软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software,2021,32(7):1999−2015 [doi: 10.13328/j.cnki.jos.006272]
©中国科学院软件研究所版权所有.
E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel: +86-10-62562563

不确定环境下 hCPS 系统的形式化建模与动态验证*

安冬冬[1], 刘 静[2], 陈小红[2], 孙海英[2]

[1](上海师范大学 信息与机电工程学院,上海 201418)

[2](华东师范大学 软件工程学院,上海 200062)

通讯作者: 刘静, E-mail: jliu@sei.ecnu.edu.cn

摘 要: 随着科技的进步,新型复杂系统,例如人机物融合系统(human cyber-physical systems,简称 hCPS),已与人类社会生活越来越密不可分.软件系统所处的信息空间与人们日常生活所处的物理空间日渐融合.物理空间内环境的复杂多变、时空数据的爆发增长以及难以预料的人类行为等不确定因素威胁着系统安全.由于系统安全需求的增长,系统的规模和复杂度随之增加所带来的一系列问题亟待解决.因此,在不确定性环境下,构造智能、安全的人机物融合系统已成为软件行业所面临的不可回避的挑战.环境不确定性使得人机物融合系统软件无法准确感知其所处的运行环境.感知的不确定性将导致系统的误判,从而影响系统的安全性.环境不确定性使得系统设计人员无法为人机物融合系统软件的运行环境提供准确的形式化规约.而对于安全要求较高的系统,准确的形式化规约是保证系统安全的首要条件.为了应对规约的不确定性,提出时空数据驱动与模型驱动相结合的建模方式,即通过使用机器学习算法,基于环境中时空数据对环境进行建模.根据安全软件的典型特征,采用动态验证的方式保证系统的安全,从而构建统一而安全的理论框架.为了展示方案的可行性,以自动驾驶车辆与人驾驶的摩托车的交互场景为例说明了在不确定性环境下的人机物融合系统的建模与验证的具体应用.

关键词: 人机物融合系统;机器学习;不确定性建模;形式化验证;统计模型检测

中图法分类号: TP311

中文引用格式: 安冬冬,刘静,陈小红,孙海英.不确定环境下 hCPS 系统的形式化建模与动态验证.软件学报,2021,32(7):1999−2015. http://www.jos.org.cn/1000-9825/6272.htm

英文引用格式: An DD, Liu J, Chen XH, Sun HY. Formal modeling and dynamic verification for human cyber physical systems under uncertain environment. Ruan Jian Xue Bao/Journal of Software, 2021,32(7):1999−2015 (in Chinese). http://www.jos.org.cn/1000-9825/6272.htm

Formal Modeling and Dynamic Verification for Human Cyber Physical Systems under Uncertain Environment

AN Dong-Dong[1], LIU Jing[2], CHEN Xiao-Hong[2], SUN Hai-Ying[2]

[1](The College of Information, Mechanical and Electrical Engineering, Shanghai Normal University, Shanghai 201418, China)

[2](Software Engineering Institute, East China Normal University, Shanghai 200062, China)

Abstract: With the development of technology, new complex systems such as human cyber-physical systems (hCPS) have become indistinguishable from social life. The cyberspace where the software system located is increasingly integrated with the physical space of

* 基金项目: 国家重点研发计划(2019YFA0706404); 国家自然科学基金(61972150); 上海市知识服务平台(ZF1213); 上海市科技计划(20ZR1416000); 上海市青年科技英才杨帆计划(21YF1432900)

Foundation item: National Key R&D Program of China (2019YFA0706404); National Natural Science Foundation of China (61972150); Knowledge Service Platform of Shanghai (ZF1213); Shanghai Science and Technology Committee (20ZR1416000); Shanghai Sailing Program (21YF1432900)

本文由“面向非确定性的软件质量保障方法与技术”专题特约编辑陈俊洁副教授、汤恩义副教授、何啸副教授以及马晓星教授推荐.

收稿时间: 2020-09-17; 修改时间: 2020-10-26; 采用时间: 2020-12-14; jos 在线出版时间: 2021-01-22

© 中国科学院软件研究所 http://www.jos.org.cn

people's daily life. The uncertain factors such as the dynamic environment in the physical space, the explosive growth of the spatio-temporal data, as well as the unpredictable human behavior are all compromise the security of the system. As a result of the increasing security requirements, the scale and complexity of the system are also increasing. This situation leads to a series of problems that remain unresolved. Therefore, developing intelligent and safe human cyber-physical systems under uncertain environment is becoming the inevitable challenge for the software industry. It is difficult for the human cyber-physical systems to perceive the runtime environment accurately under uncertain surroundings. The uncertain perception will lead to the system's misinterpretation, thus affecting the security of the system. It is difficult for the system designers to construct formal specifications for the human cyber-physical systems under uncertain environment. For safety-critical systems, formal specifications are the prerequisites to ensure system security. To cope with the uncertainty of the specifications, a combination of data-driven and model-driven modeling methodology is proposed, that is, the machine learning-based algorithms are used to model the environment based on spatio-temporal data. An approach is introduced to integrate machine learning method and runtime verification technology as a unified framework to ensure the safety of the human cyber-physical systems. The proposed approach is illustrated by modeling and analyzing a scenario of the interaction of an autonomous vehicle and a human-driven motorbike.

Key words: human cyber physical system; machine learning; uncertainty modeling; formal verification; statistical model checking

随着计算机科学技术的发展,各种新型复杂系统快速涌现,尤其是与人类社会交互和协作的复杂系统越来越受到人们的关注.信息物理融合系统(cyber physical system,简称 CPS)是一种综合计算、网络和物理环境的多维复杂系统,通过 3C 技术,即计算(computation)、通信(communication)和控制(control)的有机融合与深度协作,实现大型工程系统的实时感知、动态控制和信息服务[1].CPS 系统的本质就是以"人-机-物"融合为目标的计算技术,以实现人的控制在时间、空间等方面的延伸.人机物融合系统(human cyber-physical-system,简称 hCPS)是具有深度集成的人、网络和物理元素的智能联网系统[2],是在信息物理系统 CPS 的基础上,重点考虑系统所处的不确定环境以及环境中人的因素.例如,与驾驶员和行人交互的自动驾驶系统、与医生合作的智慧医疗系统以及与用户互动的智能家居系统等.

作为人机物系统的核心,hCPS 软件系统所处的信息空间与人们日常生活所处的物理空间日渐融合.作为人机物融合系统主体的智能体(agent)在与所处的环境(environment)不断进行交互的过程中,智能体需要像人一样具有学习能力,拥有"智能性"以应对各种环境."智能性"主要是指计算智能,是以数据为基础,通过训练建立联系进行问题求解,人工神经网络、遗传算法、模糊系统、进化程序设计等都属于计算智能.由于环境与过程的时空特性,hCPS 系统行为的响应不仅与所经历的时间有关,同时也与所处的空间相关,具体表现为计算过程要求严格的时间约束,而物理过程具有空间依赖性.同样,随着万物互联的发展,人成为 hCPS 系统中的重要组成部分.由于许多 hCPS 都发挥着至关重要的作用(例如,在能源、医疗保健、国防安全、智能驾驶等方面),因此这些系统的"安全性"也同样至关重要."安全性"是系统所在环境的函数,是对任何环境下系统零风险程度的度量[3].Sun 等人[4]指出:"确保软件的安全性是指在任何环境中各种软件的运行不会导致系统出现不可预测的风险,具有避免灾难事故发生的能力".物理空间内环境的复杂多变、时空数据的爆发增长以及难以预料的人类行为等不确定因素威胁着系统的安全性.智能体如何准确感知并捕获不确定环境中的有效信息并及时做出准确且安全的决策是目前研究人员重点关注的问题.

由于人机物融合系统所涉及的研究系统比较广泛,不同系统所涉及的人的因素各不相同,本文中我们以无人驾驶系统作为人机物融合系统的典型案例进行分析和研究.无人驾驶是典型的 hCPS 应用领域,未来将是人机共驾、有人无人驾驶车辆共存的局面.在这样的混合交通场景中,无人驾驶车辆作为智能体不可避免地需要与所处环境中的其他车辆以及车辆内部和外部的人进行交互[5].这对无人驾驶系统的设计提出了更高的要求.一方面,系统需要有应对环境的"智能性",拥有迅速且准确的决策能力,可以自主地应对驾驶过程中常常遇到的、偶发的各种各样的不确定性事件.我们知道,人类驾驶员在突发事件的情况下,会因为情绪失控而导致不理性决策产生.对人来说,这类事故是小概率事件,因而很难通过对事故的"练习"来帮助驾驶员积累处理经验.而机器学习(machine learning,简称 ML)技术可以帮助自动驾驶系统应用所感知到的时空数据进行及时而有效的计算,从而分析周边环境以产生决策.因而无人驾驶系统可以在事故发生时,精准且理性地应对,从而最大限度地

© 中国科学院软件研究所 http://www.jos.org.cn

减少事故的发生.机器学习可以帮助我们提高系统的“智能性”,但是并不能保证系统的“安全性”.数据驱动的方式可能会由于某个数据的变化导致学习出的结果是错误的,从而产生无法预料的后果,这对安全的系统是不可接受的[6−8].该方案显示出了对于数据的敏感性,由于机器学习算法本身可解释性较差的缺点导致系统无法预知所学习的结果.所以,仅仅依靠机器学习的方法无法保证无人驾驶系统的安全性.目前,无人驾驶由于安全问题的不断出现仍然面临巨大的挑战.2016 年,一辆特斯拉汽车由于传感器失灵没有及时感知到前方的障碍物而与之发生碰撞[9].2018 年,Uber 的无人驾驶汽车由于夜间识别能力较差没有及时感知到前方的行人而加以避让从而发生车祸事故,导致行人的死亡[10].城市交通环境日趋复杂,面对自动驾驶场景复杂多变、时空数据的爆发以及其他人工驾驶车辆的主观驾驶行为所导致的环境中不确定性的增加,因此,如何避免事故的发生以保证系统的“安全性”仍是目前的研究难点.

随着 hCPS 应用空间的扩大,开发具有安全保证的设计框架至关重要.在对 hCPS 建模的过程中,需要重点考虑 hCPS 动态行为的 3 个特性.(1) 混成性:hCPS 依赖行为离散的计算系统,同时也依赖行为连续的物理环境(例如:连续的温度变化、时间及空间的变化等),其异构的本质导致其行为具有混成性;(2) 随机性:hCPS 系统处在开放的环境中,不确定的环境(例如:环境中人的行为不确定、用户的行为不确定以及天气的变化等),造成 hCPS 的行为具有随机性;(3) 安全性:hCPS 经常被用于安全攸关的系统(例如:智慧医疗系统、列车控制系统和智能驾驶系等),因此需保证 hCPS 的行为必须是安全可信的.

形式化方法(formal method,简称 FM)是基于严谨的数学(逻辑)语言和精确的数学语义的方法学,主要应用于对系统软件进行建模分析和验证等工作,以保证系统的正确运行[11].面对系统规模的扩大和复杂度的增加,形式化方法越来越频繁地被应用于安全攸关软件的验证,以此来保证其安全性[12,13].模型检测(model checking,简称 MC)是通过对系统的状态空间进行全遍历,从而对系统的安全性进行自动验证[14],其本质是将一个过程或系统抽象成一个有穷状态机模型加以分析验证.传统的模型检测由于状态爆炸的问题,增大了对系统进行分析验证的难度.统计模型检测技术(statistical model checking,简称 SMC)[15]可以通过评估系统所满足的概率区间对系统进行定量分析,从而有效地解决状态过多而难以验证的问题[16−18].统计模型检测是一种高效的验证技术,常用于复杂的随机系统的验证.该工具所采用的技术主要是参数估计式和假设检验.其中,参数估计是基于所收集到的足够的样本,给出模型满足给定性质的近似概率,属于定量的结果;而假设检验是基于样本给出模型是否满足给定的性质,属于定性的结果.本文主要考虑人类驾驶车辆和自动驾驶车辆共存的混合交通流下的驾驶环境,由于人的行为以及物理环境,从而导致不确定性因素的增加.而 UPPAAL-SMC 具有定性和定量分析的能力,能够分析验证出自动驾驶车辆的相对安全性,因此,本文选用 UPPAAL-SMC 作为自动验证工具.

本文采用时空数据驱动的方法来处理环境的不确定性,目前,针对安全攸关 hCPS 的建模与验证还缺乏统一、系统的理论、方法.围绕“不确定环境下的人机物融合系统的建模与验证”这一关键科学问题,提出了以数据驱动和模型驱动相结合的创新方式构建不确定环境下的 hCPS,并对其关键支撑技术进行创新性研究.

(1) 针对系统所处的物理环境的不确定性,应用机器学习技术,以环境中的时空数据为驱动、环境感知计算为切入点,提出了不确定环境下的基于朴素贝叶斯的人类行为分类模型.

(2) 为了应对大规模系统的结构化建模,构建了不确定环境场景下的 hCPS 系统模型,使用验证工具 UPPAAL-SMC 实现对模型的动态验证.

(3) 提出基于统计模型验证的运行时动态验证技术,采取线下和线上验证相结合的方法,即通过比较参数即可快速得出验证结果,从而达到动态、实时的验证效果,进而保障系统在复杂环境中的安全运行.

本文以构建安全智能的人机物融合系统的理论框架及应用作为主要的研究目标.第 1 节主要介绍所涉及到的一些背景知识,包括概念及其主要应用.第 2 节介绍不确定环境下的感知模型,针对系统所处物理环境的不确定性,应用机器学习算法,以环境的时空数据为驱动,提出不确定环境下的基于朴素贝叶斯的人类驾驶行为分类模型.第 3 节提出线下与线上验证相结合的动态验证方法,即通过验证工具 UPPAAL-SMC 实现对模型的动态验证,从而定量评估不确定性环境以及人的行为对系统安全性的影响.第 4 节对相关工作进行比较.最后总结全文,并对未来值得关注的研究方向进行初步探讨.

© 中国科学院软件研究所 http://www.jos.org.cn

1 预备知识

1.1 朴素贝叶斯算法

朴素贝叶斯或简单贝叶斯分类器广泛应用于数据分析.朴素贝叶斯分类器[19],基于关键的约束假设,即属性 X_i 之间彼此独立,也就是 $P(X_i|X_k)=P(X_i)$.

$$P\left(\text{类别}\,|\,\text{特征1,特征2,…,特征}N=\frac{P(\text{类别})P(\text{特征1,特征2,…,特征}N)}{P(\text{特征1,特征2,…,特征}N)}\right).$$

上述公式是朴素贝叶斯分类器的原理,即在给定 n 个特征的情况下,计算事件属于某个类别的概率,概率最大的那个类别即为该事件所属类别.虽然贝叶斯分类原理很简单,但却有着高效的学习效率和准确的分类效果.原因是,我们不需要知道类别概率的精确值 $P(\theta\in 1,2,...,k)$,只需要分类器对它们进行正确的分类.朴素贝叶斯算法被广泛应用在机器学习中,相较于其他监督学习分类算法,其所需考虑的参数更少也更简单、高效.朴素贝叶斯算法主要包括 3 个阶段:准备阶段、学习阶段和预测阶段.在准备阶段,需要进行数据预处理,指定特征属性和类别,获取训练数据.在学习阶段,首先要估计每个类别出现的概率,估计每个类别下每个特征属性出现的概率,然后对于每个属性组合,分别计算其归属于每个类别的概率.在预测阶段,针对输入的特征属性集合,挑选最大概率所属的类别作为最终的分类结果.

1.2 随机混成自动机SHA

考虑到我们的模型中含有时间、速度等连续变量,信号灯表示属于离散变量.同时,驾驶员状态的迁移以及行人对于来往车辆的关注行为都带有不确定的特性.因此,我们用随机混成自动机来构建模型.随机混成自动机(stochastic hybrid automata,简称 SHA)[20]是混成自动机带有随机行为的版本.用于实现对带有随机混成行为的系统建模,即包含离散和连续变化的变量,同时状态之间迁移带有随机行为.本文关注的不确定环境下的人机物系统包含混成行为和随机行为,且对时间约束高度敏感.因此,本文将无人驾驶系统特征映射到随机混成自动机,实现自动化的模型验证.SHA 是对混成自动机 HA 进行了随机语义扩展,状态之间的迁移可基于离散概率分布[21].

随机混成自动机是一个八元组 $M=(L,l,C,Act,inv,enab,prob,F)$,其中,

- L 表示位置的有限集合;
- l 表示初始位置,$l\in L$;
- C 表示时钟的有限集合;
- Act 表示动作的有限集合;
- $inv:L\to CC(C)$表示不变式的有限集合,CC 表示时钟约束函数;
- $enab:L\times Act\to CC(C)$表示迁移触发的条件;
- $prob:L\times Act\to Dist(2^C\times L)$表示概率迁移函数;
- $F:L\to 2^{AP}$ 表示将每个位置映射到原子命题集合的标签函数.

1.3 概率计算树逻辑PCTL

概率计算树逻辑(probabilistic computation tree logic,简称 PCTL)是对计算树逻辑(computation tree logic,简称 CTL)的概率扩展,用概率运算符 P 定量扩展了 CTL 的路径量词“所有(all,简称 A)”和“存在(exists,简称 E)”.PCTL 的详细语义参见文献[22,23].PCTL 语法的状态公式ϕ和路径公式ψ的定义如下所示:

$$\phi := \text{true}\,|\,ap\,|\,\phi\wedge\phi\,|\,\neg\phi\,|\,P_{\bowtie p}(\psi),$$

$$\psi := \phi U^{\wedge}\{\leqslant k\}\phi\,|\,X\phi.$$

其中,AP 表示原子命题集合,$ap\in AP$,$p\in[0,1]$,$k\in N$,$\bowtie\in\{\leqslant,<,>,\geqslant\}$,时序操作符 X 表示 $Next$,U 表示 $Until$.状态公式ϕ中的每个状态被评估为 true 或 false.状态 s 满足关系定义如下:

$$s\models\text{true},s\models ap \quad \text{iff} \quad ap\in inv(s)$$

© 中国科学院软件研究所 http://www.jos.org.cn

$$s \models \neg\phi \quad \text{iff} \quad s \not\models \phi$$
$$s \models \phi_1 \wedge \phi_2 \quad \text{iff} \quad s \models \phi_1 \quad \text{and} \quad s \models \phi_2$$
$$s \models P_{\bowtie p}(\psi) \quad \text{iff} \quad \Pr(s \models \psi) \bowtie p$$

$s|=P_{\bowtie p}(\psi)$中 P 表示在起始状态为 s 的路径集合中满足路径公式 ψ 的概率,$P_{\bowtie p}$ 比较得出的概率 P 与给定的概率值 p 的对比结果$\bowtie \in \{\leqslant,<,>,\geqslant\}$是 true 还是 false.给定一个路径公式$\psi$,第 i 个状态和初始状态分别定义为 $\psi[i]$和$\psi[0]$.对于路径公式ψ的满足关系定义如下:

$$\psi \models X\phi \quad \text{iff} \quad \psi[1] \models \phi$$
$$\psi \models \phi_1 U^{\leqslant k} \phi_2 \quad \text{iff} \quad \exists i, 0 \leqslant i \leqslant k$$
$$\psi[i] \models \phi\_2(\forall i, 0 \leqslant i \leqslant k \quad \psi[i] \models \phi_1)$$

1.4 统计模型检测

统计模型检测(statistical model checking,简称 SMC)问题可以描述为:给定一个随机系统模型 M 和一个性质规约公式 Φ,"统计模型检测技术建立在蒙特卡洛(Monte Carlo)模拟技术之上,它能够有效地评估系统模型满足目标约束的概率区间,对系统进行验证分析".SMC 算法主要分为定性和定量两种类型,其中,定性算法用来验证"系统 M 满足约束ϕ的概率是否大于或者等于某个概率阈值 p,$p \in [0,1]$,即 $s|=P_{\bowtie p}(\psi)$"包括:single sampling plan (SSP)、sequential probability ratio(SPRT)和 Bayesian hypothesis testing(BHT).定量算法用来验证"系统 M 满足约束 ϕ_2 的概率是多少,即 $s|=P_{=?}(\psi_2)$"包括 Bayesian interval estimation(BIE)和 approximate probabilistic model checking(APMC).不同类型的 SMC 算法的主要区别在于统计参数的计算和置信度满足条件的判断这两个方面.

2 基于朴素贝叶斯分类器的人工驾驶风格分类模型

在 hCPS 软件系统所处的信息空间与人们日常生活所处的物理空间日渐融合的情况下,作为人机物融合系统主体的智能体在与所处的环境不断进行交互的过程中,智能体需要像人一样具有学习能力,拥有"智能性"来应对各种环境.无人驾驶系统作为 hCPS 的典型应用系统,无人驾驶汽车是集感知、决策和控制等功能于一体的自主交通工具,其中,感知系统代替人类驾驶过程中人的视、听、触等功能,融合摄像机、雷达等传感器采集的海量交通环境数据,精确识别各类交通元素,为自动驾驶汽车决策系统提供支撑.在无人驾驶和人工驾驶的混合交通场景中,由于人类驾驶员在驾驶过程中会疲劳或者分心,这就使得人类驾驶行为很随机.无人驾驶系统由于高精度的传感、执行、控制,其技术是规定好的,不会退化,在交互过程中,智能化的技术对于未知环境的理解,其认知理解能力是有限的.而目前的一些无人驾驶设计基于规则驾驶的一些模式来设定,有相关的一些逻辑参数,在自适应方面包括人性化方面做得很不到位,这就导致无人驾驶和人工驾驶有可能产生一定的冲突.如何保证系统在不确定的复杂环境下智能运行仍是目前无人驾驶系统设计所面临的挑战.在本节中,为了提高系统的"智能性",我们提出了基于朴素贝叶斯分类器的人工驾驶行为的分类模型(drving style classification,简称 DSC).

本节重点介绍人工驾驶行为的分类模型,一方面从环境中接收数据,将环境数据和智能体自身的相关数据作为模型输入,基于朴素贝叶斯的分类算法对周边人工驾驶车辆的驾驶行为进行分类,学习器的学习结果将作为参数输入到后续的系统模型中.

无人驾驶车辆在行驶过程中要面对大量的不确定因素,在与人工驾驶的车辆交互的过程中,由于人类驾驶员在驾驶过程中的驾驶行为具有很强的主观性,所以在无人驾驶车辆的角度来看,驾驶员的行为状态具有一定的不确定性.例如,面对相同的驾驶环境,不同的驾驶员可能会产生不同的驾驶行为.如果无人驾驶车辆不能及时、准确地判定周围人类驾驶员的驾驶行为,势必会导致本车进入不安全的状态.因此需要对周边车辆的驾驶行为进行分析.本节将重点聚焦于对人工驾驶车辆的驾驶风格进行分类.本节的目标是通过无人驾驶与人工驾驶交互所产生的数据进行分类学习,构建人类驾驶员的驾驶行为分类模型,无人驾驶车辆根据周围人工驾驶车辆的当前状态来对其驾驶行为进行分类,并预测周边车辆的未来状态.从而指导无人驾驶车辆做出智能决策.为了实现该目标,本文提出了基于朴素贝叶斯的驾驶行为分类模型,使用有监督的机器学习训练模型以对数据进

© 中国科学院软件研究所 http://www.jos.org.cn

行分类,该模型的分类准确率将作为概率参数输入到下一节的模型中.

2.1 分类流程

图 1 展示了基于朴素贝叶斯分类器的驾驶行为分类模型,共分为 3 部分,第 1 部分是数据预处理,获取训练样本,第 2 部分是朴素贝叶斯分类器的线下训练阶段,第 3 部分是在线测试阶段.第 1 个模块是数据预处理模块(data preprocess),该模块用于提取特征数据并且形成驾驶行为的特征数据集.在数据预处理阶段,带标签的数据包括无人驾驶车辆自身的行车数据以及与周边人工驾驶车辆的关系数据,将带标签的数据输入样本选取预处理器中后,特征样本存储器会存储特征样本,同时进入线下训练阶段.接着在第 2 个模块,基于贝叶斯的学习(Bayes-based learning)中,使用所得数据来训练状态预测模型,通过该模型,可以判断驾驶行为的类别.在线下训练阶段中,第 1 步是估计每个类别出现的概率,接着估计每个类别条件下每个属性值出现的概率 $p(y_i)$,并对每个属性组合计算属于每个类别的概率 $p(x|y_i)p(y_i)$,最后选择最大概率值 $p(x|y_i)p(y_i)$作为该条件数据的推测结果而输出.在第 3 阶段中,将实时收集到的时空数据输入到训练好的分类模型中,进行模型评估,最后输出对行为预测的结果.

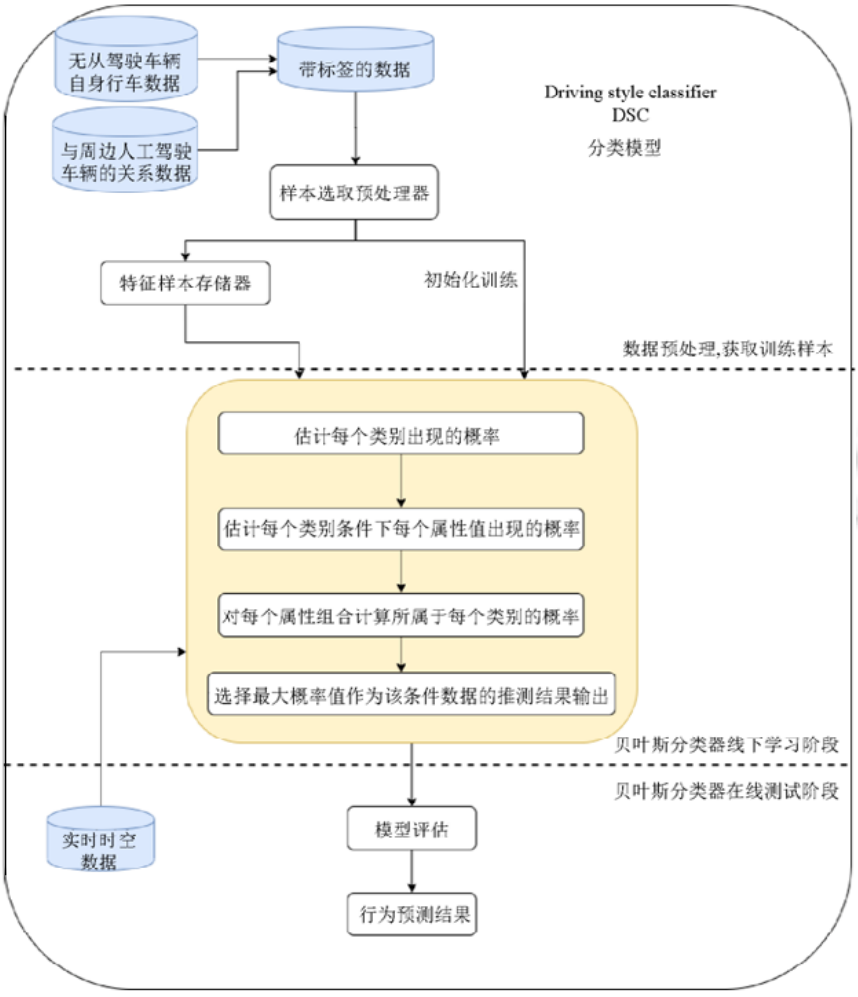

Fig.1 Driving style classification model based on naïve Bayes

图 1 基于朴素贝叶斯分类器的驾驶行为分类模型

下面详细介绍分类器线下训练流程.

© 中国科学院软件研究所 http://www.jos.org.cn

(1) 将已知的输入数据(观测值或示例)和对数据的已知响应(标签或类别)作为数据集输入;

(2) 选择参与训练的特征或者使用 PCA 降维,默认不降维,使用全部特征;

(3) 设定误分类代价,即类别准确率优先级,哪些类别一定要分对,哪些类别可接受一定的误差范围;

(4) 选择训练模型并进行模型参数配置;

(5) 输入样本数据来训练模型,采用并行训练的方式可同时训练多个分类器模型,提高训练效率,该训练模型可以对新数据的响应生成预测;

(6) 分类器效果评价结果包括散点图、混淆矩阵、ROC 曲线等;

(7) 导出训练结果和模型文件,为接下来的线上训练做准备.

2.2 模型评价指标

针对一个二分类问题,我们将实例分为如下两类:正类(positive)和负类(negative).

那么在实际分类中,会出现如下 4 种情况.

(1) 真正类(true positive,简称 TP),即被预测为正类,实际为正类;

(2) 假正类(false positive,简称 FP),即被预测为正类,实际为负类;

(3) 真负类(true negative,简称 TN),即被预测为负类,实际为负类;

(4) 假负类(flase negative,简称 FN),即被预测为负类,实际为正类.

正确率(precision): $Precision=\frac{TP}{TP+FP}$.

真阳性率(true positive rate,简称 TPR),灵敏度(sensitivity),召回率(recall):$TPR=Sensitivity=Recall=\frac{TP}{TP+FN}$.

真阴性率(true negative rate,简称 TNR),特异度(specificity): $TNR=Specificity=\frac{TN}{FP+TN}$.

假阴性率(false negative rate,简称 FNR): $FNR=\frac{FN}{TP+FN}$.

假阳性率(false positive rate,简称 FPR): $FPR=\frac{FP}{TN+FP}$.

$F1$-score: $F1\text{-score}=\frac{2\times Precision\times Recall}{Precision+Recall}$.

2.3 案例研究

如图 2 所示.

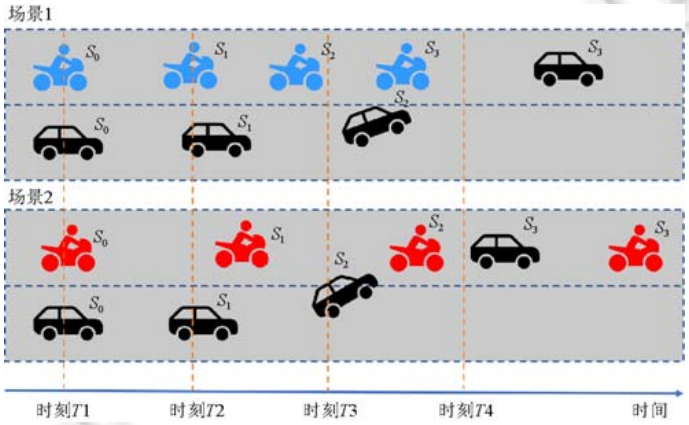

Fig.2 Senario: Autonomous vehicle change to the lane where human-drive vehicle occupied

图 2 驾驶场景:无人驾驶车辆(黑色)变道至人工驾驶车辆(蓝色、红色)所在的车道上

图 2 中,人工驾驶的车辆(蓝色、红色)在直行道上行驶,旁边的车道上由一辆无人驾驶车辆(黑色)同向行驶.

© 中国科学院软件研究所 http://www.jos.org.cn

在该驾驶场景中,无人驾驶车辆(黑色)需要变道至人工驾驶的车辆(蓝色、红色)所在的车道上.图中,$S_0,S_1,\ldots,S_n$表示不同时刻 $T_0,T_1,\ldots,T_n$ 车辆所在的不同位置.上半部分表示场景 1:蓝色人工驾驶车辆减速行驶,黑色无人驾驶车辆通过加速超车后完成变道.下半部分表示场景 2:红色人工驾驶车辆加速行驶,黑色无人驾驶车辆减速后跟在人工驾驶车辆后方完成变道.

2.3.1 数据采集及预处理

首先,需要从传感器获取时空数据,我们将智能体采集到的数据划分为以下 3 种类型,见表 1.

(1) 自身驾驶数据以及与周边人所驾驶车辆的关系数据,主要包括自动驾驶车辆的速度、加速度,与周边车辆的相对速度、相对距离等;

(2) 周边人所驾驶车辆的驾驶数据,主要包括车灯使用是否正确、刹车次数以及是否存在其他不规范的行车行为等;

(3) 周边环境中与车辆无关的信息数据,主要包括天气(例如:雨天、雾天、晴天等)、时间(例如:白天、晚上、早晚高峰等)、道路情况(例如:高速公路、城市道路、乡村道路等)等.

由于缺少实验条件,我们采用仿真的方式生成了近 200 条数据(https://github.com/DongdongAn/DrivingStyleClassification.git).在此场景中,人的行为是间接通过车辆的行驶数据来体现的,所以我们对人类所驾驶车辆的行车行为分类即为对人工驾驶员的驾驶行为进行分类.

Table 1 The input dataset

表 1 输入分类器的数据集

类别	数据
1. 与周边车辆的相关数据	
1) 自动驾驶车辆的速度	0~120km/h
2) 相对距离	1m~20m
3) 相对速度	−15km/h~15km/h
4) 周边车辆的速度	0~120km/h
5) 速度差百分比	<−10%;−10%~10%;>10%
2. 周边车辆其他数据	
1) 车灯使用是否正确	0-使用正确;1-使用错误
2) 刹车次数	0-无刹车;1-有刹车
3) 其他不合规行为	0-无;1-有
3. 周围环境	
1) 天气情况	0-雨天;1-晴天;2-雾天
2) 所处时间	0-夜晚;1-白天
驾驶行为分类	正常-Normal;轻度激进-SlightlyAggressive;激进-Aggressive

2.3.2 分类器效果评价

如图 3 所示,散点图表示了不同驾驶行为的分类情况,不同颜色的点表示不同的分类结果.如图 4 所示,在人工智能中,混淆矩阵(confusion matrix)是可视化工具,特别用于监督学习.混淆矩阵是通过将每个实测像元的位置和分类与分类图像中的相应位置和分类相比较来计算的,从而刻画出一个分类器的分类准确程度来验证分类结果的有效性.如图 5 所示,平行坐标图(parallel coordinates plot)采用的是一种数据可视化的方式.横坐标的每个标记表示样本中的一个属性,如车速、相对距离、相对速度等.纵坐标表示每个样本中该属性的值,相连而得到的一个折线表示该样本.蓝色、红色、黄色分别表示激进驾驶行为、危险驾驶行为和正常驾驶行为.直线表示分类结果与输入时样本的预分类结果一致,虚线表示模型的分类结果与输入时样本的预分类结果不一致.如图 6 所示,ROC(receiver operating characteristic)曲线的横坐标表示:1-Specificity,伪正类率(false positive rate,简称 FPR),预测为正但实际为负的样本占所有负例样本的比例;纵坐标表示:Sensitivity,真正类率(true positive rate,简称 TPR),预测为正且实际为正的样本占所有正例样本的比例.表 2 给出了线下训练结果,每个模型训练后的准确率及其他相关信息都会被显示出来.训练方法主要包括决策树(decision trees)、朴素贝叶斯(naïve Bayes classifiers)、支持向量机(support vector machines)、最近邻 *K*NN(nearest neighbor classifiers)和组合分类器.最终的结果显示,训练准确度较高的是朴素贝叶斯分类器.对于训练好的模型,我们可以输入新的数据进行分类,采

© 中国科学院软件研究所 http://www.jos.org.cn

用样本预测,将线下训练好的模型导出到工作空间,这样就可以利用模型来测试新的数据.

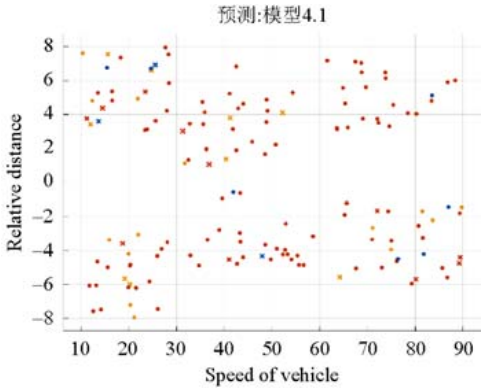

Fig.3 The classification result of different driving style
图 3 不同驾驶行为的分类情况

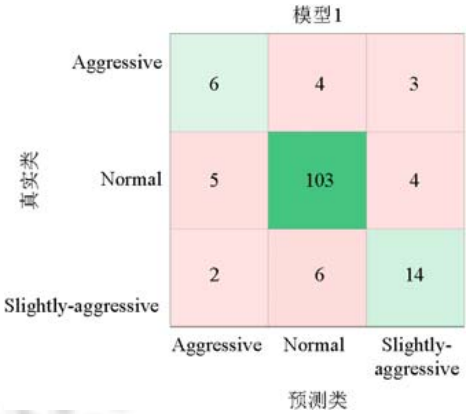

Fig.4 The classification result of confusion matrix
图 4 混淆矩阵结果

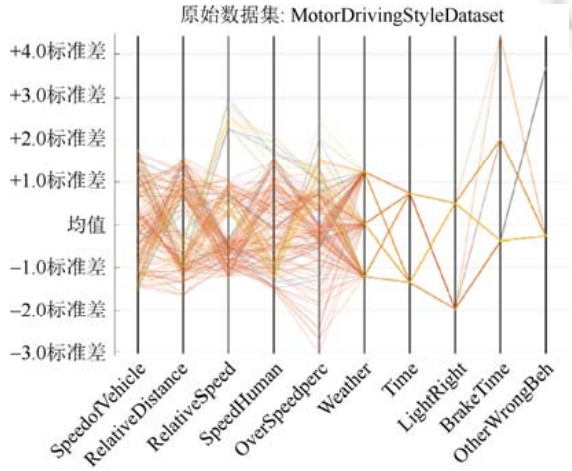

Fig.5 The classification result of parallel coordinates plot
图 5 分类结果的平行坐标图

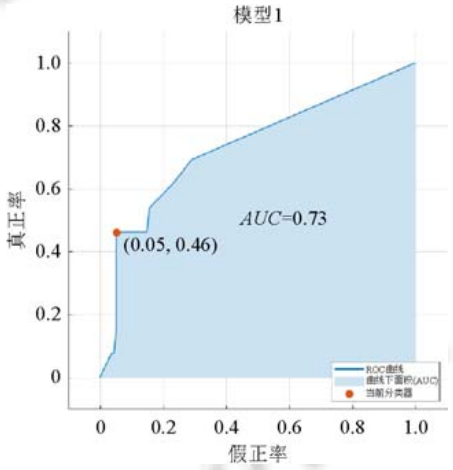

Fig.6 ROC (receiver operating characteristic)
图 6 ROC 曲线结果

3 线下与线上验证相结合的动态验证方法

在上一节我们得到 UPPAAL-SMC 可以验证的模型后,通过写入要验证的性质(query),即可对模型进行验证.但是,目前使用 UPPAAL-SMC 进行验证的时间成本较高,通常需要十几秒钟甚至更长,对于复杂的系统很可能由于要检测的状态过多而产生状态爆炸的问题.由于本文所研究的无人驾驶系统的物理车辆是行驶在不确定且复杂多变的场景中,十几秒钟后才能出现的验证结果对于系统来说是难以接受的,若车辆不能及时地对周边环境做出反应,那么对于无人驾驶车辆来说将造成无法承担的后果.所以我们从机器学习流程中线下训练和线上学习的过程得到启发,通过采取线下验证和线上验证相结合的方法,比较参数后即可快速得出验证结果,从而达到动态、实时的验证效果,进而保障系统在复杂环境中能够安全运行.

图 7 显示了线下与线上结合的动态验证方法示意图.上半部分显示的是线下的验证过程,下半部分所示为线上验证过程.在线下验证过程中,统计模型检测工具 UPPAAL-SMC 接受的是属性和 NSHA 模型,其中所要验证的属性基于不同场景下的安全需求进行形式化描述而产生,而 NSHA 模型由上一节提出的环境感知模型和线下训练场景共同建模得到.不断加入新的线下场景进行验证.这样,在线上验证过程中,通过对比相关安全数据,在验证

© 中国科学院软件研究所 http://www.jos.org.cn

结果库中查询之前线下已经验证出的结果,从而及时给予系统反馈,获得实时、动态验证的效果,使得 hCPS 系统可以智能、安全且及时地应对复杂多变的不确定性环境.对于训练好的驾驶行为分类模型,通过两组新的环境数据和人工驾驶车辆的行车数据,我们得到的人工驾驶行为预测结果为:在场景 1,T_1 时刻,*driver_style=NORMAL* 的概率为 85%;在场景 2,T_2 时刻,*driver_style=AGGRESSIVE* 的概率是 87%.将以上结果作为参数集输入到后续的 NSHA 模型中.

Table 2 The learning results of the classification learner

表 2 分类器学习结果统计

模型	准确率(%)	分类错误数量	预测速度(obs/s)	训练时间(s)
精细树	83.70	24	2 600	1.136 3
中等树	83.70	24	1 700	5.124 8
粗略树	81.60	27	1 700	5.035 7
精细 *K*NN	79.60	30	1 300	4.894 9
中等 *K*NN	78.90	31	1 900	2.064 6
粗略 *K*NN	76.20	35	1 800	2.35
余弦 *K*NN	78.90	31	2 200	3.374 7
3 次 *K*NN	76.90	34	2 800	3.265 2
加权 *K*NN	83.70	24	2 400	3.714
朴素贝叶斯	85.50	22	760	1.616 5
线性 SVM	78.90	31	1 200	3.731 1
二次 SVM	81.60	27	960	3.841 8
3 次 SVM	85.00	22	1 100	4.781 3
精细高斯 SVM	76.20	35	1 400	4.662 3
装袋树	85.00	22	200	11.638
子空间判别	82.30	26	170	12.516
子空间 *K*NN	79.60	30	130	12.403
RUSBooosted 树	76.90	34	230	12.846

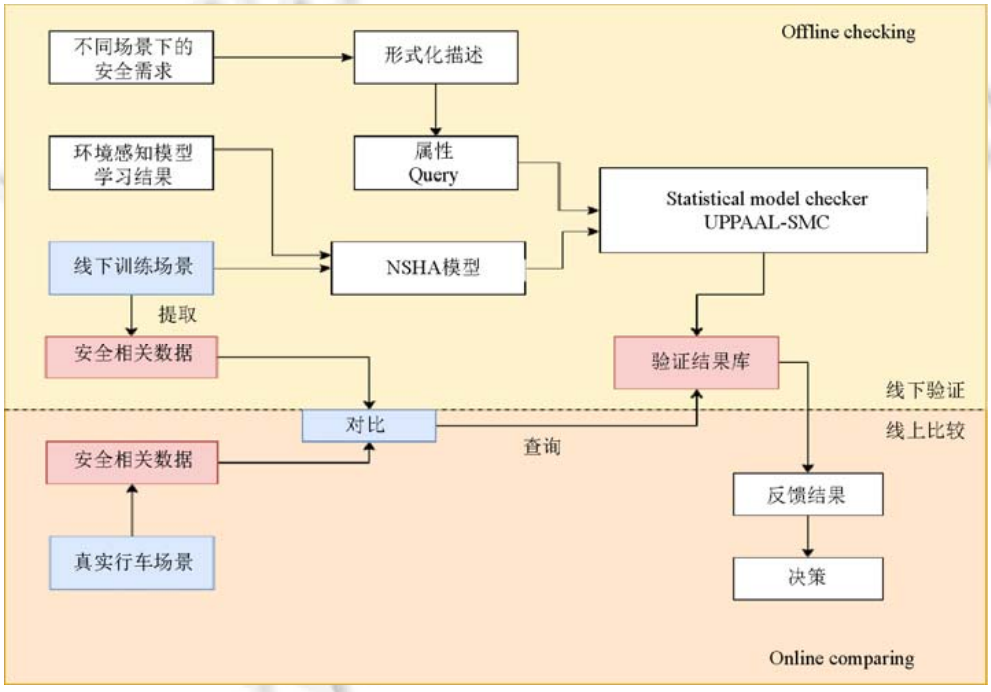

Fig.7 Offline and online verification method

图 7 线下与线上验证相结合的动态验证方法

© 中国科学院软件研究所 http://www.jos.org.cn

3.1 构建NSHA模型

将无人驾驶车辆进行建模,UPPAAL-SMC 可以进行验证 NSHA 模型.所构建出的 NSHA 模型由 5 个子模型组成,分别是组合模型 Composite、开始模型 Start、风险模型 EnvRisk、人工驾驶模型 HumanDrive 和无人驾驶模型 Autodrive.因此,随机混成自动机网络可以表示成:

$$\text{ChangeLane} = \text{Composite} \cup \text{Start} \cup \text{EnvRisk} \cup \text{HumanDrive} \cup \text{Autodrive}.$$

图 8 表示复合状态抽象成的 SHA 模型,主要包括 STOP、STRAIGHT 和 CHANGE_LANE 这 3 个状态和两个 Urgent 状态,在 Urgent 状态中时间不流失,会跳转到下一个状态.受篇幅所限,其他 SHA 模型的建立过程详见 https://github.com/DongdongAn/DrivingStyleClassification.git,这里不再一一列举.

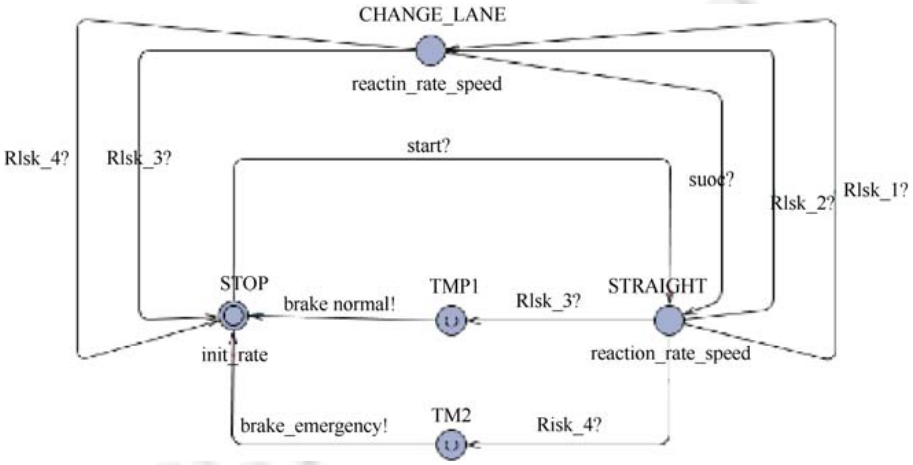

Fig.8 Composite state of SHA model

图 8 复合状态抽象成的 SHA 模型

3.2 采用UPPAAL-SMC对NSHA模型进行验证

建立好 NSHA 模型后,接下来是对其进行验证.首先需要结合 PCTL 公式定义所要验证的性质,见表 3.

(1) $P1$ 表示在 15 个时间单位内人工驾驶车辆允许无人驾驶车辆通过的概率.当人工驾驶风格分类模型预测结果为 normal driver 时,图 9 表示在模拟了 1 305 次之后,所得的概率区间是[0.549862,0.609859],其中,置信度为 0.97.概率密度分布和累积分布如图 10 所示.图中的 x 轴表示时间,y 轴分别表示概率密度和概率.图 11 表示,当人工驾驶风格分类模型预测结果为 aggressive driver 时,概率密度分布和累积分布的情况.

Table 3 Property list

表 3 性质列表

序号	性质
$P1$	Pr[<=15] (<> HumanDrive.LET_PASS)
$P2$	Pr[<=30](<>AutoDrive.CHANGE_LANE)

Fig.9 The verification result of $P1$ for normal driver

图 9 当学习出的结果是 normal driver 时性质 $P1$ 的验证结果

© 中国科学院软件研究所 http://www.jos.org.cn

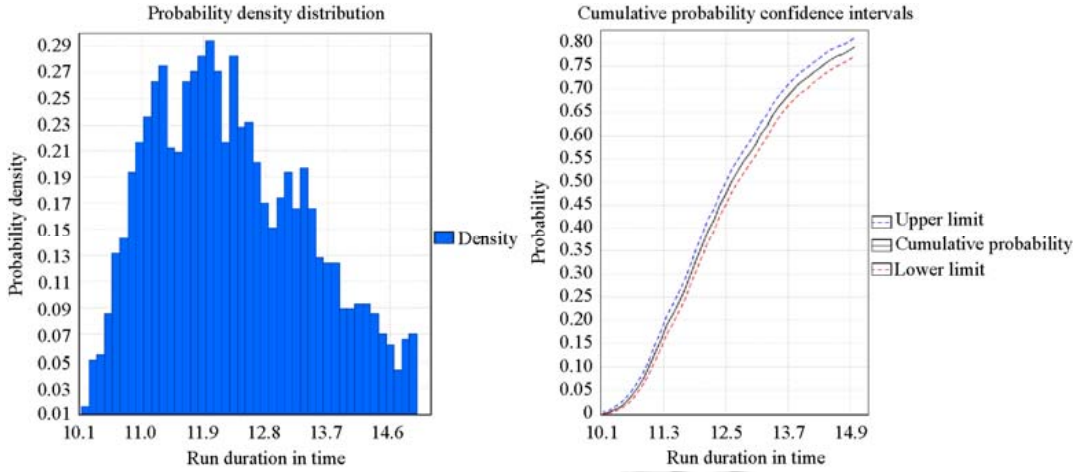

Fig.10 The probability distribution and cumulative distribution of $P1$ for normal driver

图 10 当学习出的结果是 normal driver 时,$P1$ 性质的概率密度分布和累积分布情况

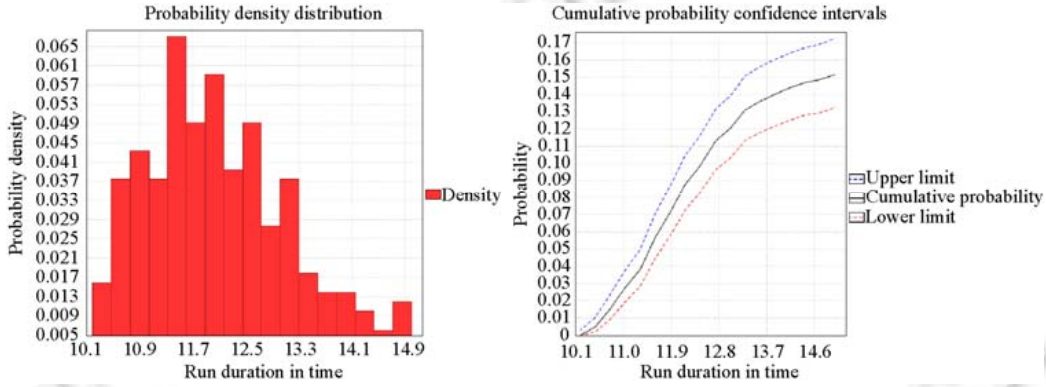

Fig.11 The probability distribution and cumulative distribution of $P1$ for aggressive driver

图 11 当学习出的结果是 aggressive driver 时,$P1$ 性质的概率密度分布和累积分布情况

(2) $P2$ 表示在 30 个时间单位内无人驾驶车辆完成变道的概率.图 12 表示,当学习出的结果是 normal driver 时,$P2$ 性质的概率密度分布和累积分布情况.图 13 表示,当人工驾驶风格分类模型预测结果为 aggressive driver 时,$P2$ 性质的概率密度分布和累积分布情况.

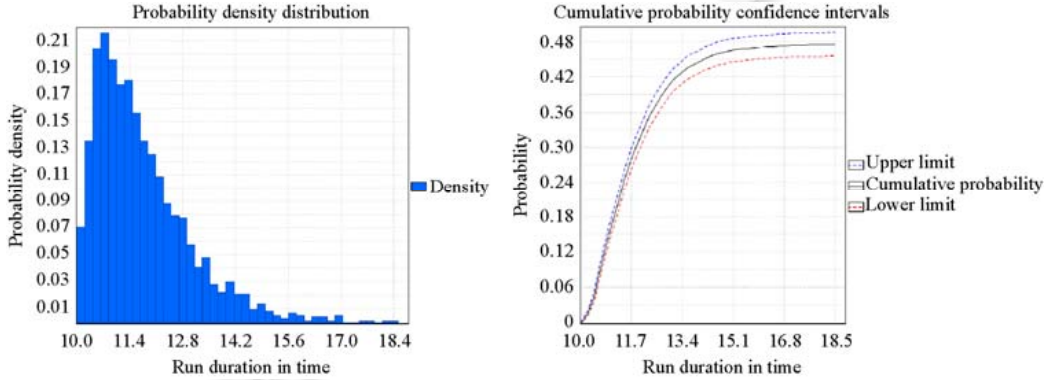

Fig.12 The probability distribution and cumulative distribution of $P2$ for normal driver

图 12 当学习出的结果是 normal driver 时,$P2$ 性质的概率密度分布和累积分布情况

© 中国科学院软件研究所 http://www.jos.org.cn

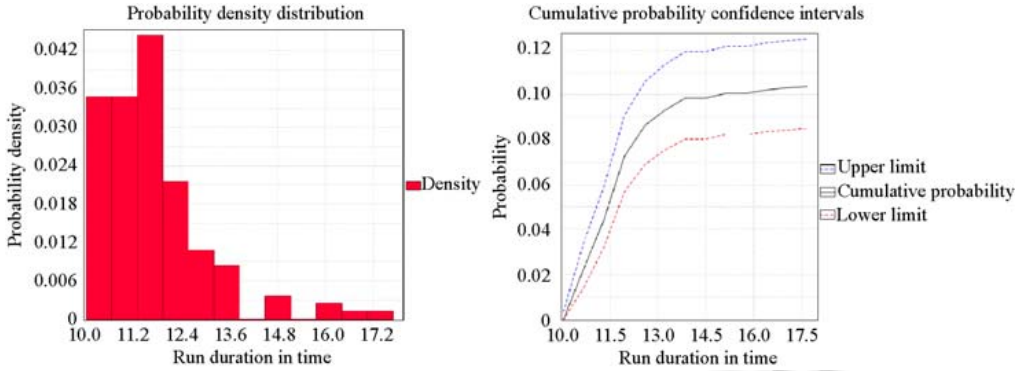

Fig.13 The probability distribution and cumulative distribution of *P*2 for aggressive driver

图 13 当学习出的结果是 aggressive driver 时,*P*2 性质的概率密度分布和累积分布情况

3.3 采用参数比较法进行动态验证

在上一节我们得到 UPPAAL-SMC 可以验证的模型后,通过写入要验证的性质,即可对模型进行验证.上一小节的验证都是线上验证的,通过设置不同的参数,可以对同一个模型进行多次验证,得到不同的验证结果.由于在风险等级较高的环境中,需要保证车辆的安全,但是验证时间在二十几秒之后才能得到,这对于系统来说是难以接受的.若车辆不能及时地对周边环境做出反应,那么对于无人驾驶车辆来说将造成无法承担的后果.对于场景 *A*,Probability uncertaintyϵ 参数为 0.01,我们得到了验证结果.那么,我们将其存入场景库中.在车辆实时运行在场景 *B* 中,如果场景 *B* 的大部分指标与 *A* 相同,个别项比场景 *A* 还要宽松,例如在相同环境下,人流量相对于场景 *A* 较少.那么,我们就不再需要对场景 *B* 进行耗时的线上验证,而只需参考场景比 *B* 更严格的场景 *A* 的验证结果即可.这种参数对比方法,大大节省了线上验证的时间,获得了动态实时的验证效果,进而保障了系统在复杂环境中的安全运行.在得到验证结果后,接下来就是决策阶段,基于不同的驾驶环境,无人驾驶车辆根据验证结果来决定是否改变车道.

(1) 当车辆行驶在高速公路上,车速大于 80km/h 时,车辆换道的概率大于 95%.

(2) 当车辆行驶在城区,车速小于 30km/h 时,车辆换道的概率大于 80%.

例如,上面案例中 *P*1 的结果为 90%,如果所处的场景为高速公路,90%<95%,所以不能完成变道.如果所处的场景为城区,90%>80%,可以进行变道.可见不同场景下指定不同的动作阈值可以保证系统在更安全的情况下进行决策并决定未来状态.

本节以无人驾驶车辆与人工驾驶车辆交互过程中的变道过程为案例,首先基于上一节提出的基于机器学习的环境感知模型得到驾驶风格分类结果,接着基于 SHA 模型对无人驾驶汽车的运动状态进行了建模,并使用统计模型检测工具 UPPAAL-SMC 来验证所建立的 NSHA 模型.

4 相关工作比较

目前,对于驾驶风格识别的研究尚处于起步阶段.人与机器最大的不同在于人是有情绪的,有些驾驶员比较激进,有些比较稳重.2016 年,谷歌公司的自动驾驶汽车在换道时与迎面而来的巴士相撞,原因就是自动驾驶汽车以为巴士会减速,而巴士司机却加速行驶.如果能够事先知道司机的驾驶风格,并结合进行预测,这场事故也许是可以避免的.当然,驾驶风格目前还没有一个准确的定义,因此分类的依据也有很多种,比如油耗、均速、跟车行为等.目前,还没有将驾驶风格识别成功应用到真实的自动驾驶系统的相关报道,但是这些研究可能是未来自动驾驶发展的一个方向.

2011 年 11 月 15 日,首个适用于汽车电子电器相关产品的功能安全标准 ISO 26262 正式发布.功能安全的设计议题在汽车领域开始受到重视.为了更好地量化驾驶场景的不确定性和风险水平以提高自动驾驶系统的安全性,Geng 等人[24]提出了利用贝叶斯方法来量化深度神经网络不确定性的方法.Gadepally 等人[25]设计了一

© 中国科学院软件研究所 http://www.jos.org.cn

个贝叶斯深度学习框架,并在模拟场景中展示了它相对于传统方法的优势.该方法是每个模块在系统中的传递和输入都服从概率分布函数,而不是一个精确的结果.另一种方法是单独评估驾驶场景下的风险水平.可以理解为,前者是从系统内部进行评估,后者是从系统外部进行评估.Yamazaki等人[26]将传感器数据输入到一个风险推理框架中,利用隐马尔可夫模型(hidden Markov model,简称 HMM)和语言模型检测不安全的车道变更事件.Yurtsever[27]引入了一个深度时空网络来推断驾驶场景的总体风险水平,从而评估车道变更的风险水平.牛津大学的Wu等人[28]提出了基于双方回合博弈的(two-player turn-based game)框架来对神经网络加以验证,用于评估安全攸关自动驾驶系统对交通标志识别的准确性.Wicke 等人[29]研究了对抗性输入扰动下的贝叶斯神经网络BNN的概率安全性,基于非凸优化的技术,开发了用于计算概率安全性的算法框架,从而验证具有数千个神经元的概率安全性并应用于自动驾驶系统.Huang 等人[30]提出了可以表达社会信任概念(如能力、性格和依赖性)的算子,将扩展逻辑 PCTL*变为概率理性时间逻辑 PRTL*,建立了一个基于人的信任度的模型验证框架,可用于自动驾驶系统以根据行人的反应进行决策.Sun 等人[31]通过将覆盖率指导的神经网络测试工具 DeepConcolic 与车辆跟踪系统集成,对深度神经网络(deep neural network,简称 DNN)进行了验证.伯克利大学的 Sanjit 等人[32]提出了一种通过深度强化学习来改善布尔逻辑回溯搜索算法的方法,这是一种对使用ML进行感知的系统进行形式分析的方法.在基于深度神经网络的感知组件上显示了该技术的有效性.实际环境中的自动驾驶决策还有周围驾驶员的意图与行为相关.目前,该技术在自动驾驶领域尚不常见.Geng 等人[33]用隐马尔可夫模型(HMM)对目标车辆的未来行为进行了预测,通过学习人类驾驶特征,将预测时间范围延长了56%.这里,主要是利用了预定义的移动行为来标记观测值,然后再使用 HMM 以数据为中心学习每种类型的特征.除此之外,还有一些其他方法,比如贝叶斯网络分类器、混合高斯模型和隐马尔可夫模型相结合[34]、支持向量机等.这一类评估的主要问题在于观测时间短,实时计算量要求高,大多数情况下,自动驾驶系统只能观测周围车辆几秒钟,因此不能使用在需要较长观察周期的复杂模型中.

针对不确定性的模型检测技术,关于随机模型检验的研究开始于 20 世纪 80 年代初,Harts 等人[35]使用离散时间马尔可夫过程建模概率程序,研究概率并发程序的终止性质和概率程序性质的证明方法.Vardi 等人[36,37]提出基于自动机理论的定性线性时间性质的验证方法,Courcoubetis 和 Yannakakis[38]研究了线性时间框架下的定性、定量验证理算时间马尔可夫链.英国牛津大学的 Kwiatkowska 团队开发出图形化随机模型检验工具PRISM[39],该工具可以验证包括 DTMC、CTMC、MDP、PA、PTA 及其 reward 扩展模型等多种类型的随机系统模型,其性质规约包括 PCTL、CSL、概率 LTL、PCTL*及其 reward 扩展等定量性质规约语言.PRISM 基于BDD(binary decision diagram)和 MTBDD(multi-terminal binary decision diagram)的符号数据结构和算法[40],通过离散事件模拟引擎来实现定量抽象精化和系统归约等验证技术,支持多目标定量性质的随机模型检验和统计随机模型检验[41].德国亚琛工业大学的 Hartmanns 等人提出了支持验证随机混成自动机、随机时间自动机的工具集 MODEST[42].通过为现有语言提供导入和导出功能,允许重复使用现有模型;并通过将它们集成在统一的建模和分析环境中而允许重复使用现有工具.MRMC(Markov reward model checker)[44]采用基于系数矩阵的数据结构和算法[43],支持精确的 on-the-fly 稳态检查和互模拟最小化等验证技术,其优势在于验证规模较小的连续时间随机系统的稳态性质效率较高.Katoen 等人近年来开发的工具 Storm[45]支持对马尔可夫链和马尔可夫决策过程的离散和连续时间变体进行分析,支持包括 JANI 和 PRISM 建模语言、动态故障树、广义随机 Petri 网和概率保护的命令语言.模块化设置可以轻松交换求解器和符号引擎,通过封装 Python API 使得 Storm 工具可以使用扩展的算法来实现快速原型制作.丹麦乌普萨拉大学的研究人员 Larsen 和 David 等人基于随机混成自动机 SHA,开发了统计模型验证工具 UPPAAL-SMC[46],统计模型检测是一种高效的验证技术,常用于复杂的随机系统验证.该工具所采用的技术主要是参数估计式和假设检验.目前使用上述工具所进行建模的系统需要设计人员在建模初期就确定好参数,以便验证.但在不确定性环境中,参数是不断变化的,采用这种验证方法无法满足时效性的要求.目前结合机器学习的统计验证模型框架的相关工作比较少,还缺乏统一的理论框架.

© 中国科学院软件研究所 http://www.jos.org.cn

5 总结与展望

为了提高无人驾驶系统面对不确定环境的“智能性”,本文提出了时空数据驱动的基于朴素贝叶斯分类器的周边人工驾驶行为分类模型 DSC.基于朴素贝叶斯的分类算法对周边人工驾驶车辆的驾驶行为进行分类并将学习器的学习结果作为参数输入到后续的系统模型中.为了提高无人驾驶系统面对不确定环境的“安全性”,我们从模型驱动的角度出发,构建结合了人工驾驶行为分类模型的计算结果的 NSHA 模型.通过将 NSHA 模型和需要验证的性质共同输入到统计模型验证工具 UPPAAL-SMC 以对系统模型进行验证.为了提高验证模型的效率,我们从机器学习流程中线下训练和线上学习的过程中得到启发,采用线下验证和线上验证相结合的方法,通过比较参数可快速得出验证结果,实现动态验证,从而帮助 hCPS 系统面对复杂多变的不确定性环境可以更及时地进行决策.综上,我们对机器学习和形式化方法的交叉领域展开了探索研究.整个过程展现了从机器学习、模型构建到统计模型验证的全过程.

在接下来的工作中,我们需要从以下几点进行拓展:由于条件限制,目前大多数研究关注于无人驾驶车辆本身的数据采集,对周边车辆的监控数据通过识别或者对比来判断危险情况,还存在一定的风险.目前无人驾驶车辆还没有“智能”到可以预判周边人类驾驶员的行为.近年来已有学者在对相关问题展开研究,例如卡内基梅隆大学相关学者对周边车辆的相关数据进行了采集,但还未开放.在今后的工作中,我们需要模拟不同情况下的危险驾驶行为的相关数据,从而帮助建立更完善的模型.目前我们使用的是统计模型检测的工具 UPPAAL-SMC.暂时没有与其他高效率的验证工具进行对比,例如概率模型检测工具 PRISM.在接下来的工作中,我们考虑使用不同的工具进行验证.目前的工作是基于监督学习的,由于强化学习在交互方面表现良好,未来我们考虑建立基于强化学习的人工驾驶行为分类模型,从而帮助无人驾驶车辆更智能地应对复杂的环境.

References:

[1] Lee EA. Cyber physical systems: Design challenges. In: Proc. of the 11th IEEE Int’l Symp. on Object and Component Oriented Real Time Distributed Computing (ISORC). 2008. 363–369. [doi: 10.1109/ISORC.2008.25]

[2] Robinson RM, Scobee DRR, Burden SA, *et al*. Dynamic inverse models in human cyber physical systems. In: Proc. of the SPIE. 2016. 9836. [doi: 10.1117/12.2223176]

[3] Leveson NG. Software safety: Why, what, and how. ACM Computing Surveys, 1986,18(2):125–163. [doi: 10.1145/7474.7528]

[4] Sun HY. Safety software testing based on multiform clocks input output transition system [Ph.D. Thesis]. Shanghai: East China Normal University, 2017 (in Chinese with English abstract).

[5] Schirner G, Erdogmus D, Chowdhury KR, *et al*. The future of human in the loop cyber physical systems. IEEE Computer, 2013, 46(1):3645. [doi: 10.1109/MC.2013.31]

[6] Geiger A, Lenz P, Urtasun R. Are we ready for autonomous driving? The Kitti vision benchmark suite. In: Proc. of the IEEE Conf. on Computer Vision and Pattern Recoginition (CVPR). 2012. 3354–3361.

[7] Elsayed GF, Shankar S, Cheung B, *et al*. Adversarial examples that fool both computer vision and time limited humans. In: Proc. of the Annual Conf. on Neural Information Processing Systems, NeurIPS. 2018. 3914–3924.

[8] Finlayson SG, Bowers JD, Ito J, Zittrain JL, Beam AL, Kohane IS. Adversarial attacks on medical machine learning. Science, 2019, 363(6433):1287–1289. [doi: 10.1126/science.aaw4399]

[9] Banks VA, Plant KL, Stanton NA. Driver error or designer error: Using the perceptual cycle model to explore the circumstances surrounding the Fatal Tesla crash on 7th May 2016. Safety Science, 2017,108:278–285. [doi: 10.1016/j.ssci.2017.12.023]

[10] Kohli P, Chadha A. Enabling pedestrian safety using computer vision techniques: A case study of the 2018 Uber Inc. selfdriving car crash. In: Proc. of the Future of Information and Communication Conf. 2019. 261–279. [doi: 10.1007/978-3-030-12388-8_19]

[11] Woodcock J, Larsen PG, Bicarregui J, *et al*. Formal methods: Practice and experience. ACM Computing Surveys, 2009,41(4):19. [doi: 10.1145/1592434.1592436]

[12] Abrial J. Formal methods: Theory becoming practice. Journal of Universal Computer Science, 2007,13:619–628.

[13] Bicarregui JC, Fitzgerald JS, Larsen PG, *et al*. Industrial practice in formal methods: A review. In: Proc. of the Int’l Symp. on Formal Methods. Springer-Verlag, 2009. 810–813. [doi: 10.1007/978-3-642-05089-3_52]

© 中国科学院软件研究所 http://www.jos.org.cn

[14] Clarke Jr EM, Grumberg O, Kroening D, *et al*. Model Checking. MIT Press, 2018.

[15] Legay A, Delahaye B, Bensalem S. Statistical model checking: An overview. In: Proc. of the Int'l Conf. on Runtime Verification. Berlin, Heidelberg: Springer-Verlag, 2010. 122–135. [doi: 10.1007/978-3-642-16612-9_11]

[16] Du DH, Cheng B, Liu J. Statistical model checking for rare-event in safety-critical system. Ruan Jian Xue Bao/Journal of Software, 2015,26(2):305−320 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/4783.htm [doi: 10.13328/j.cnki.jos.004783]

[17] Zuliani P, Platzer A, Clarke EM. Bayesian statistical model checking with application to stateflow/simulink verification. Formal Methods in System Design, 2013,43(2):338–367. [doi: 10.1145/1755952.1755987]

[18] Du DH, Zan H, Jiang KQ, Cheng B. Self-adaptive statistical model checking approach for CPS. Ruan Jian Xue Bao/Journal of Software, 2017,28(5):1128−1143 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/5216.htm [doi: 10.13328/j.cnki.jos.005216]

[19] Davarzani S, Nagahi M, Tidwell M, *et al*. Pattern recognition using machine learning for corn and Soybean yield prediction. In: Proc. of the 2020 IISE. New Orleans, 2020.

[20] Julius AA, Pappas GJ. Approximations of stochastic hybrid systems. IEEE Trans. on Automatic Control, 2009,54(6):1193–1203. [doi: 10.1109/TAC.2009.2019791]

[21] David A, Larsen KG, Legay A, *et al*. Statistical model checking of dynamic networks of stochastic hybrid automata. Francisco Javier Fuente Fernández, 2014,66:91104. [doi: 10.14279/tuj.eceasst.66.893.878]

[22] Llerena YRS, Su G, Rosenblum DS. Probabilistic model checking of perturbed MDPs with applications to cloud computing. In: Proc. of the 11th Joint Meeting on Foundations of Software Engineering. ACM, 2017. 454–464. [doi: 10.1145/3106237.3106301]

[23] Zhao X, Robu V, Flynn D, *et al*. Probabilistic model checking of robots deployed in extreme environments. In: Proc. of the AAAI Conf. on Artificial Intelligence. 2019(33):8066–8074. [doi: 10.1609/aaai.v33i01.33018066]

[24] Gal Y. Uncertainty in deep learning [Ph.D. Thesis]. University of Cambridge, 2016.

[25] McAllister R, Gal Y, Kendall A, *et al*. Concrete problems for autonomous vehicle safety: Advantages of Bayesian deep learning. In: Proc. of the Int'l Joint Conf. on Artificial Intelligence. 2017. [doi: 10.24963/ijcai.2017/661]

[26] Yamazaki S, Miyajima C, Yurtsever E, *et al*. Integrating driving behavior and traffic context through signal symbolization. In: Proc. of the IEEE Intelligent Vehicles Symp. (IV). IEEE, 2016. 642–647. [doi: 10.1109/IVS.2016.7535455]

[27] Yurtsever E, Liu Y, Lambert J, *et al*. Risky action recognition in lane change video clips using deep spatio temporal networks with segmentation mask transfer. In: Proc. of the IEEE Intelligent Transportation Systems Conf. (ITSC). IEEE, 2019. 3100–3107. [doi: 10.1109/ITSC.2019.8917362]

[28] Wu M, Wicker M, Ruan W, *et al*. A gamebased approximate verification of deep neural networks with provable guarantees. Theory Computer Science, 2020,807:298–329. [doi: 10.1016/j.tcs.2019.05.046]

[29] Wicker M, Laurenti L, Patane A, *et al*. Probabilistic safety for bayesian neural networks. CoRR, 2020, abs/2004.10281.

[30] Huang X, Kwiatkowska M, Olejnik M. Reasoning about cognitive trust in stochastic multiagent systems. ACM Trans. on Compututation Logic, 2019,20(4):21:121:64. [doi: 10.1145/3329123]

[31] Sun Y, Zhou Y, Maskell S, *et al*. Reliability validation of learning enabled vehicle tracking. In: Proc. of the IEEE Int'l Conf. on Robotics and Automation (ICRA). 2020. 9390–9396. [doi: 10.1109/ICRA40945.2020.9196932]

[32] Lederman G, Rabe MN, Seshia S, *et al*. Learning heuristics for quantified boolean formulas through reinforcement learning. In: Proc. of the Int'l Conf. on Learning Representations, ICLR. 2020.

[33] Dreossi T, Donzé A, Seshia SA. Compositional falsification of cyber physical systems with machine learning components. Journal of Automated Reasoning, 2019,63(4):10311053. [doi: 10.1007/s10817-018-09509-5]

[34] Geng X, Liang H, Yu B, *et al*. A scenario adaptive driving behavior prediction approach to urban autonomous driving. Applied Sciences, 2017,7(4):426. [doi: 10.3390/app7040426]

[35] Augustynowicz A. Preliminary classification of driving style with objective rank method. Int'l Journal of Automotive Technology, 2009,10(5):607–610. [doi: 10.1007/s12239-009-0071-8]

[36] Hart S, Sharir M, Pnueli A. Termination of probabilistic concurrent programs: (extended abstract). In: Proc. of the Symp. on Principles of Programming Languages. 1982. 16. [doi: 10.1145/582153.582154]

© 中国科学院软件研究所 http://www.jos.org.cn

[37] Vardi MY. Automatic verification of probabilistic concurrent finite state programs. Foundations of Computer Science, 1985, 327–338. [doi: 10.1109/SFCS.1985.12]

[38] Vardi MY, Wolper P. An automata theoretic approach to automatic program verification. In: Proc. of the 1st Symp. on Logic in Computer Science. IEEE Computer Society, 1986. 322–331.

[39] Courcoubetis C, Yannakakis M. The complexity of probabilistic verification. Journal of the ACM, 1995,42(4):857–907. [doi: 10.1145/210332.210339]

[40] Kwiatkowska M, Norman G, Parker D. PRISM 4.0: Verification of probabilistic real time systems. In: Proc. of the Int'l Conf. on Computer Aided Verification (CAV). 2011. 585–591. [doi: 10.1007/978-3-642-22110-1_47]

[41] Kwiatkowska M, Norman G, Parker D. Probabilistic symbolic model checking with prism: A hybrid approach. Int'l Journal on Software Tools for Technology Transfer, 2001. [doi: 10.1007/s10009-004-0140-2]

[42] Kwiatkowska M. Quantitative verification: Models techniques and tools. In: Proc. of the 6th Joint Meeting of the European Software Engineering Conf. and the ACM SIGSOFT Symp. on the Foundations of Software Engineering. 2007. 449–458. [doi: 10.1145/1295014.1295018]

[43] Hartmanns A, Hermanns H. The modest toolset: An integrated environment for quantitative modelling and verification. In: Proc. of the Int'l Conf. on Toolsand Algorithms for the Construction and Analysis of Systems. Springer-Verlag, 2014. 593–598. [doi: 10.1007/978-3-642-54862-8_51]

[44] Katoen JP, Zapreev IS, Hahn EM, *et al*. The ins and outs of the probabilistic model checker MRMC. Performance Evaluation, 2011, 68(2):90–104. [doi: 10.1016/j.peva.2010.04.001]

[45] Hensel C, Junges S, Katoen J, *et al*. The probabilistic model checker storm. CoRR, 2020, abs/2002.07080.

[46] Dehnert C, Junges S, Katoen JP, Volk M. A storm is coming: A modern probabilistic model checker. In: Proc. of the Int'l Conf. on Computer Aided Verification (CAV). 2017. [doi: 10.1007/978-3-319-63390-9_31]

附中文参考文献:

[4] 孙海英.基于多形态时钟输入输出迁移系统的安全软件测试研究[博士学位论文].上海:华东师范大学,2017.

[16] 杜德慧,程贝,刘静.面向安全攸关系统中小概率事件的统计模型检测.软件学报,2015,26(2):305−320. http://www.jos.org.cn/1000-9825/4783.htm [doi: 10.13328/j.cnki.jos.004783]

[18] 杜德慧,咎慧,姜凯强,程贝.一种面向 CPS 的自适应统计模型检测方法.软件学报,2017,28(5):1128–1143. http://www.jos.org.cn/1000-9825/5216.htm [doi: 10.13328/j.cnki.jos.005216]

安冬冬(1990－),女,博士,CCF 专业会员,主要研究领域为形式化建模与验证,统计模型检测,人机物融合系统,自动驾驶系统.

刘静(1964－),女,博士,教授,博士生导师,CCF 专业会员,主要研究领域为基于模型的高可信软件开发技术,形式化方法建模与验证.

陈小红(1982－),女,博士,副教授,CCF 专业会员,主要研究领域为需求工程,形式化方法,安全攸关系统.

孙海英(1976－),女,博士,讲师,CCF 专业会员,主要研究领域为形式化建模,形式化验证,基于形式化的测试.

© 中国科学院软件研究所 http://www.jos.org.cn