面向机器学习模型安全的测试与修复

张笑宇[1,2],沈 超[1,2],蔺琛皓[1,2],李 前[1,2],王 骞[3],李 琦[4,5],管晓宏[1,2]

(1. 西安交通大学电子与信息学部网络空间安全学院,陕西西安 710049; 2. 智能网络与网络安全教育部重点实验室(西安交通大学),陕西西安 710049; 3. 武汉大学国家网络安全学院,湖北武汉 430072; 4. 清华大学网络科学与网络空间研究院,北京 100084; 5. 中关村实验室,北京 100094)

摘 要: 近年来,以机器学习算法为代表的人工智能技术在计算机视觉、自然语言处理、语音识别等领域取得了广泛的应用,各式各样的机器学习模型为人们的生活带来了巨大的便利. 机器学习模型的工作流程可以分为三个阶段. 首先,模型接收人工收集或算法生成的原始数据作为输入,并通过预处理算法(如数据增强和特征提取)对数据进行预处理. 随后,模型定义神经元或层的架构,并通过运算符(例如卷积和池)构建计算图. 最后,模型调用机器学习框架的函数功能实现计算图并执行计算,根据模型神经元的权重计算输入数据的预测结果. 在这个过程中,模型中单个神经元输出的轻微波动可能会导致完全不同的模型输出,从而带来巨大的安全风险. 然而,由于对机器学习模型的固有脆弱性及其黑箱特征行为的理解不足,研究人员很难提前识别或定位这些潜在的安全风险,这为个人生命财产安全乃至国家安全带来了诸多风险和隐患. 研究机器学习模型安全的相关测试与修复方法,对深刻理解模型内部风险与脆弱性、全面保障机器学习系统安全性以及促进人工智能技术的广泛应用有着重要意义. 本文从不同安全测试属性出发,详细介绍了现有的机器学习模型安全测试和修复技术,总结和分析了现有研究中的不足,探讨针对机器学习模型安全的测试与修复的技术进展和未来挑战,为模型的安全应用提供了指导和参考. 本文首先介绍了机器学习模型的结构组成和主要安全测试属性,随后从机器学习模型的三个组成部分即数据、算法和实现,六种模型安全相关测试属性即正确性、鲁棒性、公平性、效率、可解释性和隐私性,分析、归纳和总结了相关的测试与修复方法及技术,并探讨了现有方法的局限. 最后本文讨论和展望了机器学习模型安全的测试与修复方法的主要技术挑战和发展趋势.

关键词: 人工智能安全; 机器学习安全; 机器学习模型测试; 机器学习模型修复; 软件测试; 软件修复

中图分类号: TP391 **文献标识码:** A **文章编号:** 0372-2112(2022)12-2884-35

电子学报 URL:http://www.ejournal.org.cn **DOI:**10.12263/DZXB.20220821

The Testing and Repairing Methods for Machine Learning Model Security

ZHANG Xiao-yu[1,2], SHEN Chao[1,2], LIN Chen-hao[1,2], LI Qian[1,2], WANG Qian[3], LI Qi[4,5], GUAN Xiao-hong[1,2]

(1. *School of Cyber Science and Engineering, Faculty of Electronic and Information Engineering, Xi'an Jiaotong University, Xi'an, Shaanxi* 710049, *China*;
2. *Key Laboratory for Intelligent Networks and Network Security (Xi'an Jiaotong University), Xi'an, Shaanxi* 710049, *China*;
3. *School of Cyber Science and Engineering, Wuhan University, Wuhan, Hubei* 430072, *China*;
4. *Institute for Network Sciences and Cyberspace, Tsinghua University, Beijing* 100084, *China*;
5. *Zhongguancun Laboratory, Beijing* 100094, *China*)

Abstract: In recent years, artificial intelligence technology led by machine learning algorithms has been widely used in many fields, such as computer vision, natural language processing, speech recognition, etc. A variety of machine learning models have greatly facilitated people's lives. The workflow of a machine learning model consists of three stages. First, the model receives the raw data which is collected or generated by the developers as the model input and preprocesses the data through preprocessing algorithms, such as data augmentation and feature extraction. Subsequently, the model defines the architecture of neurons or layers in the model and constructs a computational graph through operators(e.g., convolution and

收稿日期:2022-07-14;修回日期:2022-10-20;责任编辑:朱梅玉

基金项目:科技创新 2030——"新一代人工智能"重大项目(No.2020AAA0107702);国家自然科学基金(No.62161160337, No.U21B2018, No.U20A20177, No.62132011, No.62006181, No.U20B2049);陕西重点研发计划项目(No.2021ZD LGY01-02)

pooling). Finally, the model calls the machine learning framework function to implement the operators and calculates the prediction result of the input data according to the weights of model neurons. In this process, slight fluctuations in the output of individual neurons in the model may lead to an entirely different model output, which can bring huge security risks. However, due to the insufficient understanding of the inherent vulnerability of machine learning models and their black box characteristic behaviors, it is difficult for researchers to identify or locate these potential security risks in advance. This brings many risks and hidden dangers to personal property safety and even national security. There is great significance to studying the testing and repairing methods for machine learning model security, which can help deeply understand the internal risks and vulnerabilities of models, comprehensively guarantee the security of machine learning systems, and widely apply artificial intelligence technology. The existing testing research for the machine learning model security has mainly focused on the correctness, robustness, and other testing properties of the model, and this research has achieved certain results. This paper intends to start from different security testing attributes, introduces the existing machine learning model security testing and repair technology in detail, summarizes and analyzes the deficiencies in the existing research, and discusses the technical progress and challenges of machine learning model security testing and repairing, providing guidance and reference for the safe application of the model. In this paper, we first introduce the structural composition and main testing properties of the machine learning model security. Afterwards, we systematically summarize and analyze the existing work from the three components of the machine learning model—data, algorithm, and implementation, and six model security-related testing properties-correctness, robustness, fairness, efficiency, interpretability, and privacy. We also discuss the effectiveness and limitations of the existing testing and repairing methods. Finally, we discuss several technical challenges and potential development directions of the testing and repairing methods for machine learning model security in the future.

Key words: artificial intelligence security; machine learning security; machine learning model testing; machine learning model repairing; software testing; software repairing

1 引言

近年来,以机器学习算法为代表的人工智能技术不断地发展创新,并在计算机视觉[1~3]、自然语言处理[4~6]、语音识别[7~9]、智能医疗[10~12]等任务上取得了成功. 成熟的人工智能技术已然“走出实验室”[13],进一步惠及人类生产与生活的多个方面. 根据预测,全球人工智能市场总额将在2019年到2026年间从5.69亿美元增长到19.54亿美元[14]. 如今,开发人员可以轻松地借助TensorFlow[15],PaddlePaddle[16],MXNet[17],PyTorch[18]等人工智能框架在各种平台上开发并部署自定义的机器学习模型,并实现不同的功能. 普通用户也可以受益于谷歌[19]、百度[20]等厂商提供的人工智能服务,涵盖图像识别、语音助手、智慧医疗等多个领域. 得益于各类人工智能理论与工具的发展,机器学习算法与模型正在全球范围内被大规模地部署应用.

然而,随着各类人工智能技术的发展与应用,其核心机器学习模型的安全隐患也逐步暴露,引发了人们对机器学习模型可信性的担忧. 2016年,Propublica报道了一起机器学习公平性导致的犯罪预测歧视问题. 相似的犯人仅仅因为人种差异,在机器学习模型驱动的犯罪预测系统中得到了截然不同的风险等级,这最终可能导致恶劣的歧视事件[21]. 2018年美国亚利桑那州发生的Uber无人车事故中,基于机器学习的自动驾驶的无人车未能检测到行人,最终导致行人被撞身亡[22]. 2019年,亚马逊智能音箱Alexa被曝出安全风险,音箱中语音助手发布消极言论并劝告用户轻生,迫使亚马逊紧急调查并进行修复[23]. 这些恶性事件的根本原因在于机器学习模型中存在的安全隐患与漏洞问题. 机器学习模型中个别神经元的输出轻微浮动便可能导致截然不同的输出结果,而结果的变化可能带来巨大安全风险. 然而由于模型黑盒特性,开发者在开发阶段难以确定或定位这些潜在的安全风险.

当下,在诸如自动驾驶系统、智能医疗、恶意信息检测等安全敏感领域的应用中,机器学习模型安全相关的正确性、鲁棒性与公平性等测试属性得到了越来越多的关注. 研究人员针对机器学习模型的安全隐患提出了针对性的测试与修复方法[24~27],旨在预防和降低机器学习模型潜在安全问题造成的严重损失. 此外,多个国家与机构出台了相应的政策与法规以规范机器学习模型的开发与测试. 2017年,中国工业和信息化部发布的《促进新一代人工智能产业发展三年行动计划(2018—2020年)》中提出要开展对AI系统相关的漏洞挖掘、安全测试等安全技术攻关,推动人工智能先进技术的深度应用[28]. 2019年的美国人工智能倡议中也提出“确保技术标准最大限度减少恶意攻击可利用的漏洞”以及“减少对人工智能技术安全测试和部署方面的障碍”等要求,指出要针对机器学习的软件工程、性能、人身安全、可用性等性质建立评估标准体系[29].

由此可见,如何深入研究机器学习模型并设计安全问题相关的测试方法亟待解决并具有重大意义. 虽然现有的研究中提出了各式各样的机器学习模型安全的测试与修复方法,但是由于机器学习模型安全相关

的测试属性众多,测试与修复的方法、形式多样且各有侧重,尚未形成完整的技术体系;此外,现有研究对机器学习模型的内在脆弱性以及本身的黑盒特性理解尚不充分,限制了这些测试与修复方法的有效性. 因此,围绕机器学习模型安全的测试与修复工作,亟须对现有的研究进行科学的归纳、分析及讨论,以发现现有研究中的不足并为后续从事相关领域的研究人员提供方向性的指导.

现有的机器学习模型安全技术相关文献综述多围绕机器学习系统的通用安全问题与对应技术展开介绍,而缺乏针对机器学习模型测试与对应的问题修复研究的系统性详细介绍. 如文献[30]总结了机器学习模型的安全与隐私问题,并总结了模型对抗鲁棒性与隐私攻防相关研究工作,但未涉及机器学习模型正确性等特性的测试与修复相关内容;文献[31]总结了深度学习模型(Deep Neural Network,DNN)部署与应用中的安全性与可信性方面的研究,涵盖了验证、测试、对抗性攻击和防御以及可解释性,但是并未涉及模型的修复方面的工作. 虽然针对机器学习模型测试有少量的文献综述,但现有文章多集中于对机器学习系统的测试方法的总结,鲜有文献针对测试中暴露的漏洞与问题如何修复进行全面的介绍[32,33]. 本文专注于机器学习模型工作流程中全面的安全特性测试方法与对应的修复研究,从数据、算法与实现三个阶段,对现有的测试与修复技术展开系统的介绍,并对现有研究中的不足之处进行总结分析,旨在推动机器学习模型安全的测试与修复技术的进一步发展,并为保障机器学习模型与相关技术的安全应用提供指导和参考. 表1对相关领域综述进行了对比.

本文针对机器学习模型安全的测试与修复的研究进展进行梳理、归纳、分析及讨论. 第2节对机器学习模型的流程框架进行描述,并对模型安全的测试属性进行整理和概述. 第3节、第4节和第5节分别从机器学习模型的数据、算法、实现三方面入手,对相关的测试与修复相关研究进行归纳总结,并讨论现有研究的效果与局限. 第6节对机器学习模型安全的测试与修复技术面临的挑战及未来的研究方向进行讨论和展望. 第7节对文章进行总结.

表1 机器学习模型安全测试相关综述对比

机器学习模型安全测试相关综述	参考文献最新年份	涵盖方法内容											
		鲁棒性		正确性		公平性		效率		可解释性		隐私性	
		测试	修复	测试	修复	测试	修复	测试	修复	测试	修复	测试	修复
文献[33]	2019	—	—	√	√	—	—	—	—	—	—	—	—
文献[34]	2019	—	—	—	—	√	√	—	—	—	—	—	—
文献[30]	2019	√	√	—	—	—	—	—	—	—	—	√	√
文献[31]	2020	√	√	√	√	—	—	—	—	—	—	—	—
文献[32]	2020	√	—	√	—	√	—	√	—	√	—	√	—
本文	2022	√	√	√	√	√	√	√	√	√	√	√	√

2 机器学习模型与安全测试属性

对机器学习模型安全的测试与修复工作进行研究,首先要探明机器学习模型的结构,明确模型安全的相关测试属性. 本节主要从数据、算法、实现三个部分分析了机器学习模型的流程框架,并对机器学习模型安全相关的六种主要测试属性进行了介绍,包括正确性、鲁棒性、公平性、效率、可解释性和隐私性,最后对比了机器学习模型安全测试工作与传统软件安全测试的区别.

2.1 机器学习模型结构

在传统软件测试工作中,研究人员往往根据被测对象的组织结构进行分解,以在测试前明确可能存在问题的各个组件,并针对性地设计测试样例. 传统软件在测试中一般可以拆分为三部分:数据、算法、实现[35]. 机器学习模型的结构也可以依此分为三部分[32],并针对不同的组件展开对应的测试与修复工作. 图1展示了这三者的流程关系.

(1)数据. 机器学习模型的输入数据一般由人工收集或者生成,存在一定的扰动与错误. 因此模型的数据部分需要整理原始输入数据、清理异常数据并执行规范化、数据增强、特征提取等预处理算法[36~38],旨在从接收到的原始输入中去除噪声、错误数据,并提取数据特征.

(2)算法. 机器学习模型的算法并非如传统程序那样由人工直接设计输入输出的工作逻辑,而是通过构造模型的算子、设计模型架构、建立神经网络计算图等方法对算法流程进行规划,引导模型在给定数据上完成聚类、分类、回归等学习任务.

(3)实现. 基于预处理数据和模型计算图,模型利用机器学习框架等计算库对模型各个算子的计算功能进行实现,并在训练迭代中更新模型算子的权重,从而完成给定的机器学习任务.

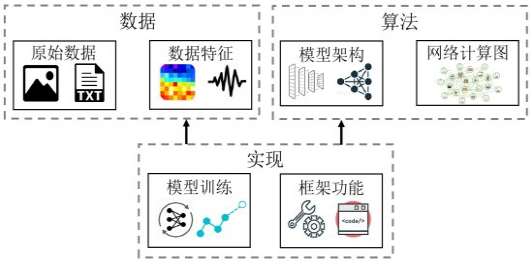


图1　机器学习模型流程框图

机器学习模型一般直接包含数据处理部分和算法程序部分,而机器学习模型的实现通常由机器学习框架隐式地提供,并为另外两个部分提供支撑. 目前针对机器学习模型的测试工作主要针对模型的数据和算法两个部分展开,并根据不同的安全测试属性提出相应的修复策略. 相对而言,现有的机器学习模型实现部分的研究工作较为有限. 在实现部分中机器学习模型会被编译并根据底层库(例如NVIDIA cuda,TensorFlow XLA等)的设置进行具体的优化. 目前框架实现环节研究以测试为主,也有研究人员提出了一些漏洞和问题的定位方法,辅助开发者进行修复,但该环节的修复工作仍处于探索阶段,需要人工对实现问题进行修正,目前没有成体系的修复方法或者技术.

2.2　机器学习模型安全相关测试属性

机器学习模型的测试特性定义了一个功能良好的、安全的、高效的机器学习模型需要具备什么样的属性. 各种各样的测试特性是模型测试与修复工作的目标. 根据现有的测试研究,本文总结了如下六种主要的机器学习模型测试特性,表2对这些特性进行了展示.

表2　机器学习模型安全测试属性总结

测试属性	测试阶段	特性描述
正确性	数据、算法、实现	模型正确行使功能并完成任务的能力
鲁棒性	数据	模型在输入干扰下正确运行的能力
公平性	数据、算法	模型不受敏感输入属性的影响的能力
效率	实现	模型执行完成指定任务的开销
可解释性	算法	模型的决策可以被观察者理解的能力
隐私性	数据、算法	模型保护相关私密数据的能力

(1)正确性(correctness). 该特性衡量模型正确行使其功能、完成给定任务的能力. 在实际场景中,模型应在给定数据集,以及任务相关的未知数据上具有可接受的正确性. 由于未知数据无法在模型的训练与评估中获取,一般实践中会将给定数据集划分为训练数据、验证数据、测试数据,通过后者模拟未知数据并使用交叉验证的方法评估模型正确性[39]. 机器学习模型的正确性问题会导致模型在特定输入下得到错误的判断结果,从而对使用者的财产安全、隐私乃至人身安全产生危害.

(2)鲁棒性(robustness). IEEE软件工程术语表[40,41]中将鲁棒性定义为:“在存在无效输入或者压力环境条件的情况下,系统与组件能够正确运行的程度”. 参考该定义,鲁棒性可衡量模型抵抗输入或者环境中的扰动并获取不受干扰影响的输出的能力,即在存在扰动的情况下,模型维持正确性的能力. 机器学习模型的鲁棒性问题会导致模型在环境扰动影响下得到错误的、甚至特定的结果,危害使用者的安全. 值得注意的是,尽管鲁棒性问题和正确性问题都会导致模型得到错误输出,但是两者本质上衡量的是模型的不同能力. 正确性衡量模型对未知数据的预测性能,即对于给定未知输入,模型能够正常预测或分类的能力;而鲁棒性更多衡量模型的预测输出不受外界干扰的能力,即模型在具有轻微扰动的原始输入上的预测结果不会显著偏离该原始输入. 测试并修复机器学习模型的鲁棒性问题可以确保模型功能不易受到环境或输入中扰动的影响.

当前模型的测试与修复研究中,对抗鲁棒性是评估模型鲁棒性的重要子属性. 对抗鲁棒性通常指代指模型在输入样本存在细微对抗扰动的情况下,模型预测输出不受对抗样本干扰或误导的能力[42].

(3)公平性(fairness). 当前研究中公平性的定义多样且不统一. 文献[43]提出了无意识公平性(fairness through unawareness)的定义,该定义衡量模型在决策中是否明确使用受保护的属性. 文献[44,45]提出了群体公平(group fairness)的概念,即基于敏感属性分类的群组有着相同的决策结果概率,敏感属性本身并不影响决策的结果. 此外,还有研究提出了反事实公平性(counter-factual fairness)[46]、个体公平性(individual fairness)[47]等概念. 不同类的数据在模型预测中准确率是否存在偏见也可以在一定程度上作为公平性的衡量标准[34]. 总体而言,公平性衡量某些受保护且与模型任务无关的属性(例如信用评级系统中的性别、种族、地理位置等属性)对机器学习模型的影响程度. 机器学习模型在任务中越不容易被这些受保护属性影响,模型就具有越优秀的公平性. 机器学习模型的公平性问题会导致模型返回存在歧视与偏见的预测结果,并损害具有特定属性群体的权益.

(4)效率(efficiency). 该特性评估模型在执行给定任务时所需的时间、内存、GPU等衡量模型性能指标的开销. 开销越大的模型,对设备的要求越高、实际部署的限制越大,效率也就越低. 现有工作通过对比相同模型在不同部署条件下的性能差异,研究相关的效率漏洞问题,并取得了一定的成果. 例如,文献[48]发现相同的LeNet模型在不同部署平台上执行分类功能时有显著的性能以及时间开销差异. 这类机器学习模型效

率问题会导致模型在特定环境下工作时带来过大的开销,从而导致计算资源和时间的浪费.

(5)可解释性(interpretability). 该特性评估观察者能够理解机器学习模型做出决策的原因的程度. 在自动驾驶、智能医疗等领域,机器学习模型往往需要协助专家做出重要判断. 然而模型中任何判断结果的偏差与异常,可能对他人的财产安全乃至人身安全产生严重威胁,因此《欧盟通用数据保护条例》[49]等法规规定,当机器学习模型做出判别时,用户有合法权利要求算法提供合理解释,从而避免歧视问题. 机器学习模型的可解释性问题会导致模型可能无法给出令人信服且公正的决策理由,从而降低模型结果的有效性与实际价值.

(6)隐私性(privacy). 该特性评估机器学习模型保护私密数据信息的能力. 文献[50]提出了差分隐私(differential privacy)的概念,旨在评估单个个体数据是否对模型结果产生重大影响. 机器学习模型的隐私性问题会导致模型相关的私密数据、受保护的信息甚至模型算法被攻击者非法窃取,从而危害用户的隐私安全、财产安全等.

2.3 机器学习模型安全测试与传统软件安全测试

面向传统软件安全的测试与面向机器学习模型安全的测试有着显著的区别,主要体现在测试对象、测试属性以及测试预言(test oracle)三方面.

(1)测试对象. 传统软件测试的测试对象一般为一段程序代码,代码按照人工设计的算法手动或者自动地生成,并可以通过编译在特定输入下实现对应的功能. 而机器学习模型测试的测试对象一般为数据、算法以及实现三部分共同组成的模型文件. 和传统软件测试不同,机器学习模型的算法程序并非经过良好的人工设计,而需要在数据集上经过充分的训练后才能够指导模型正确地完成任务. 因此在机器学习模型的测试中,需要对数据、算法、实现三部分进行测试,而传统软件测试主要测试代码的实现.

(2)测试属性. 机器学习模型具备正确性、鲁棒性、公平性、效率、可解释性、隐私性等多种安全相关的测试属性,在测试中通过测试并评估这些特性,研究人员可以找到模型设计与实现的缺陷并加以修复,从而改善模型的安全性. 传统软件测试中,主要测试软件的正确性与效率. 由于传统软件的行为是直接由代码确定的,不同于机器学习模型需要经过训练确定具体行为,因此传统软件不容易出现潜在的公平性、可解释性等问题,这类测试属性一般不作为传统软件安全测试的重点关注对象.

(3)测试预言. 传统软件测试中,研究人员可以根据代码实现的逻辑提前构造测试预言并与代码的执行结果进行验证. 而机器学习模型在模型训练与框架实现中存在大量的随机操作,因此难以提前预测给定输入的输出结果. 现有的测试工作中,一般通过同一功能的不同实现交叉验证的方法[51,52]或者构造等效关系(例如等效图[53,54]、蜕变测试[55,56]等)作为测试预言. 由于难以获得可靠的预言,在机器学习模型测试中往往存在较多的误报、漏报现象. 对于测试的结果,研究人员也往往需要人工检查机器学习模型或者框架实现来进行确认.

本文将基于上文建立的机器学习模型流程框图,从数据、算法以及实现三个部分,系统地总结并阐述当前模型安全的测试与修复方面的研究工作的特点与关注的测试属性,并讨论其中的局限与未来研究的方向.

3 模型数据测试与修复技术

机器学习模型的输入数据可能人工或者自动化地从网络、传感器等来源收集. 这些在一定程度上反映了真实物理世界的图像、音频、文字等数据往往包含各种干扰. 这些数据中的错误与扰动会严重影响模型的预测效果与工作质量,使模型无法正确完成给定任务,甚至会给模型带来安全隐患. 在本节中,本文主要从鲁棒性、公平性、正确性以及隐私性四种安全测试属性方面,介绍现有针对机器学习模型数据处理环节的主要测试工作,并阐述对应的测试属性修复研究. 表3展示了模型数据测试与修复的几种典型技术,其中测试与修复方法如果通用性不强或者存在明显局限,则在“效果”一栏标记为“弱”,反之标记为“强”.

3.1 数据鲁棒性测试与修复

3.1.1 数据鲁棒性测试研究

2013年,Szegedy等人[81]针对机器学习模型的输入数据设计了对抗攻击方法L-BFGS,通过在图片中植入肉眼不可察微小扰动,引导机器学习模型分类错误. 在此之后,对抗攻击这一鲁棒性安全问题得到了学术界与工业界越来越多的关注. 尽管研究者们对对抗样本这一鲁棒性问题进行了深入调研,但是对其具体形成机理依然没有统一的共识. Szegedy等人[81]认为对抗样本存在的原因之一是模型输入与输出映射的不连续性,以及模型训练中存在的过拟合. Goodfellow等[57]则认为对抗样例本质上是高维线性空间中数据的微小变化会导致的输出极大变化.

现有研究中,机器学习模型鲁棒性的测试工作主要分为两种. 一种是基于对抗输入的方法,首先构造存在细微扰动的样本作为模型输入,随后对模型输出结果进行观察,从而间接评估模型的鲁棒性[57,82];另一种是通过模型在数据上的输入输出状态,分析模型算法的安全数据区间甚至量化评估[83,84]. 本节主要对基于

表3　模型数据测试与修复典型技术对比总结

功能描述	方法类别	应用领域	方法描述	效果	相关工作
数据鲁棒性测评	对抗输入生成	图像、文本、音频	生成对抗样本直接测试模型	弱	文献[57]等
		图像、文本、音频	构建对抗输入生成库测试模型鲁棒性	强	文献[58]等
数据鲁棒性修复	随机化	图像	随机化变换调整输入数据	弱	文献[59]
		图像/数值数据	利用张量衰减调整模型内数据特征	强	文献[60]
	去噪	图像	压缩图像对输入数据进行去噪	弱	文献[61]
		图像	利用特征压缩的方法对数据去噪	强	文献[62]
	对抗输入检测	图像	基于模型变异检测对变异敏感的对抗样本	弱	文献[63]
		图像	评估数据的鲁棒性来区分对抗样本	弱	文献[64]
数据公平性测评	数据偏差测试	数值数据	无监督聚类采样检测数据的类不平衡	弱	文献[65]
		图像	使用自动编码器学习数据特征并检测偏差	强	文献[66]
		数值数据	检测数据分布与特征的倾斜问题	弱	文献[67]
数据公平性修复	数据集修正	主要为数值数据	修复数据集标签或内容	强	文献[68]等
	良性数据生成	图像	生成非歧视性数据以解决训练数据不均衡	强	文献[69]
		文本	构造良性数据集训练或微调模型	强	文献[70]
	修复框架与工具	数值数据	自动化诊断与修复框架	弱	文献[71]
数据正确性测评	异常数据检测工具	数值数据	检查数据示例并识别特定模式的潜在问题	弱	文献[72]
		主要为数值数据	自动化异常数据检测方法搜索框架	强	文献[73]
		图像	分析特征空间以识别异常数据并进行过滤	弱	文献[74]
数据正确性修复	数据清理工具	图像	基于自动编码器对存在噪声数据进行清理	强	文献[75]
		图像、文本	加入数据检测以在模型计算前剔除异常值	强	文献[76]
		主要为数值数据	自动化搜索数据清理方法并清理异常数据	强	文献[73]
数据隐私性测评	私密信息窃取	主要为数值数据	构造私密数据窃取攻击以测试模型隐私性	弱	文献[77]等
数据隐私性修复	基于差分隐私的数据隐私保护	图像	训练多个教师模型并聚合预测结果	强	文献[78]
	基于安全多方计算的数据隐私保护	图像、数值数据	基于安全多方计算协议交互私密数据	强	文献[79]等
	基于联邦学习的数据隐私保护	图像、数值数据	通过安全聚合等方法构建联邦学习训练模型	强	文献[80]等

构造对抗输入的数据相关鲁棒性测试与评估方法进行了总结,对模型算法进行分析的鲁棒性测试方法则在后文的算法鲁棒性测试部分进行概述.

对抗输入生成. 为进一步评估模型的鲁棒性上界,研究人员提出了基于不同目标函数、优化方法的对抗性输入生成方法,该方法通过构造对抗样本作为输入数据,测试模型抵抗对抗扰动并保持正确输出的能力. 现有的工作在计算机视觉[57,79,81,82,85,86]、自然语言处理[87~89]、音频处理[90~92]等多个领域设计了各式各样的对抗样本生成方法,通过构造并生成扰动以测试模型的对抗鲁棒性. 2015年,Goodfellow等人[57]率先提出了一种基于梯度的白盒对抗样本生成方法:快速梯度符号法(Fast Gradient Sign Method,FGSM),通过沿着梯度反方向加入扰动以拉大对抗样本与原始样本的距离,从而快速生成对抗样本. 然而该方法攻击成功率有限,尤其在定向攻击特定类别时成功率较低. 除了基于梯度的方法外,研究者们也在寻找其他生产对抗输入的方法. Brown等人[93]提出了"对抗补丁"生成算法,通过加入一个肉眼可见的图片补丁作为扰动,使得该扰动加到任何图片上都可以让原图被识别为特定类别,并在实验中进一步探索了VGG,Xception等流行模型的鲁棒性. Carlini等人[82]则改进了Szegedy等人的提出的L-BFGS方法,设计了攻击效果更优秀的目标函数,并提出了距离度量量化相似度的对抗性示例生成方法C&W算法,可以有效地突破常用的模型对抗鲁棒性防御技术,从而测试模型对抗鲁棒性的上界,在之后工作中被广泛地用于测试模型鲁棒性与对抗防御性能.

基于已有的对抗样本生成技术,研究人员构建了多种对抗鲁棒性测试库. 2016年,Papernot等人[58]和Goodfellow等人[94]构建了Cleverhans对抗样本生成库,辅助从业人员批量生成基于各种技术的对抗样本,并方便地测试目标系统面对各种对抗输入的鲁棒性. 之后,Rauber等人[95]开发了Foolbox,该工具可以生成各种对抗性扰动,并量化和比较机器学习模型的鲁棒性. 目前Foolbox提供了四十余种已发布的对抗性攻击方法的实现,并支持Keras,PyTorch等主流机器学习框架的接口,然而该工作主要专注于实现对抗性攻击方法,而没有提供相应的修复与防御方法. Nicolae等人[96]进一步设计并实现了ART工具箱,该工具箱提供了较为全面的构建对抗扰动和部署对抗防御的工具,并通过对抗

性样例测试其鲁棒性. 对抗性输入生成相关的工作旨在研究对抗样本构造方法,主要关注攻击效果而非鲁棒性的测试评估结果,纪守领等人[30]和任奎等人[97]对这些对抗样本的构造方法已进行较为全面的总结.

小结. 现有的鲁棒性测试工作更多倾向于设计对抗样本生成算法测试机器学习模型或者系统的鲁棒性上界或者直接设计对抗样本并测试在目标模型上的攻击成功率,并通过对比来说明模型之间鲁棒性的优劣,而非给出具体且量化的指标进行评价. 如何设计令人信服且高效的鲁棒性评价指标将会是未来重要的研究方向之一.

3.1.2 数据鲁棒性修复研究

为解决机器学习模型的鲁棒性问题、缓解输入中对抗扰动对模型的影响,研究者们提出了各式各样的对抗防御与鲁棒性提升方法,在此主要阐述从数据环节入手,提升并修复模型鲁棒性的工作. 数据鲁棒性的修复方法主要包含三种,分别是随机化、去噪以及对抗输入检测. 前两者主要通过对输入数据加入噪声或者降噪以降低对抗扰动对模型的干扰;第三主要通过检测并筛选对抗输入的方法,避免该类型输入危害模型的鲁棒性.

随机化. 输入样本的随机化方法通过随机变换输入数据或者对数据中加入随机噪声以降低对抗扰动对模型鲁棒性的影响. Xie等人[59]率先提出在模型前向传播过程中,通过随机调整模型输入大小以及随机填充数据的方法,减轻对抗干扰对模型的影响. 事实上,引入随机性可以在一定程度上提升模型的鲁棒性,该观点在后续的研究中得到了证实[98]. 基于这种思想,Guo等人[99]进一步提出对模型的输入图片进行随机的图像变换以及图片压缩(例如JPEG压缩、图像拼接等),以降低对抗样本的危害. Liu等人[98]提出对输入加入随机噪声以减弱对抗扰动对模型分类的影响,该方法在神经网络中添加了随机噪声层以削弱基于梯度构建的对抗扰动所造成的危害,该方法几乎没有任何额外的内存开销. 此外他们所提出的基于噪声随机梯度下降的训练过程可以确保模型依然具有良好的预测能力. Luo等人[100]进一步提出了一种随机模型特征的技术,通过随机掩盖模型层的输出特征,避免模型受到对抗扰动的不良影响. 近年,Kolbeinsson等人[60]深入模型内部的数据传递,提出了一种模型张量数据随机化技术,利用张量衰减(tensor dropout)的方法降低对抗输入对模型的影响,从而提升模型的鲁棒性和泛化性能.

去噪. 样本的去噪工作可以通过输入整流、清理等方法有效地减轻对抗扰动的效果. Xu等人[61,62]首先提出利用图像的压缩方法(例如压缩图像颜色得位深度)对模型内特征进行压缩与降噪以去除原始图像中潜在的噪声,他们的方法旨在压缩原本广阔的特征空间,从而降低输入中的对抗性扰动对输出结果的影响,该方法一定程度上减轻了对抗扰动的效果,提升了模型的鲁棒性. 然而文献[101]表明白盒攻击算法依然可以破解这类修复方法. 随着生成对抗网络GAN被广泛地应用在数据清理工作中,Samangouei等人[102]提出使用GAN清理对抗输出的扰动,其核心思想在于利用生成模型对良性输入的分布进行建模并学习其输入特征,然后重构待预测样本进而得到近似的干净样本,避免对抗扰动的影响,然而生成模型本身的训练和学习能力限制了该方法的应用,导致对未知输入的重构与预测效果有限. 除此之外,Liao等人[103]还提出从特征层面进行去噪的方法,该方法通过损失函数定义良性输入与对抗输入在模型中输出的差异,并基于该函数在训练中最小化对抗扰动对模型输出的影响. Shen等人[104]提出了APE-GAN的方法,将输入的对抗扰动作为GAN的输入,输出良性样本,在实验中取得了良好的效果性能. 近些年Yang等人[105]进一步提出了APE-GAN++方法消除输入数据中的对抗干扰,并在实验中取得比其他对抗修复方法更好的效果. Kherchouche等人[106]设计了一种基于块匹配3D滤波器的去噪器,有效地过滤模型输入中的对抗扰动,进而提升模型对抗鲁棒性. Esmaeilpour等人[107]基于GAN设计了一种语言对抗样本的去噪方法,根据合成的频谱图从给定的输入信号中重建不包含对抗扰动的一维信号,且不引入额外的噪声,有效地提升了语音系统的鲁棒性.

对抗输入检测. 研究者们还提出了一些对抗样本检测方法,以从数据中筛选出不受对抗扰动影响的良性样本,从而保障模型输入数据的安全. Metzen等人[108]使用一个经过良好训练的简单子网络扩充机器学习模型,用于区分对抗性样例和良性数据,并取得优秀的检测效果,然而该方法依然容易被训练中未遇到的对抗样本类型欺骗,将对抗性样例误判为良性数据. Wang等人[63]则发现对抗样本比良性样本对模型中的随机突变更敏感,他们设计了一种新颖的检测方法并开源了对应的工具包,该方法基于模型变异检测对抗样本,当模型神经元轻微改变时,对抗性样本的预测结果更容易产生变化,当样本变化的概率过高时,则判断为对抗样本并拒绝该样本的输入. 近些年,Zhao等人[64]基于对抗样本自身鲁棒性差的观察,提出了一种对抗样本检测方法,通过评估输入数据的鲁棒性来区分良性样本与对抗样本. 他们发现,良性样本往往具有更好的鲁棒性且离决策边界更远,而对抗样本相比之下则更靠近决策边界,分类结果也更不稳定. 此外,Gopinath等人[84]设计了DeepSafe以评估机器学习模型的鲁棒性,他们的方法可以识别并证明输入空间中具

有鲁棒性的安全区域. 他们的结论可以辅助开发者对模型输入进行过滤与选择,从而保障模型鲁棒性不受对抗样本数据的损害.

小结. 现有修复工作主要通过输入随机化、去噪以及对抗样本检测等方法修复机器学习模型的鲁棒性问题,然而依然存在局限. 输入随机化和去噪方法容易被具有较强对抗攻击能力的白盒自适应攻击[82]破解. 对抗样本的检测方法则依赖生成对抗扰动的损失函数[109],攻击者可以通过构造新的损失函数绕开检测器. 如何有效地提升模型鲁棒性,改善并处理模型的输入数据从而降低对抗扰动对模型的危害,依然是一个有待解决的难题.

3.2　数据公平性测试与修复

3.2.1　数据公平性测试研究

当前公平性测试工作主要体现在两种方面:一方面是对算法中的公平性与歧视问题进行测试,评估模型算法输出是否容易受到输入的受保护属性变化的影响,并针对相应的公平性问题设计修复方法、重训练模型等;另一方面则对数据中的公平性问题进行评估,检测并消除数据中的偏差(例如类的不均匀等),避免基于有偏差的数据训练出不公平、不准确的模型. 基于数据的公平性测试主要针对后者的数据偏差.

数据偏差测试. 研究者们致力于检测数据集中由测量偏差、采样偏差等导致的数据不均衡问题[34],确保训练数据、测试数据表现出一致的特征和分布,且敏感属性不会对分类结果产生影响. Nguyen等人[65]提出了用聚类的方法识别数据集中的类不均衡问题,然而这种聚类检测方法难以拓展到图片等高维数据. Amini等人[66]进一步设计了一种在训练过程中检测数据的分类偏差并进行自适应调整训练数据的方法. 这种方法可以在学习训练任务的同时学习训练数据的潜在结构,并发掘数据中隐藏的偏差问题和样例,并修复该偏差对模型质量的影响. 随着近些年相关数据偏差检测工作越发成熟,一些测试准则和检测工具被研究者们提出. Mullick等人[67]评估了类不均衡对二值分类器和多分类器的评价指标的影响,设计了在数据偏差环境下的指标改进和规范化方法,并通过在ImageNet数据集子集上的实证研究验证了方法的有效性. Breck等人[110]研究了模型的训练数据和实际部署的数据之间的偏差,并设计工具验证了其中的特征倾斜、分布倾斜问题,这类问题可能会在训练中影响模型的质量与类的公平性.

小结. 基于数据的公平性测试工作目前主要关注数据中的分类偏差、敏感属性(例如种族、性别)等,以及其对模型训练准确率与公平性的影响. 现有工作已经提出了一些的测试准则,并基于聚类以及数据特征学习等方法[65,66]实现了偏差检测. 目前的工作主要关注类不均衡这类偏差在数据集中的直观表现,随着数据集的特征以及偏差的标记工作[111,112]逐渐成熟,未来的测试研究趋势可能会深入到检测数据偏差的具体来源与类别,例如数据的采样偏差、代表性偏差、测量偏差等[34]以及它们对模型的影响.

3.2.2　数据公平性修复研究

公平性修复工作可以根据这些方法执行所处的阶段大致分为三种[113],即预处理(pre-processing)(在模型预测之前修复问题)、处理中(in-processing)(在模型训练与预测过程中修复问题)和后处理(post-processing)(在模型完成预测后修复问题). 其中基于数据的公平性修复工作一般都是属于预处理修复. 目前对数据公平性与偏差的研究中,一些研究者设计了检测方法的同时也提供了修复偏差数据并缓解模型偏差的改进方法[66,67,110].

数据集修正. 早期的研究中,Kamiran等人[68]提出修改数据集的方法来获得无偏数据集,从而避免数据中的偏差问题. 此外,Kamiran等人[114]对通过更改数据集类标签、在重新加权或重新采样数据等预处理方法消除歧视问题进行了研究,并在实际数据中进行了实验. 然而,该类方法均成本较高. 近年,Bender等人[112]针对nlp系统提出了“数据陈述”(data statement)的概念,例如语言多样性特征、讲述人和注释人的人口统计特征、语言的情景特征等作为数据集的特征,以提供上下文并辅助开发者更好地理解数据集潜在的偏差以及对构建系统可能的影响. 事实上,随着数据集规模的提升,修正数据集所需要的时间和人工成本越来越大,因此近年主流的修复方法更多以数据生成为主.

良性数据生成. 目前研究人员更多使用生成非歧视性数据扩充并改善数据集的方法修复数据偏差问题. Amini等人[69]通过在模型学习中捕捉数据特征的潜变量,生成非歧视性数据以解决训练数据不均衡的问题,缓解了自动驾驶系统的数据偏差对模型的影响. Sattigeri等人[115]借助生成对抗网络GAN设计了一种数据集重构方法,可以通过生成与原始数据集相似的数据,改善类与类之间的数据平衡等公平性问题. Aïvodji等人[116]提出了一种对抗性训练方法,通过消除敏感属性本身以及与其他属性的相关性,缓解敏感属性引发模型的公平性与偏差问题. 除此之外,Tomalin等人[70]还针对现有神经机器翻译(Neural Machine Translation, NMT)中数据去偏差方法存在的局限进行了研究,并提议在该领域中使用域适配(domain adaptation)技术,在不降低模型预测准确率的情况下削弱偏差对模型公平性的影响.

修复框架与工具. Holland等人[71]提出了一个数据

集诊断与分析框架,可以在模型训练之前概述数据集的变量相关性、数据分布等统计学信息,通过定性与定量的方法为数据集添加可以跨域和数据类型应用的标签,从而提供高效的数据查询与更完善的模型质量保证,降低数据偏差对模型的影响.

小结. 现有机器学习模型数据偏差与公平性的修复工作有着较为成熟的发展,近年的相关工作侧重于利用机器学习的聚类、分类等方法,构造模型学习数据偏差特征并生成相似数据,修复数据集的偏差与不平衡问题. 关于更细致的敏感属性(例如种族、性别等)的公平性问题的修复与提升工作目前也有了一定的研究[116,117]. 如何针对敏感属性的数据公平性问题设计一套高效、便捷、通用的检测与修复系统或工具可能会成为后续研究的挑战之一.

3.3 数据正确性测试与修复

3.3.1 数据正确性测试研究

正确性问题对模型性能与质量有着重大的影响. 数据编码错误和异常值等数据正确性问题会使模型无法正确执行给定任务. 目前研究者已提出多种检测异常数据的工具与方法.

异常数据检测工具. 为检测模型异常数据、确保模型输入正确性,2016年,Krishnan等人[118]设计了数据清理平台Activeclean,允许在模型训练中自动清理特定类型的异常数据并协助模型收敛,该平台根据数据对模型的价值以及数据存在异常的可能性,建议清理部分数据样本. 之后在2017年,Krishnan等人[119]设计了BoostClean系统,以检测并修复数据集中的域值冲突问题(例如属性值超出许可范围),在实验中有效地提升了模型的准确率,改善了模型正确性. 尽管这些工具具有较好的异常数据检测与清理的效果,但依然需要人工的操作. 研究人员也在探索自动化检测数据正确性问题的方法. 同样在2017年,Hynes等人[72]提出了轻量级工具Data Linter,以自动化地检测数据集中潜在的异常数据、错误编码等问题. 之后,在2019年研究者们提出的Alpha Clean工具[73]提供了更加丰富的异常数据检测功能,它使用贪婪树搜索算法自动调整数据清理管道的参数,并取得了优秀的异常数据检测与清理效果. 近些年,数据正确性的检测与验证工具更加贴近机器学习的使用场景与管道. Breck等人[110]提出了一种测试模型数据正确性的验证系统,并在TFX机器学习平台作为数据验证功能部署. 此外,Song等人[120]提出了一种语料库驱动的方法,通过推断合适的数据验证模式来自动化验证模型数据的正确性,减少了人工干预和误报率. Steindhardt等人[76]通过在输入模型前增加数据检测机制以发掘并剔除数据的异常值. Laishram等人[74]提出一种名为Curie的轻量级数据清理方式,可以识别并过滤数据集中认为添加的恶意数据与异常数据,有效地降低了机器学习模型的误报率. Rubinstein等人[121]构建了一个基于主成分分析的污染数据检测模型,以此限制异常训练数据对AI决策边界的影响.

小结. 现有的数据正确性测试工作较为成熟,相关工具可以自动化地完成包括错误编码在内的多种异常数据识别、测试以及修复,过程中几乎不需要人工参与.

3.3.2 数据正确性修复研究

数据清理是维护机器学习模型数据的重要方法. 常见的方法包括规范化数据(例如,更正拼写错误)、删除具有非法值的数据(例如,删除NaN值)、识别异常值(例如,检测编码错误)以及自动更正有问题的样本(例如,修复标签错误的样本)等. 在数据科学领域中,数据清理相关工作已有成熟的研究,Rahm等人[122]的工作对此进行了总结,虽然这些工作并非专门针对机器学习模型的数据,但同样可以用于修复、清理机器学习数据集.

数据清理工具. 目前大部分机器学习数据正确性测试的工具都可以在测试流程中一并完成数据的清理与修复工作[73,74,76,118,119]. 近些年,随着自动编码器的流行,它可以重构输入并在隐藏层中学习到输入的良好表征的特性被应用在数据清理任务中. Zhang等人[75]提出使用自动编码器对标签存在噪声的数据进行分类与清理,自动编码器可以学习类特定特征,进而发现存在噪声的异常数据并实现噪声数据分类.

小结. 机器学习模型数据正确性修复工作一定程度上受益于数据科学领域中已有的数据清洗方法,因此相关方法和工具较为成熟. 近些年相关研究方向主要是将机器学习方法与数据清洗方法相结合,并提出新的可行方法与自动化工具[74,75].

3.4 数据隐私性测试与修复

3.4.1 数据隐私性测试研究

机器学习模型隐私性测试工作主要涵盖两种,即测试模型数据的隐私性与算法的隐私性. 现有的模型数据隐私测试主要侧重于设计隐私攻击方法测试模型的私密数据信息是否会被泄露,缺乏具体的量化指标.

私密信息窃取. 私密数据窃取研究在数据隐私领域有着成熟的发展,现有的大量工作提出了各种不同的方法对模型的训练数据、隐私属性乃至特定成员信息进行窃取或推断. Fredrikson等人[123]针对药物遗传学的研究表明,提供医疗建议的模型存在数据隐私风险,攻击者在给定人口统计信息和模型的情况下,可以预测患者的遗传标记. Hitaj等人[124]进一步研究了深度学习模型的数据隐私性问题,发现了分布式、联合或去中心化的深度学习方法也无法从根本上保护用户的训

练数据隐私. 除训练数据的隐私问题外,Ateniese 等人[125]构造元分类器对训练集的隐私属性进行了攻击与测试,他们对数据隐私性测试结果表示,提供记录级隐私的差分隐私方法无法有效地防御这类属性推断攻击. Shokri 等人[77]通过训练"影子模型"的方法设计了一种成员推断攻击,可以判断某条数据是否属于模型训练集的标签.

私密数据窃取方法主要关注对隐私数据的攻击效果而非隐私性的测评结果. 近年,纪守领等人[30]针对机器学习模型安全与隐私问题进行了深入研究,并整理了近年数据隐私攻击的研究进展,对相关的攻击方法已有较为全面的总结.

小结. 目前数据隐私的相关测试与评估工作主要通过数据隐私攻击的方法来测试不同场景下模型数据的隐私安全,总体来讲,这类测试方法对隐私性的测试效果较弱且缺乏量化的评估方法. 如何设计数据隐私的量化评估指标,并集成现有的方法开发一套类似鲁棒性测试中 Cleverhans[94]的综合测试工具将会是未来的研究方向之一.

3.4.2　数据隐私性修复研究

机器学习模型隐私性修复的研究主要采用基于差分隐私(differential privacy)的方法对隐私数据或者私密属性进行保护,从而保障模型的数据隐私性. 差分隐私将隐私定义为添加或移除输入数据中的任何一条记录不会显著影响算法输出结果的一种属性[50],即模型不会对输入数据的记录中学到额外的知识.

基于差分隐私的数据隐私保护. 纪守领等人[30]的研究中已对现有的基于差分隐私的数据隐私保护进行了全面地讨论,在这里只介绍有代表性的几种隐私保护方法. Erlingsson 等人[126]提出了 RAPPOR 差分隐私机制,该机制使用随机响应技术保护用户发送至服务器的训练数据,并让用户在响应服务器查询时以一定概率返回随机值. Papernot 等人[78]提出了 PATE,这是一种保护数据隐私的通用型框架,该框架以黑盒方式将使用不相交数据训练的多个模型结合作为教师模型,并在预测阶段将教师模型的预测结果投票聚合,从而得到最终输出. 在聚合结果时还可以引入噪声打乱投票统计,进一步保护数据隐私. Salem 等人[127]对成员推断攻击进行了研究,他们提出机器学习模型训练的过拟合问题是促使成员推断攻击成功的原因之一,并基于该观点设计了随机失活和模型集成两种改进方法,以提升模型的数据隐私性. 他们的改进方法在 CIFAR-100 等图像数据集上取得了显著的效果,有效地改善了模型数据的隐私性.

基于安全多方计算的数据隐私保护. 安全多方计算(Secure Multi-party Computation,SMC)旨在解决如下场景的问题[128]:一组参与者需要共同计算某个约定的函数,每个参与者提供一个对其他人保密的输入,需要可信第三方计算函数结果并分别交给各个参与者,保证每个参与者数据的隐私性. Yao 最早在 1982 年提出了两方安全计算协议[129],之后研究人员提出了可以计算任意函数的基于密码学安全模型的安全多方计算协议[130]. 如今随着理论的发展与完善,安全多方计算协议被广泛地应用在数据挖掘、机器学习等各种领域,用来确保算法、数据等信息的隐私性. 2003 年,Vaidya 等人[131]提出了一种基于安全多方计算的 k-means 聚类方法,使得各方可以在对彼此的属性一无所知的情况下协同执行 k-means 计算. 随着机器学习技术的不断发展,安全多方计算的研究者们不再单单局限于单个简单的机器学习算法. 2017 年,Mehnaz 等人[132]提出了一个通用的、具有强安全保证的框架,使得多方能够在保障隐私的同时对数据进行训练,他们的框架设计了两种安全梯度下降算法,适用于大型数据集的多方计算场景和多种机器学习算法. Mohassel 等人[133]构建了 SecureML 方法,该方法通过将加密数据发送到两个非共谋服务器,使用安全两方计算训练线性回归、逻辑回归模型以及使用随机梯度下降方法的模型. 然而该方法在训练中使用多项式近似非线性激活函数,一定程度上改变了模型的训练方式并导致了模型精度下降. 为改善该问题,Rouhani 等人[134]提出 DeepSecure 框架,这是第一个具有可证明安全的深度学习框架,该框架基于乱码电路(Garbled Circuit,GC)协议,在模型训练中支持非线性激活函数,且无需改变训练方式,保证了模型的精度. 近年,Huang 等人[79]提出了轻量级的深度学习隐私保护框架 LPP-CNN,它在确保 CNN 模型准确性和安全性的同时,降低了安全交互过程中的计算与通信开销.

联邦学习与隐私保护. 如今,随着机器学习在工业场景的大规模部署以及对隐私性的高要求,联邦学习和相关的隐私性保护与修复方法得到了越来越多的重视. 联邦学习最早由谷歌提出,以解决私密数据需要保留在本地端而服务器端需要更新全局模型的问题[135]. 在联邦学习中,各方首先从服务端下载一个基本的共享模型,基于本地数据训练后将更新的模型参数上传至服务端;随后服务端将各方的参数(或者参数更新量)聚合至全局模型后再次共享出去,以此反复指导达到停止条件. 为确保数据隐私性,联邦学习的研究者们提出了多种隐私保护方案. Bonawitz 等人[80]提出了一种高效且鲁棒安全多方计算协议——安全聚合,该协议基于秘密共享,旨在保证设备与服务端之间通信及服务端参数聚合过程的安全性,具有较低的计算代码. Mcmahan 等人[136]利用差分隐私的方法,在全局模型聚

合过程中对相关参数进行扰动,从而保护用户级数据隐私性,他们在真实数据集的实验结果表明,在应用该方法后总体模型效用并未有显著降低,总体计算开销可接受. 除了差分隐私和安全多方计算的方法以外,Weng等人[137]还提出了基于区块链的联邦学习隐私保护方法DeepChain,该方法将区块链和安全聚合协议结合,可以同时保证本地参数在通信中的隐私性与正确性,并为整个训练过程提供可审核性.

小结. 得益于现有隐私攻击与防御技术的发展,目前研究人员已提出了差分隐私、安全多方计算等多种方法改善模型数据隐私问题,缓解训练数据窃取、成员推断等隐私攻击方法,且已有研究者对针对机器学习模型隐私问题进行全面地调研[30]. 总体来看,现有研究中大量使用差分隐私技术保障模型的训练数据的隐私性,然而差分隐私处理数据时,需要对数据的可用性与隐私性进行权衡,追求高隐私保证程度的同时也会降低模型数据的训练效果与效率. 安全多方计算和联邦学习等方法也为模型数据隐私性提供了保障,但现有研究主要关注较为简单的机器学习模型,且通信开销大,效率也有待提升. 此外,如何集成现有的模型隐私保护方法,并设计通用、便利的工具以改善模型隐私性也会是未来的研究方向之一.

4 模型算法测试与修复技术

机器学习模型的算法一般需要开发者手动选择或者设计. 传统的机器学习模型需要选择合适的超参数,而深度学习模型需要选择并连接各个算子或者模型层从而构成完整的网络结构. 随后,模型算法程序在给定数据集上训练以调整其中的参数权重,并获得完成给定分类、回归、文本处理等任务的能力. 在这个过程中,不合理的模型算法设计或者训练环节的种种问题都有可能影响模型算法的具体性能与其在任务上的表现,从而导致模型算法程序在特定输入下产生错误行为或者输出不公平预测结果. 在本节中,本文主要从正确性、公平性、可解释性以及隐私性四种安全测试属性方面,介绍现有针对机器学习模型算法程序环节的主要测试工作,并阐述对应的测试属性修复研究. 表4对比总结了主流的模型算法测试与修复技术,并对其效果与应用领域进行了评估.

4.1 算法鲁棒性测试与修复

4.1.1 算法鲁棒性测试研究

研究人员发现,机器学习模型的训练样本集不可能覆盖全部输入空间的可能性,因此无法训练出一个覆盖所有样本特征的模型,这就导致训练后模型算法的决策边界与真实决策边界不一致[58]. 因此,当输入样本存在特定扰动时,模型算法容易出现分类错误. 为评估模型算法鲁棒性,维护机器学习模型安全,现有工作提出了一系列鲁棒性评估与测试准则.

鲁棒性评估与测试准则. Moosavi-Dezfooli等人[83]提出了DeepFool方法,通过设计并计算用于欺骗机器学习模型的对抗扰动,构建对抗性样例并量化地评估机器学习模型算法的对抗鲁棒性,该工作率先对模型鲁棒性提出了一定的量化准则,计算简单且较为有效. 后续的研究中,Ruan等人[162]通过计算输入数据的L_0范数安全半径,并生成全局鲁棒性的上下界,以量化评估模型的全局对抗鲁棒性,具有较高的评估效率. Gopinath等人[84]提出了一种机器学习模型鲁棒性评估方法,在大量标记数据上聚类并识别确保模型分类正确的安全数据区域,从而评估模型的对抗鲁棒性,在数据集上安全数据区域越大的模型意味着其算法有着更好的鲁棒性. Mangal等人[163]提出了概率鲁棒性概念,以指导在更接近实际环境的非对抗性输入环境下,评估模型的鲁棒性. 近年,鲁棒性测试工作不再限制在计算机视觉领域中,而在广泛的领域中被应用. Goel等人[138]针对自然语言处理(Nature Language Processing,NLP)系统设计了自动化对抗鲁棒性评估平台,以辅助从业者便捷地评估模型的鲁棒性. Lorenz等人[164]针对自动驾驶等系统中的点云模型设计了鲁棒性测试方法,通过模拟真实世界的3D变换验证自动驾驶模型在不同场景下的鲁棒性与表现. Bhojanapalli等人[165]和Mahmood等人[85]对先进的Transformer模型进行了鲁棒性评估与度量,并研究了具有对抗扰动的大批量数据对Transformer模型的鲁棒性影响.

小结. 目前研究人员对机器学习模型算法的鲁棒性测试工作的关注相对较少,鲁棒性的研究主要仍以设计样本数据的对抗扰动为主. 当前基于算法鲁棒性测试工作虽然提出一定的鲁棒性评估标准,但是普遍缺乏一个统一的测试准则,且相关评估主要针对对抗鲁棒性,缺乏其他方面的模型鲁棒性与泛化性的评估工作. 如何设计一套综合的、通用的模型数据以及算法鲁棒性评估工具,将会是未来重要的研究方向.

4.1.2 算法鲁棒性修复研究

对抗训练. 为修复机器学习模型算法的鲁棒性问题,研究人员提出了对抗训练的方法,改进并提升模型的对抗鲁棒性. 该方法旨在利用对抗样本训练机器学习模型并提升模型对于同类型的对抗扰动的鲁棒性. Goodfellow等人[57]最早提出将对抗样本数据和训练数据一同训练模型以增强对抗鲁棒性的方法. 之后的研究[166~168]也提出了类似的观点并证明了对抗训练对提升模型的对抗鲁棒性的有效性. 然而,这些对抗训练的研究普遍采用确定性攻击算法来生成训练样本,即需要利用特定的对抗样本算法生成对应的训练数据. 近

表4　模型算法测试与修复典型技术对比总结

功能描述	方法类别	应用领域	方法描述	效果	相关工作
算法鲁棒性测评	鲁棒性评估与测试准则	图像	计算欺骗模型的最小扰动	强	文献[83]
		图像	识别模型输入空间鲁棒区域	弱	文献[84]
		文本	利用对抗样本等多种范式评估模型	强	文献[138]
算法鲁棒性修复	对抗训练	主要为图像	使用对抗样本重训练模型	强	文献[57]等
	鲁棒优化	主要为图像	使用正则化方法处理并优化模型,削弱扰动影响	强	文献[139]等
算法正确性测评	模型差异测试	图像	通过白盒交叉验证方法测试流行模型的差异行为	弱	文献[140]
		图像	变异模糊测试并最大化原始与变异输入的差异	强	文献[141]
	模型蜕变测试	数值数据	利用蜕变测试的方法测试机器学习模型属性	弱	文献[142]
		图像	设计了多个通用蜕变关系测试机器学习系统特征	强	文献[143]
		文本	针对NLP系统设计了蜕变关系并测试	弱	文献[144]
	测试充分性评估	图像	基于覆盖率的模糊测试和基于属性的测试结合	弱	文献[26]
		图像	利用神经元覆盖率等覆盖率准则进行模糊测试	强	文献[24]
	模型调试	图像	分析模型差分状态并识别模型"故障神经元"	强	文献[145]
算法正确性修复	重训练	图像	生成并机器学习系统的异常行为样例并重训练	弱	文献[140]
		图像	基于神经风格转换学习故障样本并重训练模型	弱	文献[146]
		图像、文本	应用多种策略修复模型训练问题并重训练	强	文献[25]
	模型调试修复	图像、文本	构建影响模型描述网络中数据的状态并分析错误	强	文献[147]
		主要为数值数据	调试机器学习模型算法故障并定位问题的原因	弱	文献[148]
算法公平性测评	公平性测试工具/框架	数值数据	结合了多个指标细粒度探索偏差并进行严格评估	弱	文献[149]
		数值数据	自动化生成包含敏感属性的输入并测试歧视问题	弱	文献[150]
		数值数据	在输入空间随机抽样歧视性样例并在邻域搜索	强	文献[151]
		数值数据	通过分析模型行为以发现潜在的群体公平性问题	强	文献[152]
算法公平性修复	处理中修复	数值数据	将发掘的歧视性样例放入数据集并进行重训练	弱	文献[151]
		数值数据	将公平性作为机器学习模型优化目标性	强	文献[113]
		数值数据	丢弃部分公平性与准确率优化方向矛盾神经元	强	文献[27]
	后处理修复	数值数据	拒绝对接近决策边界的输出样本	弱	文献[153]
		文本	自动检测并修复输出偏差结果并重构公平输出	弱	文献[154]
算法可解释性测评	人工可解释性测评	数值数据	调研参与者在输入变化下给出模型预期输出变化	弱	文献[155]
	自动化可解释性测评	数值数据	设计蜕变关系评测系统功能可解释性	弱	文献[156]
算法可解释性修复	可解释性提升	数值数据	使用可解释性强的算法构建模型	弱	文献[157]
		文本数据	自动学习任务中重要文字并减少无关信息	强	文献[158]
算法隐私性测评	隐私性评估	数值数据	多次运行候选算法并统计对算法隐私的侵犯程度	弱	文献[159]
	模型萃取攻击	图像	通过查询ReLU临界点的查询窃取模型参数信息	强	文献[160]
算法隐私性修复	基于加密的算法隐私保护	数值数据、图像	基于同态加密等方法对模型组件设计加密算法	强	文献[161]等
	基于安全多方计算隐私保护	数值数据、图像	设计安全多方计算协议保障模型算法信息隐私性	强	文献[80]等

年,随着生成对抗网络(Generative Adversarial Network,GAN)的流行,Lee等人[169]提出了利用非确定性生成器来生成对抗样本的方法,该方法通过构造包含生成器和分类器的GAN,不利用特定的对抗样本算法来生成具有对抗性扰动的图片.Wang等人[170]还整理现有工作并提出了一种机器学习系统鲁棒性测试工具,可以自动化地生成有助于提升模型鲁棒性的测试样例,并在基准实验中通过重训练证明了有效性.除此之外,Kim等人[171]提出了一种新的深度学习系统测试充分性标准,即深度学习系统意外充分性,这种标准假设模型应当在良好的训练数据与实际数据上的表现接近,因此通过衡量系统在两类数据上的表现差异,衡量系统对输入的实际数据以及其特征的意外程度,并通过采样、重训练的方法改进系统的鲁棒性.

鲁棒优化.为加强机器学习模型算法的鲁棒性,研究人员将各种正则化方法结合进模型,使对抗样本不会明显干扰到模型的分类结果.2009年,Xu等人[172]针对支持向量机研究了模型的鲁棒性和正则化方法的关系,他们的结果表明:对于支持向量机这种基于内核的类分类器,正则化方法和鲁棒优化具有一定程度上的

等价性,正则化支持向量机因此有着更良好的泛化性能和鲁棒性. 之后的研究者们进一步探索了各种正则化方法以提升模型鲁棒性. Demontis 等人[173]的研究进一步强调了分类器正则化与鲁棒问题的关系,并提出了一种八角范数正则化方法,可以调整线性分类器权重的稀疏性和安全性,并在图像分类和垃圾邮件检测等任务上验证了该正则化方法对模型算法鲁棒性的修复与提升. 近年,Pauli 等人[139]提出了一个训练多层神经网络的框架,该框架在训练中不仅最小化神经网络的训练损失,而且最小化其 Lipschitz 常数,通过保持其 Lipschitz 常数较小来提升算法的鲁棒性. 除了传统分类模型,研究者也关注其他模型的鲁棒性优化问题. Chen 等人[174]关注基于树的模型的鲁棒性,他们的研究证明了该类模型鲁棒性也容易受到对抗性扰动的危害,并设计了一种算法优化树模型的鲁棒性.

小结. 基于对抗训练的修复方法往往需要大量额外的计算资源和时间成本用于生成对抗样本并进行模型重训练,且对未知对抗扰动的抵抗能力较差,而鲁棒优化方法虽然可以提升模型鲁棒性,但是难以彻底防御对抗样本的攻击. 如何设计一套更高效、更通用的算法鲁棒性修复方法将会是未来研究的重要挑战. 研究者可以从对抗扰动的机理出发,结合模型可解释性研究,对模型的推理逻辑与神经元激活情况进行调试,从而避免重复的训练流程,以更高效的方法修复模型算法鲁棒性问题.

4.2 算法正确性测试与修复

4.2.1 算法正确性测试研究

为测试机器学习模型算法的正确性,判断算法程序在给定输入下是否可以正确地进行预测,目前学术界与工业界的研究者提出了各式各样的测试方法与工具. 值得注意的是,尽管算法的正确性测试和鲁棒性测试中,存在问题的模型都对给定输入产生了预期外的错误输出,但两者本质上衡量的是不同的模型安全属性. 鲁棒性测试旨在衡量模型的预测输出不受外界干扰的能力,即模型在特定输入上的预测结果和具有轻微扰动的相同输入的预测结果之间不会有显著的偏差;而正确性测试侧重于评价模型能够正确预测或分类未知输入样本的能力,例如在“大苹果”构成的图片数据集上训练的具有良好正确性的分类模型,对于“小苹果”的测试图片也应当能够进行正确的预测. 机器学习模型算法的正确性问题一般表现为模型的算法程序对特定输入的输出结果与预期产生偏差,这种输出偏差可能来自不充分的训练或者算法不合理设计.

模型差异测试. 差异测试(differential testing)是一种常见的机器学习测试方法,该方法通过在多个相似的应用程序之间进行交叉验证作为测试预言,并观察这些相似程序是否针对相同的输入产生不同的输出以检测其中的正确性问题[32]. 2017 年,Pei 等人[140]提出的 DeepXplore 基于交叉验证的测试预言对白盒自动驾驶系统、图像分类系统进行了测试,该方法率先引入了神经元覆盖的概念,并利用多个 DL 系统之间的输出差异引导测试. 在实验中,DeepXplore 发现多个流行的自动驾驶系统在特定的具有扰动的输入下会产生严重的误判,有可能危害使用者的生命安全. 之后的研究中,Guo 等人[141]提出了 DLfuzz,该方法基于差异模糊测试,对模型输入进行变异并最大化神经元覆盖率以及原始输入与变异输入之间的预测差异,因此无需额外的测试预言或者其他程序的交叉验证即可高效地发掘模型的错误行为并测试算法程序的正确性问题. DLFuzz 在和 DeepXplore 的对比实验中取得了更高的神经元覆盖率和更少的时间开销. 近年,Sun 等人[147]将软件测试领域流行的蜕变测试和变异测试的方法相结合并设计了自动化的测试方法对两个流行的机器翻译系统进行了测试. 他们的方法通过上下文相似的变异方法生成测试语句,并在翻译前后根据蜕变关系判断未被突变的部分语句是否发生超过阈值的异常扰动,从而检测其中的算法正确性问题.

然而差异测试自身存在一定的局限,对于程序之间的结果差异,测试者难以分辨存在具体的漏洞问题或者实现错误的具体程序,而且在多个程序存在错误而个别程序实现正确的情况下,这种测试方法容易对漏洞问题产生误判.

模型蜕变测试. 除了差异测试以外,传统软件测试的蜕变测试(metamorphic testing)方法在缺乏测试预言的情况下是一种受欢迎的测试方法. 它旨在设计一个或多个用来验证算法或者功能实现的必要属性(也称为蜕变关系),并构造测试样例模型以判断是否满足蜕变关系. 2008 年,Murphy 等人[142]率先利用蜕变测试的方法测试机器学习应用的属性是否正确,他们考虑到模型只关注数据样例之间的关系,不关注属性的特定值,因此提出相应的蜕变关系来测试模型的算法程序,即对数据集的所有数据的属性加或乘一个常量后,模型算法的结果不变. 之后研究者针对不同的机器学习算法提出了更多种类的蜕变关系. 在 Murphy 等人工作的基础上,Xie 等人[175]进一步对有监督的机器学习分类器设计了 6 类蜕变关系并测试了 k-最近邻和朴素贝叶斯分类器两种经典机器学习算法. Dwarakanath 等人[176]对两种流行的机器学习分类器 SVM 和 ResNet 分别进行了测试,测试中通过设计加入扰动、变化图像等蜕变关系,利用蜕变关系构造测试预言并观察分类器

的结果是否符合预期,以测试算法程序的正确性. Al-Azani等人[177]也分别使用蜕变测试的方法对朴素贝叶斯和k-最近邻机器学习算法进行了测试. Xie等人[143]设计了METTLE,这是一套基于蜕变测试的无监督机器学习验证评估系统,通过内置的11个通用的蜕变关系,评估了机器学习系统期望中应当具备的属性与特征,该系统在实验与用户评估中证明了对机器学习系统的测评的有效性. 近期,Jiang等人[144]对机器翻译系统设计了多种蜕变关系,并测试了包括亚马逊、微软在内的4个机器翻译系统的算法有效性与正确性. 他们设计了多种蜕变关系并在实验中揭示了部分机器翻译算法程序存在的错误与问题. 他们为机器翻译系统设计了语句级和文档级两种级别的蜕变关系,其中语句级涵盖了语句的同义词替换、缩写词拆分、符号替换、单词大小写转换、时态变换等,文档级包含加入随机扰动、删除部分语句等,为NLP领域的蜕变关系设计提供了基础和支撑.

测试充分性评估. 前文的测试方法旨在为机器学习模型正确性测试构建测试预言,从而协助测试模型中输出的偏差或者不一致问题. 除去这些方法,覆盖率等测试标准指导的测试充分性评估也是一类常见的机器学习模型测试方法. 传统软件工程的覆盖率准则一般指软件程序的代码、路径等在测试中被测试到的比率,用于衡量测试方法对程序源码、路径等覆盖与执行的程度. 而机器学习模型由于本身功能实现效果并不依赖具体的程序源码和路径而是与模型参数权重相关,因此无法直接套用这种传统的测试准则. 2017年,DeepXplore[140]率先针对机器学习模型引入了神经元覆盖率的概念,即被激活或者满足特定条件的神经元占总神经元数量的比例. 之后研究者们提出了各种各样的神经元覆盖率准则,旨在通过神经网络覆盖情况反映模型的学习逻辑和行为规则,进而发掘异常行为并评估模型质量[24,178~180]. 随着测试充分性评估的研究的发展,研究人员提出了各式各样的神经元覆盖率准则用于测试模型算法的正确性. Ma等人[178]提出了DeepGauge,通过神经元边界覆盖率、Top-k神经元覆盖率等神经元级别和层级别的多粒度覆盖率准则引导机器学习系统的测试. Odena等人[26]将基于覆盖率的模糊测试和基于属性的测试结合并设计了开源测试工具TensorFuzz,用于发现神经网络版本之间的差异与不一致问题. Xie等人[24]更进一步提出了DeepHunter,这是一个覆盖率引导的模糊测试系统,基于8种图像变换方法生成测试输入样例并利用神经元覆盖率、神经元边界覆盖率等5种覆盖率准则评估测试. 该系统可以有效地增加神经网络的覆盖率并检测模型迁移等行为引发的模型算法程序缺陷问题,取得了比前人更优秀的测试效果. 除此之外,Sun等人[179]基于动态符号执行技术设计了DNN输入样本生成方法,针对不同的覆盖率需求动态地生成输入样本从而有效地提升神经网络覆盖率指标. 随着自动驾驶系统的流行,其中的安全问题得到了研究者们的广泛关注. Tian等人[180]提出了DeepTest方法测试其中的错误行为与正确性问题,该方法对种子图像应用9种图像变换方法生成逼真的合成图像,以模拟真实世界驾驶图片的同时提升模型神经元覆盖率,随后对自动驾驶系统设计对应的蜕变规则以测试并发掘系统的错误行为. Braiek等人[181]将基于搜索的测试方法和基于覆盖率的测试结合,以确保生成的测试用例具有最大的多样性并且能够增加测试用例的神经元覆盖率,在实验中该测试方法检测自动驾驶模型算法缺陷问题的表现优于TensorFuzz工具.

近年,Yan等人[182]对各类神经网络覆盖率指标进行了实证研究,并观察到神经网络覆盖率对模型质量的影响并不显著. 目前机器学习模型覆盖率准则的研究逐渐从神经元覆盖率为主的研究转向各类模型充分性测试准则,例如系统意外充分性[171]、重要神经元驱动充分性[183]等.

模型调试. 研究人员通过调试模型也可以测试算法中潜在的正确性问题. Ma等人[145]设计了MODE方法测试并修复模型中的正确性问题,该方法通过分析模型差分状态,以识别导致模型错误的模型内部特征与导致错误分类的“故障神经元”,然后该方法选择对故障神经元影响重大的输入样本重新训练,进而改善模型的分类错误等问题. 近年,Zhang等人[25]针对模型中的梯度爆炸等影响模型性能与预测结果的训练问题,提出了一种自动化的测试与修复工具,用于实时监测模型训练的参数以高效地检测模型中的5种训练问题并提供自动化的修复.

除此之外,研究者们还提出了其他基于样本生成的正确性问题测试方法. Xie等人[184]针对模型量化压缩问题中引入的不一致行为提出了DiffChaser,通过在神经网络的决策边界生成输入样例以捕获模型量化压缩中引入的差异问题. Yang等人[185]设计了一种基于测试样本生成的机器学习系统恶意行为检测方法,利用进化算法和混淆策略模拟恶意样本生成测试样例从而有效地指导测试.

小结. 目前机器学习模型算法的正确性测试工作已经有了较为成熟的发展. 传统软件工程领域的测试方法虽然无法直接应用在机器学习模型测试中,但是其中的蜕变测试、差异测试、覆盖率测试准则等方法都为模型正确性测试技术提供了基础和支撑. 研究人员也设计了DeepHunter这类集成多种测试准则、且功能

较为全面的算法正确性测试工具[24]. 未来的相关测试工作的趋势将会倾向于方法的多样化和通用化,具有高度有效性与通用性的模型正确性测试工具将会是下阶段研究工作的重要目标之一.

4.2.2 算法正确性修复研究

目前主流的机器学习模型算法正确性的修复方法为重训练,即通过收集异常行为样本或者修正模型结构并重新训练,以改善模型的预测能力,保障算法正确性.

重训练. 一些研究在测试机器学习模型算法正确性的同时,往往会通过重训练的方法改进并修复原始模型的性能. 例如DeepXplore[140]生成并收集了数千个自动驾驶系统等机器学习系统的异常行为样例,并将机器学习模型在这些测试样例上重训练,从而将模型精度提升了3%. 类似的,文献[145,146,179]等也利用了重训练的方法,将生成的测试样例或者触发模型错误行为的异常输入样本作为训练数据重新训练模型并改进模型的准确率、修复其中的正确性问题. Zhang等人[25]总结了机器学习模型训练问题的特征与常用的解决方案并设计了一套自动化的修复工具,通过改进模型并重训练的方法提升存在训练问题的模型的准确率. 近年,Chen等人[186]提出利用图片的旋转、平移、缩放等蜕变关系生成新数据并扩充数据集并指导机器学习模型的训练. 他们的方法在实验中证明了扩充后的数据集上重训练的模型具有更高的性能. Yu等人[146]针对已部署机器学习模型的正确性问题,基于神经风格转换的思想设计了一种数据样本生成工具,通过学习故障样本中的未知故障模式,并将其引入训练数据中,从而实现对数据集的扩充并进行重训练. Xie等人[187]专注于循环神经网络的错误行为的解释与修复,他们的方法利用影响模型来描述网络在所有训练数据中的状态和统计行为,并对错误进行影响分析,从而有效地估计现有或新增训练样本对给定预测的影响. 他们的方法通过修复数据集样本并重训练来改进模型执行预测的效果. 尽管重训练方法可以有效地修复算法的正确性问题,但是训练需要额外的时间开销和计算资源,且未在先前的测试中暴露的正确性问题依然难以修复.

模型调试修复. Sun等人[147]针对测试的机器翻译系统的正确性问题设计了自动化的翻译结果不一致问题修复方法,他们的方法不需要训练数据和算法源码,甚至在黑盒场景下只需要执行被测翻译器和翻译输出的能力,具有较高的应用价值. 近年,Wardat等人[148]设计了DeepLocalize方法以对机器学习模型算法程序的故障进行调试与定位,他们的方法通过捕获模型中的数值错误并在训练过程中监控模型来识别造成问题的原因,并辅助开发人员调整模型. 尽管该方法确实可以辅助开发人员修复模型的算法问题,但是难以做到自动化的修复,需要较多的人工干涉.

小结. 目前机器学习算法程序正确性问题的修复工作主要以重训练为主,通过补充数据集或者修改模型并重启训练流程来提升模型准确率并避免模型错误行为. 这类方法需要较大的时间成本和较高的计算资源开销用于重新训练模型,且需要特定的测试方法先生成触发错误行为的输入样例. 如何设计低成本且高效率的模型正确性问题修复方法与测试工具将会是下阶段研究的重要目标之一.

4.3 算法公平性测试与修复

4.3.1 算法公平性测试研究

基于算法的公平性测试旨在检测并发掘机器学习模型算法环节的公平性问题与歧视问题,评估模型算法层面在执行预测、分类等功能时是否对输入的受保护属性敏感或者对特定敏感属性存在歧视问题.

公平性测试工具/框架. 2017年,Tramer等人[149]设计了FairTest,这是第一个帮助开发人员审查模型算法程序是否受到公平性、歧视性等问题影响的综合性工具. 之后,Angell等人[150]提出了Themis,该测试框架利用因果分析的方法,专注于群体歧视和因果歧视两种公平性问题,并通过自动化的方法生成包含敏感属性的待测试系统的输入进行测试,并在测试中变化敏感属性并记录输出的行为与变化. Aggarwal等人[188]将动态符号执行技术与生成测试样例的方法相结合,提出了一种自动生成公平性测试输入的技术,并在实验中表现出比Themis更好的测试有效性. Udeshi等人[151]继承并改进了Themis并提出了Aeqitas,不同于Themis关注的群体歧视,Aeqitas是一种专注于个体歧视实例的自动化公平性测试生成方法,通过在模型输入空间随机抽样获取引发模型算法歧视问题的样例,并在其邻域搜索更多的相似性质输入. 这些歧视性输入可以辅助模型重训练以改进公平性问题.

近年,研究者们的目标从设计各式各样的机器学习模型公平性测试方法逐渐转向为流行的卷积网络、循环神经网络等深度学习模型设计自动化、高效率的算法公平性测试方法或工具. 2020年,Black等人[152]提出了FlipTest,这是一种黑盒的公平性测试方法,该方法对个人和子群体在模型中的行为进行分析,以发现模型潜在的群体公平性歧视问题并展示模型行为中的相关模式. Zhang等人[189,190]针对深度学习模型的图像数据与深层次模型的特点,设计了一种轻量级、自动化的公平性测试工具,该工具基于对抗性采样的方法,通过最大化相似样本的预测差异来指导歧视性样本的搜

索与生成.

小结. 基于算法的公平性测试的相关研究主要基于传统软件工程测试的方法,设计自动化、高效率的测试样本生成方法并对模型算法公平性与歧视问题进行测试. 近期的研究主要关注点从白盒场景转移到更复杂更困难的黑盒测试场景,并且逐渐转向关注深度学习模型等流行机器学习模型. 未来随着公平性测试准则的多样化,如何设计一套通用性强、支持多种模型架构且测试效率高的算法公平性测试工具将会成为研究重点之一.

4.3.2　算法公平性修复研究

针对算法的公平性修复工作一般为处理中和后处理修复方法,即在模型训练和预测中或者之后改进并修复公平性问题.

处理中修复. 现有的部分测试工作通过添加歧视性样本并重训练的方法对算法公平性问题提供了处理中修复[151,189],例如,Aeqitas将发掘的歧视性样例放入模型并进行重训练以改进模型公平性[151]. 除重训练模型以外,Chakraborty等人[113]提出将公平性作为机器学习模型超参数优化的目标可以保持模型预测准确率并辅助模型获取更公平的预测结果. 这是一种典型的处理中的公平性修复方法. 他们在三个数据集上对性别和种族两个敏感属性构建测试并验证了这一观点. 近年,Gao等人[27]提出了轻量级的模型公平性修复工具FairNeuron,该工具结合模型调试的思路,通过在训练中检测具有公平性与准确率优化方向相矛盾的神经元,并选择性丢弃这些神经元以缓解模型的公平性问题,在实验中取得了先前工作更好的修复效果.

后处理修复. Kamiran等人[153]提出的拒绝选项分类是一种有代表性的后处理方法,该方法认为接近决策边界的样本更有可能存在歧视问题,并通过拒绝对应样本的输出结果的方法缓解潜在的歧视问题. 然而这种方法更倾向于缓解歧视问题,并不能真正修复问题. 近些年,Yang等人[154]提出了第一个针对黑盒情感分析系统的公平性问题的后处理修复方法,当检测到系统输出的偏差结果时自动化修复并重构公平的输出结果,并在实验中取得了一定的准确率提升.

小结. 目前机器学习模型算法公平性修复工作主要关注于处理中的修复方法,并提出了重训练、修改模型训练目标、修改模型神经元激活状态等方法改进算法公平性,但是在后处理阶段的修复方法较为有限,且效果有待进一步提升. 未来的研究可能会更多关注后处理等探索较少的领域,并设计自动化、通用化的模型算法修复工具.

4.4　算法可解释性测试与修复

4.4.1　算法可解释性测试研究

机器学习模型算法可解释性的研究目前主要分为两种,即人工可解释性评测与自动化可解释性评测.

人工可解释性评测. 早期的评测方法主要通过人工评估的方法对算法输出的合理解释进行评价. 2017年,Doshi-Velez等人[191]讨论了可解释性的定义并给出了一种可解释性评估方法的分类方法,该方法将可解释性评估分为三类:基于应用、基于人以及基于功能. 其中基于应用的评估需要在实际应用程序中人工进行评价试验,基于人的评估是在保证目标应用本质的前提下进行更简单的人工试验,基于功能的评估则不需要人工实验而是使用可解释性定义作为解释质量的评价标准. 之后,Slack等人[155]对上千名参与者进行了实证研究,实验中他们要求参与者们在给定输入变化的情况下给出模型的预期输出变化,然后记录不同模型的精度和完成时间. 他们的研究结果表明,决策树和逻辑回归模型比神经网络具有更好的可解释性.

自动化可解释性评测. 自动化评估可解释性比人工评估更加客观,也具备更少的时间成本与资源开销. 在自动化评估中,如何给出合理的可解释性度量是一个重要问题. Cheng等人[192]提出了一种模型识别判断可解释性的度量指标,该指标衡量模型是否通过物体周围的遮挡环境来识别并进行判断. Molnar[157]提出了基于算法类别的模型可解释性度量方法,并确定了包括逻辑回归、决策树在内的几个具有良好可解释性的机器学习模型. Zhou等人[156]基于蜕变测试的方法,针对搜索系统等设计了多种蜕变关系用于测试模型的可解释性. 他们提出的蜕变关系包括:搜索集合子集得到结果数量应当不大于搜索整个集合的结果数量;集合并集的搜索结果数量应当小于其中任何一个结合的结果数量. 他们的方法可以有效地帮助用户理解和使用机器学习系统. 近年,Ross等人[193]提出通过测量用户交互操作表示以重建目标实例的程度来量化生成模型的可解释性,他们在Amazon Mechanical Turk上大规模试验的结果表明,他们的任务可以很好地区分可解释性存在问题的纠缠模型(entangled model).

小结. 目前机器学习算法可解释性的相关研究仍处于初级阶段,相关的测试度量方法与评估技术较少,目前没有一套通用且成体系的可解释性度量标准,且相关评估更多面向传统机器学习模型,例如决策树和逻辑回归等[157]. 如何面向各种类型的机器学习模型设计一套有效且通用的可解释性评估标准与工具将会是未来研究的重点之一.

4.4.2　算法可解释性修复研究

可解释性提升. 由于机器学习可解释性研究较为

有限,目前机器学习模型算法可解释性的修复与提升方法较少. Molnar[157]提出"实现可解释性的最简单方法是仅使用能够创建可解释模型的算法子集",并在研究中给出了逻辑回归、决策树等具有良好可解释性的机器学习算法模型. Schielzeth[194]的研究表明,输入变量的中心化和标准化可以有效提高回归模型的可解释性. Chen等人[195]尝试了一些有助于改进医学分类器模型可解释性的方法,并对这些方法进行了评估. 这些方法将分类器得分转换为疾病概率度量. 他们的评估结果表明,在不影响模型性能的情况下,可以将任意尺度上的分类器得分校准为疾病概率. 近年,Chen等人[158]针对NLP任务提出了变分文字掩码的方法来自动学习任务特定的重要文字,并减少分类中的无关信息,从而最终提高了模型预测的可解释性. 实验表明,他们的方法可以有效提升模型预测精度和可解释性. Kokhlikyan等人[196]设计了一个开源模型可解释性库,对机器学习框架Pytorch支持的各类属性与算法补充了高级概述,并设计了可视化方案,辅助用户理解模型的算法构造.

小结. 目前模型可解释性的修复与提升研究较为有限,其主要针对传统机器学习模型(例如决策树、回归模型等),相关领域仍处于发展阶段. 如何给出模型可解释性的通用评估并针对各类深度学习模型设计高效的修复方法将会是未来研究的重要挑战.

4.5 算法隐私性测试与修复

4.5.1 算法隐私性测试研究

目前的机器学习模型算法隐私性的测评研究旨在评估隐私侵犯行为对模型隐私性的影响,也有研究者通过设计模型萃取攻击的方法,探索机器学习模型算法抵抗隐私侵害的能力. 目前相关的评估与测试工作仍处于探索阶段[159,197~199],相关研究较为有限.

隐私性评估. Ding等人[159]设计了反例生成器,通过多次运行候选算法,找到导致算法违反差分隐私的错误并生成反例统计对算法隐私的侵犯程度,从而辅助开发人员理解和评估模型算法的隐私性. Bichsel等人[198]设计了一种差分隐私评估系统DP-Finder,可以自动推导算法强制执行的差分隐私下限. 该系统通过抽样方法来估计反例的隐私侵犯,并使用数值优化器将反例对模型隐私的侵犯效果最大化,从而系统地、大规模地搜索侵犯隐私的行为,并对模型算法隐私性进行评估.

模型萃取攻击. Tramèr等人[199]率先研究了模型萃取方法,他们发现攻击者理论上只需要通过模型预测接口进行n+1次查询就能窃取到输入为n维的线性机器学习模型. 在实验中,他们的方法可以以近乎完美的保真度提取流行机器学习模型算法,然而该方法需要的查询次数多,且只能萃取简单的线性机器学习模型. 之后的研究者们旨在降低查询次数并提升模型的萃取效果. Wang等人[200]研究了超参数窃取攻击方法. 他们首先在模型参数处计算目标函数的梯度并构造关于超参数的线性方程组,随后利用线性最小二乘法推导近似解,以估计超参数. 他们的方法适用于各种流行的机器学习算法,如逻辑回归、支持向量机和神经网络. 近年,Carlini等人[160]提出了一种新型差分攻击,可以通过极少的查询次数有效地窃取目标模型的算法参数,并在MNIST数据集上的实验中取得了良好的效果,他们精妙的模型萃取理论得到了数学的验证,但是需要目标模型使用ReLU激活函数,这限制了该方法的实际价值. Jagielski等人[201]开发了一种基于学习的攻击,这是第一种实用的功能等效提取攻击,可以无须训练直接提取目标模型权重. 他们的方法在图像分类器上进行了实验并取得了良好的实用效果.

小结. 现有的模型算法隐私性评估工作仍处于探索阶段,相关研究较少且没有成体系的评估指标和工具. 现有算法隐私性测试研究主要通过设计模型萃取等隐私攻击方法测试模型的隐私性边界,从而侧面评价模型算法的隐私性,这类评估方法并不直观且效率较低. 如何设计一套规范、通用、高效的模型算法隐私性量化指标并设计工具与框架对模型算法隐私进行全面评估将会是未来研究的挑战之一.

4.5.2 算法隐私性修复研究

目前的模型算法隐私修复方法主要通过加密、安全多方计算等方法保障模型算法的隐私安全,防止非法访问或窃取模型的算法.

基于加密的算法隐私保护. 同态加密(Homomorphic Encryption,HE)是一种允许用户直接在密文上进行运算的加密形式,其得到的结果仍是密文,并且解密结果与对明文运算的结果一致. Xie等人[202]提出了一种隐私保护模型crypto-nets,将同态加密技术引入神经网络,他们利用已训练好的神经网络直接在密文上做预测,并返回加密预测结果. 该方法不需要数据所有者参与中间计算,从而不会泄漏关于模型的信息. Gilad-Bachrach等人[203]更进一步使加密数据可以直接输入模型,返回的加密预测结果只能由密钥持有者解密,他们的模型在MNIST数据集上的分类准确率达到了98.95%,且云端很少泄露给数据持有者额外的信息,但是他们的方法需要较大的资源开销. Hesamifard等人[204]使用低阶多项式逼近CNN的常用激活函数,并构建了对Leveled-FHE加密的密文数据进行分类的深度学习模型CryptoDL,该模型在MNIST和CIFAR-10数据集上取得了高效且准确的预测结果. 除此之外,Liu等人[161]进一步提出了MiniONN,他们为神经网络常用操作设计了多种不经意协议(oblivious protocols),只需要

在预测阶段使用密码原语,即可在预测精度损失忽略不计的情况下对算法进行加密,确保模型算法的隐私性.

基于安全多方计算的算法隐私保护. 除同态加密技术以外,研究人员还提出将安全多方计算结合到机器学习模型中以确保隐私性. 早在2000年前后,安全多方计算就作为隐私保护方法被数据科学领域(例如数据挖掘)广泛关注[205]. Bonawitz等人[80]设计了一种高效且鲁棒的安全多方通信协议,只需要一个服务器提供者,即可允许大批量地、安全地计算并聚合来自各个用户设备的模型参数更新总和,且具有较低的通讯开销和运行时开销. 随着安全多方计算技术的发展与成熟,研究人员进一步构建更加便捷、实用的安全多方计算平台,从而维护并提升模型算法与数据的隐私性. Juvekar等人[206]进一步基于同态加密技术和乱码电路设计了一种新的安全神经网络推理方案GAZELLE,该方案下,客户端可以在不向服务端透露其输入的情况下获取分类结果,同时保证服务端神经网络的隐私. Chandran等人[207]提出了EzPC框架,可以生成高效的安全两方计算协议,该框架集合了算术共享和乱码电路,使服务器无法获得无客户端具体输入输出信息,客户端也无法从服务器端获取额外的模型信息,保障了算法和数据的隐私安全. 近些年,Zheng等人[208]提出了Cerbero,这是一套端到端的协作学习平台,使各方在不共享明文数据的情况下计算学习任务,充分保障了算法与数据的隐私性. Knott等人[209]提出了CRYPTEN,这是一种安全多方计算框架,提供了全面的张量计算库,其中所有计算都通过安全的MPC执行,便于开发者使用以确保模型的隐私安全.

小结. 目前机器学习模型算法隐私修复工作已经取得了一定的进展,研究人员提出了同态加密等加密方法以及多种安全多方计算工具,辅助开发者保障模型隐私性. 然而同态加密算法和安全多方计算方法普遍存在计算开销、通信开销大等问题,导致在实际场景中可用性较差,如何让这些隐私保障方法走出学术界,进而推广并应用在工业场景中是未来阶段需要解决的问题.

5　模型实现测试与修复技术

机器学习模型的框架实现需要开发者根据各个算子与功能的预期效果,手动进行实现和编译. 在这个过程中,极有可能引入各种漏洞问题[210,211]. 这些框架漏洞会导致模型的运行效率、功能的执行效果受到影响,从而得到错误的输出结果或者额外的时间与计算开销. 目前,随着机器学习测试工作的发展与进步,越来越多的框架实现问题得到了研究人员的关注. 在本节中,本文主要从正确性和效率两方面,介绍现有针对机器学习模型框架实现环节的主要测试工作,并阐述对应的安全测试属性修复研究. 表5对主流的机器学习模型实现的测试方法进行了对比,并总结了这些方法的测试框架与测试效果.

表5　模型实现测试典型技术对比总结

功能描述	方法类别	测试框架	方法描述	效果	相关工作
实现正确性测评	实现差异测试	TenorFlow, CNTK, Theano	对比框架同一功能在输入下的输出差异	弱	文献[52]
		TenorFlow, CNTK, Theano	基于模糊测试生成不同的模型以探索框架	弱	文献[212]
		TensorFlow, PyTorch	构造等效图对同一功能实现进行对比	强	文献[53]
	实现蜕变测试	Weka, C4.5等	检查蜕变关系执行前后一致并自动化测试	弱	文献[213]
		Scikit-learn, TensorFlow	基于增减数据等蜕变关系测试框架实现	弱	文献[214]
	测试样本生成方法	TensorFlow, Keras	基于模型突变测试的方法检测实现问题	弱	文献[215]
		TensorFlow, PyTorch, MXNet	自动化提取框架功能约束并生成样例	强	文献[216]
	框架漏洞研究	TensorFlow, Torch等	调研开源社区上的机器学习框架漏洞特性	弱	文献[210]
	框架底层库测试	TVM	覆盖度指导变异低级中间表示以模糊测试	强	文献[217]
		TensorFlow	对算子误差进行了计算评估并与实际对比	强	文献[54]
实现效率测评	效率问题实证研究	TensorFlow, Caffe, Torch	实证研究不同框架上训练时间等性能差异	弱	文献[218]
		TensorFlow, CNTK, PyTorch, MXNet	测试部署环境迁移对实现的性能影响	弱	文献[48]

5.1　实现正确性测试与修复

5.1.1　实现正确性测试研究

类似于模型算法的正确性测试,针对框架实现正确性测试的相关研究主要也利用差异测试和蜕变测试的方法解决测试预言的问题.

实现差异测试. 2018年,Srisakaokul等人[219]率先提出了一种多实现测试的方法来测试WEKA等机器学习框架,通过在相同的输入上运行同一算法或者功能的多个框架实现的方法进行测试,并基于投票的方法将输出结果与多数实现结果不同的异常实现区分出来. 之后,Pham等人[52]设计了Cradle测试方法,通过对比TensorFlow,CNTK等多个机器学习框架对同一功能实现在相同输入下的输出差异,检测其中的框架漏洞问题,然而Cradle在检测中存在较高的误报率,会把复

杂模型中一些正常的层判断为存在实现问题. Guo 等人[51]进一步研究了多框架的差异测试方法,并提出了 Audee,这种测试方法可以测试包括 TensorFlow,PyTorch,CNTK,Theano 四个流行机器学习框架,检测并定位模型崩溃、实现不一致等正确性问题. 该方法有效地降低了误报率并在实际测试中检测到了这些机器学习框架的多个实现漏洞并得到了开发者的确认与修复. 除此之外,Wang 等人[220]为机器学习模型设计了一系列模型变异规则以探索框架代码各种的行为,并提出启发式策略引导模型在变异生成过程朝着放大机器学习框架之间差异的方向发展. 他们的方法可以高效地生成并变异模型,从而在多个框架上引导差异测试. 近年,Gu 等人[212]提出了 Muffin,基于模糊测试的思想生成不同的 DL 模型来探索机器学习框架,并设计了一组度量标准来衡量不同框架对模型训练过程中同一功能实现的不一致性. Zhang 等人[221]设计了 Duo 测试方法,该方法基于多框架底层算子的不同实现设计了差异测试,利用遗传算法变异算子的测试样例并通过交叉验证的方法评估各个底层算子(例如 Tanh,ReLU)实现的正确性. 这些实现差异测试的工作都依赖多框架之间的实现对比与交叉验证,难以在单个框架上直接检测并判断正确性问题,一定程度上限制了实用性.

随着不同机器学习框架对同一功能的不同实现的支持逐渐减少[222],利用模型的多框架实现进行差异测试的场景如今面临着两大问题. 首先,这类差异测试往往需要同一功能在多个框架上均有稳定的实现,然而最先进的机器学习算法往往不能够被多个框架的开发者及时实现,这限制了这类测试方法的可用性;其次,在只有单个实现的场景下,多框架差异测试技术往往需要巨大的成本去进行专门的功能实现和维护. 为解决多框架差异测试所遇到的困境,Wang 等人[53]提出了 EAGLE,这种测试方法通过在单一框架中构造等效图来替代多框架对同一功能或者模型的实现,从而实现高效率高通用性的差异测试. 该方法不再依赖多个机器学习框架的交叉验证,通过精心设计的等效规则对 TensorFlow,PyTorch 等流行机器学习框架进行测试并检测出 13 个新的底层框架实现问题,通过模型的等效图转换方法,有效地解决了差异测试场景的局限.

实现蜕变测试. 蜕变测试中,研究者们通过设计不同的蜕变关系,对机器学习功能实现的各种特性与约束进行测试. 2009 年,Murphy 等人[223]率先将传统软件工程中蜕变测试的方法应用在机器学习框架中,对包括机器学习库 WEKA 在内的 5 个应用设计了总结 25 种蜕变关系,并成功测试出 WEKA 的 SVM 算法实现的一个正确性漏洞问题. 同年,Murphy 等人[213]进一步设计了一套自动化的蜕变测试系统,并对 WEKA,C4.5 等机器学习库与工具进行测试. 在实验中该测试系统可以有效地检测出源码中插入的缺陷变异体,证明了该工具在测试正确性问题上的有效性. Ding 等人[214]设计了扩充数据集、增减数据集分类等多种蜕变关系测试机器学习框架功能的正确性. 近年,蜕变测试被广泛地应用在测试机器学习框架底层实现上,用于验证框架底层编译器等工具的功能是否正确实现. Wang 等人[224]对移动设备部署使用的神经网络加速器算子设计了蜕变关系,他们设计了度量标准来定量评估加速器算子的精度性能,并在两种流行的神经网络加速器 HiAI 和 Snapdragon 引擎上进行了实验. Xiao 等人[56]为机器学习框架底层编译器巧妙地设计了蜕变关系,通过引入结果为固定常量的算子组合以测试编译器在各种输入下的优化编译功能的正确性. 他们对四种流行的机器学习编译器进行测试并检测到超过 435 种可能导致错误的输入. 现有的蜕变测试工作已经取得了一定的成果,但是蜕变测试无法保证通过测试的框架一定没有正确性问题,且蜕变关系的设计对测试的具体效果有着显著的影响,因此该测试方法总体效果仍有待提升.

测试样本生成方法. 除此之外,研究者们还提出了不同的测试样本生成的方法,通过变异等方法生成大量的模型,从而对框架的功能与实现进行测试. 不同于传统软件程序,机器学习模型有着特殊的结构和执行逻辑,因此需要设计特别的变异方法或者测试样本生成方法. Ma 等人[215]针对机器学习系统设计了一种突变测试方法 DeepMutation,通过在数据和源码层面引入突变体并设计模型和权重的变异方法以检测并杀死引入的突变体,从而验证模型变异方法在检测框架实现正确性问题的有效性. Hu 等人[225]进一步改进了该方法,为前馈神经网络和循环神经网络设计了权重、神经元、模型层等不同层次的总计 17 个模型的变异方法以有效地检测植入的突变体. 近年,Xie 等人[216]提出了 DocTer,这是一种高效的自动化测试样例生成方法,可以从机器学习框架文档中自动化提取 API 参数约束并生成有效和无效两种测试输入以测试框架功能是否正确实现. 他们在 TensorFlow,PyTorch,MXNet 框架上的实验证明了该工具检测正确性问题的有效性,在实验中 DocTer 检测到了多个框架实现正确性问题. Luo 等人[226]设计了一种基于图的模糊测试方法以检测框架实现的质量,他们的方法通过探索模型结构、模型参数和数据输入的不同组合,实现了六种不同的突变策略以生成多种类的机器学习模型.

框架漏洞研究. 此外,Islam 等人[210]还对 GitHub 和 Stack Overflow 开源社区上的机器学习框架漏洞的特性进行了调研,他们发现大部分漏洞都会导致崩溃、性能下降或者功能受损,从而影响框架功能的正确实现. Jia

等人[211]着重关注TensorFlow的漏洞问题,他们发现超过35%的TensorFlow漏洞问题都会导致功能错误的症状,此时程序无法按预期运行,并得到错误的结果.现有的框架漏洞的研究没有形成完整的技术体系,主要以调研和分析为主,仍有待研究者们深入探索.

框架底层库测试.随着机器学习框架测试技术的发展与进步,研究者们不再局限于框架源码的正确性,而是更多关注机器学习框架底层算子和编译器的功能实现是否符合期望.前文已提到一些使用差异测试或者蜕变测试技术测试框架底层算子或者编译器等的研究工作[56,221,224].除此之外,Zhang等人[54]设计了名为Predoo的测试工具,不同于传统的模糊测试技术,该工具对7个算子的形状变量输入和误差进行了细粒度评估,在实验中,该工具可以触发TensorFlow框架的精度错误,该类型错误会导致框架无法输出正确的结果.Predoo对待测试的算子的精度提供了数学上的验证与分析,但是测评的算子较少且较为简单,未来仍有较大的研究与提升的空间.Liu等人[217]针对机器学习编译器TVM提出了Tzer测试方法.该方法通过覆盖度反馈指导,对TVM的低级中间表示进行变异,以实现更有效的模糊化测试.实验中,他们的方法检测到49个未知的实现错误,其中25个已得到修复错误,证明了该方法的有效性和价值.

小结.针对机器学习框架实现正确性的测试工作在近些年得到了广泛的关注,研究人员提出了各式各样的测试方法.类似算法程序正确性的测试工作,框架实现方面的测试主要利用差异测试和蜕变测试解决测试预言问题,并广泛使用模糊测试和变异测试的方法执行测试.目前,框架实现正确性研究的总体趋势主要从浅层次的框架源码实现向着深层次的底层算子和编译器等方向进行研究,研究内容越发深入.然而框架底层实现往往依赖编译完成的动态链接库以及cuda等第三方库,这使底层的测试工作难以直接进行.因此如何设计通用性强、效率高且能够深入测试框架实现更底层问题的测试方法,将会是研究人员的重要关注点之一.

5.1.2　实现正确性修复研究

目前大部分机器学习框架正确性测试工作都无法提供对应的修复方法,其原因主要在于复杂的机器学习框架源码与自动化程序修复问题本身的困难,目前关于这方面的研究仍处于探索阶段,对框架正确性漏洞的具体修复一般还是需要开发者在确定问题之后人工修复.现有的研究工作提出了一些可以辅助开发者修复的方法.Pham等人[52]和Guo等人[51]提出了机器学习模型正确性问题的定位技术,可以将模型触发的框架正确性漏洞定位到具体的模型层乃至层内参数组合,辅助开发者确定问题的原因并进行修复.目前研究工作主要通过测试具体漏洞问题并反馈给开发者的方法,辅助修复正确性问题[53,216,217,221].

小结.机器学习框架实现正确性的修复工作目前处于探索阶段.学术界和工业界都缺乏一套高度自动化或者成体系的修复方法与工具,研究者们主要通过测试具体漏洞问题并与开发者沟通反馈的方法辅助修复.如何设计一套自动且有效的框架或者工具实现修复辅助工具以降低人工成本,将会是未来研究的重点难题之一.

5.2　实现效率测试与修复

5.2.1　实现效率测试研究

近些年,随着机器学习测试技术的发展,模型框架实现的效率问题得到了越来越多的关注.基于框架的效率测试旨在检测、发掘框架实现中的效率漏洞,例如在特定部署环境下的同一框架实现的额外时间与计算资源开销[48]、不同框架实现在特定数据上的性能差异[218]等.这类漏洞问题往往会影响机器学习模型在不同部署环境迁移时的效率与可用性.

效率问题实证研究.Liu等人[218]首先构造大规模样例测试了TensorFlow,Caffe和Torch三种流行机器学习框架的实现,发现相同的模型功能与数据集在不同框架实现上存在显著的训练时间、准确率等性能差异,且模型的不同的框架实现对对抗性示例表现出不同程度的鲁棒性.Guo等人[48]更进一步关注不同部署环境下框架实现的性能问题,发现模型在不同部署环境中迁移会在框架实现过程中产生明显性能变化(即时间与内存开销变化),并基于观察结果发掘了一系列影响兼容性和可靠性的效率漏洞问题.Zhang等人[227]通过设计问卷调研了深度学习应用的开发与测试情况,他们的调研显示,性能与效率问题开发者们测试过程中重点关注的问题的之一,仅次于鲁棒性问题.Zhang等人[228]对机器学习中遇到的各类问题进行了实证研究并将GitHub等开源社区中发现的问题分为七类,他们的研究注意到175个机器学习问题中只有9个属于效率问题,原因可能在于该类型问题难以发觉.近年,Chen等人[229]对基于机器学习技术的移动应用程序的效率漏洞问题进行了总结,他们将包含内存问题、模型速度问题等常见漏洞问题总计分为23类,并对问题发生的具体框架功能、修复措施进行了总结.这些效率问题的实证研究尽管取得了一定的成果,但总体上仍缺乏成体系的测试理论与方法指导,仍有较大的发展空间.

小结.现有的基于框架实现的效率测试工作仍处于探索阶段,目前仍缺乏通用、成体系的测试方法,相关研究主要以调研、总结的实证研究为主.相比于正确

性问题,效率问题发生较少且较难发掘[228],因此对应的测试方法仍有待发展.

5.2.2 实现效率修复研究

现有研究对框架实现的效率问题的修复研究相当有限,一般在测试工作的讨论或总结部分提出.Chen等人[229]在研究中对移动应用程序的框架实现效率漏洞的常见修复方案进行了总结,并分类了9种修复策略,其中最常见的是重新安装框架并更换版本.目前业界并没有成体系的修复方案,一般需要开发者根据测试中发掘的性能与效率漏洞人工调试并修复底层源码.

小结.对框架实现的效率问题的测试与修复研究都处于探索阶段,目前仍没有成体系的检测或者修复方法.随着机器学习技术的发展与在工业界的大规模应用,效率问题对大规模的模型部署与使用的影响会越来越大.关于效率问题的自动化测试与修复工作可能会成为未来的重要发展方向.

6 研究难点与未来挑战

近年来,针对机器学习模型安全的测试与修复的工作已经取得了一定的成果,然而该研究整体上仍处于较为初级的阶段,相关的测试与修复方法的性能、通用性等方面有许多关键问题尚待解决,且尚未形成完整的技术体系,并且对机器学习模型安全的各类测试属性的评测指标多样且不统一.现有的测试与修复工具功能也较为单一.与此同时,以生成对抗网络(GAN)及Transformer模型为代表的深度学习模型的技术仍在不断地发展,这些模型在为机器学习任务带来更多解决方案的同时,也为模型的测试与修复工作带来了机遇和挑战.例如,研究人员一方面可以利用GAN对模型数据进行清理修复,另一方面也可以设计测试方法对Transformer等模型的公平性、正确性等安全测试属性进行测试.这使得无论是模型的测试技术还是修复方法都有着广阔的发展空间和持续的挑战.

目前机器学习模型安全的测试与修复研究主要针对正确性、鲁棒性、公平性三个模型测试属性,而针对模型的其他测试属性(即效率、可解释性以及隐私性)的测试准则、测试方法均处于初级阶段,相关研究较为有限且缺乏成体系的评估准则.因此本节首先对机器学习模型的正确性、鲁棒性、公平性三种安全测试属性在模型各个阶段的相关测试与修复技术进行了总结,随后本节梳理了效率、可解释性、隐私性三种相关研究较为有限的安全测试属性的技术进展,并对各个测试属性相应的研究难点与未来潜在的研究机遇进行了分析.

6.1 模型正确性相关技术分析

从方法设计层面上看,现有的模型正确性测试技术已经取得了一定的成果.目前相关测试与修复工作在模型的数据、算法、实现三个层面上都有涉及.其中,正确性测试与修复相关研究在数据层面方法较为成熟,得益于数据科学领域已有的技术,相关研究设计了多种工具可以自动化完成异常数据的识别、测试和清理[73,74].在算法层面,正确性测试相关工作也有较为成熟的发展,研究人员提出了蜕变测试、差异测试等方法测试模型错误行为并提供了多种方法借助重训练来调整模型算法,然而这类修复方法往往需要较大的计算资源与时间开销,且对未检测到的错误行为往往无法直接修复,总体来看存在一定的局限.在实现层面,正确性测试工作也得到了一定的发展,研究人员设计了各种测试样本生成方法、蜕变关系以及差异测试方法测试机器学习框架乃至底层算子的实现.但是机器学习框架实现的修复工作目前仍处于探索阶段,现有研究仅能为开发人员修复问题提供有限的支持.

从结果及评价体系上看,现有正确性测试方法的效果较好,目前针对数据、算法乃至框架实现已有一些通用性较强、便于使用且测试效果较好的测试工具与框架[24,53,110,216].在正确性修复方面的研究则存在一定的局限.其中现有的算法正确性修复的研究通过重训练和调试的方法虽然可以在模型修复中取得较明显的提升,但是会引入较大额外开销;而实现正确性的修复工作更是处于探索阶段,相关研究十分稀少,且已有方法仅能提供协助修复的效果.此外,在评价体系上,尽管相关测评工作已有研究者提出大量的标准准则用于指导测试,但依然缺乏一套通用的评价体系以对模型进行综合、直观的评估.

从技术挑战上看,目前模型正确性主要面临三个挑战.

(1)针对模型算法正确性的测试与评估方法种类较多,仅仅用于指导测试的不同的覆盖率准则就有数种,例如神经元边界覆盖率准则、Top-k神经元覆盖率准则等.然而,尽管现有的覆盖准则多种多样,但这些准则对模型正确性、模型质量的具体评估效果以及适用的测试场景仍有待研究.

(2)目前算法正确性修复方法普遍依赖寻找引发错误行为的样例,并利用这些样例重训练模型.如今随着深度学习模型技术的发展,训练模型往往需要数日乃至数月的时间,并占用几十甚至上百块GPU,TPU,因此重训练过程具有较大的开销.尽管研究者通过即时监控模型的方法一定程度上削减了训练的时间开销[25],但是依然需要大量的时间与计算资源.

(3)框架实现正确性问题目前缺乏有效的修复方法与工具,尽管目前已有研究者结合机器学习技术设计了软件漏洞源文件的定位方法[230,231],但准确定位具

体源码实现的问题还存在一定挑战.

从未来可能研究方向上看,现有正确性的测试与修复方法的局限性可能从如下角度突破.

(1)设计一套有效且直观的评价标准或者综合的测试工具可以在一定程度上改善现有覆盖率准则种类多样但效果参差不齐的问题. 例如Goodfellow等人[94]构建的Cleverhans鲁棒性测试样本生成库,集成了多种先进的测试样本生成方法,可以辅助开发者对模型鲁棒性进行综合的测评. 在未来的模型正确性测试的研究中,可以基于DeepHunter,TensorFuzz等开源工具[24,26]以及各类测试充分性评估方法制定一套综合的测试工具,对不同任务场景、不同种类的模型进行全面的评估.

(2)设计更有效的正确性修复方法,降低或者避免重训练带来的大量时间成本将会是未来正确性修复工作的一个重要研究方向. 现有的研究通过早停(early stopping)存在正确性问题的模型,从而避免计算资源与时间成本的浪费. 在未来的研究中,通过模型调试等方法直接定位存在正确性问题的神经元,并通过训练部分神经元或者其他轻量级修复方法进行修复,可以大大降低修复方法的时间与计算资源的开销.

(3)在模型实现正确性方面,设计自动化的框架实现问题定位与修复方法将会是未来的研究方向之一. 目前模型实现层面的修复研究仍处于探索阶段,现有的研究对检测到的正确性问题往往需要手动排查并人为修复. 未来的研究可以结合代码插桩等传统软件工程技术,实现自动化的模型正确性问题的分析与定位,从而降低人工成本,实现高效的模型修复.

6.2　模型鲁棒性相关技术分析

从方法设计层面上看,现有模型鲁棒性的测试与修复技术主要面向模型数据或者基于模型数据的生成方法实现. 总体来看,现有的模型鲁棒性的测试与修复工作发展较为全面,测试方法上研究人员提出了不同的鲁棒性评估方法,涵盖了图像、文本、音频多个领域,并设计了FGSM,DeepFool等多种对抗样本生成方法[57,83,93],用于测试模型的鲁棒性上界. 修复方面涵盖了重训练、随机化、去噪、对抗输入检测等多种方法,但其中部分方法局限较大,对白盒对抗扰动的抵抗能力较差.

从结果及评价体系上看,现有大多数鲁棒性测试方法的效果较弱,依赖生成对抗输入测试模型的鲁棒性上限,但是少有对模型鲁棒性进行直观地量化评估的指标或者方法,鲁棒性相关的评价体系目前存在一定不足. 此外现有的鲁棒性修复方法尽管种类多样,但是部分方法效果较差,存在明显的局限,例如随机化方法和去噪方法容易被具有较强对抗攻击能力的白盒自适应攻击[82]破解;对抗样本检测会减少数据集样本数量,且不能从根本上改进鲁棒性问题;而效果最好的对抗训练方法则需要大量资源生成对抗样本并进行重训练.

从技术挑战与趋势上看,目前模型鲁棒性方面主要面临两个挑战.

(1)现有的模型鲁棒性相关评估主要针对对抗鲁棒性,缺乏其他方面的模型鲁棒性或泛化性能的评估与研究工作.

(2)现有的模型鲁棒性修复工作往往采用静态的对抗防御方法,普遍对白盒自适应对抗样本的抵抗能力弱[82],且基于对抗训练等方法的鲁棒性修复需要消耗较多计算资源和时间.

从未来可能研究方向上看,现有鲁棒性的测试与修复方法的局限性可能从如下角度突破.

(1)设计一套综合的、通用的模型鲁棒性评估工具是未来重要的研究方向. 现有的模型鲁棒性评估主要关注模型的对抗鲁棒性,然而,机器学习模型的输入扰动不单单只来自精心设计的对抗噪声,现实场景中光暗变化、音频噪声以及传感器收集数据时的误差都会带来扰动. 随着在各种场景下机器学习模型的部署与应用,未来迫切需要一套综合性强且通用性高的模型鲁棒性评估工具.

(2)如何设计新的鲁棒性修复或对抗样本防御方法,并建立动态自适应的防御体系,是未来研究中的重要的问题. 现有的模型鲁棒性白盒自适应攻击对输入随机化、去噪鲁棒性修复与防御方法有着较高的攻击成功率[82]. 未来的研究应将建立动态自适应的防御体系作为研究重点之一,而不单单局限于某种静态防御方法,从而保障非受控环境下机器学习模型的鲁棒性.

6.3　模型公平性相关技术分析

从方法设计层面上看,现有模型公平性测试与修复技术在模型数据和模型算法上都有一定的研究工作,且大量的研究工作在提出测试技术的同时也给出了对应的修复或者改进方法[67,110]. 由于模型的公平性定义较为多样,因此对模型公平性的测评和修复方法种类繁多,其中数据公平性方面关注的问题涵盖数据样例偏差、类偏差、特征倾斜等,算法公平性方面针对这些公平性问题设计了多种公平性测试工具与框架,利用抽样搜索、数据生成等方法探索算法中潜在的其实问题[150,151],总体来说相关工作的发展较为全面.

从结果及评价体系上看,现有公平性测试工作效果较好,数据与算法的公平性问题测试工作利用聚类等机器学习方法以及数据生成等软件测试方法设计了多种测试工具与框架,在数值数据上取得了较好的测试效果. 在公平性修复方面,研究人员提出了数据集修

正、良性数据生成等预处理方法解决数据集中的不均衡问题,并设计了重训练以及拒绝特定输出等处理中和后处理的修复方法.尽管部分重训练、数据集修正等方法消耗大量资源和时间,且后处理方法往往修复效果较差,但是生成良性数据等修复技术依然取得了明显的公平性的改善.此外,在评价体系上,公平性问题有着完善且多样的评价标准,涵盖群体公平、个体公平等等多种公平性定义,相关研究的发展较为全面.

从技术挑战与趋势上看,目前公平性问题主要面临两个挑战.

(1)模型公平性的定义多样,相关的测试方法繁多,但缺乏一套类似DeepHunter或者Cleverhans的综合性测评工具.这类工具可以方便地帮助用户测试不同定义下的模型公平性,从而全面协助用户改善模型潜在的歧视问题.

(2)针对类不均衡等数据问题,研究人员已提出一些修复工具或者框架进行改善,但现有研究对实际场景中的更细致敏感属性的关注较少(例如种族、性别等).

从未来可能研究方向上看,现有公平性的测试与修复方法的局限性可能从如下角度突破.

(1)基于现有研究对模型公平性的各种定义,设计一套综合的模型公平性评估工具会是未来重要的研究方向.综合的评估工具可以辅助研究人员与模型开发者更好地评价模型潜在的公平性问题,从而促进机器学习模型在实际场景中的应用与部署.

(2)随着机器学习模型在实际场景中的大规模部署,未来的研究将会更关注于种族、性别等细致的敏感属性在推荐、识别等更多复杂任务中的公平性,而不再局限于分类任务中的类不均衡等公平性问题.

6.4 模型其他属性相关技术分析

从方法设计层面上看,现有模型效率测试与修复技术的研究较少,主要基于实证研究的方法对模型的框架实现进行测评,目前并没有成体系的测试与评估方法.类似模型实现的正确性修复,效率相关的修复研究也处于探索阶段,研究人员一般通过向框架开发者反馈漏洞来辅助修复,目前没有有效的修复方法或者工具.机器学习模型可解释性测试与修复技术的研究主要面向模型的算法部分,目前已经有了一定的发展,研究人员已经提出了多种人工与自动化可解释性方法,也提出了一些用于提升模型可解释性的方法与工具.相比另外两种测试属性,对模型隐私性的测试与修复技术的研究涵盖相对广泛,相关工作囊括了模型数据隐私和算法隐私两部分.由于机器学习隐私安全近些年得到了国内外的广泛关注,学术界提出了多种模型隐私修复方法,但是模型隐私性的测试与评估工作一直缺乏有效的量化评估方法,现有工作主要依靠设计隐私攻击的方法探索模型的隐私性边界.

从结果及评价体系上看,现有模型效率测试研究效果较好,能够直观表现出框架实现的效率差异,但是缺乏合适的评价方法.此外,对于不同实现之间的效率差异来自软件实现的问题还是硬件的优化问题,目前的研究没有合适的评估手段.模型的隐私性测试方面也同样面临缺乏有效的评估手段的问题,现有的隐私性测试方法往往是针对模型构造隐私攻击,尽管一些攻击方法可以轻易窃取模型数据与算法,但是依然缺乏合理的量化指标以综合地评价模型的对抗隐私攻击的性能.相对而言,现有的隐私性修复的工作已经取得了一定的成果,并形成了工具与框架,使用户可以方便地使用.模型可解释性测试方面目前已有一些自动化可解释性评测研究,但是这些方法普遍通用性较差,缺乏一套完善的可解释性评价体系.此外,现有的可解释性修复工作效果较差且大多针对传统机器学习模型,对于新兴的深度学习模型相关研究工作较为有限.

从技术挑战与趋势上看,目前效率、可解释性与隐私问题主要面临三个挑战.

(1)目前模型效率研究以调研、总结的实证研究为主,模型隐私性研究则依赖使用隐私攻击来测试模型的隐私性边界,两者在测试中都缺乏有效的问题评估手段,难以定量地判断效率问题以及隐私性问题的具体程度.

(2)随着机器学习技术的发展与工业界的深度学习模型的大规模应用,模型效率问题将会得到越发广泛的关注.目前该领域缺乏有效的检测或修复方法,如何大规模地测试模型的性能或者效率将会是未来研究中的重要挑战.

(3)深度学习技术目前被广泛地应用各个领域,然而目前模型可解释性的修复与提升研究主要针对传统机器学习模型(例如决策树、逻辑回归等),如何为各类深度学习模型可解释性设计高效的修复、提升方法将会是目前的重要挑战之一.

从未来可能研究方向上看,这些属性的测试与修复方法的局限性可能从如下角度突破.

(1)随着机器学习模型在工业界的大规模部署,在模型效率和隐私性的安全属性的测试方面,未来的研究应关注于如何提出有效的评价标准,从而对相应的属性设计量化评估方法,以客观地评判模型的效率与隐私性.

(2)针对机器学习模型实现中的效率与性能问题设计自动化测试与修复方法应为未来的重要研究方向之一.随着机器学习模型底层框架实现的发展,繁杂的代码与模型算子引发了各种各样的性能问题.同一模

型功能的底层实现多种多样,在不同的硬件与软件环境下,这些的实现有着截然不同的性能,使用不合适的实现将会导致模型的性能下降. 因此如何自动化地、大规模地测试模型的性能与效率问题,在未来的模型实现研究中具有重要意义.

(3)现有的模型可解释性的测试与修复研究主要针对较为简单的传统机器学习模型,对参数更多、更复杂的深度学习模型的研究较少. 随着深度学习技术的发展与大范围应用,为保障模型的安全性以及模型行为的可解释性,深度学习模型的可解释性研究将会是未来的重要研究方向.

7　结束语

机器学习技术的快速发展与在实际场景中的广泛应用与部署吸引了大批学术界与工业界的研究者们对机器学习模型的测试属性进行深入研究,并取得了丰硕的成果. 然而,发展迭代中的机器学习模型仍存在潜在安全隐患,现有的模型安全相关的测试与修复方法仍有提升空间,且日新月异的机器学习技术为模型安全带来了更多的机遇与挑战. 为了重新审视面向机器学习模型安全的测试与修复方法的研究现状,梳理现有的技术方法的效果与不足,明确未来研究的挑战与方向,本文系统地从机器学习模型的数据、算法与实现三个环节对正确性、鲁棒性、公平性、效率、可解释性和隐私性等六种模型安全测试属性的相关测试与修复方法进行了研究,回顾了大量有代表性且具有影响力的研究成果并对相关工作进行了科学的归纳与总结. 最后,本文指出了机器学习模型安全的测试与修复工作的研究难点与挑战,并探讨了未来研究中可能的趋势与方向,旨在推动面向机器学习模型安全的测试与修复方法研究的进一步发展.

参考文献

[1] WORTSMAN M, ILHARCO G, GADRE S Y, et al. Model soups: Averaging weights of multiple fine-tuned models improves accuracy without increasing inference time[EB/OL]. (2022-03-10)[2022-07]. https://arxiv.org/abs/2203.05482.

[2] BAO H B, DONG L, PIAO S H, et al. BEiT: BERT pre-training of image transformers[EB/OL]. (2021-06-15) [2022-07]. https://arxiv.org/abs/2106.08254.

[3] TAN M X, LE Q. Efficientnet: Rethinking model scaling for convolutional neural networks[C]//International Conference on Machine Learning. New Orleans: PMLR. org, 2019: 6105-6114.

[4] BROWN T B, MANN B, RYDER N, et al. Language models are few-shot learners[C]//34th International Conference on Neural Information Processing Systems. Vancouver: Curran Associates Inc., 2020: 1877-1901.

[5] MELIS G, KOČISKÝ T, BLUNSOM P. Mogrifier LSTM [EB/OL]. (2019-09-04) [2022-07]. https://arxiv.org/abs/1909.01792.

[6] YAMADA I, ASAI A, SHINDO H, et al. LUKE: Deep contextualized entity representations with entity-aware self-attention[EB/OL]. (2020-10-02)[2022-07]. https://arxiv.org/abs/2010.01057.

[7] KOLOBOV R, OKHAPKINA O, OMELCHISHINA O, et al. MediaSpeech: Multilanguage ASR benchmark and dataset[EB/OL]. (2021-03-30) [2022-07]. https://arxiv. org/abs/2103.16193.

[8] PARK D S, ZHANG Y, JIA Y, et al. Improved noisy student training for automatic speech recognition[EB/OL]. (2020-05-19)[2022-07]. https://arxiv.org/abs/2005.09629.

[9] XU Q T, BAEVSKI A, LIKHOMANENKO T, et al. Self-training and pre-training are complementary for speech recognition[C]//2021 IEEE International Conference on Acoustics, Speech and Signal Processing. Toronto: IEEE, 2021: 3030-3034.

[10] JHA D, RIEGLER M A, JOHANSEN D, et al. DoubleU-net: A deep convolutional neural network for medical image segmentation[C]//2020 IEEE 33rd International Symposium on Computer-Based Medical Systems. Rochester: IEEE, 2020: 558-564.

[11] SRIVASTAVA A, JHA D, CHANDA S, et al. MSRF-net: A multi-scale residual fusion network for biomedical image segmentation[EB/OL]. (2021-05-16) [2022-07]. https://arxiv.org/abs/2105.07451.

[12] WANG J F, HUANG Q M, TANG F L, et al. Stepwise feature fusion: Local guides global[EB/OL]. (2022-03-07) [2022-07]. https://arxiv.org/abs/2203.03635.

[13] STOICA I, SONG D, POPA R A, et al. A Berkeley view of systems challenges for AI[EB/OL]. (2017-12-15) [2022-07]. https://arxiv.org/abs/1712.05855.

[14] Research and Market. Edge AI Market – Forecasts from 2021 to 2026[EB/OL]. (2021-03) [2022-07]. https://www.researchandmarkets.com/reports/5308992/edge-ai-market-forecasts-from-2021-to-2026.

[15] ABADI M. TensorFlow: Learning functions at scale[J]. ACM SIGPLAN Notices, 2016, 51(9): 1.

[16] 马艳军, 于佃海, 吴甜, 等, 飞桨: 源于产业实践的开源深度学习平台[J]. 数据与计算发展前沿, 2019, 1(1): 105-115.
MA Y, YU D, WU T, et al. PaddlePaddle: An open-

source deep learning platform from industrial practice[J]. Frontiers of Data and Domputing, 2019, 1(1): 105-115. (in Chinese)

[17] CHEN T Q, LI M, LI Y T, et al. MXNet: A flexible and efficient machine learning library for heterogeneous distributed systems[EB/OL]. (2015-12-03)[2022-07]. https://arxiv.org/abs/1512.01274.

[18] PASZKE A, GROSS S, MASSA F, et al. Pytorch: An imperative style, high-performance deep learning library [C]//33rd International Conference on Neural Information Processing Systems. Vancouver: Curran Associates, Inc., 2019: 8026-8037.

[19] Google. AI and machine learning products[EB/OL]. (2022) [2022-07-11]. https://cloud. google. com/products/ai/.

[20] Baidu. Baidu Al open platform[EB/OL]. (2021)[2022-07-11]. http://ai.baidu.com/.

[21] JULIA A, JEFF L, SURYA M, et al. Machine Bias [R/OL]. (2016-05-23)[2022-07-11]. https://www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing.

[22] WAKABAYASHI D. Self-driving uber car kills pedestrian in Arizona, where robots roam[EB/OL]. (2018-03-19) [2022-07-11]. https://www.nytimes.com/2018/03/19/technology/uber-driverless-fatality.html.

[23] ELSOM J. Moment an Amazon Alexa tells a terrified mother, 29, to "stab yourself in the heart for the greater good" while reading from rogue Wikipedia text[EB/OL]. (2019-12-19)[2022-07-11]. https://www.dailymail.co.uk/news/article-7809269/Amazon-Alexa-told-terrified-moth - er-29-stab-heart-greater-good.html.

[24] XIE X F, MA L, JUEFEI-XU F, et al. DeepHunter: A coverage-guided fuzz testing framework for deep neural networks[C]//Proceedings of the 28th ACM SIGSOFT International Symposium on Software Testing and Analysis. Beijing: ACM, 2019: 146-157.

[25] ZHANG X Y, ZHAI J, MA S Q, et al. AUTOTRAINER: An automatic DNN training problem detection and repair system[C]//2021 IEEE/ACM 43rd International Conference on Software Engineering. Madrid: IEEE, 2021: 359-371.

[26] ODENA A, OLSSON C, ANDERSEN D, et al. Tensorfuzz: Debugging neural networks with coverage-guided fuzzing[C]//Proceedings of the 36th International Conference on Machine Learning. Virtual Conference: PMLR.org, 2019: 4901-4911.

[27] GAO X Q, ZHAI J, MA S Q, et al. FairNeuron: Improving deep neural network fairness with adversary games on selective neurons[EB/OL]. (2022-04-06) [2022-07-11]. https://arxiv.org/abs/2204.02567.

[28] 中华人民共和国工业和信息化部. 工业和信息化部关于印发《促进新一代人工智能产业发展三年行动计划(2018—2020 年)》的通知[EB/OL]. (2017-12-13)[2022-07-11]. https://www.miit.gov.cn/jgsj/kjs/wjfb/art/2020/art_08d153ee9e9d4676aa69d0aa12676ca1.html.

[29] The White House Office Of Science And Technology Policy. American AI Initiative One Year Annua Report[R/OL]. 2020. https://www.nitrd.gov/nitrdgroups/images/c/c1/American-AI-Initiative-One-Year-Annual-Report.pdf.

[30] 纪守领, 杜天宇, 李进锋, 等. 机器学习模型安全与隐私研究综述[J]. 软件学报, 2021, 32(1): 41-67.
JI S L, DU T Y, LI J F, et al. Security and privacy of machine learning models: A survey[J]. Journal of Software, 2021, 32(1): 41-67. (in Chinese)

[31] HUANG X W, KROENING D, RUAN W J, et al. A survey of safety and trustworthiness of deep neural networks: Verification, testing, adversarial attack and defence, and interpretability[J]. Computer Science Review, 2020, 37: 100270.

[32] ZHANG J M, HARMAN M, MA L, et al. Machine learning testing: Survey, landscapes and horizons[J]. IEEE Transactions on Software Engineering, 2022, 48(1): 1-36.

[33] BRAIEK H B, KHOMH F. On testing machine learning programs[J]. Journal of Systems and Software, 2020, 164: 110542.

[34] MEHRABI N, MORSTATTER F, SAXENA N, et al. A survey on bias and fairness in machine learning[J]. ACM Computing Surveys, 2021, 54(6): 1-35.

[35] AMERSHI S, BEGEL A, BIRD C, et al. Software engineering for machine learning: A case study[C]//2019 IEEE/ACM 41st International Conference on Software Engineering: Software Engineering in Practice. Montreal: IEEE, 2019: 291-300.

[36] JESMEEN M Z H, HOSSEN J, SAYEED S, et al. A survey on cleaning dirty data using machine learning paradigm for big data analytics[J]. Indonesian Journal of Electrical Engineering and Computer Science, 2018, 10(3): 1234-1243.

[37] KHALID S, KHALIL T, NASREEN S. A survey of feature selection and feature extraction techniques in machine learning[C]//2014 Science and Information Conference. London: IEEE, 2014: 372-378.

[38] ROH Y, HEO G, WHANG S E. A survey on data collection for machine learning: A big data - AI integration perspective[J]. IEEE Transactions on Knowledge and Data Engineering, 2021, 33(4): 1328-1347.

[39] REFAEILZADEH P, TANG L, LIU H. Cross-validation [M]//Encyclopedia of Database Systems. Boston: Springer, 2009: 532-538.

[40] SHAHROKNI A, FELDT R. A systematic review of software robustness[J]. Information and Software Technology, 2013, 55(1): 1-17.

[41] IEEE. IEEE Standard Glossary of Software Engineering Terminology[A/OL]. (1990-12-31) [2022-07-11]. https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=159342.

[42] 纪守领, 杜天宇, 邓水光, 等. 深度学习模型鲁棒性研究综述[J]. 计算机学报, 2022, 45(1): 190-206.
JI S L, DU T Y, DENG S G, et al. Robustness certification research on deep learning models: A survey[J]. Chinese Journal of Computers, 2022, 45(1): 190-206. (in Chinese)

[43] GAJANE P, PECHENIZKIY M. On formalizing fairness in prediction with machine learning[EB/OL]. (2017-10-09)[2022-07-11]. https://arxiv.org/abs/1710.03184.

[44] HARDT M, PRICE E, SREBRO N. Equality of opportunity in supervised learning[J]. Advances in Neural Information Processing Systems. Barcelona: Curran Associates Inc., 2016: 29.

[45] ZAFAR M B, VALERA I, ROGRIGUEZ M G, et al. Fairness constraints: Mechanisms for fair classification [C]//Proceedings of the 20th International Conference on Artificial Intelligence and Statistics. Fort Laud-erdale: PMLR, 2017: 962-970.

[46] KUSNER M J, LOFTUS J, RUSSELL C, et al. Counterfactual fairness[J]. Advances in Neural Information Processing Systems. Long Beach: Curran Associates Inc., 2017: 30.

[47] DWORK C, HARDT M, PITASSI T, et al. Fairness through awareness[C]//Proceedings of the 3rd Innovations in Theoretical Computer Science Conference. Beijing: ACM, 2012: 214-226.

[48] GUO Q Y, CHEN S, XIE X F, et al. An empirical study towards characterizing deep learning development and deployment across different frameworks and platforms[C]// 34th IEEE/ACM International Conference on Automated Software Engineering. San Diego: IEEE, 2019: 810-822.

[49] GOODMAN B, FLAXMAN S. European union regulations on algorithmic decision-making and a "right to explanation"[J]. AI Magazine, 2017, 38(3): 50-57.

[50] DWORK C. Differential privacy: A survey of results[C]// International Conference on Theory and Applications of Models of Computation. Berlin: Springer, 2008: 1-19.

[51] GUO Q Y, XIE X F, LI Y, et al. Audee: Automated testing for deep learning frameworks[C]//35th IEEE/ACM International Conference on Automated Software Engineering. Virtual Conference: ACM, 2020: 486-498.

[52] PHAM H V, LUTELLIER T, QI W Z, et al. CRADLE: Cross-backend validation to detect and localize bugs in deep learning libraries[C]//2019 IEEE/ACM 41st International Conference on Software Engineering. Montreal: IEEE, 2019: 1027-1038.

[53] WANG J N, LUTELLIER T, QIAN S S, et al. EAGLE: Creating equivalent graphs to test deep learning libraries [C]//2022 IEEE/ACM 44th International Conference on Software Engineering. Pittsburgh: IEEE, 2022: 798-810.

[54] ZHANG X F, SUN N, FANG C R, et al. Predoo: precision testing of deep learning operators[C]//Proceedings of the 30th ACM SIGSOFT International Symposium on Software Testing and Analysis. Virtual Conference: ACM, 2021: 400-412.

[55] SANTOS S H N, SILVEIRA B N C DA, ANDRADE S A, et al. An experimental study on applying metamorphic testing in machine learning applications[C]//Proceedings of the 5th Brazilian Symposium on Systematic and Automated Software Testing. Natal: ACM, 2020: 98-106.

[56] XIAO D W, LIU Z B, YUAN Y Y, et al. Metamorphic testing of deep learning compilers[J]. Proceedings of the ACM on Measurement and Analysis of Computing Systems, 2022, 6(1): 1-28.

[57] GOODFELLOW I J, SHLENS J, SZEGEDY C. Explaining and harnessing adversarial examples[EB/OL]. (2014-12-20)[2022-07-11]. https://arxiv.org/abs/1412.6572.

[58] PAPERNOT N, FAGHRI F, CARLINI N, et al. Technical report on the CleverHans v2.1.0 adversarial examples library[EB/OL]. (2016-10-03) [2022-07-11]. https://arxiv.org/abs/1610.00768.

[59] XIE C H, WANG J Y, ZHANG Z S, et al. Mitigating adversarial effects through randomization[EB/OL]. (2017-11-06)[2022-07-11]. https://arxiv.org/abs/1711.01991.

[60] KOLBEINSSON A, KOSSAIFI J, PANAGAKIS Y, et al. Tensor dropout for robust learning[J]. IEEE Journal of Selected Topics in Signal Processing, 2021, 15(3): 630-640.

[61] XU W L, EVANS D, QI Y J. Feature squeezing: Detecting adversarial examples in deep neural networks[EB/OL]. (2017-

04-04)[2022-07-11]. https://arxiv.org/abs/1704.01155.

[62] XU W L, EVANS D, QI Y J. Feature squeezing mitigates and detects carlini/Wagner adversarial examples[EB/OL]. (2017-05-30)[2022-07-11]. https://arxiv.org/abs/1705.10686.

[63] WANG J Y, DONG G L, SUN J, et al. Adversarial sample detection for deep neural network through model mutation testing[C]//2019 IEEE/ACM 41st International Conference on Software Engineering. Montreal: IEEE, 2019: 1245-1256.

[64] ZHAO Z, CHEN G K, WANG J Y, et al. Attack as defense: Characterizing adversarial examples using robustness[C]//Proceedings of the 30th ACM SIGSOFT International Symposium on Software Testing and Analysis. Virtual Conference: ACM. 2021: 42-55.

[65] NGUYEN G H, BOUZERDOUM A, PHUNG S L. A supervised learning approach for imbalanced data sets[C]// 2008 19th International Conference on Pattern Recognition. Tampa: IEEE, 2008: 1-4.

[66] AMINI A, SOLEIMANY A P, SCHWARTING W, et al. Uncovering and mitigating algorithmic bias through learned latent structure[C]//Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society. Honolulu: ACM, 2019: 289-295.

[67] MULLICK S S, DATTA S, DHEKANE S G, et al. Appropriateness of performance indices for imbalanced data classification: An analysis[J]. Pattern Recognition, 2020, 102: 107197.

[68] KAMIRAN F, CALDERS T. Classifying without discriminating[C]//2009 2nd International Conference on Computer, Control and Communication. Karachi: IEEE, 2009: 1-6.

[69] AMINI A, SCHWARTING W, ROSMAN G, et al. Variational autoencoder for end-to-end control of autonomous driving with novelty detection and training de-biasing[C]// 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems. Madrid: IEEE, 2018: 568-575.

[70] TOMALIN M, BYRNE B, CONCANNON S, et al. The practical ethics of bias reduction in machine translation: Why domain adaptation is better than data debiasing[J]. Ethics and Information Technology, 2021, 23(3): 419-433.

[71] HOLLAND S, HOSNY A, NEWMAN S, et al. The dataset nutrition label: A framework to drive higher data quality standards[EB/OL]. (2018-05-09) [2022-07-11]. https://arxiv.org/abs/1805.03677.

[72] HYNES N, SCULLEY D, TERRY M. The data linter: Lightweight automated sanity checking for ML data sets [C]//NIPS MLSys Workshop. Cambridge: MIT Press, 2017: 1.

[73] KRISHNAN S, WU E. AlphaClean: Automatic generation of data cleaning pipelines[EB/OL]. (2019-04-26) [2022-07-11]. https://arxiv.org/abs/1904.11827.

[74] LAISHRAM R, PHOHA V V. Curie: A method for protecting SVM Classifier from Poisoning Attack [EB/OL]. (2016-06-05)[2022-07-11]. https://arxiv.org/abs/1606.01584.

[75] ZHANG W N, WANG D, TAN X Y. Robust class-specific autoencoder for data cleaning and classification in the presence of label noise[J]. Neural Processing Letters, 2019, 50(2): 1845-1860.

[76] STEINHARDT J, KOH P W, LIANG P. Certified defenses for data poisoning attacks[C]//Proceedings of the 31st International Conference on Neural Information Processing Systems. Long Beach: Curran Associates Inc., 2017: 3520-3532.

[77] SHOKRI R, STRONATI M, SONG C Z, et al. Membership inference attacks against machine learning models [C]//2017 IEEE Symposium on Security and Privacy. San Jose: IEEE, 2017: 3-18.

[78] PAPERNOT N, ABADI M, ERLINGSSON Ú, et al. Semi-supervised knowledge transfer for deep learning from private training data[EB/OL]. (2016-10-18) [2022-07-11]. https://arxiv.org/abs/1610.05755.

[79] HUANG K, LIU X M, FU S J, et al. A lightweight privacy-preserving CNN feature extraction framework for mobile sensing[J]. IEEE Transactions on Dependable and Secure Computing, 2021, 18(3): 1441-1455.

[80] BONAWITZ K, IVANOV V, KREUTER B, et al. Practical secure aggregation for privacy-preserving machine learning[C]//Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security. Dallas: ACM, 2017: 1175-1191.

[81] SZEGEDY C, ZAREMBA W, SUTSKEVER I, et al. Intriguing properties of neural networks[EB/OL]. (2013-12-31)[2022-07-11]. https://arxiv.org/abs/1312.6199.

[82] CARLINI N, WAGNER D. Towards evaluating the robustness of neural networks[C]//2017 IEEE Symposium on Security and Privacy. San Jose: IEEE, 2017: 39-57.

[83] MOOSAVI-DEZFOOLI S M, FAWZI A, FROSSARD P. DeepFool: A simple and accurate method to fool deep

neural networks[C]//2016 IEEE Conference on Computer Vision and Pattern Recognition. Las Vegas: IEEE, 2016: 2574-2582.

[84] GOPINATH D, KATZ G, PASAREANU C S,et al. DeepSafe: A data-driven approach for assessing robustness of neural networks[C]//International Symposium on Automated Technology for Verification and Analysis. Los Angeles: Springer, 2018: 3-19.

[85] SHEN M, YU H, ZHU L H, et al. Effective and robust physical-world attacks on deep learning face recognition systems[J]. IEEE Transactions on Information Forensics and Security, 2021, 16: 4063-4077.

[86] HAN S C, LIN C H, SHEN C, et al. Rethinking adversarial examples exploiting frequency-based analysis[C]//International Conference on Information and Communications Security. Chongqing: Springer, 2021: 73-89.

[87] MU J M, WANG B H, LI Q, et al. A hard label black-box adversarial attack against graph neural networks[C]//Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security. Virtual Conference: ACM, 2021: 108-125.

[88] MAHMOOD K, MAHMOOD R, VAN DIJK M. On the robustness of vision transformers to adversarial examples [C]//2021 IEEE/CVF International Conference on Computer Vision. Montreal: IEEE, 2021: 7818-7827.

[89] BALUJA S, FISCHER I. Learning to attack: Adversarial transformation networks[C]//Proceedings of the AAAI Conference on Artificial Intelligence. Lousiana: AAAI Press, 2018, 32(1): 2687-2695.

[90] CARLINI N, WAGNER D. Audio adversarial examples: Targeted attacks on speech-to-text[C]//2018 IEEE Security and Privacy Workshops. San Francisco: IEEE, 2018: 1-7.

[91] CISSE M, ADI Y, NEVEROVA N, et al. Houdini: Fooling deep structured prediction models[EB/OL]. (2017-07-17)[2022-07-11]. https://arxiv.org/abs/1707.05373.

[92] ZHENG B L, JIANG P P, WANG Q, et al. Black-box adversarial attacks on commercial speech platforms with minimal information[C]//Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security. Virtual Conference: ACM, 2021: 86-107.

[93] BROWN T B, MANÉ D, ROY A, et al. Adversarial patch [EB/OL]. (2017-12-27)[2022-07-11]. https://arxiv.org/abs/1712.09665.

[94] GOODFELLOW I J, PAPERNOT N, MCDANIEL P. Cleverhans V0.1: An adversarial machine learning library [EB/OL]. (2016-10-03)[2022-07-11]. https://arxiv.org/abs/1610.00768v1.

[95] RAUBER J, BRENDEL W, BETHGE M. Foolbox: A Python toolbox to benchmark the robustness of machine learning models[EB/OL]. (2017-07-13) [2022-07-11]. https://arxiv.org/abs/1707.04131.

[96] NICOLAE M I, SINN M, TRAN M N, et al. Adversarial robustness toolbox v1.0.0[EB/OL]. (2018-07-03) [2022-07-11]. https://arxiv.org/abs/1807.01069.

[97] 任奎, ZHENG Tianhang, 秦湛, 等 . 深度学习中的对抗性攻击和防御[J]. Engineering, 2020, 6(3): 307-339.
REN K, ZHEBG T, QIN Z, et al. Adversarial attacks and defenses in deep learning[J]. Engineering, 2020, 6(3): 307-339. (in Chinese)

[98] LIU X Q, CHENG M H, ZHANG H, et al. Towards robust neural networks via random self-ensemble[C]//European Conference on Computer Vision. Munich: Springer, 2018: 381-397.

[99] GUO C, RANA M, CISSE M, et al. Countering adversarial images using input transformations[EB/OL]. (2017-10-31)[2022-07-11]. https://arxiv.org/abs/1711.00117.

[100] LUO T G, CAI T L, ZHANG M X, et al. RANDOM MASK: Towards robust convolutional neural networks [EB/OL]. (2020-07-27) [2022-07-11]. https://arxiv. org/abs/2007.14249.

[101] SHARMA Y, CHEN P Y. Bypassing feature squeezing by increasing adversary strength[EB/OL]. (2018-03-27) [2022-07-11]. https://arxiv.org/abs/1803.09868.

[102] SAMANGOUEI P, KABKAB M, CHELLAPPA R. Defense-GAN: Protecting classifiers against adversarial attacks using generative models[EB/OL]. (2018-03-17) [2022-07-11]. https://arxiv.org/abs/1805.06605.

[103] LIAO F Z, LIANG M, DONG Y P, et al. Defense against adversarial attacks using high-level representation guided denoiser[C]//2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition. Salt Lake City: IEEE, 2018: 1778-1787.

[104] SHEN S W, JIN G Q, GAO K, et al. APE-GAN: Adversarial perturbation elimination with GAN[EB/OL]. (2017-07-18)[2022-07-11]. https://arxiv.org/abs/1707.05474.

[105] YANG R, CHEN X Q, CAO T J. APE-GAN++: An improved APE-GAN to eliminate adversarial perturbations [J]. IAENG International Journal of Computer Science, 2021, 48(3): 827-844.

[106] KHERCHOUCHE A, FEZZA S A, HAMIDOUCHE W. Detect and defense against adversarial examples in deep

learning using natural scene statistics and adaptive denoising[J]. Neural Computing and Applications, 2022, 34(24): 21567-21582.

[107] ESMAEILPOUR M, CARDINAL P, KOERICH A L. Class-conditional defense GAN against end-to-end speech attacks[C]//ICASSP 2021 – 2021 IEEE International Conference on Acoustics, Speech and Signal Processing. Toronto: IEEE, 2021: 2565-2569.

[108] METZEN J H, GENEWEIN T, FISCHER V, et al. On detecting adversarial perturbations[EB/OL]. (2017-02-14)[2022-07-11]. https://arxiv.org/abs/1702.04267.

[109] CARLINI N, WAGNER D. Adversarial examples are not easily detected: Bypassing ten detection methods [C]//Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security. Dallas: ACM, 2017: 3-14.

[110] BRECK E, POLYZOTIS N, ROY S, et al. Data validation for machine learning[C]//Proceedings of Machine Learning and Systems. Stanford: mlsys. org, 2019: 334-347.

[111] GEBRU T, MORGENSTERN J, VECCHIONE B, et al. Datasheets for datasets[J]. Communications of the ACM, 2021, 64(12): 86-92.

[112] BENDER E M, FRIEDMAN B. Data statements for natural language processing: Toward mitigating system bias and enabling better science[J]. Transactions of the Association for Computational Linguistics, 2018, 6: 587-604.

[113] CHAKRABORTY J, XIA T P, FAHID F M, et al. Software engineering for fairness: A case study with hyperparameter optimization[EB/OL]. (2019-05-14)[2022-07-11]. https://arxiv.org/abs/1905.05786.

[114] KAMIRAN F, CALDERS T. Data preprocessing techniques for classification without discrimination[J]. Knowledge and Information Systems, 2012, 33(1): 1-33.

[115] SATTIGERI P, HOFFMAN S C, CHENTHAMARAKSHAN V, et al. Fairness GAN[EB/OL]. (2018-05-24)[2022-07-11]. https://arxiv.org/abs/1805.09910.

[116] AÏVODJI U, BIDET F, GAMBS S, et al. Local data debiasing for fairness based on generative adversarial training[J]. Algorithms, 2021, 14(3): 87.

[117] JALAL A, KARMALKAR S, HOFFMANN J, et al. Fairness for image generation with uncertain sensitive attributes[C]//Proceedings of the 38th International Conference on Machine Learning. Virtual Conference: PMLR, 2021: 4721-4732.

[118] KRISHNAN S, WANG J N, WU E, et al. ActiveClean: Interactive data cleaning for statistical modeling[J]. Proceedings of the VLDB Endowment, 2016, 9(12): 948-959.

[119] KRISHNAN S, FRANKLIN M J, GOLDBERG K, et al. BoostClean: automated error detection and repair for machine learning[EB/OL]. (2017-11-03) [2022-07-11]. https://arxiv.org/abs/1711.01299.

[120] SONG J, HE Y Y. Auto-validate: Unsupervised data validation using data-domain patterns inferred from data lakes[C]//Proceedings of the 2021 International Conference on Management of Data. Virtual Conference: ACM, 2021: 1678-1691.

[121] RUBINSTEIN B I P, NELSON B, HUANG L, et al. ANTIDOTE: understanding and defending against poisoning of anomaly detectors[C]//Proceedings of the 9th ACM SIGCOMM Conference on Internet Measurement. Chicago: ACM, 2009: 1-14.

[122] RAHM E, DO H. Data cleaning: Problems and current approaches[J]. IEEE Data Eng. Bull., 2000, 23: 3-13.

[123] FREDRIKSON M, LANTZ E, JHA S, et al. Privacy in pharmacogenetics: An end-to-end case study of personalized warfarin dosing[C]//Proceedings of the 23rd USENIX Security Symposium. Berkeley: USENIX Association, 2014, 2014: 17-32.

[124] HITAJ B, ATENIESE G, PEREZ-CRUZ F. Deep models under the GAN: Information leakage from collaborative deep learning[C]//Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security. Dallas: ACM, 2017: 603-618.

[125] ATENIESE G, FELICI G, MANCINI L V, et al. Hacking smart machines with smarter ones: How to extract meaningful data from machine learning classifiers [EB/OL]. (2013-06-19)[2022-07-11]. https://arxiv.org/abs/1306.4447.

[126] ERLINGSSON Ú, PIHUR V, KOROLOVA A. RAPPOR: randomized aggregatable privacy-preserving ordinal response[C]//Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security. Scottsdale: ACM, 2014: 1054-1067.

[127] SALEM A, ZHANG Y, HUMBERT M, et al. ML-leaks: Model and data independent membership inference attacks and defenses on machine learning models [EB/OL]. (2018-06-04)[2022-07-11]. https://arxiv.org/abs/1806.01246.

[128] 李强, 颜浩, 陈克非 . 安全多方计算协议的研究与应用[J]. 计算机科学, 2003, 30(8): 52-55.
LI Q, YAN H, CHEN K F. Research and application of secure multi-party computation protocols[J]. Computer Science, 2003, 30(8): 52-55. (in Chinese)

[129] YAO A C. Protocols for secure computations[C]//23rd Annual Symposium on Foundations of Computer Science. Chicago: IEEE, 1982: 160-164.

[130] GOLDREICH O, MICALI S, WIGDERSON A. How to play ANY mental game[C]//Proceedings of the Nineteenth Annual ACM Symposium on Theory of Computing. New York: ACM, 1987: 218-229.

[131] VAIDYA J, CLIFTON C. Privacy-preserving k-means clustering over vertically partitioned data[C]//Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. Washington: ACM, 2003: 206-215.

[132] MEHNAZ S, BELLALA G, BERTINO E. A secure sum protocol and its application to privacy-preserving multiparty analytics[C]//Proceedings of the 22nd ACM on Symposium on Access Control Models and Technologies. Indianapolis: ACM, 2017: 219-230.

[133] MOHASSEL P, ZHANG Y P. SecureML: A system for scalable privacy-preserving machine learning[C]//2017 IEEE Symposium on Security and Privacy. San Jose: IEEE, 2017: 19-38.

[134] ROUHANI B D, RIAZI M S, KOUSHANFAR F. DeepSecure: scalable provably-secure deep learning[C]//55th ACM/ESDA/IEEE Design Automation Conference. San Francisco: IEEE, 2018: 1-6.

[135] KONEČNÝ J, MCMAHAN H B, YU F X, et al. Federated learning: Strategies for improving communication efficiency[EB/OL]. (2016-10-18)[2022-07-11]. https://arxiv.org/abs/1610.05492.

[136] MCMAHAN H B, RAMAGE D, TALWAR K, et al. Learning differentially private recurrent language models[EB/OL]. (2017-10-18)[2022-07-11]. https://arxiv.org/abs/1710.06963.

[137] WENG J S, WENG J, ZHANG J L, et al. DeepChain: Auditable and privacy-preserving deep learning with blockchain-based incentive[J]. IEEE Transactions on Dependable and Secure Computing, 2021, 18(5): 2438-2455.

[138] GOEL K, RAJANI N, VIG J, et al. Robustness gym: Unifying the NLP evaluation landscape[EB/OL]. (2021-01-13)[2022-07-11]. https://arxiv.org/abs/2101.04840.

[139] PAULI P, KOCH A, BERBERICH J, et al. Training robust neural networks using lipschitz bounds[J]. IEEE Control Systems Letters, 2022, 6: 121-126.

[140] PEI K X, CAO Y Z, YANG J F, et al. DeepXplore: automated whitebox testing of deep learning systems[C]// Proceedings of the 26th Symposium on Operating Systems Principles. Shanghai: ACM, 2017: 1-18.

[141] GUO J M, JIANG Y, ZHAO Y, et al. DLFuzz: Differential fuzzing testing of deep learning systems[C]//Proceedings of the 2018 26th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering. Lake Buena Vista: ACM, 2018: 739-743.

[142] MURPHY C, KAISER G, HU L F, et al. Properties of machine learning applications for use in metamorphic testing[C]//Proceedings of the Twentieth International Conference on Software Engineering & Knowledge Engineering. San Francisco: Knowledge Systems Institute Graduate School, 2008: 867-872.

[143] XIE X Y, ZHANG Z Y, CHEN T Y, et al. METTLE: A METamorphic testing approach to assessing and validating unsupervised machine learning systems[J]. IEEE Transactions on Reliability, 2020, 69(4): 1293-1322.

[144] JIANG M Y, CHEN T Y, WANG S. On the effectiveness of testing sentiment analysis systems with metamorphic testing[J]. Information and Software Technology, 2022, 150: 106966.

[145] MA S Q, LIU Y Q, LEE W C, et al. MODE: Automated neural network model debugging via state differential analysis and input selection[C]//Proceedings of the 2018 26th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering. Lake Buena Vista: ACM, 2018: 175-186.

[146] YU B, QI H, GUO Q, et al. DeepRepair: Style-guided repairing for deep neural networks in the real-world operational environment[J]. IEEE Transactions on Reliability, 2022, 71(4): 1401-1416.

[147] SUN Z Y, ZHANG J M, HARMAN M, et al. Automatic testing and improvement of machine translation[C]//Proceedings of the ACM/IEEE 42nd International Conference on Software Engineering. Seoul: ACM, 2020: 974-985.

[148] WARDAT M, LE W, RAJAN H. DeepLocalize: Fault lo-

calization for deep neural networks[C]//2021 IEEE/ACM 43rd International Conference on Software Engineering. Madrid: IEEE, 2021: 251-262.

[149] TRAMÈR F, ATLIDAKIS V, GEAMBASU R, et al. FairTest: Discovering unwarranted associations in data-driven applications[C]//2017 IEEE European Symposium on Security and Privacy. Paris: IEEE, 2017: 401-416.

[150] ANGELL R, JOHNSON B, BRUN Y, et al. Themis: automatically testing software for discrimination[C]//Proceedings of the 2018 26th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering. Lake Buena Vista: ACM, 2018: 871-875.

[151] UDESHI S, ARORA P, CHATTOPADHYAY S. Automated directed fairness testing[C]//Proceedings of the 33rd ACM/IEEE International Conference on Automated Software Engineering. Montpellier: ACM, 2018: 98-108.

[152] BLACK E, YEOM S, FREDRIKSON M. FlipTest: Fairness testing via optimal transport[C]//Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency. Barcelona: ACM, 2020: 111-121.

[153] KAMIRAN F, MANSHA S, KARIM A, et al. Exploiting reject option in classification for social discrimination control[J]. Information Sciences, 2018, 425: 18-33.

[154] YANG Z, JAIN H, SHI J K, et al. BiasHeal: On-the-fly black-box healing of bias in sentiment analysis systems [C]//2021 IEEE International Conference on Software Maintenance and Evolution. Luxembourg: IEEE, 2021: 644-648.

[155] SLACK D, FRIEDLER S A, SCHEIDEGGER C, et al. Assessing the local interpretability of machine learning models[EB/OL]. (2019-02-09) [2022-07-11]. https://arxiv.org/abs/1902.03501.

[156] ZHOU Z Q, SUN L Q, CHEN T Y, et al. Metamorphic relations for enhancing system understanding and use[J]. IEEE Transactions on Software Engineering, 2020, 46 (10): 1120-1154.

[157] MOLNAR C. Interpretable Machine Learning[M]. Morrisville: Lulu Press, 2019.

[158] CHEN H J, JI Y F. Learning variational word masks to improve the interpretability of neural text classifiers[EB/OL]. (2020-10-01) [2022-07-11]. https://arxiv. org/abs/2010.00667.

[159] DING Z Y, WANG Y X, WANG G H, et al. Detecting violations of differential privacy[C]//Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security. Toronto: ACM, 2018: 475-489.

[160] CARLINI N, JAGIELSKI M, MIRONOV I. Cryptanalytic extraction of neural network models[C]//Annual International Cryptology Conference. Santa Barbara: Springer, 2020: 189-218.

[161] LIU J, JUUTI M, LU Y, et al. Oblivious neural network predictions via MiniONN transformations[C]//Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security. Dallas: ACM, 2017: 619-631.

[162] RUAN W J, WU M, SUN Y C, et al. Global robustness evaluation of deep neural networks with provable guarantees for the L0 norm[EB/OL]. (2018-04-16) [2022-07-11]. https://arxiv.org/abs/1804.05805.

[163] MANGAL R, NORI A V, ORSO A. Robustness of neural networks: A probabilistic and practical approach[C]// 2019 IEEE/ACM 41st International Conference on Software Engineering: New Ideas and Emerging Results. Montreal: IEEE, 2019: 93-96.

[164] LORENZ T, RUOSS A, BALUNOVIĆ M, et al. Robustness certification for point cloud models[C]//2021 IEEE/CVF International Conference on Computer Vision. Montreal: IEEE, 2021: 7588-7598.

[165] BHOJANAPALLI S, CHAKRABARTI A, GLASNER D, et al. Understanding robustness of transformers for image classification[C]//2021 IEEE/CVF International Conference on Computer Vision. Montreal: IEEE, 2021: 10211-10221.

[166] MADRY A, MAKELOV A, SCHMIDT L, et al. Towards deep learning models resistant to adversarial attacks[EB/OL]. (2017-06-19) [2022-07-11]. https://arxiv.org/abs/1706.06083.

[167] CARLINI N, KATZ G, BARRETT C, et al. Provably minimally-distorted adversarial examples[EB/OL]. (2017-09-29)[2022-07-11]. https://arxiv.org/abs/1709.10207.

[168] KURAKIN A, GOODFELLOW I, BENGIO S. Adversarial examples in the physical world[EB/OL]. (2016-07-08)[2022-07-11]. https://arxiv.org/abs/1607.02533.

[169] LEE H, HAN S, LEE J. Generative adversarial trainer: Defense to adversarial perturbations with GAN[EB/OL]. (2017-05-09) [2022-07-11]. https://arxiv.org/abs/1705.03387.

[170] WANG J Y, CHEN J L, SUN Y C, et al. RobOT: Robustness-oriented testing for deep learning systems[C]// 2021 IEEE/ACM 43rd International Conference on Software Engineering. Madrid: IEEE, 2021: 300-311.

[171] KIM J, FELDT R, YOO S. Guiding deep learning system testing using surprise adequacy[C]//2019 IEEE/ACM 41st International Conference on Software Engineering. Montrea: IEEE, 2019: 1039-1049.

[172] XU H, CARAMANIS C, MANNOR S. Robustness and regularization of support vector machines[J]. Journal of Machine Learning Research, 2008, 10: 1485-1510.

[173] DEMONTIS A, RUSSU P, BIGGIO B, et al. On security and sparsity of linear classifiers for adversarial settings[C]//Joint IAPR International Workshops on Statistical Techniques in Pattern Recognition (SPR) and Structural and Syntactic Pattern Recognition (SSPR). Mérida: Springer, 2016: 322-332.

[174] CHEN H, ZHANG H, BONING D, et al. Robust decision trees against adversarial examples[C]//International Conference on Machine Learning. Florida: PMLR, 2019: 1122-1131.

[175] XIE X Y, HO J W K, MURPHY C, et al. Testing and validating machine learning classifiers by metamorphic testing[J]. The Journal of Systems and Software, 2011, 84(4): 544-558.

[176] DWARAKANATH A, AHUJA M, SIKAND S, et al. Identifying implementation bugs in machine learning based image classifiers using metamorphic testing[C]// Proceedings of the 27th ACM SIGSOFT International Symposium on Software Testing and Analysis. Amsterdam: ACM, 2018: 118-128.

[177] AL-AZANI S, HASSINE J. Validation of machine learning classifiers using metamorphic testing and feature selection techniques[C]//International Workshop on Multidisciplinary Trends in Artificial Intelligence. Gadong: Springer, 2017: 77-91.

[178] MA L, JUEFEI-XU F, ZHANG F Y, et al. DeepGauge: multi-granularity testing criteria for deep learning systems[C]//2018 33rd IEEE/ACM International Conference on Automated Software Engineering. Montpellier: IEEE, 2018: 120-131.

[179] SUN Y C, WU M, RUAN W J, et al. Concolic testing for deep neural networks[C]//Proceedings of the 33rd ACM/IEEE International Conference on Automated Software Engineering. Montpellier: ACM, 2018: 109-119.

[180] TIAN Y C, PEI K X, JANA S, et al. DeepTest: Automated testing of deep-neural-network-driven autonomous cars[C]//Proceedings of the 40th International Conference on Software Engineering. Gothenburg: ACM, 2018: 303-314.

[181] BRAIEK H BEN, KHOMH F. DeepEvolution: A search-based testing approach for deep neural networks[C]// 2019 IEEE International Conference on Software Maintenance and Evolution. Cleveland: IEEE, 2019: 454-458.

[182] YAN S N, TAO G H, LIU X W, et al. Correlations between deep neural network model coverage criteria and model quality[C]//Proceedings of the 28th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering. Virtual Conference: ACM, 2020: 775-787.

[183] GERASIMOU S, ENISER H F, SEN A, et al. Importance-driven deep learning system testing[C]//Proceedings of the ACM/IEEE 42nd International Conference on Software Engineering: Companion Proceedings. Seoul: ACM, 2020: 322-323.

[184] XIE X F, MA L, WANG H J, et al. DiffChaser: Detecting disagreements for deep neural networks[C]//Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence. California: International Joint Conferences on Artificial Intelligence Organization, 2019: 5772-5778.

[185] YANG W, XIE T. Telemade: A testing framework for learning-based malware detection systems[C]//Workshops at the Thirty-Second AAAI Conference on Artificial Intelligence. Palo Alto: AAAI Press, 2018: 400-403.

[186] CHEN T Y, POON P L, QIU K, et al. Use of metamorphic relations as knowledge carriers to train deep neural networks[EB/OL]. (2021-04-10) [2022-07-11]. https://arxiv.org/abs/2104.04718.

[187] XIE X, GUO W, MA L, et al. RNNrepair: Automatic RNN repair via model-based analysis[C]//Proceedings of the 38th International Conference on Machine Learning. Virtual Conference: PMLR.org, 2021: 11383-11392.

[188] AGGARWAL A, LOHIA P, NAGAR S, et al. Black box fairness testing of machine learning models[C]//Proceedings of the 2019 27th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering. Tallinn: ACM, 2019: 625-635.

[189] ZHANG P, WANG J, SUN J, et al. Automatic Fairness

Testing of Neural Classifiers through Adversarial Sampling[J]. IEEE Transactions on Software Engineering, 2022: 3593-3612.

[190] ZHANG P X, WANG J Y, SUN J, et al. White-box fairness testing through adversarial sampling[C]//2020 IEEE/ACM 42nd International Conference on Software Engineering. Seoul: IEEE, 2020: 949-960.

[191] DOSHI-VELEZ F, KIM B. Towards a rigorous science of interpretable machine learning[EB/OL]. (2017-02-28)[2022-07-11]. https://arxiv.org/abs/1702.08608.

[192] CHENG C H, N¨HRENBERG G, HUANG C H, et al. Towards dependability metrics for neural networks[C]//2018 16th ACM/IEEE International Conference on Formal Methods and Models for System Design. Beijing: IEEE, 2018: 1-4.

[193] ROSS A, CHEN N N, HANG E Z, et al. Evaluating the interpretability of generative models by interactive reconstruction[C]//Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems. Yokohama: ACM, 2021: 1-15.

[194] SCHIELZETH H. Simple means to improve the interpretability of regression coefficients[J]. Methods in Ecology and Evolution, 2010, 1(2): 103-113.

[195] CHEN W J, SAHINER B, SAMUELSON F, et al. Calibration of medical diagnostic classifier scores to the probability of disease[J]. Statistical Methods in Medical Research, 2018, 27(5): 1394-1409.

[196] KOKHLIKYAN N, MIGLANI V, MARTIN M, et al. Captum: A unified and generic model interpretability library for PyTorch[EB/OL]. (2020-09-16) [2022-07-11]. https://arxiv.org/abs/2009.07896.

[197] YANG Z J, WANG B H, LI H R, et al. On detecting growing-up behaviors of malicious accounts in privacy-centric mobile social networks[C]//Annual Computer Security Applications Conference. Virtual Conference: ACM, 2021: 297-310.

[198] BICHSEL B, GEHR T, DRACHSLER-COHEN D, et al. DP-finder: Finding differential privacy violations by sampling and optimization[C]//Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security. Toronto: ACM, 2018: 508-524.

[199] TRAMÈR F, ZHANG F, JUELS A, et al. Stealing machine learning models via prediction APIs[C]//25th USENIX security symposium (USENIX Security 16). Berkeley: USENIX Association, 2016: 601-618.

[200] WANG B H, GONG N Z. Stealing hyperparameters in machine learning[C]//2018 IEEE Symposium on Security and Privacy. San Francisco: IEEE, 2018: 36-52.

[201] JAGIELSKI M, CARLINI N, BERTHELOT D, et al. High accuracy and high fidelity extraction of neural networks[C]//Proceedings of the 29th USENIX Conference on Security Symposium. Virtual Conference: USENIX Association, 2020: 1345-1362.

[202] XIE P T, BILENKO M, FINLEY T, et al. Crypto-nets: Neural networks over encrypted data[EB/OL]. (2014-12-18)[2022-07-11]. https://arxiv.org/abs/1412.6181.

[203] GILAD-BACHRACH R, DOWLIN N, LAINE K, et al. Cryptonets: Applying neural networks to encrypted data with high throughput and accuracy[C]//Proceedings of the 33nd International Conference on Machine Learning. Virtual Conference: JMLR.org, 2016: 201-210.

[204] HESAMIFARD E, TAKABI H, GHASEMI M. CryptoDL: Deep neural networks over encrypted data[EB/OL]. (2017-11-14)[2022-07-11]. https://arxiv.org/abs/1711.05189.

[205] LINDELL Y, PINKAS B. Privacy preserving data mining[C]//Advances in Cryptology — CRYPTO 2000. California: Springer, 2000: 36-54.

[206] JUVEKAR C, VAIKUNTANATHAN V, CHANDRAKASAN A. GAZELLE: A low latency framework for secure neural network inference[C]//27th USENIX Security Symposium (USENIX Security 18). Berkeley: USENIX Association, 2018: 1651-1669.

[207] CHANDRAN N, GUPTA D, RASTOGI A, et al. EzPC: Programmable and efficient secure two-party computation for machine learning[C]//2019 IEEE European Symposium on Security and Privacy. Stockholm: IEEE, 2019: 496-511.

[208] ZHENG W, DENG R, CHEN W, et al. Cerebro: A platform for multi-party cryptographic collaborative learning [C]//30th USENIX Security Symposium (USENIX Security 21). Berkeley: USENIX Association, 2021: 2723-2740.

[209] KNOTT B, VENKATARAMAN S, HANNUN A, et al. Crypten: Secure multi-party computation meets machine learning[C]//Advances in Neural Information Processing Systems. Virtual Conference: Curran Associates, Inc., 2021: 4961-4973.

[210] ISLAM M J, NGUYEN G, PAN R, et al. A comprehensive study on deep learning bug characteristics[C]//Proceedings of the 2019 27th ACM Joint Meeting on Euro-

pean Software Engineering Conference and Symposium on the Foundations of Software Engineering. Tallinn: ACM, 2019: 510-520.

[211] JIA L, ZHONG H, WANG X Y, et al. An empirical study on bugs inside tensorflow[C]//International Conference on Database Systems for Advanced Applications. Jeju: Springer, 2020: 604-620.

[212] GU J Z, LUO X C, ZHOU Y F, et al. Muffin: Testing deep learning libraries via neural architecture fuzzing [EB/OL]. (2022-04-19)[2022-07-11]. https://arxiv.org/abs/2204.08734.

[213] MURPHY C, SHEN K, KAISER G. Automatic system testing of programs without test oracles[C]//Proceedings of the eighteenth international symposium on Software Testing and Analysis. Chicago: ACM, 2009: 189-200.

[214] DING J H, KANG X J, HU X H. Validating a deep learning framework by metamorphic testing[C]/Proceedings of the 2nd International Workshop on Metamorphic Testing. Buenos Aires: IEEE, 2017: 28-34.

[215] MA L, ZHANG F Y, SUN J Y, et al. DeepMutation: Mutation testing of deep learning systems[C]//2018 IEEE 29th International Symposium on Software Reliability Engineering. Memphis: IEEE, 2018: 100-111.

[216] XIE D N, LI Y T, KIM M, et al. DocTer: Documentation-guided fuzzing for testing deep learning API functions[C]//Proceedings of the 31st ACM SIGSOFT International Symposium on Software Testing and Analysis. Virtual Conference: ACM, 2022: 176-188.

[217] LIU J W, WEI Y X, YANG S, et al. Coverage-guided tensor compiler fuzzing with joint IR-pass mutation[J]. Proceedings of the ACM on Programming Languages, 2022, 6(OOPSLA1): 73(1-26).

[218] LIU L, WU Y Z, WEI W Q, et al. Benchmarking deep learning frameworks: Design considerations, metrics and beyond[C]//2018 IEEE 38th International Conference on Distributed Computing Systems. Vienna: IEEE, 2018: 1258-1269.

[219] SRISAKAOKUL S, WU Z, ASTORGA A, et al. Multiple-implementation testing of supervised learning software[C]//Workshops at the thirty-second AAAI conference on artificial intelligence. Palo Alto: AAAI Press, 2018: 384-391.

[220] WANG Z, YAN M, CHEN J J, et al. Deep learning library testing via effective model generation[C]//Proceedings of the 28th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering. Virtual Conference: ACM, 2020: 788-799.

[221] ZHANG X F, LIU J W, SUN N, et al. Duo: differential fuzzing for deep learning operators[J]. IEEE Transactions on Reliability, 2021, 70(4): 1671-1685.

[222] Keras. Keras 2.3.0: This is also the last major release of multi-backend Keras[EB/OL]. (2019-07-18) [2022-07-11]. https://github.com/keras-team/keras/releases/tag/2.3.0.

[223] MURPHY C, SHEN K, KAISER G. Using JML runtime assertion checking to automate metamorphic testing in applications without test oracles[C]//2009 International Conference on Software Testing Verification and Validation. Denver: IEEE, 2009: 436-445.

[224] WANG C J, SHEN J, FANG C R, et al. Accuracy measurement of deep neural network accelerator via metamorphic testing[C]//2020 IEEE International Conference on Artificial Intelligence Testing. Oxford: IEEE, 2020: 55-61.

[225] HU Q, MA L, XIE X F, et al. DeepMutation: A mutation testing framework for deep learning systems[C]// 2019 34th IEEE/ACM International Conference on Automated Software Engineering. San Diego: IEEE, 2019: 1158-1161.

[226] LUO W S, CHAI D, RUN X Y, et al. Graph-based fuzz testing for deep learning inference engines[C]//Proceedings of the 43rd International Conference on Software Engineering. Madrid: IEEE, 2021: 288-299.

[227] ZHANG X F, YANG Y L, FENG Y, et al. Software engineering practice in the development of deep learning applications[EB/OL]. (2019-10-08)[2022-07-11]. https://arxiv.org/abs/1910.03156.

[228] ZHANG Y H, CHEN Y F, CHEUNG S C, et al. An empirical study on TensorFlow program bugs[C]//Proceedings of the 27th ACM SIGSOFT International Symposium on Software Testing and Analysis. Amsterdam: ACM, 2018: 129-140.

[229] CHEN Z P, YAO H H, LOU Y L, et al. An empirical study on deployment faults of deep learning based mobile applications[C]//Proceedings of the 43rd International Conference on Software Engineering. Madrid: IEEE, 2021: 674-685.

[230] LAM A N, NGUYEN A T, NGUYEN H A, et al. Bug localization with combination of deep learning and information retrieval[C]//2017 IEEE/ACM 25th International

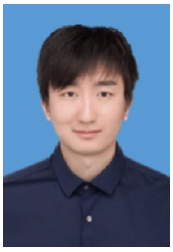
Conference on Program Comprehension. Buenos Aires: IEEE, 2017: 218-229.

[231] QI B H, SUN H L, YUAN W, et al. DreamLoc: A deep relevance matching-based framework for bug localization[J]. IEEE Transactions on Reliability, 2022, 71(1): 235-249.

作者简介

张笑宇 男,出生于1999年,河南郑州人. 西安交通大学网络空间安全学院博士研究生. 主要研究领域为人工智能软件测试.
E-mail: zxy0927@stu.xjtu.edu.cn

沈 超(通讯作者) 男,出生于1985年,重庆人. 博士,西安交通大学教授、博士生导师. 主要研究领域为可信人工智能、人工智能安全和信息物理系统安全.
E-mail: chaoshen@mail.xjtu.edu.cn



蔺琛皓 男,出生于1989年,陕西西安人. 博士,西安交通大学特聘研究员、博士生导师. 主要研究领域为人工智能安全、对抗机器学习、智能身份认证.
E-mail: linchenhao@xjtu.edu.cn

李 前 男,出生于1992年,陕西宝鸡人. 博士,西安交通大学助理教授. 主要研究领域为人工智能安全、对抗机器学习.
E-mail: qianlix@xjtu.edu.cn

王 骞 男,出生于1980年,湖北武汉人. 博士,武汉大学教授、博士生导师. 主要研究领域为人工智能安全、云计算安全与隐私、无线系统安全、应用密码学.
E-mail: qianwang@whu.edu.cn

李 琦 男,出生于1979年,浙江临安人. 博士,清华大学副教授、博士生导师. 主要研究领域为互联网和云安全、移动安全、机器学习与安全、大数据安全、区块链与安全.
E-mail: qli01@tsinghua.edu.cn

管晓宏 男,出生于1955年,四川泸州人. 博士,西安交通大学教授、博士生导师,中国科学院院士. 主要研究领域为网络信息安全、网络化系统、电力系统优化调度.
E-mail: xhguan@xjtu.edu.cn