软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software,2023,34(1):103−129 [doi: 10.13328/j.cnki.jos.006409]

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel: +86-10-62562563

人工智能软件系统的非功能属性及其质量保障方法综述*

叶仕俊, 张鹏程, 吉顺慧, 戴启印, 袁天昊, 任 彬

(河海大学 计算机与信息学院, 江苏 南京 211100)

通信作者: 张鹏程, E-mail: pchzhang@hhu.edu.cn

摘 要: 随着神经网络等技术的快速发展, 人工智能被越来越多地应用到安全关键或任务关键系统中, 例如汽车自动驾驶系统、疾病诊断系统和恶意软件检测系统等. 由于缺乏对人工智能软件系统全面和深入的了解, 导致系统时常发生严重错误. 人工智能软件系统的功能属性和非功能属性被提出以加强对人工智能软件系统的充分认识和质量保障. 经调研, 有大量研究者致力于功能属性的研究, 但人们越来越关注于人工智能软件系统的非功能属性. 为此, 专注于人工智能软件系统的非功能属性, 调研了 138 篇相关领域的论文, 从属性定义、属性必要性、属性示例和常见质量保障方法几个方面对目前已有的研究工作进行系统的梳理和详细的总结, 同时重新定义和分析了非功能属性之间的关系并介绍了人工智能软件系统研究中可以用到的开源工具. 最后, 展望了人工智能软件系统非功能属性的未来研究方向和挑战, 以期为该领域的研究人员提供参考.

关键词: 人工智能软件系统; 非功能属性; 质量保障; 神经网络

中图法分类号: TP311

中文引用格式: 叶仕俊, 张鹏程, 吉顺慧, 戴启印, 袁天昊, 任彬. 人工智能软件系统的非功能属性及其质量保障方法综述. 软件学报, 2023, 34(1): 103–129. http://www.jos.org.cn/1000-9825/6409.htm

英文引用格式: Ye SJ, Zhang PC, Ji SH, Dai QY, Yuan TH, Ren B. Survey on Non-functional Attributes for AI-enabled Software Systems and Quality Assurance Methods. Ruan Jian Xue Bao/Journal of Software, 2023, 34(1): 103–129 (in Chinese). http://www.jos.org.cn/1000-9825/6409.htm

Survey on Non-functional Attributes for AI-enabled Software Systems and Quality Assurance Methods

YE Shi-Jun, ZHANG Peng-Cheng, JI Shun-Hui, DAI Qi-Yin, YUAN Tian-Hao, REN Bin

(College of Computer and Information, Hohai University, Nanjing 211100, China)

Abstract: With the rapid development of neural network and other technologies, artificial intelligence has been widely applied in safety-critical or mission-critical systems, such as autopilot systems, disease diagnosis systems, and malware detection systems. Due to the lack of a comprehensive and in-depth understanding of artificial intelligence software systems, some errors with serious consequences occur frequently. The functional attributes and non-functional attributes of artificial intelligence software systems are proposed to enhance the adequate understanding and quality assurance of artificial intelligence software systems. After investigation, a large number of researchers are devoted to the study of functional attributes, but people are paying more and more attention to the non-functional attributes of artificial intelligence software systems. This paper investigates 138 papers in related fields, systematically combs the existing research results from the aspects of attribute necessity, attribute definition, attribute examples, and common quality assurance methods, and summarizes the research work on non-functional attributes of artificial intelligence software systems. At the same time, a summary and relationship analysis are presented on the non-functional attributes of artificial intelligence software systems. The open source tools that can be used in the research of artificial intelligence software system are surveyed. Finally, the thoughts on potential future research directions and challenges are summarized on non-functional attributes of artificial intelligence software systems, which, hopefully, will provide references

* 基金项目: 国家重点研发计划 (2018YFC0407901); 江苏省自然科学基金 (BK20191297); 中央高校基本科研业务费 (B210202075)
收稿时间: 2020-12-17; 修改时间: 2021-03-15, 2021-06-17; 采用时间: 2021-06-29; jos 在线出版时间: 2021-08-03
CNKI 网络首发时间: 2022-11-15

for researchers interested in the related directions.

Key words: artificial intelligence software systems; non-functional attributes; quality assurance; neural network

随着计算机硬件、云计算和移动互联等领域的重大突破, 人工智能 (artificial intelligence, AI) 被逐渐应用于包括癌细胞识别、农业土壤监测、广告精准投放、金融欺诈识别、信用评价以及自动驾驶汽车等[1]安全关键或任务关键系统中. 本文将这类模拟人的某些思维过程进行学习、推理、规划、思考等智能行为的系统统称为人工智能软件系统. 人工智能软件系统具有较为复杂的结构, 如数字图片识别系统中一个简单的 4 层卷积神经网络就具有超过 13 000 个参数, 一些人类无法发现的微小扰动就有可能导致神经网络出现相反的判断. 不幸的是, 由于缺乏对人工智能软件系统的全面认识和评估标准, 导致在高风险应用中时常发生一些重大错误, 引起了人们对于人工智能软件系统质量保障问题的关注和研究.

研究者从人工智能软件系统的分析、测试、验证等方面对人工智能软件系统的质量保障进行了探索[2]. 依据测试、验证的结果, 研究人员可以有依据、有方向地提出保障思路. 当前, 针对人工智能软件系统相比传统软件存在的差异问题, 研究人员对传统软件测试方法进行改进和创新, 产生了各式各样的测试方法. 如 Pei 等人提出了一种深度学习系统白盒测试方法——DeepXplore[3], 该方法通过使用生成的测试用例对模型进行重新训练来提高模型的准确率, Pei 等人在实验中首次使用神经元覆盖率作为对抗性样本生成的度量指标, 同时他们以错误行为发现次数和训练数据污染抵抗能力作为模型度量指标; Cisse 等人引入一种名为 Houdini 的对抗性示例生成方法[4], 该方法可基于语音识别、自然语言处理等应用模型的测试评估指标生成对抗性示例, 并且在实验中使用 PCKh (percentage of correctly detected keypoints)、结构相似性指数 (structural similarity index, SSIM) 和可感知性 (perceptibility) 作为模型方法的度量指标; Kurakin 等人利用快速梯度符号法 (fast gradient sign method, FGSM)、基础迭代法 (basic iterative method, BIM)、最相似迭代法 (iterative least-likely class method, ILCM) 等算法生成对抗性样本[5], Kurakin 等人还将所生成的对抗性样本打印并通过模拟真实世界中的目标分类场景进行测试, 并在实验中使用重构率 (destruction rate) 作为模型性能的度量指标.

测试、验证等方法多样化的同时也产生了系统评估标准不统一的问题, 为此研究人员在基于传统软件评估指标基础上逐渐形成了人工智能软件系统的属性指标. 该指标的提出一方面加强了从业人员对人工智能软件系统的认知, 另一方面提高了人工智能软件系统的质量保障力度. 人工智能软件系统的软件属性是指其为帮助用户实现目标或处理问题所需要的条件, 也是人工智能软件系统及其组件要满足标准和规范所要具备的要求. 这些属性指标对人工智能软件系统的性能维护和质量保障起到指向性的重要作用, 具体可分为功能属性 (functional attributes, FAs) 和非功能属性 (non-functional attributes, NFAs). 其中功能属性描述系统功能, 它是指人工智能软件系统为实现用户需求和业务需求所必须具备的能力. 而非功能属性反映系统质量、特性和约束. 作为功能属性的补充, 非功能属性是依赖于功能属性而存在的属性, 它们紧密约束和限制着人工智能软件系统, 对系统的质量保障具有非常重要的作用.

对人工智能软件系统使用非功能属性指标进行评估能指出系统存在的不足; 即人工智能软件系统需要在什么方面改进才能保障系统模型的质量. 当前研究中, 包括文献 [6,7] 在内的一些工作对人工智能软件系统的功能属性已进行一定程度的概括, 但尚且缺乏对非功能属性进行系统研究的相关工作. 为了填补这个空白, 本文将以人工智能软件系统非功能属性作为研究对象, 如图 1 所示, 详细地总结包含鲁棒性、安全性、数据隐私性、公平性、可解释性和可用性在内的非功能属性.

目前王赞等人[1]、Zhang 等人[6]和 Vinayagasundaram 等人[7]在针对人工智能领域的测试工作进行系统总结的同时, 对人工智能软件系统的非功能属性也进行了一定程度的讨论. 王赞等人从包括测试度量指标在内的多个角度来系统梳理深度神经网络测试的相关工作; Zhang 等人从范围更广的机器学习测试方面进行总结, 对机器学习测试流程、测试组件、测试属性和应用场景等部分作出总结; Vinayagasundaram 等人则从整个人工智能软件系统的体系结构角度对度量标准进行定义, 以衡量系统软件质量. 此外还有 Gilpin 等人[8]和 Mehrabi 等人[9]对人工智能软件系统的单个非功能属性进行系统地梳理和说明. 令人遗憾的是, 这几项相关工作的内容主要讲述人工智能研

究领域的测试和人工智能的体系结构, 对人工智能非功能属性的研究并未形成系统、有效的覆盖性总结. 本文将关注人工智能软件系统的非功能属性. 如图 2 所示, 将从属性定义、属性必要性、非功能属性的示例以及常见质量保障方法几个方面系统地梳理人工智能软件系统中常见的非功能属性. 其中, 属性定义以文字描述和公式表示两种形式给出非功能属性的定义; 属性必要性讲述非功能属性在人工智能软件系统开发中的重要性; 属性示例中以汽车自动驾驶系统等例子来呈现非功能属性; 常见质量保障方法部分总结现有的人工智能软件系统非功能属性的质量保障方法和研究思路. 本文还将对非功能属性之间的关系进行定义和总结分析, 并展望人工智能软件系统非功能属性的未来研究方向和挑战, 以期为该领域的研究人员提供参考.

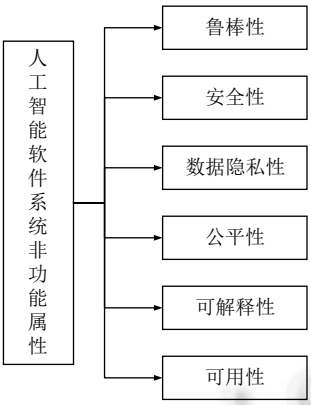

图 1 人工智能领域常见非功能属性

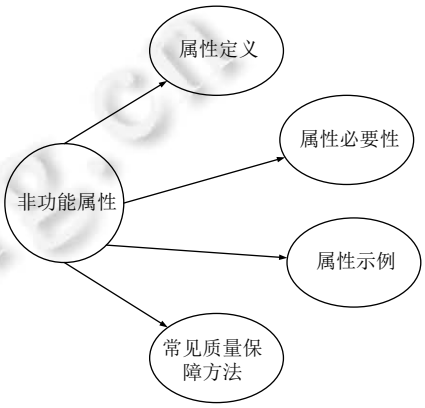

图 2 非功能属性研究范畴

为了对该研究问题进行系统的梳理和分析, 本文首先将“Non-Functional Attributes/Properties for Artificial Intelligence”“Robustness/Security/Data Privacy/Fairness/Interpretability/Usability of Artificial Intelligence”等设为搜索关键词, 在国内外重要的学术搜索引擎 (例如 Google 学术搜索、Springer、DBLP、CiteSeerX、CNKI 等) 中检索出相关论文; 随后, 我们筛选并移除了与该综述问题无关的论文. 有 3 名熟悉人工智能软件系统的研究人员参与了这一过程, 并通过讨论来消除分歧; 接着, 通过查阅论文中的相关工作和研究人员的已发表的论文列表, 以及通过已搜索到的文献的引用和被引用获取更多相关论文; 本文最终确定并引用了有关论文 138 篇. 论文的发表时间概况如图 3 所示, 最终所选中的论文有 75 篇发表在 CCF 评级为 A, B 的各领域顶级期刊和会议上, 其中人工智能领域 (38 篇)、软件工程及系统和程序设计语言领域 (15 篇)、计算机科学理论领域 (1 篇)、网络与信息安全领域 (18 篇)、计算机体系结构及平行分布计算和存储系统领域 (1 篇)、数据库及数据挖掘和内容检索领域 (1 篇)、人机交互与普适计算领域 (1 篇). 除此之外, 本文引用的文献还包括 arXiv (17 篇)、CCF 评级为 C 类会议期刊论文 (7 篇)、软件学报 (2 篇)、其他 CCF 未收录的会议和期刊论文 (32 篇) 和相关书籍 (5 本). 图 4 展示了不同非功能属性的论文分布情况.

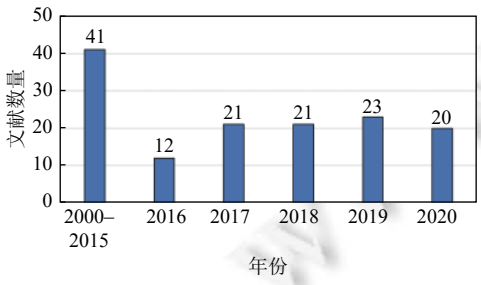

图 3 论文发表时间概况

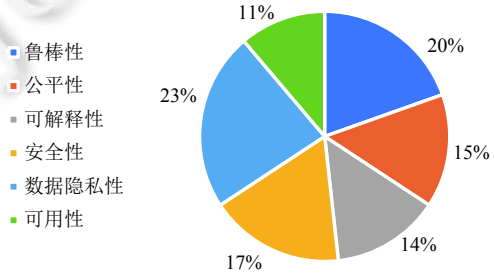

图 4 不同非功能属性研究工作占比

本文第 1 节从属性定义、属性必要性、示例和常见质量保障方法几个方面对人工智能软件系统的非功能属性进行详细的总结梳理; 第 2 节对人工智能软件系统非功能属性进行归纳, 并对它们之间存在的关系进行探讨; 第 3 节对本文所列举的人工智能软件系统非功能属性的现有工具或开源项目进行总结; 第 4 节总结全文, 并展望非功能属性的未来研究方向和挑战.

1 人工智能软件系统非功能属性

在人工智能软件系统发展前期, 人们过于关注人工智能软件系统的功能属性, 从而对于非功能属性的认知远远低于功能属性, 这不利于对人工智能软件系统的认知和保障工作. 近年来, 人们越来越关注非功能属性的研究, 将其作为人工智能软件发展的新突破.

常见的人工智能软件系统功能属性有正确性 (correctness) 和过拟合程度 (overfitting). 正确性是系统能够正确处理各种输入并产生正确输出的能力 (概率)[6]. 正确性的研究主要集中于数据[10]、模型[11]和统计方法[12,13]这 3 个方面. 过拟合程度是衡量模型算法由于过度的拟合当前可用数据而导致无法很好地拟合未来数据或者可靠地预测未来结果的程度指标[14]. 关于过拟合的研究主要集中于产生过拟合的原因 (复杂的模型[15]、稀疏的样本[16−18]) 以及模型过拟合验证[19,20]两个方向.

功能属性描述了人工智能软件系统所具有的实现各种需求的能力. 而非功能属性依赖于功能属性, 是系统特性和约束的表征. 人工智能软件系统常见的非功能属性有鲁棒性、安全性、数据隐私性、公平性、可解释性和可用性等. 本节将以属性定义、属性必要性、属性示例和常见质量保障方法 4 个方面对以上非功能属性进行详细的阐述.

1.1 鲁棒性

1.1.1 鲁棒性定义

当前, 不同的研究对鲁棒性 (robustness) 有不同的定义.

(1) 鲁棒性在 IEEE 软件工程标准术语中定义为[21,22]: “在无效输入或者压力环境下, 系统或者系统组件可以正确运行的程度.” 借鉴这一定义, 用公式 (1) 给出鲁棒性在人工智能软件系统中的定义; 假设 S 为人工智能软件系统; $E(S)$ 是 S 的正确性; $\vartheta(S)$ 是对人工智能软件系统 S 的任意组件 (例如数据、学习程序或者框架) 经过干扰后的系统, 则人工智能软件系统 S 的鲁棒性 r 就是 $E(S)$ 和 $E(\vartheta(S))$ 之差的度量, 即:

$$r = E(S) - E(\vartheta(S)) \tag{1}$$

因此, 鲁棒性衡量了人工智能软件系统对干扰的抵抗能力.

(2) Huber 从统计学的角度系统地给出了鲁棒性 3 个层面的概念[23]:

① 对于人工智能软件系统中所有学习模型的基本要求; 模型具有较高的精度或有效性;

② 对于模型所出现的较小偏差, 只能对算法性能产生较小的影响 (主要是噪声, noise);

③ 对于模型所出现的较大偏差, 不可对算法性能产生“毁灭性”的影响 (主要是离群点, outlier).

(3) 对抗鲁棒性是鲁棒性的常见类别, 其主要利用对抗性输入来增强系统的鲁棒性. 参照 Katz 等人[24]的工作可将对抗鲁棒性分为局部对抗鲁棒性和全局对抗鲁棒性如公式 (2), 公式 (3) 所示.

首先是局部对抗鲁棒性, 假设: x 是人工智能学习模型 h 的一个测试输入; x'是通过对 x 进行对抗扰动而生成的另一个测试输入; $h(x)$ 和 $h(x')$ 是 x、x'在模型 h 上的输出. 对于任意 x'如果满足公式 (2), 则称模型 h 在输入 x 处是 δ-局部鲁棒性的.

$$\forall x' : \|x - x'\|_p \leqslant \delta \rightarrow h(x) = h(x') \tag{2}$$

局部对抗鲁棒性要求在给定输入的领域内的所有样本都使用相同的标签进行分类. 换言之, 局部对抗鲁棒性所关注的是对一个特定测试输入的鲁棒性, 而全局对抗鲁棒性是针对所有输入的鲁棒性.

全局对抗鲁棒性定义如下, 假设: x 是人工智能学习模型 h 的一个测试输入; x'是通过对 x 进行对抗扰动而生成的另一个测试输入; $h(x)$ 和 $h(x')$ 是 x、x'在模型 h 上的输出. 如果任意的 x 和 x'满足公式 (3), 则称模型 h 是 ε-全局鲁棒性的:

$$\forall x, x' : \|x - x'\|_p \leqslant \delta \rightarrow h(x) - h(x') \leqslant \varepsilon \tag{3}$$

上述工作皆表明: 与正确性和过拟合程度不同, 鲁棒性是衡量系统在存在噪声干扰下的正确性[6,25], 即一个具有优异鲁棒性的系统在噪声干扰下应该能够很好地抵抗干扰、保持自身性能.

1.1.2 鲁棒性必要性

鲁棒性是人工智能软件系统的一种重要非功能属性[26,27]. 随着人工智能逐渐在决策和自主系统中扮演至关重要的角色, 例如在自动驾驶系统、恶意软件检测系统和医疗诊断系统中, 人工智能软件系统被发现在测试和应用中时常由于环境、输入等因素的变化导致系统出错. 正如 Hamon 等人的工作提出; 当前正在开发的人工智能系统距离实现自治系统所具有的最低鲁棒性、安全性要求还很遥远[28]. 如何提高系统鲁棒性以减少系统出错成为亟待解决的问题.

1.1.3 鲁棒性示例

为更清楚地解释鲁棒性, 本文以汽车自动驾驶系统 (autonomous driving system, ADS) 为示例. PC Magazine 将自动驾驶汽车定义为“由计算机控制的自动驾驶的汽车”[29]. 自动驾驶系统被应用于汽车上来保证在无人操作情况下, 汽车能够感知周围环境并且安全行驶. 关于自动驾驶系统的研发试验始于 20 世纪 20 年代, 并于 1977 年成功在日本实验了第一款半自动汽车. 近年来, 快速发展的自动驾驶汽车呈现出实用化趋势.

就在人们欣喜于汽车自动驾驶系统的市场化时, ADS 却经常暴露一些致命漏洞. 自动驾驶系统中的识别功能是汽车行驶过程中所做决策的重要依据之一, 包括了标志识别、人物识别和道路识别等. 但本田汽车公司所研发的汽车自动驾驶系统在一次测试中被发现; 日本拉面“天下一品”的商标和交通标志“禁止驶入”会被汽车错误识别为同一种标识; 在 2018 年亚利桑那州所发生的无人驾驶汽车致命事故[30]中, 系统模型将行人错误地识别为物体, 从而导致了错误的行驶决策. 本节所要研究的鲁棒性就是能够表示系统在干扰环境下维持本身正确性的属性.

1.1.4 常见质量保障方法

当前人工智能软件系统鲁棒性质量保障的主流方法为对抗学习, 依据现有的工作可以将对抗学习分为以下 2 种.

(1) 对抗性样本生成

基于 Szegedy 等人[31]和 Wang 等人[32]的工作, 攻击者通常可以通过对少量正确标记的输入进行扰动来构造使系统错误输出的输入, 这些输入被称之为对抗性样本. 现有的方法大都是通过研究和构造对抗性样本来增强系统模型的鲁棒性[33−35]. Carlini 等人[36]依据距离度量方法创建了一组攻击样本, 该方法对当前多种防御方法具有很强的攻击能力; Tjeng 等人[25]提出使用测试输入与其最接近的对抗示例之间的距离来衡量鲁棒性.

(2) 对抗性学习训练

这类方法通过在人工智能软件系统模型优化过程中增加噪声、随机性或对抗性损失来实现, 目的是使训练后的模型更加鲁棒. 比如 Gao 等人通过识别和删除 DNN 模型中不必要的功能来限制攻击者生成对抗性样本的能力[37]; Cheng 等人通过对抗稳定性训练来提高 NMT 模型的鲁棒性[38,39], 他们对神经机器翻译模型 (neural machine translation, NMT) 中的编码器和解码器加入词汇级别和特征级别的扰动, 训练的结果显示 NMT 的鲁棒性和翻译性能被大幅度地提升; Gehr 等人[40]和 Ross 等人[41]分别使用经典抽象解释和输入梯度正则化来增强模型的鲁棒性和可解释性; Kwiatkowska 等人[42]对深度神经网络的对抗性扰动进行研究, 他们介绍了针对深度神经网络所开发的自动验证和测试技术并利用该技术确保模型决策在有限输入扰动时的安全性和鲁棒性.

除对抗学习外, 还包括对人工智能软件系统鲁棒性建模和构建指标的方法. 比如 Fawzi 等人[43]对一般噪声情况下非线性分类器的鲁棒性进行了首次定量分析并证明了实验结论对于各种最新 DNN 和数据集的优异泛用性; Bastani 等人[44]提出了 3 种用于度量神经网络鲁棒性的指标, 并基于鲁棒性编码的线性程序设计了一种新算法来近似这些指标; Banerjee 等人[45]探索了使用贝叶斯深度学习来建模深度神经网络内部的误差传播, 从而在不进行大量故障注入实验的情况下, 对神经网络对硬件错误的敏感性进行数学建模.

现有的针对人工智能软件系统鲁棒性的质量保障方法研究仍然存在一些方面的问题. 一方面是方法的研究方向比较单一, 主要的研究基本聚焦于通过对抗学习来提升系统的鲁棒性, 同时大部分研究在关于对抗攻击、数据

集规模和消耗时间的权衡上也存在不足; 另一方面, 无论是对抗性样本的生成过程还是在对抗学习的训练过程中, 研究者们提出的算法过于地依赖某个特定的模型或环境, 虽然在该模型上能取得较好的效果, 但是算法在其泛用性和迁移性上存在一定的问题.

1.2 安全性

1.2.1 安全性定义

Zhang 等人在其工作中将安全性定义为[6]: 人工智能软件系统的安全性 (Security) 是指系统抵御通过操纵或非法访问模型学习组件造成的潜在伤害、危险或损失的能力.

同鲁棒性相似的是, 安全性也存在抵御外部攻击的要求; 不同的是, 鲁棒性是衡量系统受干扰后保持自身正确性的能力, 安全性则衡量保护系统内部组件不受破坏或降低破坏程度的能力. 也就是说, 鲁棒性的衡量将依据噪声等因素干扰下系统的正确性的程度, 而系统安全性的评估则更多地依赖于外部攻击对模型本身的破坏程度.

人工智能软件系统的安全性公式定义如公式 (4) 所示, 假设: S 为人工智能软件系统; $eval(S)$ 是系统 S 的安全评估函数; $\theta(S)$ 是被操纵或非法访问系统模型学习组件 (例如数据、学习程序或者框架) 之后的人工智能软件系统; 则系统的安全性为 $eval(S)$ 和 $eval(\theta(S))$ 之差的度量:

$$s = eval(S) - eval(\theta(S)) \tag{4}$$

1.2.2 安全性必要性

自动驾驶汽车的市场化虽然令人期待, 但目前的自动驾驶系统仍然存在许多问题, 包括系统无法在任意环境中保持绝对的决策正确性以及无法在任何情况下保证系统组件和功能安全. 前者属于鲁棒性的研究动因, 后者属于本节安全性的研究范围.

可以说, 安全问题是自动驾驶系统研发试验的核心. 对于安全关键或任务关键领域的人工智能软件系统来说, 安全性一直是衡量系统性能的一个非常重要的指标.

1.2.3 安全性示例

以汽车自动驾驶系统为例, Uber 和 Tesla 的自动驾驶汽车皆发生过由于识别系统故障而导致的严重交通安全事故. 频发的自动驾驶汽车事故, 引发了人们对于高风险人工智能应用的安全性担忧.

在传统的交通工具中, 汽车的行驶安全主要是人为操控, 再辅以另外一些安全功能 (例如车道保持, 紧急制动等) 设计, 以帮助和减少风险. 即人为控制和安全机制组成了传统汽车的安全保障体系. 在自动驾驶汽车中, 人工智能应用于包括感知、控制、路由等多个系统模块中. 自动驾驶系统采取的是归纳训练, 即没有明确的要求或设计见解. 这导致系统不可能涵盖所有未知案例, 当汽车行驶过程中出现未知案例时就有可能导致事故的发生.

当前自动驾驶汽车主要采用危险方位报警指示器来增加安全性, 采取了包括故障树分析 (FTA)、向前向后搜索 (FBS)、失效模式和影响分析 (FMEA)、危险与可操作性研究 (HAZOP) 在内的多种组合算法和策略.

1.2.4 常见质量保障方法

本小节将当前人工智能软件系统安全性质量保障的研究工作主要分为 3 类.

(1) 第 1 类对人工智能软件系统中常见的攻击方式进行研究, 以此来防御针对系统的大部分攻击.

第 1 种攻击为模型反演攻击. 当前许多互联网公司都推出了人工智能服务即开发平台, 为众多开发者提供便利的人工智能技术支持, 但同时也为攻击者提供了新的攻击方式. 由于模型是由一系列参数所决定, 攻击者可通过求解模型参数的攻击方式来实现模型剽窃, 从而对模型持有者造成损失. 这种攻击被称为模型反演攻击.

Tramèr 等人[46]在 2016 年的工作中提到这种攻击方式; 对于一个输入为 n 维的线性模型, 依据解方程的思想 (即一个 n 元方程, 至少拥有 n 个等式就能求解), 攻击者只需要通过 API 接口进行 $n+1$ 次查询便可以窃取到这个模型; Batina 等人[47]也证明了假设攻击者已知所使用的神经网络结构, 便可仅通过单发边信道测量就可以对神经网络的输入进行逆向工程; Yang 等人[48]通过辅助集和截断技术的有效结合, 提出了一种即使目标模型为黑盒也能进行模型反演的攻击方法.

第 2 种常见攻击方式是完整性攻击, 常见的完整性攻击有数据下毒攻击[49]和对抗样本攻击. 下毒攻击多发生

在模型的训练过程, 表现为攻击者修改模型现有的训练集或者增加额外的恶意数据[49], 最后影响模型的完整性和模型训练的结果; 对抗样本攻击发生在模型的预测阶段, 表现为在预训练数据中添加肉眼无法识别的微小扰动, 使模型的结果发生错误.

(2) 第 2 类从人工智能软件系统构建出发, 通过建立安全性论证体系、威胁分析及保护机制和安全性系统设计原则来加强人工智能系统的安全保障.

改进人工智能软件系统安全性贯穿于系统构建全过程. 即开发者要建立安全性论证体系; 识别相关的危害和安全要求、确定潜在的危害根源、针对每个危害制定一个缓解策略、提供证据证明缓解措施得到适当的实施.

在人工智能软件系统构建过程中, 开发者需要针对所设计的系统进行详尽的安全性考虑. 包括系统的安全需求(策略)、威胁模型分析、攻击面分析和系统保护机制分析. Papernot 等人[50]在其工作中对攻击面和威胁模型进行了讨论, 他们将系统决策过程中的攻击面分成了 4 个阶段; 获取物理世界信息阶段、信息转换成数字化表征形式阶段、表征进入机器学习模型训练阶段、模型返回物理世界进行推理预测阶段. 以汽车自动驾驶系统为例, 4 个攻击面对应于: 1) 利用行车传感器获取诸如停车的交通标志 (获取物理世界信息阶段); 2) 将得到的信息转化成图片进入预处理阶段 (转化成数字表征), 再将图像转化成 3 维的张量 (表征转化); 3) 应用模型对其进行分类置信度的计算 (表征进入机器学习模型训练阶段); 4) 反馈到物理世界中, 让车辆刹车 (模型返回物理世界进行推理预测阶段).

关于威胁模型, Suresh 提出了建立威胁模型的 5 个步骤[51].

第 1 步是分析系统、定义外部实体和要保护的资产. 安全性设计的第 1 步是为了了解系统运行的环境以及攻击者所要攻击的资产目标. 以汽车自动驾驶系统为例, 资产包括系统固件、系统数据和学习模型等, 外部实体则包括了将与人工智能软件系统交互的用户和潜在的攻击者或对手.

第 2 步是识别潜在对手、攻击面和威胁. 对于大部分人工智能软件系统来说, 潜在对手主要为远程软件攻击者和网络攻击者, 除此之外还包括通常很容易被忽略的恶意内部攻击者和攻击方式非常复杂的高级硬件攻击者. 威胁和系统安全级别的设置主要根据攻击的严重性来决定的. 目前确定系统安全威胁一般使用 STRIDE 模型(欺骗身份、篡改数据、否认、信息泄露、拒绝服务和提升权限), 通过分析 STRIDE 威胁模型, 可以揭示应用程序的泄漏和被攻击的范围. Yang 等人[52]在其工作中对 STRIDE 威胁模型的 6 种威胁类型进行了详细的讨论. 评估攻击的严重性可以使开发者能够适当的分配有限的开发资源, Houmb 等人[53]提出一种风险水平估算模型, 该模型可将风险水平作为频率和影响估算的条件概率得出. 频率和影响估算值是从“通用漏洞评分系统 (CVSS)”中得出的. 该模型适用于漏洞级别 (就像 CVSS 一样), 并且能够将漏洞组成服务级别.

第 3 步是确定应对威胁的高级安全目标. 针对不同的对象, 开发者可以设定不同的安全目标来保持系统的 6 个安全要素; 保密性、完整性、可利用性、真实性、安全生命周期以及不可否认性. 依据对人工智能软件系统发起的攻击类型来判断每个要素的风险. 基于以上信息, 开发者就基本可以确定人工智能软件系统将面对的威胁和可采取的对策.

第 4 步是为每个安全目标定义安全需求. 例如从“安全生命周期”的目标出发, 开发者可以确定系统的最小权限和孤立性原则、维持安全的系统状态、系统访问控制策略以及安全系统的初始化执行.

第 5 步是将所有信息整合进威胁摘要表. 将基于前 4 步工作得到的信息合并到威胁摘要表中, 最终得到当前人工智能软件系统的威胁模型.

开发者在设计系统时须遵守基本的安全设计原则.

- 最小权限原则; 应该给予系统组件实现其功能所需的最低权限. 目的是最大限度地降低受损组件的影响.
- 孤立性原则; 组件之间的交互作用不应超过必要的程度. 目标是减少可信计算基数 (TCB) 的大小.

最小权限原则将通过减少每个系统组件的非必要权限, 防止高权限组件受损引发系统崩溃. 即防止如图 5 所示, 当网络组件受到木马病毒等攻击而受损时, 由于网络组件的高权限将会导致整个系统的运行陷入混乱或停滞的情况. McGraw 等人在论述软件安全时提到在设计和架构层次上, 系统必须是一致的, 并呈现一个考虑到安全原则 (如最小特权原则) 的统一安全架构[54]; Gollmann 等人在其关于计算机安全的工作中[55]强调安全系统的设计应该遵循 Ref13 所制定的最小特权原则等保护原则.

孤立性原则要求尽量降低组件间的耦合度, 减少人工智能软件系统 TCB (trusted computing base) 的大小. 如图 6 所示, 由于组件间的低耦合, 网络组件遭受破坏所产生的影响将会被限制在有限的非 TCB 中, 不会对人工智能软件系统的安全造成实质的威胁. TCB 是系统中负责建立安全需求的组件, 如果任一 TCB 遭到破坏表示系统安全受到威胁, 相反非 TCB 组件遭到破坏但系统安全性仍然保留, 因而在较差的系统设计中, TCB 等同于整个系统.

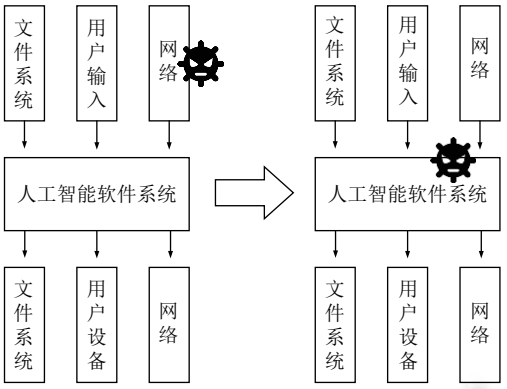

图 5 最小权限原则

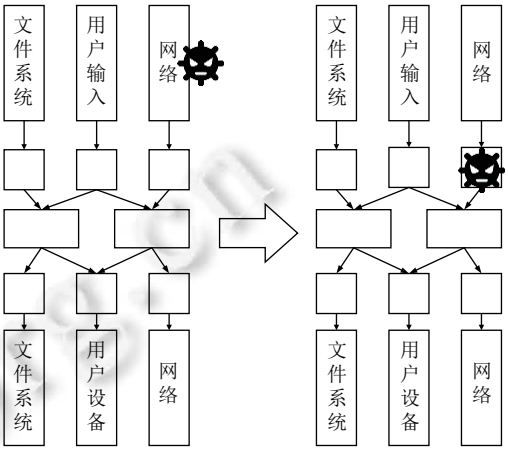

图 6 孤立性原则

(3) 第 3 类工作致力于为人工智能软件系统设计特定于任务的自定义安全协议.

由于经典的通用安全计算协议无法扩展至现实世界的人工智能应用中, 为了解决这个问题, 机器学习领域近几年的工作结合了不同的安全计算基数来设计特定于任务的自定义协议. 包括约束优化[56]和 K 近邻分类[57]等. Agrawal 等人[58]对 DNN 训练算法和安全协议两方面进行了优化, 提出了 QUOTIENT 方法. 该方法采用了归一化和自适应梯度法, 不仅在安全性上提升显著, 在训练速度方面相较之前的工作[59−61]也有明显的改善.

现有的针对人工智能软件系统安全性的质量保障方法研究存在几个不足. 首先, 虽然现有的研究已经提出了各种方法来增强人工智能软件系统的安全性, 但是对于安全性基准的研究依然缺少. 例如当研究人员在不同工作中使用了不同数据集时, 很难比较这些方法的优缺点. 其次, 过高的安全性将会带来更大的算法开销、更低的泛化性能问题, 这 3 个方面如何进行适当的平衡是在设计和构建人工智能软件系统时需要注意的.

1.3 数据隐私性

1.3.1 数据隐私性定义

人工智能软件系统中的隐私性 (privacy) 是系统保留私有数据信息的能力[50].

对于数据隐私性的公式定义, 本文采用 Dwork 等人工作中的差分隐私来定义[62]; 如公式 (5) 所示, 假定 A 是随机算法; D_1 和 D_2 是两个仅在一个实例上不同的训练数据集; S 是 A 的输出集的子集; 参数 ε 是隐私预算 (较小的预算会产生更强的隐私保证); 参数 δ 是故障率 (可以决定对于由 ε 所定义的边界不成立的容忍程度). 则差分隐私可用下列公式表示:

$$\Pr[A(D_1)\in S]\leqslant e^{\varepsilon}*\Pr[A(D_2)\in S]+\delta \tag{5}$$

基于上述定义, 可知差分隐私可以确保 D_1 和 D_2 仅在一个实例中有所不同时, 学习者不应该从 D_1 中获得比从 D_2 中更多的信息.

1.3.2 数据隐私性必要性

在传统的机器学习领域, 研究者们关注模型对于结果预测的准确性, 同时又希望模型在不同场景中都有较强的泛化能力 (指应用能力). 但在大数据时代, 数据的来源是复杂且敏感的, 个人的数据信息将直接同个人的隐私相关. 模型在增强泛化能力的同时扩大了使用范围也增加了泄露数据和信息的危险度.

为此欧盟通用数据保护法规 (GDPR)[63]和加利福尼亚 CCPA[64]等监管法规相继出台来保证人工智能软件系统

应用的数据隐私性.

1.3.3 数据隐私性示例

人工智能已经形成了一个商业模式, 机器学习即服务 (MLaaS) 领域的发展和应用为大量非专业的数据持有者提供了构建私有模型的快捷方式, 但随之也增加了用户隐私暴露的危险性. 原因在于数据持有者在使用 MLaaS 时可能使用到攻击者设计的可窃取数据的预训练模型, 攻击者可以通过解码模型参数轻松得到用户的隐私数据. 文献 [65] 中的工作也提到; 尽管机器学习平台是安全可靠的, 但机器学习模型提供者提供的算法未必可信.

在医学领域中, 人工智能技术主要被用于医疗诊断和药物推荐. 训练模型所需的大量医疗数据和病人的隐私信息联系紧密, 因而保护这些数据的隐私性是必要的. 但是, Fredrikson 等人在 2014 年针对药物推荐系统的研究工作却发现可以通过基于机器学习模型和病人的人口属性统计信息逆向推理得到病患的遗传资料[66]. Fredrikson 等人在 2015 年的研究工作中介绍了如何给定模型的输出结果去推断模型的输入数据[67]. 他们利用系统提供的查询接口进行大量的查询, 根据查询的输入和返回的输出构建了一个无限逼近原始模型的新模型, 最后基于新模型逆向恢复原始模型的训练数据.

除医学行业以外, 数据的隐私性对于汽车行业也是非常重要的. 未来自动驾驶汽车市场化后, 汽车行业将不仅成为数据的消费者, 还将成为数据的主要生产者. 据 Google 公司统计, 在其自动驾驶汽车实验中, 平均一辆汽车每秒产生了超过 1 GB 的数据. 如此庞大的数据涉及到汽车使用者非常详细的个人隐私信息, 这些数据对于汽车制造商、移动运营商、保险公司、酒店和其他希望与自动驾驶汽车或汽车使用者打交道的服务和产品提供者来说, 具有巨大的价值.

1.3.4 常见质量保障方法

目前人工智能软件系统数据隐私性的质量保障有以下几种常见方法.

(1) 对数据进行匿名化处理

这类方法主要用于一些涉及微数据的应用程序 (包括医学数据保护方面), 大多基于系统全局记录、本地记录和群集的匿名化方式使得系统中的数据记录无法相互区分来达到隐私保护的目的. Sweeney 等人提出了一种名为 k-anonymity 的隐私保护模型[68], 这种模型可以使得数据的持有者在保证数据仍然可用的前提下, 在发布其私有数据时他人无法从中得到敏感的个人信息. 匿名化方法的缺点在于匿名数据分析的工作量非常大, 因而这种方法无法总是实现有效的数据匿名, 并且现有的大部分方法未充分地考虑数据属性的实用性. 目前也有研究人员在进行数据匿名化方法的改进研究, 例如 Gao 等人基于多维匿名化的概念, 提出了一种基于属性顺序敏感的实用性匿名化方法[69].

(2) 防御数据窃取攻击的方法

相比其他领域, 机器学习体系结构更容易获得, 但是想要质量较高的标记数据进行训练的成本却是更高. 一般通过云服务上的预训练模型能够产生这些昂贵的标记数据, 但是也会使得攻击者能够复制模型来得到这些标记数据. 为了应对这个问题, 研究者们提出通过省略类别概率来限制基于云的模型所提供的信息, 但是这种方法会极大地影响模型的实用性. Papernot 等人提出了一种窃取神经网络模型以生成对抗性示例的方法[70]. 他们假定攻击者的训练数据量有限, 并使用基于雅可比的启发式方法来查找定义目标模型决策边界的对抗性示例. 据我们所知, 这是有效减轻模型数据窃取攻击的第一项研究. Lee 等人提出了一种方法[71], 该方法通过为输出添加扰动来抵御数据窃取攻击, 同时能保持模型的实用性. 他们在神经网络中增加了一个可以应用于大多数神经网络分类器并且可以添加微小可控扰动的激活层, 同时测试了各种可能的噪声形式, 发现“反向 Sigmoid”是最有效的防御方法. 经过 Lee 等人的评估, 这种方法要么完全阻止了窃取, 要么在攻击者了解防御的最坏情况下将窃取过程减慢了 64 倍; Prashar 等人[72]提出一种分布式的隐私保护方法, 该方法通过禁止用户共享其模型的参数来防止恶意攻击者破坏训练并复制用户的私人数据.

(3) 为人工智能模型制定隐私保护协议

Gascón 等人[73]提出了用于计算线性回归模型的隐私保护协议. Gascón 等人的工作改进了 Nikolaenko 等人[74]的隐私保护 Ridge 回归方法, 证明了其方法具有优异的可扩展性和极高的处理效率. Nikolaenko 等人在 2013 年的

工作[75]中使用矩阵分解方法来对诸如推荐系统中的用户数据隐私进行保护; 同样是对预测服务进行隐私保护, Liu 等人[76]提出 MiniONN——这是将现有神经网络转换为支持合理保护隐私的遗忘神经网络的第 1 种方法, Liu 等人的工作表明 MiniONN 在响应延迟和消息大小方面胜过现有工作.

现有的针对人工智能软件系统数据隐私性的质量保障方法研究存在几个不足. 首先, 由于人工智能系统的复杂程度非常高, 在系统数据的保护过程中掺杂了对庞大模型参数的密码计算, 使得现有针对人工智能软件系统的数据隐私保护效率较低, 在实时运行的人工智能系统中尤其明显. 其次, 对数据本身的加密保护是提高人工智能软件系统数据隐私性的常见方法, 与之相比较, 在模型的训练和推理阶段的隐私性研究则略显不足[77].

1.4 公平性

1.4.1 公平性定义

Corbett-Davies 等人将需要保护以防止导致不公平的特征称为受保护特征或受保护属性和敏感属性[78]. 法律认可的受保护特征包括种族、肤色、性别、宗教、国籍、国籍、年龄、怀孕、家庭状况、残疾状况、退伍军人身份和遗传信息.

目前有许多的工作[6,79−81]都对公平性 (fairness) 作了定义, 本文将对这些工作所述的公平性进行了分类, 主要包括一般公平性、群体公平性和个体公平性, 并使用公式进行定义. 假设需要对群体中的每个个体进行决策时, 使用 X 表示一组个体; Y 表示 X 的真实标签集合; S 为当前待测的已训练好的人工智能软件系统; B 为敏感属性集.

(1) 一般公平性: 一般性公平是指只要 S 的决策过程中没有明确使用敏感属性, 则该人工智能算法就是公平的. 这种定义对于人工智能软件系统中公平性的定义和确保公平成本相对较低. 但是有时 X 中的非受保护属性包含着与受保护属性相关的敏感信息[79,82]. 另外将受保护属性排除在外也会影响模型的正确性, 严重的将会产生无效的预测结果[83].

(2) 群体公平性: 群体公平性是指基于受保护属性所选择的群体所作的决策概率相等. 群体公平性中有两种常见的公平标准类型:

① 人口均等公平性[84]: 它要求 S 所作决策应独立于敏感属性. 以贷款问题为例, 人口均等公平性是指银行应该独立于种族和宗教等受保护属性以相同或者几乎相同的选择率向两个群体提供贷款. 假设 X_1 和 X_2 是 X 中基于敏感属性 $b \in B$ 所划分的两个群体组, 如果人工智能软件系统 S 满足公式 (6) 则称 S 存在人口均等公平性.

$$P\{S(x_i)=1|x_i \in X_1\} - P\left\{S(x_j)=1|x_j \in X_2\right\} < \varepsilon \tag{6}$$

其中, ε 为 0 或者无线接近于 0.

② 机会均等公平性和均等赔率公平性: 这种群体公平性是 Hardt 等人提出的[85]. 和人口均等公平性所不同的是, 均等赔率公平性增加了一个固定决策结果标签 $y \in Y$ 为前提. 如果人工智能软件系统 S 独立于敏感属性和决策结果标签且满足公式 (7) 则称 S 存在均等赔率公平性,

$$P\{S(x_i)=1|x_i \in X_1, Y=y\} = P\left\{S(x_j)=1|x_j \in X_2, Y=y\right\} \tag{7}$$

这意味着对于受保护和不受保护的群体, 积极类别中的群体被正确分配为积极结果的可能性和负面类别中的群体被错误分配为积极结果的可能性都应相同[80]. 换言之, 均等赔率公平性指出, 受保护和不受保护的群体对真阳性和假阳性的比率应相等.

当决策结果的标签 y 为 1 (1 代表积极) 时, 均等赔率公平性也称为机会均等公平性[85]. 以贷款问题为例, 机会均等公平性要求即使两个群体组的真实盈利率相等, 银行也要依据每个群体中按时偿还贷款的客户比例分别制定选择率. 如果人工智能软件系统 S 独立于敏感属性、决策结果标签 y 为 1 且满足公式 (8) 则称存在机会均等公平性.

$$P\{S(x_i)=1|x_i \in X_1, Y=y=1\} = P\left\{S(x_j)=1|x_j \in X_2, Y=y=1\right\} \tag{8}$$

这意味着对于受保护和不受保护的群体, 积极类别的群体被分配为积极结果的可能性应该相等[80]. 换言之, 机会均等公平性指出受保护和不受保护的群体应具有相等的真阳性比率.

(3) 个体公平性: 个体公平性是指群体中的相似个体在相同敏感属性环境中所做决策相同. Dwork 等人在其工

作[86]中提出了使用基于任务的相似性度量来描述相似个体对. 根据 Dwork 等人的工作, 可以将个体公平性定义为人工智能软件系统 S 要保证两个相似的个体 x_1、 $x_2 \in X$ 通过 S 获得的决策相似. 如公式 (9) 所示:

$$P\{S(x_1)|x_1 \in X\} = P\{S(x_2)|x_2 \in X\},\ \text{if}\ dis(x_1, x_2) < \epsilon \tag{9}$$

其中, dis 是衡量个人相似度的距离衡量标准.

1.4.2 公平性必要性

公平性也是属于人工智能软件系统的一个非常重要的属性. 以机器学习为例, 机器学习算法被广泛用于电影推荐、贷款申请[87]、商业活动和雇佣[88,89]. 模型可以学习人类的教学内容 (即以训练数据的形式), 而人类可能会对认知产生偏见, 从而影响收集或标记的数据以及设计的算法, 最终导致偏见问题产生. 因此, 由人工智能软件系统做出的决定要保证以正确的方式和合理的理由来避免人权、歧视、法律和其他道德问题, 确保不同使用者基于上述领域的公平.

人工智能软件系统的不公平将会导致两个后果. 第 1 种危害是影响社会资源或机会的公平分配[90]. 不公平的决策损害了受歧视人群获得社会援助的机会. 第 2 种危害是会损害受歧视人群的正面印象, 降低人们对受歧视人群的身份认同[91]. 不公平的决策增加了不同人群对不公平现象的接受能力, 同时也加固了对受歧视人群的负面印象.

1.4.3 公平性示例

肤色和性别是人工智能软件系统最常见的不公平特征. 类似的研究有: Buolamwini 等人[90]引入一个按性别和肤色类型均平衡的面部数据集, 他们使用该数据集评估了 3 种商业性别分类系统, 结果表明肤色较黑的女性是分类错误最多的群体 (错误率高达 34.7%), 肤色较浅的男性的最大错误率仅为 0.8%; Angwin 等人[92]对犯罪风险评估工具 COMPAS 进行评估发现, COMPAS 认为黑色人种有 44.9% 的可能性再犯罪, 白色人种只有 23.5% 的再犯罪概率. 实际上黑色人种的再犯罪概率为 28.0%, 而白色人种为 47.7%. 上述现象表明工具 COMPAS 对不同肤色人群存在歧视决策; Wang 等人[93]发现软件开发团队的男女比例对所研发软件的性别偏见产生了较大的影响.

公平性问题在自动驾驶系统中主要表现为两类伦理问题. 第 1 类是当面临危险情况时, 不同人群的保护优先级问题. 这一类属于哲学中经典的火车司机困境问题; 即在车辆无法刹车时, 自动驾驶系统选择向前行驶撞向 5 个行人还是转向岔道撞向一人. 第 2 类是在法律或道德的约束下, 不同人群对系统的使用权限问题. 这一类的示例是车上乘客突发疾病, 自动驾驶系统是否可以为了把乘客及时送到医院进行闯红灯等操作. 两类伦理问题如何有效地消除已成为汽车自动化驾驶系统市场化的迫切需求.

1.4.4 常见质量保障方法

本节从不公平性来源、模型训练过程的公平操作和公平系统设计要求 3 个方面对人工智能系统公平性质量保障进行梳理, 同时总结了当前人工智能软件系统公平性质量保障的常见方法, 包括中间表示、竞争学习、因果分析、公平测试和不公平缓解.

了解系统不公平的原因有利于研究者从源头对其进行消除和遏制. 根据 Barocas 等人的工作[94]存在以下 5 个不公平性的原因.

(1) 倾斜的样本: 一旦样本发生指向性的初始偏差, 这种偏差可能会随着时间的流逝而加剧;

(2) 被污染的标签: 数据标签由于人为打标签的偏向性而产生偏差;

(3) 有限的特征: 可能缺乏充足的信息或无法可靠地收集特征, 从而在建立特征与标签之间的联系时误导了模型;

(4) 样本规模的差异: 如果少数群体和多数群体的数据高度不平衡, 则很难很好地模拟少数群体;

(5) 代理: 某些特征是敏感属性的代理, 即使排除了敏感属性也可能导致模型出现偏差.

模型的训练过程是人工智能软件系统构建的核心步骤, 在训练阶段增加一些操作来保障模型的公平性是很有必要的. 例如在对数据进行预处理时, 要清理数据集以减少特征集和敏感属性之间的相关性; 在训练模型的过程中可以给模型增加约束, 例如奇偶校验约束, 因为模型的训练过程就是一个约束优化的过程; 在对系统进行后期处理时, 要注意调整训练的模型使其与敏感属性不相关.

除此之外, 公平系统还需将实际的使用环境和使用人群信息考虑到系统的设计要求中, 使各种环境中的各种

使用者不会产生受歧视的心理. 公平系统的设计要求包括:

● 识别所有的环境实体: 考虑所有的利益相关者, 包括他们的背景和特征.

● 保证系统对于环境的状态需求 (REQ): 比如系统应具备什么功能和质量属性?系统可能产生什么样的危害?是否符合当地法律及政策的规定?

● 识别环境和系统之间的接口: 什么样的数据将被人工智能软件系统感知?人工智能软件系统将执行什么类型的动作?

● 确定环境假设 (ENV): 利益相关者如何与系统互动?互动的对抗性、滥用性以及不公平的互动行为导致怎样后果?

● 制定足以满足 REQ 的软件规范 (SPEC): 主要包含人工智能软件系统应该努力实现哪种类型的公平性?

● 测试环境规格是否符合要求: 持续监控公平性指标并定时产生用户报告.

以下列举近年来研究人员对公平性探索的一些工作.

(1) 中间表示、竞争学习

Dwork 等人在文献 [86] 中首次讨论了通过学习一种中间表示来获得公平模型, 该中间表示对已经清洗过的数据变体进行编码. Edwards 等人[95]的工作则表明, 可以通过与竞争对手进行竞争性学习来尝试从公平模型的预测中预测敏感变量, 并最终实现公平.

(2) 因果分析

人工智能软件系统的过去数据可能存在偏差, 由此可能导致在一些诸如保险、贷款、雇用和预测性警示等领域产生具有法律或道德后果的不公平性决策. 对此, Kusner 等人[82]使用因果推断工具开发了一个公平性建模框架, 通过对照模型决策在反事实世界和现实世界中的一致性来判断决策对于某个群体的公平性. Chiappa 等人[96]引入了一种反事实方法来减轻沿不公平途径产生的影响, 这种方法克服了以往存在的导致个人特殊信息丢失的问题.

Johnson 等人提出的 Themis 方法通过使用因果分析来考虑群体公平性[97,98]. Themis 制定了一套公平性分数来作为衡量公平性的标准, 并使用随机测试生成技术来评估歧视的程度, 相比较而言, Udeshi 等人提出的 Aequitas 方法[99]把侧重点放在了个体公平性上. Aequitas 方法旨在发现歧视性输入以及对理解个体公平必不可少的那些输入. Aequitas 方法的首先会对输入空间进行随机采样来发现歧视性输入, 随后通过搜索这些输入的领域来查找更多的歧视性输入.

(3) 公平测试

Tramer 等人[100]首先提出了“公平性缺陷”的概念. Tramer 等人认为敏感属性和算法的输出在统计学上的联系是一个公平性缺陷并且在其工作中将这种缺陷命名为“不合理的关联”. Tramer 等人提出了一个综合测试工具 FairTest, FairTest 能够产生“易于理解”的错误报告来帮助开发人员测试和调试公平性错误. 目前 FairTest 已经运用于包括图像分类、收入预测和医疗保健预测等领域; Zhang 等人[101]提出了一种可以有效生成 DNN 模型不公平示例的方法, 他们的方法和同类方法对比, 能够在更少的时间内生成更多的不公平示例.

(4) 不公平缓解

在最新的一些工作中, 研究者将发现不公平现象后实施的缓解措施作为新的研究方向. 例如 Biswas 等人[102]创建了一个评估不公平缓解措施的基准, 该基准能够评估不同缓解措施的缓解效果, 为人们选择缓解措施提供了一定的指导性; Chakraborty 等人[103]提出了结合预处理和实时处理的方法 Fairway 来消除训练数据和训练模型中的不公平现象. 他们的研究结果表明, Fairway 方法可以在一个学习模型中发现偏差并减轻偏差, 而不会严重损害该模型的预测性能.

现有的针对人工智能软件系统公平性的质量保障方法研究仍然存在一些问题. Caton 等人[104]提出了几点不足; 首先是对于模型公平性和模型性能之间权衡的不足, 因为在实施提高公平性的方法时往往会损害模型的性能[105,106]; 其次, 除了模型性能和公平性之间的权衡外, 目前仍没有共识性工作对个体和群体公平性之间的权衡作出回应. 当前的公平性指标无法将个体公平和群体公平结合在一起研究[107,108], 并且许多有效解决群体公平性的方法经常会导致内部个体公平性的破坏[109].

1.5 可解释性

1.5.1 可解释性定义

与公平性不同, 为人工智能软件系统可解释性下定义的工作并不多. 本文参照 Biran 等人[110]以及 Miller[111]的工作, 将人工智能软件系统的可解释性定义如下: 人工智能软件系统可解释性 (interpretability) 是指系统使用者或观察人员能够理解人工智能软件系统所作决策的原因的程度.

在如图 7 所示的人工智能黑盒算法中. 用户往对他们来说相当于一个黑盒的人工智能软件系统中添加新数据并从黑盒模型中得到结果, 但关于预测结果的来由用户无法获知. 可解释性要求用户对黑盒模型产生预测结果的原因具有一定的理解.

图 7 黑盒算法

参考对公平性的定义, 对人工智能软件系统的可解释性进行公式表示; 使用 X 表示系统使用者或者观察人员; S 为当前待测的已经训练好的人工智能软件系统; Y 表示人工智能软件系统 S 对于任意输入所产生的真实标签. 对于人工智能软件系统 S, 如果利益相关者 x_1、 $x_2 \in X$ 对任意相同输入所产生的决策 y 的理解程度相同或非常相近, 则称人工智能软件系统 S 具有可解释性. 如公式 (10) 所示,

$$dis(de_{x_1}(Y=y), de_{x_2}(Y=y)) < \varepsilon \tag{10}$$

其中, de_{x_i} 表示 x_i 对于决策结果的理解程度 (可以制订一套可解释性评分指标来表示), dis 是衡量两个人的理解程度的距离衡量指标.

Zhang 等人在[6]中提出可解释性包含两个方面.

(1) 透明度, 即模型如何工作;

(2) 事后解释, 即其他可以从模型中得出的信息, 包括模型决策的解释.

1.5.2 可解释性必要性

人工智能软件系统的可解释性改善并不是一个新研究. 但越来越庞大和复杂的模型使人们很难用人类的语言来解释系统为什么会做出某种决定 (即决策的不透明化). 这是一些人工智能工具在有可解释性需求的应用领域中使用率仍然很低的原因之一. 尤其是人们又证明了 DNN 很容易被“愚弄”而发生错误[112]特别是在图像分类[31,113,114]和自然语言处理领域[115]. 这些意外“愚弄”行为和非故意歧视的存在更加凸显了对人工智能软件系统决策进行解释的必要性.

此外, 随着人工智能应用的扩展和相关法律法规的完善将会激发对更可解释的人工智能软件系统的需求. 例如自 2018 年起生效的《通用数据保护条例》(GDPR)[63]引入了一套与人工智能可解释性相关的权利, 特别是为任何受该人工智能软件系统决定影响的个人提供了要求解释的可能性.

增强人工智能软件系统的可解释性意义重大. 一方面, 它能够使得人类理解某些人工智能软件系统做出最终决策的逻辑和原因, 另一方面能加深人类对系统决策建立的信任, 同时也能尽量避免系统决策所造成的安全问题.

1.5.3 可解释性示例

Hamon 等人在文献 [28] 中描述了人工智能软件系统在大学中的应用, 即大学生奖学金自动评定系统. 该系统综合申请人班级平均成绩和申请人上一学年的学习成绩自动评定奖学金. 对于被拒绝的奖学金申请, 系统会返回反事实解释来说明拒绝申请人的理由.

而在汽车自动驾驶系统中, 人工智能软件系统的可解释性不仅表现为正常情况下汽车行驶决策的可解释程度, 还包括极端情况下决策的可解释程度. 用户一般对前者的可理解程度较高, 因为在一般情况下汽车的行驶决策是依照安全要求和遵守交通规则所作出的. 与之相比极端情况下的决策就比较难以解释, 但是充分理解极端情况的决策对于司机和他人的安全又是极为重要的.

1.5.4 常见质量保障方法

本节从降低模型复杂性、提高模型透明度和构建易解释网络等几个方向来梳理总结当前人工智能软件系统的可解释性质量保障研究工作.

(1) 降低模型复杂性、提高模型透明度

常用的深度网络使用大量的基本计算操作来得出决策; 例如, ResNet[61], 这是一种流行但复杂的图像分类体系结构, 它合并了约 5×10^7 个学习参数, 并执行了约 1 010 个浮点操作来对单个图像进行分类. 对深度网络的解释方法是找到降低这些操作复杂性的方法. 这可以通过创建同原始模型相似但更易于解释的代理模型来完成, 也可以通过创建决策树和显著图以突出显示与决策最相关部分来实现.

第 1 种是代理模型. Ribeiro 等人[116]的 LIME 可以很好地说明代理模型方法. LIME 通过探测输入扰动来解释黑盒人工智能软件系统, 然后使用该数据构建局部线性模型, 该线性模型可充当输入附近的完整模型的简化代理. Ribeiro 等人的结果表明, 该方法可用于识别对跨类型模型和问题域的决策影响最大的输入区域. Dong 等人[117]的方法不仅能够自适应的构建代理模型, 还能通过使用概率性推理语言来提升对模型决策的表达解释能力.

第 2 种方法是决策树. 最早是从 20 世纪 90 年代开始将神经网络分解为决策树, 一开始主要集中在浅层网络研究, 之后逐渐推广到深度神经网络. Zilke 等人的 DeepRED[118]是一个典型的代表, DeepRED 将专为浅层网络设计的 CRED[119]算法扩展到拥有任意多个隐藏层的深层网络. DeepRED 采用包括 RxREN[120]和 C4.5[121]在内的多种策略来简化决策树, 它虽然能够构建与原始网络非常接近的树, 但是所生成的树可能比较庞大导致可伸缩性受到限制. Aldrich 等人[122]的 ANN-DT 也是一种决策树方法, 它是使用采样来创建决策树.

第 3 种方法是显著图. Zeiler 等人的遮挡程序[123]提供了一个显著图的示例, 他们对网络进行了反复的测试, 通过将部分输入遮挡创建了一个图, 该图能够显示数据的哪些部分实际上对网络输出产生了影响. Simonyan 等人[124]的工作表示: 当可以直接检查深层网络参数时, 通过直接计算输入梯度可以更有效地创建显著图.

第 4 种方法是自动规则提取, 这是一种能够提高神经网络透明度的方法, Andrews 等人[125]总结了现有的规则提取技术, 并以包括表达力、透明性和规则质量在内的 5 个维度对它们进行了分类.

(2) 构建易解释网络

此外, Leilani 等人在文献 [7] 中提出, 可以采用几种不同的方法来创建易于解释的网络.

第 1 种网络是注意力网络. 基于注意力的网络学习提供对输入信息或内部特征进行加权的功能, 以引导信息对网络的其他部分可见. 基于注意力的方法在解决很多问题上取得了显著的成功, 例如允许自然语言翻译模型以合适的非顺序结构处理单词等. 注意力网络也被应用于医学图像诊断[126]和视觉问答[127]等领域. 尽管控制注意力的单元不是出于易于解释的目的训练的, 但它们确实会直接通过无网络传递显示其信息地图, 该地图可以用作决策解释的依据.

第 2 种是分离式表征. 分离式表征具有描述独立变异因素的个体维度, 但分离潜在因素一直是研究的难题, 以往是通过使用主成分分析[128]、独立成分分析[129]和非负矩阵分解[130]等技术来解决. 现在较为流行的是通过训练深层网络来显式的学习分离式表征. 如 Chen 等人[131]提出了 InfoGAN, 该方法以减少潜在因素之间的牵连为目的来训练可生成对抗网络; Zhang 等人[132]建议使用特殊损失函数来激励前馈网络分离它们的单元, 同时可以使用这些能够检测到有意义修补程序的单元来创建可解释的卷积神经网络而不是难以解释的混合模式.

第 3 种是解释生成. 主要是人工智能软件系统自动设计生成人类可理解的解释语句. 该方法已经在视觉问答系统[133]和细粒度图像分类系统[134]中进行应用. 产生解释的生成器使用大量人工注释过的数据进行训练, 最终产生人类可理解的人工智能软件系统决策解释语句, 文献 [135] 中还使用视觉指向和文字说明的多模式解释, 在用户个人评价和信任度实验中取得优异的表现.

(3) 反事实推理

Wachter 等人[136]在 2017 年首次提出反事实推理方法, 这是一种增强模型可解释性的新方法, 它通过对模型已预测的结果进行否定再进行重新的表征来构建一种可能的假设. 给定一个输入数据点和黑盒模型, Wachter 等人将反事实推理定义为一个数据点并尽可能靠近输入数据点, 他们的实验显示模型给出了和输入数据点不同的预测结果. 例如, 用户被模型拒绝给予贷款, 则反事实推理可能是“如果您的收入每年增加 5 000 美元, 您的信用评分就会

提高 30 点, 您的贷款就会获得批准”. Wachter 等人认为反事实推理是向用户解释模型结果的一种方式, 这样就可以识别出改变决策的可行方式以获得正向的预测.

现有的针对人工智能软件系统可解释性的质量保障方法研究存在几个不足. 首先是针对复杂人工智能系统特别是大型 DNN 系统所做决策的可解释性和理论理解水平仍处于起步阶段, 对多层高度复杂的系统决策无法作出令人满意和信服的解释[137]; 其次是当前为解决人工智能软件系统可解释性的各种方法是孤立的, 很少工作涉及将不同方法加以合并来实现更加有效的解释, 即缺少可以根据给定的解释目的和类型来得到最好解释的方法.

1.6 可用性

1.6.1 可用性定义

ISO 9241 标准将可用性 (Usability) 定义为“产品被指定用户在指定使用环境中以有效性、效率和满意度实现指定目标的程度”[138]. 基于 ISO 标准以及 Ardito 等人[139]和 Ponce 等人[140]的工作, 本文从用户体验角度出发将人工智能软件系统的可用性分为以下 3 个维度:

(1) 有效性 (effectiveness, *EFE*). 现有文献缺乏对人工智能软件系统有效性的明确定义, 本文将其定义为: 有效性是指人工智能软件系统对于符合要求的所有输入都能正确运行并产生用户预期结果;

(2) 效率 (efficiency, *EFI*). 人工智能软件系统的效率是指系统的预测速度[6];

(3) 满意度 (satisfaction, *SA*). ISO 9421-210 标准将其定义为: 人们对于针对使用或期望使用的产品、系统或者服务的认知印象和接受度[138].

从用户体验角度来说, 有效性要求人工智能软件系统能对用户符合要求的输入产生达到用户期望的输出, 和功能属性中正确性不同的是, 有效性要求模型的输出是合乎用户期望、合乎逻辑的输出, 正确性则是要求模型的输出必定正确; 人工智能软件系统的效率是指系统对用户输入的响应时间, 因而具有优异可用性的人工智能软件系统必须能对用户的请求作出及时的处理. Baeza-Yates 等人[141]认为效率是模型选择和框架选择要考虑的重要特征, 有时甚至比正确性更重要, Krik 等人[142]将不同算法的效率作为比较复杂性的指标之一, Musliner 等人[143]的工作指出, 以自动驾驶系统为代表的实时领域人工智能软件系统要求具有足够快的响应速度, 能够在可预测、适应需求和不确定环境中做出及时的行为; 满意度则是用户使用一个产品或系统全部过程的全部感受, 包括系统执行任务的操作成本等. ISO 对其补充了 3 个影响的因素; 系统、用户和使用环境.

本文对人工智能软件系统的可用性用公式 (11) 表示, 假设: *EFE*、*EFI*、*SA* 分别表示人工智能软件系统有效性、效率和满意度, f 函数是可用性评估函数, 则人工智能软件系统 S 的可用性 $U(S)$ 表示为:

$$U(S) = f(EFE, EFI, SA) \tag{11}$$

1.6.2 可用性必要性及示例

国际标准 ISO 10289 对产品质量制定了严密的评估标准, 包括外部质量、内部质量和使用质量[144]. 同样地, 软件的使用质量对人工智能软件质量评估也是非常重要的. 良好可用性的系统能够改善用户的使用体验, 而拥有反馈良好的用户群体是商业化系统的生存根本.

一项调查 B2B 网站用户流失原因的工作[145]表明: 有 37% 的用户流失原因是因为 B2B 网站差强人意的设计和复杂糟糕的交互体验. 因为如果系统无法让用户高有效、高效率和高满意的达成指定目标, 用户可能会停止使用该系统, 甚至将会寻求其他的解决途径来达成用户的目的. 对于商业化的人工智能系统来说, 用户流失造成将会造成毁灭性的影响.

1.6.3 常见质量保障方法

本节将从可用性系统设计要求和可用性系统测试及改进两个方面对当前人工智能软件系统的可用性质量保障进行梳理.

(1) 可用性系统设计

可用性人工智能软件系统设计要求是开发人员对目标系统的技术描述、功能描述和缺陷描述, 是可用性系统的规范, 通常需要考虑 4 个方面: 用户需求、心智模型的构建、差错处理和系统的反馈与控制.

第 1 个是用户需求. 用户需求的建立是实现人工智能软件系统可用性的核心一环, 但其存在两个注意点. 首先

要确定用户想要完成的任务的需求, 其次需对这些任务在自动化或者增强执行之间做出选择; 在一些用户缺乏执行任务的知识、能力或缺乏趣味的任务中可以采用自动化执行的方式, 例如系统的预测任务; 在一些难以传达用户对系统需求的任务中可以采用增强执行的方式, 例如汽车的驾驶任务.

第 2 个是心智模型的构建. 心智模型是为解释人同外界心理活动过程而构造的一种对比性的描述. 人们常用它来了解周围世界并同周围环境进行互动. 心智模型是根植于人的内心, 它影响着个人采取的行动、建立的假设、持有的成见. 建立一个和人工智能软件系统情况相符的心智模型能够为系统提供有效的指导行动, 从而增加用户对于系统可用性的正面评价; 相反, 当用户的心智模型与人工智能软件系统情况不相符时, 会让用户产生一种个人构想无法实现的落差感, 使得用户的满意度降低. 构建心智模型的目标是为了对人类的思维信息构建和处理机制进行研究和探索, 同时也为提升相应的人工智能软件系统的可用性提供新的体系结构和技术方法. 需要注意几点: (1) 用户对系统的看法, 这在很大程度上反映用户的满意程度; (2) 根据模型来计划系统执行动作; (3) 保持系统与用户的心理模型一致, 这是由于用户的心智模型随着时间的推移而不断变化; (4) 构建简易模型, 设计者可以根据先前的经验快速搭建一个粗略的用户模型来减少构建周期.

第 3 个是差错处理. 差错处理是纠正系统故障的一环, 差错处理的实现主要分为 4 个步骤: 1) 定义错误和故障, 错误主要分为用户错误和系统错误; 前者是用户自己所犯的错误 (例如点击了错误的按键), 后者是系统无法提供正确的结果 (例如推荐系统向用户推荐了糟糕的电影); 2) 检测并记录错误的发生; 3) 识别错误的来源, 例如用户错误可能是不匹配的心智模型或者糟糕的用户体验设计导致的, 系统错误可能是模型精度差、训练数据有问题导致的; 4) 提供可操作的错误处理指南.

最后一个环节为反馈和控制, 系统直接或间接地收集用户的反馈数据能够帮助系统提高可用性. 反馈数据的收集分为隐式反馈和显式反馈, 隐式反馈是指系统收集的用户行为数据, 例如一天中的使用时间、点击方式等; 显式反馈是由用户提示或主动提供的调查、评分、反馈表格等方式收集的反馈信息. 反馈的设计注意要将反馈与改进人工智能交互结合起来, 要尽可能地减少从确认用户反馈到系统响应的时间, 最后还需为用户提供一种调整人工智能行为的控制方式.

(2) 可用性系统测试及改进

经验用户测试作为一种开发可用系统的标准技术被广泛使用, 它主要是通过实际用户来测试系统从而确定系统存在的可用性方面问题. 虽然很多人认为在严谨使用时, 这种可以表达人为主观态度的方法确实有用, 但在实际软件系统开发中, 尤其在难以获得领域专家意见时, 经验用户测试成本过于昂贵和缓慢. 为此 Card 等人[146]提出 GOMS 模型, GOMS 模型由实现指定目标所需方法的描述组成, John 等人[147]列出了 GOMS 在实际可用性设计中的许多成功应用. Kieras 等人[148]介绍了一种基于计算机的工具 GLEAN, 该工具比手动构建可用性预测模型 (例如 GOMS 模型) 和经验用户测试节省了大量的测试时间.

对人工智能软件系统效率的改进是提高可用性的常见方法, 大多集中于数据[149,150]、Bug 研究[151]和实时策略[143,152,153]. 例如 Spieker 等人[149]研究了 3 种训练数据约简方法, 目的是在模型训练过程中找到与原始训练数据相似特征的较小子集, 从而可以提高模型构建速度, 该方法在模型预测速度上同样也有很好的效果; Zhang 等人[151]对 Stack Overflow 和 GitHub 中关于 TensorFlow 相关 Bug 提交进行研究, 发现在关于机器学习模型的 175 个 Bug 中, 属于效率问题的只有 9 个 (5.1%). Zhang 等人将其原因归结为用户的现有期望不高; Musliner 等人[143]的工作指出了将响应及时性和人工智能方法结合的 3 个方向; 将人工智能嵌入实时系统中、将实时反应嵌入到人工智能软件系统中以及将人工智能技术和实时子系统耦合成并行的协作组件.

现有的针对人工智能软件系统可用性的质量保障方法研究仍然存在一些问题. 首先是对于可用性 3 个维度的研究存在一定的偏重, 对于有效性和满意度的研究远不如效率维度的研究丰富; 其次, 虽然当前可以利用经验用户测试方法和 GLEAN 工具对人工智能软件系统可用性进行评估, 但仍然存在成本高、耗时多的缺点, 并且在评估标准上也缺少统一性的指标.

2 人工智能软件系统非功能属性总结及关系分析

人工智能非功能属性涉及人工智能软件系统评估标准, 通常与训练后的人工智能软件系统学习模型的行为有

关. 虽然已有工作对人工智能软件系统的非功能属性进行了一定程度的研究[1,6−9], 但遗憾的是大部分的工作只集中于某一个非功能属性, 缺少对非功能属性的总结和属性之间关系的梳理研究. 为了填补这个空白, 本文根据现有的相关工作和分析, 对上述非功能属性进行系统总结并对非功能属性之间的关系作详细的分析.

非功能属性的总结见表 1, 表中列举了上文总结的 6 个非功能属性. 其中, 定义/种类; 基于调研的相关工作归纳了每个非功能属性的定义; 重要性描述; 从整个人工智能软件系统的角度对每个非功能属性的重要性进行描述; 常见质量保障方法; 整合了第 1 节中 6 个非功能属性的常见质量保障方法, 更方便研究者的查找.

表 1 非功能属性小结

名称	定义/种类	重要性描述	常见质量保障方法
鲁棒性	人工智能软件系统发生故障时仍然能较好地完成预定工作	衡量人工智能软件系统受到外部干扰时维持自身正确性的能力	对抗性样本[25,31−36] 对抗性训练[37−42] 其他[43−45]
安全性	人工智能软件系统能够避免或减少攻击以保护自身安全	衡量人工智能软件系统抵御外部攻击保护系统内部组件不受破坏或降低破坏程度的能力	针对模型反演的研究[1,46−48] 完整性攻击[49] 威胁模型[50−53] 安全原则[54,55] 安全协议[54−62]
数据隐私性	人工智能软件系统能够保护自身数据安全	衡量人工智能软件系统保留私有数据信息的能力	数据匿名化[68−69] 防御数据窃取[70−72] 隐私保护协议[73−76]
公平性	人工智能软件系统所做决策对一切有关的事物公正、平等的对待、公平分配	衡量人工智能软件系统独立于敏感属性所做决策的能力	中间表示和公平测试[86,88,100−101] 竞争学习和因果分析[82,95−99] 不公平缓解[102,103]
可解释性	利益相关者对人工智能软件系统决策原因的理解程度	衡量人工智能软件系统所做决策易于理解的能力	代理模型[116−125] 解释性网络[7,126−135] 反事实推理[136]
可用性	用户对人工智能软件系统有效性、效率和满意的评估	衡量人工智能软件系统满足用户需求和便宜操作的能力	可用性测试[146−148] 效率改进[149−153]

从表 1 能看出鲁棒性、安全性和数据隐私性是更偏于衡量外部因素对人工智能软件系统影响的指标, 而公平性、可解释性、可用性则是衡量模型本身对外界 (例如用户等利益相关者) 影响的指标.

非功能属性之间关系如图 8 所示, 本文将属性之间的关系分为促进、阻碍和尚无明确定义. 其中, (1) 促进关系 (positive); 实线表示两者之间存在促进关系, 即一个属性提高时, 另一个属性也会随之提高; (2) 阻碍关系 (negative); 虚线表示两者之间存在阻碍关系, 即一个属性提高时, 另一个属性将会随之降低; (3) 尚无明确定义关系 (unclear); 两属性间不存在连线, 表明当前尚无工作对其关系作出明确定义. 考虑到非功能属性之间未必存在双向的关系, 在现有工作的基础上, 本文在图 8 中使用双向/单向箭头来表示属性间关系的方向性. 我们对图 8 中存在的属性关系进行研究分析, 如表 2 所示一共形成 16 组分析结果, 表 2 中还列出了表明属性 1 到属性 2 之间关系的相关工作[40,41,47,154−162].

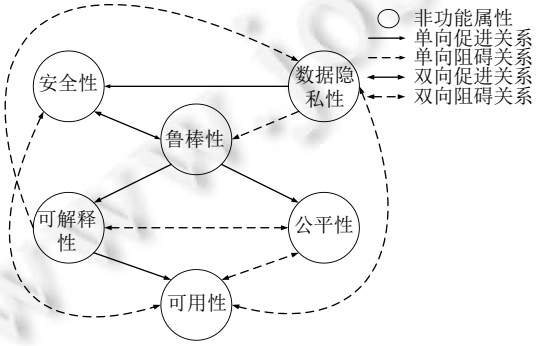

图 8 非功能属性间关系

表 2 非功能属性关系分析

属性1	属性2	关系(促进/阻碍/尚无明确定义)	相关工作
鲁棒性	安全性	促进	Gehr等人[40]提出的AI^{2}方法通过改善精度来提升模型鲁棒性, 同时实验结果也表明该模型防御攻击的安全性也得到了提升
	公平性	促进	Yeom等人[154]通过对模型施加随机平滑(提高鲁棒性方法)来设置最小度量并证明了模型的个体公平性
	可解释性	促进	Ross等人[41]的工作发现, 经过对抗训练的模型能表现更好的可解释性
安全性	鲁棒性	促进	Gehr等人[40]提出的AI^{2}方法通过改善精度来提升模型鲁棒性, 同时实验结果也表明该模型防御攻击的安全性也得到了提升
	可用性	阻碍	Braz等人[155]认为人工智能安全系统中安全性和可用性存在冲突问题, 他们提出了新的安全验证方法来缓解冲突问题
数据隐私性	鲁棒性	阻碍	Phan等人[156]通过放松模型隐私约束, 设计了具有可证明的保留机制来增强DP深层神经网络的鲁棒性;Lecuyer等人[157]认为差分隐私机制无法让低置信度数据具有可证明的鲁棒性
	安全性	促进	Papernot等人[47]将ML安全性和隐私性的研究进行系统化, 并提出分析数据隐私性将促进ML安全性的结论
	可用性	阻碍	Arul等人[158]认为在IPTV推荐系统里, 用户的使用体验和用户信息的提供是反比的关系, 即隐私性和可用性存在矛盾特性
公平性	可解释性	阻碍	Kleinberg等人[159]在进行决策算法研究中发现; 尽管模型可解释性和公平性似乎出自不同目标的动机, 但他们之间存在着根本矛盾
	可用性	阻碍	Manavalan等人[160]在对推荐网络的代理结构研究中发现; 在典型的推荐网络中, 有用性和公平性是成反比的
可解释性	数据隐私性	阻碍	Troiano等人[161]认为构建模型数据执行路径的模糊规则可解释性越强, 模型隐私泄露风险就越高
	公平性	阻碍	Kleinberg等人[159]在进行决策算法研究中发现; 尽管模型可解释性和公平性似乎出自不同目标的动机, 但他们之间存在着根本矛盾
	可用性	促进	Niemöller等人[162]在人工智能软件系统大数据分析中, 对算法模型的解释和模型影响的用户体验进行了研究, 结果表明; 具有更明确解释的算法可以提高用户的心理感知评分, 从而影响用户体验
可用性	安全性	阻碍	Braz等人[155]认为人工智能安全系统中安全性和可用性存在冲突问题, 他们提出了新的安全验证方法来缓解冲突问题
	数据隐私性	阻碍	Arul等人[158]认为在IPTV推荐系统里, 用户的使用体验和用户信息的提供是反比的关系, 即隐私性和可用性存在矛盾特性
	公平性	阻碍	Manavalan等人[160]发现; 在典型推荐网络中, 有用性和公平性是成反比

表 2 的关系分析显示, 基于现有的非功能属性关系研究工作, 鲁棒性同其他非功能属性的关系全表现为促进关系; 而公平性和可用性同其他非功能属性的关系全表现为阻碍关系; 安全性、数据隐私性和可解释性同其他非功能属性的关系既存在促进关系也存在阻碍关系, 但以阻碍关系为主; 表 2 中还显示现有非功能属性关系研究中, 安全性和公平性同其他非功能属性之间关系的研究相对较少.

3 人工智能软件系统非功能属性研究中的开源工具/开源项目

为了更好地支持研究者对人工智能软件系统的非功能属性进行研究分析, 本文除了将各非功能属性作为关键词在例如 Google 学术搜索、Springer、DBLP 等国内外重要的学术搜索引擎搜索之外, 还在例如 Github 等开源社区上对相关工作中用到的开源工具或开源项目进行了统计, 具体分析结果见表 3. 表中列举了近几年内的 10 种涉及鲁棒性、公平性、安全性、数据隐私性和可解释性等非功能属性的工具, 同时还介绍了每个工具的编程语言、发表时间、相关论文等简要信息并提供了工具的下载地址.

表 3 中 PRODeep[163]是中国科学院大学开发的 DNN 鲁棒性验证平台, PRODeep 结合了基于约束、基于抽象和基于优化的鲁棒性检查算法, 具有模块化架构, 可以轻松比较不同的算法; MAGICAL[164]是加州大学伯克利分校

研发的一款基准套件, 该套件可通过对模仿学习算法在实践中可能遇到的各种分布偏移的鲁棒性进行量化, 从而对泛化进行系统地评估; Fairness 是一款用于计算不同敏感属性之间算法公平性的度量工具, 其中度量的指标是根据二元分类任务中的模型预测来计算的; AIF360[165]是 IBM 研究所开发的一个可扩展的开放源码库, 它可以帮助检测和减轻整个 AI 应用程序生命周期中机器学习模型的不公平偏差; Aequitas[166]是开源的偏差审计工具包, 它重点关注标准 ML 指标及其对保护属性不同子组的评估; Fairness Measures[167]是一个度量分类和排序方案中不公平现象的开源项目, 它提供了调查公平性的数据集、度量指标和算法; Audit AI 在标准的机器学习程序中进行各种统计显著性测试, 以检测不同组或群体间的偏见和区别; Black-Box Rippe[168]提供了一种可以生成极小准确率损失的替代模型开源项目, 该项目一方面可以作为测试算法防窃取的安全能力, 另一方面可以作为替代模型加强算法可解释性; GAN-Leaks[169]是德国 Helmholt 信息安全中心所提供的一种可以检测目标模型数据隐私泄露能力的通用模型, 该模型可以在各种设置中实例化, 并且适用于各种深度生成模型; ICML2017 年最佳论文[170]提出了 Influence Functions 方法, 该方法通过影响函数来加强对黑盒模型的理解, 有学者提供了该方法的开源实现.

表 3 现有工具/开源项目总结

工具/项目名称	覆盖非功能属性	编程语言	发表时间	相关论文	下载地址
PRODeep	鲁棒性	C++	2020	[163]	https://github.com/ISCAS-PMC/PRODeep
MAGICAL	鲁棒性	Python	2020	[164]	https://github.com/qxcv/magical/
Fairness	公平性	R语言	2019	—	https://github.com/kozodoi/Fairness
AIF360	公平性	Python/R语言	2018	[165]	https://github.com/Trusted-AI/AIF360
Aequitas	公平性	Python	2018	[166]	https://github.com/dssg/aequitas
Fairness Measures	公平性	Python	2017	[167]	https://github.com/megantosh/fairness_measures_code/tree/master
Audit AI	公平性	Python	2019	—	https://github.com/pymetrics/audit-ai
Black-Box Ripper	安全性/可解释性	Python	2020	[168]	https://github.com/antoniobarbalau/black-box-ripper
GAN-Leaks	数据隐私性	Python	2020	[169]	https://github.com/DingfanChen/GAN-Leaks
Influence Functions	可解释性	Python	2020	[170]	https://github.com/nimarb/pytorch_influence_functions

4 总结和展望

随着人工智能软件系统被越来越广泛地应用于不同领域, 尤其是在一些安全攸关的领域, 人工智能软件系统的质量也受到人们的重视. 本文提供了针对人工智能软件系统中常见非功能属性的综述, 具体来说, 从非功能属性的定义、必要性、示例和常见质量保障方法这 4 个方面对该领域进行了系统的梳理. 最后, 对人工智能软件系统常见非功能属性之间的关系进行了详细的分析, 并总结了现有开源工具和开源项目, 希望能对相关领域的研究提供帮助.

尽管目前已经有很多的工作围绕着人工智能软件系统非功能属性进行展开, 但是该领域仍然面临着许多不足和挑战. 本文将对现存的主要问题和挑战进行概括, 以期未来有更多的研究围绕这些问题和挑战进行, 从而进一步保障人工智能软件系统的质量.

(1) 目前的工作大多偏向于鲁棒性和安全性等表现较为直观的非功能属性, 对于一些难以直接表示的非功能属性, 诸如可用性等属性的工作相对而言较少.

(2) 现有的非功能属性定义的泛用性相差较大, 例如人工智能软件系统的公平性. 由于公平性与具体领域相关, 例如信贷、教育、就业和住房等, 因此难以形成较为统一的共识.

(3) 虽然现在对人工智能属性的研究一般采用独立处理的办法, 但是考虑其根本原因时, 这些属性并不是严格相互独立的. 目前比较缺少对属性之间相关性的研究, 缺少总结分析非功能属性间关系的工作.

References:

[1] Wang Z, Yan M, Liu S, Chen JJ, Zhang DD, Wu Z, Chen X. Survey on testing of deep neural networks. Ruan Jian Xue Bao/Journal of

Software, 2020, 31(5): 1255−1275 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/5951.htm [doi: 10.13328/j.cnki.jos.005951]

[2] China Computer Federation. Quality assurance of artificial intelligence software system—CNCC technology forum. 2020 (in Chinese). https://www.ccf.org.cn/Media_list/cncc/2020-10-11/709302.shtml

[3] Pei KX, Cao YZ, Yang JG, Jana S. DeepXplore: Automated whitebox testing of deep learning systems. In: Proc. of the 26th Symp. on Operating Systems Principles. Shanghai: Association for Computing Machinery, 2017. 1–18. [doi: 10.1145/3132747.3132785]

[4] Cisse M, Adi Y, Neverova N, Keshet J. Houdini: Fooling deep structured prediction models. arXiv:1707.05373, 2017.

[5] Kurakin A, Goodfellow IJ, Bengio S. Adversarial examples in the physical world. In: Proc. of the 5th Int'l Conf. on Learning Representations (ICLR). Toulon: OpenReview.net, 2017.

[6] Zhang JM, Harman M, Ma L, Liu Y. Machine learning testing: Survey, landscapes and horizons. IEEE Trans. on Software Engineering, 2022, 48(1): 1–36. [doi: 10.1109/TSE.2019.2962027]

[7] Vinayagasundaram B, Srivatsa SK. Software quality in artificial intelligence system. Information Technology Journal, 2007, 6(6): 835–842. [doi: 10.3923/itj.2007.835.842]

[8] Gilpin LH, Bau D, Yuan BZ, Bajwa A, Specter M, Kagal L. Explaining explanations: An overview of interpretability of machine learning. In: Proc. of the 5th Int'l Conf. on Data Science and Advanced Analytics (DSAA). Turin: IEEE, 2018. 80–89. [doi: 10.1109/DSAA.2018.00018]

[9] Mehrabi N, Morstatter F, Saxena N, Lerman K, Galstyan A. A survey on bias and fairness in machine learning. ACM Computing Surveys, 2022, 45(6): 115. [doi: 10.1145/3457607]

[10] Qin Y, Wang HY, Xu C, Ma XX, Lu J. SynEva: Evaluating ML programs by mirror program synthesis. In: Proc. of the IEEE Int'l Conf. on Software Quality, Reliability and Security (QRS). Lisbon: IEEE, 2018. 171–182. [doi: 10.1109/QRS.2018.00031]

[11] Chen WJ, Gallas BD, Yousef WA. Classifier variability: Accounting for training and testing. Pattern Recognition, 2012, 45(7): 2661–2671. [doi: 10.1016/j.patcog.2011.12.024]

[12] Japkowicz N. Why question machine learning evaluation methods? (An illustrative review of the shortcomings of current methods). In: AAAI Workshop on Evaluation Methods for Machine Learning. Boston: AIAA, 2006. 6–11.

[13] Chen WJ, Samuelson FW, Gallas BD, Kang L, Sahiner B, Petrick N. On the assessment of the added value of new predictive biomarkers. BMC Medical Research Methodology, 2013, 13: 98. [doi: 10.1186/1471-2288-13-98]

[14] Roelofs R, Fridovich-Keil S, Miller J, Shankar V, Hardt M, Recht B, Schmidt L. A meta-analysis of overfitting in machine learning. In: Proc. of the 33rd Conf. on Neural Information Processing Systems (NeurIPS). Vancouver, 2019. 9175–9185.

[15] Hawkins DM. The problem of overfitting. Journal of Chemical Information and Computer Sciences, 2004, 44(1): 1–12. [doi: 10.1021/ci0342472]

[16] Chan HP, Sahiner B, Wagner RF, Petrick N. Classifier design for computer-aided diagnosis: Effects of finite sample size on the mean performance of classical and neural network classifiers. Medical Physics, 1999, 26(12): 2654–2668. [doi: 10.1118/1.598805]

[17] Sahiner B, Chan HP, Petrick N, Wagner RF, Hadjiiski L. Feature selection and classifier performance in computer-aided diagnosis: The effect of finite sample size. Medical Physics, 2000, 27(7): 1509–1522. [doi: 10.1118/1.599017]

[18] Fukunaga K, Hayes RR. Effects of sample size in classifier design. IEEE Trans. on Pattern Analysis and Machine Intelligence, 1989, 11(8): 873–885. [doi: 10.1109/34.31448]

[19] Zhang JM, Barr ET, Guedj B, Harman M, Shawe-Taylor J. Perturbed model validation: A new framework to validate model relevance. arXiv:1905.10201, 2019.

[20] Werpachowski R, György A, Szepesvári C. Detecting overfitting via adversarial examples. In: Proc. of the 33rd Conf. on Neural Information Processing Systems (NIPS). Vancouver, 2019. 7856–7866.

[21] IEEE. IEEE Std 610.12-1990 IEEE standard glossary of software engineering terminology. IEEE, 1990. 1–84. [doi: 10.1109/IEEESTD.1990.101064]

[22] Shahrokni A, Feldt R. A systematic review of software robustness. Information and Software Technology, 2013, 55(1): 1–17. [doi: 10.1016/j.infsof.2012.06.002]

[23] Huber PJ. Robust Statistics. Hoboken: John Wiley & Sons, Inc., 2004. [doi: 10.1002/0471725250]

[24] Katz G, Barrett C, Dill DL, Julian K, Kochenderfer MJ. Reluplex: An efficient SMT solver for verifying deep neural networks. In: Proc. of the 29th Int'l Conf. on Computer Aided Verification (CAV). Heidelberg: Springer, 2017. 97–117. [doi: 10.1007/978-3-319-63387-9_5]

[25] Tjeng V, Xiao KY, Tedrake R. Evaluating robustness of neural networks with mixed integer programming. In: Proc. of the 7th Int'l Conf. on Learning Representations. New Orleans: OpenReview.net, 2019.

[26] Chung L, Nixon BA, Yu E, Mylopoulos J. Non-Functional Requirements in Software Engineering. Boston: Springer, 2020. [doi: 10.1007/978-1-4615-5269-7]

[27] Afzal W, Torkar R, Feldt R. A systematic review of search-based testing for non-functional system properties. Information and Software Technology, 2009, 51(6): 957–976. [doi: 10.1016/j.infsof.2008.12.005]

[28] Hamon R, Junklewitz H, Sanchez I. Robustness and explainability of artificial intelligence: From technical to policy solutions. Luxembourg: Publications Office of the European Union, 2020. [doi: 10.2760/57493]

[29] PCMag. Self-driving car. https://www.pcmag.com/encyclopedia/term/self-driving-car

[30] Becic E, Zych N, Ivarsson J. Vehicle automation report HWY18MH010. Washington: National Transportation Safety Board Office of Highway Safety, 2019.

[31] Szegedy C, Zaremba W, Sutskever I, Bruna J, Erhan D, Goodfellow IJ, Fergus R. Intriguing properties of neural networks. In: Proc. of the 2nd Int'l Conf. on Learning Representations (ICLR). Banff, 2014.

[32] Wang YZ, Jha S, Chaudhuri K. Analyzing the robustness of nearest neighbors to adversarial examples. In: Proc. of the 35th Int'l Conf. on Machine Learning (ICML). Stockholmsmässan: PMLR, 2018. 5120–5129.

[33] Cheng MH, Yi JF, Chen PY, Zhang H, Hsieh CJ. Seq2Sick: Evaluating the robustness of sequence-to-sequence models with adversarial examples. In: Proc. of the 34th AAAI Conf. on Artificial Intelligence. New York: AAAI, 2020. 3601–3608. [doi: 10.1609/aaai.v34i04.5767]

[34] Jakubovitz D, Giryes R. Improving DNN robustness to adversarial attacks using jacobian regularization. In: Proc. of the 15th European Conf. on Computer Vision (ECCV). Munich: Springer, 2018. 525–541. [doi: 10.1007/978-3-030-01258-8_32]

[35] Goswami G, Ratha NK, Agarwal A, Singh R, Vatsa M. Unravelling robustness of deep learning based face recognition against adversarial attacks. In: Proc. of the 32nd AAAI Conf. on Artificial Intelligence. New Orleans: AAAI, 2018. 6829–6836.

[36] Carlini N, Wagner D. Towards evaluating the robustness of neural networks. In: Proc. of the IEEE Symp. on Security and Privacy. San Jose: IEEE, 2017. 39–57. [doi: 10.1109/SP.2017.49]

[37] Gao J, Wang BL, Lin ZM, Xu WL, Qi YJ. Deepcloak: Masking deep neural network models for robustness against adversarial samples. In: Proc. of the 5th Int'l Conf. on Learning Representations (ICLR). Toulon: OpenReview.net, 2017.

[38] Cheng Y, Tu ZP, Meng FD, Zhai JJ, Liu Y. Towards robust neural machine translation. In: Proc. of the 56th Annual Meeting of the Association for Computational Linguistics (Vol. 1: Long Papers). Melbourne: Association for Computational Linguistics, 2018. 1756–1766. [doi: 10.18653/v1/P18-1163]

[39] Cheng Y, Jiang L, Macherey W. Robust neural machine translation with doubly adversarial inputs. In: Proc. of the 57th Annual Meeting of the Association for Computational Linguistics. Florence: Association for Computational Linguistics, 2019. 4324–4333. [doi: 10.18653/v1/P19-1425]

[40] Gehr T, Mirman M, Drachsler-Cohen D, Tsankov P, Chaudhuri S, Vechev M. AI2: Safety and robustness certification of neural networks with abstract interpretation. In: Proc. of the IEEE Symp. on Security and Privacy. San Francisco: IEEE, 2018. 3–18. [doi: 10.1109/SP.2018.00058]

[41] Ross AS, Doshi-Velez F. Improving the adversarial robustness and interpretability of deep neural networks by regularizing their input gradients. In: Proc. of the 32nd AAAI Conf. on Artificial Intelligence. New Orleans: AAAI, 2018. 1660–1669.

[42] Kwiatkowska M. Safety and robustness for deep learning with provable guarantees. In: Proc. of the 35th IEEE/ACM Int'l Conf. on Automated Software Engineering. Virtual Event: IEEE, 2020. 1–3. [doi: 10.1145/3324884.3418901]

[43] Fawzi A, Moosavi-Dezfooli SM, Frossard P. Robustness of classifiers: From adversarial to random noise. In: Proc. of the 30th Int'l Conf. on Neural Information Processing Systems (NIPS). Barcelona: Curran Associates Inc., 2016. 1632–1640.

[44] Bastani O, Ioannou Y, Lampropoulos L, Vytiniotis D, Nori AV, Criminisi A. Measuring neural net robustness with constraints. In: Proc. of the 30th Int'l Conf. on Neural Information Processing Systems (NIPS). Barcelona: Curran Associates Inc., 2016. 2621–2629.

[45] Banerjee SS, Cyriac J, Jha S, Kalbarczyk ZT, Lyer RK. Towards a bayesian approach for assessing fault tolerance of deep neural networks. In: Proc. of the 49th Annual IEEE/IFIP Int'l Conf. on Dependable Systems and Networks-Supplemental Volume (DSN-S). Portland: IEEE, 2019. 25–26. [doi: 10.1109/DSN-S.2019.00018]

[46] Tramèr F, Zhang F, Juels A, Reiter MK, Ristenpart T. Stealing machine learning models via prediction APIs. In: Proc. of the 25th USENIX Conf. on Security Symp. Austin: USENIX Association, 2016. 601–618.

[47] Batina L, Bhasin S, Jap D, Picek S. Poster: Recovering the input of neural networks via single shot side-channel attacks. In: Proc. of the ACM SIGSAC Conf. on Computer and Communications Security. London: Association for Computing Machinery, 2019. 2657–2659. [doi: 10.1145/3319535.3363280]

[48] Yang ZQ, Zhang JY, Chang EC, Liang ZK. Neural network inversion in adversarial setting via background knowledge alignment. In:

Proc. of the ACM SIGSAC Conf. on Computer and Communications Security. London: Association for Computing Machinery, 2019. 225–240. [doi: 10.1145/3319535.3354261]

[49] Shen SQ, Tople S, Saxena P. Auror: Defending against poisoning attacks in collaborative deep learning systems. In: Proc. of the 32nd Annual Conf. on Computer Security Applications. California: Association for Computing Machinery, 2016. 508–519. [doi: 10.1145/2991079.2991125]

[50] Papernot N, McDaniel P, Sinha A, Wellman MP. SoK: Security and privacy in machine learning. In: Proc. of the IEEE European Symp. on Security and Privacy. London: IEEE, 2018. 399–414. [doi: 10.1109/EuroSP.2018.00035]

[51] Marisetty S. Five steps to successful threat modelling. 2019. https://community.arm.com/iot/b/internet-of-things/posts/five-steps-to-successful-threat-modelling

[52] Yang ZM, Zhang ZG. The study on resolutions of STRIDE threat model. In: Proc. of the 1st IEEE Int'l Symp. on Information Technologies and Applications in Education. Kunming: IEEE, 2007. 271–273. [doi: 10.1109/ISITAE.2007.4409285]

[53] Houmb SH, Franqueira VNL. Estimating ToE risk level using CVSS. In: Proc. of the Int'l Conf. on Availability, Reliability and Security. Fukuoka: IEEE, 2009. 718–725. [doi: 10.1109/ARES.2009.151]

[54] McGraw G. Software security. IEEE Security & Privacy, 2004, 2(2): 80–83. [doi: 10.1109/MSECP.2004.1281254]

[55] Gollmann D. Computer security. WIREs Computational Statistics, 2010, 2(5): 544–554. [doi: 10.1002/wics.106]

[56] Kilbertus N, Gascón A, Kusner MJ, Veale M, Gummadi KP, Weller A. Blind justice: Fairness with encrypted sensitive attributes. In: Proc. of the 35th Int'l Conf. on Machine Learning (ICML). Stockholmsmässan: PMLR, 2018. 2635–2644.

[57] Songhori EM, Hussain SU, Sadeghi AR, Koushanfar F. Compacting privacy-preserving k-nearest neighbor search using logic synthesis. In: Proc. of the 52nd Annual Design Automation Conf. San Francisco: Association for Computing Machinery, 2015. 36. [doi: 10.1145/2744769.2744808]

[58] Agrawal N, Shamsabadi SA, Kusner MJ, Gascón A. QUOTIENT: Two-party secure neural network training and prediction. In: Proc. of the ACM SIGSAC Conf. on Computer and Communications Security. London: Association for Computing Machinery, 2019. 1231–1247. [doi: 10.1145/3319535.3339819]

[59] Mohassel P, Rindal P. ABY^3: A mixed protocol framework for machine learning. In: Proc. of the ACM SIGSAC Conf. on Computer and Communications Security. Toronto: ACM, 2018. 35–52. [doi: 10.1145/3243734.3243760]

[60] Mohassel P, Zhang YP. SecureML: A system for scalable privacy-preserving machine learning. In: Proc. of the IEEE Symp. on Security and Privacy. San Jose: IEEE, 2017. 19–38. [doi: 10.1109/SP.2017.12]

[61] He KM, Zhang XY, Ren SQ, Sun J. Deep residual learning for image recognition. In: Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR). Las Vegas: IEEE, 2016. 770–778. [doi: 10.1109/CVPR.2016.90]

[62] Dwork C. Differential privacy: A survey of results. In: Proc. of the 5th Int'l Conf. on Theory and Applications of Models of Computation. Xi'an: Springer, 2008. 1–19. [doi: 10.1007/978-3-540-79228-4_1]

[63] Voigt P, von dem Bussche A. The EU General Data Protection Regulation (GDPR): A Practical Guide. Cham: Springer, 2017. [doi: 10.1007/978-3-319-57959-7]

[64] Wikipedia. California consumer privacy act. 2018. https://en.wikipedia.org/wiki/California_Consumer_Privacy_Act

[65] Song CZ, Ristenpart T, Shmatikov V. Machine learning models that remember too much. In: Proc. of the ACM SIGSAC Conf. on Computer and Communications Security. Dallas: Association for Computing Machinery, 2017. 587–601. [doi: 10.1145/3133956.3134077]

[66] Fredrikson M, Lantz E, Jha S, Lin S, Page D, Ristenpart T. Privacy in pharmacogenetics: An end-to-end case study of personalized warfarin dosing. In: Proc. of the 23rd USENIX Conf. on Security Symp. San Diego: USENIX Association, 2014. 17–32.

[67] Fredrikson M, Jha S, Ristenpart T. Model inversion attacks that exploit confidence information and basic countermeasures. In: Proc. of the 22nd ACM SIGSAC Conf. on Computer and Communications Security. Denver: Association for Computing Machinery, 2015. 1322–1333. [doi: 10.1145/2810103.2813677]

[68] Sweeney L. *k*-Anonymity: A model for protecting privacy. Int'l Journal of Uncertainty, Fuzziness and Knowledge-based Systems, 2002, 10(5): 557–570. [doi: 10.1142/S0218488502001648]

[69] Gao AQ, Diao LH. Privacy preservation for attribute order sensitive workload in medical data publishing. Ruan Jian Xue Bao/Journal of Software, 2009, 20(S1): 314–320 (in Chinese with English abstract).

[70] Papernot N, McDaniel P, Goodfellow I, Jha S, Celik ZB, Swami A. Practical black-box attacks against machine learning. In: Proc. of the 2017 ACM on Asia Conf. on Computer and Communications Security. Abu Dhabi: Association for Computing Machinery, 2017. 506–519. [doi: 10.1145/3052973.3053009]

[71] Lee T, Edwards B, Molloy I, Su D. Defending against neural network model stealing attacks using deceptive perturbations. In: Proc. of

the IEEE Security and Privacy Workshops. San Francisco: IEEE, 2019. 43–49. [doi: 10.1109/SPW.2019.00020]

[72] Prashar A, Monroy SAS. A secure algorithm for deep learning training under GAN attacks. In: Proc. of the Int'l Conf. on Communications, Computing, Cybersecurity, and Informatics (CCCI). Sharjah: IEEE, 2020. 1–6. [doi: 10.1109/CCCI49893.2020.9256566]

[73] Gascón A, Schoppmann P, Balle B, Raykova M, Doerner J, Zahur S, Evans D. Privacy-preserving distributed linear regression on high-dimensional data. Proceedings on Privacy Enhancing Technologies, 2017, 2017(4): 345–364. [doi: 10.1515/popets-2017-0037]

[74] Nikolaenko V, Weinsberg U, Ioannidis S, Joye M, Boneh D, Taft N. Privacy-preserving ridge regression on hundreds of millions of records. In: Proc. of the IEEE Symp. on Security and Privacy. Berkeley: IEEE, 2013. 334–348. [doi: 10.1109/SP.2013.30]

[75] Nikolaenko V, Ioannidis S, Weinsberg U, Joye M, Taft N, Boneh D. Privacy-preserving matrix factorization. In: Proc. of the ACM SIGSAC Conf. on Computer & Communications Security. Berlin: Association for Computing Machinery, 2013. 801–812. [doi: 10.1145/2508859.2516751]

[76] Liu J, Juuti M, Lu Y, Asokan N. Oblivious neural network predictions via MiniONN transformations. In: Proc. of the ACM SIGSAC Conf. on Computer and Communications Security. Dallas: Association for Computing Machinery, 2017. 619–631. [doi: 10.1145/3133956.3134056]

[77] Rigaki M, Garcia S. A survey of privacy attacks in machine learning. arXiv:2007.07646, 2020.

[78] Corbett-Davies S, Goel S. The measure and mismeasure of fairness: A critical review of fair machine learning. arXiv:1808.00023, 2018.

[79] Gajane P, Pechenizkiy M. On formalizing fairness in prediction with machine learning. arXiv:1710.03184, 2017.

[80] Verma S, Rubin J. Fairness definitions explained. In: Proc. of the IEEE/ACM Int'l Workshop on Software Fairness (FairWare). Gothenburg: IEEE, 2018. 1–7. [doi: 10.23919/FAIRWARE.2018.8452913]

[81] Saxena NA, Huang K, DeFilippis E, Radanovic G, Parkes DC, Liu Y. How do fairness definitions fare?: Examining public attitudes towards algorithmic definitions of fairness. In: Proc. of the AAAI/ACM Conf. on AI, Ethics, and Society (AIES). Honolulu: Association for Computing Machinery, 2019. 99–106. [doi: 10.1145/3306618.3314248]

[82] Kusner MJ, Loftus J, Russell C, Silva R. Counterfactual fairness. In: Proc. of the 31st Int'l Conf. on Neural Information Processing Systems. Long Beach: Curran Associates Inc., 2017. 4069–4079.

[83] Grgić-Hlača N, Zafar MB, Gummadi KP, Weller A. The case for process fairness in learning: Feature selection for fair decision making. In: Proc. of the Symp. on Machine Learning and the Law at the 29th Conf. on Neural Information Processing Systems. Barcelona, 2016.

[84] Zafar MB, Valera I, Rodriguez MG, Gummadi KP. Fairness constraints: Mechanisms for fair classification. In: Proc. of the 20th Int'l Conf. on Artificial Intelligence and Statistics (AISTATS). Fort Lauderdale: PMLR, 2017. 962–970.

[85] Hardt M, Price E, Srebro N. Equality of opportunity in supervised learning. In: Proc. of the 30th Int'l Conf. on Neural Information Processing Systems. Barcelona: Curran Associates Inc., 2016. 3323–3331.

[86] Dwork C, Hardt M, Pitassi T, Reingold O, Zemel R. Fairness through awareness. In: Proc. of the 3rd Innovations IN Theoretical Computer Science Conf. (ITSC). Massachusetts: Association for Computing Machinery, 2012. 214–226. [doi: 10.1145/2090236.2090255]

[87] Nabi R, Malinsky D, Shpitser I. Learning optimal fair policies. In: Proc. of the 36th Int'l Conf. on Machine Learning (ICML). Long Beach: PMLR, 2019. 4674–4682.

[88] Rieke A, Bogen M. Help wanted: An examination of hiring algorithms, equity, and bias. 2018. https://www.upturn.org/reports/2018/hiring-algorithms/

[89] Cohen L, Lipton ZC, Mansour Y. Efficient candidate screening under multiple tests and implications for fairness. arXiv:1905.11361v1, 2019.

[90] Buolamwini J, Gebru T. Gender shades: Intersectional accuracy disparities in commercial gender classification. In: Proc. of the Fairness, Accountability and Transparency (FAT). New York: PMLR, 2018. 77–91.

[91] Sweeney L. Discrimination in online ad delivery. Communications of the ACM, 2013, 56(5): 44–54. [doi: 10.1145/2447976.2447990]

[92] Angwin J, Larson J, Mattu S, Kirchner L. Machine bias. 2016. https://www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing

[93] Wang Y, Zhang M. Reducing implicit gender biases in software development: Does intergroup contact theory work? In: Proc. of the 28th ACM Joint Meeting on European Software Engineering Conf. and Symp. on the Foundations of Software Engineering (ESEC/SIGSOFT FSE). Virtual Event: Association for Computing Machinery, 2020. 580–592. [doi: 10.1145/3368089.3409762]

[94] Barocas S, Selbst AD. Big data's disparate impact. California Law Review, 2016, 104: 671–732. [doi: 10.2139/ssrn.2477899]

[95] Edwards H, Storkey AJ. Censoring representations with an adversary. In: Proc. of the 4th Int'l Conf. on Learning Representations. San Juan, 2016.

[96] Chiappa S. Path-specific counterfactual fairness. In: Proc. of the 33rd AAAI Conf. on Artificial Intelligence. Honolulu: AAAI, 2019. 7801–7808. [doi: 10.1609/aaai.v33i01.33017801]

[97] Angell R, Johnson B, Brun Y, Meliou A. Themis: Automatically testing software for discrimination. In: Proc. of the 26th ACM Joint Meeting on European Software Engineering Conf. and Symp. on the Foundations of Software Engineering (ESEC/SIGSOFT FSE). Lake Buena Vista: Association for Computing Machinery, 2018. 871–875. [doi: 10.1145/3236024.3264590]

[98] Johnson B, Brun Y, Meliou A. Causal testing: Finding defects' root causes. arXiv:1809.06991, 2020.

[99] Udeshi S, Arora P, Chattopadhyay S. Automated directed fairness testing. In: Proc. of the 33rd ACM/IEEE Int'l Conf. on Automated Software Engineering. Montpellier: Association for Computing Machinery, 2018. 98–108. [doi: 10.1145/3238147.3238165]

[100] Tramèr F, Atlidakis V, Geambasu R, Hsu D, Hubaux JP, Humbert M, Juels A, Lin H. FairTest: Discovering unwarranted associations in data-driven applications. In: Proc. of the IEEE European Symp. on Security and Privacy (EuroS&P). Paris: IEEE, 2017. 401–416. [doi: 10.1109/EuroSP.2017.29]

[101] Zhang PX, Wang JY, Sun J, Dong GL, Wang XY, Wang XG, Dong JS, Dai T. White-box fairness testing through adversarial sampling. In: Proc. of the ACM/IEEE 42nd Int'l Conf. on Software Engineering. Pittsburgh: Association for Computing Machinery, 2020. 949–960. [doi: 10.1145/3377811.3380331]

[102] Biswas S, Rajan H. Do the machine learning models on a crowd sourced platform exhibit bias? An empirical study on model fairness. In: Proc. of the 28th ACM Joint Meeting on European Software Engineering Conf. and Symp. on the Foundations of Software Engineering (ESEC/SIGSOFT FSE). Virtual Event: Association for Computing Machinery, 2020. 642–653. [doi: 10.1145/3368089.3409704]

[103] Chakraborty J, Majumder S, Yu Z, Menzies T. Fairway: A way to build fair ML software. In: Proc. of the 28th ACM Joint Meeting on European Software Engineering Conf. and Symp. on the Foundations of Software Engineering (ESEC/SIGSOFT FSE). Virtual Event: Association for Computing Machinery, 2020. 654–665. [doi: 10.1145/3368089.3409697]

[104] Caton S, Haas C. Fairness in machine learning: A survey. arXiv:2010.04053, 2020.

[105] Haas C. The price of fairness—A framework to explore trade-offs in algorithmic fairness. In: Proc. of the 40th Int'l Conf. on Information Systems (ICIS). Munich: Association for Information Systems, 2019.

[106] Berk R, Heidari H, Jabbari S, Kearns M, Roth A. Fairness in criminal justice risk assessments: The state of the art. Sociological Methods & Research, 2021, 50(1): 3–44. [doi: 10.1177/0049124118782533]

[107] Chouldechova A. Fair prediction with disparate impact: A study of bias in recidivism prediction instruments. Big Data, 2017, 5(2): 153–163. [doi: 10.1089/big.2016.0047]

[108] Kleinberg J, Ludwig J, Mullainathan S, Rambachan A. Algorithmic fairness. Aea Papers and Proc., 2018, 108: 22–27. [doi: 10.1257/pandp.20181018]

[109] Speicher T, Heidari H, Grgic-Hlaca N, Gummadi KP, Singla A, Weller A, Zafar MB. A unified approach to quantifying algorithmic unfairness: Measuring individual & group unfairness via inequality indices. In: Proc. of the ACM SIGKDD Int'l Conf. on Knowledge Discovery & Data Mining. London: Association for Computing Machinery, 2018. 2239–2248. [doi: 10.1145/3219819.3220046]

[110] Biran O, Cotton C. Explanation and justification in machine learning: A survey. In: Proc. of the IJCAI-17 Workshop on Explainable Artificial Intelligence (XAI). 2017. 8–13.

[111] Miller T. Explanation in artificial intelligence: Insights from the social sciences. Artificial Intelligence, 2019, 267: 1–38. [doi: 10.1016/j.artint.2018.07.007]

[112] Nguyen A, Yosinski J, Clune J. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In: Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR). Boston: IEEE, 2015. 427–436. [doi: 10.1109/CVPR.2015.7298640]

[113] Moosavi-Dezfooli SM, Fawzi A, Fawzi O, Frossard P. Universal adversarial perturbations. In: Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR). Honolulu: IEEE, 2017. 86–94. [doi: 10.1109/CVPR.2017.17]

[114] Papernot N, McDaniel P, Jha S, Fredrikson M, Celik ZB, Swami A. The limitations of deep learning in adversarial settings. In: Proc. of the IEEE European Symp. on Security and Privacy (EuroS&P). Saarbruecken: IEEE, 2016. 372–387. [doi: 10.1109/EuroSP.2016.36]

[115] Jia RB, Liang P. Adversarial examples for evaluating reading comprehension systems. In: Proc. of the 2017 Conf. on Empirical Methods in Natural Language Processing (EMNLP). Copenhagen: Association for Computational Linguistics, 2017. 2021–2031. [doi: 10.18653/v1/D17-1215]

[116] Ribeiro MT, Singh S, Guestrin C. "Why should I trust you?": Explaining the predictions of any classifier. In: Proc. of the 22nd ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining. San Francisco: Association for Computing Machinery, 2016. 1135–1144. [doi: 10.1145/2939672.2939778]

[117] Dong GL, Wang JY, Sun J, Zhang Y, Wang XY, Dai T, Dong JS, Wang XG. Towards interpreting recurrent neural networks through probabilistic abstraction. In: Proc. of the 35th IEEE/ACM Int'l Conf. on Automated Software Engineering (ASE). Melbourne: IEEE, 2020. 499–510.

[118] Zilke JR, Mencía EL, Janssen F. DeepRED–rule extraction from deep neural networks. In: Proc. of the 19th Int'l Conf. on Discovery Science (DS). Bari: Springer, 2016. 457–473. [doi: 10.1007/978-3-319-46307-0_29]

[119] Sato M, Tsukimoto H. Rule extraction from neural networks via decision tree induction. In: Proc. of the IJCNN2001. Int'l Joint Conf. on Neural Networks. Washington: IEEE, 2001. 1870–1875. [doi: 10.1109/IJCNN.2001.938448]

[120] Augasta MG, Kathirvalavakumar T. Reverse engineering the neural networks for rule extraction in classification problems. Neural Processing Letters, 2012, 35(2): 131–150. [doi: 10.1007/s11063-011-9207-8]

[121] Quinlan JR. C4.5: Programs for Machine Learning. San Mateo: Morgan Kaufmann, 1993. [doi: 10.1007/BF00993309]

[122] Schmitz GPJ, Aldrich C, Gouws FS. ANN-DT: An algorithm for extraction of decision trees from artificial neural networks. IEEE Trans. on Neural Networks, 1999, 10(6): 1392–1401. [doi: 10.1109/72.809084]

[123] Zeiler MD, Fergus R. Visualizing and understanding convolutional networks. In: Proc. of the 13th European Conf. on Computer Vision (ECCV). Zurich: Springer, 2014. 818–833. [doi: 10.1007/978-3-319-10590-1_53]

[124] Simonyan K, Vedaldi A, Zisserman A. Deep inside convolutional networks: Visualising image classification models and saliency maps. In: Proc. of the 2nd Int'l Conf. on Learning Representations. Banff, 2014.

[125] Andrews R, Diederich J, Tickle AB. Survey and critique of techniques for extracting rules from trained artificial neural networks. Knowledge-based Systems, 1995, 8(6): 373–389. [doi: 10.1016/0950-7051(96)81920-4]

[126] Zhang ZZ, Xie YP, Xing FY, McGough M, Yang L. MdNet: A semantically and visually interpretable medical image diagnosis network. In: Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR). Honolulu: IEEE, 2017. 3549–3557. [doi: 10.1109/CVPR.2017.378]

[127] Lu JS, Yang JW, Batra D, Parikh D. Hierarchical question-image co-attention for visual question answering. In: Proc. of the 30th Int'l Conf. on Neural Information Processing Systems (NIPS). Barcelona: Curran Associates Inc., 2016. 289–297.

[128] Jolliffe IT. Principal components in regression analysis. In: Jolliffe IT, ed. Principal Component Analysis. New York: Springer, 1986. 129–155. [doi: 10.1007/978-1-4757-1904-8_8]

[129] Hyvärinen A, Oja E. Independent component analysis: Algorithms and applications. Neural Networks, 2000, 13(4-5): 411–430. [doi: 10.1016/S0893-6080(00)00026-5]

[130] Berry MW, Browne M, Langville AN, Pauca VP, Plemmons RJ. Algorithms and applications for approximate nonnegative matrix factorization. Computational Statistics & Data Analysis, 2007, 52(1): 155–173. [doi: 10.1016/j.csda.2006.11.006]

[131] Chen X, Duan Y, Houthooft R, Schulman J, Sutskever I, Abbeel P. InfoGAN: Interpretable representation learning by information maximizing generative adversarial nets. In: Proc. of the 30th Int'l Conf. on Neural Information Processing Systems (NIPS). Barcelona: Curran Associates Inc., 2016. 2180–2188.

[132] Zhang QS, Wu YN, Zhu SC. Interpretable convolutional neural networks. In: Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR). Salt Lake City: IEEE, 2018. 8827–8836. [doi: 10.1109/CVPR.2018.00920]

[133] Antol S, Agrawal A, Lu JS, Mitchell M, Batra D, Zitnick CL, Parikh D. VQA: Visual question answering. In: Proc. of the IEEE Int'l Conf. on Computer Vision (ICCV). Santiago: IEEE, 2015. 2425–2433. [doi: 10.1109/ICCV.2015.279]

[134] Hendricks LA, Akata Z, Rohrbach M, Donahue J, Schiele B, Darrell T. Generating visual explanations. In: Proc. of the 14th European Conf. on Computer Vision (ECCV). Amsterdam: Springer, 2016. 3–19. [doi: 10.1007/978-3-319-46493-0_1]

[135] Huk Park D, Anne Hendricks L, Akata Z, Rohrbach A, Schiele B, Darrell T, Rohrbach M. Multimodal explanations: Justifying decisions and pointing to the evidence. In: Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR). Salt Lake City: IEEE, 2018. 8779–8788. [doi: 10.1109/CVPR.2018.00915]

[136] Wachter S, Mittelstadt B, Russell C. Counterfactual explanations without opening the black box: Automated decisions and the GDPR. Harvard Journal of Law & Technology, 2018, 31(2): 841–887. [doi: 10.2139/ssrn.3063289]

[137] Carvalho DV, Pereira EM, Cardoso JS. Machine learning interpretability: A survey on methods and metrics. Electronics, 2019, 8(8): 832. [doi: 10.3390/electronics8080832]

[138] ISO. ISO 9241-11: 2018(en) Ergonomics of human-system interaction—Part 11: Usability: Definitions and concepts. Geneva: ISO, 2018.

[139] Ardito C, Costabile MF, De Marsico M, Lanzilotti R, Levialdi S, Roselli T, Rossano V. An approach to usability evaluation of e-learning applications. Universal Access in the Information Society, 2006, 4(3): 270–283. [doi: 10.1007/s10209-005-0008-6]

[140] Ponce P, Balderas D, Peffer T, Molina A. Deep learning for automatic usability evaluations based on images: A case study of the

usability heuristics of thermostats. Energy and Buildings, 2018, 163: 111–120. [doi: 10.1016/j.enbuild.2017.12.043]

[141] Baeza-Yates R, Liaghat Z. Quality-efficiency trade-offs in machine learning for text processing. In: Proc. of the IEEE Int'l Conf. on Big Data. Boston: IEEE, 2017. 897–904. [doi: 10.1109/BigData.2017.8258006]

[142] Kirk M. Thoughtful Machine Learning: A Test-driven Approach. Sebastopol: O'Reilly Media Inc., 2014.

[143] Musliner DJ, Hendler JA, Agrawala AK, Durfee EH, Strosnider JK, Paul CJ. The challenges of real-time AI. Computer, 1995, 28(1): 58–66. [doi: 10.1109/2.362628]

[144] ISO. ISO 10289: 1999(en) Methods for corrosion testing of metallic and other inorganic coatings on metallic substrates—Rating of test specimens and manufactured articles subjected to corrosion tests. Geneva: ISO, 1999.

[145] Povolná L. Marketing communications on B2B markets. In: Proc. of the 13th Int'l Bata Conf. for Ph.D. Students and Young Researchers (DOKBAT). Zlín, 2017. 278–285. [doi: 10.7441/dokbat.2017.29]

[146] Card SK, Moran TP, Newell A. Computer text-editing: An information-processing analysis of a routine cognitive skill. Cognitive Psychology, 1980, 12(1): 32–74. [doi: 10.1016/0010-0285(80)90003-1]

[147] John BE, Kieras DE. The GOMS family of analysis techniques: Tools for design and evaluation. 1994. https://apps.dtic.mil/sti/pdfs/ADA309174.pdf

[148] Kieras DE, Wood SD, Abotel K, Hornof A. GLEAN: A computer-based tool for rapid GOMS model usability evaluation of user interface designs. In: Proc. of the 8th Annual ACM Symp. on User Interface and Software Technology. Pittsburgh: Association for Computing Machinery, 1995. 91–100. [doi: 10.1145/215585.215700]

[149] Spieker H, Gotlieb A. Towards testing of deep learning systems with training set reduction. arXiv:1901.04169, 2019.

[150] Kaur J, Mann KS. AI based healthcare platform for real time, predictive and prescriptive analytics using reactive programming. Journal of Physics: Conf. Series, 2017, 933: 012010. [doi: 10.1088/1742-6596/933/1/012010]

[151] Zhang YH, Chen YF, Cheung SC, Xiong YF, Zhang L. An empirical study on TensorFlow program bugs. In: Proc. of the 27th ACM SIGSOFT Int'l Symp. on Software Testing and Analysis. Amsterdam: Association for Computing Machinery, 2018. 129–140. [doi: 10.1145/3213846.3213866]

[152] Buro M. Real-time strategy games: A new AI research challenge. In: Proc. of the 8th Int'l Joint Conf. on Artificial Intelligence. Acapulco: Morgan Kaufmann, 2003. 1534–1535.

[153] Buro M, Furtak TM. RTS games and real-time AI research. In: Proc. of the Behavior Representation in Modeling and Simulation Conf. (BRIMS). 2004. 51–58.

[154] Yeom S, Fredrikson M. Individual fairness revisited: Transferring techniques from adversarial robustness. In: Proc. of the 29th Int'l Joint Conf. on Artificial Intelligence (IJCAI). Yokohama: ijcai.org, 2020. 437–443. [doi: 10.24963/ijcai.2020/61]

[155] Braz C, Robert JM. Security and usability: The case of the user authentication methods. In: Proc. of the 18th Conf. on Interaction Homme-Machine (IHM). Montreal: Association for Computing Machinery, 2006. 199–203. [doi: 10.1145/1132736.1132768]

[156] Phan NH, Vu MN, Liu Y, Jin RM, Dou DJ, Wu XT, Thai MT. Heterogeneous gaussian mechanism: Preserving differential privacy in deep learning with provable robustness. In: Proc. of the 28th Int'l Joint Conf. on Artificial Intelligence (IJCAI). Macao: ijcai.org, 2019. 4753–4759. [doi: 10.24963/ijcai.2019/660]

[157] Lecuyer M, Atlidakis V, Geambasu R, Hsu D, Jana S. Certified robustness to adversarial examples with differential privacy. In: Proc. of the IEEE Symp. on Security and Privacy. San Francisco: IEEE, 2019. 656–672. [doi: 10.1109/SP.2019.00044]

[158] Arul T, Anagnostopoulos NA, Katzenbeisser S. Privacy & usability of IPTV recommender systems. In: Proc. of the IEEE Int'l Conf. on Consumer Electronics (ICCE). Las Vegas: IEEE, 2019. 1–2. [doi: 10.1109/ICCE.2019.8662046]

[159] Kleinberg J, Mullainathan S. Simplicity creates inequity: Implications for fairness, stereotypes, and interpretability. In: Proc. of the 2019 ACM Conf. on Economics and Computation (EC). Phoenix: Association for Computing Machinery, 2019. 807–808. [doi: 10.1145/3328526.3329621]

[160] Manavalan P, Singh MP. Emerging properties of knowledge sharing referral networks: Considerations of effectiveness and fairness. In: Proc. of the 7th Int'l Workshop on Agents and Peer-to-peer Computing. Budapest: Springer, 2012. 13–23. [doi: 10.1007/978-3-642-31809-2_2]

[161] Troiano L, Rodríguez-Muñiz LJ, Ranilla J, Díaz I. Interpretability of fuzzy association rules as means of discovering threats to privacy. Int'l Journal of Computer Mathematics, 2012, 89(3): 325–333. [doi: 10.1080/00207160.2011.613460]

[162] Niemöller J, Washington N. Subjective perception scoring: Psychological interpretation of network usage metrics in order to predict user satisfaction. Annals of Telecommunications, 2017, 72(7-8): 431–441. [doi: 10.1007/s12243-017-0575-6]

[163] Li RJ, Li JL, Huang CC, Yang PF, Huang XW, Zhang LJ, Xue B, Hermanns H. PRODeep: A platform for robustness verification of deep neural networks. In: Proc. of the 28th ACM Joint Meeting on European Software Engineering Conf. and Symp. on the Foundations

of Software Engineering (ESEC/SIGSOFT FSE). Virtual Event: Association for Computing Machinery, 2020. 1630–1634. [doi: 10.1145/3368089.3417918]

[164] Toyer S, Shah R, Critch A, Russell S. The MAGICAL benchmark for robust imitation. In: Proc. of the 34th Conf. on Neural Information Processing Systems. Vancouver, 2020.

[165] Bellamy RKE, Dey K, Hind M, Hoffman SC, Houde S, Kannan K, Lohia P, Martino J, Mehta S, Mojsilovic A, Nagar S, Ramamurthy KN, Richards J, Saha D, Sattigeri P, Singh M, Varshney KR, Zhang YF. AI Fairness 360: An extensible toolkit for detecting, understanding, and mitigating unwanted algorithmic bias. arXiv:1810.01943, 2018.

[166] Saleiro P, Kuester B, Hinkson L, London J, Stevens A, Anisfeld A, Rodolfa KT, Ghani R. Aequitas: A bias and fairness audit toolkit. arXiv: 811.05577, 2018.

[167] Zehlike M, Castillo C, Bonchi F, Baeza-Yates R, Hajian S, Megahed M. Fairness measures: Datasets and software for detecting algorithmic discrimination. https://fairnessmeasures.github.io/Pages/About

[168] Bărbălău A, Cosma A, Ionescu RT, Popescu M. Black-Box Ripper: Copying black-box models using generative evolutionary algorithms. In: Proc. of the 34th Conf. on Neural Information Processing Systems. Vancouver, 2020.

[169] Chen DF, Yu N, Zhang Y, Fritz M. GAN-Leaks: A taxonomy of membership inference attacks against generative models. In: Proc. of the ACM SIGSAC Conf. on Computer and Communications Security. Virtual Event: Association for Computing Machinery, 2020. 343–362. [doi: 10.1145/3372297.3417238]

[170] Koh PW, Liang P. Understanding black-box predictions via influence functions. In: Proc. of the 34th Int'l Conf. on Machine Learning (ICML). Sydney: PMLR, 2017. 1885–1894.

附中文参考文献:

[1] 王赞, 闫明, 刘爽, 陈俊洁, 张栋迪, 吴卓, 陈翔. 深度神经网络测试研究综述. 软件学报, 2020, 31(5): 1255–1275. http://www.jos.org.cn/1000-9825/5951.htm [doi: 10.13328/j.cnki.jos.005951]

[2] 中国计算机学会. 人工智能软件系统的质量保障丨CNCC技术论坛. 2020. https://www.ccf.org.cn/Media_list/cncc/2020-10-11/709302.shtml

[69] 高爱强, 刁麓弘. 医疗数据发布中属性顺序敏感的隐私保护方法. 软件学报, 2009, 20(S1): 314–320.

叶仕俊(1996－), 男, 硕士, 主要研究领域为人工智能软件测试, 文本分类.

戴启印(1996－), 男, 硕士, 主要研究领域为人工智能软件测试, 图像检索.

张鹏程(1981－), 男, 博士, 教授, 博士生导师, CCF 高级会员, 主要研究领域为智能化软件工程, 服务计算, 数据科学, 边缘计算, 区块链.

袁天昊(1997－), 男, 硕士, 主要研究领域为人工智能软件测试, 语音识别.

吉顺慧(1987－), 女, 博士, 副教授, CCF 专业会员, 主要研究领域为软件建模, 分析, 测试, 验证.

任彬(1997－), 男, 硕士, 主要研究领域为人工智能软件测试, 图像检索.