

# 环境感知自适应软件的运行时输入验证技术综述



王慧妍 徐经纬 许畅

计算机软件新技术国家重点实验室(南京大学) 南京 210023

南京大学计算机科学与技术系 南京 210023

(cocowhy1013@gmail.com)

**摘要** 随着软件智能化与大数据时代的到来,环境感知自适应软件作为智能软件中的代表趋于流行。环境感知自适应软件有两大特征:1)“环境感知”,即能够通过传感器等设备感知周围环境并采集环境数据;2)“自适应”,即能够基于采集的环境数据自适应地进行软件决策。这类软件的主要表现特征为在运行时刻能够动态感知周边环境的变化并进行交互,从而做出决策。此外,随着大数据时代的到来,越来越多的人工智能模型被使用并被期望能够帮助环境感知自适应软件更好地实现自适应机制,使其能够更加智能地通过与环境的感知交互来做决策。一方面,由于运行时环境复杂,该类软件的运行时环境情况往往难以估计和预料,使得其在实际部署后运行在复杂环境中的可靠性很难通过事先测试得到有效保障,这也成为了这类软件在运行时得到有效质量保障所面临的一大挑战。而另一方面,此类软件对人工智能模型的应用与人工智能模型基于统计的核心特征,使得其在运行时刻选择应用人工智能模型来进行辅助决策也存在一定的局限性,这更加剧了保障此类软件在运行时刻质量的难度。因此,如何能够在此类软件的实际部署运行时更好地保障其运行质量与可靠性成为了当今智能软件工程的一个广泛研究的问题。与此同时,输入验证被认为是保障运行时刻软件质量的一大常用手段,它通过对软件输入进行有效识别,来避免不合适的输入在运行时刻被输入软件而影响软件行为。基于此,文中对环境感知自适应软件的运行时输入验证技术进行总结与综述,基于此类软件的两大特征,从“环境感知”方面的环境数据感知模块的输入验证及“自适应”方面决策模块的输入验证两个方面,分别对已有技术进行调研与综述。同时,文中还探讨了对环境感知自适应软件的运行时输入验证技术问题中的主要性能挑战,为实现更加高效的输入验证做框架性总结。最后,还对人工智能技术广泛应用于环境感知自适应软件的现状带来的对此类软件额外决策的挑战做了讨论与分析,已有工作对此挑战的探索也让此类软件进一步成熟,并为其未来集成决策逻辑演化从而达到软件自成长的理想提供支撑。通过对相关技术的综述,试图为相关领域的科研工作者勾画一个对环境感知自适应软件在运行时刻较清晰的质量保障框架,为未来的相关研究提供可能的方向与角度。

**关键词:** 环境感知自适应软件;运行时质量保障;输入验证;软件可靠性;约束检查

**中图法分类号** TP311

## Survey on Runtime Input Validation for Context-aware Adaptive Software

WANG Hui-yan, XU Jing-wei and XU Chang

State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210023, China

Department of Computer Science and Technology, Nanjing University, Nanjing 210023, China

**Abstract** With the widespread of intelligence and big data, context-aware adaptive software, one representative of intelligent software, has gained increasing popularities. It has two key characteristics: 1) “context-aware”, referring to the ability of becoming aware of environments through ubiquitous sensors. 2) “adaptive”, referring to the ability of making adaptations based on collected contexts. As such, context-aware adaptive software can at runtime sense its surrounding environment and make adaptations smartly. Besides, with the growing development of artificial intelligence (AI) technologies, more AI models have been applied in context-aware adaptive software for smarter adaptations. Therefore, on one hand, due to the complexity of environments at runtime, the software suffers from severe reliability issues during its deployment, which is difficult to avoid by sole testing due to the lack of practically controllable environments, thus leading to great challenges for its runtime reliability assurance. On the other hand, the application of AI models in context-aware adaptive software further aggravates its reliability issues. As such, how to maintain the runtime reliability of context-aware adaptive software has been a widely-open research problem in intelligent software engineering, while input validation has shown promising in this field by identifying and isolating unexpected inputs from be-

到稿日期:2020-03-18 返修日期:2020-05-14 本文已加入开放科学计划(OSID),请扫描上方二维码获取补充信息。

基金项目:国家自然科学基金重点项目(61932021,61802170)

This work was supported by the Key Program of the National Natural Science Foundation of China (61932021,61802170).

通信作者:许畅(changxu@nju.edu.cn)

ing fed into the software in order to avoid possible uncontrollable consequences at runtime. In this article, we survey techniques on runtime input validation for context-aware adaptive software concerning its two key characteristics: “context-aware” and “adaptive”. Meanwhile, we also dig into the reliability issue problem concerning its cost-effectiveness in solving, and overview the concerned research framework. Finally, we discuss some latest concerns of context-aware adaptive software at present and in future, and present how context-aware adaptive software supports the emergence of self-growing software in the vision. As a summary, we survey existing efforts on runtime input validation for context-aware adaptive software, and aim to form a structured framework for the potential solutions on its reliability issues. We wish this may shed some light on relative researchers in future.

**Keywords** Context-aware adaptive applications, Runtime quality assurance, Input validation, Software reliability, Constraint checking

## 1 引言

随着智能化技术与大数据时代的到来,越来越多的环境感知自适应软件(context-aware adaptive software)趋于流行。作为现代智能软件的一大代表,这类软件具有两大明显的特征:1)“环境感知”,即此类软件与环境交互密切,可以通过各类传感器、摄像设备等对环境进行感知,搜集环境数据;2)“自适应”,即此类软件能够基于由感知环境收集的数据对自身行为进行自适应调整和相关决策。此类软件具有对环境智能解析与反馈的优势,因此受到了很多关注,且被广泛应用于生活的各个场景,如能够根据室内情况自动控制并调节温度和湿度的智能家居系统,能够根据路况拥堵情况自动推荐路线的智能导航应用等。与此同时,随着人工智能技术的广泛应用,越来越多的人工智能模型在环境感知自适应软件中的使用更赋予了此类软件新的生命力,使其能够更加智能地实现“自适应”决策,如各类热门的无人驾驶汽车应用通过应用各类人工智能模型来实现自动驾驶等。

然而,一方面,此类软件与环境的交互频繁,而现实运行环境往往十分复杂且不可预料,这为其运行时的可靠性带来了巨大的挑战;另一方面,此类软件对各类人工智能模型的使用也为其在运行过程中的决策引入了更多的不确定性,从而进一步增加了保障其部署运行时可靠性的难度。综上,此类软件在运行时刻的可靠性面临重大挑战,甚至可能在实际运行时出现各种异常行为。例如,热门的无人驾驶应用,在运行时刻受到现实环境的天气、光照,甚至周围各类感知信号等的干扰而出现决策异常,Tesla 自动驾驶车就由于运行时受到强光照影响对物体识别不准确而导致了严重事故<sup>[1]</sup>。因此,针对此类部署于复杂环境中并与之交互密切的智能软件,有效保障其运行时的可靠性至关重要。输入验证作为一种保障运行时刻的质量的常用手段,可以通过对运行时刻的输入进行监控验证来避免运行时刻接受异常输入从而导致软件行为异常的现象。本文主要调研针对环境感知自适应软件进行运行时输入验证的相关技术,分析其如何通过输入验证的方式来保障此类软件在运行时刻的可靠性。首先对环境感知的自适应软件进行介绍,从“环境感知”与“自适应”两大特征出发,分别对“环境感知”部分针对外部感知数据的输入验证工作与“自适应”部分针对内部决策机制的输入验证技术进行调研与综述,力图从此类软件感知环境到处理决策两大子任务出发,对运行时输入验证技术进行介绍,并讨论与两大子任务相关的各类输入验证技术如何相辅相成,以保障整个软件在运行时刻

的整体可靠性。本文通过对相关技术的综述,期望为未来研究寻求可能的科研契机与突破方向。本文的主要内容包括:

(1)简单介绍环境感知自适应软件的背景与工作流程框架,并在此框架上分析其运行时面临的问题与挑战,明确问题定义(见第2节)。

(2)调研问题的相关技术,分析其工作内容及可能存在的不足(见第3节)。

1)对于环境感知的输入验证工作,主要区分其是对环境感知数据的直接处理,还是对抽象后上下文的质量保障,并在后文分别对两者进行探讨。

2)对于内部自适应决策的输入验证工作,主要区分其不同的决策逻辑形式,即传统基于规则的决策逻辑与新型基于学习的决策逻辑,并在后文分别对两者进行探讨。

(3)对未来的研究方向进行探讨,提出未来可能的科研重点与突破角度(见第4节)。

## 2 问题与挑战

### 2.1 环境感知自适应软件

环境感知自适应软件,是指能够有效利用上下文信息(例如位置、环境温度、GPS数据等)进行推断,并基于此自适应地做出任务决策的软件。如图1所示,其内部模块可以按照功能与任务目的的不同分为环境感知与自适应决策两个子模块。

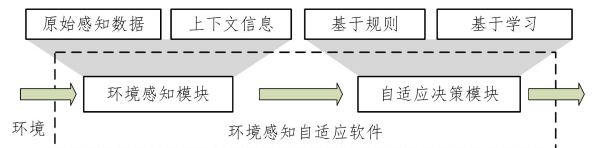


图1 环境感知自适应软件的流程

Fig. 1 Process of context-aware adaptive software

环境感知模块主要通过物理部署的传感器、摄像头等感知设备来进行信息读取,以获取感知数据并对其进行处理;获取物理传感器的原始数据(如GPS信息、RFID数据、拍摄图像、温度等);基于此,根据应用需求综合各原始数据,抽象形成与应用相关的上下文信息。

自适应决策模块主要根据获得的上下文信息来进行与应用相关的任务决策。例如,对于智能家居系统<sup>[2]</sup>,任务要求根据感知到的房间物理温度来决策是否需要合理调节室温;而对于自动驾驶系统,任务则是根据感知到的汽车周围障碍物的位置、速度等信息来决策当前是否需要调控汽车的行驶状

态(如加速、减速、变道等)。传统的自适应决策模块通常是基于规则式的,即主要通过制定感知和决策之间的应对规则来决策,其中应对规则主要用于明确感知状况自主决策的对应关系,如 ADAM<sup>[3]</sup> 中为货物运输应用建模的基于规则的决策实例。而近年来,随着人工智能的兴起,越来越多的环境感知自适应软件选择采用新型的基于学习的决策方式来进行决策,并期望提升决策的智能性。此类基于学习的决策模块的核心为,通过人工智能方式事先从大样本数据中训练学习得到人工智能模型(如各类神经网络模型),并将其用于运行时时刻对感知到的环境状况进行决策行为的自动预测。相比传统基于规则的决策模式,这类新型的基于学习的决策模式更加智能,一方面能够减少编写自适应逻辑的人工工作量,另一方面能够更加智能地处理更多复杂的实际情况。因此,基于学习的决策模块被广泛应行于各类环境感知自适应软件中,如各类自动驾驶系统、自然语言识别工具、物体识别软件等。

2.2 问题描述与挑战分析

环境感知自适应软件的一大特征就是与环境交互频繁,但是由于环境具有复杂性与不可预知性,此类软件在部署运行后的运行可靠性往往面临着巨大的挑战。无论是感知模块还是决策模块,都存在着由于环境的复杂性而导致的质量问题,从而影响其部署后运行的可靠性。而此类质量问题在开发与测试过程中往往由于难以对环境进行充分建模与模拟而很难被发现,但在运行时刻暴露往往会带来严重的后果,例如 Uber 无人车在运行过程中由于对夜间环境决策不准确而造成了冲撞行人, Tesla 自动驾驶模式对强光照下的物体识别不准确撞到了公交车等,因此在复杂环境中保障软件运行时刻的质量刻不容缓。

本文首先对环境感知自适应软件在运行时可能存在的质量问题的主要来源进行深入分析,主要包含两部分。1)运行时感知环境所采用的物理传感器受噪声、信号干扰等影响存在不可避免的感知误差,从而造成感知的原始数据与后续抽象形成的上下文信息都有可能存在质量问题,对整体软件产生影响,进而导致在此基础上的决策存在偏差。2)决策模块本身存在不确定性,其主要针对基于学习的决策模块,与传统基于规则的决策模块不同,由于缺少显式人为编写的决策逻辑,难以事先有效验证其在不同感知环境情况下进行自适应决策的预测结果的合理性。同时,由于主流学习模块基于统计训练的本质特点是从大数据样本中挖掘特征,并通过泛化性计算运行时时刻感知环境下统计指标最优的决策方案,因此应用基于学习模块的此类软件可能受到学习模块训练样本数据代表性不足、实际预测环境情况复杂、训练模型本身质量不佳等各类因素的影响,使得决策模型在运行时时刻对软件当前感知到的环境下的最优决策方案的预测并不准确,外在表现为学习模型实际预测的准确性通常难以达到 100%,从而使此类软件本身在运行时刻的行为不可靠。

解决这一问题的常用方式就是在运行时时刻进行输入验证(runtime input validation),即在运行时时刻对软件本身相关的输入主体进行验证,从而有效识别可能造成软件行为不可靠的输入,并对其加以过滤或修复。本文研究的主要问题就是,如何对此类软件的各个模块进行有效的输入验证。我们针对

前文所述的环境感知自适应软件的两大特征与其运行时质量问题的两大来源,从软件核心的两大模块即环境感知模块与自适应决策模块出发,分别分析模块存在的质量问题与其原因,并介绍针对相应模块如何进行有效的输入验证,最终总结两大模块的输入验证工作如何相辅相成以保障环境感知自适应软件在运行时的整体可靠性。

3 研究现状

本节将针对环境感知与自适应决策两大模块分别调研相关的在运行时进行输入验证的工作。针对环境感知模块,我们将从感知的原始数据和抽象形成的上下文信息这两个主体进行调研;而针对自适应决策模块,我们将从当前两大主流决策逻辑,即基于规则与基于学习的决策逻辑,来调研相关的输入验证工作。

3.1 环境感知模块

感知模块的质量问题主要是物理感知设备受到噪声、信号干扰等因素的影响而造成感知数据不准确,主要表现为感知的原始数据以及基于此抽象的应用相关上下文信息出现偏差或缺失,从而使得依赖于此的应用决策出现严重的质量问题,进而影响整体软件的运行时可靠性。

环境感知模块的主要工作内容有两点:1)通过传感器等物理设备从环境中获取原始感知数据;2)基于原始感知数据,在考虑应用层面需求的情况下进行环境建模,抽象出应用相关的上下文信息。因此,下文对感知模块的输入验证调研,将根据验证主体(原始感知数据或上下文数据)不同,分别从这两个方面来总结已有工作。

3.1.1 原始感知数据

原始感知数据,是指通过传感器等物理设备获取的未经后续复杂处理的环境数据等,是物理设备感知噪声、信号干扰等因素所带来的直接质量问题的载体。其中常用的环境感知手段,如 GPS 定位、RFID 标记、位移传感器、温度/湿度测量器等,均可能在被感知的过程中出现严重的质量问题<sup>[4-5]</sup>,从而造成感知数据与实际数据之间的偏差。例如,典型 RFID 技术标记中常见的错读与交叉读<sup>[6]</sup>,就有可能造成 30%~40% 的 RFID 数据采集偏差<sup>[3]</sup>。

我们通过分析原始感知数据的特征观察到,此类数据由于感知手段往往来源于同源的传感设备,且常可以建模为纯数值形式,因此对原始感知数据的输入验证往往从感知数据的具体数值或实际采集频率出发进行深入分析,从而进行感知数据的验证与修正。已有工作一部分关注于在传感器感知数据中针对不同主体检测其中可能的异常数据,例如 Khoussainova<sup>[7]</sup> 利用约束规则主要针对 RFID 数据中的异常数据进行检测并汇报, Subramaniam<sup>[8]</sup> 和 Zhuang<sup>[9]</sup> 则关注于如何对 WSN 传输的感知数据进行异常检测。另外一部分工作关注于如何自动化地对传感器数据中的系统误差进行修正,其中主要有两大类方法,一类是基于数值的修正方法,另一类是基于滤波的修正方法。前者通过对传感器数据感知误差进行建模,来理解物理传感器在感知过程中的误差情况,例如通过事先建立的误差模型或者误差函数,在运行时时刻对数据进行自动修正<sup>[10]</sup>;后者则通过滤波进行辅助,直接在采集过程中降



低误差对感知数据的影响,对数据进行平滑处理<sup>[3,11-12]</sup>。由于此部分工作往往与物理传感器设备硬件条件关系密切,且更多属于信号处理方向的技术,因此本文不做过多讨论。

### 3.1.2 上下文信息数据

上下文信息数据,是指那些根据应用需求对原始感知数据进行建模并抽象的信息数据,与应用需求相关,且往往具备一定的语义含义。例如,利用原始传感 GPS 数据与 RFID 技术数据进行交叉分析,可以抽象得到应用感兴趣的某一特定 ID 实例的实际位置变化速度或路径情况等。对于上下文信息数据的验证工作,由于上下文信息本身的建模往往与应用语义相关,因此属于软件层面的工作。

我们首先分析上下文信息数据的特征。上下文信息数据属于原始传感器数据经过处理分析后的产物,通常可追溯到多种传感设备,且由于来源多样,数据格式通常异构,因此难以采用与验证传统原始感知数据类似的针对同源数值进行清洗的验证方式,即仅从数值或采集频率层面进行验证或修复。同时,由于上下文信息数据的主要特征在于其与应用需求密切相关的语义含义,因此主流的对上下文信息的输入验证工作通常借助于对其含义的把握,通过利用来源于物理知识或应用需求的约束规则来建模上下文信息数据需要满足的语义含义,从而检测当前上下文信息是否存在语义违反的问题。主流工作将该类约束规则称为对上下文信息的一致性约束规则<sup>[13]</sup>,并要求上下文信息满足该规则,否则称当前上下文信息数据违反一致性要求(或出现不一致性),基于此对上下文信息进行有效的语义含义验证。主流的对上下文信息的研究工作主要存在两个维度,一个维度关注于对当前应用相关的上下文信息进行验证并报告不一致信息,另一个维度关注于对不一致信息进行有效处理,从而获取高质量的上下文信息用于运行。

针对如何有效验证并发现当前上下文信息是否存在不一致的问题,已有工作通常关注于针对上下文一致性检测的检测效果与效率的平衡,即希望在得到较好的检测效果的同时,能够在运行时刻满足应用对检测效率的需求。例如, Xu 等<sup>[13]</sup>从上下文信息的验证技术效率出发,基于一致性约束提出增量式的上下文一致性检测,其能够高效地验证运行时的上下文信息,且相比 Nentwich 等<sup>[14]</sup>提出的传统无增量式检测方式有显著的效率提升;相似地,后续 Xu 等<sup>[15]</sup>与 Sui 等<sup>[16]</sup>同样立足于验证技术效率层面,借助于 CPU 的多线程处理能力与 GPU 的高性能计算能力,使上下文信息的验证效率得到了进一步的提升。Wang 等<sup>[17]</sup>则从上下文信息的验证时机出发,提出了一种能够智能地决定上下文信息验证时机的调度策略,此调度策略能够搭配不同的上下文验证技术,普遍地提升了一致性检测验证的整体效率。

针对存在不一致性的上下文信息,同样有一系列工作关注于如何有效定位并修复存在质量问题的上下文,这也是另一项重要的针对上下文信息进行运行时验证的具体研究方向。其中,以 Insuk 等<sup>[18]</sup>与 Ranganathan 等<sup>[19]</sup>为代表的学者建议根据用户意愿来分析定位此类有问题的上下文。除此之外,还有一系列工作建议以丢弃上下文为主。例如, Chomicki 等<sup>[20]</sup>采用 drop-latest 策略,期望通过将造成不一致的最近一

次上下文信息或任意一上下文信息丢弃,从而达到修复不一致性的目的; Bu 等<sup>[21]</sup>则建议丢弃造成不一致问题的所有上下文信息,即 drop-all 策略; Xu 等<sup>[22]</sup>则通过对上下文信息与汇报的不一致进行参与度计数的方式,对上下文可能存在问题的程度进行评估,从而丢弃造成不一致性可能性最大的上下文(将具有最大计数),即 drop-bad 策略。也有另外一些工作在对可能存在问题的上下文进行验证的同时对其进行修复。例如, Xu 等<sup>[23-24]</sup>通过对各类上下文的修复可能带来的影响进行评估,提出根据评估结果选择能够带来应用最小 side-effect 的修复策略。另外一类流派则研究如何合理看待应用程序上下文中可能存在的的多不一致问题,探讨是否存在在运行时验证中可压缩或可容忍的不一致<sup>[25-26]</sup>。

总的来说,在运行时对原始感知数据与上下文信息数据的质量进行有效检测并修复,是保障环境感知自适应软件在运行时的可靠性的基础。该类研究从保障软件最底层环境数据质量的角度,探讨如何有效地从根源上避免外部感知输入中可能存在的质量问题,从而避免其影响后续软件决策。然而,由于这部分验证技术或单纯从感知数据出发,或只考虑感知数据语义的基本一致性要求,并未深入探讨应用程序内部的决策逻辑,因此验证通过或问题被修复的上下文输入在后续决策模块中仍然有可能造成决策质量问题,从而影响软件运行时刻决策的可靠性,因此针对决策模块进行独立的运行时输入验证仍然不可或缺。

## 3.2 自适应决策模块

本节将调研针对自适应决策模块进行运行时输入验证的相关技术,分析如何从决策角度对自适应决策模块的输入进行验证,从而避免不期望的输入对决策的影响。由于传统基于规则的决策与基于学习的决策在架构上存在重大区别,下文将从传统基于规则的决策与基于学习的决策两大类来分别分析已有技术。

### 3.2.1 基于规则的自适应决策

传统基于规则的决策由于具有显示的规则,其运行逻辑较明显,通常是以类似状态—动作—状态转换关系的形式,通过规则显示来规定何种状态下进行何种决策。例如,在典型的智能家居系统中,用户可以自定义一系列决策规则,如当房间温度高于某个值时自动打开空调进行降温,或者厨房一氧化碳监测设备监测到一氧化碳浓度高于某个值时自动打开换气设备进行换气。

由于此类基于规则的自适应决策的规则逻辑明显,且运行时刻决策逻辑相关的软件状态易监控,因此已有工作对其决策进行运行时输入验证通常是通过对输入在运行时的应用状态进行实时特征的监控来实现的。例如, Raz 等<sup>[27]</sup>对数值来源进行抽取形成值约束,并实时监测运行过程中是否出现任何值违反情况; Zheng 等<sup>[28]</sup>通过挖掘各程序点需要满足的约束规则信息来实时监控运行; Nadi 等<sup>[29]</sup>则从代码中挖掘配置约束来帮助进行监控; Xu 等<sup>[3]</sup>对运行时刻的常见错误类型进行总结,并监控运行时是否出现这类错误; Xu 等<sup>[30]</sup>提出基于方法调用的上下文来区分程序异常行为状态; Yang 等<sup>[31]</sup>针对上下文感知程序进行运行时反例汇报,并最大化反例出现的概率; Qin 等<sup>[32]</sup>挖掘并抽取程序运行不定式,并利用抽

取的各种不定式来帮助在运行时刻识别程序中出现的违反不定式的异常状态,从而识别不符合预期的输入。

3.2.2 基于学习的自适应决策

新型基于学习的决策与传统基于规则的决策的最大不同点在于,前者内置核心的人工智能模型本身基于统计的特性,决策本身的正确性存在不确定性,因此容易导致软件运行时出现异常决策。然而,由于运行时刻模型本身固定,且模型本身在决策过程中的程序状态难以理解与分析(例如神经网络模型的中间状态往往是异常高维的数组),已有工作对其进行运行时验证主要集中在直接对模型输入进行验证,并评估模型输入是否是异常输入。

针对人工智能模型的异常输入判定,下面将介绍相关工作的几大主要流派。

一大流派考虑运行时输入是否与本身模型在开发过程中训练时的大样本数据存在分布出入(即输入样本是否存在 out-of-distribution 或 OOD 问题),若有则认为输入验证失败。例如,Hendrycks 等<sup>[33]</sup>基于对输入在模型预测时相关的最终 softmax 分数来辅助区分模型输入是否 OOD;Liang 等<sup>[34]</sup>提出 ODIN 方法来进一步提升利用 softmax 分数进行 OOD 输入与非 OOD 输入区分的验证效果;Lee 等<sup>[35]</sup>提出的 Mahalanobis 方法,利用不同类下的条件 Gaussian 分布对输入进行置信度分数计算,从而有效验证 OOD 输入;Kim 等<sup>[36]</sup>则通过分析运行输入的 surprise 程度来帮助验证输入是否足够有价值;Zhang<sup>[37]</sup>在 DeeRoad 技术中对输入预测行为的投影进行最小距离分析,并将结果作为子模块验证模型的输入。

另一大流派关注于 adversarial 样本<sup>[38]</sup>(即受人为攻击插入扰动后的非自然样本),并寻找手段来帮助验证此类样本在运行时的有效性。例如,Wang 等<sup>[39]</sup>模仿软件的变异测试,对此类样本行为在模型变异体上的异常表现进行观察,并提出 mMutant 方法对其进行有效验证;Xu 等<sup>[40]</sup>利用 feature squeezing 来检测输入是否存在任何攻击痕迹,并利用额外模型进行输入区分;Meng 等<sup>[41]</sup>与 Feinman 等<sup>[42]</sup>同样致力于开发新的用于输入区分的模型作为 detector;Liang 等<sup>[43]</sup>另辟蹊径,将攻击痕迹看作图像噪声,并利用噪声平滑技术帮助进行输入验证;Gebhart 等<sup>[44]</sup>则将模型看作信息流图,提出利用持续同调技术来可视化输入在模型中的信息流向,以辅助检测攻击痕迹,从而验证输入。

还有另外一些工作特别地考虑了运行时输入与模型本身的匹配程度,由于模型本身具备的能力有限,因此对于特定输入,需要在考虑与部署模型适配的情况下进行验证。Syn-Eva<sup>[45]</sup>和 VISON<sup>[46]</sup>对决策核心部署的预测模型进行辅助模型的自动生成,并基于此对运行时输入与当前模型的适配程度进行打分,从而帮助进行输入验证;DISSECTOR 工作<sup>[47]</sup>则提出直接对输入在模型中的预测行为进行剖析,并观察其行为模式,从而对模型进行有效地输入验证。

4 未来的研究方向

随着自动化技术与大数据的蓬勃发展,环境感知自适应软件得到了进一步的发展,在复杂环境中如何利用运行时输入验证来保障其运行可靠性也得到了相关领域工作者的广泛

研究。然而,在验证的效率与效果,以及输入验证技术对软件的适用程度等方面,仍然存在研究空间。

(1)在环境感知部分的上下文信息验证方面,已有工作在验证效果上的有效性明显,其中以 Wang 等提出的 GEAS 策略<sup>[17]</sup>搭配 Xu 等提出的 PCC 检测技术<sup>[13]</sup>的组合对实际上下文信息验证的效果与效率最佳,但其验证效率与现实复杂场景下的环境变化频率之间仍然存在一定距离。更严重的是,由于大数据时代下环境数据的变化速度与日俱增,这使得实际应用场景对上下文验证的高效性的实际需求也随之增长,因此在如何不断追求检测效率的同时,寻找到最佳的检测效果,始终是个值得研究的课题。

(2)基于学习的决策模式属于新型人工智能与软件结合的表现形式,还处于新的发展阶段,这样的新型决策模式在环境感知自适应软件的运行过程中如何获得高效的输入验证,仍然存在很大挑战。相比着重从输入样本层面验证其是否符合预期(OOD 或 adversarial),在运行时刻针对特定部署模型的输入验证来考量部署模型的实际能力并有针对性地验证输入,确实能够更加具有针对性地实现运行时的输入验证,然而 DISSECTOR<sup>[47]</sup>作为这一方面的代表工作,虽然考虑了模型在特定能力范围下对输入验证的初步探索,但对实际输入验证中的效果仍然存在不少改进空间,这既需要软件工程技术对输入进行相关分析,也受制于人工智能领域对模型本身能力的理解与研究。而在验证效率方面,验证高效性的需求也在不断地加剧,主要原因有两点:1)环境变化频繁导致应用对输入验证高效性的需求增长;2)与软件集成的人工智能模型本身越来越复杂。因此,如何高效地验证实际输入对决策模块的影响,依旧存在很大的挑战与进步空间。

(3)本文调研中已有的输入验证技术基本遵循软件决策逻辑在部署运行时刻保持独立且不发生变化的基本假设,这基本符合当下软件的特征。然而,随着环境感知自适应软件的智能化,尤其是与人工智能模型结合技术的进一步发展,运行过程中软件决策逻辑受模型学习能力的促使而不断演化的可能性不可忽视,甚至可能在未来成为软件决策逻辑的常态特征。因此,当运行部署中的软件不停学习与演化,其内部决策逻辑也同时不断变化时,已有的各项输入验证技术是否能够依旧适用,以及如何开发新的适用于此场景的输入验证技术,使得其能够在决策逻辑演化的过程中依旧高效保障软件的可靠运行,都是需要未来不断研究与探索的课题。

**结束语** 受人工智能技术的影响,环境感知自适应软件更加智能地与环境进行着交互,也随之让其在运行时面临着环境复杂性与人工智能预测的不确定性带来的巨大可靠性挑战。本文综述了已有相关工作如何在运行时刻通过输入验证的方式来帮助环境感知自适应软件更加可靠地运行。本文着重调研了针对环境感知自适应软件感知模块与决策模块这两大模块的质量验证。其中,基于学习的决策模块的输入验证作为传统软件与人工智能技术结合的核心代表,是当下对环境感知自适应软件研究的热门课题,这也更加促使环境感知自适应软件更稳健地向未来自成长型软件理念迈步。本文对此模块的调研工作,也为未来自成长型软件的产生与质量保障提供了支撑。

## 参 考 文 献

- [1] EI K X, CAO Y Z, YANG J F, et al. DeepXplore: Automated Whitebox Testing of Deep Learning Systems[C]// Proceedings of the 26<sup>th</sup> Symposium on Operating Systems Principles (SOSP). ACM, 2017: 1-18.
- [2] BU L, XIONG W, LIANG C J M, et al. Systematically Ensuring the Confidence of Real-time Home Automation IoT Systems [J]. ACM Transactions on Cyber-Physical Systems, 2018, 2(3): 1-23.
- [3] XU C, CHEUNG S C, MA X X, et al. ADAM: Identifying Defects in Context-aware Adaptation [J]. The Journal of Systems and Software (JSS), 2012, 85(12): 2812-2828.
- [4] JEFFERY S R, GAROFALAKIS M, FRANKLIN M J. Adaptive Cleaning for RFID Data Streams[C]// International Conference on Very Large Data Bases (VLDB). 2006: 163-174.
- [5] WU Z L, LI C H, NG J K Y, et al. Location Estimation via Support Vector Regression [J]. IEEE Trans. on Mobile Computing (TMC), 2007, 6(3): 311-321.
- [6] RAO J, DORAISWAMY S, THAKKAR H, et al. A deferred cleansing method for RFID data analytics[C]// Proceedings of the 32<sup>nd</sup> International Conference on Very Large Data Bases (VLDB). Seoul, Korea, 2006: 175-186.
- [7] KHOUSSAINOVA N, BALAZINSKA M, SUCIU D. Towards Correcting Input Data Errors Probabilistically Using Integrity Constraints[C]// 5<sup>th</sup> International ACM Workshop on Data Engineering for Wireless and Mobile Access. Chicago, Illinois, USA, 2006: 43-50.
- [8] SUBRAMANIAM S, PALPANAS T, PAPADOPOULOS D, et al. Online Outlier Detection In Sensor Data Using Non-parametric Models[C]// Proceedings of the 32<sup>nd</sup> International Conference on Very Large Data Bases (VLDB). New York: ACM Press, 2006: 187-198.
- [9] ZHUANG Y, CHEN L. In-network outlier cleaning for data collection in sensor networks[C]// Proceedings of the 1<sup>st</sup> International VLDB Workshop on Clean Databases (CleanDB). New York: ACM Press, 2006: 41-48.
- [10] YANG W H, XU C, LIU Y P, et al. Verifying Self-adaptive Applications Suffering Uncertainty[C]// Proceedings of the 29<sup>th</sup> IEEE/ACM International Conference on Automated Software Engineering (ASE). 2014: 199-209.
- [11] JAIN A, CHANG E Y, WANG Y F. Adaptive Stream Resource Management Using Kalman Filters [C] // Proceedings of the ACM SIGMOD International Conference on Management of Data (SIGMOD). ACM Press, 2004: 11-22.
- [12] DESHPANDE A, GUESTRIN C, MADDEN S R. Using Probabilistic Models for Data Management in Acquisitional Environments[C]// 2<sup>nd</sup> Biennial Conference on Innovative Data Systems Research. Asilomar, California, USA, 2005: 1-13.
- [13] XU C, CHEUNG S C, CHAN W K, et al. Partial Constraint Checking for Context Consistency in Pervasive Computing [J]. ACM Transactions on Software Engineering and Methodology (TOSEM), 2010, 19(3): 1-61.
- [14] NENTWICH C, CAPRA L, EMMERICH W, et al. Xlinkit: A Consistency Checking and Smart Link Generation Service [J]. ACM Transaction on Internet Technology (TOIT), 2002, 2(2): 151-185.
- [15] XU C, LIU Y P, CHEUNG S C, et al. Towards Context Consistency by Concurrent Checking for Internetwork Applications [J]. Science China Information Sciences (SCIS), 2013, 56(8): 1-20.
- [16] SUI J, XU C, CHEUNG S C, et al. Hybrid CPU-GPU Constraint Checking: Towards Efficient Context Consistency [J]. Information and Software Technology (IST), 2016, 74: 230-242.
- [17] WANG H Y, XU C, GUO B Y, et al. Generic Adaptive Scheduling for Efficient Context Inconsistency Detection [J]. IEEE Transactions on Software Engineering (TSE), 2019, PP(99): 1-1.
- [18] INSUK P, LEE D, HYUN S J. A Dynamic Context-conflict Management Scheme for Group-aware Ubiquitous Computing Environments[C]// Proceedings of the 29<sup>th</sup> Annual International Computer Software and Applications Conference. IEEE Computer Society, 2005: 359-364.
- [19] RANGANATHAN A, CAMPBELL R H. An infrastructure for context-awareness based on first order logic [J]. Personal and Ubiquitous Computing (PUC), 2003, 7(6): 353-364.
- [20] CHOMICKI J, LOBO J, NAQVI S. Conflict Resolution Using Logic Programming [J]. IEEE Transactions on Knowledge and Data Engineering (TKDE), 2003, 15: 244-249.
- [21] BU Y, GU T, TAO X, et al. Managing Quality of Context in Pervasive Computing[C]// Proceedings of the Sixth International Conference on Quality Software. IEEE Computer Society, 2006: 193-200.
- [22] XU C, CHEUNG S C, CHAN W K, et al. Heuristics-based Strategies for Resolving Context Inconsistencies in Pervasive Computing Applications[C]// Proceedings of the 28<sup>th</sup> International Conference on Distributed Computing Systems (ICDCS). 2008: 713-721.
- [23] XU C, CHEUNG S C, CHAN W K, et al. On Impact-oriented Automatic Resolution of Pervasive Context Inconsistency[C]// Proceedings of the 6<sup>th</sup> Joint Meeting of the European Software Engineering Conference and the ACM SIGSOFT Symposium on the Foundations of Software Engineering (ESEC/FSE). 2007: 569-572.
- [24] XU C, MA X X, CAO C, et al. Minimizing the Side Effect of Context Inconsistency Resolution for Ubiquitous Computing [C]// Proceedings of the 8<sup>th</sup> ICST International Conference on Mobile and Ubiquitous Systems (MOBIQUITOUS). 2011: 285-297.
- [25] NUSEIBEH B, EASTERBROOK S, RUSSO A. Making Inconsistency Respectable in Software Development [J]. The Journal of Systems and Software, 2001, 58: 171-180.
- [26] XU C, XI W, CHEUNG S C, et al. CINA: Suppressing the Detection of Unstable Context Inconsistency [J]. IEEE Transactions on Software Engineering (TSE), 2015, 41(9): 842-865.
- [27] RAZ O, KOOPMAN P, SHAW M. Semantic anomaly detection in online data sources[C]// Proceedings of the 24<sup>th</sup> International Conference on Software Engineering. 2002: 302-312.



- [28] ZHENG W, LYU M, XIE T. Test selection for result inspection via mining predicate rules[C]//Proceedings of the 31<sup>st</sup> International Conference on Software Engineering. 2009;215-225.
- [29] NADI S, BERGER T, KASTNER C, et al. Where Do Configuration Constraints Stem From An Extraction Approach and An Empirical Study [J]. IEEE Transactions on Software Engineering, 2015, 41(8):820-841.
- [30] XU K, TIAN K, YAO D, et al. A Sharper Sense of Self; Probabilistic Reasoning of Program Behaviors for Anomaly Detection with Context Sensitivity[C]//Proceedings of the 46th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN). 2016;467-478.
- [31] YANG W H, XU C, PAN M X, et al. Efficient Validation of Self-adaptive Applications by Counterexample Probability Maximization [J]. The Journal of Systems and Software (JSS), 2018, 138:82-99.
- [32] QIN Y, XIE T, XU C, et al. CoMID; Context-based Multi-invariant Detection for Monitoring Cyber-physical Software [J]. IEEE Transactions on Reliability (TR), 2020, 69(1):106-123.
- [33] HENDRYCKS D, GIMPEL K. A baseline for detecting misclassified and out-of-distribution examples in neural networks[C]//Proceedings of 5<sup>th</sup> International Conference on Learning Representations (ICLR). 2017.
- [34] LIANG S, LI Y X, SRIKANT R. Principled detection of out-of-distribution examples in neural networks[C]//Proceedings of 5<sup>th</sup> International Conference on Learning Representations (ICLR). 2018.
- [35] LEE K M, LEE K, LEE H, et al. A Simple Unified Framework for Detecting Out-of-Distribution Samples and Adversarial Attacks[C]//Advances in Neural Information Processing Systems (NIPS). 2018;7167-7177.
- [36] KIM J H, FELDT R, YOO S. Guiding deep learning system testing using surprise adequacy[C]//Proceedings of the 41<sup>st</sup> ACM/IEEE International Conference on Software Engineering (ICSE). 2019.
- [37] ZHANG M S, ZHANG Y Q, ZHANG L M, et al. DeepRoad: GAN-based metamorphic testing and input validation framework for autonomous driving systems[C]//Proceedings of the 33<sup>rd</sup> ACM/IEEE International Conference on Automated Software Engineering (ASE). ACM, 2018;132-142.
- [38] GOODFELLOW I J, SHLENS J, SZEGEDY C. Explaining and harnessing adversarial examples[C]//The 32<sup>nd</sup> International Conference on Machine Learning (ICML). 2015.
- [39] WANG J Y, DONG G L, SUN J, et al. Adversarial Sample Detection for Deep Neural Network through Model Mutation Testing[C]//In Proceedings of the 41<sup>st</sup> ACM/IEEE International Conference on Software Engineering (ICSE). 2019.
- [40] XU W L, EVANS D, QI Y. Feature Squeezing; Detecting Adversarial Examples in Deep Neural Networks[C]//Network and Distributed System Security Symposium. 2018.
- [41] MENG D Y, CHEN H. MagNet; a Two-Pronged Defense against Adversarial Examples[C]//Proceedings of ACM Conference on Computer and Communications Security (CCS). 2017.
- [42] FEINMAN R, CURTIN R R, SHINTRE S, et al. Detecting Adversarial Samples from Artifacts[J]. arXiv:1703.00410, 2017.
- [43] LIANG B, LI H, SU M, et al. Detecting Adversarial Examples in Deep Networks with Adaptive Noise Reduction [J]. IEEE Transactions on Dependable & Secure Computing, 2017, PP (99).
- [44] GEBHART T, SCHRATER P. Adversary Detection in Neural Networks via Persistent Homology [J]. arXiv:1711.10056, 2017.
- [45] QIN Y, WANG H Y, XU C, et al. SynEva; Evaluating ML Programs by Mirror Program Synthesis[C]//Proceedings of the International Conference on Software Quality, Reliability and Security (QRS). 2018;171-182.
- [46] CHEN Z Q, WANG H Y, XU C, et al. Vision; Evaluating Scenario Suitableness for DNN Models by Mirror Synthesis[C]//In Proceedings of the 26<sup>th</sup> Asia-Pacific Software Engineering Conference (APSEC). IEEE Computer Society, 2019;78-85.
- [47] WANG H Y, XU J W, XU C, et al. DISSECTOR; Input Validation for Deep Learning Applications by Crossing-layer Dissection [C]//Proceedings of the 42<sup>nd</sup> ACM/IEEE International Conference on Software Engineering (ICSE). 2020.



**WANG Hui-yan**, born in 1993, postgraduate, is a member of China Computer Federation. Her main research interests include context management, input validation, and intelligent software testing and analysis.



**XU Chang**, born in 1977, Ph.D, professor, is a senior member of China Computer Federation. His main research interests include big data software engineering, and intelligent software testing and analysis.