基于深度学习的恶意代码检测综述

宋亚飞，张丹丹，王 坚，王亚男，郭新鹏
（空军工程大学防空反导学院，西安，710051）

摘要 恶意代码检测是网络空间安全领域的重要研究方向之一。在简要阐述恶意代码检测重大研究价值的基础上，结合国内外研究现状，总结了现有的基于深度学习的恶意代码检测技术及方法。首先，分别从静态、动态和混合检测方法多方面地梳理了传统检测技术，其次，分别从基于序列特征、图像可视化和数据增强的恶意代码特征提取方法出发，对基于深度学习的恶意代码分类识别方法进行了总结，最后，对基于深度学习的恶意代码特征提取与识别方向的技术难点和未来发展趋势进行了分析与展望。

关键词 恶意代码；恶意代码分类；恶意代码检测；深度学习；网络空间安全
DOI 10.3969/j.issn.2097-1915.2024.04.014
中图分类号 TP309.5 **文献标志码** A **文章编号** 2097-1915(2024)04-0094-13

Review of Malware Detection Based on Deep Learning

SONG Yafei, ZHANG Dandan, WANG Jian, WANG Yanan, GUO Xinpeng
(Air Defense and Antimissile School, Air Force Engineering University, Xi'an 710051, China)

Abstract Rapid and accurate identification of unknown malware and its variants is one of the important research directions in the field of cyberspace security. Based on a brief description of the significant research value of malware detection, the existing deep learning-based malware detection techniques and methods are summarized in consideration of the current situation of domestic and foreign research. Firstly, the traditional detection techniques are sorted out from static, dynamic and hybrid detection methods respectively. Secondly, the malware classification and identification methods based on deep learning are summarized from the malware feature extraction methods based on sequence features, image visualization and data enhancement. Finally, the technical difficulties and future development trends of malware feature extraction and identification based on deep learning are analyzed and foreseen.

Key words malware; malware classification; malware detection; deep learning; cyberspace security

恶意代码，也称为恶意软件，是一组旨在通过非授权操作破坏或窃取计算机系统中信息的程序或指令集合，以对数字系统的完整性、机密性和功能产生影响为目的，常见的恶意代码有勒索软件、后门、木马等[1]，详细类别如表1所示。恶意代码攻击导致个人和企业遭受网络威胁的风险急剧增加，如2017年的Wanna Cry勒索病毒攻击150个国家的30万名用户，造成高达80亿美元的损失[2]。《2021年上

收稿日期：2023-07-21
基金项目：国家自然科学基金(61806219，61703426，61876189)；陕西省科学基金(2021JM-226)
作者简介：宋亚飞(1988—)，男，河南汝州人，副教授，博士后，研究方向为机器学习及其在目标识别和入侵检测等领域中的应用。E-mail：yafei_song@163.com

引用格式：*宋亚飞，张丹丹，王坚，等. 基于深度学习的恶意代码检测综述[J]. 空军工程大学学报，2024，25(4)：94-106. SONG Yafei, ZHANG Dandan, WANG Jian, et al. Review of Malware Detection Based on Deep Learning[J]. Journal of Air Force Engineering University, 2024, 25(4): 94-106.*

半年我国互联网网络安全监测数据分析报告》显示，国家互联网应急中心捕获恶意程序样本约 2 307 万个，其日均传播次数达 582 万余次，涉及恶意程序家族约 20.8 万个[3]，由恶意代码攻击带来的网络威胁居高不下。当前主流的恶意代码攻击技术有代码注入[4]、缓冲区溢出[5]、端口复用[6]、协同攻击[7]等。为增强恶意代码的隐蔽性，攻击者使用大量变形技术如代码转置、重排等，创建恶意代码变体以逃避检测[8]。恶意代码检测的实质是分类问题，算法核心是检测代码与恶意代码之间的相似性，以此来判断是否为恶意代码，检测人员面对的不仅是固有结构的恶意代码，还有依靠混淆方法爆发式增长的变体。因此，如何高效准确地检测恶意代码，是目前网络安全领域的研究热点。

表 1　恶意代码类型

类型	行为
广告软件	通过提高商业广告的曝光率创造收益，或未经用户授权收集用户信息
勒索软件	通过加密或锁定系统中的文件限制用户的访问权限，胁迫受害者缴纳一定数额的赎金后才可解除限制
后门	绕过系统的安全防护机制，安装到系统上，便于攻击者访问
特洛伊木马	具有自动更名、自我隐藏复制的伪装式病毒装成合法程序诱使用户安装，或嵌入后门功能使攻击者绕过安全程序进入系统收集重要信息以及控制系统
间谍软件	未经用户允许进行一系列间谍活动并收集用户的敏感信息
病毒	利用操作系统的漏洞，对用户系统的功能以及数据造成损坏，使系统失去可用性，并可以通过设备进行传播
蠕虫	与病毒类似，都是对系统造成巨大损坏，但它可以通过网络进行自我复制和传播
Rootkit	加载到系统内核中的驱动级恶意程序，可以隐藏其他程序的进程并保留 Root 权限，通常与木马、后门等多种恶意软件结合，用于隐藏恶意踪迹

近年来，基于深度学习的恶意代码分析方法已成为继传统方法、数据挖掘和模糊测试后的首要方法，其识别流程如图 1 所示。深度学习受学术界广泛重视的原因，除了在理论层面实现重大突破外，还得益于其与在人力物力上花销巨大的传统方法不同。深度学习是机器学习的一个子集，其功能和架构受人类大脑启发，能够用于自动特征提取、识别图像和文本等其他基于信号处理的隐藏模式。传统的方法，如静态、动态或混合分析方法，从恶意样本中提取不同级别的特征进行识别和分类，无法高效准确地执行。深度学习作为一种基于分层的结构，前几层负责特征提取，最后一层用于分类，通过使用有监督、无监督和混合架构进行检测。目前国内外对基于深度学习的恶意代码检测技术进行全面综述的文献甚少。文献[9～13]对恶意代码检测技术进行了全面的回顾和综述，涵盖了该领域的大部分研究内容。然而，关于深度学习在恶意代码检测方面的应用研究在其中仅占据了较小的一部分。其中，申培等[9]对恶意代码检测研究进行了综述，但其中提及的关于深度学习的检测研究寥寥无几。王志文等[10]对基于机器学习的恶意软件识别研究进行了全面综述，然而，对于基于图像、序列化及数据增强的恶意代码检测技术的总结尚不够完善。目前关于深度学习的恶意代码检测研究仍被忽视，存在一定的不足。因此，本文对近年来基于深度学习的恶意代码检测相关研究工作进行了归类和梳理，以样本特征表示的获取方法为主线，对不同类型的恶意代码检测方法进行全面地调研分析，从传统恶意代码检测方法、基于深度学习的恶意代码检测技术、当前恶意代码检测方法所面临的问题 3 个方面系统地进行探究，总结了近些年针对恶意代码检测的新兴技术，分析基于深度学习的恶意代码检测方法所面临的形势，提出未来可能的研究方向，旨在为提高恶意代码检测效率提供新思路，进一步推动该领域的发展。

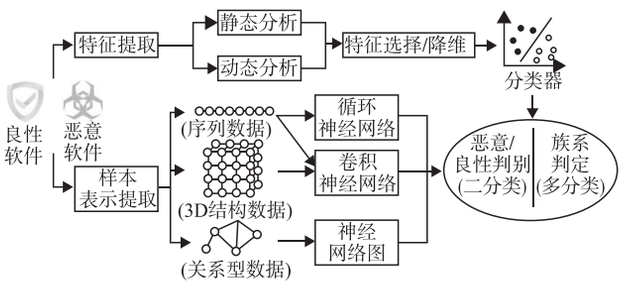

图 1　恶意代码识别流程

1　传统恶意代码检测方法

传统恶意代码检测方法主要通过人工分析恶意代码函数进行检测，不仅时间和资源成本高昂，且检测效率低、实用性差。依据是否需要运行程序可将传统恶意代码检测技术分为静态检测、动态检测与混合检测。

1.1　静态检测方法

静态检测依据汇编及二进制文件的结构和其调用函数等来提取特征，主要应用于软件开发和测试阶段。

Griffin 等[14]结合汇编代码和马尔可夫模型，改变以往使用单一特征码进行检测的模式，实现了多复合特征码检测。Yamaguchi 等[15]将抽象语法树、

控制流图和程序依赖图合并到联合数据结构中，使用图像遍历的方式对恶意代码的源代码进行分析。张炜康等[16]通过构建特征数据库保存敏感应用程序编程接口(application programming interface，API)序列特征，将样本的 API 序列与特征数据库进行匹配以进行检测。Coull 等[17]对反汇编二进制代码进行解析，将单个字节嵌入与原始语义相联系以分析样本特征。Kumar 等[18]集成了可移植可执行文件标头字段的原始值与派生值，并输入机器学习分类器以检测恶意代码。Alasmary 等[19]使用抽象图结构来分析恶意代码，通过控制流图构建物联网恶意代码检测机制。Taheri 等[20]使用汉明距离查找第一最近邻、所有最近邻、加权所有最近邻和 k 中心近邻的恶意样本间的相似性。Rezaei 等[21]使用基于标头和可移植可执行文件结构提取的特征来训练机器学习模型。随着黑客的攻击手段不断升级，使用静态分析方法检测经混淆技术操作的恶意代码变种时易发生误报或者漏报，并且难以处理动态类型的数据和复杂的程序行为。恶意代码攻击方式的先进化使得仅能提取部分特征的静态分析方法略显单薄。

1.2 动态检测方法

动态检测是将恶意代码样本在诸如虚拟环境、模拟器和沙盒等安全环境中运行，监测样本调用系统资源的行为，得到恶意代码运行特征，主要应用于软件运行和维护阶段。

赵晓君等[22]提出了利用半监督聚类算法分析 API 调用序列的恶意代码检测方法。Shaid 等[23]通过在虚拟环境中执行样本来收集恶意代码行为信息，然后使用颜色映射将收集到的 API 调用序列和操作系统资源转换为彩色图像，通过统计方法计算这些图像之间的相似度。Galal 等[24]通过 API 挂钩技术从提取的 API 调用和恶意代码运行的参数中收集信息并推断出特有的恶意代码行为。Mohaisen 等[25]提出一种仅与高级事件发生的时间顺序有关的系统，该系统通过映射关系的简洁级联捕获样本的执行动态，应用 n-gram 文档分类技术对恶意代码家族进行分类。Amer 等[26]使用词嵌入来理解恶意代码 API 调用序列中的上下文关系，基于 API 调用序列的行为进行建模生成语义转换矩阵来进行恶意代码检测。Anderson 等[27]提出了基于流量上下文和背景知识的 TLS 指纹构建方法和 TrafficAV 模型的流量荷载特征检测方法，对网络系统和移动端恶意代码行为进行检测和分析。部分恶意程序通过替换系统服务修改内存数据结构以便于自我隐藏，导致一般的取证工具难以发现其运行过程[28]。Tien 等[29]和 Kumara 等[30]提出了基于虚拟机侦测和利用虚拟机的内存取证技术，以获得流程详细信息的语义视图。部分恶意代码在运行时会申请更多的内存等资源，通过检测硬件信息使用情况可以发现异常信息。Bridges 等[31]基于 CPU 功率配置文件特征，提出一种无监督恶意代码检测方法。Sayadi 等[32]侧重研究硬件特征与恶意代码之间的关系，通过分析集成的处理器硬件性能计数器等数据对恶意代码进行实时检测。潘绯等[33]基于物理特征的认证对无线传感网中节点行为进行分析，识别其中的恶意节点。基于程序执行时的流程、行为、流量和物理等特征的动态分析对于恶意代码二进制转换检测很有用，且不会对二进制转换产生不利影响，但大多数动态分析方法仅适用于小规模恶意代码检测，他们使用过多的资源进行静态和动态视觉分析以确定恶意代码的行为，以及虚拟环境的使用非常耗时，这都会导致大规模恶意代码检测的计算开销增加。

1.3 混合检测方法

对于加壳的恶意代码，难以对其进行静态检测，通过动态分析可检测到虚拟环境中运行的加壳恶意代码的隐藏行为，但这种方式存在安全隐患，且实用性与可扩展性不强。基于此，混合检测方法应运而生，首先动态分析加壳恶意代码并进行脱壳处理，再对脱壳代码进行静态分析。这种结合动态检测与静态检测的方式，能够有效利用两者优势进而全面地获取恶意代码特征[34]，主要应用于软件分析阶段。

Islam 等[35]提出多特征集成的恶意代码分类方法，将静态特征中的函数长度频率与可打印字符串信息等和动态特征中的 API 函数特征与执行程序的轨迹信息等结合为集成特征馈入分类器训练。梁光辉等[36]提取恶意代码的模糊哈希特征和动态行为特征，并结合无监督聚类学习与有监督分类学习用于恶意代码检测。但由于其复杂的检测模式往往需要消耗大量资源，需要手动分析程序的行为，对分析者的技能要求较高，难以迎合大规模数据集的应用需求。

1.4 模糊测试

模糊测试是一种常用的恶意代码检测技术，它通过向目标程序输入大量随机或者半随机的数据，来检测程序是否会出现异常或崩溃。模糊测试可以帮助发现程序中的漏洞和错误，进而提高软件的安全性和稳定性。

在 2005 年之前，模糊测试还处于发展的早期阶段。这一时期的模糊技术主要是利用随机突变的黑盒模糊。第一个模糊系统是在 1990 年由 Miller 建

立的[37]。这是一个典型的随机模糊方法，它通过输入随机字符串来测试目标程序，观察程序是否崩溃，继而分析崩溃。该模糊器包含 2 个程序：Fuzz 和 Ptyjig。Fuzz 生成一个随机字符流，供目标程序使用，Ptyjig 通过编写脚本自动测试交互式实用程序。评估显示，在 7 个版本的 UNIX 中，90 个不同的实用程序中有 24%以上发生了崩溃。从那时起，模糊技术广泛应用于软件安全检测。

覆盖引导式模糊由于其反馈机制比随机模糊更有效[38]。Liu 等[39]提出了用于恶意代码分类的 MalFuzz 框架。MalFuzz 使用覆盖引导式模糊测试的思想，通过不断更新模型状态来更快地检测模型分类错误。该方法使用第 1 层和最后一层神经元值来近似表示模型状态，进一步解决了模型状态表示问题。考虑到新的覆盖率计算问题，MalFuzz 使用快速近似最近邻算法来计算新的覆盖率。实验结果表明，MalFuzz 可以快速探索模型的新状态，并生成更多使模型出错的测试用例。然而，生成测试用例以通过应用程序中复杂的检查对于这个模糊测试器来说仍然是具有挑战性的，并且覆盖引导的模糊测试所使用的反馈回路并没有将应用程序行为与输入结构联系起来，从而需要改进输入生成。

此外，还有研究人员引入哈希算法来提高模糊测试的效率。Nitin 等[40]提出了一种用于恶意软件分析的模糊导入哈希方法，以提高相似度检测的准确性和分析的性能。该方法结合了模糊哈希法和导入哈希法，将模糊导入哈希方法的相似性检测性能与独立模糊哈希（sdeep、SDHASH 和 mvHASH-B）和导入哈希方法进行比较，结果表明，提出的模糊导入哈希在总体检测率上有一定的提高。然而，这种方法难以被广泛地用作一种通用的分析方法。

模糊测试提供了一种检测程序漏洞的新方法，但对于具有复杂结构化输入数据的程序，它的测试结果很差，因为大多数突变输入在程序执行的早期步骤中被拒绝，无论是在解析还是在校验和验证过程中。

1.5　验证类技术

基于签名的检测技术不可避免地意味着恶意代码的抽象定义，即为检测规则。为了有效定义检测规则，需要网络、安全、操作系统等方面的专业知识。不恰当的检测规则操作会产生大量误报，使安全系统性能不佳，甚至导致安装网络安全系统的整个网络瘫痪。专家创建的检测规则具有精度高的优点，创建错误或效率低下的可能性较小，但缺点是创建和验证检测规则需要花费大量时间。

针对这一问题，Kim 等[41]提出并开发了一种基于签名的检测规则生成与验证系统，该系统利用潜在狄利克雷分配算法进行特征提取和流量分析。该项研究对开发的系统进行了实验，实验结果表明，相较于以往的方法，该系统在检测规则的创建与验证过程中，展现出了更快的速度和更高的准确性。除此之外，基于模型检查的恶意代码检测方法也比传统方法更准确地捕获安全漏洞的语义，从而实现更高的检测率。Holzer 等[42]使用基于经典计算树逻辑的表达性规范语言形式化程序的恶意行为，以集成规范开发过程。这些研究都支持着未来的自动化恶意代码分析和规范提取。

2　基于深度学习的恶意代码检测

深度学习作为一种强大的人工智能工具，已应用于语音识别和计算机视觉等军事科技相关领域[43-44]。传统上对恶意代码的研究主要采用静态、动态和混合分析，但自 2011 年 Nataraj 等人提出并实现了基于恶意代码可视化的机器学习分类方法，尤其是得出了比传统方法更行之有效的结论，研究者们的视线开始向机器学习和可视化方向转移，如决策树、深度学习网络和迁移学习等[45-48]。Nataraj 等[49]提出同一恶意代码家族的灰度图像往往呈现出相似的纹理，将样本的二进制结构特征转换为图像纹理特征进行分析更有分类优势。他们将压缩的二进制可执行文件结构转换为灰度图，然后提取 GIST 特征并使用 K-最近邻（K-nearest neighbor，KNN）分类技术进行分类。在此基础上，Han 等[50]利用图像相似矩阵技术对恶意代码变体进行分类，通过收集操作码指令等二进制信息在图像矩阵上生成 RGB 彩色像素，使用选择性区域匹配计算图像矩阵之间的相似度。Han 等[51]将恶意代码二进制的分割信息转换为灰度图，然后根据这些灰度图生成熵图，使用直方图相似性度量方法来度量图像的相似性。Vu 等[52]将二进制文件中的字节编码和排列成图像，并与机器学习相结合。这些方法有助于处理大量未打包的恶意代码二进制样本，但利用熵图对压缩恶意代码二进制样本进行相似性计算是一项非常困难的任务。机器学习的主要缺点是手动地进行特征提取[53]，模型无法自动提取和学习恶意图像的特征，这使得基于机器学习的恶意代码检测方法难以应对日益增加的恶意代码及其变种，应运用更智能的框架或分类器进行分类。Greengard 等[54]的研究表明，大多恶意代码都是由已知恶意代码变种而来，其差异性不到 2%。恶意代码可视化与以图像形式翻译恶意代码的原始二进制文件有关，通过

可视化的方式能够直观地区分和比较图像形式的恶意代码文件的不同代码部分[55]。因此，本文主要集中在近几年基于深度学习和可视化的恶意代码检测与分类方法及其创新。

2.1 基于图像可视化的深度学习方法

早在 2011 年，Nataraj 等[49]就提出以灰度图像的形式表示恶意代码二进制文件，基于图像的 GIST 特征来量化纹理的相似性，并利用 KNN 算法对转换后的灰度图像进行分类。研究提出的恶意代码可视化对解决现代恶意代码分类问题起到了不可或缺的作用，并成为恶意代码检测领域的里程碑，研究者们后续开展了很多相关研究。可视化方法是通过提取纹理特征检测图像之间的视觉相似性，其中细微变化通过可视化可以度量，这也证明了同一家族结构的相似性和执行恶意代码文件的非必要性。本文展示了基于深度学习的恶意代码检测方法，并对这些工作进行了比较，见表 2。

受此启发，Han 等[50]提出将恶意代码二进制文件映射到 RGB 图像矩阵，首先对恶意代码二进制文件进行分解，然后提取操作码序列并存储在块中，每个块都由 Simhash 和 djb2 哈希函数处理生成图像矩阵的坐标和 RGB 像素信息，在比较 RGB 图像矩阵时，使用了选择区域匹配。这种新颖的彩色图像表示方法利用了高维度特征，在每像素包含更多数据信息的彩色图像中能够提取更饱满的特征，有效地解除恶意代码图像处理模式单一和手动特征提取的限制。然而，由于分类技术的不成熟，实现的分类结果不尽人意。Pal 等[56]对原始数据进行预处理，应用均值归一化使训练数据的亮度归一化到每个图像维度，对原始数据进行零分量分析将图像的边缘转换得更尖锐，实验在不同的卷积神经网络(convolutional neural network，CNN)模型上进行。研究表明，对原始数据使用预处理技术能够有效提高分类精度，基于数学的预处理技术将训练图像的原始数据和边缘特征转换得更深刻，而 CNN 从这些边缘中检测到更多优化的特征，实现了提高精度的目标。Kornish 等[63]提出将汇编代码转换为图像是一个耗时的过程，因此作者使用汉明距离将原始的二进制样本表示为图像，并使用了预先训练好的深度 CNN 模型 Alexnet、VGG16 和 VGG19 用于分类。Ni 等[62]从恶意代码中提取操作码序列进行编码并基于哈希函数转换成图像，并研究了哈希算法、主块选择和双线性插值等图像处理方法对模型性能的影响，并利用 CNN 对恶意图像进行分类以比较序列之间的相似性。

许多研究人员直接沿用文献[49]中提出的同一家族的恶意代码变体具有相同的纹理和视觉特征的概念，将恶意代码二进制文件转换为灰度图像，将其作为神经网络的输入。Kalash 等[55]将灰度图像馈入预训练好的 VGG 模型进行分类，经过超参数微调后的模型在经典的 Malimg 和 Kaggle 恶意代码数据集上都获得了出色的结果。高宁等[76]提出基于挤压激励网络的恶意代码检测方法，在 CNN 中引入挤压和激励(squeeze-and-excitation，SE)模块，充分利用彼此在特征提取方面的优势。褚堃等[77]将 Android 字节码映射成二维表示，在此基础上，提出一种融合深度可分离卷积和全局注意力模块的深度学习模型。得益于深度学习的优势，该方法可以自动学习 Android 应用程序特征，并且无须使用反编译工具。

在此基础上，近几年涌现了许多基于多通道图像恶意代码检测方法，Singh 等[64]将恶意代码二进制文件通过 RGB 值表示为彩色图像矩阵，应用经典深度神经网络 ResNet-50 对恶意代码家族进行分类。作者同时使用 RGB 彩图和灰度图进行实验，结果表明 RGB 彩图取得显著成效。Naeem 等[65,69]引入了物联网领域的恶意代码检测问题，提出将恶意代码和良性二进制文件转换为彩色图像，然后分别应用机器学习和深度学习进行恶意代码检测以检测物联网中的恶意活动。实验结果表明，使用 CNN 模型的深度学习方法获得了优秀的准确率，但由于模型运行时间不足，对大规模数据集无效。蒋考林等[71]利用 AlexNet 神经网络提取由样本代码转化成的多通道图像特征并综合运用多通道图像特征提取和局部响应归一化(local response normalization，LRN)等技术进行有效分类。王润正等[78]将反汇编获取的恶意样代码的区段特征转化为 RGB 图像，在深度可分离卷积网络中引入通道域和空间域注意力机制，多维度地提取恶意图像纹理特征。实验结果表明，该方法在微软提供的 Kaggle 恶意代码数据集上获得了 98.38%的准确率。任卓君等[72]使用色谱来表示操作码指令的频次，并依据对应颜色向量在 RGB 空间中的次序来重排操作码，将可视化的结果馈入深度融合网络学习后，在 Kaggle 数据集上取得了 98.50%的分类正确率。

空军工程大学宋亚飞团队对智能化恶意代码检测方法展开了大量研究与分析[75,79-80]。其中，文献[79]提出了一种适用于恶意软件图像检测的深度学习模型，该模型结合了 DenseNet-BC 网络和注意力机制，实现了多模块结构下对恶意图像特征的有效提取。文献[75]提出了基于多尺度特征融合与通道注意力机制的轻量化卷积神经网络结构，有效融合

了不同层次的图像纹理特征。文献[80]基于灰度图像处理建立了一种多尺度卷积核混合作用的卷积神经网络模型，提出具有捷径结构的深度大内核卷积和标准小内核卷积共同作用的混合卷积核模块，有针对性地对恶意图像结构进行分析，采用深度卷积作为基本模块的网络在深度方面的延伸实现了对特征的深层挖掘。

文献[58，70，81～84]融合不同的深度学习架构进行目标特征提取。Tobiyama 等[58]应用 RNN 提取 API 调用序列特征并转换为图像，然后应用 CNN 对特征图像进行分类。Chandra 等[81]将长短期记忆网络(long short-term memory，LSTM)和身份初始化循环神经网络(identity initialized recurrent neural network，IRNN)并行独立执行得到 LSTM 和 IRNN 输出的平均值。该方法利用 2 种 RNN 结构提高图像分类性能，实验结果证明了循环神经网络(recurrent neural network，RNN)在恶意图像分类领域也大有可观。Pooja 等[82]提出将 CNN 与双向长短时记忆网络(bi-directional long short term memory，Bi-LSTM)结合用于恶意代码分类，该方法应用 CNN 从恶意图像中提取优化后的特征，并在扁平输出后应用 Bi-LSTM 进行分类。经过超参数微调和大尺寸图像输入的调整后，实验取得了令人满意的成效。Vasan 等[70]集成了高性能的预训练深度学习模型并将其成功应用于包装和未包装的恶意代码。作者利用深度预训练的深层次体系结构模型 VGG16 和 ResNet-50 对彩色图像进行特征提取，将提取出的 2 种模型的特征组合成一个 6 144 维的特征向量，然后将这些特征向量输入 SVM 以对恶意代码进行预测。陈小寒等[83]使用 RNN 处理操作码序列，并用 CNN 对特征图像进行分类，该方法利用哈希算法将原始编码与 RNN 的预测编码融合，生成特征图像，同时关联了原始恶意代码信息和时序特征。蒋瑞林等[84]将深度可分离卷积、SENet 通道注意力机制和灰度共生矩阵并联以学习恶意代码灰度图像纹理特征。

表 2　基于深度学习的恶意代码检测方法的比较

文献	年份	数据集	方法	特征	准确率/%	结论	存在问题
Nataraj[49]	2011	Malimg	数据挖掘 & K-最近邻算法	GIST 特征	97.18	纹理特征和可视化对于恶意代码检测是有效的	需手动提取特征；纹理分析的计算成本高
Pal[56]	2016	CIFAR 10	卷积神经网络	自动提取特征	68.00	证明预处理是有效的	准确率较低
Hardy[57]	2016	Comodo 云数据集	堆叠自编码器	API 调用序列	96.85	能够检测未知恶意代码	实验结果不理想
Tobiyama[58]	2016	NTT 安全数据集	RNN&CNN	API 调用序列	96.00	API 调用序列能够作为时间序列数据被输入到 RNN 中	RNN 在可变长度序列上的反向传播训练时间过长
Dong[59]	2016	KDD-96 数据集	RBM&SVM	HTTP 响应代码、请求类型等	81.00	基于 RBM 的模型能够应用于攻击检测	实验结果不理想
Kim[60]	2017	Kaggle	tGAN	自动提取特征	96.39	检测零日恶意代码所需时间更短	仅能生成样本
Kalash[55]	2018	Malimg	M-CNN	自动提取特征	98.52	获得了较高精度的分类结果	输入图像尺寸过大
Cui[61]	2018	Malimg	CNN	自动提取特征	94.50	解决了数据不平衡问题，检测时间更低	识别效果不好
Ni[62]	2018	Kaggle	CNN	自动提取特征	99.26	达到高准确率	需要反汇编恶意代码以计算特征
Kornish[63]	2018	Kaggle	CNN	自动提取特征	98.00	能够直接处理原始网络流量数据	识别精度不高；输入图像尺寸大
Singh[64]	2019	Malimg	Deep CNN	自动提取特征	98.98	采用彩色图像实现了更高的精度，不需要代码提取、执行和反编译	采用固定图像大小处理
Naeem[65]	2019	Malimg	DCNN	自动提取特征	98.18	实现了更高的检测精度和更短的检测时间	物联网设备的大小和资源有限，需要轻量级深度学习模型
Hsiao[66]	2019	Virusshare	SCNN	自动提取特征	92.00	可以使用一小组训练数据来训练模型	准确率较低；超高维图像处理

(续)表 2 基于深度学习的恶意代码检测方法的比较

文献	年份	数据集	方法	特征	准确率/%	结论	存在问题
Lu[67]	2019	Malimg	GAN	自动提取特征	84.00	成功应用 GAN	准确率较低
Jain[68]	2020	Malimg	CNN&ELM	自动提取特征	97.70	证明一维数据分析是有效的	网络体系结构较浅
Naeem[69]	2020	Leopard 移动 & windows 数据集	DCNN	自动提取特征	98.47	解决了 Android 恶意软件问题	图像维度较大;训练时间也比较长
Vasan[70]	2020	Malimg	VGG16&ResNet-50	自动提取特征	99.50	证明一组神经网络能够有效提取特征	方法的复杂性和深度预训练模型导致时间开销大
蒋考林[71]	2021	Malimg	AlexNet	自动提取特征	97.80	证明了多通道图像更有利于基于 AlexNet 的神经网络进行识别	难以应对数据集中存在不同分类粒度类别的情况
任卓君[72]	2021	Kaggle	CNN&SVM	自动提取特征	98.50	成功实现以空间填充曲线遍历 RGB 颜色空间重排样本中的操作码	不适用于大规模的图像分类
唐永旺[73]	2021	VxHeaven	Bi-LSTM&SA	自动提取特征	92.60	基于 Bi-LSTM 和自注意力机制的恶意代码检测方法是切实可行的	在大规模数据量下训练模型和检测均需要消耗大量的时间
郑钰[74]	2022	Kaggle	CNN	自动提取特征	99.92	灰度图像叠加混合序列作为多特征能够有效提高准确率	未证明模型的泛化能力
Kumar[47]	2022	Malimg	DTMIC	自动提取特征	98.92	迁移学习能够有效应用于恶意代码检测;具有良好的泛化性	准确率有待提升
王硕[75]	2023	Malimg	FFSE	自动提取特征	99.04	使用双线性插值法对恶意图像进行放缩能够提高模型准确率	归一化过程中恶意图像纹理特征变化给模型分类带来影响
Rustam[48]	2023	Malimg	Bi-KNN	自动提取特征	99.00	采用双模型迁移学习的恶意代码检测方法	检测时间较长

图神经网络能够与恶意代码检测技术相结合,以图的表征形式能够展现代码中更丰富的语法和语义特征,帮助提高恶意代码检测的效率和准确性。例如,利用图神经网络和自然语言处理算法[85,86]提取二进制程序的控制流图并进行相似性检测,可以找出恶意代码在其他软件中的位置。此外,图嵌入算法将控制流图(control flow graph,CFG)中的图表示编码为一个数值特征向量,Feng 等[87]首次将此方法应用于恶意代码检测。虽然这种方法存有局限性,如生成的码本质量受到训练数据集规模的限制,图嵌入的运行开销随码本中 CFG 的数量线性增加,但其思路具有启发性。通过将控制流和数据流转化为图的表示形式,并利用各种图神经网络发掘控制流图和数据流图中的依赖关系,能够做到精确定位脆弱代码。

2.2 基于序列特征的深度学习方法

早期的恶意代码分类检测方法需要进行人工特征提取,主要是通过可执行文件的系统调用序列特征、API 调用序列特征和操作码序列特征等进行特征提取,使用深度学习技术能够免去手动提取的物力资源消耗。Hardy 等[57]通过堆叠自编码器(autoencoder,AE)提取可执行文件的系统调用序列特征,并进行监督参数调优来检测未知的恶意代码。该方法的实验结果并不理想,但成功地将 AE 应用在恶意代码检测领域。Jain 等[68]应用极限学习机(extreme learning machine,ELM)对二进制恶意代码进行分类,并比较了基于 CNN 和 ELM 模型的实验。结果表明,用于训练 ELM 模型的时间更少,并且在一维数据处理上获得了更高的分类精度,在处理二维数据时 ELM 模型也更快。唐永旺等[73]结合了 Bi-LSTM 和自注意力机制的方法优势,采用 Bi-

LSTM 学习恶意样本的字节流序列，随后利用自注意力机制计算 Bi-LSTM 输出的时间步隐状态的线性加权和，将其作为序列的深层特征。沈元等[88]将 CNN 与改进的 LSTM 模型相结合，基于样本函数的控制流图对恶意样本进行反汇编解析，通过构建自定义函数的反汇编代码文本和整个样本的系统函数调用图对高级持续攻击（advanced persistent threat，APT）组织样本进行分类。郑钰等[74]设计实现基于 CNN 的多特征融合的多组件分类器，内容包括图像组件、序列组件和融合组件，融合了恶意代码灰度图像特征和带有 API 函数调用与操作码的混合序列特征，分类器经训练后用于检测恶意代码类别。

近年来，Powershell 由于其易用性强、隐蔽性高的特点被广泛应用于 APT 攻击中。刘岳等[89]将随机森林特征组合与深度学习相结合，使用随机森林生成原始数据的新表征，并输入神经网络进行训练用于检测 Powershell 恶意代码。高宇航等[90]提出构建双向门控循环网络（bi-directional gate recurrent unit，Bi-GRU）与注意力机制的融合网络提取 Powershell 恶意代码的上下文语义信息，利用 Powershell 恶意代码的语义特征实现恶意代码家族分类。

Android 系统作为全球最受欢迎及市场占有率最高的移动端操作系统，其安全问题受到广泛关注。传统的技术手段难以胜任现实的检测任务，有科研人员使用深度学习技术抵御 Android 恶意软件。杨宏宇等[91]提出基于双通道 CNN 的 Android 恶意软件检测模型。该方法将应用程序中提取的原始操作码序列和新生成的指令功能序列分别作为 CNN 的 2 个通道输入迭代训练并调参。受文本分类常用的 TextCNN 模型的启发，李凡等[92]对恶意 Android 应用程序进行了源码获取与信息过滤，从正负样本集中分别随机选取训练集和测试集，随后进行代码向量化操作，使用特征向量训练深度转移网络（deep transfer CNN，DTCNN）-LSTM 模型参数，最后对测试集中的恶意 Android 应用进行分类识别。实验表明，融合模型在对恶意 Android 应用的安全语义理解和局部信息抽取上表现良好。作者强调，基于深度网络结构自动获取特征表达能力的端到端检测模型依然是未来发展的方向。吴月明等[93]将恶意 Android 应用的程序语义提取为函数调用图，采用抽象 API 技术将调用图转换为抽象图，基于图特征训练构建恶意 Android 应用分类器 SriDroid。

此外，基于网络流量数据和 PDF 文档特征的深度学习方法也大有可观。针对现有的网络恶意流量检测方法依赖统计特征而忽略时序特征的问题，赵忠斌等[94]通过在特征提取算法中融合多头注意力机制来解决。该方法将网络流量以会话为单位截取固定长度的流量字节，以词嵌入的方式进行编码，输入特征提取算法获得时序特征并馈入分类器从而实现对恶意流量的检测。针对现有恶意 PDF 文档检测方法数据集样本少导致模型欠拟合的问题，俞远哲等[95]基于 Ward 最小方差聚类方法，从 PDF 文档中提取常规特征和结构特征，过滤出特征簇最小方差，将不同的聚合特征数输入 CNN 进行训练，确定出最优的聚合特征数。

2.3　基于数据增强的深度学习方法

实际中，恶意代码数据集样本严重不平衡，导致深度学习模型性能降低，因此数据增强方法应运而生。Dong 等[59]将支持向量机（support vector machine，SVM）和受限波尔兹曼机（restricted boltzmann machine，RBM）相结合，为解决数据不平衡问题，采用合成少数类过采样技术（synthetic minority over-sampling technique，SMOTE），并使用海量数据集进行实验。由于深度学习能够评估数据中相似模式的特性，该方法成功应用于检测和分类的网络分析中。Cui 等[61]将原始恶意二进制文件转换成的灰度图像输入卷积神经网络进行分类，基于蝙蝠算法设计了一种有效的数据均衡方法 DRBA，以解决 Malimg 数据集中不同家族之间存在的数据不平衡问题。实验结果表明该方法能有效避免过拟合，然而，与最近研究发现的恶意代码分类方法相比，在准确性方面不尽人意。Hsiao 等[66]选择采用单样本学习方法，作者使用了文献[49]中提出的方法可视化恶意代码，并成功应用具有双胞胎结构的孪生神经网络（siamese neural network，SNN）进行分类。

许多研究人员提出使用对抗训练以及改进的生成对抗网络（generative adversarial network，GAN）用以数据增强。GAN 由生成器和判别器组成，生成器生成与原始数据相似的样本，判别器学习模式与区分原始数据。刘延华等[96]提出对抗训练驱使的数据增强方法，将反汇编工具提取的 API 调用特征映射为二值特征向量，利用沃瑟斯坦生成对抗网络（wasserstein generative adversarial network，WGAN）构建良性样本库，为恶意代码躲避检测器提供更具迷惑性的扰动样本。Kim 等[95]提出使用转移生成对抗网络（transferred generative adversarial network，tGAN）生成新的样本来解决恶意代码数据集不平衡问题，内置 AE 模块对 GAN 进行预训练，克服了原始 GAN 的局限性。Lu 等[67]实现了将深度卷积生成对抗网络（deep convolutional

generative adversarial networks，DCGAN）用于恶意代码分类，并解决了数据不平衡问题。DCGAN 将深度 CNN 与 GAN 结合用于无监督学习，相比较于单独使用 GAN，实验结果增加了 6% 的分类精度。朱晓慧等[97]将图像处理技术与 WGAN-gp 相结合，通过缩放处理图像高概率地保留原始隐含特征，使用 WGAN-gp 训练经缩放处理后的恶意图像以学习样本分布规律，进而有效地生成充足均衡的新数据。王栋等[98]基于一维卷积神经网络（one dimensional CNN，1D-CNN）和半监督生成对抗网络（semi-supervised GAN，SGAN）构造半监督深度卷积生成对抗网络 SGAN-CNN 用于灰度恶意图像分类。

3 总结与展望

恶意代码检测领域面临着许多关键挑战，这些挑战主要源自恶意代码的不断演变和高级化的形式，包括恶意代码的隐藏深度、变种多样性、零日漏洞的利用等。近年来，随着恶意代码检测技术的不断发展和研究的不断深入，涌现出许多关于恶意代码特征提取与分类识别的新技术和新方法，并向智能化、体系化逐步发展。今后，恶意代码检测仍将是网络空间安全领域的重要研究热点之一，还有许多亟待研究解决的问题。通过对现有相关工作的分析总结，基于深度学习的恶意代码检测未来在以下方面需要进一步研究突破。

3.1 模型的可解释性

深度学习模型，特别是 CNN 和 RNN 此类常用的特征提取网络，虽然能够进行高效的学习和推断，但其决策过程往往缺乏透明度，研究人员需要理解模型为何将某个文件判别为恶意或良性，以便对其决策的正确性进行评估，这在恶意代码检测等安全性至关重要的领域中可能构成挑战。

未来的研究可以致力于开发更具解释性的深度学习模型，其中最具代表性的包括神经图模型和可解释的深度学习模型。神经图模型是一种能够处理图结构数据的深度学习模型，而可解释的深度学习模型则是一种能够解释模型决策和输出的深度学习模型，可以更好地理解和解释模型决策和输出的结果。另外，知识图谱模型也可以用于恶意代码检测，通过将恶意代码和良性代码的特征和行为表示为知识图谱中的节点和边，从而更准确地识别和解释恶意代码的行为和特征。这些更具解释性的深度学习模型将有助于提高恶意代码检测的精度和理解其决策过程。

3.2 模型的泛化能力

深度学习在恶意图像处理、特征提取、分类识别等领域得到广泛应用，并取得了显著的成果，然而，许多文献中提出的解决方法仅针对于特定的恶意代码数据集，对数据集的数量、质量都有较高的要求，缺乏泛化性。现有的基于深度学习的恶意代码特征提取和分类识别，尽管可以高精度地完成目标分类识别，然而，在特定数据集上训练的模型可能在识别新出现的、未曾遇见过的恶意代码时表现不佳。此现象的产生原因可能在于这些新型恶意代码并未被包含在训练数据集中，或是其行为模式有别于训练集中的已知恶意代码。因此，基于深度学习理论框架，如何在小样本的条件下完成恶意代码的高精度实时分类识别，并对未知目标具有一定的泛化性、迁移性，同时网络具有一定的可解释性，仍是制约恶意代码分类识别的技术障碍，亟待研究突破。

下一步工作可致力于提高模型的泛化性能，包括采用数据增强、正则化、迁移学习和对抗训练等方法技术。数据增强可以增加训练数据集的多样性，通过随机修改良性代码或恶意代码的一部分来生成新的样本，从而使得模型在训练过程中接触到更多的情况和变化。使用正则化技术可以限制模型的复杂度和过拟合，从而降低模型对训练数据的过拟合程度，提高其对未知数据的泛化性能。通过迁移学习技术能够将已经训练好的模型应用于新的数据集上。采用对抗训练方法可以增加模型的鲁棒性和泛化性能，通过增加对抗样本等手段来提高模型对未知数据的适应性。这些方法的应用可以提高模型的泛化性能，使其能够更好地适应并识别新出现的恶意代码。

3.3 恶意代码数据集的质量和规模

现实生活中，对网络空间产生威胁的恶意代码形态是复杂多样的，其数据集样本因规格和格式而不同。当前的恶意代码数据集存在种种问题，如恶意代码形态多样化、标准数据集较少、数据分布不均衡及数据缺乏标签等。这可能导致模型在训练和测试时面临困难，尤其是在未见过的恶意代码形态或缺乏标签的数据上。与此同时，用于训练不同深度学习模型的数据集也难以反映真实世界的恶意代码。在训练深度学习模型时，其性能受到数据集规模的影响，因此数据集的不平衡问题也是恶意代码分类中的另一大挑战。研究表明，对原始数据的预处理能够提高性能，但也消耗时间。尤其是在恶意代码可视化研究中，许多研究人员为提高分类精度而使用大尺寸图像，这显著增加了训练时间。以上都是在恶意代码数据集上需要进一步研究解决的

问题。

未来可以进一步研究改进数据收集和处理方法，提高数据的质量和规模，或者开发更强大的模型，能够更好地处理这些问题。在数据收集层面，应致力于实现数据集的多元化，以覆盖更广泛的场景和情况，这能够帮助模型更好地理解和适应各种不同的环境和情况。在数据清洗和预处理阶段，对于原始数据的处理，应致力于提高数据的质量和可靠性，包括去除数据中的噪音、填充缺失值和解决异常值等问题，以提供更准确、更完整的数据。在数据增强和处理环节，可以利用各种技术手段增加数据的多样性和规模。同时，对数据的时序关系进行合理处理、对特征进行优化选择，也可以提升数据的可用性和模型的性能。通过以上策略可以有效地提升数据的质量与规模，同时开发出更加强大的模型来更好地解决各类问题。

参考文献

[1] SU J W, VASCONCELLOS D V, PRASAD S, et al. Lightweight Classification of IoT Malware Based on Image Recognition[C]//HIRONORI K. 2018 IEEE 42nd Annual Computer Software and Applications Conference (COMPSAC). Tokyo: IEEE, 2018: 664-669.

[2] MOHURLE S, PATIL M. A Brief Study of Wannacry Threat: Ransomware Attack 2017[J]. International Journal of Advanced Research in Computer Science, 2017, 8(5): 1938-1940.

[3] 国家互联网应急中心. 2021 年上半年我国互联网网络安全监测数据分析报告[EB/OL]. (2021-07-31)[2023-05-11]. https://www.cert.org.cn/publish/main/46/2021/20210731090556980286517/20210731090556980286517_.html.

[4] 马博林，张铮，陈源，等. 基于指令集随机化的抗代码注入攻击方法[J]. 信息安全学报，2020，5(4)：30-43.

[5] 邵思豪，高庆，马森，等. 缓冲区溢出漏洞分析技术研究进展[J]. 软件学报，2018，29(5)：1179-1198.

[6] 刘强，邓亚平，徐震，等. 隐藏木马检测技术的研究[J]. 计算机工程，2006，32(1)：180-182.

[7] 冯晓萌，孙秋野，王冰玉，等. 基于蠕虫传播和 FDI 的电力信息物理协同攻击策略[J]. 自动化学报，2022，48(10)：2429-2441.

[8] YADAV B, TOKEKAR S. Recent Innovations and Comparison of Deep Learning Techniques in Malware Classification: A Review[J]. International Journal of Information Security Science, 2021, 9(4): 230-247.

[9] 申培，刘福龙，桑海伟. 恶意代码检测研究综述[J]. 重庆理工大学学报(自然科学)，2022，36(11)：212-218.

[10] 王志文，刘广起，韩晓晖，等. 基于机器学习的恶意软件识别研究综述[J]. 小型微型计算机系统，2022，43(12)：2628-2637.

[11] 郭沁怡. 恶意代码检测技术研究综述[J]. 电脑知识与技术，2023，19(13)：79-81，93.

[12] 李豪，钱丽萍. 恶意代码可视化检测技术研究综述[J]. 软件导刊，2022，21(5)：9-16.

[13] 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.

[14] GRIFFIN K, SCHNEIDER S, HU X, et al. Automatic Generation of String Signatures for Malware Detection[C]//12th International Symposium on Recent Advances in Instrusion Detection (RAID2009), Saint-Malo: Springer, 2009: 101-120.

[15] YAMAGUCHI F, GOLDE N, ARP D, et al. Modeling and Discovering Vulnerabilities with Code Property Graphs[C]//2014 IEEE Symposium on Security and Privacy. Berkeley, CA: IEEE, 2014: 590-604.

[16] 张玮康. 基于恶意代码 API 的静态检测技术研究[D]. 西安：西安电子科技大学，2018.

[17] COULL S E, GARDNER C. Activation Analysis of a Byte-Based Deep Neural Network for Malware Classification[C]//2019 IEEE Security and Privacy Workshops (SPW). San Francisco, CA: IEEE, 2019: 21-27.

[18] 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.

[19] ALASMARY H, KHORMALI A, ANWAR A, et al. Analyzing and Detecting Emerging Internet of Things Malware: A Graph-Based Approach[J]. IEEE Internet of Things Journal, 2019, 6(5): 8977-8988.

[20] TAHERI R, GHAHRAMANI M, JAVIDAN R, et al. Similarity-Based Android Malware Detection Using Hamming Distance of Static Binary Features[J]. Future Generation Computer Systems, 2020, 105: 230-247.

[21] REZAEI T, HAMZE A. An Efficient Approach for Malware Detection Using PE Header Specifications[C]//2020 6th International Conference on Web Research (ICWR). Tenran: IEEE, 2020: 234-239.

[22] 赵晓君，王小英，张咏梅，等. 基于恶意代码行为分析的入侵检测技术研究[J]. 计算机仿真，2015，32(4)：277-280.

[23] SHAID S Z, MAAROF M A. Malware Behaviour Visualization[J]. Jurnal Teknologi, 2014, 70(5): 325-330.

[24] GALAL H S, MAHDY Y B, ALLATIEA M. Behavior-Based Features Model for Malware Detection[J]. Journal of Computer Virology and Hacking Techniques, 2016, 12(2): 59-67.

[25] MOHAISEN A, ALRAWI O, PARK J, et al. Network-Based Analysis and Classification of Malware Using Behavioral Artifacts Ordering[EB/OL]. (2018-12-08)[2023-05-20]. http://arxiv.org/abs/1901.01185.

[26] AMER E, ZELINKA I. A Dynamic Windows Malware Detection and Prediction Method Based on Contextual Understanding of API Call Sequence[J]. Computers & Security, 2020, 92: 101760.

[27] ANDERSON B, MCGREW D. Accurate TLS Fingerprinting Using Destination Context and Knowledge Bases[EB/OL]. (2020-09-03)[2023-04-25]. http://arxir.org/abs/2009.01939.

[28] 李鹏超,刘彦飞. 基于删除 PE 文件头的恶意代码内存取证方法[J]. 信息网络安全, 2021, 21(12): 38-43.

[29] TIEN C W, LIAO J W, CHANG S C,et al. Memory Forensics Using Virtual Machine Introspection for Malware Analysis[C]//2017 IEEE Conference on Dependable and Secure Computing. Taipei:IEEE, 2017: 518-519.

[30] KUMARA M A A, JAIDHAR C D. Leveraging Virtual Machine Introspection with Memory Forensics Todetect and Characterize Unknown Malware Using Machine Learning Techniques at Hypervisor[J]. Digital Investigation, 2017, 23: 99-123.

[31] BRIDGES R, JIMÉNEZ J H, NICHOLS J, et al. Towards Malware Detection via cpu Power Consumption: Data Collection Design and Analytics[C]//2018 17th IEEE International Conference on Trust, Security And Privacy in Computing and Communications/12th IEEE International Conference on Big Data Science and Engineering (TrustCom/BigDataSE). New York:IEEE, 2018: 1680-1684.

[32] SAYADI H, PATEL N, SASAN A, et al. Ensemble Learning for Effective Run-Time Hardware-Based Malware Detection: A Comprehensive Analysis and Classification[C]//Proceedings of the 55th Annual Design Automation Conference. San Francisco, CN: IEEE,2018: 1-6.

[33] 潘绯. 基于物理特征的认证及恶意节点检测研究[D]. 成都: 电子科技大学, 2019.

[34] 戴超,庞建民,张一弛,等. 基于语义特征的恶意代码检测综述[J]. 信息工程大学学报, 2018, 19(1): 106-113.

[35] 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.

[36] 梁光辉,摆亮,庞建民,等. 一种基于混合学习的恶意代码检测方法[J]. 电子学报, 2021, 49(2): 286-291.

[37] MILLER B P, FREDRIKSEN L, SO B. An Empirical Study of the Reliability of UNIX Utilities[J]. Communications of the ACM, 1990, 33(12): 32-44.

[38] RAWAT S, JAIN V, KUMAR A, et al. VUzzer: Application-aware Evolutionary Fuzzing[C]//Proceedings of 2017 Network and Distributed System Security Symposium. San Diego, CA: Internet Society, 2017: 1-14.

[39] LIU Y, YANG P, JIA P, et al. MalFuzz: Coverage-guided Fuzzing on Deep Learning-Based Malware Classification Model[J]. PLOS one, 2022, 17(9): e0273804.

[40] NAIK N, JENKINS P, SAVAGE N, et al. Fuzzy-Import Hashing: A Malware Analysis Approach[C]//2020 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE). Glasgow, UK: IEEE, 2020: 1-8.

[41] KIM S, LEE S. Automatic Malware Detection Rule Generation and Verification System[J]. Journal of Internet Computing and Services, 2019, 20(2): 9-19.

[42] HOLZER A, KINDER J, VEITH H. Using Verification Technology to Specify and Detect Malware[C]//International Conference on Computer Aided Systems Theory. Berlin, Heidelberg: Springer, 2007: 497-504.

[43] XIANG Q, WANG X D, SONG Y F, et al. One-Dimensional Convolutional Neural Networks for High-Resolution Range Profile Recognition via Adaptively Feature Recalibrating and Automatically Channel Pruning[J]. International Journal of Intelligent Systems, 2021, 36(1): 332-361.

[44] XIANG Q, WANG X D, LAI J, et al. Multi-Scale Group-Fusion Convolutional Neural Network for High-Resolution Range Profile Target Recognition[J]. IET Radar, Sonar & Navigation, 2022, 16(12): 1997-2016.

[45] GOPINATH M, SETHURAMAN S C. A Comprehensive Survey on Deep Learning Based Malware Detection Techniques[J]. Computer Science Review, 2023, 47: 100529.

[46] MACAS M, WU C M, FUERTES W. A Survey on Deep Learning for Cybersecurity: Progress, Challenges, and Opportunities[J]. Computer Networks, 2022, 212: 109032.

[47] KUMAR S, JANET B. DTMIC: Deep Transfer Learning for Malware Image Classification[J]. Journal of Information Security and Applications, 2022, 64: 103063.

[48] RUSTAM F, ASHRAF I, JURCUT A D, et al. Malware Detection Using Image Representation of Malware Data and Transfer Learning[J]. Journal of Parallel and Distributed Computing, 2023, 172:

32-50.

[49] 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. New York:ACM, 2011: 1-7.

[50] HAN K S, LIM J H, IM E G. Malware Analysis Method Using Visualization of Binary Files[C]//Proceedings of the 2013 Research in Adaptive and Convergent Systems. New York:ACM,2013: 317-321.

[51] 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.

[52] VU D L, NGUYEN T K, NGUYEN T V, et al. HIT4Mal: Hybrid Image Transformation for Malware Classification[J]. Transactions on Emerging Telecommunications Technologies, 2020, 31(11): e3789.

[53] HASAN M, ISLAM M M, ZARIF M I I, et al. Attack and Anomaly Detection in IoT Sensors in IoT Sites Using Machine Learning Approaches[J]. Internet of Things, 2019, 7: 100059.

[54] GREENGARD S. Cybersecurity Gets Smart[J]. Communications of the ACM, 2016, 59(5): 29-31.

[55] 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). Paris:IEEE, 2018: 1-5.

[56] PAL K K, SUDEEP K S. Preprocessing for Image Classification by Convolutional Neural Networks [C]//2016 IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT). Bangalore:IEEE,2016: 1778-1781.

[57] 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). Las Vegas, Nevada:WorldComp, 2016: 61.

[58] TOBIYAMA S, YAMAGUCHI Y, SHIMADA H, et al. Malware Detection with Deep Neural Network Using Process Behavior[C]//2016 IEEE 40th Annual Computer Software and Applications Conference (COMPSAC). Atlanta,GA: IEEE, 2016: 577-582.

[59] DONG B, WANG X. Comparison Deep Learning Method to Traditional Methods Using for Network Intrusion Detection[C]//2016 8th IEEE International Conference on Communication Software and Networks (ICCSN). Beijing: IEEE, 2016: 581-585.

[60] KIM J Y, BU S J, CHO S B. Malware Detection Using Deep Transferred Generative Adversarial Networks[C]//Neural Information Processing: 24th International Conference, ICONIP 2017. Guangzhou: Springer International Publishing, 2017: 556-564.

[61] 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.

[62] NI S, QIAN Q, ZHANG R. Malware Identification Using Visualization Images and Deep Learning [J]. Computers & Security, 2018, 77: 871-885.

[63] KORNISH D, GEARY J, SANSING V, et al. Malware Classification Using Deep Convolutional Neural Networks[C]//2018 IEEE Applied Imagery Pattern Recognition Workshop (AIPR). Washington, DC: IEEE, 2018: 1-6.

[64] SINGH A, HANDA A, KUMAR N, et al. Malware Classification Using Image Representation[C]//Cyber Security Cryptography and Machine Learning: Third International Symposium, CSCML 2019. Beer-Sheva: Springer International Publishing, 2019: 75-92.

[65] NAEEM H. Detection of Malicious Activities in Internet of Things Environment Based on Binary Visualization and Machine Intelligence[J]. Wireless Personal Communications, 2019, 108(4): 2609-2629.

[66] HSIAO S C, KAO D Y, LIU Z Y, et al. Malware Image Classification Using One-Shot Learning with Siamese Networks [J]. Procedia Computer Science, 2019, 159: 1863-1871.

[67] LU Y, LI J. Generative Adversarial Network for Improving Deep Learning Based Malware Classification [C]//2019 Winter Simulation Conference (WSC). National Harbor,MD:IEEE, 2019: 584-593.

[68] JAIN M, ANDREOPOULOS W, STAMP M. Convolutional Neural Networks and Extreme Learning Machines for Malware Classification[J]. Journal of Computer Virology and Hacking Techniques, 2020, 16 (3): 229-244.

[69] 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[J]. Ad Hoc Networks, 2020, 105: 102154.

[70] VASAN D, ALAZAB M, WASSAN S, et al. Image-Based Malware Classification Using Ensemble of CNN Architectures (IMCEC)[J]. Computers & Security, 2020, 92: 101748.

[71] 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.

[72] 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.

[73] 唐永旺,刘欣.基于 Bi-LSTM 和自注意力的恶意代码检测方法[J].计算机应用与软件,2021,38(3):327-333.

[74] 郑珏,欧毓毅.基于卷积神经网络与多特征融合恶意代码分类方法[J].计算机应用研究,2022,39(1):240-244.

[75] 王硕,王坚,王亚男,等.一种基于特征融合的恶意代码快速检测方法[J].电子学报,2023,51(1):57-66.

[76] 申高宁,陈志翔,王辉,等.基于挤压激励网络的恶意代码家族检测方法[J].信息技术与网络安全,2022,41(6):1-9.

[77] 褚堃,万良,马丹,等.深度可分离卷积在 Android 恶意软件分类的应用研究[J].计算机应用研究,2022,39(5):1534-1540.

[78] 王润正,高见,仝鑫,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.

[79] 李一萌,李成海,宋亚飞,等.基于 Attention-DenseNet-BC 的恶意软件家族分类方法[J].计算机科学,2021,48(10):308-314.

[80] 张丹丹,宋亚飞,刘曙.MalMKNet:一种用于恶意代码分类的多尺度卷积神经网络[J].电子学报,2023,51(5):1359-1369.

[81] CHANDRA B, SHARMA R K. On Improving Recurrent Neural Network for Image Classification[C]// 2017 International Joint Conference on Neural Networks (IJCNN). Anchorage,AK:IEEE, 2017: 1904-1907.

[82] POOJA B, KUMAR G S. Detection of Malware Using Deep Learning Techniques [J]. International Journal of Scientific and Technology Research, 2020, 9: 1688-1691.

[83] 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.

[84] 蒋瑞林, 覃仁超.基于深度可分离卷积的多神经网络恶意代码检测模型[J]. 计算机应用, 2023, 43(5):1527-1533.

[85] YU Z D,CAO R,TANG Q Y,et al. Order Matters: Semantic Aware Neural Networks for Binary Code Similarity Detection [J]. Proceedings of the AAAI Conference on Artificial Intelligence, 2020, 34(1): 1145-1152.

[86] LI Z,ZOU D,XU S,et al. VulDeePecker:A Deep Learning-Based System for Vulnerability Detection[C]//Network and Distirbuted Systems Security(NDSS) Symposium 2018. Sam Diego,CA:[s. n.],2018:1-95.

[87] FENG Q,ZHOU R,XU C,et al. Scalable Graph-Based Bug Search for Firmware Images[C]//Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security. Vienna:ACM,2016:480-491.

[88] 沈元,严寒冰,夏春和,等.一种基于深度学习的恶意代码克隆检测技术[J]. 北京航空航天大学学报, 2022, 48(2): 282-290.

[89] 刘岳,刘宝旭,赵子豪,等.基于特征组合的 Powershell 恶意代码检测方法[J].信息安全学报,2021,6(1):40-53.

[90] 高宇航,彭国军,杨秀璋,等.基于深度学习的 PowerShell 恶意代码家族分类研究[J].武汉大学学报(理学版),2022,68(1):8-16.

[91] 杨宏宇,那玉琢.一种 Android 恶意软件检测模型[J].西安电子科技大学学报,2019,46(3):45-51.

[92] 李凡,易军凯.代码向量深度学习的恶意 Android 应用检测方法[J].计算机应用研究,2021,38(2):549-552,558.

[93] 吴月明,齐蒙,邹德清,等.图卷积网络的抗混淆安卓恶意软件检测[J].软件学报,2023,34(6):2526-2542.

[94] 赵忠斌,蔡满春,芦天亮.融合多头注意力机制的网络恶意流量检测[J].数据与计算发展前沿,2022,4(5):60-67.

[95] 俞远哲,王金双,邹霞.基于特征集聚和卷积神经网络的恶意 PDF 文档检测方法[J].信息技术与网络安全,2021,40(8):35-41.

[96] 刘延华,李嘉琪,欧振贵,等.对抗训练驱动的恶意代码检测增强方法[J].通信学报,2022,43(9):169-180.

[97] 朱晓慧,钱丽萍,傅伟.基于生成对抗网络增强恶意代码的方法[J].计算机工程与设计,2021,42(11):3034-3042.

[98] 王栋,杨珂,玄佳兴,等.基于半监督生成对抗网络的恶意代码家族分类实现[J].计算机工程与科学,2022,44(5):826-833.

(编辑:刘勇)