计算机研究与发展
Journal of Computer Research and Development
DOI: 10.7544/issn1000-1239.202330450
60(10): 2234−2254, 2023

基于人工智能的物联网恶意代码检测综述

刘奇旭[1,2] 刘嘉熹[1,2] 靳 泽[1,2] 刘心宇[1,2] 肖聚鑫[1,2] 陈艳辉[1,2] 朱洪文[1,2] 谭耀康[1,2]
[1](中国科学院信息工程研究所 北京 100085)
[2](中国科学院大学网络空间安全学院 北京 100049)
(liuqixu@iie.ac.cn)

Survey of Artificial Intelligence Based IoT Malware Detection

Liu Qixu[1,2], Liu Jiaxi[1,2], Jin Ze[1,2], Liu Xinyu[1,2], Xiao Juxin[1,2], Chen Yanhui[1,2], Zhu Hongwen[1,2], and Tan Yaokang[1,2]
[1](*Institute of Information Engineering*, *Chinese Academy of Sciences*, *Beijing* 100085)
[2](*School of Cyber Security*, *University of Chinese Academy of Sciences*, *Beijing* 100049)

Abstract In recent years, with the large-scale deployment of Internet of things (IoT) devices, there has been a growing emergence of malicious code targeting IoT devices. IoT security is facing significant threats from malicious code, necessitating comprehensive research on IoT malware detection techniques. Following the remarkable achievements of artificial intelligence (AI) in fields such as computer vision (CV) and natural language processing (NLP), the IoT security field has witnessed numerous efforts in AI-based malware detection as well. By reviewing relevant research findings and considering the characteristics of IoT environments and devices, we propose a classification method for the primary motivations behind research in this field and analyze the research development in IoT malware detection from two perspectives: malware detection techniques towards IoT device limitation mitigation and IoT malware detection techniques towards performance improvement. This classification method encompasses the relevant research in IoT malware detection, which also highlights the unique characteristics of IoT devices and the current limitations of the IoT malware detection field. Finally, by summarizing existing research, we extensively discuss the challenges present in AI-based malware detection and present three possible directions for future research that consists of combining foundation models in IoT malware code detection, improving the safety of detection models, and combining zero trust architecture in this field.

Key words Internet of things (IoT); malware; artificial intelligence (AI); detection technology; cyber security

摘 要 近年来,随着物联网(Internet of things, IoT)设备的大规模部署,针对物联网设备的恶意代码也不断出现,物联网安全面临来自恶意代码的巨大威胁,亟需对物联网恶意代码检测技术进行综合研究.随着人工智能(artificial intelligence, AI)在计算机视觉和自然语言处理等领域取得了举世瞩目的成就,物联网安全领域也出现了许多基于人工智能的恶意代码检测工作.通过跟进相关研究成果,从物联网环境和设备的特性出发,提出了基于该领域研究主要动机的分类方法,从面向物联网设备限制缓解的恶意代码检测和面向性能提升的物联网恶意代码检测2方面分析该领域的研究发展现状.该分类方法涵盖了物联

收稿日期: 2023-06-05; **修回日期**: 2023-08-14
基金项目: 中国科学院青年创新促进会(2019163); 中国科学院战略性先导科技专项项目(XDC02040100); 中国科学院网络测评技术重点实验室项目; 网络安全防护技术北京市重点实验室项目
This work was supported by the Youth Innovation Promotion Association CAS (2019163), the Strategic Priority Research Program of Chinese Academy of Sciences (XDC02040100), the Project of CAS Key Laboratory of Network Assessment Technology, and the Project of Beijing Key Laboratory of Network Security and Protection Technology.
通信作者: 靳泽(jinze@iie.ac.cn)

网恶意代码检测的相关研究，充分体现了物联网设备独有的特性以及当前该领域研究存在的不足. 最后通过总结现有研究，深入讨论了目前基于人工智能的恶意代码检测研究中存在的问题，为该领域未来的研究提出了结合大模型实现物联网恶意代码检测，提高检测模型安全性以及结合零信任架构 3 个可能的发展方向.

关键词 物联网；恶意代码；人工智能；检测技术；网络空间安全

中图法分类号 TP309.5；TP391

近年来，物联网（Internet of things, IoT）在智能家居、智能手表、智能健康、供应链管理等领域中被大量使用，“万物互联”已经成为当今时代的主流. 截至 2022 年，全球物联网设备的数量已达 131 亿，预计到 2030 年物联网设备数量将会逼近 300 亿，与此同时，物联网市场的收益也逐步攀升，年收入增长率高达 13.60%[1].

随着物联网设备的大量使用，人们的生活质量显著提高，然而，物联网的蓬勃发展也为攻击者提供了温床，生产厂商更多关注设备销量和收益，对物联网设备的安全问题很难投入足够的研究. 此外，物联网设备还存在使用弱口令、不及时更新安全补丁等问题. 因此，物联网设备比传统的台式机、笔记本电脑等设备更容易被恶意代码攻击，成为恶意攻击的目标. 恶意代码，又称为恶意软件，是指能够在计算机系统中进行非授权操作，并使系统执行攻击者希望其执行的操作，以实施破坏或窃取信息的代码. 恶意代码可能以蠕虫、病毒、远控木马、僵尸程序以及勒索软件等形态出现，以不同攻击形态出现的恶意代码破坏计算机、服务器、客户端或计算机网络，或在不知情的情况下损害用户的计算机安全和隐私，给企业和个人造成巨大的经济损失. 例如，攻击者通过精心设计恶意代码，利用物联网设备使用默认凭证或弱凭证的漏洞控制设备，并执行进一步的攻击.

Sonic Wall 的报告[2] 指出，截止到 2022 年，针对物联网设备的恶意代码攻击同比增加了 77%，无疑给物联网安全造成巨大威胁. 2016 年，物联网恶意软件 Mirai 通过使用默认用户名和密码感染设备，创下了最大的分布式拒绝服务（distributed denial of service, DDoS）攻击记录，引起了众多安全研究人员[3-4] 对物联网恶意代码的关注. Mirai 的源代码不久后泄露，一定程度上导致了以物联网设备为目标的新型恶意软件家族如 Gafgyt, Reaper, satori[5] 等的出现，这些恶意代码对物联网设备的安全和用户的隐私及财产安全都产生了严重威胁，物联网恶意代码检测技术已经成为物联网安全领域研究的重要组成部分.

为了缓解恶意代码带来的巨大安全风险，近年来，安全研究人员开始逐渐关注物联网领域的恶意软件检测工作. 传统的物联网恶意代码检测和设备保护方法主要依靠特征库的积累和恶意软件分析人员的人工分析，但是由于恶意软件的爆发式增长[4,6–10]，传统方式缺乏效率且难以应对未知的安全风险. 安全研究人员开始尝试将在图像分类、文本分析等领域取得了巨大成功的人工智能（artificial intelligence, AI）技术应用于恶意代码检测领域并得到了很好的效果[11–14]. 目前，基于人工智能技术的物联网恶意代码检测研究逐渐成为主流. 但是，相较于传统的台式机、服务器等设备，物联网设备上的恶意代码检测技术不仅面临着基于人工智能的恶意代码检测技术普遍需求更高检测准确率等问题，而且要应对物联网设备自身特性引发的 2 大挑战[15]：1）物联网恶意代码能够感染使用多种不同 CPU 架构的设备，而不同 CPU 架构的指令集不同，导致无法将基本使用相同架构的传统设备中成熟的恶意代码特征提取和检测方法直接应用到物联网恶意代码的检测中. 2）由于物联网设备一般体积较小，需要部署在各种不同的环境中，物联网设备受到内存空间小、电量少等资源限制，需要部署相对于传统设备更轻量级的检测系统. 因此基于人工智能的物联网恶意代码检测研究在提高模型的检测效果的同时，需要解决当前物联网设备自身的特性带来的独特挑战.

随着物联网设备地逐步普及，许多物联网安全相关研究也被提出[16]，基于人工智能的物联网恶意代码检测相关研究也不断涌现，本文对 2018 年以来网络与信息安全领域四大顶级会议以及期刊等来源的基于人工智能的物联网恶意代码检测相关研究进行了大量的调研和分析，总结现有研究工作的特点和不足，为进一步的研究提供了系统性的参考. 图 1 展示了历年来基于人工智能的物联网恶意代码检测文章数量，文章数量的增长速度逐步加快，说明随着人工智能和物联网技术的快速发展，越来越多的检测方案被提出.

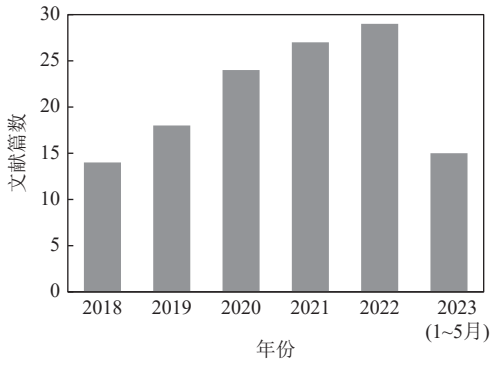


Fig. 1 AI-based IoT malware detection representative research statistics from 2018 to May 2023

图 1 2018—2023 年 5 月基于人工智能的物联网恶意代码检测代表性研究统计

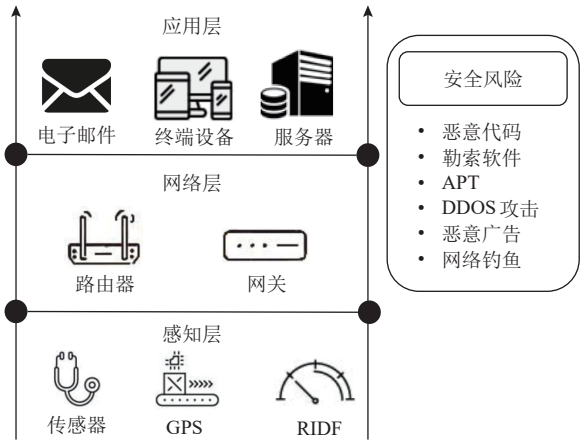


Fig. 2 Structural layers of IoT system and the threats they face

图 2 物联网系统的结构层次及其面临的安全威胁

为了更加直观地展现目前研究工作的侧重点以及存在的问题，本文总结了物联网领域与传统设备领域的检测工作的差异性，从新的角度提出一种新的分类方法.

本文主要有 3 个方面的贡献：

1）本文调研了自 2018 年以来基于人工智能技术的物联网恶意代码检测工作，深入分析了这些研究工作提出的基于人工智能的检测技术以及其技术特点，对本领域的发展进程进行了全面的梳理.

2）本文从物联网设备和系统自身的特性出发，围绕物联网恶意代码检测的主要研究动机，从面向物联网设备限制缓解的恶意代码检测和面向性能提升的物联网恶意代码检测 2 个角度对当前的研究工作进行了分类研究.

3）基于对物联网恶意代码检测工作的全面调研总结，本文对当前的工作进行了深入的分析，总结了应用人工智能技术的检测当前仍存在的不足和面临的挑战，并展望了未来基于人工智能的物联网恶意代码检测研究的方向.

1 研究背景

1.1 物联网相关概念及特性

本节从物联网的基本概念入手，介绍物联网的基本架构和物联网设备的内部架构，进而梳理了影响运行在物联网设备上的恶意代码检测技术设计的特性.

学术界通常将物联网系统划分为感知层、网络层和应用层[17-18]，物联网系统的结构层次和安全风险及威胁如图 2 所示.

感知层关联到各类物联网设备，这些设备中通常内置了许多用于收集外部信息的传感器以及用于数据传输的无线连接模块[19]，传感器收集的数据被发送到应用层. 网络层定义了各类通信协议与传输协议，负责感知层与应用层之间的数据交换. 应用层包括云平台和搭载于物联网设备上的应用程序(application, APP). 其中各层的物联网设备都面临着不同的安全威胁，而各个层次中的物联网设备都可能受到恶意代码攻击.

具体到设备而言，每个物联网设备的内部组成自下而上可以分为硬件层、系统层和用户层，如图 3 所示.

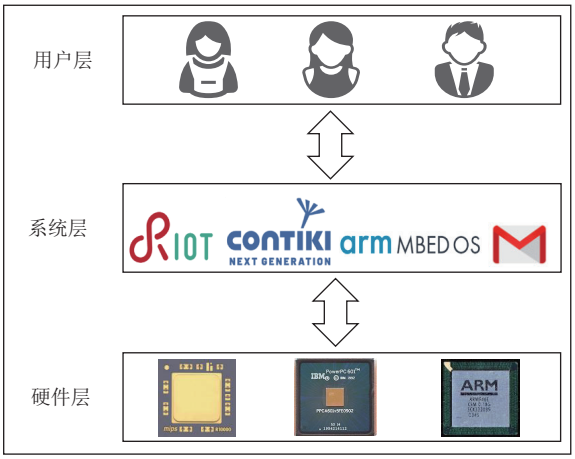


Fig. 3 Internal architecture of IoT devices

图 3 物联网设备内部架构

硬件层常见的处理器架构包括 ARM、MIPS、PowerPC(PPC)、SPARC、SuperH 等[20]，供应商往往会根据物联网设备不同的功能需求选择基于不同架构的处理器. 这些架构均为 32 b 精简指令集处理器计算机(reduced instruction set computer, RISC)，其被广泛应用于微处理器的嵌入式系统设计，低能耗的特

性使得这些架构适用于移动通信、航空航天、智能传感器等设备中. 例如, 基于 MIPS 架构开发的 Sensor Hub 被广泛应用于可穿戴设备[21]. 台式机和服务器等设备上常用的 x86 和 x86_64 等架构为复杂指令集处理器计算机(complex instruction set computer, CISC), 由于其复杂性与高能耗, 较少出现在物联网设备上.

物联网设备的系统层包括操作系统和应用程序, 为物联网设备功能的实现提供支撑. 物联网设备的操作系统具有低功耗、安全、通信协议支持和云端连接功能. 常见的物联网操作系统包括 RIOT、Contiki、ARM mbed、eLinux 等[22].

物联网设备的用户层面向用户提供交互界面并接受用户控制.

通过分析物联网设备的组成, 可以梳理出物联网设备区别于传统台式机等设备的 2 个特性:

1) 根据物联网设备的独特功能需求使用不同的 CPU 架构, 不同 CPU 架构所使用的指令集、寄存器等也存在差异.

2) 物联网设备结构上的局限性导致大多数物联网设备的内存容量较小, 可分配的计算资源也相对较少.

由于存在多种 CPU 架构以及资源限制等与传统台式机和服务器等设备不同的特性, 攻击者往往会专门开发针对物联网设备的恶意代码. 1.2 节中, 我们将详细探讨这些特性给物联网恶意代码检测带来的独特挑战.

1.2 物联网恶意代码检测挑战

本节首先介绍了近年来活跃的物联网恶意代码, 然后结合物联网设备和运行在其上的恶意代码的特性指出了基于人工智能的物联网恶意代码检测技术面临的独特挑战.

著名的物联网恶意代码 Mirai 在 2016 年 10 月的网络攻击[23]引起了全球关注, 大量运行嵌入式 Linux 系统且使用弱密码或默认凭证的 IoT 设备被 Mirai 感染, 并被组织成庞大的僵尸网络用于发动大规模的 DDoS 攻击, 攻击导致大半个美国互联网瘫痪. 近年来, IoT 平台的主要威胁依然是以 Mirai、Gafgyt 等为代表的主流僵尸网络家族, 也有陆续出现一些变体和新家族如 Tsunami、Mozi、VPNfilter 等. 僵尸网络家族也不再满足于挖矿和 DDOS 攻击, 2022 年发表的研究工作[24]提出一种被称为通过物联网操纵需求(manipulation of demand via IoT, MadIoT)的新型潜在攻击, 此攻击表明物联网恶意代码已对国家的基础设施构成了重大威胁.

Mirai、Tsunami、VPNfilter 等物联网恶意代码已被发现在多种 CPU 架构上运行[25-27]. 鉴于第 1.1 节中提到的不同物联网设备通常会使用大量不同的 CPU 架构的特性, 攻击者可以通过在不同架构上编译恶意代码, 然后向使用不同架构的物联网设备广泛投放, 以实现大规模的攻击. 这为物联网恶意代码检测带来了第 1 个独特的挑战.

在不同 CPU 架构上编译的程序具有不同的指令集, 在台式机等设备上基于人工智能的恶意代码检测中, 通常会选用一些特征来进行检测, 例如可执行连接格式(executable linking format, ELF)文件头的数据特征和操作码等. 但这些特征依赖于编译程序的系统所使用的 CPU 架构[27]. 这些特征高度依赖于特定的体系结构, 无法用于跨架构物联网恶意软件检测. 因此, 针对物联网恶意代码的检测需要选择跨架构通用的特征, 以解决大量来自不同 CPU 架构的恶意代码带来的挑战.

物联网设备的资源限制特性带来了基于物联网的恶意代码检测的第 2 个独特挑战. 当前的大多数物联网设备, 如可穿戴设备、智能家居、智能电表、无人机集群等, 由于设备体积小、能耗低, 其可以使用的资源受到限制. 例如, 只拥有少量的内存空间、有限的计算能力、低带宽以及低电量[28-31]. 现有恶意代码检测方法大多忽略了物联网设备资源受限的问题. 检测模型的复杂度往往较高, 导致计算成本也高. 因此, 设计基于机器学习或深度学习的轻量级恶意代码检测模型变得至关重要. 这些模型需要通过降低算法复杂度等方式适应物联网设备资源的限制, 从而实现可部署性.

上述 2 个物联网恶意代码检测所遭遇的独特挑战, 源自物联网的 CPU 架构多样性以及物联网设备的资源限制. 这 2 种挑战是物联网设备限制挑战的 2 个方面.

此外, 物联网恶意代码层出不穷, 及时检测到恶意代码攻击, 提高检测模型的准确率, 降低误报率, 也是物联网设备安全和恶意代码检测的一大挑战.

1.3 物联网恶意代码检测常用数据集

由于物联网领域正处于蓬勃发展阶段, 针对物联网设备的攻击也层出不穷, 基于人工智能技术的物联网恶意代码检测研究使用多种数据集训练其设计的机器学习或深度学习检测模型. 我们通过大量的文献调研工作总结了检测工作中常用的物联网恶意代码和恶意流量数据集.

1.3.1 物联网恶意可执行文件数据集

为了给未来的检测工作减少数据收集的负担，本节介绍的物联网恶意代码数据集符合3点要求：1)数据集包含来自多种架构的恶意和良性二进制可执行程序；2)数据集包含近年新构建的样本；3)数据集开源可用，易于获得.

TWISC(Taiwan information security center)研究中心2021年发布的开源数据集[32]包含36 328个样本，包括各种来源的ELF恶意软件和19 975个良性软件，其中样本来源的CPU架构有MIPS、ARM、x86、Super-H4和PPC等. 文献[26]的工作包括收集并开源了一个名为Badthings的恶意样本的物联网恶意代码数据集，此数据集排除了主要存在于服务器、台式机和笔记本电脑中的x86和x86_64恶意软件以及安卓恶意软件，包含来自多种CPU架构的1 66 772个恶意ELF二进制文件. 文献[33]收集了另外一个包含来自不同架构的恶意和良性样本的物联网数据集firmware. IoTPoT[6]通过部署大量蜜罐截获物联网恶意代码样本，开源并定时更新其数据集. 此外，VirusTotal[34]和VirusShare[35]也是很多论文收集物联网恶意代码数据的重要来源. 各种开源物联网恶意代码数据集展示在表1中.

Table 1 Open Source IoT Malware Datasets

表1 开源物联网恶意代码数据集

数据类型	数据集	支持的CPU架构	数据收集方式
可执行文件	TWISC[32]	MIPS，ARM，PPC，SPARC，X86，X86_64	互联网下载
	Badthings[26]	MIPS，ARM，PPC，SPARC，SH4	互联网下载
	firmware[33]	MIPS，ARM，PPC，SPARC，X86，X86_64	互联网下载
	IoTPoT[6]	MIPS，ARM，PPC，SPARC，X86，X86_64，m68k	部署蜜罐捕获
	VirusTotal[34]	多种常见物联网CPU架构	用户自主提交
	VirusShare[35]	多种常见物联网CPU架构	部署检测设备捕获
流量数据包	IoT-23[36]	多种常见物联网CPU架构	真实环境捕获
	Bot-IoT[37]	多种常见物联网CPU架构	实验室模拟
	ToN_IoT[38]	多种常见物联网CPU架构	真实环境捕获
	MedBIoT[39]	多种常见物联网CPU架构	真实环境捕获
	Kitsune[40]	多种常见物联网CPU架构	实验室模拟

1.3.2 物联网恶意流量数据集

物联网恶意流量有较多开源数据集，IoT-23数据集[36]收集了从飞利浦智能LED灯、Somfy智能门锁和亚马逊Echo等多个设备上捕获的运行恶意软件和良性程序时的pcap文件. 新南威尔士大学的研究人员[37-38]贡献了2个流量数据集，分别在实验室模拟现实网络环境，部署恶意代码收集攻击数据，恶意代码部署在智能气象站、智能冰箱、智能灯光控制、远程车库门开关和智能恒温器等真实物联网应用场景中，其中Bot-IoT数据集[37]有超过7 300万条流量数据. ToN_IoT数据集[38]由来自物联网传感器和Ubuntu系统等真实物联网设备的网络流量组成. MedBIoT数据集[39]收集了Mirai等僵尸网络在拥有83个包括智能锁、智能开关等设备的中型网络中的攻击流量. Kitsune数据集[40]收集自一个包括恒温器、婴儿监视器、网络摄像头、低成本的安全摄像头和门铃等物联网设备在内的由3台电脑和9台物联网设备组成的小型网络.

上述5个物联网恶意流量数据集也展示在表1中，由于流量数据集从多种不同设备中获得，流量数据在不同CPU架构上也没有区别，因此统一标记的流量数据来自多种常见物联网CPU架构.

2 物联网恶意代码检测技术分类方法

2.1 现有综述论文分类方法

2008年，针对物联网设备的恶意代码首次被发现[41]，之后几年内才开始大规模出现并引起工业界与研究人员的共同关注[42-43]. 为了介绍当前的物联网恶意代码检测分类方法，本节对现有的综述文章及其物联网恶意代码检测技术的分类方法进行了介绍.

文献[44]聚焦于跨架构物联网恶意软件检测和分析方法，对着眼于解决多架构限制的机器学习物联网恶意软件检测技术的最新研究进行总结，从静态检测特征选取的角度进行了分类分析，将目前的物联网恶意软件检测技术分为基于度量、基于图或

树、基于序列和相互依赖 4 种. 基于度量的特征包括 ELF 文件头、字符串、系统调用、操作码等，基于图或树的特征表示包括控制流图(control flow graph, CFG)、函数调用图(function call graph, FCG)等，基于序列的特征包括字节码、转换为图片等，相互依赖的特征关注 ELF 文件与外部环境之间的关系，包括二进制文件的路径信息等. 然而，该综述的分类方案只考虑了物联网恶意代码静态检测技术，没有考虑物联网恶意软件检测领域中大量使用动态检测技术的相关研究，并且只局限于总结面向跨架构限制的检测技术，没有考虑针对物联网系统自身的其他特性提出的更多恶意代码检测方案.

Ngo 等人[45]对截止到 2020 年的物联网恶意软件静态检测的主要技术论文及其优缺点进行了综述，他们将物联网恶意软件检测方法分为 2 类：未使用图的方法和基于图的方法. 这个分类考虑了现阶段图神经网络的快速发展以及在恶意代码检测领域中的大量应用，但是忽略了动态特征. 此外，随着近几年物联网领域的飞速发展，物联网恶意代码领域也有大量新的研究，但其可能没有涵盖最新的物联网恶意软件检测技术.

文献 [46] 将物联网恶意软件检测方法从使用的技术角度分为基于区块链技术的检测、基于图像技术的检测、基于机器学习的检测和移动恶意软件检测. 但是文献 [46] 的分类方法不够清晰，例如，基于图像技术的检测本质上也是将二进制程序的特征转换为图片形式表示，再使用机器学习技术检测，这也属于基于机器学习检测的一部分.

文献 [31,47−48] 探讨了近年来在保护用户数据及系统安全方面广受关注的联邦学习(federated learning, FL)技术的研究现状，并对应用联邦学习进行物联网恶意代码检测的工作进行了全面分析. 这些文献主要关注在资源受限的物联网设备上应用联邦学习进行检测工作，但对物联网恶意软件检测领域的整体评估尚不完备.

文献 [43, 49] 对 2008—2019 年活跃的物联网恶意代码及恶意代码家族进行了详细调研和梳理，并分别提出了检测分类方案.

虽然现有的物联网恶意代码检测相关综述都很好地总结了特定方向的工作，但是由于物联网领域发展时间短，近年来针对物联网恶意软件的综述文章较少，并且目前的综述文献大多聚焦于具体的检测技术和方法，在较小范围内细分检测技术无法全面展现整个领域当前的研究现状. 基于此，本文对 2018 年以来发表的基于人工智能的物联网恶意代码检测高质量工作进行调研，补充现有的综述工作，从更高的角度提出涵盖范围更大的分类和总结.

2.2 本文分类方法

为了提供一个全新的视角，使研究人员能够全面了解物联网恶意代码检测技术的进展，本文对 2018 年以来在网络与信息安全领域顶级会议和期刊上发表的物联网恶意代码检测相关研究进行了调研. 对这些研究工作解决的问题、主要贡献、使用的机器学习和深度学习算法、物联网恶意代码数据集以及检测效果等方面进行了详细分析. 同时，本文提出了一种新的分类方法，从物联网检测研究的主要动机的角度进行分类，本文分类框架如图 4 所示.

具体而言，相较于现有的综述文献所采用的特定静态或动态分析技术，或是以人工智能算法模型的角度进行分类，本文所提出的分类方法主要着眼

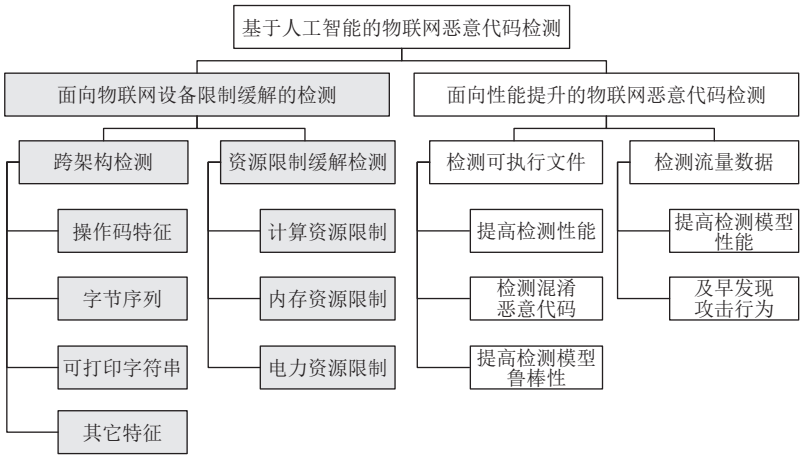


Fig. 4 IoT malware detection taxonomy method proposed by this paper

图 4 本文提出的物联网恶意代码检测分类方法

于基于人工智能的物联网恶意代码检测研究所致力解决的主要问题，即研究动机. 如图 4 左侧浅色阴影框所示，物联网恶意代码检测的一类研究侧重于缓解物联网设备特有的架构及资源限制带来的恶意代码检测挑战，包括跨架构的恶意代码检测和针对物联网设备资源限制的恶意代码检测技术. 如图 4 右侧无浅色阴影框所示，另一类面向检测模型性能提升，这些研究从恶意代码检测的通用检测和分类方法入手，通过使用不同的机器学习算法或特征，提高人工智能算法模型的检测准确率，包括基于流量的恶意行为检测和基于二进制可执行文件的检测技术.

面向物联网设备限制的恶意代码检测技术的主要研究动机是解决物联网设备自身特性导致的恶意代码检测限制. 如 1.1 节中所述，随着物联网在各行各业的不断普及，由于物联网领域的设备存在多样性，而这些设备需要达到高性能、低能耗的标准和更高的安全性要求，无法通过单一处理器架构满足多种设备的不同需求，因而物联网领域存在多种处理器架构. 物联网设备常用的处理器架构包括 MIPS、ARM、PPC 和 SPARC 等，而物联网恶意软件可以在异构设备[50]中传播，这使得运行在物联网设备上的恶意代码的特性与台式机等设备中被限制在有限种类的 CPU 架构上运行的传统恶意软件不同[26]，CPU 架构的差异导致相同的恶意行为呈现出不同的特征[27]，因此无法直接将其它设备上已经较为成熟的机器学习恶意代码检测模型应用到物联网恶意代码检测中. 这部分研究通过分析运行在不同架构上物联网恶意代码的特点，选择新的恶意代码特征，提出在多种架构中检测物联网恶意代码的方法. 另外，物联网设备还具有资源有限、需要持续在线连接、缺乏安全保护等独特的性质. 现有基于人工智能的恶意代码检测方法大多没有考虑计算成本等问题，难以直接部署到物联网环境中. 因此，设计可在资源受限物联网设备上部署的轻量级且准确的恶意代码检测框架也是当前本领域的一个热门研究方向.

除了面向物联网设备和环境独有的特性和限制而设计的检测技术，一部分研究面向物联网恶意代码检测模型的性能提升，其主要研究动机是通过更好地设计检测算法提高现有的物联网恶意代码二进制可执行文件检测方法的准确率和检测效率，减少检测系统的漏报和误报，以及通过基于流量的检测及时检测攻击行为以减少系统安全保障开销.

本文提出的分类方法涵盖了物联网恶意代码检测相关研究工作重点关注的 2 个方面，即物联网设备特性导致的检测技术需要解决的问题和通用恶意代码检测技术在提高检测模型准确率方面有待完善等一般性问题. 首先，如 2.1 节中所述，与其他方法相比，本文分类方法涵盖了物联网恶意代码检测的相关研究，体现了物联网环境和设备独有的特点，而其他分类方法与普通的恶意代码检测分类思路基本类似. 其次，现有的分类方法过分聚焦于具体的人工智能算法，如基于联邦学习的检测技术等，并在更小的范围内细分检测算法，而没有关注物联网领域更高层次的特性. 此外，本文提出的分类方法可以充分体现当前物联网恶意代码研究面临的问题，而现有的分类方法无法为未来的研究提供基于物联网设备特性的发展趋势分析，因此本文分类方法更适合物联网安全领域.

3 面向物联网设备限制缓解的恶意代码检测

物联网设备广泛存在于日常生活使用的物品中. 与传统的台式机、笔记本电脑、智能手机等设备不同，物联网设备由于应用环境的多样性，采用了多种流行的 CPU 架构和操作系统. 同时，与传统计算设备相比，物联网设备的体积通常较小，这限制了其可用的计算资源和内存空间等资源. 这些特性导致一般基于人工智能的恶意代码检测技术难以直接应用于物联网设备上. 为了解决物联网设备的多种架构和资源限制问题，恶意代码检测方法通过有针对性的特征选择和算法设计，提出了适用于物联网环境的高性能恶意代码检测技术.

3.1 跨架构检测技术

物联网设备使用多种 CPU 架构，常用的架构有十多种[51]，这导致了在不同架构上编译的物联网恶意软件之间的差异，使得安全人员无法直接使用传统的 Windows 或移动环境中的分析方法检测物联网恶意代码[51-52]，也无法将在某个设备上获得的恶意样本集直接应用于面向物联网恶意代码检测的人工智能算法. 为了进行跨架构物联网恶意软件检测，可以使用包含大量来自不同 CPU 架构的恶意软件样本的数据集，选择跨架构恶意软件特征，训练机器学习检测模型区分良性和恶意样本. 实现跨架构检测技术的核心在于选择可以跨架构检测出物联网恶意代码的特征，设计高性能的模型，以及使用包含大量不同 CPU 上编译的物联网恶意代码样本数据训练模型. 本节首先梳理了当前研究中用到的跨架构物联网恶意代码数据集，然后从研究工作使

用的不同跨架构特征角度对相关研究工作进行了梳理.

3.1.1 结合操作码的融合特征

操作码是计算机程序的机器语言指令的一部分，用于指定下一步要执行的操作[53]. 操作码指令体现了程序运行时要执行的行为和函数调用等操作. 使用操作码特征进行恶意代码检测的技术最初由 Bilar[54] 提出，近年来出现了很多基于操作码的恶意代码检测工作，这些工作结合了操作码的出现频率[55]、操作码序列[56]、基于图像的操作码[57] 等特征，采用精心设计的机器学习和深度学习算法. 在这些工作中，所选取的特征在很大程度上决定了检测模型的效果. 然而，目前许多研究工作主要依赖于单一特征的恶意代码机器学习检测技术. 这些研究工作[58] 报告的恶意代码检测准确率在 94%~96%，但这些方法通常需要较长的处理时间和较大的计算量[59]. 因此，操作码作为一种有效特征在恶意代码检测领域被广泛应用于基于机器学习的检测模型[54-57,60-61]，在物联网恶意代码检测中也得到了应用.

面对大量运行在不同架构中的物联网恶意代码，不同 CPU 导致来自同一家族的恶意代码的操作码存在一定的区别. 由于单一的操作码特征无法高效准确地实现检测，解决跨架构问题的物联网恶意代码检测方法在使用操作码特征的同时，结合了物联网恶意代码二进制程序的其他特征，实现了融合特征选择，并结合机器学习和深度学习模型来实现物联网恶意代码的跨架构检测.

日本国家信息和通信技术研究所的研究团队[25,52] 在他们的检测工作中选取了操作码和程序运行时的 API 调用序列特征. 他们依据先前研究的发现[62],即加壳后的二进制程序熵值会明显提高，首先计算数据集中每个二进制程序的熵以判断样本是否加壳并从数据集中删除加壳的样本. 然后使用 IDA 得到程序的汇编代码，从中提取操作码序列，同时在沙箱中使用 strace 命令记录样本运行时的 API 调用序列，并使用 N-gram 算法处理操作码和 API 调用序列. 文献 [6] 在 2 个工作中分别应用支持向量机（support vector machine, SVM）和 *k*-最近邻（*k*-nearest neighbor, kNN）算法检测从 IoTPOT 收集的跨架构物联网恶意代码样本，实验结果显示，操作码特征在 ARM 架构上的检测效果优于 API 特征，而 API 特征在 MIPS 上的检测效果略优于操作码特征. 文献 [6] 对检测跨架构物联网恶意代码的特征的方法的有效性进行了分析和验证，但是没有考虑加壳的恶意样本，很难保证在真实物联网环境中的检测效果. Tien 等人[63] 实现了跨指令集架构（instruction set architectures, ISAs），即跨 CPU 的物联网恶意代码检测. 他们选取操作码指令与 ELF 文件的相关属性这 2 组特征，其中包括架构名称、文件大小、外部库、是否加壳、函数数量、是否连接网络等 7 个 ELF 文件特征，以及逻辑、控制、内存、堆栈、输出、算数等不同功能类型的 12 个操作码特征，并在包含 30 000 多个跨架构 IoT 恶意软件样本的数据集中验证了所选特征的有效性，训练和测试了 3 种机器学习模型，其中 CNN 取得了最好的检测效果，在物联网恶意代码家族分类中得到 98.37% 的检测准确率. 此工作同样受限于使用的脱壳技术，无法检测使用复杂反汇编技术的恶意软件.

2020 年，Vasan 等人[58] 提出了一个检测跨架构物联网恶意代码威胁的模型，采用基于操作码和信息增益（information gain, IG）的异构特征选择方法以学习不同层次的语义特征表示，信息增益可以对抗恶意代码常用的垃圾代码混淆，从而准确地检测跨架构的物联网恶意软件. Vasan 等人又提出了名为 MTHAEL 的轻量级堆叠集成模型，使用可以在 MIPS、ARM、PPC 和 Intel X86-64 等架构上传播的 15 482 个恶意代码样本和同样来自多种 CPU 架构的 5 655 个良性样本组成的大数据集上进行训练和测试. MTHAEL 集成了 RNN 和 CNN 这 2 个子网络，把他们嵌入到多头神经网络中，以更好地结合来自每个子网络的预测. 文献 [58] 所提的方法在跨架构样本检测中得到 97.02% 的高检测准确率，并验证了 MTHAEL 面向对抗性攻击的鲁棒性.

除了 API 调用等特征，其他层面也存在不依赖于平台架构的特征，中间表示（intermediate representation, IR）就是其中之一. Vex 中间表示在 Valgrind[64] 和 Angr[65] 等著名的程序分析工具中被使用，文献 [59] 基于 Vex 中间表示和基于控制流特征提取的动态[33]（control flow-based features extraction dynamic, CFD）规划算法实现了一种跨架构 ELF 文件特征选择方法，此方法从操作码对应的 IR 语句中的 Vex 中间表示，调用 N-gram 算法提取基于控制流的特征，然后使用本文工作中收集的开源混合架构数据集训练了一个基于支持向量机的物联网恶意代码检测模型. 基于中间表示的跨架构物联网恶意软件检测方法目前还很少，未来值得被进一步探索.

3.1.2 字节序列

字节序列是恶意代码检测中经常被使用的特征之一[66-67]，在物联网恶意软件检测中也有应用. Wan

等人[68-69]在2020年的2项基于机器学习的检测工作中,同样使用了字节序列特征.他们从ELF程序的入口点开始提取字节作为算法的输入,使用N-gram算法处理字节序列表示为数值向量,在由7种不同CPU架构的超过2万多个样本组成的数据集上训练支持向量机分类器.文献[70]通过对不同CPU上物联网恶意代码样本的详细分析,基于程序主要功能启动时源代码在相同的恶意软件家族不同变体中通常不会改变这一原理,选取恶意软件入口点的字节序列特征,从跨架构恶意代码样本中提取特征后,训练了一个精心设计的Bi-GRU-CNN检测模型,实验结果证明选取的字节序列特征能够准确地区分恶意软件和良性软件.但是基于字节序列特征的方法无法处理加壳的恶意代码,当前的工作大多基于物联网恶意软件还没有大规模使用混淆技术这一发现[70]而开展研究.

3.1.3 可打印字符串

ELF文件中的可打印字符串具有跨平台泛化能力,同时具有高可访问性和高可理解性[27],具体来说,可打印字符串包含与源代码密切相关的基本识别信息,因此可以捕获不同CPU架构上编译的同一系列恶意软件的共同特征.此外,研究人员可以直接从恶意软件的二进制文件中提取可打印字符串,特征提取效率较高且不需要耗费大量内存和计算资源.由3.1.1节可知,二进制文件的操作码依赖体系结构,需要结合其他特征才能取得较好的检测效果,而可打印字符串特征可以直接体现运行在不同CPU架构上的恶意软件样本的共同特征,不需要结合二进制程序的其它特征即可应用于物联网恶意代码检测模型.

Alhanahnah等人[50]使用N-gram算法从ELF文件中提取可打印字符串序列特征,额外选取了二进制程序汇编代码的函数总数、指令总数、重定向指令数、算术指令数、逻辑指令数、传输指令数等6个高级统计特征与可打印字符串序列一起应用于跨架构物联网恶意软件检测模型.文献[27]从ELF文件的函数名称、API名称、代码和代码注释中提取了可打印字符串,以及可打印字符串的数量和长度等特征,在从VirusTotal[34]收集的12万个运行在x86、MIPS、ARM、SPARC、x86-64、PPC和未知类型的CPU架构上的恶意软件ELF文件上提取可打印字符串特征,训练和测试了包括支持向量机在内的3个机器学习分类模型.此项研究工作的实验充分验证了所提出的方法在跨架构CPU样本上的性能,训练模型时使用来自x86、ARM和MIPS等3种常见的CPU架构的样本作为训练集,并使用未知架构和应用较少的架构样本作为测试集,模型得到了平均98%的检测准确率,同时也减少了训练时间.

3.1.4 其他特征

除了被应用最多的操作码和可打印字符串特征,解决跨架构问题的物联网恶意代码检测相关工作中还选取了其他一些特征,包括函数调用图[71]、系统调用函数[72]等.

Wu等人[71]首先使用Radare2[73]对输入二进制文件执行静态分析并创建函数调用图,使用Graph2vec[74]对从物联网恶意软件二进制文件中提取的FCGs进行图嵌入;然后将图嵌入特征与图结构特征相结合,建立物联网恶意软件族分类的训练模型.为验证所提方案的有效性和效率,在一个包含超过10万个物联网恶意软件样本的数据集上进行了实验.这些恶意软件样本分别针对7种不同的CPU架构进行了编译.实验结果显示,支持向量机算法的分类性能最好,在跨架构数据集上5折分层交叉验证的准确率达到了98.88%,但是基于函数调用图的方法需要较长的时间从ELF文件中提取图,在大数据集中会造成很大的时间消耗.Li等人[75]也提取二进制文件的函数调用图,在涵盖5种不同处理器架构的数据集上训练了一个基于图神经网络(graph neural network, GNN)的跨架构物联网恶意软件检测系统.

文献[76]首先构建物联网恶意软件的系统进化树,接着应用基于最小描述长度(minimum description length, MDL)准则的新聚类算法处理待测样本,此研究的一个优点是考虑了恶意样本每天都在快速大量增加的现状,为了保证检测模型对新恶意样本的检测能力,提出了一种直接添加样本的在线处理算法,通过跳过系统进化树重建降低实际操作的计算量,同时保持了恶意代码聚类精度.

物联网恶意代码检测工作大多基于静态分析,基于静态分析可以直接对二进制程序进行反编译及分析等操作,但是静态分析无法直接分析加壳或混淆的恶意代码,此类程序可以使用动态分析.文献[72]使用单一系统调用集特征实现物联网恶意代码检测,此项研究检测了在ARM和Intel X86-32上编译的恶意代码,通过创建2个使用对应CPU的虚拟机以实现动态检测,选取样本在虚拟机中运行时由strace命令跟踪和记录下来的系统调用作为特征,训练了一个名为MDABP的基于平台即服务(platform as a service, PaaS)的物联网恶意软件检测模型.但是该工作

仍然面临很多问题，如创建虚拟机的过程比较复杂耗时，部分样本无法在虚拟机上运行等，这些问题都在都在一定程度上影响本文方法的实际部署.

3.1.5 小 结

在3.1节中我们讨论了面向大量跨架构样本的物联网恶意软件检测工作，并总结在表2中. 当前解决物联网恶意软件跨架构问题的方法通过对不同CPU上编译程序的分析研究，选取字节序列、可打印字符串、控制流图等特征，并设计基于不同机器学习算法的检测模型，在大规模跨架构恶意样本数据集上得到了高准确率，但是仍存在当前大多数基于静态分析的跨架构检测方法无法应对混淆或加壳的恶意代码，基于动态分析的跨架构检测方法存在着不可以统一部署的虚拟机环境等不足.

Table 2 Comparison of AI-based Cross-Architecture IoT Malware Detection Techniques

表2 基于人工智能的跨架构物联网恶意代码检测技术对比

数据来源	特征类别	人工智能算法	数据集	支持的架构	年份
文献 [25]	操作码及API	SVM，kNN	IoTPOT	ARM，MIPS，MIPSE	2019
文献 [52]	操作码及API	SVM，kNN	IoTPOT	ARM，MIPS，MIPSEL	2018
文献 [59]	操作码及CFG	SVM	firmware	ARM，MIPS，PPC，SPARC，X86，X86_64	2019
文献 [63]	操作码及ELF文件特征	CNN	VirusTotal	ARM，MIPS，PPC，SPARC	2020
文献 [58]	操作码及IG	RNN-CNN	IoTPoT，VirusShare	ARM，MIPS，PPC	2020
文献 [68]	字节序列	SVM	VirusTotal	ARM，MIPS，PPC，SPARC，X86、X86_64	2020
文献 [69]	字节序列	SVM	VirusTotal	ARM，MIPS，PPC，SPARC，X86，X86_64	2020
文献 [70]	字节序列	Bi-GRU-CNN	TWISC	ARM，MIPS，X86，SuperH4，PPC	2022
文献 [50]	可打印字符串	聚类	IoTPOT	ARM，MIPS，PPC，SPARC	2018
文献 [27]	可打印字符串	RF，kNN，SVM	VirusTotal	ARM，MIPS，X86，X86-64，PPC，SPARC	2020
文献 [71]	图相关特征	RF，kNN，SVM，MLP，LR	VirusTotal	ARM，MIPS，SPARC，X86，X86-64，PPC	2023
文献 [75]	图相关特征	GNN		ARM，MIPS，SPARC，PPC，X86-64	2021
文献 [72]	系统调用	KMM	VirusTotal	ARM，Intel X86	2023

3.2 面向资源限制的检测技术

大部分物联网设备计算资源非常有限并且内存空间很小[28,30-31,77]. 这些物联网设备上的资源限制导致目前许多适用于通用计算设备的安全防护功能难以在物联网上实现[18,78]，严重制约了物联网安全的发展. 因此，设计计算复杂度低、耗能少，占用内存少的可以缓解资源限制的恶意代码检测系统对于物联网安全是至关重要的.

3.2.1 面向计算资源限制的检测技术

传统的基于机器学习的恶意代码检测方法主要依赖于特征工程，为了提高准确率，这些方法会从恶意软件文件中提取大量不同类型的特征，给分类带来了很高的复杂性[79]. 此外，一般的基于深度学习的恶意代码检测方法，模型复杂且计算成本大，在智能城市、智能家居、智能医院等物联网环境中是不可持续的[80]. 面向计算资源限制的检测方法从选取低维特征和降低算法的复杂度的角度进行了研究.

文献 [79,81−82] 均通过降低特征维度的方法减少它们的物联网恶意代码检测技术对资源的消耗. Qiao 等人[79] 的方法基于 Word2Vec[83] 算法，Word2Vec由谷歌公司开发，是当前比较流行的使用神经网络进行词嵌入的技术. 他们使用 Word2Vec 算法提取二进制程序的十六进制字节和汇编指令的词向量，将每个样本中提取出的向量连接得到新的特征向量，然后训练基于多层感知器（multilayer perception, MLP）的检测模型. 该方法在特征提取阶段既不需要专家经验，又不需要数据依赖，在降低特征维数的同时避免了过度拟合的问题.

文献 [81] 通过精心设计的特征工程方法降低物联网恶意代码流量数据的特征维度，该文献分别使用了基于相关性的4个统计指标，即方差分析、皮尔逊相关系数、互信息和卡方检验，在特征工程之后设计了在各个特征之间进行投票的阶段，最终选择了19个特征. 该文分别训练了3种集成和6种非集成机器学习模型，支持向量机模型和随机森林（random forest, RF）模型得到了100%的检测准确率. 但是该文献只在1个物联网数据集上评估了检测模型，模型的鲁棒性和泛化能力未被验证. Lee 等人[82] 将每个操作码都转换成一个根据其功能分类的操作码类别，从操作码序列中新提取了3种类别特征：操作码分

类序列、操作码分类熵直方图和最大序列模式. 与一般的操作码序列特征相比, 基于操作码类别的 3 种特征表示所需的数据量更少, 因此对计算资源的需求较少, 同时较低的维度具有训练时间更短的优势. 降低特征维度的方法还包括基于恶意样本的视觉表示[80,84]、设计耗费较低计算成本的图像表示方法和特征提取方法, 结合人工智能方法实现资源受限物联网设备上的恶意代码检测. 基于视觉表示的恶意代码检测方法均基于一个假设, 即物联网恶意代码的视觉表示与良性程序有明显区别. Dhanya 等人[84]将可执行文件的字节码生成为 256×256 的 Markov 矩阵, 将矩阵转换为图像, 其中字节序列转换为像素, 得到 256×256 的图像. 使用处理得到的图像训练卷积神经网络(convolutional neural network, CNN), 训练数据集包括了混淆的恶意代码, 实验的结果证明了以二进制程序的 Markov 图像为输入的 CNN 模型对物联网恶意代码的混淆和概念漂移具有弹性. 文献 [80] 提出了基于蚁群优化器(ant clony optimization, ACO)的特征选择方法, 该方法使用物联网恶意代码网络流量的视觉表示作为模型的输入, 在使用低维度特征的同时提升了支持向量机分类器的检测结果.

降低算法复杂度方面, Phu 等人[85]针对早前基于 CFG 结构图提取特征的检测方法存在 NP-hard 难题并且算法复杂度高的问题, 提出了基于动态规划的 C500-CFG 算法, 使用 Angr[65] 的 CFGEmulated 方法提取 ELF 文件的 CFG, 在包含 7 000 个 MIPS 架构上运行的 ELF 程序的数据集中使用 C500-CFG 算法构建 C500 树, 使用 N-gram 算法提取 C500 树的控制流特征, 提取特征的平均时间为 10 s, 最长特征提取时间为 40 s. 实验结果表明 N-gram 算法速度更快, 并且使用更少的内存, 适用于计算资源受限的物联网环境. 在 C500-CFG 算法的基础上, Phu 等人[33]提出了 CFD 算法并将 CFD 算法应用于 MIPS 架构样本的检测. 进一步地, 将 CFD 算法与 ELF 二进制文件的中间表示结合, 实现了低算法复杂度的跨架构恶意软件检测的特征选择方法 CFDVex.

3.2.2 面向内存限制的检测技术

物联网设备为了便于使用、移动和部署, 一般体积较小, 这导致物联网设备的内存和存储空间通常会受到限制[47,86]. 有限的内存容易溢出, 使一般基于机器学习的恶意代码检测系统难以被直接部署在物联网设备中. 面向内存限制的检测技术[27,87]精心设计了轻量级模型, 实现内存占用更小、速度更快的物联网恶意代码检测系统.

2021 年, Giaretta 等人[88]实现了一种名为 LiMNet 的新型轻量级记忆网络(memory networks)检测物联网恶意软件流量以进行僵尸网络早期检测, 不同于一般以网络数据包为中心的设计方法, LiMNet 以物联网设备为中心, 使用记忆网络的组件理解每个物联网设备的行为. 检测模型的输入是构建的特征图, 节点是物联网设备, 节点之间的交互根据网络数据包的源地址和目的地址确定, LiMNet 从图中节点之间的交互流中提取因果关系, 将相关的节点级信息存储在内部结构中, 并使用这些信息来识别僵尸程序. 文献 [87] 同样设计了一个轻量级网络, 在将二进制文件原始字节转换成的 Markov 图像的基础上, 针对物联网设备内存受限的特点修改了经典的卷积神经网络, 提出了轻量级卷积神经网络(lightweight convolutional neural network, LCNN), 该网络在 CNN 中加入了深度卷积(depthwise convolution)和通道洗牌(channel shuffle), 其卷积层的设计与著名的轻量级网络 ShuffleNetV2[89] 基本相同, 但是单元数量更少. 与其他基于深度学习的方法相比, LCNN 模型的大小只有 1 MB, 而 VGG16 的模型有 552.57 MB, 由此可见, LCNN 可以在保持准确性的同时显著减少训练模型所需的资源消耗.

3.2.3 面向电力限制的检测技术

由于物联网设备受到体积和硬件限制的原因[90], 用于给物联网设备供电的电池通常容量不高, 并且许多物联网设备在部署后, 电池一般不需要短期更换, 因此设备上程序的运行受到电量的限制. 轻量级和高速的检测模型可以解决电力资源限制问题, 然而, 最近的研究[91-93]提出了更有效的方案, 这些检测方案可以独立于设备可用资源进行部署, 从根源上规避了物联网环境中部署检测系统受到的电力资源限制问题.

文献 [91] 以嵌入式设备的电磁辐射为分析对象, 选择树莓派 2B 作为目标设备, 在执行恶意代码时检测设备外部的电磁辐射, 处理数据中的噪声, 并使用软件分析保护机制扩展后的数据集来训练卷积神经网络模型. 此检测模型不依赖设备和系统架构, 也不会导致物联网设备的计算开销, 并且实验结果也证明了此模型对于未知的混淆样本具有较高的鲁棒性. DeepPower[92] 监控被用于保护设备的功率信号, 通过分析侧信道功率信号推断物联网恶意软件活动, 首先快速检测出可疑功率信号, 然后使用基于注意力的 Seq2Seq 模型实现对可疑信号的细粒度分析. 训练的深度学习模型在检测 Mirai 恶意软件时体现了很

好的鲁棒性，可以及时检测物联网恶意代码入侵. Azmoodeh 等人[93]通过监控物联网设备的电量使用情况实现检测. 通过记录所有运行的进程的电量消耗情况，使用电量消耗数据训练支持向量机模型，实现了检测.

3.2.4 小 结

在 3.2 节中我们分类介绍了目前以解决物联网设备受到计算资源、内存空间以及电力资源等限制为研究动机的物联网恶意代码检测方法，并总结在表 3 中. 当前针对物联网资源限制的研究工作相比传统的基于机器学习或深度学习的方法，只需要少量的计算资源和内存资源，检测速度快. 但是，当前模型大多侧重于解决资源限制问题，只在来自单一架构的数据集上训练和测试模型，没有在不同架构上编译的恶意程序中验证模型的检测效果，模型的鲁棒性和泛化能力有限.

Table 3 Comparison of Resource-constrained AI-based IoT Malware Detection Techniques

表 3 基于人工智能资源限制的物联网恶意代码检测技术对比

数据来源	资源限制类别	限制缓解方案	特征选择/处理方法	人工智能算法	年份
文献 [79]	计算资源限制	降低特征维度	Word2Vec	MLP	2021
文献 [80]	计算资源限制	降低特征维度	ACO	SVM	2023
文献 [81]	计算资源限制	降低特征维度	相关性分析	Random Forest，Bagging，Stacking，SVM，LR，kNN	2021
文献 [82]	计算资源限制	降低特征维度	无	SVM，Random Forest，Decision Tree	2023
文献 [84]	计算资源限制	降低特征维度	图像化	CNN	2023
文献 [33]	计算资源限制	降低算法复杂度	N-gram	CFD	2020
文献 [85]	计算资源限制	降低算法复杂度	C500-CFG	SVM	2019
文献 [88]	内存限制	设计轻量级模型	无	LiMNet	2021
文献 [87]	内存限制	设计轻量级模型	图像化	LCNN	2021
文献 [91]	电力限制	设计可独立于物联网设备部署的检测	图像化	CNN	2021
文献 [92]	电力限制	设计可独立于物联网设备部署的检测	过滤电信号噪声	Seq2Seq	2020
文献 [93]	电力限制	设计可独立于物联网设备部署的检测	无	SVM	2018

4 面向性能提升的物联网恶意代码检测

第 3 节详细介绍了物联网设备和环境所面临的挑战和限制，并针对解决物联网设备特性带来的恶意代码检测问题进行了研究，取得了一系列的研究成果. 而本节主要介绍旨在提高所使用的机器学习和深度学习模型性能和检测效果的物联网恶意代码检测工作. 这些工作针对恶意代码检测领域的一般性问题，主要关注于提升基于人工智能算法的检测方法的准确率，降低误报率以及提高检测速度等方面.

在本节中，我们根据所分析对象的不同，将基于通用技术的研究工作划分为基于二进制可执行文件的检测技术和基于流量的检测技术 2 类. 这 2 类检测技术均将机器学习或深度学习用于物联网恶意代码检测，并通过选择不同的特征或修改人工智能模型来提高检测效果.

4.1 基于可执行文件分析的检测性能提升

在物联网环境中，绝大多数物联网系统依赖于基于 Linux 的操作系统，其上运行的程序为 ELF 文件[94]，分析物联网系统中的恶意 ELF 文件并结合深度学习算法实现检测系统已经是物联网安全领域的重要部分. 除针对物联网环境和设备的跨架构和资源限制特性的检测方法外，基于二进制可执行文件的检测技术还解决恶意代码检测面临的一些普遍问题，包括提高模型检测效果和模型对攻击的鲁棒性[95-100]、检测使用各种混淆技术的恶意代码[101-103]等，另有一小部分工作通过对 ELF 文件的详细分析，对比了同为 ELF 文件的物联网恶意代码与安卓恶意代码[104-105].

Dib 等人[96]从可执行二进制文件中提取可打印字符串特征并结合二进制文件转换成的图片，以及使用超过 7 万个最新的物联网恶意代码样本训练了一个结合 CNN 和长短期记忆（long short-term memory, LSTM）网络的检测模型，模型的准确率较现有方法有所提升，但是此方法没有考虑混淆的恶意代码样本. 类似地，文献 [99] 同样使用待测样本转换成的 RGB 图片作为其检测模型的输入，并利用深度迁移学习，通过融合 ResNet18[106]、MobileNetV2[107] 和 Dense-

Net161[108]这3个卷积神经网络,提高了模型的检测和分类性能.

另一方面,OGCNN-RWD[98]是一种基于最优图卷积神经网络的勒索软件检测技术,OGCNN-RWD使用图卷积神经网络(graph convolutional neaval network, GCNN)模型,通过和谐搜索算法(harmony search algorithm, HSA)进行参数选择.模糊模式树(fuzzy pattern tree, FPT)[109]在物联网恶意代码检测中也得到了应用,该树状结构具备处理模糊性和不可见条件的能力,提高了模糊方法对恶意代码变化的鲁棒性.此外,文献[97]提取控制流图相关特征并输入模糊模式树,实验结果表明使用模糊树和快速模糊树方法的检测结果优于使用支持向量机、决策树、k最近邻和随机森林等其他机器学习算法.但是模糊模式树算法计算复杂度高,运行时间长,不便于在物联网环境中实际部署.Yumlembam等人[100]提出基于图神经网络(graph neural network, GNN)的分类器用于检测恶意软件.他们的研究表明,基于GNN的检测模型鲁棒性较差,针对此问题,他们引入了名为VGAE-MalGAN的对抗样本生成模型,并通过实验证明使用VGAE-MalGAN生成的对抗样本进行再训练可以提高检测方法的鲁棒性.

面向物联网架构及资源限制的恶意代码检测技术侧重于解决物联网环境和设备特性对恶意代码检测系统造成的限制,但未考虑使用混淆技术的恶意代码对检测方法造成的影响[25,27,50,70].因此,面向性能提升的物联网恶意代码检测工作[101-103]针对这一问题展开了探索,提升了在混淆恶意样本上的检测准确率.

Darabian等人[101]采用序列模式挖掘技术,提取二进制可执行样本操作码序列的最大频繁模式作为特征,并利用开源的多态恶意软件创建工具构建了6个多态恶意代码数据集,在这些数据集以及来自ARM架构的恶意样本集上,他们训练了包括k-最近邻、支持向量机在内的多个机器学习模型.为了减轻使用多态和代码混淆技术的恶意代码对标准检测方法的影响,文献[102]提出了基于迁移学习的预训练Inception-v3模型检测框架,该框架能够对模型进行微调,其输入为恶意代码二进制文件转换成的RGB图片.实验证明,基于图像的恶意软件模型不需要特征工程,构建速度快,能够对抗代码混淆,并且在各项指标上优于使用类似技术的其他方法.

2-MaD[103]是一种2阶段的物联网恶意代码混合检测方案,旨在部署于智能城市环境中以保护物联网设备免受混淆恶意软件的攻击.2-MaD首先在第1阶段执行静态分析,提取样本的操作码特征,训练双向长短期记忆(bidirectional long-short term memory, Bi-LSTM)模型进行检测.随后,在第2阶段中,对可能被静态分析误报为良性的恶意代码进一步地动态分析,提取虚拟机的行为日志中的进程内存信息,训练EfficientNet-B3[110]模型再次检测恶意代码.相较于单一的静态检测或动态检测,该方法具有更低的误报率,但是由于在动态分析阶段执行一个样本需要2 min时间,2-MaD技术的时间消耗大.HyMalD[111]是另一个混合检测工具,与2-MaD不同,HyMalD同时执行基于静态分析和基于动态分析的物联网恶意代码检测.静态检测部分提取物联网恶意代码的操作码特征训练了Bi-LSTM模型;动态检测部分提取样本在沙箱中运行产生的进程、文件和注册表行为作为特征并转换为RGB图像训练SPP-Net模型[112].实验结果显示,相较于单一的静态检测方法,HyMalD具有较低的漏报率和更高的准确率.

物联网系统主要运行基于Linux系统的ELF文件,而攻击安卓系统的恶意软件也基于Linux系统[105].随着物联网设备的普及,分析物联网恶意软件与其他基于Linux系统的恶意软件之间的差异有助于构建更有效的检测系统.

文献[104]通过提取物联网和安卓恶意代码样本的控制流图,并从图的大小、直径、最短路径分布、节点数量和中介中心性等图论相关的图属性对2种恶意代码样本进行对比分析,发现安卓恶意软件的节点数量更多,与物联网恶意软件相比安卓恶意软件具有更高的复杂性.Alasmary等人[105]的研究是在文献[104]工作基础上的扩展,他们分别分析了近3 000个物联网和安卓恶意软件样本以及良性样本数据的控制流图相关特征,并提取了平均最短路径、度中心性和中介中心性等具有高区分度的特征.然后,他们利用机器学习和深度学习方法训练了基于这些特征的物联网恶意代码检测模型,其中卷积神经网络检测模型得到了误报率最低且准确率最高的实验结果.然而,这一系列对比研究的不足是因为没有考虑其检测模型对代码混淆和对抗样本攻击的鲁棒性.

4.2 基于流量分析的检测性能提升

很大一部分物联网恶意软件以僵尸程序的形态出现[113].一旦感染物联网设备,僵尸程序会与攻击者的命令与控制(command and control, C&C)服务器进行通信,并根据黑客的指令执行恶意攻击,如DDoS攻击等[114].僵尸网络是由被僵尸程序感染的设备组成的网络[115].随着物联网设备部署数量的迅速增加,

针对物联网的 DDoS 攻击流量也达到了前所未有的水平[116-118]. 及时检测此类攻击并断开受感染设备与网络的链接对物联网安全至关重要. 基于流量的检测是对设备上运行的二进制可执行程序检测系统的补充, 它使安全管理人员可以在攻击发生的早期阶段实时发现可能的恶意行为, 并减少系统安全保障开销.

基于流量的检测通常被视为基于动态特征的行为检测技术, 在物联网恶意代码检测领域中, 由于僵尸程序的广泛存在, 也有一些基于流量的恶意行为检测方法研究[15,114,119-124]. 例如, N-BaIoT[119] 从受感染的物联网设备中提取网络行为快照, 并使用深度自动编码器检测异常网络流量. Jamal 等人[15] 利用 ToN_IoT 数据集[38] 提取网络数据包的源端口、目的端口、时间戳、链接状态等特征, 并训练了基于深度学习的检测模型. Alharbi 等人[122] 使用主成分分析 (principal component analysis, PCA) 方法对特征进行降维, 并利用开源的 IoT-23 数据集[36] 提取特征后训练随机森林分类器, 以提高机器学习模型的检测性能并降低过拟合的风险. 文献 [121] 将关联规则学习[125] 应用于物联网流量分析, 除了选取流量包的端口等特征, 还使用其他报头信息并行进行所有报头信息规则挖掘, 以便检测未知恶意流量.

文献 [114] 将 FastGRNN[126] 用于物联网恶意代码产生流量的检测, FastGRNN 相比于传统的 RNN 模型具有更低的复杂性, 能够提供快速的训练和攻击检测能力. 分布式模块化的检测方案 EDIMA[120] 不仅包含基于机器学习算法的检测模型, 而且包括一个定期使用新捕获的流量重新训练机器学习模型的模型构造器模块和一个数据包流量特征数据库, EDIMA 用于存储提取的已知恶意数据特征向量列表并及时更新, 通过定期进行模型重新训练以应对大量新型物联网恶意代码引起的概念漂移等问题. 物联网恶意软件分析系统 BOTA[124] 是基于机器学习和基于规则的异构分类器的集合, 可在高速计算机网络上对受感染的物联网设备及时实现可解释的检测.

4.3 小　结

本节对第 4 节中介绍的物联网安全领域面向性能提升和基于通用人工智能技术进行的恶意代码检测研究进行总结, 并展示在表 4 中.

基于可执行文件分析的检测性能提升方法通过利用深度学习方法自动提取特征, 减少对专家知识的依赖, 提高检测模型的性能, 但是当前的很多检测研究没有考虑对抗样本等攻击对机器学习和深度学习模型的安全威胁. 基于流量分析的检测性能提升方法为攻击早期阶段检测物联网恶意代码提供了支持. 然而, 由于攻击流量数量巨大, 用于训练检测系统的数据集无法实时更新, 导致基于流量的检测系统的准确率可能随时间降低.

Table 4　Comparison of Performance-enhancing IoT Malware Detection Techniques

表 4　性能提升的物联网恶意代码检测技术对比

数据来源	检测对象	技术重心	人工智能算法	数据集来源	年份
文献 [96]	二进制可执行文件	提高检测准确率	CNN+LSTM	VirusTotal，VirusShare	2021
文献 [99]	二进制可执行文件	提高检测准确率	ResNet18+MobileNetV2+DenseNet161	未公开	2022
文献 [98]	二进制可执行文件	提高检测准确率	GCNN	未公开	2023
文献 [97]	二进制可执行文件	提高检测准确率	FPT	VirusShare	2019
文献 [101]	二进制可执行文件	检测混淆恶意代码	SVM	未公开	2020
文献 [102]	二进制可执行文件	检测混淆恶意代码	Inception-v3	未公开	2022
文献 [103]	二进制可执行文件	检测混淆恶意代码	Bi-LSTM+EfficientNet-B3	VirusTotal	2021
文献 [111]	二进制可执行文件	检测混淆恶意代码	Bi-LSTM+SPP-Net	未公开	2022
文献 [100]	二进制可执行文件	提高检测模型鲁棒性	VGAE-MalGAN	未公开	2023
文献 [119]	流量数据	及时发现攻击行为	Auto-encoder	未公开	2018
文献 [15]	流量数据	提高检测模型性能	ANN	ToN_IoT	2022
文献 [122]	流量数据	提高检测模型性能	RF	IoT-23	2022
文献 [114]	流量数据	提高检测模型性能	FastGRNN	MedBIoT	2020
文献 [120]	流量数据	及时发现攻击行为	LR，kNN，RF，AdaBoost	未公开	2019

5 挑战与机遇

本节对当前研究工作中存在的问题和未解决的挑战进行了总结,基于当前已有的研究工作存在的不足提出了未来的研究方向. 表 5 展示了当前的挑战与机遇.

Table 5 Challenges and Opportunities

表 5 挑战与机遇

类别	具体内容
面临的挑战	缺少针对物联网设备特性的评估指标; 缺少统一可用的动态分析环境; 缺乏对攻击的鲁棒性.
未来研究方向	结合大模型实现物联网恶意代码检测; 提高检测模型安全性; 结合零信任架构实现检测.

5.1 当前工作面临的挑战

1) 缺少针对物联网设备特性的评估指标. 目前基于人工智能检测的相关研究在评价检测方案的效果时,通常选择用于评估机器学习和深度学习算法性能的评价指标,包括准确率、精确度、召回率和 $F1$ 值以及漏报率和误报率等. 这些指标在应用于各种场景的人工智能模型中都被使用,虽然检测准确率等指标可以反映模型的检测能力,但是由于物联网设备还存在独有的特性和限制,这些通用的评估指标无法全面评估物联网设备上恶意代码的检测模型. 例如,针对物联网中存在多种不同架构,不同架构上的恶意代码可能存在较大区别的特性,应在检测准确率等评价之外对检测模型的跨架构检测能力进行定量评估. 此外,由于物联网设备的资源限制,我们在调研中也发现很多解决资源限制问题的研究设计了基于人工智能的轻量级检测模型,但是这些轻量级模型的资源消耗量等数据同样没有定量评价标准.

2) 缺少统一可用的动态分析环境. 目前使用的机器学习和深度学习的物联网恶意代码检测工作中使用动态分析方法的较少[72,103,111]. 虽然自从 IoTPOT[6] 被公开以来,其已经成为物联网恶意代码分析数据集的稳定来源,但是动态分析过程中研究人员试图通过样本的执行提取自定义的特征,如运行的进程等,这需要一个可以直接部署使用的开源沙箱. 随着物联网恶意代码技术的迭代,其检测沙箱环境逃避动态分析,因此当前需要的动态分析环境需要有模拟真实设备环境的反逃逸设置,同时便于快速部署多个虚拟环境,以适应物联网恶意代码多种版本运行在不同 CPU 架构上的特点.

3) 缺乏对攻击的鲁棒性. 随着近年来人工智能技术在各个领域的蓬勃发展,人工智能的安全性也逐渐成为被研究人员关注的领域. 在对基于人工智能的物联网恶意代码检测技术的大规模调研中,我们发现大多数检测物理联网恶意代码的机器学习模型在设计时没有考虑针对模型的攻击,由于真实部署环境中存在大量攻击者,检测模型的安全实际上面临着较大风险. 最近几年也有许多针对人工智能恶意代码检测工具的对抗性攻击研究[127-129],目前常见的攻击包括对模型训练数据的数据投毒攻击、针对人工智能模型的对抗样本攻击等.

5.2 未来研究方向

为了实现更高效、更实用的基于人工智能的物联网恶意代码检测技术,为物联网安全领域的发展注入新的活力,本节在总结现有研究的基础上,提出了 3 个未来可能的研究方向,旨在为研究人员进一步的工作提供参考.

1) 结合大模型实现物联网恶意代码检测. 近年来随着人工智能领域的突破性技术 ChatGPT 的出现,大模型也成为研究的热点. 随着计算能力的提高,通过亿级数据训练的大模型拥有着远远超出一般机器学习和深度学习的性能. AI 大模型与安全领域相结合,可以实现快速分析恶意代码,引入大模型辅助物联网恶意代码检测,使用训练好的大模型直接检测恶意代码将会极大提高恶意代码检测技术的效率. 同时可以通过大量数据的训练以产生针对物联网特性的新的检测评估指标,实现高效准确的物联网恶意代码检测和完善的模型检测效果评估.

2)提高检测模型安全性. 我们对当前研究工作的分析中指出了目前基于人工智能的物联网恶意代码检测技术缺乏模型鲁棒性保障的问题. 针对真实环境的部署中检测模型可能遭到的对抗样本攻击,未来的研究工作应该考虑提高检测模型的安全性,这也是未来人工智能安全领域模型安全的研究方向之一. 可以通过在设计物联网恶意代码检测模型时引入对抗训练和随机化数据等方法提高检测模型的安全性.

3) 结合零信任架构实现检测. 零信任安全模型是近年来提出的安全防御新架构. 由于物联网环境中设备数量巨大且正在快速扩张,传统的防御方案无法在物联网设备中得到大规模应用. 结合基于不

信任任何设备或用户原则的零信任架构要求对设备实时认证和授权访问，攻击者无法通过感染一个设备快速感染联网的其他设备，可以有效减轻基于人工智能的恶意代码检测系统的负担.

6 总　　结

近年来，物联网安全越来越受到重视，一方面由具于物联网设备数量激增，另一方面也因为针对物联网设备的恶意代码等威胁不断涌现. 物联网恶意代码检测在物联网安全中扮演着举足轻重的角色. 本文对2018年至今的基于人工智能的物联网恶意代码检测技术进行了大规模调研，从物联网设备区别于一般台式机等设备的特性导致的问题出发，提出了面向主要研究动机的分类方法，从物联网设备限制缓解的恶意代码检测方面和方面性能提升的物联网恶意代码检测方面对当前的检测工作进行了分析和梳理. 基于对现有研究的全面总结，分析了该领域当前面临的挑战，并提出了未来的研究方向.

作者贡献声明：刘奇旭负责论文的总体规划、指导以及论文的撰写；刘嘉熹负责论文主要内容的调研和撰写；靳泽负责论文结构的梳理；刘心宇、肖聚鑫负责相关工作的调研和梳理；陈艳辉负责论文内容的梳理和校对；朱洪文、谭耀康负责相关文献的整理和内容校对.

参　考　文　献

[1] STATISTA. Number of Internet of things (IoT) connected devices worldwide from 2019 to 2021 [EB/OL]. [2023-05-29]. https://www.statista.com/statistics/1183457/iot-connected-devices-worldwide/

[2] Wall S. 2022 Cyber threat report[R]. Santa Clara, CA: Palo Alto Networks, 2022: 5−7

[3] Antonakakis M, April T, Bailey M, et al. Understanding the Mirai botNet[C] //Proc of the 26th USENIX Security Symp. Berkeley, CA: USENIX Association, 2017: 1093−1110

[4] Griffioen H, Doerr C. Examining Mirai's battle over the Internet of things[C] //Proc of the 27th ACM SIGSAC Conf on Computer and Communications Security (CCS). NewYork: ACM, 2020: 743−756

[5] CNCERT. IoT security threat intelligence report (Oct 2022)[R]. Beijing: National Computer Network Emergency Response Technical Team/Coordination Center of China, 2022: 1−426 (in Chinese)
(国家互联网应急中心. 物联网安全威胁情报(2022年10月)[R]. 北京：国家计算机网络应急技术处理协调中心, 2022: 1−426)

[6] Pa Y M P, Suzuki S, Yoshioka K, et al. IoTPOT: Analysing the rise of IoT compromises[C] //Proc of the 9th USENIX Workshop on Offensive Technologies (WOOT). Berkeley, CA: USENIX Association, 2015: 1−9

[7] Breitenbacher D, Homoliak I, Aung Y L, et al. HADES-IoT: A practical host-based anomaly detection system for IoT devices[C] //Proc of the 24th ACM Asia Conf on Computer and Communications Security. New York: ACM, 2019: 479−484

[8] Çetin O, Gañán C, Altena L, et al. Cleaning up the Internet of evil things: Real-world evidence on ISP and consumer efforts to remove mirai[C] //Proc of the 26th Annual Network and Distributed System Security Symp (NDSS). Reston, VA: The Internet Society, 2019: 1−5

[9] Xu Yiwen, Jiang Yu, Yu Lu, et al. Brief industry paper: Catching IoT malware in the wild using HoneyIoT[C] //Proc of the 27th Real-Time and Embedded Technology and Applications Symp (RTAS). Piscataway, NJ: IEEE, 2021: 433−436

[10] Dang Fan, Li Zhenhua, Liu Yunhao, et al. Understanding fileless attacks on Linux-based IoT devices with honeycloud[C] //Proc of the 17th Annual Int Conf on Mobile Systems, Applications, and Services. New York: ACM, 2019: 482−493

[11] Downing E, Mirsky Y, Park K, et al. DeepReflect: Discovering malicious functionality through binary reconstruction[C] //Proc of the 30th USENIX Security Symp. Berkeley, CA: USENIX Association, 2021: 3469−3486

[12] Alrawi O, Ike M, Pruett M, et al. Forecasting malware capabilities from cyber attack memory images[C] //Proc of the 30th USENIX Security Symp. Berkeley, CA: USENIX Association, 2021: 3523−3540

[13] Wang Qi, Wajih H, Ding Li, et al. You are what you do: Hunting stealthy malware via data provenance analysis[C] //Proc of the 27th Annual Network and Distributed System Security Symp (NDSS). Reston, VA: The Internet Society, 2020: 1−17

[14] Chen Jinyin, Hu Keke, Yu Yue, et al. Software visualization and deep transfer learning for effective software defect prediction[C] //Proc of the ACM/IEEE 42nd Int Conf on Software Engineering. New York: ACM, 2020: 578−589

[15] Jamal A, Hayat M F, Nasir M. Malware detection and classification in IoT network using ANN[J]. Mehran University Research Journal Of Engineering & Technology, 2022, 41(1): 80−91

[16] Zhang Shuqin, Bai Guangyao, Li Hong, et al. IoT security knowledge reasoning method of multi-source data fusion[J]. Journal of Computer Research and Development, 2022, 59(12): 2735−2749 (in Chinese)
(张书钦, 白光耀, 李红, 等. 多源数据融合的物联网安全知识推理方法[J]. 计算机研究与发展, 2022, 59(12): 2735−2749)

[17] Zhang Yuqing, Zhou Wei, Peng Anni. Survey of internet of things security[J]. Journal of Computer Research and Development, 2017, 54(10): 2130−2143 (in Chinese)
(张玉清, 周威, 彭安妮. 物联网安全综述[J]. 计算机研究与发展, 2017, 54(10): 2130−2143)

[18] Yang Yiyu, Zhou Wei, Zhao Shangru, et al. Survey of IoT security research: Threats, detection and defense[J]. Journal on

Communications, 2021, 42(8): 188−205 (in Chinese)
(杨毅宇, 周威, 赵尚儒, 等. 物联网安全研究综述: 威胁, 检测与防御[J]. 通信学报, 2021, 42(8): 188−205)

[19] Chen Jiongyi, Diao Wenrui, Zhao Qingchuan, et al. IoTFuzzer: Discovering memory corruptions in IoT through app-based fuzzing[C] //Proc of the 25th Annual Network and Distributed System Security Symp (NDSS). Reston, VA: The Internet Society, 2018: 1−15

[20] Symantec. Symantec: Security response [EB/OL]. [2023-05-16]. https://www.symantec.com/connect/blogs/iot-devicesbeing-increasingly-used-ddos-attacks

[21] Egg S. Standing egg chooses MIPS CPUs for sensor hubs targeting mobile, IoT, wearables and automotive [EB/OL]. [2023-05-29]. https://electroiq.com/files/files/ables-and-automotive_xqh4b2rmudu6skntuhhadd/

[22] Peng Anni, Zhou Wei, Jia Yan, et al. Survey of the Internet of things operating system security[J]. Journal on Communications, 2018, 39(3): 22−34 (in Chinese)
(彭安妮, 周威, 贾岩, 等. 物联网操作系统安全研究综述[J]. 通信学报, 2018, 39(3): 22−34)

[23] Hackett R. Why a hacker dumped code behind colossal website trampling botnet [EB/OL]. [2023-05-16]. https://finance.yahoo.com/news/why-hacker-dumped-code-behind-145847907.html?

[24] Shekari T, Cardenas A A, Beyah R. MaDIoT 2.0: Modern high-wattage IoT botnet attacks and defenses[C] //Proc of the 31st USENIX Security Symp. Berkeley, CA: USENIX Association, 2022: 3539−3556

[25] Ban T, Isawa R, Huang S Y, et al. A cross-platform study on emerging malicious programs targeting IoT devices[J]. IEICE Transactions on Information and Systems, 2019, 102(9): 1683−1685

[26] Alrawi O, Lever C, Valakuzhy K, et al. The circle of life: A large-scale study of the IoT malware lifecycle[C] //Proc of the 30th USENIX Security Symp. Berkeley, CA: USENIX Association, 2021: 3505−3522

[27] Lee Y T, Ban T, Wan T L, et al. Cross platform IoT-malware family classification based on printable strings[C] //Proc of the 19th Int Conf on Trust, Security and Privacy in Computing and Communications (TrustCom). Piscataway, NJ: IEEE, 2020: 775−784

[28] Clements A A, Almakhdhub N S, Saab K S, et al. Protecting bare-metal embedded systems with privilege overlays[C] //Proc of the 2017 IEEE Symp on Security and Privacy (SP). Piscataway, NJ: IEEE, 2017: 289−303

[29] Xu Yiwen, Yin Zijing, Hou Yiwei, et al. MIDAS: Safeguarding IoT devices against malware via real-time behavior auditing[J]. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2022, 41(11): 4373−4384

[30] Zhou Jie, Du Yufei, Shen Zhuojia, et al. Silhouette: Efficient protected shadow stacks for embedded systems[C] //Proc of the 29th USENIX Security Symp. Berkeley, CA: USENIX Association, 2020: 1219−1236

[31] Imteaj A, Thakker U, Wang Shiqiang, et al. A survey on federated learning for resource-constrained IoT devices[J]. IEEE Internet of Things Journal, 2021, 9(1): 1−24

[32] TWISC Research Centers. IoT-based end-to-end system security [EB/OL]. [2023-05-29].https://www.twisc.org/research-centers/

[33] Phu T N, Tho N D, Hoang L H, et al. An efficient algorithm to extract control flow-based features for IoT malware detection[J]. The Computer Journal, 2020, 64(4): 599−609

[34] VirusTotal. VirusTotal [EB/OL]. [2023-05-19]. https://www.virustotal.com/gui/home/upload

[35] VirusShare. VirusShare [EB/OL]. [2023-05-26]. https://virusshare.com/

[36] Parmisano A, Garcia S, Erquiaga M J. A labeled dataset with malicious and benign IoT network traffic[EB/OL]. [2023-06-04]. https://www.stratosphereip.org/datesets-iot23

[37] Koroniotis N, Moustafa N, Sitnikova E, et al. Towards the development of realistic botnet dataset in the Internet of things for network forensic analytics: Bot-IoT dataset[J]. Future Generation Computer Systems, 2019, 100: 779−796

[38] Moustafa N. The ToN_IoT datasets [EB/OL]. [2023-05-24]. https://www.unsw.adfa.edu.au/unsw-canberra-cyber/cybersecurity/ADFA-ton-iot-Datasets/

[39] Guerra-Manzanares A, Medina-Galindo J, Bahsi H, et al. MedBIoT: Generation of an IoT botnet dataset in a medium-sized IoT network[C] //Proc of the 6th Int Conf on Information Systems Security and Privacy (ICISSP). Setúbal, Portugal: Scitepress, 2020: 207−218

[40] Mirsky Y, Doitshman T, Elovici Y, et al. Kitsune: An ensemble of autoencoders for online network intrusion detection[J]. arXiv preprint, arXiv: 1802.09089, 2018

[41] Vailshery L S. Malware for network devices [EB/OL]. [2023-05-29]. https://www.statista.com/statistics/1183457/iot-connected-devices-worldwide/

[42] Ronen E, Shamir A, Weingarten A O, et al. IoT goes nuclear: Creating a ZigBee chain reaction[C] //Proc of the 2017 IEEE Symp on Security and Privacy (S&P). Piscataway, NJ: IEEE, 2017: 195−212

[43] Costin A, Zaddach J. IoT malware: Comprehensive survey, analysis framework and case studies[R]. Las Vegas, LA: Black Hat, 2018: 1−9

[44] Raju A D, Abualhaol I Y, Giagone R S, et al. A survey on cross-architectural IoT malware threat hunting[J]. IEEE Access, 2021, 9: 91686−91709

[45] Ngo Q D, Nguyen H T, Le V H, et al. A survey of IoT malware and detection methods based on static features[J]. ICT Express, 2020, 6(4): 280−286

[46] Clincy V, Shahriar H. IoT malware analysis[C] //Proc of the 43rd Annual Computer Software and Applications Conf (COMPSAC). Piscataway, NJ: IEEE, 2019: 920−921

[47] Imteaj A, Mamun Ahmed K, Thakker U, et al. Federated learning for resource-constrained IoT devices: Panoramas and state of the art[J]. Federated and Transfer Learning, 2022, 27: 7−27

[48] Venkatasubramanian M, Lashkari A H, Hakak S. IoT malware analysis using federated learning: A comprehensive survey[J]. IEEE Access, 2023, 11: 5004−5018

[49] Vignau B, Khoury R, Hallé S, et al. The evolution of IoT malwares,

from 2008 to 2019: Survey, taxonomy, process simulator and perspectives[J]. Journal of Systems Architecture, 2021, 116: 102143

[50] Alhanahnah M, Lin Qicheng, Yan Qiben, et al. Efficient signature generation for classifying cross-architecture IoT malware[C] //Proc of the Conf on Communications and Network Security (CNS). Piscataway, NJ: IEEE, 2018: 1−9

[51] Cozzi E, Graziano M, Fratantonio Y, et al. Understanding Linux malware[C] //Proc of the 2018 IEEE Symp on Security and Privacy (S&P). Piscataway, NJ: IEEE, 2018: 161−175

[52] Ban T, Isawa R, Yoshioka K, et al. A cross-platform study on IoT malware[C] //Proc of the 11th Int Conf on Mobile Computing and Ubiquitous Network (ICMU). Piscataway, NJ: IEEE, 2018: 1−2

[53] Davie H. Assemblers and Loaders: DW Barron, Macdonald and Janes Computer Monographs[M]. Amsterdam, Netherlands: Elsevier, 1979

[54] Bilar D. Opcodes as predictor for malware[J]. International Journal of Electronic Security and Digital Forensics, 2007, 1(2): 156−168

[55] Yewale A, Singh M. Malware detection based on opcode frequency[C] //Proc of the 2016 Int Conf on Advanced Communication Control and Computing Technologies (ICACCCT). Piscataway, NJ: IEEE, 2016: 646−649

[56] Santos I, Brezo F, Nieves J, et al. Idea: Opcode-sequence-based malware detection[C] //Proc of the Engineering Secure Software and Systems: Second Int Symp (ESSoS). Berlin: Springer, 2010: 35−43

[57] Venkatraman S, Alazab M. Use of data visualisation for zero-day malware detection[J]. Security and Communication Networks, 2018, 2018: 1−13

[58] Vasan D, Alazab M, Venkatraman S, et al. MTHAEL: Cross-architecture IoT malware detection based on neural network advanced ensemble learning[J]. IEEE Transactions on Computers, 2020, 69(11): 1654−1667

[59] Phu T N, Hoang L H, Toan N N, et al. CFDVex: A novel feature extraction method for detecting cross-architecture IoT malware [C] //Proc of the 10th Int Symp on Information and Communication Technology. New York: ACM, 2019: 248−254

[60] O'Kane P, Sezer S, McLaughlin K, et al. SVM training phase reduction using dataset feature filtering for malware detection[J]. IEEE Transactions on Information Forensics and Security, 2013, 8(3): 500−509

[61] Zhang Jixin, Qin Zheng, Yin Hui, et al. A feature-hybrid malware variants detection using CNN based opcode embedding and BPNN based API embedding[J]. Computers & Security, 2019, 84: 376−392

[62] Lyda R, Hamrock J. Using entropy analysis to find encrypted and packed malware[J]. IEEE Security & Privacy, 2007, 5(2): 40−45

[63] Tien C W, Chen S W, Ban T, et al. Machine learning framework to analyze IoT malware using elf and opcode features[J]. Digital Threats: Research and Practice, 2020, 1(1): 1−19

[64] Nethercote N, Seward J. Valgrind: A framework for heavyweight dynamic binary instrumentation[J]. ACM Sigplan Notices, 2007, 42(6): 89−100

[65] Shoshitaishvili Y, Wang Ruoyu, Salls C, et al. Sok: (State of) the art of war: Offensive techniques in binary analysis[C] //Proc of the 2016 IEEE Symp on Security and Privacy (S&P). Piscataway, NJ: IEEE, 2016: 138−157

[66] Raff E, Barker J, Sylvester J, et al. Malware detection by eating a whole exe[J]. arXiv preprint, arXiv: 1710.09435, 2017

[67] Yakura H, Shinozaki S, Nishimura R, et al. Neural malware analysis with attention mechanism[J]. Computers & Security, 2019, 87: 101592

[68] Wan T L, Ban T, Lee Y T, et al. IoT-malware detection based on byte sequences of executable files[C] //Proc of the 15th Asia Joint Conf on Information Security (AsiaJCIS). Piscataway, NJ: IEEE, 2020: 143−150

[69] Wan T L, Ban T, Cheng S M, et al. Efficient detection and classification of Internet-of-things malware based on byte sequences from executable files[J]. IEEE Open Journal of the Computer Society, 2020, 1: 262−275

[70] Chaganti R, Ravi V, Pham T D. Deep learning based cross architecture Internet of things malware detection and classification[J]. Computers & Security, 2022, 120: 102779

[71] Wu C Y, Ban T, Cheng S M, et al. IoT malware classification based on reinterpreted function-call graphs[J]. Computers & Security, 2023, 125: 103060

[72] Zhao Yang, Kuerban A. MDABP: A novel approach to detect cross-architecture IoT malware based on PaaS[J]. Sensors, 2023, 23(6): 3060

[73] pancake. Radare2 [EB/OL]. [2023-05-29]. https://rada.re/r/

[74] Narayanan A, Chandramohan M, Venkatesan R, et al. Graph2vec: Learning distributed representations of graphs[J]. arXiv preprint, arXiv: 1707.05005, 2017

[75] Li Chuangfeng, Shen Guangming, Sun Wei. Cross-architecture Intenet-of-things malware detection based on graph neural network[C] //Proc of the 2021 Int Joint Conf on Neural Networks (IJCNN). Piscataway, NJ: IEEE, 2021: 1−7

[76] He Tianxiang, Han Chansu, Isawa R, et al. Scalable and fast algorithm for constructing phylogenetic trees with application to IoT malware clustering[J]. IEEE Access, 2023, 11: 8240−8253

[77] Redini N, Machiry A, Wang Ruoyu, et al. Karonte: Detecting insecure multi-binary interactions in embedded firmware[C] //Proc of the 2020 IEEE Symp on Security and Privacy (S&P). Piscataway, NJ: IEEE, 2020: 1544−1561

[78] CAICT. IoT security white paper (2018) [R]. Beijing: China Academy of Information and Communications Technology, 2018: 1−426(in Chinese)
(中国信息通信研究院. 物联网安全白皮书(2018)[R]. 北京：中国信息通信研究院，2018：1−426)

[79] Qiao Yanchen, Zhang Weizhe, Du Xiaojiang, et al. Malware classification based on multilayer perception and Word2Vec for IoT security[J]. ACM Transactions on Internet Technology, 2021, 22(1): 1−22

[80] El-Ghamry A, Gaber T, Mohammed K K, et al. Optimized and efficient image-based IoT malware detection method[J]. Electronics,

2023, 12(3): 708

[81] Yaokumah W, Appati J K, Kumah D. Machine learning methods for detecting Internet-of-things (IoT) malware[J]. International Journal of Cognitive Informatics and Natural Intelligence, 2021, 15(4): 1−18

[82] Lee H, Kim S, Baek D, et al. Robust IoT malware detection and classification using opcode category features on machine learning[J]. IEEE Access, 2023, 11: 18855−18867

[83] Mikolov T, Chen Kai, Corrado G, et al. Efficient estimation of word representations in vector space[J]. arXiv preprint, arXiv: 1301.3781, 2013

[84] Dhanya K, Vinod P, Yerima S Y, et al. Obfuscated malware detection in IoT Android applications using Markov images and CNN[J]. IEEE Systems Journal, 2023, 17(2): 2756−2766

[85] Phu T N, Hoang L, Toan N N, et al. C500-CFG: A novel algorithm to extract control flow-based features for IoT malware detection [C] //Proc of the 19th Int Symp on Communications and Information Technologies (ISCIT). Piscataway, NJ: IEEE, 2019: 568−573

[86] CSA GCR. IoT security key technologies white paper[R]. Hong Kong, CSA GCR, 2023(in Chinese)
(云安全联盟大中华区. 物联网安全关键技术白皮书[R]. 香港: 云安全联盟大中华区, 2023)

[87] Yuan Baoguo, Wang Junfeng, Wu Peng, et al. IoT malware classification based on lightweight convolutional neural networks[J]. IEEE Internet of Things Journal, 2021, 9(5): 3770−3783

[88] Giaretta L, Lekssays A, Carminati B, et al. LiMNet: Early-stage detection of IoT botnets with lightweight memory networks [C] //Proc of the 26th European Symp on Research in Computer Security. Berlin: Springer, 2021: 605−625

[89] Ma Ningning, Zhang Xiangyu, Zheng Haitao, et al. ShuffleNet V2: Practical guidelines for efficient cnn architecture design[C] //Proc of the European Conf on Computer Vision (ECCV). Berlin: Springer, 2018: 116−131

[90] Javed F, Afzal M K, Sharif M, et al. Internet of things (IoT) operating systems support, networking technologies, applications, and challenges: A comparative review[J]. IEEE Communications Surveys & Tutorials, 2018, 20(3): 2062−2100

[91] Pham D P, Marion D, Mastio M, et al. Obfuscation revealed: Leveraging electromagnetic signals for obfuscated malware classification[C] //Proc of the Annual Computer Security Applications Conf. New York: ACM, 2021: 706−719

[92] Ding Fei, Li Hongda, Luo Feng, et al. DeepPower: Non-intrusive and deep learning-based detection of IoT malware using power side channels[C] //Proc of the 15th ACM Asia Conf on Computer and Communications Security. New York: ACM, 2020: 33−46

[93] Azmoodeh A, Dehghantanha A, Conti M, et al. Detecting crypto-ransomware in IoT networks based on energy consumption footprint[J]. Journal of Ambient Intelligence and Humanized Computing, 2018, 9: 1141−1152

[94] Maxfield C. Embedded markets study-Integrating IoT and advanced technology designs, application development & processing environments[R]. Cambridge, MA: AspenCore, 2017

[95] Samantray O P, Tripathy S N. IoT-malware classification model using byte sequences and supervised learning techniques[C] //Proc of Int Conf on the Next Generation of Internet of Things (ICNGIoT 2021). Berlin: Springer, 2021: 51−60

[96] Dib M, Torabi S, Bou-Harb E, et al. A multi-dimensional deep learning framework for IoT malware classification and family attribution[J]. IEEE Transactions on Network and Service Management, 2021, 18(2): 1165−1177

[97] Dovom E M, Azmoodeh A, Dehghantanha A, et al. Fuzzy pattern tree for edge malware detection and categorization in IoT[J]. Journal of Systems Architecture, 2019, 97: 1−7

[98] Alsubaei F S, Alshahrani H M, Tarmissi K, et al. Graph convolutional neural network based malware detection in IoT-cloud environment[J]. Intelligent Automation & Soft Computing, 2023, 36(3): 2897−2914

[99] Atitallah S B, Driss M, Almomani I. A novel detection and multi-classification approach for IoT-malware using random forest voting of fine-tuning convolutional neural networks[J]. Sensors, 2022, 22(11): 4302−4324

[100] Yumlembam R, Issac B, Jacob S M, et al. IoT-based Android malware detection using graph neural network with adversarial defense[J]. IEEE Internet of Things Journal, 2023, 10(10): 8432−8444

[101] Darabian H, Dehghantanha A, Hashemi S, et al. An opcode-based technique for polymorphic Internet of things malware detection[J]. Concurrency and Computation: Practice and Experience, 2020, 32(6): 5173

[102] Naeem H, Alshammari B M, Ullah F. Explainable artificial intelligence-based IoT device malware detection mechanism using image visualization and fine-tuned CNN-based transfer learning model[J]. Computational Intelligence and Neuroscience, 2022, 7: 1−17

[103] Baek S, Jeon J, Jeong B, et al. Two-stage hybrid malware detection using deep learning[J]. Human-Centric Computing and Information Sciences, 2021, 11(27): 10−22967

[104] Alasmary H, Anwar A, Park J, et al. Graph-based comparison of IoT and Android malware[C] //Proc of the 7th Int Conf on CSoNet. Berlin: Springer, 2018: 259−272

[105] 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

[106] He Kaiming, Zhang Xiangyu, Ren Shaoqing, et al. Deep residual learning for image recognition[C] //Proc of the IEEE Conf on Computer Vision and Pattern Recognition. Piscataway, NJ: IEEE, 2016: 770−778

[107] Sandler M, Howard A, Zhu Menglong, et al. MobileNetV2: Inverted residuals and linear bottlenecks[C] //Proc of the IEEE Conf on Computer Vision and Pattern Recognition. Piscataway, NJ: IEEE, 2018: 4510−4520

[108] Huang Gao, Liu Zhuang, Van Der Maaten L, et al. Densely connected convolutional networks[C] //Proc of the IEEE Conf on

Computer Vision and Pattern Recognition. Piscataway, NJ: IEEE, 2017: 4700−4708

[109] Senge R, Hüllermeier E. Top-down induction of fuzzy pattern trees[J]. IEEE Transactions on Fuzzy Systems, 2010, 19(2): 241−252

[110] Tan Mingxing, Le Quoc. EfficientNet: Rethinking model scaling for convolutional neural networks[C] //Proc of the Int Conf on Machine Learning. New York: PMLR, 2019: 6105−6114

[111] Jeon J, Jeong B, Baek S, et al. Hybrid malware detection based on Bi-LSTM and SPP-Net for smart IoT[J]. IEEE Transactions on Industrial Informatics, 2022, 18(7): 4830−4837

[112] He Kaiming, Zhang Xiangyu, Ren Shaoqing, et al. Spatial pyramid pooling in deep convolutional networks for visual recognition[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2015, 37(9): 1904−1916

[113] Cozzi E, Vervier P A, Dell'Amico M, et al. The tangled genealogy of IoT malware[C] //Proc of the Annual Computer Security Applications Conf. New York: ACM, 2020: 1−16

[114] Alzahrani H, Abulkhair M, Alkayal E. A multi-class neural network model for rapid detection of IoT botnet attacks[J]. International Journal of Advanced Computer Science and Applications, 2020, 11(7): 688−696

[115] Puri R. Bots & botnet: An overview[J]. SANS Institute, 2003, 3: 58

[116] Kolias C, Kambourakis G, Stavrou A, et al. DDoS in the IoT: Mirai and other botnets[J]. Computer, 2017, 50(7): 80−84

[117] Bertino E, Islam N. BotNets and Internet of things security[J]. Computer, 2017, 50(2): 76−79

[118] Hallman R, Bryan J, Palavicini G, et al. IoDDoS-the Internet of distributed denial of sevice attacks[C] //Proc of the 2nd Int Conf on Cnternet of Things, Big Data and Security. Setúbal, Portugal: Scitepress, 2017: 47−58

[119] Meidan Y, Bohadana M, Mathov Y, et al. N-BaIoT—network-based detection of IoT botnet attacks using deep autoencoders[J]. IEEE Pervasive Computing, 2018, 17(3): 12−22

[120] Kumar A, Lim T J. EDIMA: Early detection of IoT malware network activity using machine learning techniques[C] //Proc of the 5th World Forum on Internet of Things (WF-IoT). Piscataway, NJ: IEEE, 2019: 289−294

[121] Ozawa S, Ban T, Hashimoto N, et al. A study of IoT malware activities using association rule learning for darknet sensor data[J]. International Journal of Information Security, 2020, 19: 83−92

[122] Alharbi A, Hamid M A, Lahza H. Predicting malicious software in IoT environment based on machine learning and data mining techniques[J]. International Journal of Advanced Computer Science and Applications, 2022, 13(8): 497−506

[123] Waqas M, Kumar K, Laghari A A, et al. Botnet attack detection in Internet of things devices over cloud environment via machine learning[J]. Concurrency and Computation:Practice and Experience, 2022, 34(4): e6662

[124] Uhříček D, Hynek K, Čejka T, et al. BOTA: Explainable IoT malware detection in large networks[J]. IEEE Internet of Things Journal, 2023, 10(10): 8416−8431

[125] Agrawal R, Imieliński T, Swami A. Mining association rules between sets of items in large databases[C] //Proc of the 1993 ACM SIGMOD Int Conf on Management of Data. New York: ACM, 1993: 207−216

[126] Kusupati A, Singh M, Bhatia K, et al. FastGRNN: A fast, accurate, stable and tiny kilobyte sized gated recurrent neural network[J]. Advances in Neural Information Processing Systems, 2018, 31: 9031−9042

[127] Hu Weiwei, Tan Ying. Generating adversarial malware examples for black-box attacks based on GAN[J]. arXiv preprint, arXiv: 1702.05983, 2017

[128] Anderson H S, Kharkar A, Filar B, et al. Learning to evade static PE machine learning malware models via reinforcement learning[J]. arXiv preprint, arXiv: 1801.08917, 2018

[129] Suciu O, Coull S E, Johns J. Exploring adversarial examples in malware detection[C] //Proc of the 2019 IEEE Security and Privacy Workshops (S&PW). Piscataway, NJ: IEEE, 2019: 8−14

Liu Qixu, born in 1984. PhD, professor, PhD supervisor. His main research interests include network attack and defense technology, cyber-attacks discovery, and attribution and forensic.

刘奇旭, 1984 年生. 博士, 教授, 博士生导师. 主要研究方向为网络攻防技术、网络攻击发现、溯源与取证.

Liu Jiaxi, born in 1997. PhD candidate. Her main research interests include malware analysis, network attack and defense technology, and machine learning.

刘嘉熹, 1997 年生. 博士研究生. 主要研究方向为恶意代码分析、网络攻防技术、机器学习.

Jin Ze, born in 1995. PhD. His main research interests include IoT security, and network attack and defense technology.

靳　泽, 1995 年生. 博士. 主要研究方向为物联网安全和网络攻防技术.

Liu Xinyu, born in 1997. PhD candidate. Her main research interests include Web security and Android security.

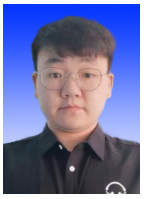
刘心宇, 1997 年生. 博士研究生. 主要研究方向为 Web 安全、Android 安全.

Xiao Juxin, born in 1999. PhD candidate. His main research interests include network attack and defense technology, IoT security, and traffic analysis.

肖聚鑫, 1999 年生. 博士研究生，主要研究方向为网络攻防技术、物联网安全、流量分析.

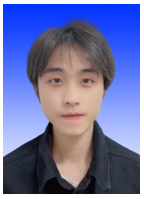
Chen Yanhui, born in 1996. PhD candidate. His main research interests include network attack and defense, malware, and machine learning.

陈艳辉, 1996 年生. 博士研究生. 主要研究方向为网络攻防、恶意软件、机器学习.



Zhu Hongwen, born in 1998. Master candidate. His main research interests include Web security and malware detection.

朱洪文, 1998 年生. 硕士研究生. 主要研究方向为 Web 安全、恶意代码检测.



Tan Yaokang, born in 1998. Master candidate. His main research interests include Web security and program analysis.

谭耀康, 1998 年生. 硕士研究生. 主要研究方向为 Web 安全、程序分析.