

2.2 基于深度学习的恶意软件可视化检测

将恶意软件可视化与深度学习相结合,以实现恶意软件的分类和检测,是一种主流方法。

目前,深度学习技术,如卷积神经网络和循环神经网络等,已经成为广泛研究的热点。这些技术能够自动学习数据集中的特征,并实现高效的分类和检测。因此,基于深度学习的恶意代码检测得到了较快发展。其中,常用的方法是将恶意代码转换为灰度图或彩色图,并将其输入深度学习模型中进行学习和分类。

2015 年, Davis 等^[40]提出了一种基于反汇编的恶意代码矢量化方法。该方法使用反汇编的十六进制机器码作为编码源,将每个十六进制数字转换为 4 bit 的二进制数字,并将其填充到 64 bit 中。由于二进制数字只有 0 或 1,因此将每个二进制位乘以 255 后,可以得到像素灰度值 0 或 255。使用该方法转换后的灰度图中,每行都对应着一条机器码,具有很好的可视化解释性。

轩勃娜等^[41]提出了一种基于改进 CNN 的恶意软件 RGB 可视化分类方法,可抵御恶意软件变种和混淆,首先,通过基于 RGB 图像的特征表示方法,强化了恶意软件的语义关系,生成更富纹理信息的图像,深挖了恶意代码原始与变种的关系;其次,利用坐标注意力模块扩展空间信息,应对恶意软件的加密和混淆问题;最后,借助空洞空间金字塔池化改进 CNN 模型,克服图像尺寸归一化引起的信息损失和冗余。

Cui 等^[42]在文献[25]的基础上进行了改进,使用蝙蝠算法结合卷积神经网络的方法解决了不同恶意代码家族之间的数据不均衡问题,并应用深度学习进行分类和学习,避免了过拟合问题。

Kalash 等^[43]首先将恶意软件的二进制文件转换为灰度图像,然后训练了一个卷积神经网络来对其进行分类。之后,作者在 Malimg 和 Big2015 两个数据集上进行了测试,结果显示其准确率分别达到了 98.52% 和 98.99%。

田志成等^[44]提出了一种 PE 文件格式的恶意代码对抗样本检测方法,旨在应对在程序功能无关区域进行添加或修改的对抗样本攻击。该方法利用模型解释技术提取端到端的恶意代码检测模型的决策依据作为特征,并借助异常检测方法准确识别这些对抗样本。值得注意的是,该方法作为一种附加模块无须对原始模型进行修改。相对其他防御方法,如对抗训练等,该方法表现出更高的效率和更强的泛化能力,可有效应对多种对抗样本攻击。

王博等^[45]在恶意代码检测领域提出了一种新的方法,将每个二进制比特串切割成长度为 8 bit 的子串,并将每连续 3 个子串分别对应 RGB 通道。这种方法可以对任意长度的恶意代码进行处理,并将其转换为图像数据,基于序列分割的彩色图片可视化方法如图 7 所示。相较于文献[42],该方法的准确率提高了 1.66%。然而,该方法所设计的 CNN 结构参数量较大,且采用的数据集样本量过少,未能完全体现所设计 CNN 结构的优势。未来需要更多的研究来验证该方法的性能和可靠性。

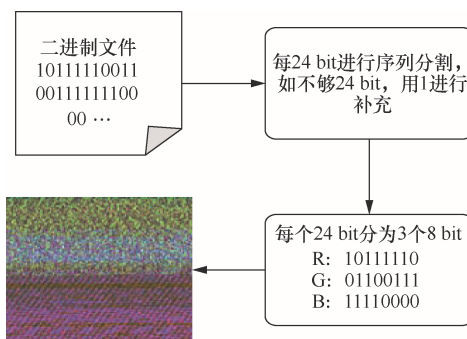


图7 基于序列分割的彩色图片可视化方法
Figure 7 Color image visualization method based on sequence segmentation

Chu 等^[46]通过灰度图转化技术将恶意代码同源分类问题转化为图像分类问题,利用基于恶意软件可视化图像的卷积神经网络构建分类器,并在 Big2015 数据集上进行了验证。实验结果表明,该分类器的分类精度高达 98.6%。

Vasan 等^[47]采用了文献[25]中的 Nataraj 矢量化方法将恶意软件转化为二维数组,随后添加了彩色映射以生成彩色图片,并利用调整过后的 VGG16 网络模型对其进行检测和分类。该方法对于混淆的恶意软件及其变种具有很高的效率和实用性。

针对文献[45]中模型数量过多的问题,蒋考林等^[48]提出了一种与其将恶意代码可视化为彩色图像类似的方法,但不同之处在于在末尾数量不足的情况下使用 0 进行填充,并使用 Alexnet 进行训练和分类。相比文献[45],文献[48]方法分类准确率提高了 1.8%。

除了前述方法,许多文献在结合可视化和卷积神经网络的基础上,引入了创新点。例如,一些作者使用 Simhash 或 Minhash 算法生成特征图像,以达到更好的分类效果。

Ni 等^[49]提出了一种 MCSC (multi-characteristic simhashing code) 算法,用于提取恶意代码特征。该算法结合了操作码序列和局部敏感哈希技术,通过获取恶意代码反汇编后汇编指令中的操作码序列进行特征提取,将每个恶意代码编码为相同长度的二进制 Simhash 值,随后,将 Simhash 值的每个二进制位转换为像素值(0 转换为 0, 1 转换为 255),从而得到恶意软件的图像。在该过程中,采用了多哈希、主块选择和双线性插值等技

术来提高算法性能。

Sun 等^[50]提出了一种结合恶意代码静态分析、RNN 和 CNN 的方法,旨在利用时序特征与原始代码信息的关联性来提高恶意代码分析的鲁棒性。该方法首先将原始代码与 RNN 预测码进行融合,并使用 Minhash 算法生成特征图像;然后,利用训练好的 CNN 对特征图像进行分类,实现恶意代码的检测和分类。

陈小寒等^[51]的研究既考虑了恶意软件的原始信息,又考虑了其与时序特征相关联的能力,通过二者的有机结合获得了更加丰富、可用于分类的信息。该研究首先从汇编代码中提取操作码序列,并采用 Simhash 算法将每段恶意代码映射为 64 bit 哈希值;其次,基于 Simhash 值映射的灰度图像可视化方法(如图 8 所示),将哈希值映射为一个坐标,通过对每个坐标出现的次数进行灰度值调整,生成恶意软件的灰度图像;随后,研究者利用双向递归神经网络处理灰度图像,得到预测码,同时,对预测码进行上述操作;最后,将原始代码与预测码特征图像结合,利用 CNN 对图像进行分类,取得了出色的分类效果。值得注意的是,该研究中所设计的 CNN 结构参数量较大,但作者对数据集进行了充分的实验设计和验证,从而保证了分类效果的可靠性和可复现性。

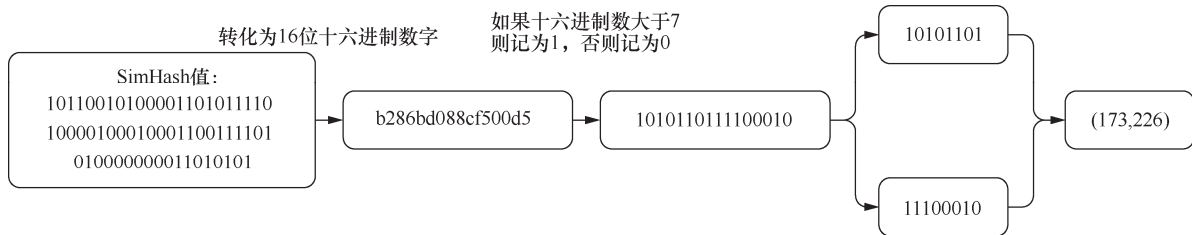


图8 基于 Simhash 值映射的灰度图像可视化方法
Figure 8 Gray image visualization method based on Simhash value mapping

也有一些研究者将深度学习和熵值或熵图相结合,用于恶意软件的检测和分类。

Gibert 等^[52]提出了一种针对恶意程序的检测和分类方法,将程序内容表示为一个熵流,其中每个值都描述了在文件中特定位置的一小块代码的熵量;随后,应用小波变换来描述熵能的变化,并且利用深度学习方法从熵值的流表示的可执行文件中找到区别模式,以实现对于恶意软件的检

测和分类。实验结果表明,该方法取得了良好的效果。

Xiao 等^[53]提出了一种将恶意代码样本转换为基于结构熵的熵图的方法,采用深度卷积神经网络作为特征提取器,自动从熵图中提取共享模式,并使用支持向量机对恶意代码进行检测和分类。实验结果表明,将深度卷积神经网络与熵图结合可以提供更好的恶意软件家族识别模式,熵

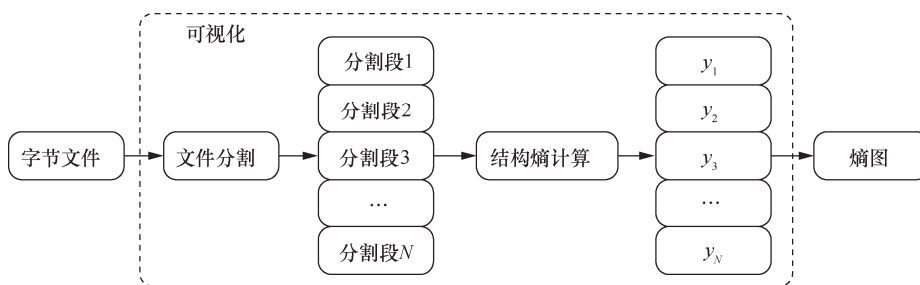


图 9 基于结构熵计算与熵图的恶意软件可视化方法

Figure 9 Malware visualization method based on structural entropy calculation and entropy graph

图比灰度图更具代表性，可以更好地表现核心家族信息。基于结构熵计算与熵图的恶意软件可视化方法如图 9 所示。

除此之外，还有一些研究者将深度学习与恶意软件的灰度图纹理特征、操作码频率、空间填充曲线、注意力图以及汇编指令的词向量等结合，以实现恶意软件的分类和学习。

Yan 等^[54]提出了一种新的恶意代码检测方法 MalNet，其首先生成恶意代码灰度图像，并同时利用 IDA 工具对操作码序列进行提取；随后使用 CNN 和长短时记忆（LSTM, long short-term memory）网络分别从灰度图和操作码序列中进行学习和分类，检测准确率达到 99.88%，分类准确率达到 99.36%。该方法通过将深度学习应用于灰度图像和操作码序列，取得了极高的检测和分类准确率。

乔延臣等^[55]在恶意软件检测中采用了一种基于汇编代码的方法。作者首先获取了恶意软件的汇编代码，并使用 Word2Vec 算法得到词向量；接下来按照 Top100 的顺序，排列每个样本上计算出的汇编指令词向量，将每个样本转换为维度为 100 的矩阵。通过归一化操作，将每个矩阵转换为灰度图像，并使用 CNN 进行分类。实验结果表明该方法可以有效地对恶意代码进行分类。

Yakura 等^[56]提出了一种新的恶意软件分析方法，该方法利用带有注意力机制的 CNN 处理恶意软件转化后的图像，计算出注意力图并提取具有代表性的恶意家族字节序列。实验结果表明，该方法能够提取有用的恶意家族信息，为人工分析提供帮助。

Zhao 等^[57]提出了一种基于灰度图像和纹理

分割的恶意代码分类框架，通过将恶意软件的信息反映在新的纹理特征图像空间中，实现了对恶意代码的分类。为适应该框架，作者还提出了一种专门设计的 CNN 进行学习和分类。

任卓君等^[58]在对文献[61]分析的基础上，针对样本数量较少、缺乏普适性的问题，提出了一种基于操作码频率的恶意软件可视化检测分类方法。该方法首先从反汇编的恶意样本中提取操作码字符序列；然后对其进行采样与转换；接着利用设计的色谱标记样本集中出现数量最多的前 15 种操作码，按照空间填充曲线遍历 RGB 颜色空间的顺序进行填充，使得相同颜色的操作码聚集在一起。基于操作码频率的恶意代码可视化分析方法如图 10 所示。通过这种方式，恶意软件信息被转换为具有代表性的可视化结果，可视化结果中同族相似，异族相异，具有很好的分类效果。

Ren 等^[59]提出了一种基于恶意软件二进制序列的可视化分析方法，该方法包括归一化、映射、学习与分类 3 个步骤。在第一个步骤中，通过填充不同的颜色来区分可显示字符和不可显示字符，分别使用绿色、黑色、白色、紫色标记可显示字符、字节值为 0 的字符、字节值为 255 的字符和其他字节。在第二个步骤中，采用 6 种空间填充曲线对恶意软件进行可视化。在第三个步骤中，使用 VGG19 卷积神经网络进行学习和分类。

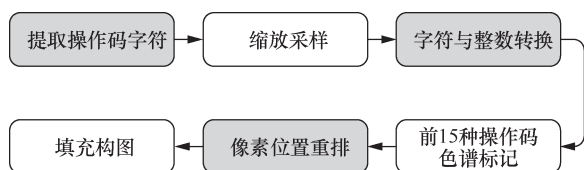


图 10 基于操作码频率的恶意代码可视化分析方法

Figure 10 Visual analysis method of malicious code based on opcode frequency

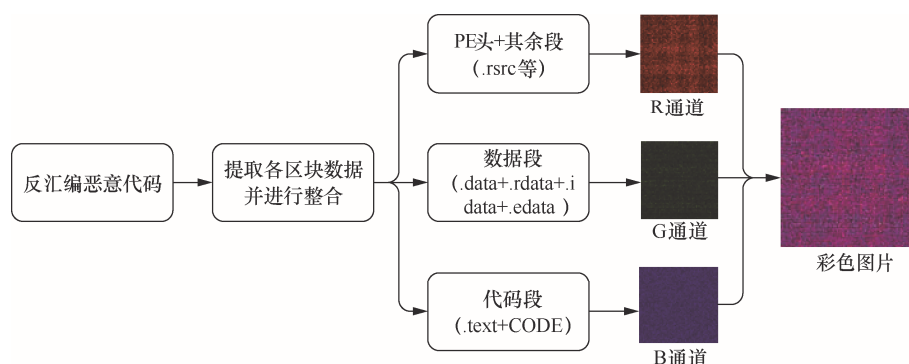


图 11 基于区段的彩色图片可视化方法
Figure 11 Color image visualization method based on segment

王润正等^[60]使用反汇编工具提取恶意代码中的不同区块数据, 然后对代码段和数据段进行分离并进行可视化操作。由于各个区块所代表的恶意代码信息不同, 使用该方法可以更直观地表现出不同恶意家族之间的差异性。基于区段的彩色图片可视化方法如图 11 所示, 实验结果表明, 采用该方法可以进一步提高恶意代码分类的准确性。

Pinhero 等^[61]进行了一系列实验, 结合恶意软件可视化和深度学习分类, 提出了一种新方法, 通过对不同维度的可执行文件进行视觉图像分析, 评估了经过微调的 CNN 模型在恶意软件识别上的性能。实验结果表明, 使用彩色视觉图像和马尔可夫图像可以取得最佳结果。此外, 应用 Gabor 滤波器提取纹理特征和熵图像进一步改进了分类准确性, 最佳的 CNN 模型在多个实验中展现出较高的 F-measure 值。未来将考虑扩展数据集和研究多模态图像分类器的性能。

Anandhi 等^[62]将恶意软件可视化为马尔可夫图像, 并利用 Gabor 滤波器提取纹理特征。作者通过构建 VGG-3 和 DenseNet 模型, 并对 DenseNet 进行微调, 实现了实时恶意软件检测和分类。在实验评估中, 作者发现 DenseNet 分别以 99.94% 和 98.98% 的准确率识别了 Malimg 和 BIG2015 样本。此外, 与现有方法相比, 作者提出的方法在根据预测时间、F1 分数和准确性进行校准的恶意软件文件分类方面表现出更好的性能。

Huang 等^[63]提出了一种恶意软件检测方法。该方法结合了恶意软件可视化技术和卷积神经网络, 并采用混合可视化的方式进行静态和动态分

析。作者通过 Cuckoo Sandbox 生成的行为分析报告, 将样本的静态特征和动态信息转换为可视化图像, 并在此基础上训练检测模型。实验结果表明, 混合可视化方法在恶意软件检测方面具有较好的性能, 优于仅使用静态分析的检测方法。

Moussas 等^[64]提出一种基于两级人工神经网络 (ANN) 的新型恶意软件检测系统, 解决了将现有恶意软件可视化为图像的变体的识别问题。研究人员从数据集中提取了关键的图像特征, 并用这些特征训练了 ANN 模型。该模型能够检测和分类数据集中的其他样本, 并通过第二级 ANN 对产生混淆的恶意软件家族进行分类。所提出的两级 ANN 方法在简洁性、准确性和速度方面表现出色, 易于实施且运行速度快。因此, 该方法可应用于防病毒软件、智能防火墙、Web 应用等领域。

Darem 等^[65]提出了一种半监督方法, 结合深度学习、特征工程、图像转换和处理技术, 用于检测混淆恶意软件。在检测混淆恶意软件方面, 作者所提出的方法达到了 99.12% 的检测准确率, 显著优于其他方法。

Conti 等^[66]提出了一种基于 GEM 图像的深度学习学习方法, 该方法更适用于较浅的卷积神经网络 (CNN) 架构, 以提高训练和分类效率。作者通过结合灰度矩阵图像、GLCM 纹理特征、马尔可夫图像和熵图像, 提出了一种新颖的恶意软件可视化方式, 同时与现有的 CNN 架构兼容。实验表明, 结合 GEM 图像格式和浅层 CNN 架构, 在传统的恶意软件分类问题上取得了有竞争力的结果, 并优于现有的少样本恶意软件分类研究。

Falana 等^[67]提出了一种基于深度卷积神经网络和深度生成对抗神经网络的集成技术 (Mal-Detect)，用于分析、检测和分类恶意软件。Mal-Detect 将恶意软件和良性文件转换为 RGB 图像，并利用深度生成对抗神经网络生成新的恶意软件图像，然后对生成的恶意软件图像、原始恶意软件和良性文件图像进行预处理，使用深度卷积神经网络训练数据集并提取关键特征。作者对 Mal-Detect 在 3 个基准数据集 (MaleVis、Mallimg 和 Virushare) 上进行了评估，结果显示其平均准确率为 96.77%。这些结果表明，Mal-Detect 可用于检测各类恶意软件。

Chaganti 等^[68]提出了一个名为 EfficientNetB1 的框架，用于对恶意软件家族进行分类。作者通过评估 3 种不同类型的恶意软件图像表示的性能，并对所提模型的网络参数进行有效缩放来对其进行评估。此外，作者对基于 CNN 的预训练模型进行了广泛的性能评估，比较了它们在恶意软件家族分类中的模型性能和计算效率。研究结

果表明，使用具有固定图像宽度的字节级表示的恶意软件获得了比使用文件大小和基于字节编码的图像宽度选择表示的恶意软件更好的性能。

Mallik 等^[69]提出了一种基于卷积循环的恶意软件分类技术。该方法通过将恶意软件样本转换为灰度图像，并利用卷积神经网络捕捉结构相似性，能够有效地对恶意软件进行分类。随后，作者通过堆叠的 BiLSTM 层处理提取的特征，通过合并 BiLSTM 层和 VGG16 层生成的输出，对恶意软件样本进行最终的恶意软件家族分类。

Chauhan 等^[70]提出了一种恶意软件分类方法，该方法利用不同颜色模式 (包括 RGB、HSV、灰度和 BGR) 将恶意软件文件转化为图像，以更清晰地展现其中的模式，用于图像分类。综合评估分类准确性、结果一致性、召回率、F1 得分和精确度等指标的结果表明，该方法具备使恶意软件分类更加高效的潜力。恶意软件可视化结合深度学习的相关文献如表 3 如所示，其中，“-”表

表 3 恶意软件可视化结合深度学习的相关文献
Table 3 Literature of malware visualization combined with deep learning

文献与年份	图像类型	特征	深度学习模型	分类准确率/检测准确率	数据集
2018 年 ^[54]	灰度	操作码序列	MalNet (CNN (BaseNet、VGGNet)、LSTM)	99.8%/99.36%	BIG2015
2018 年 ^[43]	灰度	-	VGG16 改进	Maling: 98.52%/- BIG2015: 99.97%/-	Maling、BIG2015
2019 年 ^[55]	灰度	汇编指令词向量	Lenet5 改进	98.56%/-	BIG2015
2019 年 ^[56]	灰度	字节序列，注意力图	添加注意力机制的自定义 CNN	-	VX Heaven
2020 年 ^[45]	彩色	-	VGG16 改进	96.16%/-	Maling
2020 年 ^[47]	彩色	彩色映射	VGG16	Maling: 98.82%/- IoT-android:97.35%/-	Maling 、IoT- android mobile dataset
2021 年 ^[51]	彩色	操作码频率	RNN、自定义 CNN	98.8%/-	BIG2015
2021 年 ^[58]	彩色	操作码频率、空间填充曲线	VGG16	98.50%/-	BIG2015
2021 年 ^[59]	彩色	空间填充曲线	VGG19	98.24%/99.02%	Kaspersky
2021 年 ^[48]	彩色	-	Alexnet	97.8%/-	Maling
2021 年 ^[60]	彩色	区段特征	DSCAM	98.38%/-	VirusShare
2021 年 ^[62]	Markov 图像	纹理特征	DenseNet 改进	Maling: 99.94%/- BIG2015: 98.98%/-	Maling、BIG2015
2021 年 ^[64]	灰度	-	ANN	99.13%/-	Maling
2022 年 ^[67]	彩色	-	Mal-Detect(DCNN、DGAN)	96.77%/-	MaleVis、Mallimg 和 Virushare
2023 年 ^[41]	彩色	二进制和汇编信息、API 信息间的语义关系	空洞空间金字塔池化改进的 CNN	Kaggle: 99.48% DataCon: 97.78%	Kaggle、DataCon

示文献中没有明确的提及。

深度学习结合可视化方法在恶意软件检测与分类方面表现出准确率高的特点。然而,目前仍存在以下需要进一步发展的方面。第一,图片提取的特征中大部分仍为静态特征,为应对混淆样本与加壳样本,可以考虑添加一些动态特征。第二,某些深度学习算法中设计的模型结构参数量过大,因此下一步需要对模型进行优化,减少参数量。第三,基于生成对抗网络(GAN, generative adversarial network)的数据增强技术在图像领域获得了广泛应用,可以考虑在恶意代码可视化图像上利用 GAN 进行图像增强,并将其与深度学习算法相结合,以提高分类效果。针对数据集分布不均衡的问题,尝试使用 GAN 解决该问题,以增强模型的泛化能力,提高恶意软件检测的准确率。最后,针对彩色图像,可以考虑将 3 个不同类型的信息分别转换为 RGB 三通道,进行彩色图片可视化,以进一步提高分类效果。

针对相同的恶意软件灰度图,机器学习方法和深度学习方法各具特色和优势。机器学习方法主要依靠手工特征提取和分类器,而深度学习方法则通过多层神经网络学习数据的抽象表示。机器学习方法具有简单易懂、容易实现的优点,适

用于小数据集,但因为需要人工选择特征和分类器,可能无法发现数据的潜在模式。在可视化结合机器学习方法中,可视化有助于分析和理解恶意软件的特征和行为,并提高模型的可解释性。通过可视化,研究人员可以直观地观察恶意软件在不同特征空间中的分布,从而优化特征选择、提取和分类器构建。深度学习方法无须手动进行特征提取和选择,从而更好地适应不同的数据分布和任务,具有更好的泛化能力和分类准确率。但深度学习方法需要更大的计算资源和训练时间,也需要更多的数据来支持模型训练。在可视化结合深度学习方法中,可视化不仅可以提高模型的可解释性,还有助于理解深度学习模型的工作原理和决策过程。综上所述,可视化结合机器学习方法和可视化结合深度学习方法各有优势和特点,选择恶意软件检测方法时应根据具体应用场景和需求综合考虑数据集大小、计算资源和时间限制、模型准确度和实现复杂度等因素。

为了方便读者理解,对恶意软件可视化检测进行文献逻辑图整理,如图 12 所示。

2.3 其他可视化方法

除了使用机器学习和深度学习方法进行恶意软件检测外,还有一些学者使用其他可视化方法

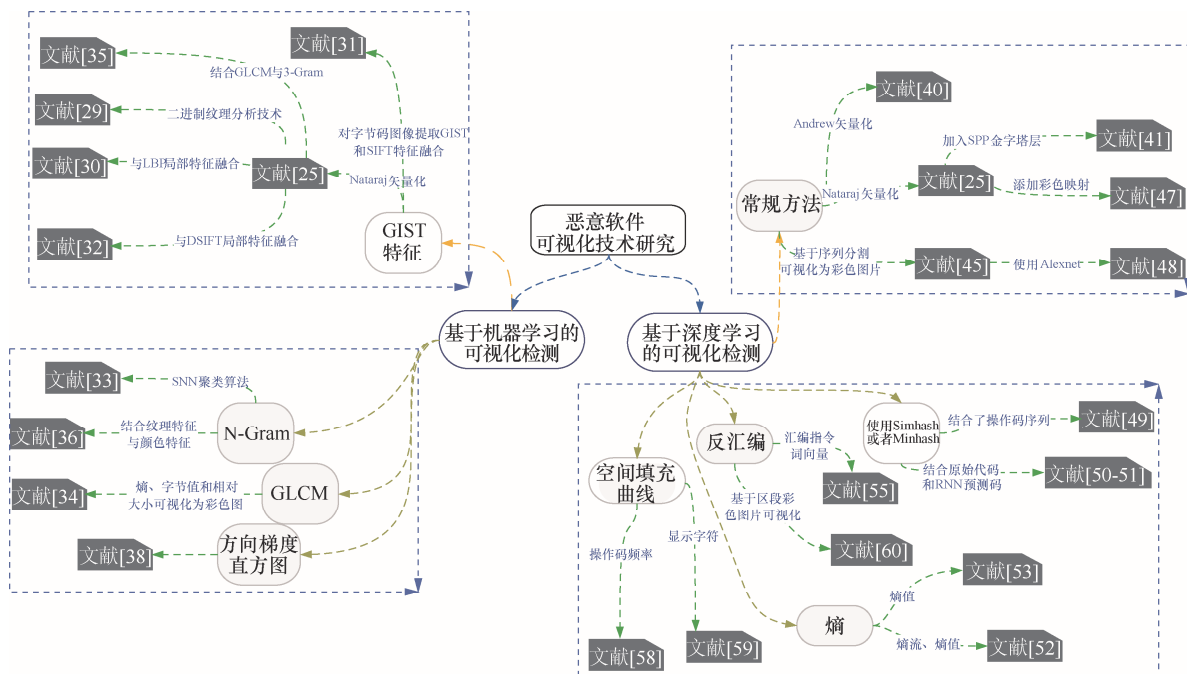


图 12 对恶意软件可视化检测进行文献逻辑图整理

Figure 12 Visual detection of malware for literature logic diagram arrangement

进行恶意软件检测。

Han 等^[71]提出了一种新的可视化分析方法, 通过将恶意软件的二进制信息转换为图像矩阵进行分析。该方法分为 3 个步骤: 首先从二进制文件中提取二进制信息; 接着生成将二进制信息记录为 RGB 彩色像素的图像矩阵; 最后计算它们中的相似性, 以便检测和分类恶意代码。此外, 作者还使用 Simhash 和 djb2 函数计算像素点的坐标值。在文献[72]中, Han 等在文献[25]Nataraj 向量化的基础上增加了熵图, 使用熵图的特征进一步提升恶意软件图像分类的效果, 对原本图像纹理特征提取方法和相似度衡量方案提供了一个新的思路。具体步骤为: 将 PE 二进制文件作为输入转化为位图图像, 并计算位图每一行的熵值, 基于熵值生成熵图, 最后将位图图像和熵图作为恶意软件的特征存储在数据库中, 通过计算熵图的相似性, 利用这些特征对恶意软件进行检测和分类。

此外, Paturi 等^[73]提出了一种利用可视化技术表达分层聚类的方法。他们使用标准化压缩距离来度量恶意软件程序之间的代码相似性, 并通过相似度量计算生成相似度得分矩阵, 从而确定正在检查的代码或二进制文件是否属于特定的恶意软件家族; 接着使用层次聚类将恶意软件家族分成集群, 并进行可视化展示。

韩晓光等^[74]提出了一种基于图像分析和灰度图检测的恶意软件检测方法。该方法首先将恶意软件转换为灰度图像, 然后采用纹理分割算法对灰度图像进行分块, 接着使用灰度共生矩阵算法提取灰度图像分块后的纹理特征, 并基于此建立纹理特征索引结构进行恶意软件检测。

3 结束语

本文全面阐述与分析了恶意代码检测技术, 特别关注了可视化领域的最新发展: 在传统非可视化检测方面, 本文从静态、动态和混合检测 3 个方面进行了概述; 在可视化检测方面, 从可视化结合机器学习、可视化结合深度学习和单独使用可视化 3 个角度进行了研究和综述。本文总结了当前恶意代码检测技术所面临的问题, 并提出了几个未来探究的方向。

第一, 现有的可视化方法仍然以灰度图和图

像相似性纹理为基础, 对于字节分布相对均匀的恶意代码并不适用, 需要进一步探索新的可视化方法。

第二, 目前可视化结合深度学习的恶意软件检测方法, 还是以卷积神经网络和循环神经网络为主, 使用图卷积神经网络和生成对抗网络进行恶意代码检测的方法较少, 因此可以考虑提出一种将恶意代码可视化、图卷积神经网络和生成对抗网络相结合的检测方法。

第三, 将恶意代码可视化为彩色图像后结合机器学习或深度学习进行恶意软件检测分类的研究仍然较少, 彩色图像拥有 3 个通道, 能反映恶意代码更加丰富的特性信息, 进而对检测和分类有更好的效果, 因此可以多尝试新的方法将恶意代码可视化为彩色图后进行分类。

第四, 在对恶意代码可视化为彩色图片时, 可以考虑 RGB 三通道分别代表 3 个不同类型的信息, 如 R 通道代表灰度图, G 通道代表字符信息, B 通道代表文件头信息。

第五, 在目前对恶意代码可视化为彩色图的方法中, PE 文件居多, 而非 PE 文件由于结构较为复杂, 无法有效提取本地特征, 因此需要对其可视化方法进行进一步的研究。

第六, 分类模型中选取的特征仍以全局特征居多, 从而导致分类模型对复杂的恶意软件的检测结果不够稳定, 只能应用于少量具有标记图像特征的恶意样本, 因此可以考虑更多结合全局特征与局部特征的方法, 提高模型的鲁棒性。此外, 有些特征提取方法需要高昂的计算成本, 如 GIST、SURF、DSIFT、LBP 和 GLCM 等, 尤其是在大规模数据集上的应用效率较低。因此, 如何通过更有效的方式降低特征提取的成本, 以从原始的二进制数据中提取更丰富的信息, 值得进一步探讨。

参考文献:

- [1] 国家互联网应急中心. 网络安全信息与动态周报[EB]. National Internet Emergency Center. Network security information and dynamic weekly report[EB].
- [2] IWAMOTO K, WASAKI K. Malware classification based on extracted API sequences using static analysis[C]//Proceedings of the Asian Internet Engineering Conference. 2012: 31-38.

- [3] IMRAN M, AFZAL M T, QADIR M A. Similarity-based malware classification using hidden Markov model[C]//2015 Fourth International Conference on Cyber Security, Cyber Warfare, and Digital Forensic (CyberSec). 2015: 129-134.
- [4] HARDY W, CHEN L, HOU S, et al. DL4MD: a deep learning framework for intelligent malware detection[C]//Proceedings of the International Conference on Data Science (ICDATA). The Steering Committee of The World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp). 2016: 61.
- [5] SCHULTZ M G, ESKIN E, ZADOK F, et al. Data mining methods for detection of new malicious executables[C]//Proceedings 2001 IEEE Symposium on Security and Privacy (S&P 2001). 2000: 38-49.
- [6] KOLTER J Z, MALOOF M A. Learning to detect malicious executables in the wild[C]//Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. 2004: 470-478.
- [7] KOLTER J Z, MALOOF M A. Learning to detect and classify malicious executables in the wild[J]. Journal of Machine Learning Research, 2006, 7(12).
- [8] KANG B J, YERIMA S Y, MC-LAUGHLIN K, et al. N-opcode analysis for android malware classification and categorization[C]//2016 International Conference on Cyber Security and Protection of Digital Services (Cyber Security). 2016: 1-7.
- [9] TIAN R, BATTEN L M, VERSTEEG S C. Function length as a tool for malware classification[C]//2008 3rd International Conference on Malicious and Unwanted Software (MALWARE). 2008: 69-76.
- [10] SIDDIQUI M, WANG M C, LEE J. Detecting internet worms using data mining techniques[J]. Journal of Systemics, Cybernetics and Informatics, 2009, 6(6): 48-53.
- [11] KONG D, YAN G. Discriminant malware distance learning on structural information for automated malware classification[C]//Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data mining. 2013: 1357-1365.
- [12] BAYSA D, LOW R M, STAMP M. Structural entropy and metamorphic malware[J]. Journal of Computer Virology and Hacking Techniques, 2013, 9(4): 179-192.
- [13] LI B, ROUNDY K, GATES C, et al. Large-scale identification of malicious singleton files[C]//Proceedings of the Seventh ACM on Conference on Data and Application Security and Privacy. 2017: 227-238.
- [14] KUMAR A, KUPPUSAMY K S, AGHILA G. A learning model to detect maliciousness of portable executable using integrated feature set[J]. Journal of King Saud University-Computer and Information Sciences, 2019, 31(2): 252-265.
- [15] FIRDAUSI I, ERWIN A, NUGROHO A S. Analysis of machine learning techniques used in behavior-based malware detection[C]//2010 2nd International Conference on Advances in Computing, Control, and Telecommunication Technologies. 2010: 201-203.
- [16] ZOLKIPLI M F, JANTAN A. An approach for malware behavior identification and classification[C]//2011 3rd International Conference on Computer Research and Development. 2011: 191-194.
- [17] RIECK K, TRINIUS P, WILLEMS C, et al. Automatic analysis of malware behavior using machine learning[J]. Journal of Computer Security, 2011, 19(4): 639-668.
- [18] ANDERSON B, QUIST D, NEIL J, et al. Graph-based malware detection using dynamic analysis[J]. Journal in Computer Virology, 2011, 7(4): 247-258.
- [19] KIM H, KIM J, KIM Y, et al. Improvement of malware detection and classification using API call sequence alignment and visualization[J]. Cluster Computing, 2019, 22(1): 921-929.
- [20] 陈佳捷, 彭伯庄, 吴佩泽. 基于动态行为和机器学习的恶意代码检测方法[J]. 计算机工程, 2021, 47(3): 166-173.
- CHEN J J, PENG B Z, WU P Z. Malicious code detection method based on dynamic behavior and machine learning[J]. Computer Engineering, 2021, 47(3): 166-173.
- [21] ISLAM R, TIAN R, BATTEN L M, et al. Classification of malware based on integrated static and dynamic features[J]. Journal of Network and Computer Applications, 2013, 36(2): 646-656.
- [22] SANTOS I, DEVESA J, BREZO F, et al. Opem: a static-dynamic approach for machine-learning-based malware detection[C]//International Joint Conference CISIS'12-ICEUTE'12-SOCO'12, 2013: 271-280.
- [23] 金炳初. 基于多特征的恶意软件分类方法[D]. 太原: 太原理工大学, 2020.
- JIN B C. Malware classification method based on multiple features[D]. Taiyuan: Taiyuan University of Technology, 2020.
- [24] 陈志锋, 李清宝, 张平, 等. 基于数据特征的内核恶意软件检测[J]. 软件学报, 2016, 27(12): 3172-3191.
- CHEN Z F, LI Q B, ZHANG P, et al. Kernel malware detection based on data features[J]. Journal of Software, 2016, 27(12): 3172-3191.
- [25] NATARAJ L, KARTHIKEYAN S, JACOB G, et al. Malware images: visualization and automatic classification[C]//Proceedings of the 8th International Symposium on Visualization for Cyber Security. 2011: 1-7.
- [26] TORRALBA A, MURPHY K P, FREEMAN W T, et al. Context-based vision system for place and object recognition[C]//IEEE International Conference on Computer Vision. 2003: 273-273.
- [27] OLIVA A, TORRALBA A. Modeling the shape of the scene: a holistic representation of the spatial envelope[J]. International Journal of Computer Vision, 2001, 42(3): 145-175.
- [28] SIAGIAN C, ITTI L. Rapid biologically-inspired scene classification using features shared with visual attention[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2007, 29(2): 300-312.
- [29] NATARAJ L, YEGNESWARAN V, PORRAS P, et al. A comparative assessment of malware classification using binary texture analysis and dynamic analysis[C]//Proceedings of the 4th ACM Workshop on Security and Artificial Intelligence. 2011: 21-30.
- [30] 刘亚姝, 王志海, 严寒冰, 等. 抗混淆的恶意代码图像纹理特征描述方法[J]. 通信学报, 2018, 39(11): 44-53.
- LIU Y S, WANG Z H, YAN H B, et al. Anti-obfuscation malicious code image texture feature description method[J]. Journal on Communications, 2018, 39(11): 44-53.
- [31] 刘亮, 刘露平, 何帅, 等. 一种基于多特征的恶意代码家族静态标注方法[J]. 信息安全研究, 2018, 4(4): 322-328.

- LIU L, LIU L P, HE S, et al. A static labeling method for malware family based on multiple features[J]. *Journal of Information Security Research*, 2018, 4(4): 322-328.
- [32] NAEEM H, GUO B, NAEEM M R, et al. Identification of malicious code variants based on image visualization[J]. *Computers & Electrical Engineering*, 2019, 76: 225-237.
- [33] LIU L, WANG B, YU B, et al. Automatic malware classification and new malware detection using machine learning[J]. *Frontiers of Information Technology & Electronic Engineering*, 2017, 18(9): 1336-1347.
- [34] FU J, XUE J, WANG Y, et al. Malware visualization for fine-grained classification[J]. *IEEE Access*, 2018, 6: 14510-14523.
- [35] 郎大鹏, 丁巍, 姜昊辰, 等. 基于多特征融合的恶意代码分类算法[J]. *计算机应用*, 2019, 39(8): 2333-2338.
- LANG D P, DING W, JIANG H C, et al. Malware classification algorithm based on multi-feature fusion[J]. *Journal of Computer Applications*, 2019, 39(8): 2333-2338.
- [36] 李劭杰, 王晨, 史峻. 基于多特征随机森林的恶意代码检测[J]. *计算机应用与软件*, 2020, 37(10): 328-333.
- LI S J, WANG C, SHI Y. Malware detection based on random forest with multiple features[J]. *Computer Applications and Software*, 2020, 37(10): 328-333.
- [37] ANDERSON B, STORLIE C, LANE T. Improving malware classification: bridging the static/dynamic gap[C]//*Proceedings of the 5th ACM Workshop on Security and Artificial Intelligence*. 2012: 3-14.
- [38] 卢喜东, 段哲民, 钱叶魁, 等. 一种基于深度森林的恶意代码分类方法[J]. *软件学报*, 2020, 31(5): 1454-1464.
- LU X D, DUAN Z M, QIAN Y K, et al. A deep forest-based malware classification method[J]. *Journal of Software*, 2020, 31(5): 1454-1464.
- [39] KANCHERLA K, MUKKAMALA S. Image visualization based malware detection[C]//*2013 IEEE Symposium on Computational Intelligence in Cyber Security (CICS)*. 2013: 40-44.
- [40] DAVIS A, WOLFF M. Deep learning on disassembly data[J]. *BlackHat USA*, 2015.
- [41] 轩勃娜, 李进. 基于改进CNN的恶意软件分类方法[J]. *电子学报*, 2023, 51(5): 1187-1197.
- XUAN B N, LI J. Malware classification method based on improved CNN[J]. *Acta Electronica Sinica*, 2023, 51(5): 1187-1197.
- [42] CUI Z, XUE F, CAI X, et al. Detection of malicious code variants based on deep learning[J]. *IEEE Transactions on Industrial Informatics*, 2018, 14(7): 3187-3196.
- [43] KALASH M, ROCHAN M, MOHAMMED N, et al. Malware classification with deep convolutional neural networks[C]//*2018 9th IFIP International Conference on New Technologies, Mobility and Security (NTMS)*. 2018: 1-5.
- [44] 田志成, 张伟哲, 乔延臣, 等. 基于模型解释的PE文件对抗性恶意代码检测[J]. *软件学报*, 2023, 34(4): 1926-1943.
- TIAN Z C, ZHANG W Z, QIAO Y C, et al. Adversarial malicious code detection for PE files based on model interpretation[J]. *Journal of Software*, 2023, 34(4): 1926-1943.
- [45] 王博, 蔡弘昊, 苏旻. 基于VGGNet的恶意代码变种分类[J]. *计算机应用*, 2020, 40(1): 162-167.
- WANG B, CAI H H, SU Y. Malware variant classification based on VGGNet[J]. *Journal of Computer Applications*, 2020, 40(1): 162-167.
- [46] CHU Q, LIU G, ZHU X. Visualization feature and CNN based homology classification of malicious code[J]. *Chinese Journal of Electronics*, 2020, 29(1): 154-160.
- [47] 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.
- [48] 蒋考林, 白玮, 张磊, 等. 基于多通道图像深度学习的恶意代码检测[J]. *计算机应用*, 2021, 41(4): 1142-1147.
- JIANG K L, BAI W, ZHANG L, CHEN J, et al. Malware detection based on multi-channel image deep learning[J]. *Journal of Computer Applications*, 2021, 41(4): 1142-1147.
- [49] NI S, QIAN Q, ZHANG R. Malware identification using visualization images and deep learning[J]. *Computers & Security*, 2018, 77: 871-885.
- [50] SUN G, QIAN Q. Deep learning and visualization for identifying malware families[J]. *IEEE Transactions on Dependable and Secure Computing*, 2018, 18(1): 283-295.
- [51] 陈小寒, 魏书宁, 覃正泽. 基于深度学习可视化的恶意软件家族分类[J]. *计算机工程与应用*, 2021, 57(22): 131-138.
- Chen X H, Wei S N, Qin Z Z. Malware family classification based on deep learning visualization[J]. *Computer Engineering and Applications*, 2021, 57(22): 131-138.
- [52] GIBERT D, MATEU C, PLANES J, et al. Classification of malware by using structural entropy on convolutional neural networks[C]//*Proceedings of the AAAI Conference on Artificial Intelligence*. 2018: 32.
- [53] XIAO G, LI J, CHEN Y, et al. MalFCS: an effective malware classification framework with automated feature extraction based on deep convolutional neural networks[J]. *Journal of Parallel and Distributed Computing*, 2020, 141: 49-58.
- [54] YAN J, QI Y, RAO Q. Detecting malware with an ensemble method based on deep neural network[J]. *Security and Communication Networks*, 2018, 2018.
- [55] 乔延臣, 姜青山, 古亮, 等. 基于汇编指令词向量与卷积神经网络的恶意代码分类方法研究[J]. *信息安全学报*, 2019(4): 20-28.
- QIAO Y C, JIANG Q S, GU L, et al. Research on malicious code classification method based on assembly instruction word vector and convolutional neural network[J]. *Information Network Security*, 2019(4): 20-28.
- [56] YAKURA H, SHINOZAKI S, NISHIMURA R, et al. Neural malware analysis with attention mechanism[J]. *Computers & Security*, 2019, 87: 101592.
- [57] ZHAO Y, XU C, BO B, et al. Maldeep: a deep learning classification framework against malware variants based on texture visualization[J]. *Security and Communication Networks*, 2019.
- [58] 任卓君, 陈光, 卢文科. 恶意软件的操作码可视化方法研究[J]. *计算机工程与应用*, 2021, 57(18): 130-134.
- REN Z J, CHEN G, LU W K. Research on opcode visualization method for malware[J]. *Computer Engineering and Applications*, 2021, 57(18): 130-134.
- [59] REN Z, BAI T. Malware visualization based on deep learning[C]//*2021 14th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI)*.

2021: 1-5.

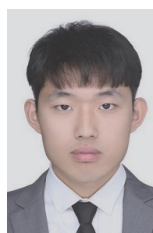
- [60] 王润正, 高见, 全鑫, 等融合注意力机制的恶意代码家族分类研究[J]. 计算机科学与探索, 2021, 15(5): 881-892.
- WANG R Z, GAO J, TONG X, et al. Research on malware family classification with attention mechanism fusion[J]. Journal of Frontiers of Computer Science and Technology, 2021, 15(5): 881-892.
- [61] PINHERO A, ANUPAMA M L, VINOD P, et al. Malware detection employed by visualization and deep neural network[J]. Computers & Security, 2021, 105: 102247.
- [62] ANANDHI V, VINOD P, MENON V G. Malware visualization and detection using DenseNets[J]. Personal and Ubiquitous Computing, 2021: 1-17.
- [63] HUANG X, MA L, YANG W, et al. A method for windows malware detection based on deep learning[J]. Journal of Signal Processing Systems, 2021, 93: 265-273.
- [64] MOUSSAS V, ANDREATOS A. Malware detection based on code visualization and two-level classification[J]. Information, 2021, 12(3): 118.
- [65] DAREM A, ABAWJY J, MAKKAR A, et al. Visualization and deep-learning-based malware variant detection using OpCode-level features[J]. Future Generation Computer Systems, 2021, 125: 314-323.
- [66] CONTI M, KHANDHAR S, VINOD P. A few-shot malware classification approach for unknown family recognition using malware feature visualization[J]. Computers & Security, 2022, 122: 102887.
- [67] FALANA O J, SODIYA A S, ONASHOGA S A, et al. Mal-detect: an intelligent visualization approach for malware detection[J]. Journal of King Saud University-Computer and Information Sciences, 2022, 34(5): 1968-1983.
- [68] CHAGANTI R, RAVI V, PHAM T D. Image-based malware representation approach with efficient net convolutional neural networks for effective malware classification[J]. Journal of Information Security and Applications, 2022, 69: 103306.
- [69] MALLIK A, KHETARPAL A, KUMAR S. ConRec: malware classification using convolutional recurrence[J]. Journal of Computer Virology and Hacking Techniques, 2022, 18(4): 297-313.
- [70] CHAUHAN D, SINGH H, HOODA H, et al. Classification of malware using visualization techniques[C]//International Conference on Innovative Computing and Communications (ICICC 2022). 2022: 739-750.
- [71] HAN K S, LIM J H, IM E G. Malware analysis method using visualization of binary files[M]//Proceedings of the 2013 Research in Adaptive and Convergent Systems. 2013: 317-321.
- [72] HAN K S, LIM J H, KANG B, et al. Malware analysis using visualized images and entropy graphs[J]. International Journal of Information Security, 2015, 14(1): 1-14.
- [73] PATURI A, CHERUKURI M, DONAHUE J, et al. Mobile malware visual analytics and similarities of attack toolkits (malware gene analysis)[C]//2013 International Conference on Collaboration Technologies and Systems (CTS). 2013: 149-154.
- [74] 韩晓光. 恶意代码检测关键技术研究[D]. 北京: 北京科技大学, 2015.

HAN X G. Research on key techniques for malicious code detection[D]. Beijing: University of Science and Technology Beijing, 2015.

[作者简介]



王金伟 (1978—), 男, 内蒙古呼伦贝尔人, 博士, 南京信息工程大学教授、博士生导师, 主要研究方向为多媒体版权保护、多媒体取证、多媒体加密和数据认证。



陈正嘉 (1999—), 男, 江苏徐州人, 南京信息工程大学硕士生, 主要研究方向为网络安全、信息安全、恶意软件检测。



谢雪 (1989—), 男, 吉林长春人, 中国科学技术大学博士生, 主要研究方向为网络安全、多媒体取证。



罗向阳 (1978—), 男, 湖北荆门人, 信息工程大学教授、博士生导师, 主要研究方向为图像隐写和隐写分析技术。



马宾 (1973—), 男, 山东济宁人, 齐鲁工业大学教授、博士生导师, 主要研究方向为可逆信息隐藏、多媒体取证、隐写与隐写分析。