

轨迹异常检测研究综述*

李超能^{1,2}, 冯冠文^{1,2}, 姚航^{1,2}, 刘如意^{1,2}, 李宇楠^{1,2}, 谢琨^{1,2}, 苗启广^{1,2}

¹(西安电子科技大学 计算机科学与技术学院, 陕西 西安 710071)

²(西安市大数据与视觉智能关键技术重点实验室 (西安电子科技大学), 陕西 西安 710071)

通信作者: 苗启广, E-mail: qgmiao@xidian.edu.cn



摘要: 传感器技术的飞速发展催生大量交通轨迹数据, 轨迹异常检测在智慧交通、自动驾驶、视频监控等领域具有重要的应用价值. 不同于分类、聚类和预测等轨迹挖掘任务, 轨迹异常检测旨在发现小概率、不确定和罕见的轨迹行为. 轨迹异常检测中一些常见的挑战与异常值类型、轨迹数据标签、检测准确率以及计算复杂度有关. 针对上述问题, 全面综述近 20 年来轨迹异常检测技术的研究现状和最新进展. 首先, 对轨迹异常检测问题的特点与目前存在的研究挑战进行剖析. 然后, 基于轨迹标签的可用性、异常检测算法原理、离线或在线算法工作方式等分类标准, 对现有轨迹异常检测算法进行对比分析. 对于每一类异常检测技术, 从算法原理、代表性方法、复杂度分析以及算法优缺点等方面进行详细总结与剖析. 接着, 讨论开源的轨迹数据集、常用的异常检测评估方法以及异常检测工具. 在此基础上, 给出轨迹异常检测系统架构, 形成从轨迹数据采集到异常检测应用等一系列相对完备的轨迹挖掘流程. 最后, 总结轨迹异常检测领域关键的开放性问题, 并展望未来的研究趋势和解决思路.

关键词: 轨迹数据; 异常检测; 数据挖掘; 机器学习; 深度学习

中图法分类号: TP311

中文引用格式: 李超能, 冯冠文, 姚航, 刘如意, 李宇楠, 谢琨, 苗启广. 轨迹异常检测研究综述. 软件学报, 2024, 35(2): 927-974. <http://www.jos.org.cn/1000-9825/6996.htm>

英文引用格式: Li CN, Feng GW, Yao H, Liu RY, Li YN, Xie K, Miao QG. Survey on Trajectory Anomaly Detection. Ruan Jian Xue Bao/Journal of Software, 2024, 35(2): 927-974 (in Chinese). <http://www.jos.org.cn/1000-9825/6996.htm>

Survey on Trajectory Anomaly Detection

LI Chao-Neng^{1,2}, FENG Guan-Wen^{1,2}, YAO Hang^{1,2}, LIU Ru-Yi^{1,2}, LI Yu-Nan^{1,2}, XIE Kun^{1,2}, MIAO Qi-Guang^{1,2}

¹(School of Computer Science and Technology, Xidian University, Xi'an 710071, China)

²(Xi'an Key Laboratory of Big Data and Intelligent Vision (Xidian University), Xi'an 710071, China)

Abstract: The rapid advancement of sensor technology has resulted in a vast volume of traffic trajectory data, and trajectory anomaly detection has a wide range of applications in sectors including smart transportation, autonomous driving, and video surveillance. Trajectory anomaly detection, unlike other trajectory mining tasks like classification, clustering, and prediction, tries to find low-probability, uncertain, and unusual trajectory behavior. The types of anomalies, trajectory data labels, detection accuracy, and computational complexity are all frequent issues in trajectory anomaly detection. In view of the above problems, the research status and latest progress of trajectory anomaly detection technology in the past two decades are comprehensively reviewed. First, the characteristics of trajectory anomaly detection and the current research challenges are analyzed. Then, the existing trajectory anomaly detection algorithms are compared and analyzed based on the classification criteria such as the availability of trajectory labels, the principle of anomaly detection algorithms, and the working mode of offline or online algorithms. For each type of anomaly detection technology, the algorithm principle, representative

* 基金项目: 科技创新 2030—“新一代人工智能”重大项目 (2022ZD0117103); 国家自然科学基金 (62272364, 62002271); 陕西省重点研发计划 (2020LSP3-15); 中国成人教育协会“十四五”成人继续教育科研规划重点课题 (2021-414ZA); 陕西高等继续教育教学改革研究课题 (21XJZ004); 广西可信软件重点实验室研究课题 (KX202061); 青岛市科技计划重点研发专项 (21-1-2-18-xx)
收稿时间: 2022-05-23; 修改时间: 2023-02-28, 2023-06-14; 采用时间: 2023-06-30; jos 在线出版时间: 2023-11-08
CNKI 网络首发时间: 2023-11-10

method, complexity analysis and algorithm advantages and disadvantages are summarized and analyzed in detail. Then, the open source trajectory datasets, commonly used anomaly detection evaluation methods and anomaly detection tools are discussed. On this basis, the architecture of the trajectory anomaly detection system is presented, and a series of relatively complete trajectory mining processes from trajectory data collection to anomaly detection application are formed. Finally, the significant open issues in the domain of trajectory anomaly detection are discussed, as well as potential research trends and solutions.

Key words: trajectory data; anomaly detection; data mining; machine learning; deep learning

随着全球定位系统 (global positioning system, GPS)、WiFi、高分辨率相机、视频监控以及无线传感器网络等技术的不断发展,各种移动物体的轨迹数据迅速增长,包括行人活动轨迹、交通车辆轨迹、海域船舶轨迹和飞行器轨迹等,这些轨迹为我们提供了前所未有的信息来理解移动物体及其行为。同时,人们对基于位置的服务 (location-based service, LBS) 需求不断增加,促进了智能交通系统、复杂社交网络和自动视频监控的广泛应用。所以,如何从大规模轨迹数据中发掘有用的知识已成为重要的研究课题,并受到计算机科学、社会学、生物学、地理学等众多领域的关注。

轨迹挖掘的主要任务包括轨迹分类^[1,2]、轨迹聚类^[3,4]、轨迹预测^[5,6]、轨迹表示^[7]、轨迹生成^[8-10]、轨迹模式挖掘^[11,12]、轨迹异常检测^[13-15]等。早期的轨迹数据是以时间和位置的形式从摄像机记录的监控视频中提取而来。近年来,得益于定位设备和位置跟踪设备的普及,各种大小移动物体的轨迹信息都可以被实时跟踪。例如,交通管理部门通过车载 GPS 设备监控城市交通和引导路线^[16],动物学家在动物身上安装遥感设备来研究它们的生态行为^[17],气象学家利用雷达和卫星来分析测量台风登陆情况^[18]。采集到的海量轨迹数据需要有效的算法进行处理,以促进轨迹数据挖掘的研究和相关应用。然而,由于探测器设备故障、传输链路失真、紧急事故发生或其他可能的影响因素等各种原因,跟踪采集到的轨迹数据不可避免地会受到破坏,并且往往包含不符合数据模型一般分布的数据项,称为离群值或异常值。

Chandola 等人^[19]对异常的定义为:异常是数据中不符合正常行为定义的模式。轨迹异常值是指与轨迹数据中频繁出现的轨迹模式不同,或是不符合预期模式的事件和观测值。从数据挖掘的角度来看,异常检测是将异常观测值从正常观测值中识别分离出来。在轨迹挖掘的背景下,这意味着发现与数据集中其余轨迹在某些相似性度量上存在局部或全局不一致的轨迹。

异常检测在现实中有许多跨领域的应用,例如,网络入侵检测^[20,21]、金融和电信欺诈检测^[22,23]、工业故障检测^[24,25]、医学诊断^[26,27]、生物信息学^[28,29]等。这些领域包括复杂的数据类型,如图像、视频、文本、时间序列、生物序列等,这对异常检测算法在复杂高维数据上的检测效果提出了较高的要求。同样地,轨迹异常检测近年来备受关注,在研究和工业应用中也具有重要的意义和实用价值。图 1 为 2010-2022 年轨迹异常检测研究发表论文数。从图中我们可以看出,在过去的 10 年里,这一领域的研究兴趣越来越浓厚。在实践中,轨迹异常检测可用于分析视频监控轨迹^[30]、检测智能交通异常^[31,32]、预警人群聚集事件^[33,34]、识别出租车驾驶欺诈行为^[35,36]等。另外,轨迹异常检测在自动驾驶、航空海事、边境安全等领域也有广泛的应用,下面我们只考虑几个具有代表性的应用领域进行简单介绍。

(1) 轨迹数据清洗

有效的数据挖掘必须建立在高质量数据基础之上,轨迹数据在采集传输过程中,由于受到大气层、GPS 设备自身弊端或其他信号遮挡干扰等因素的影响,获取的部分轨迹数据可能包含“脏数据”,通过结合其他数据清洗方法和轨迹异常检测过程可以提高数据质量。轨迹数据的清洗和校正方法是训练高质量模型和轨迹信息挖掘的关键^[11,14]。

(2) 交通视频监控

在智慧交通监控领域,车辆的异常行为,如突然加速、偏离道路、位置漂移等,可能暗示存在与酒后驾驶、交通碰撞、道路违章有关的危险^[30,35,36],及时查明和评估感兴趣区域内的异常轨迹对于主动采取解决措施至关重要。若多条异常轨迹出现在同一条道路中,还可以为城市路网的建设及规划提供参考。

(3) 自动驾驶导航

自动驾驶汽车结合各种传感器来感知周围环境并进行导航规划^[16,31,32]。提前检测感知周围交通个体的异常行

为轨迹, 可以帮助无人车更加快捷地选择合理的行驶路径, 避开危险路段, 从而提升自动驾驶系统的安全性和可靠性, 降低道路事故率。

(4) 动物习性分析

动物习性分析^[17,28]是轨迹异常挖掘的一个非常重要的应用, 动物的异常习惯或移动倾向可能存在不符合群体常规运动模式的事件, 这些异常的发现可以极大地促进生物学家们的研究兴趣。

(5) 气象异常检测

气象异常检测一般是针对飓风、台风和旋风等自然灾害的发生进行提前预警^[18]。例如, 通过对飓风轨迹进行监测和记录, 及时发现飓风运动过程中的异常现象, 可以为飓风的灾情预报提供辅助。

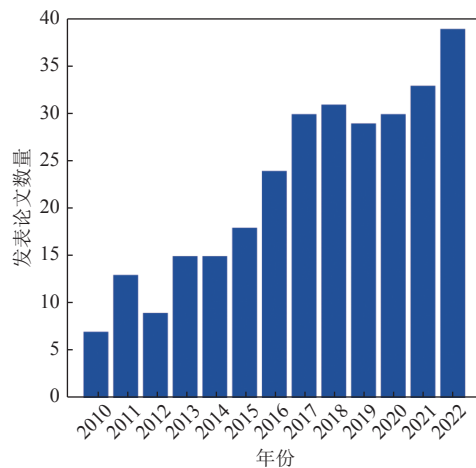


图1 2010–2022年轨迹异常检测研究发表论文数

本文第1节介绍轨迹异常检测领域现有的代表性综述及其主要内容。在第2节中, 对轨迹数据相关的定义进行概述, 并分析异常轨迹产生的原因以及常见的异常类型。第3节主要剖析目前轨迹异常检测研究面临的挑战与待解决问题。第4节对公开文献中已发表的轨迹异常检测算法进行总结和分类, 并详细分析每种算法的基本原理、代表性方法、复杂度、优点和局限性等。第5节对近年来轨迹异常检测的代表性方法进行了综合分类, 总结了现有文献中常用的轨迹数据集, 讨论了异常检测算法评估指标以及用于异常检测的开源工具。第6节给出了轨迹异常检测系统架构, 并对轨迹异常检测未来研究方向和发展趋势进行了分析和展望。

1 轨迹异常检测相关综述

由于轨迹信息量过载, 而且在现实应用场景中, 摄像机位置、采样频率以及场景结构存在差异, 所获得的运动轨迹通常在时间和空间特征上会有所不同, 无法通过人工分析来完成异常检测任务。因此, 需要设计自动的异常检测算法来对轨迹信息进行挖掘与分析。经过几十年的理论发展, 学者们对轨迹异常检测进行了深入研究, 提出了大量分析轨迹信息的新方法, 特别是针对不同的应用领域, 发表了轨迹异常检测研究综述。Chen等人^[37]从算法原理、特点和时间复杂度等方面回顾了3种典型的轨迹异常检测方法, 分别为基于统计的方法、基于距离的方法、基于密度的局部异常检测方法。通过考虑两个不同维度的交通数据集: 旅行时间和交通流量, 使用南京城市交通数据对3种算法进行了比较研究。此外, 一些人工合成的异常轨迹被引入到交通流数据, 以观察不同算法对它们的检测效果。结果表明, 经典的异常检测算法对于检测交通数据集中的异常轨迹点是可行有效的。Zheng^[38]总结了不同的轨迹表示形式, 包括序列、矩阵、图形、张量等, 以及不同的轨迹数据预处理任务, 如噪声过滤、地图匹配和轨迹压缩等。回顾了轨迹、子轨迹上的异常和噪声点检测, 以及基于轨迹的异常事件识别, 包括事故、指挥、抗议、运动、庆祝、灾难和其他事件。Bhowmick等人^[39]介绍了处理城市交通数据中轨迹异常值的相关工作, 根据处理步

骤中使用的方法将现有的轨迹异常检测算法进行了分类,分别为基于距离、基于密度和基于轨迹历史相似度的异常检测方法,并建议使用轨迹异常检测来识别不规则交通事件,如道路事故或车辆故障造成的交通拥堵等。Meng 等人^[40]从 3 个角度分析了现有的轨迹异常检测算法,首先是考虑多属性的算法,即尽可能多地探索轨迹关键属性,例如速度、方向、位置、时间,来表示原始轨迹并与其他轨迹进行比较。其次,用于衡量轨迹之间差异的距离度量方式。第三,改进现有算法降低时间和空间复杂度。Zhang 等人^[41]回顾了城市异常分析的最新研究成果,概述了城市异常的 4 种类型:交通异常、人群异常、环境异常和个体异常。此外,文献[41]总结了从不同设备获得的各种类型的城市数据集,包括轨迹、出行记录、社交媒体、监控摄像头等,对城市异常检测与预测技术中存在的问题进行了全面分析,讨论了研究挑战和待解决问题。Belhadi 等人^[42]研究了轨迹异常检测算法在不同工业领域的应用,包括智能城市交通、视频监控、气候变化领域等。通过 3 种不同的分类法对现有的轨迹异常检测算法进行分类:基于应用的方法(智能交通、视频分析)、基于输出的方法(轨迹、子轨迹)和基于算法的异常检测方法(距离、密度、模式挖掘),并在不同大小规模的轨迹数据集上对 10 种异常检测算法进行了评估。Djenouri 等人^[31]介绍了与轨迹异常检测有关两个问题。第 1 个问题是群体轨迹异常检测,第 2 个问题是个体轨迹异常和群体轨迹异常的偏离点检测,并将轨迹异常检测算法分为两类:只能检测轨迹异常的离线方法和能够识别产生异常的子轨迹的在线方法。

表 1 总结了现有的轨迹异常检测代表性综述文献及其主要内容,可以看出,现有工作都集中在总结基于统计、轨迹间距离、轨迹密度等传统的异常检测方法。随着深度学习技术在异常检测研究的广泛应用,我们需要对新技术新方法进行对比分析,以扩展轨迹异常检测的应用领域。与以往的综述文献相比,本文回顾了近 20 年轨迹异常检测方法的研究进展,并按照不同的分类标准对每种异常检测算法进行了深入全面的分析,包括算法原理、优点和局限性、计算复杂度以及应用场景等,最后指出未来可能的研究方向。通过本文的总结和分析,期望为轨迹异常检测未来的研究提供一些有价值的参考。

表 1 轨迹异常检测相关综述

文献	年份/来源	综述内容简介
Chen 等人 ^[37]	2010/期刊	基于统计、距离、密度的轨迹异常检测方法
Zheng ^[38]	2015/期刊	轨迹预处理;子轨迹异常检测;异常事件识别
Bhowmick 等人 ^[39]	2018/专著	基于距离、密度、轨迹历史相似度的异常检测方法;不规则交通事件检测
Meng 等人 ^[40]	2019/期刊	轨迹多属性方法、距离度量方法、低复杂度方法
Zhang 等人 ^[41]	2020/期刊	交通、人群、环境等城市异常检测
Belhadi 等人 ^[42]	2020/期刊	基于应用、输出、算法的轨迹异常检测方法
Djenouri 等人 ^[31]	2021/期刊	群组轨迹异常检测和轨迹偏离点检测

2 轨迹相关符号与定义

轨迹点记录了移动对象在不同时刻下的空间位置信息,相邻点之间有着密切的上下文关系,这些有序的点组成的集合即为轨迹。方便起见,本文做如下定义。

定义 1 (轨迹点). 轨迹点是由导航应用(如谷歌地图、百度地图、腾讯地图)、拼车平台(如 Didi Taxi、Hello-bike)、配备 GPS 的车辆(如 Apollo、Waymo)采集的 GPS 坐标。时空轨迹点可以表示为 $p = (x, y, t)$, 包括位置信息 (x, y) (即经度纬度或局部坐标)以及时间戳 t 。

定义 2 (轨迹). 轨迹是由时空轨迹点组成的序列,表示为 $T = \langle p_1, p_2, \dots, p_k \rangle$, 其中 $p_i (i \in \{1, 2, \dots, k\})$ 表示轨迹点, k 为轨迹点个数。

定义 3 (轨迹数据集). 轨迹数据集是一系列轨迹的集合,表示为 $T = \{T_1, T_2, \dots, T_n\}$, 其中 n 为轨迹数目。

定义 4 (异常轨迹). 轨迹异常检测的目标是从来自真实采集或虚拟仿真的海量轨迹数据中识别出异常轨迹。根据真实世界的交通规则和行驶规律以及移动对象速度、方向的变化,异常轨迹一般包括:① 复杂多变的交通状

况导致的剧烈变速; ② 轨迹长时间偏离道路或位置漂移; ③ 违反交通规则逆向行驶等. 以车辆轨迹为例, 本文将异常交通流轨迹归纳为以下 8 种类型: 突然加速、长时间停止、频繁变速、左右摆动、逆向行驶、突然停止、偏离道路范围、剧烈转弯, 如图 2 所示, 图中虚线部分表示该段轨迹存在异常.

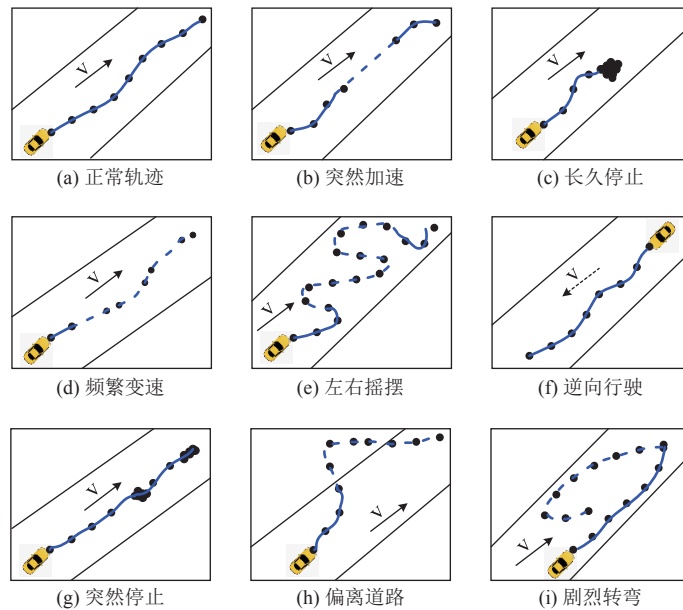


图 2 车辆轨迹异常类型

3 轨迹异常检测研究的问题与挑战

3.1 研究问题的特点

不同于大多数轨迹挖掘问题和任务, 它们都有明确的数据分布模式, 异常检测旨在发现小概率、不确定和罕见的事件, 导致轨迹异常检测问题具有独特的复杂性, 对检测算法的稳定性和鲁棒性提出了较高要求. 简单总结, 轨迹异常检测问题具有如下特点.

3.1.1 异常轨迹类型多样

在轨迹数据采集过程中移动对象自身出现异常行为, 或者受到环境、设备故障等客观因素干扰, 在不同的采集场景下, 可能会产生各种各样的异常轨迹. Chandola 等人^[19]研究了 3 种完全不同类型的异常, 即点异常、条件异常、集合异常. 参照 Chandola 等人^[19]的分类方式, 以交通流轨迹为例, 点异常是指空间上远离大多数轨迹分布的个别轨迹点, 如车辆逆向行驶、偏离历史轨迹、剧烈拐弯、超出道路监测范围. 条件异常, 又称为上下文异常, 是指在特定上下文中才会出现的异常轨迹, 而在其他情况下视为正常行为, 如车辆突然加速、突然停止、强行变道、违规超车. 集合异常是指整体而言将轨迹看作异常类型, 而其中的局部片段可能不是异常, 如轨迹不连续或缺失、轨迹点稀疏. 可以看到, 上述多种异常类型仅针对车辆轨迹, 如果考虑交通流中其他移动对象, 如非机动车、行人等, 产生的异常轨迹数量和类型将会更多, 因此, 异常类型多样性是轨迹异常检测问题的主要特点.

3.1.2 轨迹类别比例失衡

轨迹类别不平衡问题是指轨迹数据集中正常轨迹和异常轨迹数量极不均衡. 异常通常是罕见的数据实例, 而正常实例通常占数据的绝大部分. 样本不平衡在其他应用中也是一种非常常见的现象. 例如, 在欺诈交易检测中, 欺诈交易的订单一般占总交易数量极少部分; 工厂中产品质量检测问题, 合格产品的数量远大于不合格产品数; 信用卡的征信问题中往往也是正常样本居多. 由于异常事件的缺乏, 人工很难收集异常数据实例, 这导致在大多数应用中无法使用大规模的标记轨迹数据直接进行异常检测.

3.1.3 异常轨迹不可预测

异常轨迹受到多种外部因素的影响,其产生一般都是未知和随机的,因此很难对异常轨迹提前做出有效的预测,理想情况是借助移动对象历史轨迹信息进行参考判断.另外,异常轨迹是不规则的,因此,一类异常轨迹可能表现出与另一种异常轨迹完全不同的特征.例如,恐怖袭击事件中嫌疑对象在预谋、实施暴恐、藏匿、逃跑等不同阶段的行为轨迹多具有异常性,一般不同阶段的轨迹特征也不相同,而嫌疑对象的轨迹又往往隐藏在众多繁杂的轨迹之中,这导致安防部门无法预测嫌疑人行踪轨迹,提前采取安全措施.

3.1.4 异常轨迹缺乏定义

轨迹异常检测的关键是有效地发现和表示轨迹的正常路径,作为时空序列数据的代表,轨迹数据随着时间推移在不断变化,所以,在不同的地理位置,轨迹可能分别表现为正常和异常,即正常和异常轨迹行为边界通常没有精确的定义.例如,轨迹序列中的子轨迹可能是异常的,或者整个轨迹序列相对于一组正常的轨迹序列集合可能是异常的.

3.2 研究挑战

第 3.1 节所述复杂的问题性质给传统的轨迹异常检测方法和目前广泛使用的深度学习方法带来了许多检测挑战.一些常见的困难与输入数据的性质、异常值类型、轨迹数据标签、检测准确率以及计算复杂度有关.近年来有些挑战性问题的解决已经得到很好的解决,但大部分尚未彻底找到有效的解决途径,总结起来,目前轨迹异常检测任务存在的关键挑战如下.

3.2.1 缺乏标签信息的轨迹模式学习

由于收集大规模带标记异常轨迹数据的难度和成本,有监督的异常检测算法往往难以应用,因为它假设标定正常和异常类别标签的训练轨迹数据具有可用性.在过去的 10 年中,主要的研究工作集中在不需要任何标记训练数据的无监督异常检测.然而,无监督方法没有任何真实异常的先验知识,它们在很大程度上依赖于对异常分布的假设.通常来说,可以收集或手动标记正常轨迹数据和少量的异常数据^[14,35,43,44],或者借助仿真算法生成部分异常轨迹并注入数据集^[13,45,46],利用这些有限的标记数据来学习正常和异常轨迹的特征对于准确的异常检测至关重要.其他的研究方法如半监督和弱监督轨迹异常检测,前者假设存在一组有标注的轨迹训练数据,后者假设有一些异常类的标签,但类别标签是不完整或不准确的.所以,目前面临的主要挑战是如何使用少量标记轨迹数据学习正常和异常轨迹特征,以及如何学习泛化性较高的模型,由给定的标记数据发现新的异常轨迹.

3.2.2 轨迹异常检测召回率低

由于异常轨迹非常罕见且不均匀,所以很难检测到所有异常.许多正常轨迹被错误地分类为异常,真实而复杂的异常却被遗漏了.在轨迹异常检测评估中,通常召回率的权重分配会远高于准确率,这是由实际应用的目标决定的.例如,在车辆轨迹异常检测中,漏报付出的代价比误报高很多,因为一旦漏报,异常轨迹就难以被及时发现,可能导致潜在的交通事故,而误报的影响则可以通过进一步的检查进行消除.尽管近年来出现了很多异常检测方法,但当前检测效果最好的方法,特别是基于无监督的方法,在真实数据集上仍然产生较高的误报率^[47,48].如何减少误报和提高检测召回率是当前轨迹异常检测中最困难的挑战之一.

3.2.3 抗噪声轨迹异常检测

在轨迹序列数据中,异常检测算法的性能很容易受到噪声的影响.然而,在现实场景采集的轨迹数据中噪声是普遍存在的,人们可能无法获得算法所要求的“干净数据”用以训练.许多半监督、弱监督轨迹异常检测方法假设给定的标记数据是干净的,但这样的假设很可能造成异常轨迹和噪声数据标签混合,降低模型学习轨迹特征的能力.有研究将原始数据分为正常数据和噪声、异常数据两部分,然后交替训练以提高模型的抗噪声能力^[49].但由于噪声数量不定且不规则地分布在数据空间,在异常检测过程中仍然对算法造成一定的干扰.所以,将异常轨迹跟噪声数据区分开来也是一项具有挑战性的任务.

3.2.4 在线轨迹异常检测

现有的轨迹异常检测方法大都是为离线学习而设计的,因为从模型训练开始就假定所有的训练轨迹数据都是

可用的, 基于批量训练数据, 对固定的模型参数进行调整, 然后重复用于异常检测. 相反, 设计用于在线学习的异常检测算法会随着每次新的训练轨迹的观察而增量地更新模型参数. 为了在异常发生时迅速采取对策, 需要支持对异常轨迹的有效在线检测. 然而, 因为轨迹往往以非常高的速度产生, 而且数据规模巨大, 以北京市为例, 在一周时间里出租车生成的 GPS 轨迹点个数可达 1 500 万^[50], 所以如何开发低延时响应的在线轨迹异常检测算法对研究人员来说也是极大的挑战.

3.2.5 可解释的轨迹异常检测

可解释性表示模型能够使用人类可认知的说法进行呈现和理解的程度^[51,52], 现有的轨迹异常检测研究基本侧重于设计精确的检测模型, 也就是由数据驱动的黑盒模型, 忽视了对已识别异常的解釋能力. 许多新发表工作都验证了可以在特定异常检测任务上取得良好的性能, 但仍然需要从更为详细的角度理解得出结论的原因. 为模型赋予较强的可解释性有利于确保其公平性和鲁棒性, 说明输入到输出之间各状态的因果关系, 提升算法的可信度. 从轨迹异常检测方法推导合理的异常解释仍然是一个未解决的问题, 特别是对于结构复杂的模型. 开发具有可解释性的异常检测模型至关重要, 但如何平衡模型的可解释性和有效性也是一个主要挑战.

4 轨迹异常检测研究现状分析

轨迹异常检测作为一项重要的研究课题, 已经在众多领域得到了应用. 尽管当前面临着许多棘手的挑战, 但随着研究方法的不断深入与改进, 研究人员已经提出了各种异常检测算法来解决这些问题. 纵观轨迹异常检测近 20 年的研究成果, 本文根据不同的分类标准, 将现有方法分为 3 大类. 具体来说, 根据轨迹数据标签的可用性, 异常检测算法包括基于有监督、弱监督、半监督、无监督的方法; 根据算法原理或研究方法, 轨迹异常检测有基于距离、密度、分类、聚类、统计、图、模式学习的方法; 根据异常检测算法的工作方式, 可以分为在线和离线的轨迹异常检测. 详细分类情况如图 3 所示. 本节根据图 3 所示的分类标准, 对公开文献中已发表的轨迹异常检测算法进行总结与分析, 并详细剖析每种算法的基本原理、代表性方法、复杂度、优缺点等.

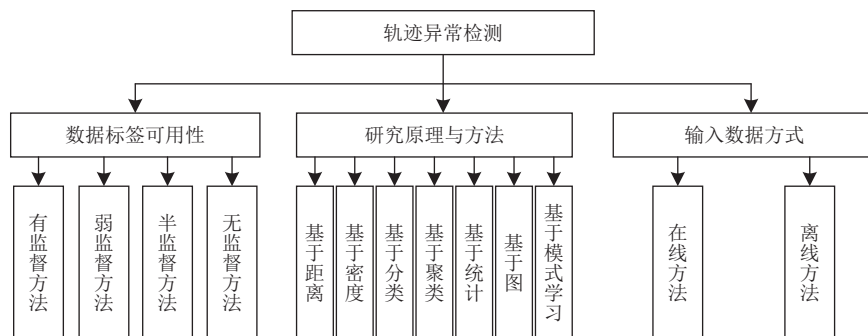


图 3 轨迹异常检测算法分类

4.1 基于数据标签的可用性

根据轨迹数据标签的可用性, 本节将轨迹异常检测研究分为有监督、弱监督、半监督、无监督等方法.

4.1.1 有监督轨迹异常检测方法

有监督的异常检测方法在性能上优于无监督异常检测, 因为有监督方法在模型构建过程中使用了标记数据. 有监督异常检测从一组带标签的轨迹数据中学习分离边界 (训练阶段), 然后使用学习的模型将测试轨迹分类为正常或异常 (测试阶段). 通常, 基于有监督的异常检测方法包含两个子任务, 即轨迹特征提取和轨迹分类. 深度模型则需要成千上万的训练数据来有效地学习轨迹特征表示, 以区分不同的类别. 由于缺乏干净的数据标签, 监督异常检测方法往往不如半监督和无监督方法使用广泛.

Mondal 等人^[53]使用基本的模糊理论来检测视频中移动对象的速度和路径偏差. 该方法首先根据每条轨迹将

视频帧图像分割为一组片段并得到每个分割区域的图像,然后计算出每个区域的平均速度,接着,求解了训练集的平均路径偏差,并根据所有轨迹路径偏差的模糊性进行分类.在实现速度模糊后,通过支持向量机检测到特定区域的异常轨迹.Laxhammar 等人^[54-56]借助 Conformal Prediction 理论^[57](即利用历史经验来确定最新预测的置信度水平)在轨迹异常检测方面做了大量工作,提出了基于 Hausdorff 距离的最近邻异常检测器 SHNN-CAD 用于在线学习和轨迹序列异常检测.作为一种轻量级的监督异常检测算法,SHNN-CAD 为异常阈值的调整提供了合理的方法.作者研究了算法在如图 4 所示的轨迹数据集(含标签信息)上在线学习和连续异常检测的性能,图 4 中灰色轨迹标记为正常,黑色轨迹标记为异常.结果表明,SHNN-CAD 以最小的参数调整实现了较好的分类效果.Ma 等人^[58]提出了基于循环神经网络的轨迹距离度量方法来计算轨迹间相似性并进行异常检测.该方法首先使用自编码器来捕获轨迹的动态特征,两条轨迹之间的距离由基于学习模型的重构误差定义.然后,利用提出的度量标准,基于最近邻方法完成轨迹异常检测.值得注意的是,该方法能够以有监督或无监督的方式处理不同场景中的异常轨迹,作者在 4 个开源的带标签轨迹数据集上完成了验证.

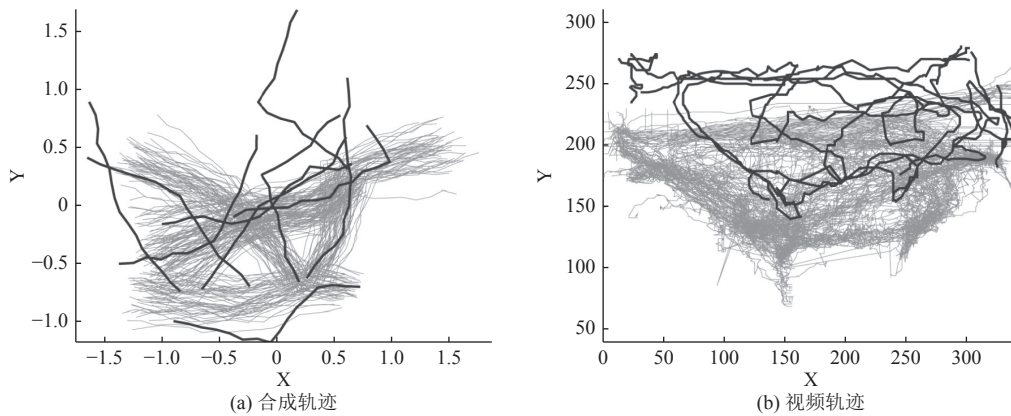


图 4 轨迹实例可视化

有监督的轨迹异常检测方法计算复杂度取决于输入轨迹数据的规模和算法模型的复杂程度.例如,基于深度学习的异常检测模型,计算复杂度会随着模型隐藏层数量线性增加,需要更多的模型训练和更新时间.有监督的异常检测方法优点在于比半监督和无监督方法拥有更高的准确率,局限性是监督方法需要对各种正常类和异常类轨迹进行准确的标记,而这在实际应用中往往是不容易获取的.

4.1.2 弱监督轨迹异常检测方法

正如第 4.1.1 节所述,尽管当前监督学习方法已经取得了巨大成功,但是由于数据标注过程的成本太高,很多情况下难以获得全部的真值标签.而无监督学习过程相对困难,因此,采取弱监督的策略是可取的^[59].通常来说,弱监督学习可以分为 3 种类型:第 1 类是不完全监督,即只有训练集的一个很小的子集是有标签的,其他数据没有标签.第 2 类是不确切监督,即数据只有粗粒度的标签.第 3 种是不准确监督,即数据标签不总是真值,有可能存在标签错误的情况.

近年来关于弱监督异常检测的研究侧重于视频中的异常事件和异常轨迹检测.Sultani 等人^[60]提出了一种基于深度多实例排序的弱监督算法,本质上是采取两阶段学习策略的框架,分别通过两个网络模型完成特征提取和异常分值获取,考虑到异常标记可能会受到噪声的影响,以及异常事件的连续性属性,给异常检测部分增加了稀疏约束和时间平滑约束.文献^[60]另外一个贡献是提出了一个真实场景下的大规模异常事件检测数据集 UCF-Crime.该数据集包含 13 种异常事件,共有 1900 个视频,划分有 1610 个视频做训练集,其中正常类别 800 个,异常类别 810 个.测试部分包括 290 个视频,其中正常类别有 150 个.UCF-Crime 在视频数量和异常事件类型上都远超以往数据集,对近几年视频异常检测方向的研究起到了很大的帮助作用.Zhong 等人^[61]将文献^[60]中的多实例学习问题转换为噪声标签下的监督学习问题,此处的噪声标签是指异常视频中正常片段的错误标注.通过对异常部分的

标注进行一定的噪声清除,可以直接在弱监督异常检测的情况下采用一个全监督动作分类器,以生成更可靠的分类结果.作者在UCF-Crime数据集上经过了3次标签噪声清除后得到了最好的表现结果.文献[60,61]中的方法虽然可以有效地检测异常,但是当异常事件与正常事件只有细微的差异时,很容易产生误检.为了解决这一问题,Tian等人^[62]提出了一种鲁棒的时间特征量学习模型RTFM,该方法通过训练特征量学习函数来有效地识别正样本实例,大大提高了多实例学习方法对异常视频中负样本的鲁棒性.RTFM还借助注意力机制捕获长期和短期的时间依赖性来更好地学习特征.作者在UCF-Crime等4个数据集上的实验表明,RTFM显著提升了多实例学习方法对细微异常事件的判别能力.近年来,有监督学习和无监督学习方法已经在轨迹异常检测问题中得到了成功的应用,而研究人员对介于二者之间的弱监督学习技术也产生了浓厚的兴趣.由于标注数据的获取成本较高,有研究者尝试用迁移学习和数据众包的方式来解决标签缺乏问题.

4.1.3 半监督轨迹异常检测方法

半监督学习的目的是利用易于获取的未标记数据来提高监督分类器的性能.半监督轨迹异常检测方法假设所有训练数据只有一种类别标签,在输入空间和特征学习空间中彼此接近的轨迹点有可能共享同一个标签,不属于多数类的测试轨迹则被标记为异常.

Sillito等人^[63]提出了一种新的学习框架来发现视频监控数据中的异常行为轨迹,在人工操作的帮助下,逐步训练区分正常轨迹和异常轨迹模式的分类器.作者从运动轨迹的角度考虑行人的行为,并使用3次样条曲线的控制点来参数化轨迹.然后,使用增量式半监督学习方法,可以在未标记的轨迹和偶尔正常的轨迹中发现异常行为.现有的大部分轨迹异常检测方法参数敏感且需人工手动调节,导致算法缺乏稳定性和可扩展性.针对上述问题,黄添强等人^[64]提出一种基于半监督技术的异常检测方法STOD,根据已知的轨迹信息确定敏感参数,并从整体与局部相结合的角度设计轨迹相似性度量,以发现更深层次有意义的异常轨迹.大多数关于轨迹异常检测的研究都旨在发现相邻时空轨迹之间的差异,但忽略了正常轨迹之间以非常规模式相互作用而产生的异常轨迹.为了解决这个问题,Wang等人^[65]提出了一种基于稀疏主题模型的半监督方法来检测视频监控中的异常行为轨迹,该方法对获取运动信息具有较强的鲁棒性.首先,为了增强轨迹点的空间和时间关系,采用Fisher核方法描述每条轨迹,并将视频看作由轨迹生成的可视文字组成的文档.然后,文中提出了稀疏主题模型,它不仅可以捕获运动模式,还可以对可视文字进行编码,实现视频场景的稀疏表示.最后,应用半监督学习方法增强模型的识别能力,提高异常检测的性能.视频监控场景中异常事件的特点是移动对象存在不规则的轨迹模式,Aljaloud等人^[66]最近提出了一种新颖的不规则感知半监督深度学习模型IA-SSLM用于检测异常事件.该方法利用半监督深度学习模型从有限数量的标记数据样本中自动学习特征表示,在训练阶段从标记和未标记数据中提取有用的场景特征以提高异常检测性能.

基于半监督的轨迹异常检测方法计算复杂度与有监督检测方法相似,主要取决于模型隐藏层数量和轨迹数据规模.由于存在一种类别的标签可用,半监督异常检测方法往往取得比无监督方法更好的检测性能,但是在基于半监督方法的模型中,隐藏层提取的轨迹特征可能无法代表较少的异常轨迹,因而容易出现过拟合问题.

综合第4.1.2节和第4.1.3节的内容可以看到,研究人员在基于轨迹信息的视频异常事件检测方面做了大量工作,在检测算法不断创新的同时,也提出了多种场景下的异常数据集,对轨迹异常检测和视频异常事件检测的研究提供了强有力的支持.本文对现有研究中几个典型的基准数据集进行简单介绍,每个数据集分别选取了3个代表性的场景示例,如图5(a)-(d)所示.其中,图5(a)为UCF-Crime数据集^[60],共包含13种异常事件,视频总时长128h.图5(b)所示为Violent Flows数据集^[67],包括246个真实世界的人群暴力视频(123个暴力和123个非暴力).该数据集唯一的目的是评估人群暴力分类算法,这是一个相对较新的数据集,还没有得到广泛测试.图5(c)为UCF Web Crowd Abnormality数据集^[68],该数据集收集自托管网站,一共由20个视频组成,其中,行人或正常人群行为相关视频有12个,逃跑恐慌、示威者冲突、人群打斗等异常行为视频有8个.图5(d)所示为UMN Unusual Crowd Activity数据集^[69],该数据集采集自明尼苏达大学,由逃逸事件的11个不同场景视频组成.每个视频片段都以正常行为开始,以一系列异常行为结束.其他视频异常检测数据集,如Shanghai-Tech^[70]、XD-Violence^[71]、UCSD^[72]、CUHK Avenue^[73]等也常被用来评估异常检测算法的性能,在此不作详细介绍.



图 5 视频异常检测数据集

4.1.4 无监督轨迹异常检测方法

无监督模式下的轨迹异常检测算法不需要带标注的训练数据,因此在研究和工业领域得到了最广泛的应用.当轨迹数据没有标签信息时,训练数据和测试数据之间通常没有区别,基于无监督的异常检测方法隐含的假设是数据集中正常轨迹所占的比例远大于异常轨迹,否则,就会产生很高的误报率.传统的无监督学习主要通过聚类方法将正常轨迹与异常轨迹区分开来,随着深度学习模型的出现,自编码器、生成对抗网络等生成式模型成为无监督异常检测的主要架构.

Das 等人^[74]针对拥挤场景提出了一种改进的基于聚类的无监督异常轨迹检测算法。该方法主要包括 4 个步骤, 首先, 使用多目标跟踪器提取拥挤场景视频中所有移动对象的轨迹; 然后提取轨迹数据多个特征 (例如, 密度、形状、中间位置、标准差等) 并将它们转换为独立的特征空间; 接着将 Mean Shift 聚类算法应用于上述特征矩阵以获得不同的聚类; 最后设计基于信息熵的检测器来检测异常轨迹。文中还使用投票表决机制来选择在大多数情况下表现出异常行为的轨迹。Zhao 等人^[75]提出了一个轨迹异常检测与定位框架, 该框架由异常轨迹估计和一致性运动对象构建两个阶段组成。同时, 为了更好地捕捉场景中轨迹点的细粒度时空信息, 文中提出了一种称为基于轨迹点的光流直方图 (PT-HOF) 的新特征。在第 1 阶段, 提取轨迹新特征 PT-HOF 并通过无监督自编码器网络对其进行编码, 用于估计每条轨迹异常的可能性。在第 2 阶段中, 将局部区域内运动相似的轨迹点聚类为一致性运动对象。最后, 结合轨迹异常的可能性和一致性运动对象, 采用投票表决机制检测和定位异常轨迹。在现有的轨迹异常检测研究中, 一些研究人员认为, 可以通过重构正常轨迹的特征来确定测试轨迹是否异常, 因为当异常轨迹或事件发生时, 往往伴随着场景和轨迹分布的变化。事实上, 交通异常通常涉及很大的时间和空间范围, 场景变化也具有多样性, 所以, 简单地利用轨迹重构思想来设计异常检测模型并不能保证其稳定性和健壮性。为此, Zhao 等人^[76]基于轨迹跟踪方法, 提出了一个鲁棒的框架解决无监督交通异常检测问题, 即从视频背景图像序列中检测和跟踪潜在的异常。首先, 设计了一种新的多目标跟踪算法, 从视频背景序列中获取轨迹。然后, 提出了一种自适应的无监督道路掩码生成方法, 以过滤道路区域以外的错误异常检测。最后, 结合获取的轨迹信息, 利用决策模型来确定轨迹异常的可能性, 并精确定位异常事件的开始时间。现有的一些轨迹异常检测算法不能充分考虑轨迹的特性, 检测结果缺乏可解释性, 同时也受到轨迹空间属性的限制。针对上述问题, Wang 等人^[77]提出了一种基于深度表示的无监督轨迹异常检测算法 TAD-FM, 从数据获取与处理到最终异常检测, 算法主要分为 4 个阶段。在轨迹数据划分阶段将每条轨迹划分为线段。在深度特征提取阶段, 首先将轨迹段转换为浅层特征序列, 然后将浅层特征序列输入到基于自编码器的深度特征融合模型中, 得到轨迹段的深度特征序列表示。在聚类阶段, 采用经典的基于密度的聚类算法 DBSCAN^[78], 通过度量融合特征序列之间的余弦距离进行聚类。在异常检测阶段, 通过比较融合特征序列之间的余弦相似度提取异常轨迹段, 然后考虑异常轨迹段在整体轨迹中的比例, 最后检测异常轨迹。

从上述研究文献可以看出, 传统的无监督异常检测通常基于聚类方法实现, 而自编码器是无监督轨迹异常检测方法中最常用的深度模型, 模型复杂度主要取决于聚类迭代次数、网络参数和隐藏层数量等, 而且训练自编码器的计算复杂度远高于聚类方法。无监督的异常检测方法可以学习固有的数据特征, 以更好地分类正常和异常轨迹, 而且无监督方法也是相对经济有效的选择, 因为它不需要带标记的数据来训练算法。但是, 无监督方法对噪声数据非常敏感, 在准确性方面的表现通常不如监督或半监督方法。另外, 在使用自编码器的模型中, 有许多超参数需要不断调整才能取得最佳检测结果, 这对于异常检测算法优化来说也存在一定的挑战。

基于数据标签可用性的异常检测方法详细分类情况对比及代表性方法如表 2 所示。

表 2 基于轨迹数据标签可用性的异常检测方法分类

方法类别	特点	劣势	代表性方法
有监督方法	需要带标签数据、检测性能相对最优、方法可靠性较高	准确的数据标注成本较高、算法应用场景有限	Mondal 等人 ^[53] 、Laxhammar 等人 ^[54] 、Ma 等人 ^[58]
弱监督方法	极小部分数据有标签、包含粗粒度标签、存在标签错误的情况	获得标签信息难度较大、算法应用场景有限	Sultani 等人 ^[60] 、Zhong 等人 ^[61] 、Tian 等人 ^[62]
半监督方法	同时使用有标签和无标签数据、检测性能优于无监督方法	标签混合数据容易导致过拟合、应用范围有限	黄添强等人 ^[64] 、Aljaloud 等人 ^[66]
无监督方法	不需要带标注的数据、应用场景广泛	检测性能不如监督方法、对噪声数据敏感	Das 等人 ^[74] 、Zhao 等人 ^[75] 、Wang 等人 ^[77]

4.2 基于算法原理

4.2.1 基于距离的轨迹异常检测方法

基于距离的轨迹异常检测方法的基本思想是, 如果一条轨迹距离大多数其他轨迹来说都很远, 则该轨迹可能

为异常轨迹. 基于距离的方法在轨迹异常检测研究初期就已经得到了广泛的应用, 因为这类方法更容易确定轨迹之间有意义的距离度量方式, 而不是确定统计分布模型. 另外, 现有文献也提供了多种可供选择的度量轨迹间距离的方法. 本节首先对常用的轨迹间距离或轨迹相似性度量方法进行总结, 主要包括 Euclidean 距离^[79]、DTW 距离^[80]、Hausdorff 距离^[81]、LCSS 距离^[82]、基于 HMM 的距离^[83].

轨迹是用二维或三维坐标表示的时间序列, 在这里, 方便起见, 将轨迹表示为二维坐标形式, 如公式 (1), 公式 (2) 所示.

$$((a_1^x, a_1^y), (a_2^x, a_2^y), \dots, (a_n^x, a_n^y)) \quad (1)$$

$$((b_1^x, b_1^y), (b_2^x, b_2^y), \dots, (b_m^x, b_m^y)) \quad (2)$$

公式 (1), 公式 (2) 表示长度分别为 N 和 M 的二维轨迹 A 与 B . 以下是 5 种轨迹相似性度量方法 (详细见对应参考文献).

(1) Euclidean 距离^[79]

欧氏距离是最常见的两点或多点之间的距离表示法, 采用两条轨迹上对应点之间的平均距离, 该方法计算简单, 但需要注意的是, 欧氏距离容易受到噪声的影响, 而且计算时两条轨迹的长度必须相同. 轨迹 A 、 B 之间欧氏距离计算如下:

$$D_1(A, B) = \frac{1}{N} \sum_{n=1}^N [(a_n^x - b_n^x)^2 + (a_n^y - b_n^y)^2]^{\frac{1}{2}} \quad (3)$$

(2) DTW 距离^[80]

当两条轨迹等长时, 可以使用欧氏距离来度量二者的相似性. 但是当轨迹长度不一样时, 欧氏距离就难以发挥作用了. 为此, 国外学者提出了动态时间弯曲距离 (dynamic time warping, DTW) 作为一种新的相似性度量方法, 通过调节时间点之间的对应关系, 能够寻找两个任意长度轨迹序列之间的最佳匹配路径 W , 可以更有效地度量轨迹之间的相似性. 由于 DTW 不要求两条轨迹序列中的点一一对应, 因此适用范围更广泛. 但 DTW 时间复杂度更高, 计算代价较为昂贵. DTW 距离可以表示为:

$$D_2(A, B) = \min \left\{ \frac{1}{K} \left[\sum_{k=1}^K w_k \right]^{\frac{1}{2}} \right\} \quad (4)$$

其中, w_k 为匹配路径 W 的第 k 个元素.

(3) Hausdorff 距离^[81]

Hausdorff 距离是描述两组点集之间相似程度的一种度量方法, 具体到轨迹集合, A 与 B 之间的 Hausdorff 距离定义为:

$$D_3(A, B) = \max\{d(A, B), d(B, A)\} \quad (5)$$

其中,

$$d(A, B) = \max_{a \in A} \min_{b \in B} \|a - b\| \quad (6)$$

(4) LCSS 距离^[82]

DTW 和欧氏距离对轨迹中的噪声点都比较敏感, 因为包括噪声在内的所有点算法都需要进行匹配. 为了解决这一问题, 研究人员提出了最长公共子序列 (longest common subsequence, LCSS) 来更鲁棒地度量轨迹间的距离. LCSS 的基本思想是允许忽略一些距离较远的轨迹点, 以增强算法对噪声数据的鲁棒性. 假设 $Head(A)$ 是轨迹 A 的前 $N-1$ 个点, $Head(B)$ 是轨迹 B 的前 $M-1$ 个点, 给定间隔阈值 δ 和距离 ε , 对于轨迹数据 A 与 B , 它们的最长公共子序列的长度为:

$$LCSS_{\delta, \varepsilon}(A, B) = \begin{cases} 0, & A \text{ 或 } B \text{ 为空} \\ 1 + LCSS_{\delta, \varepsilon}(Head(A), Head(B)), & \text{如果 } \|a_N - b_M\| < \varepsilon \text{ 并且 } |N - M| \leq \delta \\ \max(LCSS_{\delta, \varepsilon}(Head(A), B), LCSS_{\delta, \varepsilon}(A, Head(B))), & \text{其他} \end{cases} \quad (7)$$

基于上述公式, 最长公共子序列的相似度为:

$$D_4(\delta, \varepsilon, A, B) = 1 - \frac{LCSS_{\delta, \varepsilon}(A, B)}{\max(N, M)} \quad (8)$$

(5) 基于 HMM 的距离^[83]

基于隐马尔可夫模型 (hidden Markov model, HMM) 的距离度量方法由 Porikli^[83]提出, 其中每条轨迹采用 HMM 进行拟合. 基于 HMM 的距离定义为:

$$D_5(A, B) = |d(A; \lambda_A) + d(B; \lambda_B) - d(A; \lambda_B) - d(B; \lambda_A)| \quad (9)$$

其中, $d(A; \lambda_A)$ 和 $d(B; \lambda_B)$ 表示轨迹来自自身模型的可能性, $d(A; \lambda_B)$ 和 $d(B; \lambda_A)$ 表示该轨迹由其他轨迹模型生成的可能性.

另外, 还有基于编辑距离的 EDR (edit distance on real sequence)^[84]和 ERP (edit distance with real penalty)^[85]以及 Fréchet 距离^[86]等度量方法也常应用于轨迹相似性分析和异常检测. 基于距离的轨迹异常检测及代表性方法如图 6 所示.



图 6 基于距离的轨迹异常检测方法

Knorr 等人^[87, 88]最早引入了基于距离的异常值的概念, 并提出了轨迹异常检测算法. 他们将原始轨迹转换成由几个独立属性组成的对象, 即从轨迹中抽取几个关键特征来表示原始轨迹. 这些特征包括: 表示轨迹起点和终点属性的坐标信息, 表示长度属性的轨迹点个数, 表示方向属性的轨迹点切线方向的最大值和最小值以及表示速度属性的整条轨迹中的最大速度和最小速度等. 在轨迹异常检测阶段主要采用基于距离的异常点检测方法. 此类方法的优点是原理简单, 借助轨迹间距离计算方法即可实现, 但是随着轨迹长度和复杂度逐渐增加, 如果仅仅通过上述几个全局特征来判断异常轨迹显然是不准确的, 而且此类方法是以整条轨迹作为异常检测单位, 因此可能无法检测出偏离路径的异常子轨迹.

针对轨迹异常检测算法检测异常子轨迹能力有限的问题, Lee 等人^[89]介绍了一种新的轨迹异常点检测框架, 该框架将整条轨迹分割成一组线段, 然后对这组线段进行异常点检测, 优点是可以从轨迹数据中检测异常子轨迹. 在此基础上, 文中提出了轨迹异常检测算法 TRAOD, TRAOD 包括两个阶段, 首先在分割阶段, 对轨迹进行粗粒度和细粒度两级划分策略, 保证轨迹划分的准确性; 然后在检测阶段, 采用基于距离和密度的方法检测异常轨迹. 实验结果表明, TRAOD 能够从真实的轨迹数据中准确地检测出异常子轨迹, 但该方法时间复杂度较高.

TRAOD^[89]中所使用的距离度量仅涉及子轨迹形状和局部运动模式的不同, 并没有消除子轨迹之间的共同偏差. 刘良旭等人^[90]提出了一种基于 R-Tree 的轨迹异常检测算法, 该算法以 k 个连续轨迹点构成的片段作为基本比较单元, 然后采用 Hausdorff 距离度量比较单元之间的相似性, 当目标轨迹没有足够的相似比较单元时, 将其判断为异常轨迹. 相比 TRAOD 算法来说, 该方法具有较高的计算效率. 接着, Liu 等人^[91]又提出了一种基于相对距离的轨迹异常检测新框架 RTOD, 在 RTOD 中, 首先使用相对距离来衡量轨迹段之间的差异, 然后根据距离度量正式定义异常轨迹. 为了提高时间性能, 文中使用 R-Tree 和局部特征相关矩阵来消除不相关的轨迹段. 实验结果表明, RTOD 方法比文献 [89, 90] 中的算法具有更高的检测效率.

Laxhammar 等人^[55]研究了基于数据驱动的轨迹异常检测, 提出了一种新的参数轻量级在线学习和自动轨迹异常检测算法 SNN-CAD. SNN-CAD 算法的关键特性是异常检测的虚警率校准良好, 如果训练数据与新的正常数据是独立同分布的, 则期望的虚警率等于既定的异常阈值. 因此, 与之前提出的算法相比, SNN-CAD 不需要任何

指定的异常阈值. 文中提出了两种基于 Hausdorff 距离的无参数非相似性度量方法, 用于衡量任意长度的两个多维轨迹间的非相似性. 分别基于模拟和真实视频轨迹的两个公共轨迹数据集, 实证研究了与 SNN-CAD 的学习和分类性能相关的基本特性以及两种不同的度量方法.

为了提高轨迹异常检测的准确率, Zhu 等人^[92]同时考虑时间和空间异常, 提出了时间依赖的异常检测算法 TPRO. 对于每一组具有相同来源和目的地的轨迹, 首先设计一个随时间变化的传递图, 在不同的时间段, 可以得到 top- k 条最受欢迎的路线作为参考路线. 对于这段时间内的待检测轨迹, 如果在空间维度和时间维度上与所选路线存在较大差异, 则将其标记为异常. 为了定量地度量轨迹与路线之间的差异, 文中提出了一种基于时空编辑距离的时变距离度量方法. Wang 等人^[93]专注于出租车轨迹异常检测研究, 提出了基于编辑距离和层次化聚类的轨迹异常检测算法, 以推断司机欺诈行驶路线和不良交通事故的发生. 首先, 从出租车轨迹中获取所有穿过相同源-目的地对的出租车轨迹, 并将这些轨迹作为聚类对象. 其次, 改进编辑距离算法以度量轨迹的相似性. 然后, 采用基于最佳聚类数的自适应层次聚类方法来区分正常轨迹和异常轨迹. 为了结合轨迹上下文信息进行异常检测, San 等人^[94]提出了一组用于轨迹描述的新特征, 以及一种新的上下文感知距离 CaD, 该距离由轨迹角度差的加权平均值, 欧氏距离, 以及每条轨迹上的点数等参数构建. 基于 CaD 距离, 文中提出了一种无监督方法来识别从视频监控系统中提取的异常行人轨迹. 该方法主要分为 4 个阶段: 特征提取、距离矩阵计算、聚类和异常检测. 特征提取采用线性分割算法对轨迹进行简化产生子轨迹向量. 然后, 使用基于上下文感知的距离 CaD 计算轨迹间的距离矩阵. 距离矩阵提供了一种无监督学习方法来提取轨迹的同质组. 接着, 对每个同质组的轨迹进行聚类. 最后, 在每个聚类簇中, 使用 k 近邻 (k -nearest neighbor, k NN) 方法检测异常轨迹.

对于任何基于距离的异常检测方法来说, 合适的距离度量都是相当重要的, 不仅局限于轨迹异常检测领域的应用, 距离度量方法在轨迹分类和聚类等研究中也发挥着重要作用. 从本节开头部分总结的轨迹间距离的度量方法可以看出, 基于距离的异常检测算法时间复杂度从 $O(N)$ 到 $O(N \times M)$ 不等, 所以选取合适的距离度量方式, 可以有效降低算法计算成本. 基于距离的轨迹异常检测算法优点在于原理简单, 在文献中有丰富的理论支持, 实现过程不需要依赖各种假设的分布来拟合数据. 然而, 这类算法最坏的时间复杂度可能会达到 $O(N^2)$, 计算成本过高, 无法处理大规模场景的轨迹数据. 此外, 这些方法对计算过程中引入的超参数也比较敏感.

4.2.2 基于密度的轨迹异常检测方法

第 4.2.1 节中所述的基于距离的轨迹异常检测算法都需要选择一个合适的全局距离阈值来判断轨迹是否异常, 用全局距离阈值检测到的异常轨迹可以视为全局异常. 然而, 轨迹数据集可能呈现出复杂多变的分布特征, 很多情况下异常轨迹与其邻域的轨迹密度相关, 特别是当轨迹局部密集时, 由于距离阈值选取困难, 基于距离的方法无法很好地检测异常轨迹. 基于密度的方法与基于距离的轨迹异常检测方法密切相关, 因为密度通常是由距离来定义的. 基于密度的异常检测方法的核心思想是: 低密度区域内的值被判定为异常点, 而正常点则出现在相对稠密的区域. 尽管基于密度的异常检测方法相比基于距离的方法拥有更复杂的异常点建模机制, 但是基于密度的检测方法的简单性和有效性使得这类方法仍然被广泛地应用于轨迹异常检测.

Breunig 等人^[95]引入了一种新的局部异常值概念, 即根据局部邻域的密度来度量目标数据被判定为异常值的程度, 并提出了局部异常因子 (local outlier factor, LOF), LOF 是最早的基于密度的异常检测方法之一. 对于任何给定的数据实例, LOF 得分等于该数据实例的 k 个最近邻数据的平均局部密度与数据实例本身的局部密度之比. 为了计算数据实例的局部密度, 文中首先找到以该数据为中心, 包含其 k 个最近邻数据的最小超球的半径. 然后, 局部密度的计算方法是用 k 除以这个超球的体积. 对于位于稠密区域的正常数据, 其局部密度与邻近数据相似, 而对于异常数据, 其局部密度则低于邻近数据. 因此, 异常数据将获得更高的 LOF 分数. 在后期的研究中, 学者们逐渐提出了 LOF 方法的多种变体形式, 其中有些变体以不同的方式计算目标数据的局部密度, 有些变体对原始 LOF 方法进行了调整, 以适应更复杂的数据类型, 其中具有代表性的工作有: Tang 等人^[96]对 LOF 进行了改进, 提出了一种基于连接的异常因子 (connectivity-based outlier factor, COF) 方法, 该方法与 LOF 非常相似, 唯一的区别是计算数据 k 邻域的方式不同. 在 COF 中, 数据实例的邻域以增量方式计算. 首先, 将与目标数据距离最近的实例添加到邻域集. 然后, 添加到邻域集的下一个实例与现有邻域集的距离是所有剩余数据实例中最小的. 单独的数据实例

和一组实例之间的距离定义为给定数据和属于该集合的任意实例之间的最小距离。最后, 数据邻域以这种方式增长, 直到大小为 k 。数据邻域计算结束后, COF 采取跟 LOF 相同的方式完成异常评分。Goldstein 等人^[97]通过不同应用领域的数据集比较了 COF 和 LOF 等异常检测方法, 发现 LOF 的球形密度估计不利于有效检测异常值。现有的异常检测方法常用于一维或二维数据, Tang 等人^[98]针对大规模城市交通视频数据, 提出了一种基于密度的有界 LOF (bounded LOF, BLOF) 异常检测方法。该方法首先通过主成分分析对时空交通数据进行降维, 然后将 BLOF 分别应用在二维和三维数据域上进行对比测试与评估。此外, 文中还将经典 LOF 方法^[95]扩展到了三维数据域进行异常检测。从上述方法原理介绍可以看出, LOF 算法及其变体方法的时间复杂度均为 $O(N^2)$, 其中 N 表示数据集大小。虽然 LOF 算法可以有效检测异常点, 但它的时间复杂度相对较高, 导致这类方法往往不适合交通流轨迹等大型数据集异常检测任务。为了克服 LOF 方法的缺点, 研究人员基于密度的思想设计了复杂度更低、检测效率更高、精度更准的轨迹异常检测算法。

为了捕捉异常运动轨迹的演化本质, Ge 等人^[99]提出了一种进化的轨迹异常检测方法 TOP-EYE, 该方法考虑了两种类型的异常轨迹: 方向上的异常轨迹和密度上的异常轨迹。将连续空间离散为小网格, 利用概率模型将网格中轨迹的方向信息转化为向 8 个方向移动的概率。因此, 方向趋势是通过一段时间内大量轨迹的移动方向进行汇总而产生的。然后, 一旦有些个体沿着与汇总方向完全不同的方向穿过该区域, 通过测量被观测个体与汇总方向之间的相似性, 就可以实时标记为异常点。每个网格内的轨迹密度被估计为穿过这个网格的轨迹数目。轨迹异常评分可以根据网格中的轨迹密度来度量。TOP-EYE 方法的优点是可以提前识别出进化的异常轨迹。

Liu 等人^[100]提出了一种基于密度的轨迹异常点检测方法 DBTOD, 该方法弥补了 TRAOD^[89]算法在轨迹局部密集时无法检测到异常的缺陷。DBTOD 引入的轨迹密度由两个分量组成: 子轨迹间的距离和给定范围内子轨迹的数量。类似于 TRAOD, DBTOD 算法也包括分割和检测两个阶段。在分割阶段, 采用 TRAOD 中的两级划分策略对每条轨迹进行分割。在检测阶段, 采用基于密度而不是基于距离的检测算法发现异常子轨迹。DBTOD 算法的优点是它既能检测异常子轨迹也可以检测局部异常轨迹。同样针对 TRAOD 算法无法从局部密集轨迹中检测异常点的问题, Luan 等人^[101]提出了一种基于局部密度的轨迹异常检测算法 LDTRAOD。该算法利用分割检测框架计算每条划分轨迹的局部密度, 并根据局部密度计算其局部异常因子。如果局部异常因子大于检测阈值, 则该条划分轨迹被标记为异常。Lan 等人^[102]使用轨迹序列的编辑距离^[84]和密度来检测工人的异常轨迹, 以及及时发现工作场所的紧急情况与安全事故。文中提出的异常检测方案包括两个阶段: 在离线阶段确定距离阈值与密度阈值, 在线阶段则完成轨迹异常判断。使用距离阈值计算输入轨迹的邻居密度。然后, 如果输入轨迹的密度小于密度阈值, 则将其标记为异常。为了检测长时间交通状况不佳的异常区域, Kong 等人^[103]提出了一种长时交通流轨迹异常检测方法 LoTAD。LoTAD 包括以下步骤: 首先, 创建时空轨迹段 TS, 这一步的目的是从公交轨迹和公交车站线路数据库创建 TS 数据库。然后, 计算轨迹异常指数。使用曼哈顿距离度量方法计算每个路段 TS 的密度, 并应用 LOF 算法查找异常路段。最后, 确定交通异常区域。对公交车站线路数据应用 K-means 算法^[104]寻找相似区域, 将每个区域所属路段的 LOF 值相加得到该区域的异常评分。Huang 等人^[105]结合了 k 近邻和 LOF 算法各自的优点, 提出了 KNN-LOF 方法, 用于船舶行为轨迹异常检测。该方法首先通过 k 近邻对船舶异常数据候选集进行过滤, 然后利用 LOF 算法计算轨迹局部偏差指数, 最后通过设置阈值来判断船舶异常轨迹。

在基于密度的轨迹异常检测方法中, 轨迹密度估计过程是非参数的, 不依赖任何假设的分布来拟合数据。经典的基于密度的方法如 LOF^[95]、TOP-EYE^[99]等已经成为许多后续研究的参考基准, 其他基于密度的方法主要是弥补 TRAOD^[89]算法在轨迹局部密集时无法检测到异常的缺陷。由于基于密度的方法是通过轨迹邻域密度来判断异常轨迹, 所以这类方法在轨迹局部异常检测方面相比基于距离的算法具有较大的优势。尽管一些基于密度的方法已经被证明具有更好的异常检测性能, 但是在大多数情况下, 它们的计算成本还是相对较高, 而且算法对参数设置也很敏感, 比如确定邻居大小、异常密度阈值等。

4.2.3 基于分类的轨迹异常检测方法

基于分类的异常检测方法在训练阶段使用大量轨迹数据学习分类模型, 然后在测试阶段使用学习的分类器将待评估轨迹分类为正常或异常, 分类器的学习过程可以采用有监督或无监督的方法。基于分类的轨迹异常检测算

法假设分类器可以从给定的轨迹数据中学习正常轨迹和异常轨迹的特征差异,从而具备区分轨迹类别的能力.根据训练阶段标签的可用程度,可以将基于分类的轨迹异常检测方法分为两类:多类轨迹异常检测 (multi-class trajectory anomaly detection, MCTAD) 和单类轨迹异常检测 (one-class trajectory anomaly detection, OCTAD).多类轨迹异常检测方法假设训练数据包含属于多个正常类的标记轨迹,分类器需要学会区分每个具体的正常类和其余类.如果测试轨迹没有被任何分类器分类为正常,则认为该轨迹为异常,如图 7(a) 所示.单类轨迹异常检测假设所有训练轨迹数据只有一个类标签.分类器通过单类别分类算法(如单类支持向量机、单类核 Fisher 判别器)来学习正常轨迹周围的判别边界,任何不在学习边界内的轨迹数据都被视为异常轨迹,如图 7(b) 所示.

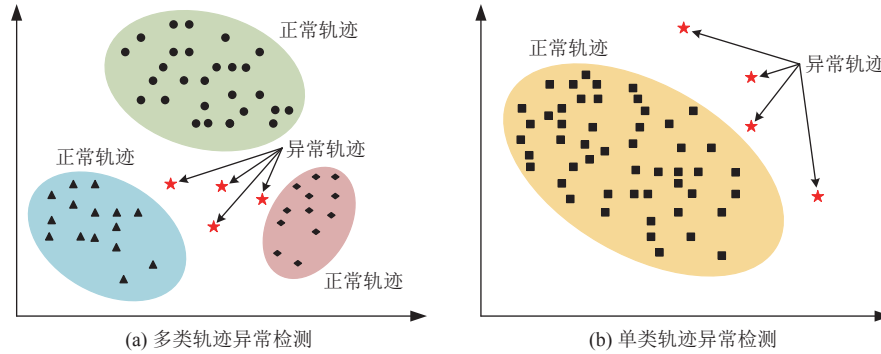


图 7 基于分类的轨迹异常检测

Li 等人^[106]提出了一种用于移动对象轨迹异常点检测的运动分类器 Motion-Alert,该分类器执行过程包括以下 3 个步骤.首先,从移动对象路径中提取时空相关的运动特征 motif,每条路径都由一系列 motif 表达式组成,这些表达式与时间和位置有关的值相关联.然后,为了发现移动对象运动过程中的异常,采用基于 motif 的泛化方法对相似的对象运动轨迹片段进行聚类.最后,将移动对象放入多层特征空间中,由可以处理高维特征空间的分类器进行分类,以区分异常轨迹和正常轨迹.在 Motion-Alert 算法的基础上, Li 等人^[107]又提出了一种基于分类的轨迹异常检测算法 ROAM. ROAM 框架由 3 个部分组成:基于 motif 的特征空间、特征空间层次自动提取、基于规则的层次分类器.基于 motif 的特征空间通过将轨迹划分为 motif 来寻找轨迹移动中最具代表性的模式,并基于 motif 构建具有关联属性的多维特征空间.特征空间层次自动提取通过检查轨迹的模式,自动提取特征空间中的层次结构,这有利于分析多条轨迹特征间的复杂关系.基于规则的层次分类器用于探索层次特征空间,并找到有效的分类区域以完成轨迹异常检测.实验表明 ROAM 能够有效地检测移动目标的异常轨迹.但是,该方法在处理复杂场景时,基于 motif 的轨迹划分可能会覆盖部分局部异常,导致无法检测到所有区域的异常轨迹. Piciarelli 等人^[108]提出了一种基于单类支持向量机 (one-class support vector machine, OCSVM) 的轨迹学习和异常检测算法.在该方法中,每条轨迹由原始轨迹均匀采样点的固定维度特征向量表示,在没有轨迹异常点分布先验信息的情况下完成轨迹分类.此外,该算法还可以自动检测和消除训练轨迹数据中的异常.现有大多数方法在异常检测过程中会忽略轨迹局部异常,为了充分考虑子轨迹中存在的局部微小异常, Yang 等人^[109]提出了一个基于轨迹分割和多实例学习的三阶段局部异常检测框架 TRASMIL. TRASMIL 首先将每条运动轨迹分割为独立的子轨迹,并提出了具有多样性和粒度的分割质量度量指标.其次,采用序列学习模型对分割后的子轨迹进行建模.最后,将异常轨迹和子轨迹分别视为包和实例,利用多实例学习对异常轨迹和子轨迹进行检测.

基于分类的轨迹异常检测可以利用强大的算法来区分不同类别的轨迹,算法复杂度往往取决于所使用的分类器,例如,训练决策树的速度一般很快,而 SVM 等涉及二次优化的方法训练成本较高.随着各种分类器的提出,特别是基于深度学习技术的分类器层出不穷,为进一步研究基于分类的异常检测方法提供了坚实的理论基础.但是,基于分类的方法依赖于正常轨迹的精确标签信息,而这在实际应用中往往是不容易获取的.

4.2.4 基于聚类的轨迹异常检测方法

聚类用于将相似的数据实例分组成簇,属于一种无监督的技术,因为它不需要任何先验知识.基于聚类的方法

用于发现强关联的数据组, 而异常检测的目的是发现与其他数据弱关联或无关联的数据. 尽管聚类和异常检测之间存在本质上的区别, 但是在聚类结果中, 包含较少数据对象、规模较小的簇可能被标记为异常值, 因此, 可以用聚类方法来检测异常点. 同时, 也有大量的研究致力于利用聚类结果来定义异常, 如聚类大小、目标数据到聚类中心的距离、聚类中心之间的距离、聚类成员关系等. 到目前为止, 有很多聚类算法可以用来检测数据集中的异常值, 如 K-means^[104]、DBSCAN^[78]、CURE^[110]、STING^[111]、WaveCluster^[112]等. 基于聚类的异常检测方法假设正常数据实例属于数据集中的某个簇, 而异常数据不属于任何簇, 或者正常数据属于大型密集集群, 而异常数据属于小型或稀疏集群. 基于聚类的异常检测方法的性能很大程度上取决于聚类算法在捕获正常数据的分布结构方面的有效性. 对于轨迹数据来说, 可以利用其时间、空间、邻居信息等特征进行聚类, 以发现轨迹异常行为模式为目标, 设计基于聚类的轨迹异常检测方法.

Piciarelli 等人^[113]提出了一种在线轨迹聚类算法, 该算法采用树状结构组织聚类, 加上概率信息, 可以用于检测视频监控系统中的异常事件. 在线聚类是在获取传入数据时进行聚类计算, 主要目的是避免经典的两步聚类法 (即数据收集和离线处理), 进而使用聚类信息进行移动物体在线行为分析. 为了检测拥塞轨迹异常, Ying 等人^[114]提出了一种基于聚类方法的拥塞异常检测框架 CTOD, 该框架主要包括相似性度量和聚类两个阶段. 为了度量轨迹段之间的相似性, 通过结合空间维度和时间维度, 引入了基于最小边界框的度量方法. 然后, 将 DBSCAN 聚类算法应用于整条轨迹的所有路段, 聚类结果中超密集轨迹簇被视为拥塞轨迹异常点, 类似于 TRAOD 算法, CTOD 也可以检测异常子轨迹.

大多数轨迹聚类算法都是将相似的轨迹作为一个整体进行分组, 但是这种聚类方式可能会遗漏公共子轨迹而无法检测到轨迹间的相似部分, 为了解决这一问题, Lee 等人^[115]提出了一种新的轨迹聚类框架, 该框架将轨迹划分为一组线段, 然后将相似线段分组到同一个簇中. 该框架的优点是可以从轨迹数据中发现公共子轨迹. 在此基础上, 文中提出了轨迹聚类算法 TRACCLUS, 该算法由轨迹划分和聚类分组两阶段构成. 在第 1 阶段, 提出了一种基于最小描述长度原理的形式化轨迹划分算法. 在第 2 阶段, 提出了一种基于密度的轨迹聚类算法, 将不同于大多数轨迹段的部分标记为异常点. 为了生成高质量的轨迹特征并达到较好的聚类结果, Lee 等人^[116]提出了一种特征生成框架 TraClass, 该框架通过对轨迹进行划分, 并探索两种类型的聚类 (基于区域的聚类和基于轨迹的聚类) 来生成特征层次结构. 基于区域的聚类在不借助任何运动模式的情况下捕获基于区域的高层次特征, 而基于轨迹的聚类使用运动模式捕获轨迹的低层次特征, 两种类型的聚类可以协同提取区域和子轨迹的特征.

TPRO^[92]算法专注于找出历史轨迹数据集中的所有异常值, 但在大多数情况下, 人们更关心算法对历史轨迹之外的任意新轨迹的异常检测结果. 为此, Zhu 等人^[117]提出了 TPRO 算法的改进版, 即基于时间依赖的流行路线实时轨迹异常检测方法 TPRRO. TPRRO 是一种实时检测算法, 它主要包含离线轨迹预处理和在线异常检测两个阶段. 在离线预处理步骤中, 根据历史轨迹数据构建时间依赖转移指数 TTI 和时间依赖转移图 TTG, 这两种高效的数据结构可以记录在不同的时间段内每条道路经过的轨迹数. 在 TTG 的帮助下, TPRRO 可以准确地检索指定时间段内最受欢迎的 k 条路线. 在在线异常检测步骤中, 利用 TTI 和 TTG 缓存加快检测速度, 有效地提高了算法响应时间. 实验结果表明, TPRRO 比 TPRO 具有更好的轨迹异常检测效率.

为了快速而准确地判断给定的轨迹是否属于异常值, Lv 等人^[44]在真实出租车轨迹数据上做了较为深入的研究, 并提出了两种基于轨迹聚类的异常检测算法. 首先, 文中提出了一种称为前缀树搜索的异常检测算法 PTS 作为研究基准, PTS 方法虽然能得到准确的检测结果, 但耗时较长. 为了提高异常检测效率, 文中提出了基于 K-medoids^[118]思想的原型异常轨迹检测算法 PBOTD. K-medoids 算法将轨迹数据划分为 k 个聚类, 每个聚类都由一个 medoid 作为代表性的数据来表示. 然后通过比较聚类结果中选择的有代表性的轨迹与待评估的轨迹, 快速判断测试轨迹是否异常. 但由于 k 的选择和局部最优问题, PBOTD 方法不能达到很高的检测精度. 为了克服这个缺陷, 文中又提出了一种基于密度聚类的异常轨迹检测算法 DBOTD. DBOTD 使用 DBSCAN 方法进行聚类, 并选择核心路径作为各个聚类的代表性轨迹, 偏离代表性轨迹的点即判断为异常值, 与 PBOTD 算法相比, DBOTD 能较好地提取代表聚类结果的轨迹, 更加有效地进行异常值检测. 基于聚类的异常检测通常需要选择合适的 k 作为算法输入, 为了消除 k 值选择的主观影响, Kumar 等人^[119]针对小数据和大数据场景案例, 提出了两种新的轨迹异常检测方法 iVAT+

和 clusiVAT+, 并将它们应用于真实的公共轨迹数据集. 文中的异常检测方法包括两个阶段, 第 1 阶段使用相似性度量对轨迹进行分组, 将具有相似路径但运动方向可能不同的轨迹分为一组. 第 2 阶段对生成的每个聚类使用方向相似性来分离方向相反的轨迹. 从轨迹集群中, 通过识别与同一集群中的其他轨迹距离太远的轨迹, 或通过识别轨迹数量太少的集群来获得轨迹异常值. 实验表明, 这种两阶段聚类方法可以在真实轨迹数据集上生成自然且信息丰富的轨迹集群, 有利于更加准确地进行轨迹异常检测.

基于聚类的轨迹异常检测不需要数据分布的先验知识, 属于无监督的方法, 所以在异常检测研究中可以广泛应用. 特别是随着深度学习技术的发展, 基于深度聚类的方法可以更加有效地提取轨迹特征, 与传统聚类方法相比, 更容易发现异常值. 但是, 聚类过程可能会受到训练数据中噪声的干扰, 导致聚类表示产生偏差而影响异常检测效果. 大多数聚类方法都依赖于用户预先指定的聚类数目 k , k 值的选择也给实现数据的精确聚类带来了一定的困难. 另外, 聚类方法的时空复杂度会随着聚类数目 k 的增加呈线性增长, 这导致基于聚类的异常检测方法不适合实时性要求较高的应用场景.

4.2.5 基于统计的轨迹异常检测方法

基于统计的异常检测是最早也是相对简单的异常值检测技术^[120,121], 从本质上来讲, 现有对各种类型数据进行异常检测的应用都可以归纳为这种方法. 基于统计的异常检测也称为基于模型的异常检测方法, 它的基本思想是将一个随机模型作用于给定的数据, 然后应用统计推断测试来确定待测数据是否属于该模型, 异常检测遵循的基本假设是: 正常的实例出现在随机模型分布的高概率区域, 而异常则出现在随机模型分布的低概率区域. 基于统计的异常检测方法通常分为参数方法和非参数方法两大类, 可以使用监督、半监督、无监督等形式来实现. 参数方法在给定的数据中假设了潜在的分佈模型, 并从已知的数据中估计分佈模型的参数, 而非参数方法没有任何分佈模型的先验知识假设. 本文将目前使用统计方法检测异常值的代表性研究按参数方法和非参数方法进行分类并讨论, 需要注意的是, 由于随机模型的多样化, 文中讨论的方法不仅可以应用于轨迹异常检测, 多数方法也可以扩展到其他类型的数据完成异常值检测任务. 基于统计的轨迹异常检测及代表性方法如图 8 所示.

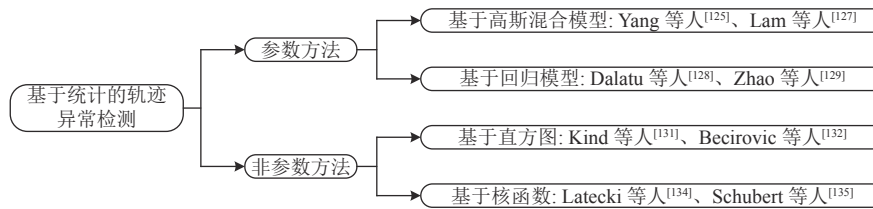


图 8 基于统计的轨迹异常检测方法

如前文所述, 参数方法假设正常数据是由带参数的随机模型生成的, 模型参数由给定数据来估计, 测试实例的异常评分通过概率密度函数计算得到. 根据假设的数据分佈类型, 参数方法可以进一步分为基于高斯混合模型^[122]的方法和基于回归模型^[123]的方法.

高斯模型是用于检测异常值的最常用的统计方法之一, 在高斯模型中, 训练阶段使用最大似然估计方法 (maximum likelihood estimation, MLE)^[124]对高斯分佈进行均值和方差估计. 在测试阶段, 采用统计不一致性检验 (如箱线图、均值-方差检验) 来完成异常检测. Yang 等人^[125]引入了基于范例的全局最优高斯混合模型 (Gaussian mixture model, GMM), 并提出了一种无监督的异常点检测方法. 在该方法中, 首先应用一个可证明的全局最优期望最大化算法拟合 GMM 到给定的数据集, 高斯分佈以每个数据点为中心, 因此, 估计的混合比例可以解释为所有数据点成为聚类中心的概率. 然后将每个数据点上的异常因子定义为混合比例的加权和, 其中权重表示与其他数据点的相似性. 所以, 文中提出的异常因子表达了数据集的全局特性, 这与大多数局部异常检测方法形成了对比. 另外, 该方法可以通过在给定的数据集集中的每个数据点拟合 GMM, 来解决基于聚类的方法在噪声污染存在时无法检测异常值的问题. 复杂非高斯过程可能会发生动态操作场景转换, 从而导致传统的异常检测方法性能下降. 为了提升算法鲁棒性, Tang 等人^[126]提出了一种基于子空间学习和高斯混合模型的能量分解异常点检测方法. 与主成

分分析 (principal component analysis, PCA) 相比, 子空间学习的局部保留投影算法 (locality preserving projections, LPP) 能够有效地保持数据的邻域结构, 揭示数据固有的流形结构, 并使得异常点远离正常数据. Lam 等人^[127]提出了一种基于 GMM 的方法来自动检测城市道路十字路口交通数据中的异常事件. 首先将视频格式记录的交通数据通过统计转换为时空交通信号, 然后使用 PCA 将时空交通信号投影到二维坐标平面以进行降维. GMM 方法认为交通数据是由高斯分布的混合形成的, 文中假设训练数据服从高斯分布, 然后设置置信区域完成测试数据异常检测.

在时间序列数据中, 使用回归模型进行异常检测已被广泛研究. 针对不同异常检测问题的数据属性, 用户选择的回归模型可以是线性的, 也可以是非线性的. 通常来说, 基于回归模型的方法包括两个步骤: 第 1 步为训练阶段, 即构建一个适合数据的回归模型, 然后第 2 步为测试阶段, 通过评估每个数据实例来测试回归模型, 当实际值与回归模型产生的预期值之间出现显著偏差时, 该数据实例被标记为异常值, 偏差的大小可以用作测试实例的异常评分. Dalatu 等人^[128]通过分析接受者操作特性曲线 (receiver operating characteristic curve, ROC 曲线), 对异常值检测的线性回归模型和非线性回归模型进行了比较研究. 结果表明, 非线性模型比线性模型拥有更高的评价指标, 更加适合异常检测任务. 所以, 在多数应用中普遍采用非线性模型进行异常检测. 为了提高风洞系统马赫数预测的准确性, Zhao 等人^[129]研究了基于回归模型的异常点检测框架来改进马赫数预测结果. 具体来说, 该方法使用 Bagging 技术来减小训练阶段未知异常值的影响, 然后, 引入极值分析的概念为计算出的异常值确定合适的检测阈值. 文中还采用了基于 Hampel 标识符^[130]的鲁棒缩放方法来缓解异常值对数据缩放的影响. 实验结果表明, 如果考虑关于因变量的先验知识, 基于回归模型的异常检测器通常会有更好的性能.

非参数方法的统计模型结构不是预先定义的, 而是根据给定的数据来确定. 与参数方法相比, 这类方法通常对数据做出较少的假设. 在异常检测中, 常用的非参数方法有基于直方图的方法和基于核函数的方法.

最简单的非参数统计技术是使用直方图来维护正常数据的轮廓, 这种方法也被称为基于频率或基于计数的方法. 基于直方图的方法在入侵检测和欺诈检测中广泛应用, 因为在这些场景中数据的行为可以使用直方图模型有效地捕获. 基于直方图的异常检测方法包括两个步骤: 第 1 步是基于训练数据中该特征的不同值构建直方图. 在第 2 步中, 检查测试实例是否落在直方图的任何一个矩形区域 (Bin) 中, 如果是, 则测试实例属于正常数据, 否则为异常值. 基于直方图方法的一个主要变体是根据每个测试实例落入的矩形区域的高度 (频率) 为其分配一个异常评分. 为了识别异常的网络流量行为, 提升网络服务的安全性, Kind 等人^[131]提出了一种新的基于特征和直方图的异常检测方法. 具体来说, 该方法包括以下 4 个步骤: 第 1 步, 选择特征和构建直方图. 有些流量异常可能在一个特征维度上表现不明显, 但可以借助更多特征或特征组合来识别. 第 2 步, 映射到度量空间. 将训练直方图的向量映射到一个度量空间, 两个相似的直方图在空间上很接近, 而两个不同的直方图在空间上距离很远. 第 3 步, 聚类 and 模型提取. 在训练过程中, 将相似的直方图聚类在一起. 第 4 步, 检测与分类. 在检测过程中, 对网络的运行进行监控, 并构建不同特征的向量. 然后, 将这些向量与已构建的模型进行比较, 以衡量观察到的在线行为是否存在异常. Becirovic 等人^[132]将基于直方图的异常检测方法应用到了金融领域, 主要目的是在小额信贷公司交易总账中发现异常交易. 实验表明, 在大多数情况下, 检测到的异常并非会计环境意义上的异常, 真实的异常通常是错误的密码输入、支付失败、交易取消等.

核密度估计 (kernel density estimation, KDE)^[133]是检测异常值的另一种常用的非参数方法. 由于核密度估计方法不利用有关数据分布的先验知识, 对数据分布不附加任何假定, 是一种从数据样本本身出发研究数据分布特征的方法, 因而, 在统计学理论和应用领域均受到高度的重视. 基于核密度估计的异常检测方法原理相对简单, 在已知数据概率分布的情况下, 如果测试数据实例出现在观测中, 则认为该数据实例的概率密度很大, 可以将它判断为正常数据, 否则为异常值. Latecki 等人^[134]提出了一种利用核函数进行异常值检测的无监督方法. 首先, 使用变量核对非参数密度估计进行改进, 得到一个鲁棒的局部密度估计. 然后通过比较每个点的局部密度与其相邻点的局部密度来检测异常点. 与经典的基于密度的异常检测方法相比, 该方法在大多数情况下具有更好的检测性能. 然而, 该方法在现实生活中的大型高维数据集上缺乏适用性. 为了解决这个问题, Schubert 等人^[135]分析了密度估计和异常值检测的相互作用, 通过对这两个过程进行清晰的解耦, 提出了一种基于核密度估计的异常检测算法.

KDEOS. KDEOS 首先通过 KDE 建立密度估计, 然后进行鲁棒的统计, 最后, 将基于密度估计的异常评分归一化. 上述步骤中的任何一步都可以修改, 以此增加了异常检测方法在大型数据集上的适用性和可扩展性. 为了提升智能交通系统的可靠性, 研究人员需要对交通数据中的异常值进行筛查, 针对这一问题, Xu 等人^[136]提出了一种基于核密度估计的异常点检测算法. 首先, 利用 KDE 算法得到交通数据概率密度函数的最优估计. 然后, 借助概率密度函数构建置信度函数, 对检测数据的可信度进行评估. 最后, 根据数据的可信度对其进行筛选, 从而检测出异常值. 综上所述, 大多数基于 KDE 的异常检测方法主要缺点是通常具有高计算成本和维数灾难问题, 这导致它们在实践中缺乏可靠性. 尽管与其他非参数方法相比, KDE 的异常检测性能更好, 但采用基于 KDE 的方法来解决轨迹异常检测问题的文献总体相对较少.

基于统计的轨迹异常检测方法的计算复杂度取决于用来拟合数据的统计模型的性质. 假设需要拟合指数函数族的单参数分布, 例如高斯分布、泊松分布、多项式分布等, 复杂度在数据大小和属性数量上通常是线性的. 假设拟合复杂分布, 例如混合模型、隐马尔可夫模型, 使用期望最大化方法等迭代估计技术, 算法收敛速度可能很慢. 对于基于核密度估计的方法来说, 根据数据规模大小, 可能具有二次时间复杂度, 计算成本相对较高.

基于统计的轨迹异常检测方法原理简单, 易于实现, 在数学上有合理的理论支撑, 而且可以在无监督的方式下运行, 不需要任何带标记的训练数据. 但是, 基于统计的方法依赖于数据分布模型的假设, 由于缺乏关于数据潜在分布的先验知识, 所产生的检测结果对于实际应用而言大多是不可靠的. 多数统计模型只适用于单变量特征空间, 在处理多变量数据时会产生很高的计算成本, 因此, 不适合实时性要求较高的应用场景. 另外, 基于统计的方法也不适用于处理高维数据.

4.2.6 基于图的轨迹异常检测方法

从本质上看, 物理学、生物学、社会科学和信息系统等广泛学科中的数据实例都具有内在的联系, 在异常检测时需要考虑数据集中对象的相互关系, 图为有效捕获相互依赖的数据对象之间的长期相关性提供了强大的支撑. 通过在数据对象之间引入连接边, 图自然地表示了对象之间的相互依赖, 图表示还有助于复杂数据集中对象之间相互作用的表达. 例如, 在城市交通数据集中, 交通个体类型包括车辆、行人、非机动车等, 将每个交通个体看作图中的节点, 它们之间的连线即为图中的边关系, 可以用来表达个体之间的相互作用, 由于个体类型、行驶速度、前进方向等各有差异, 所以个体间相互作用大小也不同. 但是, 通过图中的节点、边以及权重等, 可以清晰地表达复杂交通中的交互关系. 基于图的异常检测目标是根据给定的数据, 构造表达数据特征的图结构, 找出图中不同于其他数据对象的点、边、子结构等. 基于图的异常检测主要包括静态图的异常检测与动态图的异常检测. 静态图的异常检测利用频繁的子结构和子图模式来发现异常, 找出与图中观察到的模式显著偏离的节点、边、子结构. 动态图的异常检测主要是时间维度异常模式的发现, 对应于异常事件发生的时间戳或者对异常事件发生贡献最大的节点、边、子结构等. 随着图论的发展以及图数据的普及, 基于结构化图的异常检测研究越来越多, 因此, 图可以作为解决多个应用领域异常检测问题的统一框架.

大规模出租车出行记录中的异常轨迹检测因其庞大的数据量和复杂的语义而存在着巨大的技术挑战. Zhang^[137]提出了一种基于图的方法 MoNav-TT, 用于检测大规模城市交通网络中的出租车出行异常. MoNav-TT 算法分为两个阶段: 第 1 阶段为预处理, 通过计算行程特征 (上车和下车位置) 与道路网络之间的相似性, 将每次行程的上车位置和下车位置与其最近的道路段进行匹配. 如果相似性距离大于距离阈值, 则出租车行程被判断为一级异常. 第 2 阶段为最短路径的确定, 使用 MoNav 算法^[138]为每个唯一的上车和下车节点对计算最短路径. 然后将计算出的最短路径距离与司机记录的行程距离进行比较, 如果计算出的距离大于记录的距离, 则该行程被标记为二级异常. MoNav-TT 算法在纽约市上亿条出租车行程记录中进行了测试, 在检测出租车出行异常的同时还可以帮助交通部门了解纽约市不同地区之间的联系强度. 针对长序列数据中的子序列异常检测问题, Boniol 等人^[139]提出了一种适用于领域未知子序列异常检测的无监督方法 Series2Graph. 该方法基于子序列低维嵌入的图表示, 它不需要任何标记实例和无异常数据, 并且允许使用相同的模型来检测不同长度的异常. 在合成数据集和真实数据集上的实验结果表明, Series2Graph 在数据特征未知的情况下, 可以正确检测单个和重复出现的异常, 并且在准确性、通用性和执行时间方面也表现出了良好的性能.

Akoglu 等人^[140]对基于图的异常检测技术进行了全面和结构化的综述. 该综述给出了在不同设置下算法分类的一般框架, 包括: 无监督与(半)监督方法、静态图与动态图、属性图与普通图. 文中分析了这些方法的有效性、可扩展性、通用性以及鲁棒性等. 此外, 该综述还强调了异常归因的重要性, 并介绍了有助于挖掘异常根本原因的相关技术, 以便进一步分析和判断异常. Das Bhattacharjee 等人^[141]提出了一种基于图的上下文感知异常检测框架, 通过用户画像分析来识别异常用户活动. 文中提出的基于查询自适应图的优化方法充分利用用户模型之间的相似性以及它们各自与查询的相似性来筛选用户画像, 以实现更可靠的异常检测. Wang 等人^[142]受邻域方法以及利用图表示和随机游走的模型的启发, 提出了一种新的异常检测模型, 该模型基于各种随机游走过程的平稳分布的不同变化模式. 文中提出的模型通过使用自动确定的邻域大小序列, 从原始数据集构建一个加权有向邻域图的集合. 利用局部信息的不同方面对预定义的图执行马尔可夫随机游走过程, 以推导出每个对象的异常值. 本文在第 4.2.1 节中介绍了基于距离的异常检测方法, 但是当应用程序面临大型数据集时, 基于距离的方法时间复杂度较高, 计算资源开销较大. 针对上述问题, Ahmed 等人^[143]通过有向加权图分别构造了正常图与异常图结构, 并提出了一种基于图的轨迹异常检测方法, 该方法采用基于图节点的相似性度量结果来比较不同组的轨迹并判断异常. 在真实数据集的实验表明, 该方法的速度和精度明显优于基于机器学习的异常检测方法.

图具有强大的数据表示能力, 可以很自然地表达数据实例之间的相互依赖关系, 而异常检测领域的问题大多是关系型问题, 异常事件的本质可以表现为数据对象之间相互关系的变化, 所以, 基于图的方法是异常检测研究领域一个重要的分支, 可以广泛应用于多种数据类型和实际场景. 虽然基于图的异常检测方面已经出现了大量可参考的工作, 但是, 实际问题中所涉及的数据集通常体积庞大且内容丰富, 有时候还需要考虑噪声等附加信息的影响, 这可能导致图模型结构非常复杂, 所以, 算法时间复杂度也远高于传统的异常检测方法. 另外, 现有基于图的异常检测方法多数是基于静态图展开研究, 这类算法无法适用于实时性要求较高的应用. 对轨迹异常检测来说, 在未来的研究中, 可以利用动态图实时建模轨迹与周围环境的交互作用, 捕获轨迹空间和时间维度的变化, 设计检测效率更高, 可扩展性更强, 鲁棒性更好的图异常检测方法.

4.2.7 基于模式学习的轨迹异常检测方法

在早期的研究中, 基于模式学习的轨迹异常检测通常采用机器学习方法设计分类器, 分类器通过观察轨迹数据的分布, 在训练过程中不断优化分类边界, 分类器的分类性能决定了异常检测算法的准确性. 近年来, 随着深度学习技术在高维数据、序列数据、空间数据、图等数据类型方面的应用, 研究人员提出了大量基于深度学习的轨迹异常检测方法, 这类方法的主要思想是通过设计深度学习模型, 让算法学习正常轨迹和异常轨迹之间的时空特征差异来识别异常. 在各种现实应用中已经证明, 基于深度学习的异常检测方法的表现明显优于其他方法, 所以, 本节介绍的基于模式学习的轨迹异常检测主要以深度学习方法为主.

循环神经网络 (recurrent neural network, RNN) 可以直接从海量时间序列中自动学习数据的时空特征, 因此, 被广泛应用于语音识别、视频理解、轨迹预测、轨迹异常检测等时序数据挖掘任务. 基于深度学习的轨迹异常检测通常借助长短时记忆网络 (long short-term memory, LSTM) 或门控循环单元网络 (gated recurrent unit, GRU) 等 RNN 模型来完成轨迹时序建模与特征学习. 通过训练自编码器 (auto-encoder, AE)^[144], 生成对抗网络 (generative adversarial network, GAN)^[145]等生成式模型及其变体, 对输入的轨迹进行重构, 然后对比重构后的轨迹和原始输入轨迹之间的差异来检测异常轨迹. RNN 模型充分考虑了所观察时间段内交通轨迹的运动特征, 可以学习正常轨迹和异常轨迹之间的形态差异, 使用正常轨迹数据训练的模型无法很好地对异常轨迹进行重构, 因此, 会产生较大的重构误差, 通过重构误差对比即可完成轨迹异常检测.

最近, 一些研究人员针对基于深度学习的异常检测方法进行了全面的调查与研究. Chalapathy 等人^[146]回顾了基于深度学习的异常检测方法在不同领域中的应用, 并评估了这些方法的有效性. 文中根据算法所采用的基本假设, 将现有深度学习异常检测方法进行了分类, 在每个类别中概述了算法基本原理、关键假设以及相关变体, 文中还对比分析了所有算法的优势和劣势, 最后总结了现有研究中未解决的问题以及未来面临的挑战. Di Mattia 等人^[147]调研了基于 GAN 的最新的异常检测方法并评估了它们的优缺点, 在不同数据集上验证了 GAN 异常检测模型的有效性, 同时突出原始文章与文中实验之间的差异, 最后作者提供了使用 GAN 进行异常检测的完整开源工具箱,

该工具箱旨在为所有希望直接使用异常检测算法的科研人员提供一种模块化的现成解决方案. Pang 等人^[148]综述了基于深度学习的异常检测研究并进行了综合分类, 主要包括 3 个高级类别的研究进展和 11 个细粒度类别方法. 文章分别回顾了这些分类下相关方法的目标函数、基本假设、优缺点等, 最后讨论了异常检测研究中未来可能遇到的挑战以及解决这些问题的新思路. 为了确定异常检测方法共同的基本原理以及各种方法通常隐含的假设, Ruff 等人^[149]在经典的“浅层”方法和最新的“深度”方法之间建立了联系, 并展示了这种联系如何在上述两种类型的方法中交叉扩展使用. 文章进一步提供了对现有主流方法的实验评估, 并使用最近的可解释性技术丰富了这些方法. 最后, 文中概述了异常检测问题中关键的开放性挑战, 由此确定了未来异常检测研究的具体路径.

基于深度学习的轨迹异常检测通常借助 LSTM 或 GRU 等 RNN 模型来完成轨迹时序建模与特征学习. 通过训练 AE 与 GAN 等生成式模型及其变体, 对输入的轨迹进行重构, 然后对比重构后的轨迹和原始输入轨迹之间的差异来检测异常轨迹. 本节将从以 AE 或 GAN 及其变体模型为基本架构两方面, 对现有的基于深度学习的轨迹异常检测方法进行分类, 并梳理相关代表性方法的发展脉络. 图 9 展示了基于自编码器及其变体的轨迹异常检测方法发展脉络.

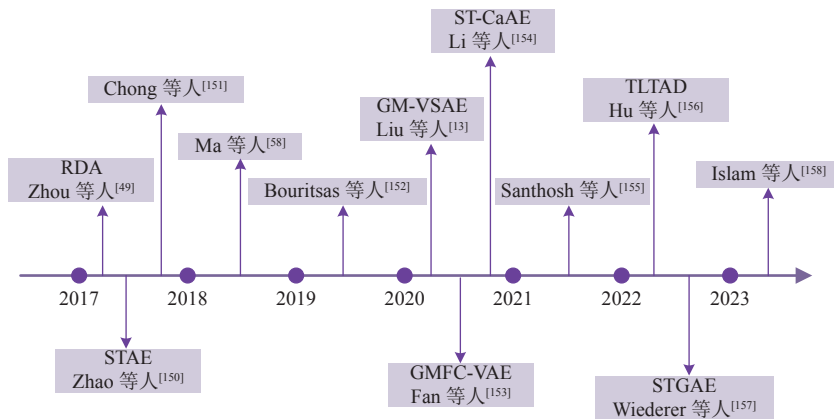


图 9 基于自编码器及其变体的轨迹异常检测方法发展脉络

自编码器模型的目标是以无监督的方式学习轨迹数据的潜在表示. 自编码器由编码器和解码器网络组成, 编码器接受原始的轨迹输入并提取固定尺寸的特征表示作为隐空间变量 (latent representation), 隐空间表示进一步作为解码器网络的输入, 解码器网络通过训练重构原始输入轨迹. 经过上述过程, 自编码器充分提取了解释轨迹数据特征的有意义的信息, 这与主成分分析等降维技术类似. 图 10 描述了自编码器的模型结构, 可以看出, 自编码器是一种具有相同输入和输出的网络模型架构.

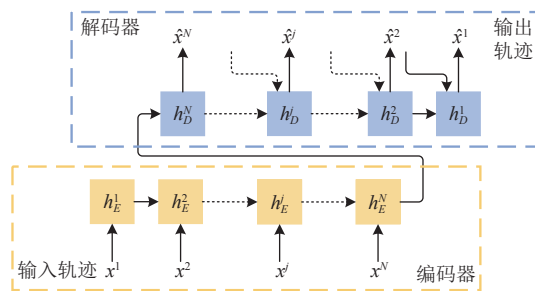


图 10 自编码器网络结构

如图 10 所示, 编码阶段负责从输入轨迹中提取特征, 编码器以原始轨迹 x 为输入, 在每个时间步计算隐藏状态 h_E^j . 在解码阶段, 轨迹 x 的隐空间表示被输入解码器完成重构与输出. 解码器接收前一个解码器的隐藏状态 h_D^{j-1} 和

上一步输出 \hat{x}^{j-1} 来计算当前时间步的隐藏状态 h_D^j 和重构输出 \hat{x}^j . 自编码器模型中编码和解码计算过程如公式 (10).

$$\begin{cases} \mathbf{h}_E = \sigma_1(\mathbf{w}_1^T \mathbf{x} + \mathbf{b}_1) \\ \hat{\mathbf{x}} = \sigma_2(\mathbf{w}_2^T \mathbf{h}_E + \mathbf{b}_2) \end{cases} \quad (10)$$

其中, \mathbf{x} 为模型输入, \mathbf{w}_1^T 和 \mathbf{w}_2^T 分别为编码器和解码器的权值矩阵, \mathbf{h}_E 表示编码器的输出, σ_1 和 σ_2 为激活函数, 用来对隐空间的输出进行非线性激活, \mathbf{b}_1 和 \mathbf{b}_2 为偏置向量, $\hat{\mathbf{x}}$ 为自编码器的输出, 自编码器模型训练的目标即最小化 $\hat{\mathbf{x}}$ 和 \mathbf{x} 之间的差异. 下面, 本文按照时间顺序对基于自编码器及其变体的轨迹异常检测方法进行描述.

自编码器模型原理简单, 易于实现, 在深度学习轨迹异常检测中得到了广泛的探索. Zhou 等人^[49]提出了一种鲁棒的自编码器模型 RDA, 该模型不仅可以保持深度自编码器发现高质量非线性特征的能力, 还可以检测异常值和噪声, 而无需借助任何“干净”的训练数据. 此外 RDA 还具有异常检测研究中容易忽略的模型去噪能力. 真实视频场景中的异常事件检测是一个具有挑战性的问题, 现有的方法大多使用局部空间区域的手工特征来识别异常. Zhao 等人^[150]提出了一种时空自编码器模型 STAE, 该模型利用深度神经网络自动学习视频表示, 并通过三维卷积操作从空间和时间维度提取运动特征. 由于大多数异常检测数据集仅包含外观异常或不自然的运动异常, 因此, 作者还从大量真实世界的交通监控视频中收集了一个新的数据集. Chong 等人^[151]提出了一种适用于拥挤场景视频异常检测的新架构, 该架构主要包括两个组件: 视频空间特征提取器和时间编码器-解码器, 分别用于表示视频空间特征和学习视频空间特征的时间演变. 文中仅使用由正常场景组成的视频数据进行端到端模型训练, 目标是使学习模型所重构的输入视频与输出视频之间的重构误差最小. 模型经过训练后可以学习正常场景的时空特征, 所以对于正常视频的重构误差较小, 而由异常场景组成的视频重构误差相对较大. 通过大量验证实验可以确定统一的异常阈值, 最终完成测试视频的异常检测以及异常定位. Ma 等人^[58]采用简单而有效的最近邻法来处理不同情况下的轨迹异常检测问题, 文中提出了一种新的轨迹相似性度量方法, 该方法基于 RNN 构造的自编码器来计算轨迹之间的距离. 自编码器用来捕获轨迹的动态特征, 轨迹之间的距离由学习模型的重构误差定义, 然后, 利用文中提出的相似性度量标准, 完成基于最近邻的轨迹异常检测. 为了评估轨迹间距离对不同异常类型的敏感性, 文中度量了 6 种异常轨迹模式与正常轨迹的相似性, 常见的异常模式包括: 轨迹平移、轨迹偏离、逆向行驶、回路行驶、等待停止、瞬间加速. 文献 [58] 另一个重要贡献是引入了 4 种真实场景交通轨迹数据集并手动进行了异常标注, 这对轨迹异常检测研究具有重要的使用价值.

Bouritsas 等人^[152]提出了一种 Seq2Seq (sequence to sequence) 架构用于实时检测行人异常轨迹. 在该方法中, 行人轨迹通过基于 LSTM 的自编码器进行表示, 编码器网络将每条轨迹样本的信息编码到隐空间, 即特征相似的轨迹在隐空间中距离较近. 解码器通过使用从隐空间采样的输入来学习生成重构轨迹. 通过最小化输入轨迹和重构轨迹之间的距离, 可以对模型进行离线训练. 接着, 预训练好的模型可以直接用于在线检测任意轨迹段中的异常. 现有的大多数轨迹异常检测算法不能处理轨迹数据的复杂性和多样性, 也不支持在线高效的异常检测, 为了解决上述挑战, Liu 等人^[13]提出了一种基于深度生成模型的轨迹异常在线检测方法, 即高斯混合变分序列自编码器 GM-VSAE. GM-VSAE 模型能够捕获包含在轨迹中的复杂顺序信息, 从轨迹中发现不同类型的正常路径, 并将它们表示在连续的隐空间中, 支持高效的在线异常检测. 由于缺乏可用于轨迹异常检测的标记数据, 为了避免手动标记异常, 文中将生成的两种异常轨迹注入数据集进行算法评估. Fan 等人^[153]提出了一种端到端的高斯混合全卷积自编码器模型 GMFC-VAE 来检测视频中的异常行为. GMFC-VAE 模型的基本思想是正常样本可以与高斯混合模型的至少一个高斯分量相关联, 而不能与任何高斯分量相关联的测试样本则被识别为异常. GMFC-VAE 编码器-解码器结构采用全卷积网络 (fully convolutional network, FCN), 以保持输入视频帧图像和输出特征映射之间的相对空间坐标. 基于各高斯混合分量的联合概率, 文中提出了一种基于样本能量的图像测试块异常评分方法. Li 等人^[154]提出了一种名为 ST-CaAE 的级联分类器用于检测监控视频中的异常行为, 该分类器包括时空对抗自编码器 ST-AAE 和时空卷积自编码器 ST-CAE 两个自编码器模型, ST-AAE 和 ST-CAE 可以充分利用来自视频数据的时间和空间信息. 此外, 文中还采用一个双流框架来融合外观和运动异常, 以获得更完整的检测结果. 智能交通系统和自动驾驶系统的通信和计算部件很容易受到黑客技术的攻击与破坏, 从而导致威胁生命的安全问题. 为此,

Ashraf 等人^[32]面向智能交通系统提出了一种基于深度学习的入侵检测系统 (intrusion detection system, IDS), 该系统设计了基于 LSTM 的自编码器模型来识别自动驾驶系统中的黑客入侵事件, 可以发现车载网络、车辆之间通信以及车辆与基础设施网络中的异常活动。

通过分析监控视频中运动对象的轨迹可以理解其行为, Santhosh 等人^[155]提出了基于卷积神经网络 (convolutional neural network, CNN) 和变分自编码器 (variational autoencoder, VAE) 的轨迹异常检测模型来识别常见的交通异常. 通过训练 CNN-VAE 混合网络模型进行轨迹分类和异常检测. 在公开的监控视频数据集上的实验结果表明, 文中设计的混合模型可以成功地识别常见的交通异常, 如车辆不遵循车道行驶、突然发生速度变化、突然停止行驶、行驶方向错误等. 海上运输船舶的异常轨迹可能会导致沉船事故, 造成巨大的经济损失和人员伤亡. Hu 等人^[156]提出了一种基于迁移学习的轨迹异常检测策略 TLTAD. 在 TLTAD 中, 基于 LSTM 的变分自编码器用于提取正常轨迹之间的潜在联系, 而图变分自编码器用于探索正常轨迹之间的空间相似性. 然后, 将迁移学习用于减少模型训练时间, 并开发了一种高效的数据集转换机制来提高模型迁移效率. 为了让自动驾驶汽车快速识别周围的异常驾驶情况以避免交通事故, Wiederer 等人^[157]提出时空图自编码器 STGAE 来学习正常的驾驶行为. 为了进行轨迹异常检测, 首先对 STGAE 学习的轨迹特征表示进行核密度估计, 然后在低密度区域检测异常. 由于缺乏用于自动驾驶中异常检测的多智能体轨迹数据集, 文中还开发了一种多智能体交通模拟器, 并基于此模拟器创建了包含正常和异常驾驶行为的数据集 MAAD, 然后在该数据集上验证了 STGAE 方法对轨迹间交互作用的建模能力以及异常检测效果. 近年来, 随着城市智能交通与监控系统的普及, 自动驾驶车辆的异常行为检测受到了越来越多的关注. 然而, 目前存在的主要挑战是: (1) 缺乏足够的轨迹数据来训练有效的异常检测模型; (2) 无法获得涵盖所有可能异常类型的轨迹数据. 虽然第 1 个挑战可以通过长时间的轨迹数据采集来解决, 但对于第 2 个挑战而言, 需要从数据采集和算法设计的角度出发, 开发更加鲁棒的异常检测模型. 为此, Islam 等人^[158]为监控视频数据中的异常检测提供了一个高效而鲁棒的框架, 并提出了一种融合 2D-CNN 和 RNN 的混合模型. 具体来说, 该混合模型使用 CNN 作为输入视频的特征提取器, 然后输入到基于 RNN 的自编码器进行序列学习和异常检测. 同时, 该模型是轻量级的, 可以确保其在边缘设备中的适用性.

自编码器和变分自编码器等基于数据重构的方法可以对不同类型的数据直接进行处理, 通用性较强, 自编码器模型的不同变体形式也可以用来进行异常检测. 然而, 在正常数据上训练的自动编码器模型不能充分重构异常数据的假设需要通过经验进行评估. 另外, 数据重构的目标函数主要用于降维或数据压缩, 而不是异常检测. 因此, 最终得到的数据表示是对潜在规律的一般总结, 这些规律并没有针对异常检测问题进行优化.

生成对抗网络 GAN 是由 Goodfellow 等人^[145]在 2014 年提出的一种生成式模型, 它可以根据从训练数据中观察到的数据分布来生成样本. 与上文介绍的自编码器模型不同, GAN 网络模型包含生成器和判别器两个互相对抗的神经网络, 从而将博弈论的思想引入到无监督学习中. 简化的 GAN 网络结构如图 11 所示, 通过训练生成器 G 来学习函数 $G: Z \rightarrow X$, 即训练数据从潜在空间 Z 到原始数据空间 X 的映射. 输入生成器的噪声 z ($z \in Z$) 基本服从均匀分布, 所以可以从 Z 进行采样以生成新的样本 $G(z)$. 相反, 训练判别器 D 的目的是区分来自 X 的真实样本和生成样本 $G(z)$. 因此, GAN 最终学习到的函数是 $D: X \rightarrow [0, 1]$, 以便判别器估计出输入样本属于训练数据给出的真实数据分布的概率. 在训练过程中, 通过最小-最大博弈 $\min_G \max_D V(D, G)$ 对生成器 G 和判别器 D 进行优化, 文献 [145] 中定义的损失函数如公式 (11).

$$\min_G \max_D V(D, G) = \mathbb{E}_{x \sim p_{\text{data}}(x)} [\log D(x)] + \mathbb{E}_{z \sim p_z(z)} [\log(1 - D(G(z)))] \quad (11)$$

GAN 训练过程中, 生成器 G 试图“欺骗”判别器 D 将生成的样本 $G(z)$ 分类为真实样本, 而判别器 D 则试图将所有样本正确分类. 理论上, 这两个网络的持续对抗训练可以产生高质量的样本, 使得判别器 D 无法区分生成样本和从原始数据分布中抽取的样本.

一般来说, GAN 具有以无监督的方式从任意领域学习数据的潜在特征的能力, 如在图像生成、图像转换、视频预测、视频生成等任务上均得到了广泛应用. 因此, GAN 也作为一种流行的深度异常检测方法, 可以用于检测多元数据中的异常模式. 基于 GAN 的轨迹异常检测方法通常旨在学习生成器网络 G 的潜在特征空间, 使该潜在

空间能够很好地捕获给定轨迹数据包含的正常属性, 然后将真实轨迹与生成轨迹之间某种形式的残差定义为异常评分. 具体来说, 在训练阶段, GAN 网络学习正常轨迹的数据分布, 即得到的网络模型只能生成或重构正常轨迹. 在测试阶段, 将随机采样的噪声向量和待检测轨迹作为输入, 然后利用训练好的 GAN 模型对待检测轨迹进行重新生成. 需要注意的是, 此时 GAN 网络中生成器的参数是固定的, 即模型只能生成落在正常数据分布的轨迹. 如果生成的待测轨迹和原始输入一样或者接近, 表明待检测轨迹为正常轨迹, 否则是异常轨迹. 图 12 展示了基于 GAN 及其变体的轨迹异常检测方法发展脉络. 下面, 本文按照时间顺序对基于 GAN 及其变体的轨迹异常检测代表性方法进行描述与分析.

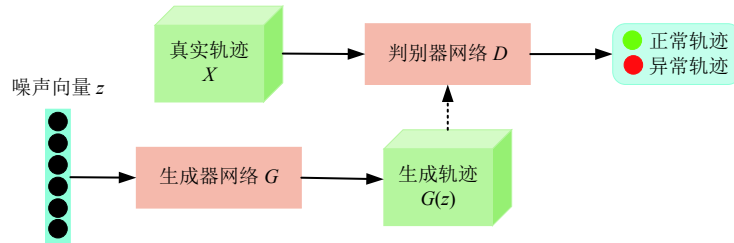


图 11 生成对抗网络结构

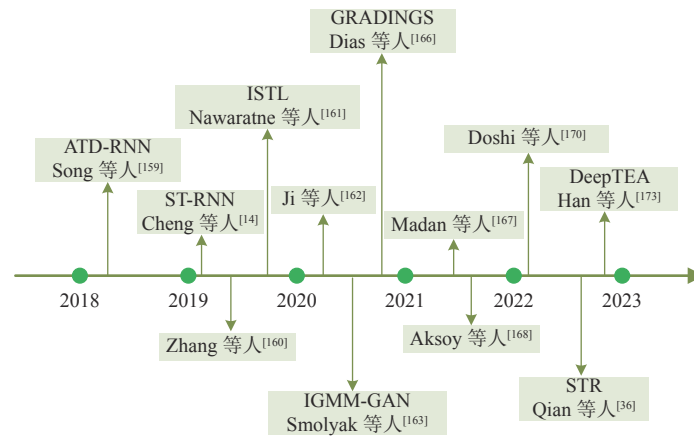


图 12 基于 GAN 及其变体的轨迹异常检测方法发展脉络

为了捕获轨迹数据的时序信息, 同时考虑异常检测过程中可能出现的数据稀疏性问题, Song 等人^[159]提出了一种基于 RNN 的异常检测方法 ATD-RNN. ATD-RNN 利用 RNN 捕获轨迹的序列信息并描述异常轨迹与正常轨迹之间的内在特征, 通过扩展大量历史轨迹来缓解潜在的数据稀疏性问题. 需要注意的是, 实验部分轨迹数据采用了聚类算法对轨迹进行分层聚类以获得标签信息, 而不是手动标记异常. 文献^[159]证明了 RNN 可以有效地对轨迹数据进行时序建模, 受此启发, Cheng 等人^[14]改进了 ATD-RNN 模型并提出了一种新的轨迹异常检测算法 ST-RNN. ST-RNN 克服了传统轨迹异常检测方法不能有效利用整条轨迹信息的问题, 而且在异常检测过程中不需要连续查询数据集, 降低了计算复杂度和空间资源消耗. ST-RNN 还引入了注意力机制, 注意力机制可以自动挖掘历史轨迹上的有价值信息, 提高轨迹表示的质量, 在一定程度上避免了 RNN 中的长期依赖问题. Zhang 等人^[160]提出了一种基于时空数据的分解方法来检测城市异常, 及时避免由大规模人群聚集或交通事故等异常事件造成的生命和财产损失. 该方法将时空特征输入全连接神经网络来估计正常城市动态, 并通过去除真实动态中的正常分量来获得异常城市动态. 在异常评分模块中, 将提取的异常分量输入通用异常检测器, 对特定时段和区域内的异常程度进行评分.

Nawaratne 等人^[161]提出了一种增量时空学习器 ISTL 来解决实时视频监控中的异常检测和定位问题. ISTL 是

一种无监督的深度学习方法,它可以随着时间的推移主动学习正常行为的时空模式.如果检测到的视频帧为异常,则将分类后的帧发送给人类观察者进行验证,在人类观察者反馈之后,标记为正常的视频帧将被用来连续训练 ISTL 模型,并更新关于正常视频的特征信息,所以,人类观察者反馈的目的是主动地向学习模型提供动态变化的正常行为. Ji 等人^[162]创新性地提出基于 LSTM 预测网络的 Seq2Seq 轨迹建模方法 SL-Modelling,以及基于时空和语义信息的异常轨迹检测方法.基于 LSTM 的特征提取模块可以直接得到不同长度正常轨迹的序列信息,用于后续异常检测.然后,引入 Hausdorff 距离^[81]和语义兴趣序列的概念,充分利用轨迹的时空和语义信息.最后,计算模型预测结果与待检测轨迹的时空和语义相似度,将两种相似度度量的加权值与阈值进行比较,以检测异常轨迹. Smolyak 等人^[163]考虑将 GAN 应用于移动数据以生成真实的车辆轨迹和速度分布.现有的基于 GAN 的异常检测算法通常无法处理多模态数据,为此,文中引入了无限高斯混合模型 IGMM^[164]以及双向 GAN 网络 BiGAN^[165],并提出了使用 IGMM 改进的 GAN 异常检测模型 IGMM-GAN,以同时生成真实的轨迹数据和检测异常.根据估计的多模态密度,IGMM-GAN 还可以进一步对异常值进行定义. Dias 等人^[166]从概率密度估计的角度来处理轨迹数据分析任务,提出了一种带归一化流的聚合异常检测方法 GRADINGS. GRADINGS 使用归一化流方法为每个轨迹段估计概率密度,预计被判定为正常的轨迹段比异常的轨迹段拥有更高的概率密度估计.然后,将这些段的概率密度聚合合成统一的轨迹异常分数.

目前,基于深度学习的监控视频异常检测方法基本利用了视频中有限的时间信息,如连续运动帧之间的光流信息等. Madan 等人^[167]考虑使用轨迹的长期依赖来进行异常检测,并提出了一种基于轨迹预测的异常检测方法.首先,通过在两个常用的数据集 Shanghai-Tech^[70]和 CUHK Avenue^[73]上运行跟踪器来获取轨迹.然后,假设通过融合基于时间的异常检测系统和基于空间的异常检测系统能够达到最优的检测效果,并在两个基于空间的异常检测方法上验证了该假设.飞行轨迹容易受到天气、空域利用率、飞行员认知等各种复杂因素的影响,为了保证航空空间的安全, Aksoy 等人^[168]提出了一种混合模型对飞行轨迹进行异常检测.该混合架构由自编码器和 GAN 两部分构成.自编码器中编码器和解码器部分均由 LSTM 实现,异常检测由 GAN 网络完成,GAN 能够捕获历史飞行轨迹的分布,在生成器部分生成时间顺序依赖的轨迹,而判别器则根据分类误差预测轨迹异常趋势. Fu 等人^[169]提出了一种基于双向 LSTM (bi-directional LSTM, Bi-LSTM) 和注意力机制的轨迹异常检测算法,该算法的核心思想是将轨迹异常点检测问题转化为有监督的二分类问题.模型整体输入是具有一定长度的轨迹点,输出为轨迹点类型,该模型能够自动学习具有运动特征的正常点与相邻异常点之间的差异. Doshi 等人^[170]提出了一种基于车辆检测和背景建模的交通视频异常检测方法.文中使用高斯混合模型首先消除场景中的移动车辆.然后,使用预训练的语义分割模型来消除停车场中的车辆和一些错误检测.利用回溯异常检测算法计算检测区域的结构相似性统计量,以确定异常发生的时间.轨迹异常检测还可以用来识别出租车司机的欺诈行为,对于出租车服务,一般可以通过驾驶时间和驾驶距离两个指标来衡量司机是否存在欺诈行为.为了提高异常检测的精度,准确识别司机的违规行为, Qian 等人^[36]提出了一种基于时空关系的在线异常出租车轨迹检测方法 STR. STR 的基本思想是给定从起点至终点的位移,当乘客行程开始时,位移应该不断增加,如果行驶时间和行驶距离不在正常范围内,则位移可能不会遵循这一趋势,因此将该轨迹识别为异常.在真实出租车轨迹数据集上的评估表明,STR 具有更低的误检率和更短的检测时间,有利于在线检测异常轨迹.

基于 GAN 的轨迹异常检测方法优点在于现有大量可用的 GAN 相关的模型和理论,而且 GAN 在生成正常轨迹方面表现出了良好的性能,这增强了模型检测从隐空间重构的异常轨迹的能力.但是,GAN 网络的训练过程往往存在收敛较慢、模式崩溃等问题,导致异常检测模型的训练难度较大.生成器网络可能会被误导,并生成正常轨迹分布外的数据实例,特别是当给定数据集的真实分布很复杂或者训练数据包含大量异常值时.另外,基于 GAN 的异常评估分数可能是次优的,因为它们建立在生成器网络上,主要目标是为了数据合成,而不是异常检测.基于模式学习的异常检测方法可以克服传统异常检测方法对轨迹分布密度和相似性度量的依赖,深度学习模型有助于解决轨迹特征提取、高维数据和非结构化数据处理等问题,可以更有效地检测大规模数据中的异常值.此外,基于模式学习的方法不需要显式地定义轨迹的正常模式和异常类型.基于模式学习的方法计算复杂度与模型结构密切

相关, 一般来说, 深度异常检测模型计算开销相对较大. 基于深度学习的异常检测技术仍处于探索阶段, 在未来研究中具有广阔的应用前景.

4.3 基于算法工作方式

根据轨迹异常检测算法的工作方式, 可以把所有的方法分为离线异常检测方法和在线异常检测方法. 第 4.1 节和第 4.2 节中早期的多数异常检测方法都是为离线学习而设计的, 在离线方法中, 所有的训练轨迹数据从算法设计开始就假设是可用的, 根据一批训练数据, 对固定的模型参数和异常阈值进行最终估计和优化, 然后反复用于其他数据集的异常轨迹检测. 相比之下, 设计用于在线学习的异常检测算法, 在观察到每个新的训练轨迹时增量地更新模型参数. 离线学习算法的计算复杂度通常比在线算法高, 这可能会限制离线算法在实际场景中的应用. 另外, 离线算法只能检测到轨迹中的异常点, 而通过在线学习的异常检测算法可以识别出引起轨迹异常的子轨迹.

4.3.1 离线轨迹异常检测方法

为了从出租车的 GPS 轨迹中发现异常的驾驶模式, 自动检测出租车欺诈行为, Zhang 等人^[35]提出了一种基于隔离的轨迹异常检测算法 iBAT. iBAT 由轨迹数据预处理和异常检测两个步骤组成. 在预处理阶段, 将城市地图划分为大小相等的网格单元, 并将所有穿越同一起点-目的地单元对的轨迹进行分组. 在异常检测阶段, iBAT 没有使用基于距离或密度的度量方法, 而是通过引入自适应的隔离森林 iForest^[171], 从大量表现相似的轨迹中找出不规则的异常实例. 通过划分轨迹集合直到所有的轨迹都被隔离完毕来生成随机树, 随机树的生成过程为隔离异常轨迹提供了更短的路径, 最终 iBAT 将以较少步骤迅速隔离的轨迹识别为异常. 同样针对出租车欺诈检测问题, Zhou 等人^[172]提出了另一种类型的异常轨迹检测方法, 专门用于检测出租车司机在未激活计价器的情况下搭载乘客的行为. 该方法的基本思想是利用出租车轨迹数据和计价器记录, 通过将轨迹中的每个点与计价器记录相匹配, 寻找未被计价器记录但与常规计价器行程相似的驾驶轨迹. 在真实出租车轨迹数据集上的实验结果表明, 该方法可以有效地检测未激活计价器的情况下出租车异常运行的次数.

4.3.2 在线轨迹异常检测方法

在线轨迹异常检测算法对轨迹数据或视频监控系统中的事件进行在线实时分析, 算法反应时较低, 能够及时完成异常行为的检测, 在线异常检测方法通常比离线方法具有更加广泛的应用. 但是当轨迹规模较大时, 在线方法可能会出现检测漏报或误报的现象.

Chen 等人^[43]在 iBAT^[35]方法的基础上进行了改进, 提出了基于隔离的在线轨迹异常检测算法 iBOAT. iBOAT 算法的目标是实时地发现异常出租车子轨迹, 并提示潜在的欺诈行为, 避免乘客在行程中走不必要的弯路. iBOAT 主要分为两个步骤, 首先, 在预处理阶段, 文中对出租车轨迹的处理方法与文献^[35]相同, 最终, 将异常轨迹检测问题转化为从具有相同起点-终点单元对的所有轨迹中寻找异常轨迹的问题, 并将少数不同于正常轨迹簇的轨迹定义为异常. 然后, 在异常检测阶段, iBOAT 算法基于隔离轨迹的思想, 利用自适应窗口策略检测异常轨迹, 即异常子轨迹与大多数轨迹隔离, 而正常轨迹将会由大量的轨迹点支撑, 轨迹的支撑点越少, 其异常程度就越高. 具体来说, iBOAT 通过维护一个最新传入的轨迹点的自适应窗口来与历史轨迹集进行比较. 当自适应窗口增加一个新的轨迹点时, 通过删除与自适应窗口中子轨迹不一致的轨迹来修剪历史轨迹集. 只要自适应窗口中子轨迹的支撑点在异常阈值以上, 该窗口中就会不断增加新的轨迹点. 如果支撑点下降到异常阈值以下, 则自适应窗口将减少到只包含最新的轨迹点. 最后, 根据异常子轨迹的长度和密度计算出一个即时异常分数, 以实时判定异常轨迹偏离正常轨迹的程度.

现有的出租车轨迹异常检测方法如 iBAT^[35]和 iBOAT^[43]本质上都是采用基于计数的策略来区分轨迹类别. 为了检测城市出租车欺诈行为, Han 等人^[173]提出了一种基于深度概率模型的在线轨迹异常检测方法 DeepTEA. DeepTEA 采用 CNN 捕获不同地点的交通多样性特征, 由于交通状况实时频繁变化, 该方法还使用 RNN 模型来建模不断变化的交通动态. 为了实时检测异常行为, 减少算法训练的时间成本, 文中提出了 DeepTEA 的快速版本, 支持在新的轨迹点到来时更新异常评分.

综合上述轨迹异常检测现有研究可以发现, 异常检测问题复杂的特性给传统的轨迹异常检测方法和目前广泛

使用的深度学习方法带来了许多检测挑战. 一些常见的挑战包括缺乏可供学习的带标签数据、异常检测召回率低、在线异常检测、抗噪声异常检测以及可解释的异常检测等. 从前文研究现状分析中得知, 近年来部分挑战性问题已经得到很好的解决, 其中也不乏一些代表性的方法. 例如, 为了在线处理轨迹数据的复杂性和多样性, 目前主流的思想是基于轨迹模式学习^[13,152,157], 通过深度神经网络有效学习轨迹数据的时空特征, 动态更新异常检测模型, 以适应不断变化的数据模式, 或者使用自适应方法来处理新的异常类型. 为了解决带标签轨迹数据匮乏的问题, 通常可以通过采集或者人工标注的方法获取异常轨迹数据集^[14,35,43,44,58], 也有部分研究借助交通仿真模拟算法生成带异常轨迹的数据集^[13,45,46], 这在很大程度上可以节省手动标注造成的人力和时间成本. 而在缺乏准确异常轨迹标签的情况下, 最有效的思路是设计无监督或弱监督^[61]的方法, 如基于聚类的方法^[74,77]或者基于生成模型的方法^[49,75,151]来完成轨迹特征学习与重构, 通过对比重构误差判断异常轨迹. 传统基于距离和密度的轨迹异常检测方法往往召回率很低, 随着机器学习^[113,115]和深度学习方法^[163,169]的不断发展和应用, 轨迹异常检测的召回率得到了显著的提升. 最后, 有的轨迹异常检测结果普遍缺乏可解释性, 为了提高异常检测的可解释性, 可以使用特征选择和特征工程技术^[40,53,77], 充分考虑车辆速度、加速度、方向变化等特征. 另外, 可视化也是提高模型方法可解释性的有力工具, 可以通过绘制实时轨迹数据分布图、热力图、时间序列图等方式实现轨迹数据可视化^[54-56], 以便更直观地理解异常检测结果.

5 轨迹数据集、异常检测评估指标、开源工具

5.1 轨迹异常检测方法综合分类

为了更直观地对比第 4 节中总结剖析的轨迹异常检测方法, 本文在表 3 中对具有代表性的算法进行了综合分类 (按照第 4 节中文献引用的先后顺序), 以帮助相关研究人员快速了解异常检测领域的研究现状.

通过表 3 可以观察到: 大多数轨迹异常检测方法在无监督模式下运行, 即使存在少量有监督或半监督方法, 数据标签也是通过人工或算法标注来获得; 在轨迹异常检测研究初期, 研究者们主要关注基于距离、密度以及统计的方法. 随着机器学习理论的不断成熟, 基于分类、聚类以及图论的方法逐渐被提出. 近几年由于深度学习技术在各种应用领域的普及, 基于轨迹模式学习的异常检测方法层出不穷; 从表 3 还可以看出, 现有方法基本以离线检测方式为主, 在未来的研究中, 可以对在线异常检测方法进行深入研究; 从交通和飞行轨迹异常检测到监控视频中的异常事件识别, 轨迹异常检测算法都发挥了重要作用.

表 3 轨迹异常检测方法分类

异常检测算法		轨迹标签可用性				研究原理与方法						工作方式		应用场景	
算法及文献出处	年份	有监督	弱监督	半监督	无监督	距离	密度	分类	聚类	统计	图	模式学习	离线		在线
SHNN-CAD ^[54]	2013	√	—	—	—	√	—	—	—	—	—	—	—	√	交通轨迹
Ma等人 ^[58]	2018	√	—	—	—	—	—	—	—	—	—	√	√	—	交通轨迹
Sultani等人 ^[60]	2018	—	√	—	—	—	—	—	—	—	—	√	√	—	监控视频
Zhong等人 ^[61]	2019	—	√	—	—	—	—	—	—	—	—	√	√	—	监控视频
RTFM ^[62]	2021	—	√	—	—	—	—	—	—	—	—	√	√	—	监控视频
Sillito等人 ^[63]	2008	—	—	√	—	—	—	—	—	—	—	√	√	—	监控视频
IA-SSLM ^[66]	2021	—	—	√	—	—	—	—	—	—	—	√	√	—	监控视频
Das等人 ^[74]	2018	—	—	—	√	—	—	—	√	—	—	—	√	—	交通轨迹
TAD-FM ^[77]	2020	—	—	—	√	—	—	—	√	—	—	—	√	—	交通轨迹
TRAOD ^[89]	2008	√	—	—	—	√	—	—	—	—	—	—	√	—	交通轨迹
RTOD ^[91]	2012	√	—	—	—	√	—	—	—	—	—	—	√	—	交通轨迹
SNN-CAD ^[55]	2011	√	—	—	—	√	—	—	—	—	—	—	√	—	交通轨迹
TPRO ^[92]	2015	√	—	—	—	√	—	—	—	—	—	—	√	—	交通轨迹

表3 轨迹异常检测方法分类(续)

异常检测算法		轨迹标签可用性				研究原理与方法						工作方式		应用场景	
算法及文献出处	年份	有监督	弱监督	半监督	无监督	距离	密度	分类	聚类	统计	图	模式学习	离线		在线
CaD ^[94]	2019	—	—	—	√	√	—	—	—	—	—	—	√	—	交通轨迹
BLOF ^[98]	2016	√	—	—	—	—	√	—	—	—	—	—	√	—	监控视频
TOP-EYE ^[99]	2010	√	—	—	—	—	√	—	—	—	—	—	√	—	交通轨迹
DBTOD ^[100]	2013	√	—	—	—	—	√	—	—	—	—	—	√	—	交通轨迹
LoTAD ^[103]	2018	√	—	—	—	—	√	—	—	—	—	—	√	—	交通轨迹
KNN-LOF ^[105]	2019	√	—	—	—	—	√	—	—	—	—	—	√	—	船舶轨迹
Li等人 ^[106]	2006	√	—	—	—	—	—	√	—	—	—	—	√	—	交通轨迹
ROAM ^[107]	2007	√	—	—	—	—	—	√	—	—	—	—	√	—	交通轨迹
TRASMIL ^[109]	2013	√	—	—	—	—	—	√	—	—	—	—	√	—	交通轨迹
Piciarelli等人 ^[113]	2006	—	—	—	√	—	—	—	√	—	—	—	—	√	监控视频
CTOD ^[114]	2009	—	—	—	√	—	—	—	√	—	—	—	√	—	交通轨迹
Lee等人 ^[115]	2007	—	—	—	√	—	—	—	√	—	—	—	√	—	交通轨迹
TPRRO ^[117]	2017	—	—	—	√	—	—	—	√	—	—	—	—	√	交通轨迹
PBOTD ^[44]	2017	—	—	—	√	—	—	—	√	—	—	—	√	—	交通轨迹
iVAT+ ^[119]	2017	—	—	—	√	—	—	—	√	—	—	—	√	—	交通轨迹
Lam等人 ^[127]	2017	√	—	—	—	—	—	—	—	√	—	—	√	—	交通轨迹
Xu等人 ^[136]	2016	√	—	—	—	—	—	—	—	√	—	—	√	—	交通轨迹
MoNav-TT ^[137]	2012	√	—	—	—	—	—	—	—	—	√	—	√	—	交通轨迹
Boniol等人 ^[139]	2020	—	—	—	√	—	—	—	—	—	√	—	√	—	交通轨迹
RDA ^[49]	2017	—	—	—	√	—	—	—	—	—	—	√	√	—	监控视频
STAE ^[150]	2017	—	—	—	√	—	—	—	—	—	—	√	√	—	监控视频
Chong等人 ^[151]	2017	—	—	—	√	—	—	—	—	—	—	√	√	—	监控视频
Bouritsas等人 ^[152]	2019	—	—	—	√	—	—	—	—	—	—	√	√	—	交通轨迹
GM-VSAE ^[13]	2020	—	—	—	√	—	—	—	—	—	—	√	—	√	交通轨迹
GMFC-VAE ^[153]	2020	—	—	—	√	—	—	—	—	—	—	√	√	—	监控视频
ST-CaAE ^[154]	2020	—	—	—	√	—	—	—	—	—	—	√	√	—	监控视频
Santhosh等人 ^[155]	2021	—	—	—	√	—	—	—	—	—	—	√	√	—	交通轨迹
TLTAD ^[156]	2022	√	—	—	—	—	—	—	—	—	—	√	√	—	船舶轨迹
STGAE ^[157]	2022	√	—	—	—	—	—	—	—	—	—	√	√	—	交通轨迹
ATD-RNN ^[159]	2018	—	—	—	√	—	—	—	—	—	—	√	√	—	交通轨迹
ST-RNN ^[14]	2019	—	—	—	√	—	—	—	—	—	—	√	√	—	交通轨迹
ISTL ^[161]	2019	—	—	—	√	—	—	—	—	—	—	√	√	√	监控视频
Ji等人 ^[162]	2020	—	—	—	√	—	—	—	—	—	—	√	√	—	飞行轨迹
IGMM-GAN ^[163]	2020	—	—	—	√	—	—	—	—	—	—	√	√	—	交通轨迹
Madan等人 ^[167]	2021	—	—	—	√	—	—	—	—	—	—	√	√	—	监控视频
Aksoy等人 ^[168]	2021	—	—	—	√	—	—	—	—	—	—	√	√	—	飞行轨迹
Fu等人 ^[169]	2020	√	—	—	√	—	—	—	—	—	—	√	√	—	交通轨迹
Doshi等人 ^[170]	2021	—	—	—	√	—	—	—	—	—	—	√	√	—	交通轨迹
STR ^[36]	2021	—	—	—	√	—	—	—	—	—	—	√	—	√	交通轨迹
iBAT ^[35]	2011	—	—	—	√	—	—	—	√	—	—	—	√	—	交通轨迹
iBOAT ^[43]	2013	—	—	—	√	—	—	—	√	—	—	—	—	√	交通轨迹
DeepTEA ^[173]	2022	√	—	—	—	—	—	—	—	—	—	√	—	√	交通轨迹

5.2 开源轨迹数据集

由于收集大规模带标签异常轨迹的难度和成本较高, 轨迹异常检测研究中最大的障碍是缺乏真实的异常轨迹数据集. 许多研究人员对真实轨迹数据集通过手动或算法来标记少量的异常轨迹^[14,35,43,44,58,159], 或者利用交通流仿真算法合成部分异常轨迹^[13,45,46], 以此来评估提出的异常检测方法. 但是, 合成数据中的异常特征可能与真实异常轨迹不同, 无法充分体现异常检测方法的性能. 所以, 在轨迹异常检测算法评估中, 通常需要同时使用真实世界采集的数据集和合成数据集. 需要注意的是, 对于真实世界的数据集, 出于隐私和安全方面的考虑, 许多数据无法公开访问. 本文表 4 总结了轨迹异常检测研究中常用的数据集^[174-214].

表 4 中, 这些数据集涵盖了以各种数据类型呈现的不同应用领域. 文中总结的数据集大致可以分为 5 类: 车辆轨迹 (汽车、出租车、卡车等)、行人轨迹、飞行器轨迹 (飞机、航拍器)、船舶轨迹、其他轨迹 (飓风、动物). 车辆轨迹数据集多数采集自各个国家的大城市及其郊区地带, 特别是人口众多的地区, 如北京、上海、洛杉矶、纽约等. 近年来, 随着自动驾驶技术的不断发展, 许多用于无人车测试的数据集 (如 Argoverse^[183]、ApolloScape^[184]) 被相继提出, 以提高自动驾驶系统的安全性. 在行人轨迹方面, 除了现有研究中常用的 UCY^[199] 和 ETH^[200] 等代表性数据, 本文还调研了有关新型冠状病毒 (COVID-19) 患者的活动轨迹数据集, 该部分数据可以为研究人员和政府部门研究疫情传播与防控提供重要的参考. 其他数据如飞行器轨迹、船舶轨迹、飓风和动物轨迹等可用的数据集相对较少. 表 4 中调研的数据集采集持续时间从小时到年不等, 采样频率通常是秒或分钟水平, 轨迹数量由 GPS 报告的轨迹数或轨迹点统计得到, 由于数据集中轨迹数目较多, 往往无法获得准确的轨迹条数, 为了方便对比, 文中以数量级的形式表示每个数据集中的轨迹数量.

表 4 开源轨迹数据集

类别	数据集名称	采集时间/地点	采集时长	轨迹条数量级	类型/异常标签
车辆	Porto ^[174]	2013–2014年/葡萄牙	1.5 y	百万	真实轨迹/无
	NGSIM ^[175]	2004年/美国	1.5 h	千万	真实轨迹/无
	NYC ^[176]	2009–2021年/美国	12 y	十亿	真实轨迹/无
	T-drive ^[177]	2008年/中国	7 d	百万	真实轨迹/无
	Greek Trucks ^[178]	2011年/希腊	33 d	千	真实轨迹/无
	highD ^[179]	2018年/德国	16.5 h	十万	真实轨迹/无
	inD ^[180]	2020年/德国	10 h	十万	真实轨迹/无
	roundD ^[181]	2020年/德国	6 h	十万	真实轨迹/无
	Mirror-Traffic ^[182]	2020年/中国	3.6 h	千	真实轨迹/无
	Argoverse ^[183]	2019年/美国	13 d	百万	真实轨迹/无
	ApolloScape ^[184]	2019年/中国	1.5 h	百万	真实轨迹/无
	Interaction ^[185]	2019年/美国	16.5 h	百万	真实轨迹/无
	Waymo Open Dataset ^[186]	2020年/美国	6.4 h	百万	真实轨迹/无
	Cityscapes ^[187]	2016年/德国	270 d	百万	真实轨迹/无
	KITTI ^[188]	2011年/德国	5 d	百万	真实轨迹/无
	nuScenes ^[189]	2019年/美国、新加坡	5.5 h	百万	真实轨迹/无
	TRAF ^[190]	2019年/印度	10 h	十万	真实轨迹/无
	Lyft Level 5 ^[191]	2019–2020年/美国	120 d	十万	真实轨迹/无
	METEOR ^[192]	2021年/印度	16 h	百万	真实轨迹/无
	D ² -city ^[193]	2019年/中国	90 d	百万	真实轨迹/无
	Shanghai ^[194]	2009–2011年/中国	2 y	百亿	真实轨迹/无
	Hangzhou ^[194]	2010年/中国	1 y	十亿	真实轨迹/无
	Beijing ^[195]	2011年/中国	90 d	千万	真实轨迹/无
	VMT ^[196]	2013年/中国	7 d	千	真实轨迹/无

表4 开源轨迹数据集(续)

类别	数据集名称	采集时间/地点	采集时长	轨迹条数量级	类型/异常标签
车辆	TRAFFIC ^[197]	2016年/中国	3 h	百	真实轨迹/有
	CROSS ^[198]	2011年/美国	30 d	万	真实轨迹/有
行人	GeoLife ^[199]	2007–2012年/中国	5 y	万	真实轨迹/无
	UCY ^[200]	2007年/以色列	20 h	万	真实轨迹/无
	ETH ^[201]	2009年/瑞士	30 h	万	真实轨迹/无
	Stanford Drone Dataset ^[202]	2016年/美国	10 h	万	真实轨迹/无
	TrajNet ^[203]	2018年/德国	15 h	万	真实轨迹/无
	Oxford Town Center ^[204]	2009年/英国	1 h	千	真实轨迹/无
	PIE ^[205]	2019年/加拿大	6 h	十万	真实轨迹/无
	JAAD ^[206]	2017年/北美、东欧	10 d	万	真实轨迹/无
	DS4C-PPP ^[207]	2020年/韩国	180 d	万	真实轨迹/无
	BDBC COVID-19 ^[208]	2020年/中国	90 d	千	真实轨迹/无
飞行器	LocaRDS ^[209]	2021年/欧洲	1 y	千万	真实轨迹/无
	ZUMAVD ^[210]	2017年/瑞士	2 h	万	真实轨迹/无
船舶	Ushant ^[211]	2019年/法国	180 d	万	真实轨迹/无
	Cargo ^[212]	2015年/地中海	30 d	百万	真实轨迹/无
其他	HURDAT2 ^[213]	1851–2020年/大西洋	170 y	万	真实轨迹/无
	Movebank ^[214]	2010–至今/全球	11 y	亿	真实轨迹/无

注: 表中y表示“年”, d表示“天”, h表示“小时”

5.3 轨迹异常检测评估指标

近年来, 关于轨迹异常检测的研究集中在引入新的检测方法以降低误检率和检测时间, 很少有研究提出更好的检测评估技术. 与传统分类问题相比, 异常检测算法的性能评估更为复杂, 轨迹异常检测本质上属于二分类任务. 为此, 机器学习研究人员已经提供了几种度量方法来评估检测算法的性能^[215]. 首先, 根据轨迹真实标签和模型预测标签的组合情况, 异常检测算法共有4种检测结果, 即: 真正例(true positive, TP)、真反例(true negative, TN)、假正例(false positive, FP)、假反例(false negative, FN), 具体分类情况如表5所示.

表5 二分类结果混淆矩阵

真实类别	预测类别	
	异常轨迹	正常轨迹
异常轨迹	TP	FN
正常轨迹	FP	TN

本文总结了常用的异常检测算法评价方法和指标, 包括精度(accuracy, ACC)、准确率(precision, P)、召回率(recall, R)、 F_1 值(F_1 -score, F_1)等. 其中, ACC 表示正确分类轨迹数占所有轨迹数的比例, P 定义为正确分类的异常轨迹数占所有预测为异常轨迹数的比例, R 为正确分类的异常轨迹数占实际异常轨迹总数的比例. 在轨迹异常检测任务中, 正常轨迹数占总数据的比例远远超过异常轨迹, 而且这类任务的目标是尽可能多地识别到异常轨迹, 所以多数研究方法重点关注召回率指标 R . 另外, 由于 F_1 是一种调和平均数, 它同时考虑了准确率 P 和召回率 R , 所以它对于样本数量不均衡的二分类数据来说是一种比精度 ACC 更好的度量方式, F_1 值越高, 表明模型分类效果越好. 上述评价指标的计算方法如公式(12)–(15).

$$ACC = \frac{TP + TN}{TP + TN + FP + FN} \quad (12)$$

$$P = \frac{TP}{TP+FP} \quad (13)$$

$$R = \frac{TP}{TP+FN} \quad (14)$$

$$F_1 = \frac{2 \cdot P \cdot R}{P+R} \quad (15)$$

根据表 5 所示的混淆矩阵, 还可以得到另外两个评估指标: 真正类率 (true positive rate, TPR), 有时候也称为灵敏度 (sensitivity), 该指标等同于召回率 R . 假正类率 (false positive rate, FPR), 表示被错误分类的正常轨迹占有正常轨迹的比例, TPR 和 FPR 的计算方法如公式 (16) 和公式 (17).

$$TPR = \frac{TP}{TP+FN} \quad (16)$$

$$FPR = \frac{FP}{FP+TN} \quad (17)$$

除了上述定量评价指标, 还可以通过绘制指标间的关系曲线来更直观地完成异常检测算法评价, 常见的关系曲线有 P-R 曲线 (precision-recall, P-R 曲线) 和 ROC 曲线, 分别如图 13(a) 和图 13(b) 所示.

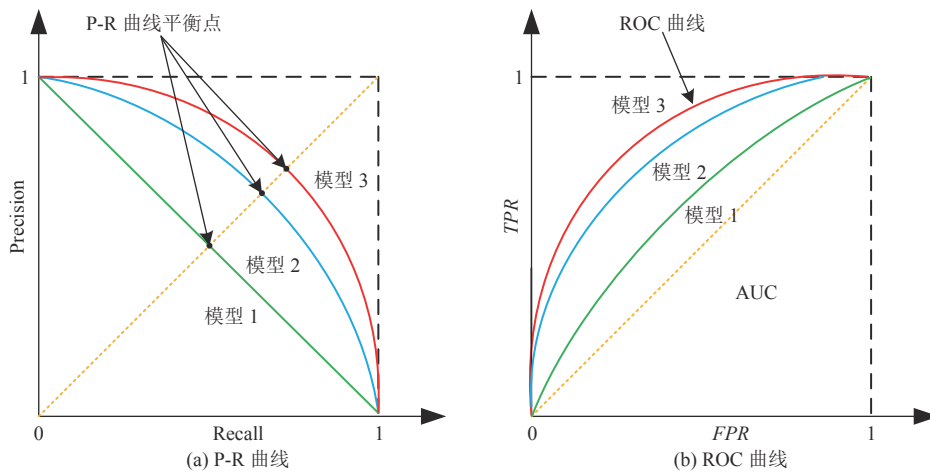


图 13 分类模型评价指标

P-R 曲线以准确率 P 作为纵坐标, 召回率 R 作为横坐标, 对于同一个模型, 通过调整分类阈值, 可以得到不同的 P-R 数值对, 从而绘制 P-R 曲线. 通过计算模型与坐标轴围成的面积大小来比较分类器的性能, 模型与坐标轴围成的面积越大, 则模型的性能越好. 当曲线之间存在交叉情况时, 曲线下面积一般不容易估算, 所以衍生出了“平衡点 (break-event point, BEP)”的概念, 即当 $P=R$ 时的分类阈值, 平衡点的取值越高, 模型性能越好, 例如在图 13(a) 中, 基于 BEP 的比较认为模型 3 的检测性能优于模型 2 和模型 1. ROC 曲线又称为接受者操作特征曲线, 曲线对应的纵坐标是 TPR , 横坐标是 FPR . ROC 曲线的绘制过程与 PR 曲线类似, 曲线上每个点都对应一个分类阈值. ROC 曲线下方的面积叫做 AUC (area under curve, AUC), AUC 的值介于 0-1 之间, AUC 作为数值可以直观地评价异常检测算法的性能, AUC 越大, 代表模型的性能越好. 例如在图 13(b) 中, 基于 AUC 的比较认为模型 3 的检测性能优于模型 2 和模型 1. 与 P-R 曲线相比, ROC 曲线相对来说更稳定, 当正负样本数量相当的时候, ROC 和 P-R 曲线的结果趋势相似, 但是当负样本数量明显大于正样本时, ROC 曲线的效果通常优于 P-R 曲线. 但对于同一模型而言, P-R 曲线和 ROC 曲线都可以说明一定的问题, 而且二者有一定的相关性, 因此, 可以借助这两种曲线综合评价.

5.4 轨迹异常检测开源工具

为了轨迹异常检测研究成果的快速应用, 并且加速异常检测算法的研究进程, 研究人员针对不同的数据类型

和应用场景, 已经开发了许多可以直接使用的异常检测工具. 本文总结了异常检测研究过程中几种常用的开源工具, 相关介绍如表 6 所示.

表 6 轨迹异常检测开源工具

工具名称	实现语言	工具简介	访问链接
PyOD ^[216]	Python	PyOD是一个用于检测多变量数据异常的Python工具包, 包括30多种异常检测模型, 已被成功应用于许多机器学习研究和商业项目	https://github.com/yzhao062/pyod
PyGOD	Python	PyGOD是一个大型的图数据异常检测仓库, 包括10余种基于图的异常检测算法, 并且支持多个层次的异常检测, 如节点、边、图的异常检测	https://github.com/pygod-team/pygod https://docs.pygod.org/en/latest
SUOD	Python	SUOD是一种用于大规模无监督异构异常检测器训练和预测的加速框架, 主要侧重于高维数据的降维、复杂模型的近似, 以及提高分布式系统中任务负载不平衡的执行效率, 同时控制异常检测性能下降	https://github.com/yzhao062/suod
DeepADoTS	Python	DeepADoTS 工具包提供了7种基于深度学习的时间序列异常检测技术的实现与评价	https://github.com/OpenSource/DeepADoTS
PySAD ^[217]	Python	PySAD是一个开源的多元数据异常检测框架, 包括多种在线异常检测模型和实验对比	https://github.com/selimfirat/pysad
TrajViz ^[218]	Java	TrajViz是一款利用GUI界面可视化轨迹模式和异常的软件, 该GUI 使用户能够以可变长度方式对轨迹中的异常模式进行交互式可视化	https://github.com/flash121123/TrajViz
Curve	Java/Python	Curve是一款用于时间序列数据异常标注的开源工具	https://github.com/baidu/Curve
TagAnomaly	R	TagAnomaly是一款为异常检测模型创建数据标签的工具, 适用于单变量和多变量数据	https://github.com/microsoft/TagAnomaly

不难看出, 目前轨迹异常检测工具库的选择还比较有限, 由于大部分异常检测算法都是无监督学习, 所以对工具包集成开发工作来说存在很大的挑战. 本文介绍的几种不同语言实现的异常检测工具包基本都是由研究人员开发的, 所以算法理论性和鲁棒性较强. 总体来看, 基于 Python 语言实现的异常检测工具库相对比较全面, Java 和 R 语言开发的工具库在轨迹可视化方面得到了广泛的应用, 另外, 也有研究人员基于 Matlab 开发了异常检测工具库.

6 轨迹异常检测未来研究趋势展望

6.1 轨迹异常检测系统架构

传感器技术和移动互联网技术的飞速发展催生了大量移动对象的轨迹数据, 这些数据蕴含了车辆、行人、动物的各种行为信息, 可以刻画移动个体和群体的时空动态性, 同时对城市道路交通、车辆驾驶行为监控、人群聚集性事件检测等应用具有重要的价值. 为了实现真实轨迹数据的利用价值, 近年来, 学术领域和工业应用领域针对轨迹数据预处理、轨迹数据存储管理、轨迹异常检测等问题开展了大量研究工作, 并形成了一套相对完备的轨迹数据挖掘流程. 图 14 描述了轨迹异常检测系统架构, 该架构由 4 个层次组成, 包括轨迹数据采集和预处理、轨迹可视化、轨迹异常检测、轨迹异常检测推广与应用. 利用日益成熟的定位感知技术, 可以获取各种类型移动个体的轨迹数据, 首先需要对采集的原始轨迹数据进行预处理, 包括轨迹清洗、轨迹分段、轨迹压缩等操作, 通过预处理过程可以减小轨迹数据冗余, 降低数据存储规模, 接着使用索引结构将轨迹数据进行高效地存储管理. 然后, 为了便于进一步研究与分析已存储的轨迹数据, 通常使用可视化工具将其直观地展现出来. 接着, 针对不同类型、不同场景的轨迹数据, 设计合适的异常检测算法, 对规范化的轨迹数据进行提取并完成异常检测. 最后, 结合可视化的异常检测结果, 分析不同场景下的轨迹数据信息和个体行为模式.

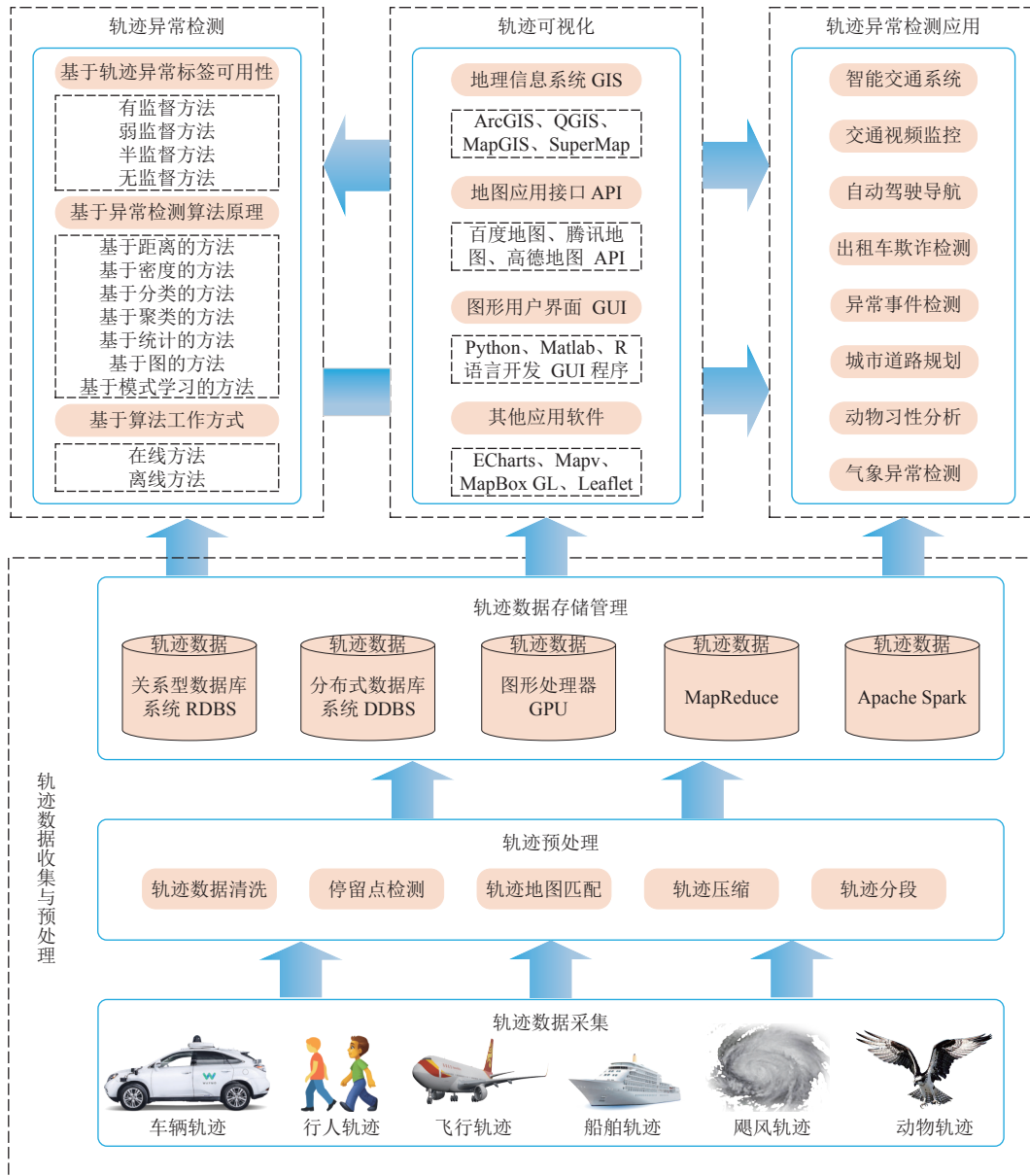


图 14 轨迹异常检测系统架构

下面简要介绍图 14 所示系统架构的内容与操作。

(1) 轨迹数据采集和预处理

通过 GPS 或移动终端设备可以实时探测车辆、行人、动物等个体的移动状态并采集局部位置信息,但是原始轨迹数据采集过程可能会受设备故障、传输信号不佳等因素影响,导致采集到的轨迹包含噪声数据或者存在位置漂移等异常,所以,在轨迹数据挖掘相关任务之前,需要进行大规模数据预处理。通常来说,轨迹数据预处理包括以下操作:数据清洗、数据分段、数据压缩、停留点检测以及地图匹配。数据清洗的目的是从轨迹上去除一些可能由定位系统的弱信号引起的噪声点和冗余点,这些点不仅会增加后续轨迹数据存储的规模,还会给检测精度带来较大影响。轨迹分段目的是根据前后轨迹点的距离和时间间隔进行轨迹段的分割,保证同一轨迹段在时空上具有连续性。轨迹压缩是在保持轨迹基本形态的同时,对轨迹关键形状点进行提取,以此实现轨迹长度压缩,同时可以减少数

据存储的开销。停留点检测通过算法识别运动物体在一定距离阈值范围内停留一段时间的位置, 停留点一般代表了轨迹途中的标志性建筑或景点等, 所以比轨迹中的其他点承载更多的语义信息。轨迹地图匹配将轨迹上的每个点投影到该点实际对应的地图路段上, 这可以帮助修正轨迹数据的精度, 地图匹配操作一般针对车辆轨迹数据。通过上述轨迹预处理操作, 实现了对原始轨迹数据的质量提升。接着, 将各种移动物体的轨迹数据按照不同的格式存储到目标轨迹数据库。随着城市交通轨迹数据量越来越庞大, 特别是在北京、纽约、东京等大城市, 传统的关系型数据库由于数据存取效率较低, 无法满足海量轨迹数据的存储管理需求。因此, 在处理大型交通数据时, 可以借助不同领域的先进技术: 高性能计算 (high performance computing, HPC)^[219]通过使用高端硬件设备或将多个独立单元的计算能力进行整合, 并利用并行架构加速解决方案, 以此来处理标准计算设备无法完成的数据密集型计算任务。目前比较流行的并行架构有图形处理器 (graphics processing unit, GPU)^[220]、超级计算机集群^[221]、MapReduce^[222]、Spark^[223]等。近年来, 学术界和工业界已经开发了高效的分布式数据库系统 (distributed database system, DDBS) 来存储管理轨迹数据, 将轨迹存储为一系列时间连续的点, 并且可以执行搜索、对比等轨迹高级查询功能。

(2) 轨迹可视化

轨迹可视化是轨迹数据挖掘系统不可缺少的部分, 通过轨迹可视化可以直观地显示轨迹数据的分布与走向, 同时可以加强异常轨迹数据的可解释性。常用的轨迹可视化方法有地理信息系统 (geographic information system, GIS)、地图应用程序编程接口 (application programming interface, API)、图形用户界面 (graphical user interface, GUI) 等。主流的 GIS 软件有 ArcGIS 和 QGIS, 通过集成开发环境可以配置各种可视化参数。百度地图、腾讯地图、高德地图等位置服务商面向用户提供的 API 能够实现在线轨迹可视化。同时, 也可以根据不同的场景需求, 基于 Python、Matlab、R 等语言自主开发 GUI 程序, 完成轨迹可视化分析。另外, 在时空数据可视化渲染方面, 也涌现出了很多优秀的应用软件, 如 ECharts、Mapv、MapBox GL、Leaflet 等。

(3) 轨迹异常检测

轨迹异常检测是从轨迹存储管理系统保存的规范化轨迹数据中找出与频繁出现的轨迹模式不同的数据, 可以是在某些相似度指标上与其他数据显著不同的轨迹, 也可以是不符合预期模式的观察事件等。轨迹异常检测方法的基本设计原则是针对特定的应用场景, 算法在时间效率和空间开销方面须要满足一定的条件。根据不同的设计原理与思想, 常见的异常检测算法有基于距离或密度度量的方法、基于轨迹分类或聚类的方法、基于轨迹特性统计的方法、基于静态图或动态图的方法以及基于轨迹模式学习的方法等。同时, 根据异常检测算法的工作方式, 通常可以设计离线方法和在线方法。为了充分发挥轨迹异常检测的效用, 在实际推广应用中, 应该设计响应速度更快、具有实时处理能力的在线方法。需要注意的是, 上述方法都可以通过有监督、半监督、无监督等方式来实现。

(4) 轨迹异常检测应用

轨迹异常检测可以用于发现道路交通事故、分析视频监控行为、预警城市聚集事件以及识别出租车欺诈行为等。另外, 轨迹异常检测在自动驾驶导航、动物习性分析、自然气象预测等方面也有广泛的应用。结合轨迹可视化结果, 将检测到的异常轨迹应用于上述领域, 进一步分析不同场景下的轨迹数据信息和个体行为模式。

6.2 轨迹异常检测研究趋势展望

轨迹异常检测研究具有重要的理论价值和实践意义, 本文对近 20 年来轨迹异常检测领域取得的研究成果进行了全面综述与分析, 从综述内容可以看出, 尽管轨迹异常检测技术取得了长足的进展, 但由于轨迹数据的独特性和复杂性, 给数据驱动的轨迹异常分析带来了基础和理论上的挑战, 在轨迹异常检测研究和应用方面仍然存在许多开放性问题有待解决。下面, 本文强调了几个关键的开放性问题, 并提出了轨迹异常检测未来研究的一些潜在方法和思想, 希望对相关研究人员提供有用的参考与指导。

(1) 大规模异常标注轨迹数据集与可视化

数据是一切研究的基础, 由于缺乏大规模带标签的异常轨迹数据, 现有轨迹异常检测工作集中在不需要任何标记数据的无监督方法, 也有部分研究人员通过手动标记或合成数据来获得带标签轨迹。然而, 异常轨迹数据在现实世界中的产生机制是极其复杂多样的, 仅通过简单的标注规则几乎不能模拟现实世界中的异常行为, 所以, 该领域需要更多具有挑战性的标注数据集来促进相关异常检测技术的发展。在设计异常检测算法的同时, 有必要收集大规模的真实异常轨迹数据, 然后结合监督或半监督等方法, 开发泛化性能较高的异常检测模型。

随着智能终端感知设备数量的急剧增加,在未来的研究中,可以充分利用 5G、云计算、边缘计算等技术,使计算处理和数据存储更靠近轨迹数据生成的位置,实时采集处理交通轨迹,实现轨迹数据的大规模收集与控制,同时还可以降低人工采集与标注成本.另外,随着高分遥感、北斗卫星、物联网等技术的不断成熟,道路交通地图由传统导航地图转变为了高精度地图.高精度地图可以更准确更全面地体现交通特征,同时记录交通轨迹的详细信息,如轨迹行进状态、轨迹周围环境、轨迹属性变化、道路属性等,所以,借助高精度地图丰富的交通元素,也可以收集大规模轨迹数据及其标注信息.另外,如何对智能终端设备产生的海量轨迹数据进行有效的收集、存储、查询、分析也是当前研究中的热点问题.随着用户对轨迹数据存储的要求越来越高,由于传统的关系型轨迹数据库表结构和数据类型过于固定,已经难以满足时空轨迹数据的存储和管理需求.而大规模时空数据库能够存储、管理时间序列以及空间地理位置相关的数据,并且可以同时处理轨迹数据的时间和空间属性,易于集成、适用场景广泛.面向轨迹挖掘与分析的时空数据库需具备高性能计算、高可靠服务、多场景支持、低存储消耗的能力,针对上述问题与需求,如何设计高效的时空数据库索引和查询优化方法、提高分布式时空计算能力、实现 GPU 加速等都是未来亟待研究的内容.

轨迹数据可视化对异常检测与其他轨迹挖掘任务都具有重要的应用价值,目前轨迹可视化方法与工具基本完成了二维空间平面内的轨迹展示,随着研究需求和可视化技术的逐步提高,在未来可以基于高精度地图开发更直观的三维轨迹可视化系统,通过多维数据展示与管理,为轨迹挖掘任务提供全面的数据分析和高效的实时交互.同时,为了高度还原轨迹行进过程,还可以借助游戏引擎渲染、虚拟现实(增强现实、混合现实)体感设备、三维重建等技术手段,使三维可视化场景更加真实逼真,全方位增强轨迹可视化效果.另外,随着计算机视觉、数字孪生等技术的发展,或许在不远的将来,元宇宙下的交通轨迹可视化平台也将成为现实.最直接的应用即自动驾驶车辆的轨迹监测与驾驶系统安全性测试.例如,如果使用真车或真人在元宇宙中的模拟形象来监测其轨迹状态,或者直接使用该模拟形象作为自动驾驶系统安全性的测试目标,这样可以更真实地反映交通个体的运动模式,甚至在轨迹发生异常的情况下,比如车辆或行人出现碰撞时,对行人的伤害程度、道路交通的拥堵影响以及车辆的损坏程度等信息都可以得到展示与反馈,并且,在元宇宙中,可视化平台体现出的车辆轨迹相关的监测和测试数据,还可以进一步指导现实生活中车辆的重新生产与改造,从而促进自动驾驶技术的发展与应用.

(2) 基于深度学习方法的轨迹异常检测

深度学习方法在捕获复杂的时间、空间依赖关系以及学习轨迹序列特征方面具有优越的性能,尽管深度学习方法在轨迹挖掘其他任务上已经成功应用并推广,然而,针对轨迹异常检测问题的深度学习方法相对较少.基于深度学习的轨迹异常检测方法主要思想是通过设计高效的深度学习模型来学习正常轨迹和异常轨迹之间的特征差异,从而完成异常判断.目前常用的模型架构是自编码器和 GAN 等生成模型,一般来说使用正常轨迹训练的模型可以有效地重构正常轨迹,而异常轨迹的重构误差较大.在轨迹挖掘领域,轨迹特征学习主要在于场景中多条轨迹之间的相互关系建模与交互作用表达,大多数方法直接借助 LSTM 或 GRU 等 RNN 模型来完成轨迹时序建模与特征学习.

由于 RNN 模型在处理数据长时间依赖关系的时候存在一定的局限性,在未来的研究中,可以借助图神经网络(graph neural network, GNN)^[224]强大的数据建模能力,将场景中的移动物体看作图的顶点,而移动物体之间的相互作用通过图中的边进行表达,这样即可完成移动物体间全局交互关系建模以及时空特征学习.为了减小全局交互建模带来的高复杂度运算,还可以考虑引入注意力机制,通过实现融合了注意力的图神经网络(graph attention network, GAT)^[225]来对轨迹间相互作用进行建模,这样不仅可以更加明确模型需要关注的轨迹特征,还可以避免大量复杂的矩阵运算.最近,Transformer^[226]网络在自然语言处理和计算机视觉领域取得了突破性的进展,Transformer 模型同样使用注意力机制来建模数据之间的时间依赖性,但是,相比传统 RNN 模型,Transformer 明显改善了时间建模效果.在未来的研究中,也可以基于 Transformer 分别设计轨迹时间和空间特征学习模型,然后利用自编码器和 GAN 等架构完成轨迹重构与异常检测.然而,现阶段基于 Transformer 的模型仅限应用于非结构化数据序列,无法处理轨迹特征提取过程中的时间-空间关系结构图,所以,如何将 Transformer 模型拓展到图等结构化数据,并直接建模轨迹间相互作用也有待进一步研究.由于收集大规模带标记异常轨迹数据的难度和成本,有监督的异常检测算法往往难以应用,因此,还可以借助对比学习的思想,通过自监督学习避免对数据集进行大量的标注.对比学习近年来已经在计算机视觉领域得到了广泛的研究与关注^[227,228],它是一种基于对比思想的判别式表示学习方

法, 通过将数据分别与正样本实例和负样本实例在特征空间进行对比来学习样本的特征表示, 主要用来做无监督的表示学习。在未来的研究中, 可以充分利用对比学习的特点, 结合自编码器等生成式模型来构建合理的正负轨迹样本, 在检测过程中更好地重构正常轨迹并抑制异常轨迹的重构, 进一步提升异常检测的效果。

另外, 与传统方法相似, 现有的深度轨迹异常检测方法主要集中在单一数据源上, 多模态轨迹异常检测是一个尚未开发的研究领域, 而传统方法难以弥补多模态信息缺失带来的影响。从理论上讲, 深度学习可以从不同类型的原始轨迹数据中学习特征表示并进行异常检测, 同时能够处理来自不同数据源的轨迹信息来学习统一的特征。因此, 深度学习为多模态轨迹异常检测提供了技术支撑。在未来的研究中, 可以借助高精度地图, 同时考虑场景中的动态视觉特征、静态视觉特征以及场景地图信息等多模态数据来构建深度轨迹异常检测模型。

(3) 在线轨迹异常检测

现有的轨迹异常检测技术大都是离线学习的方式, 即模型训练前已经假定所有的轨迹数据都是已知可用的, 通过不断训练得到最优模型参数, 然后将此模型重复作用于所有待测轨迹, 并完成异常检测。相反, 设计用于在线学习的异常检测算法会随着每次新的训练轨迹的观察而增量地更新模型参数。在线轨迹异常检测算法对轨迹数据或视频监控系统中的事件进行在线实时分析, 算法反应时较低, 能够及时完成异常行为的检测。在线异常检测方法有利于轨迹异常检测的推广与落地, 所以, 相比离线方法具有更加深远的研究意义。在未来的研究中, 首先, 可以根据大量正常的历史轨迹训练基础的异常检测模型, 然后, 在异常检测过程中实时在线更新基础模型。将待测轨迹输入到基础模型, 初步判断其是否为异常轨迹, 若模型判定为异常轨迹, 则对其进行标注, 若判断为正常轨迹, 则将基础模型进行实时更新, 并将该轨迹作为异常检测模型的训练数据, 从而使得模型持续适应轨迹数据分布的动态变化。

(4) 轨迹异常检测评估与开源工具

目前轨迹异常检测研究都致力于开发新的方法来提高检测精度, 但数据挖掘研究的一个主要挑战是如何有效评估新设计的算法, 目前最常用的评估方法大多基于机器学习二分类评价指标, 如精度、准确率、召回率等, 很少有研究提出更好的异常检测评估技术。在未来的研究中, 可以通过引入多维轨迹特征对比, 例如, 对模型检测出的异常轨迹与正常轨迹的速度、加速度、角度、轨迹点个数等特征进行对比, 同时结合二分类评价指标, 设计基于集成方法的评价规则, 从更广泛的角度对异常检测结果进行评估。需要注意的是, 大多数研究者对异常检测问题的主要关注点在于算法的有效性和执行效率, 所以, 设计评估方法时需要考虑诸如数据集大小、数据维度、参数设置等因素的影响。

在轨迹异常检测开源工具方面, 现有的异常检测工具对于输入数据的格式与规范没有统一的表示, 导致选择工具时存在一定的局限性, 使用者无法充分利用工具库中的异常检测算法。在今后的工具库开发中, 关于轨迹数据的输入格式亟待进行统一, 或者开发适配性较强的异常检测工具, 以接受不同数据源格式的轨迹。另外, 现有的异常检测工具大多数基于 CPU 实现, 因此, 无法应用于实时高并发的异常检测场景, 在未来有待进一步研究基于高性能并行计算的异常检测工具, 以满足更高的使用需求。更重要的是, 由于轨迹数据的独特性, 研究过程中不可避免地会涉及地理、地形、地图等关键信息, 为了构建安全可控的技术体系, 在轨迹异常检测工具以及可视化系统的研究与开发中, 要注重建设国产自主可控软件, 真正实现轨迹异常检测领域核心技术国产化。

(5) 可解释的轨迹异常检测

通过轨迹异常检测算法完成异常轨迹识别后, 还需要说明被检测异常的含义, 解释异常产生的原因。可解释的异常检测方法对于理解模型检测结果、减少潜在风险以及采取决策行动至关重要, 同时可解释的反馈也可以增强模型的透明度, 及时发现模型漏洞并改进。然而, 在大量的文献中, 关于轨迹异常检测的可解释性和可信性的研究相对较少, 到目前为止, 还没有任何异常检测方法或轨迹挖掘技术可以解决这个问题。随着异常检测任务和轨迹数据越来越复杂, 可解释性和可信性的问题变得越来越紧迫, 在未来的研究中, 可以通过考虑轨迹几何形状、轨迹模式、异常发生时间、异常轨迹邻居等周围干扰因素, 利用因果推理方法, 分别研究上述因素对异常检测结果的影响, 开发能够进一步揭示异常值含义的算法。一般来说, 干扰因素和异常检测结果存在一定的因果关系, 即轨迹时空特征提取决定了异常检测结果。因此, 在充分考虑相关干扰因素的情况下, 以因果关系为依据的异常轨迹判定会更加具有公平性和透明性。基于因果推理的方法可以直观地描述轨迹特征变化对异常检测结果的影响, 如果需要考虑异常检测模型中相关变量的本质意义及其在异常检测过程中的贡献, 可以借助模型可解释性工具, 例如未知模型的局部解释方法^[229]、SHAP 工具包^[230]等。

7 总结

轨迹异常检测在视频监控、智能交通、自动驾驶、航空海事、边境安全等诸多领域有着广泛的应用价值。由于轨迹数据复杂的时空特性,导致异常检测中一些常见的困难与轨迹数据的性质、异常类型、轨迹数据标签、检测准确率以及计算复杂度有关。同时,给传统的轨迹异常检测方法和最新的深度学习方法带来了诸多挑战。针对目前面临的问题与挑战,本文全面综述了近 20 年来轨迹异常检测技术的研究现状和最新进展。首先,对轨迹异常检测问题的特点与目前存在的研究挑战进行分析。然后,基于轨迹标签的可用性、异常检测算法原理、算法工作方式等分类标准,对现有轨迹异常检测算法进行了分类与剖析。对于每一类异常检测技术,本文从算法原理、复杂度、优缺点等方面进行了详细的分析,并对 50 余种具有代表性的轨迹异常检测算法进行了综合分类。接着,讨论了开源的轨迹数据集,常用的异常检测评估方法与指标以及异常检测工具。在此基础上,本文给出了轨迹异常检测系统架构,形成了从轨迹数据采集到异常检测应用等一系列相对统一的轨迹挖掘流程。最后,文中总结了轨迹异常检测领域关键的开放性问题的,并展望了未来的研究趋势和潜在思路,期待能有更多研究人员参与到轨迹异常检测的工作中,也希望本文能对轨迹异常检测领域未来的研究提供一定的参考价值。

References:

- [1] da Silva CL, Petry LM, Bogorny V. A survey and comparison of trajectory classification methods. In: Proc. of the 8th Brazilian Conf. on Intelligent Systems. Salvador: IEEE, 2019. 788–793. [doi: [10.1109/BRACIS.2019.00141](https://doi.org/10.1109/BRACIS.2019.00141)]
- [2] Gao Q, Zhang FL, Yao FM, Li AL, Mei L, Zhou F. Adversarial mobility learning for human trajectory classification. IEEE Access, 2020, 8: 20563–20576. [doi: [10.1109/ACCESS.2020.2968935](https://doi.org/10.1109/ACCESS.2020.2968935)]
- [3] Wang W, Xia F, Nie HS, Chen ZK, Gong ZG, Kong XJ, Wei W. Vehicle trajectory clustering based on dynamic representation learning of Internet of vehicles. IEEE Trans. on Intelligent Transportation Systems, 2021, 22(6): 3567–3576. [doi: [10.1109/TITS.2020.2995856](https://doi.org/10.1109/TITS.2020.2995856)]
- [4] Yang YQ, Cai JH, Yang HF, Zhang JF, Zhao XJ. TAD: A trajectory clustering algorithm based on spatial-temporal density analysis. Expert Systems with Applications, 2020, 139: 112846. [doi: [10.1016/j.eswa.2019.112846](https://doi.org/10.1016/j.eswa.2019.112846)]
- [5] Ma YX, Zhu XG, Zhang SB, Yang RG, Wang WP, Manocha D. TrajectoryPredict: Trajectory prediction for heterogeneous traffic-agents. In: Proc. of the 33rd AAAI Conf. on Artificial Intelligence. Honolulu: AAAI Press, 2019. 6120–6127. [doi: [10.1609/aaai.v33i01.33016120](https://doi.org/10.1609/aaai.v33i01.33016120)]
- [6] Sun JH, Jiang QH, Lu CW. Recursive social behavior graph for trajectory prediction. In: Proc. of the 2020 IEEE/CVF Conf. on Computer Vision and Pattern Recognition. Seattle: IEEE, 2020. 660–669. [doi: [10.1109/CVPR42600.2020.00074](https://doi.org/10.1109/CVPR42600.2020.00074)]
- [7] Cao HL, Tang HN, Wang F, Xu YJ. Survey on trajectory representation learning techniques. Ruan Jian Xue Bao/Journal of Software, 2021, 32(5): 1461–1479 (in Chinese with English abstract). <http://www.jos.org.cn/1000-9825/6210.htm> [doi: [10.13328/j.cnki.jos.006210](https://doi.org/10.13328/j.cnki.jos.006210)]
- [8] Zhang XX, Ma J, Cheng ZL, Huang SN, Ge SS, Lee TH. Trajectory generation by chance-constrained nonlinear MPC with probabilistic prediction. IEEE Trans. on Cybernetics, 2021, 51(7): 3616–3629. [doi: [10.1109/TCYB.2020.3032711](https://doi.org/10.1109/TCYB.2020.3032711)]
- [9] Amirian J, van Toll W, Hayet JB, Pettré J. Data-driven crowd simulation with generative adversarial networks. In: Proc. of the 32nd Int'l Conf. on Computer Animation and Social Agents. Paris: ACM, 2019. 7–10. [doi: [10.1145/3328756.3328769](https://doi.org/10.1145/3328756.3328769)]
- [10] Cai PD, Sun YX, Wang HL, Liu M. VTGNet: A vision-based trajectory generation network for autonomous vehicles in urban environments. IEEE Trans. on Intelligent Vehicles, 2021, 6(3): 419–429. [doi: [10.1109/TIV.2020.3033878](https://doi.org/10.1109/TIV.2020.3033878)]
- [11] Zhu J, Huang CQ, Yang M, Cheong Fung GP. Context-based prediction for road traffic state using trajectory pattern mining and recurrent convolutional neural networks. Information Sciences, 2019, 473: 190–201. [doi: [10.1016/j.ins.2018.09.029](https://doi.org/10.1016/j.ins.2018.09.029)]
- [12] Gao Q, Zhang FL, Wang RJ, Zhou F. Trajectory big data: A review of key technologies in data processing. Ruan Jian Xue Bao/Journal of Software, 2017, 28(4): 959–992 (in Chinese with English abstract). <http://www.jos.org.cn/1000-9825/5143.htm> [doi: [10.13328/j.cnki.jos.005143](https://doi.org/10.13328/j.cnki.jos.005143)]
- [13] Liu YD, Zhao KQ, Cong G, Bao ZF. Online anomalous trajectory detection with deep generative sequence modeling. In: Proc. of the 36th Int'l Conf. on Data Engineering. Dallas: IEEE, 2020. 949–960. [doi: [10.1109/ICDE48307.2020.00087](https://doi.org/10.1109/ICDE48307.2020.00087)]
- [14] Cheng YY, Wu B, Song L, Shi C. Spatial-temporal recurrent neural network for anomalous trajectories detection. In: Proc. of the 15th Int'l Conf. on Advanced Data Mining and Applications. Dalian: Springer, 2019. 565–578. [doi: [10.1007/978-3-030-35231-8_41](https://doi.org/10.1007/978-3-030-35231-8_41)]
- [15] Mao JL, Jin CQ, Zhang ZG, Zhou AY. Anomaly detection for trajectory big data: Advancements and framework. Ruan Jian Xue Bao/Journal of Software, 2017, 28(1): 17–34 (in Chinese with English abstract). <http://www.jos.org.cn/1000-9825/5151.htm> [doi: [10.13328/j.cnki.jos.005151](https://doi.org/10.13328/j.cnki.jos.005151)]
- [16] Adaramola BA, Salau AO, Adetunji FO, Fadodun OG, Ogundipe AT. Development and performance analysis of a GPS-GSM guided

- system for vehicle tracking. In: Proc. of the 2020 Int'l Conf. on Computation, Automation and Knowledge Management. Dubai: IEEE, 2020. 286–290. [doi: [10.1109/ICCAKM46823.2020.9051533](https://doi.org/10.1109/ICCAKM46823.2020.9051533)]
- [17] Remelgado R, Leutner B, Safi K, Sonnenschein R, Kuebert C, Wegmann M. Linking animal movement and remote sensing-mapping resource suitability from a remote sensing perspective. *Remote Sensing in Ecology and Conservation*, 2018, 4(3): 211–224. [doi: [10.1002/rse2.70](https://doi.org/10.1002/rse2.70)]
- [18] Wu ZH, Huang YB, Zhang Y, Zhang LF, Lei HC, Zheng HP. Precipitation characteristics of typhoon Lekima (2019) at landfall revealed by joint observations from GPM satellite and S-band radar. *Atmospheric Research*, 2021, 260: 105714. [doi: [10.1016/j.atmosres.2021.105714](https://doi.org/10.1016/j.atmosres.2021.105714)]
- [19] Chandola V, Banerjee A, Kumar V. Anomaly detection: A survey. *ACM Computing Surveys*, 2009, 41(3): 15. [doi: [10.1145/1541880.1541882](https://doi.org/10.1145/1541880.1541882)]
- [20] Ahmed M, Mahmood AN, Hu JK. A survey of network anomaly detection techniques. *Journal of Network and Computer Applications*, 2016, 60: 19–31. [doi: [10.1016/j.jnca.2015.11.016](https://doi.org/10.1016/j.jnca.2015.11.016)]
- [21] Yan XD, Xu Y, Xing XF, Cui BJ, Guo ZH, Guo TB. Trustworthy network anomaly detection based on an adaptive learning rate and momentum in IIoT. *IEEE Trans. on Industrial Informatics*, 2020, 16(9): 6182–6192. [doi: [10.1109/TII.2020.2975227](https://doi.org/10.1109/TII.2020.2975227)]
- [22] Chaudhary K, Yadav J, Mallick B. A review of fraud detection techniques: Credit card. *Int'l Journal of Computer Applications*, 2012, 45(1): 39–44.
- [23] Al-Hashedi KG, Magalingam P. Financial fraud detection applying data mining techniques: A comprehensive review from 2009 to 2019. *Computer Science Review*, 2021, 40: 100402. [doi: [10.1016/j.cosrev.2021.100402](https://doi.org/10.1016/j.cosrev.2021.100402)]
- [24] Zhou XK, Hu YY, Liang W, Ma JH, Jin Q. Variational LSTM enhanced anomaly detection for industrial big data. *IEEE Trans. on Industrial Informatics*, 2021, 17(5): 3469–3477. [doi: [10.1109/TII.2020.3022432](https://doi.org/10.1109/TII.2020.3022432)]
- [25] Meshram A, Haas C. Anomaly detection in industrial networks using machine learning: A roadmap. In: Beyerer J, Niggemann O, Kühnert C, eds. *Machine Learning for Cyber Physical Systems*. Berlin, Heidelberg: Springer, 2017. 65–72. [doi: [10.1007/978-3-662-53806-7_8](https://doi.org/10.1007/978-3-662-53806-7_8)]
- [26] Shin DH, Park RC, Chung K. Decision boundary-based anomaly detection model using improved AnoGAN from ECG data. *IEEE Access*, 2020, 8: 108664–108674. [doi: [10.1109/ACCESS.2020.3000638](https://doi.org/10.1109/ACCESS.2020.3000638)]
- [27] Baur C, Wiestler B, Albarqouni S, Navab N. Deep autoencoding models for unsupervised anomaly segmentation in brain MR images. In: Proc. of the 4th Int'l MICCAI Brainlesion Workshop. Granada: Springer, 2019. 161–169. [doi: [10.1007/978-3-030-11723-8_16](https://doi.org/10.1007/978-3-030-11723-8_16)]
- [28] Cao Y, Geddes TA, Yang JYH, Yang PY. Ensemble deep learning in bioinformatics. *Nature Machine Intelligence*, 2020, 2(9): 500–508. [doi: [10.1038/s42256-020-0217-y](https://doi.org/10.1038/s42256-020-0217-y)]
- [29] Karim R, Beyan O, Zappa A, Costa IG, Rebholz-Schuhmann D, Cochez M, Decker S. Deep learning-based clustering approaches for bioinformatics. *Briefings in Bioinformatics*, 2021, 22(1): 393–415. [doi: [10.1093/bib/bbz170](https://doi.org/10.1093/bib/bbz170)]
- [30] Santhosh KK, Dogra DP, Roy PP. Anomaly detection in road traffic using visual surveillance: A survey. *ACM Computing Surveys*, 2021, 53(6): 119. [doi: [10.1145/3417989](https://doi.org/10.1145/3417989)]
- [31] Djenouri Y, Djenouri D, Lin JCW. Trajectory outlier detection: New problems and solutions for smart cities. *ACM Trans. on Knowledge Discovery from Data*, 2021, 15(2): 20. [doi: [10.1145/3425867](https://doi.org/10.1145/3425867)]
- [32] Ashraf J, Bakhshi AD, Moustafa N, Khurshid H, Javed A, Beheshti A. Novel deep learning-enabled LSTM autoencoder architecture for discovering anomalous events from intelligent transportation systems. *IEEE Trans. on Intelligent Transportation Systems*, 2021, 22(7): 4507–4518. [doi: [10.1109/TITS.2020.3017882](https://doi.org/10.1109/TITS.2020.3017882)]
- [33] Khezerlou AV, Zhou X, Tong L, Li YH, Luo J. Forecasting gathering events through trajectory destination prediction: A dynamic hybrid model. *IEEE Trans. on Knowledge and Data Engineering*, 2021, 33(3): 991–1004. [doi: [10.1109/TKDE.2019.2937082](https://doi.org/10.1109/TKDE.2019.2937082)]
- [34] Vahedian A, Zhou X, Tong L, Li YH, Luo J. Forecasting gathering events through continuous destination prediction on big trajectory data. In: Proc. of the 25th ACM SIGSPATIAL Int'l Conf. on Advances in Geographic Information Systems. Redondo: ACM, 2017. 34. [doi: [10.1145/3139958.3140008](https://doi.org/10.1145/3139958.3140008)]
- [35] Zhang DQ, Li N, Zhou ZH, Chen C, Sun L, Li SJ. iBAT: Detecting anomalous taxi trajectories from GPS traces. In: Proc. of the 13th Int'l Conf. on Ubiquitous Computing. Beijing: ACM, 2011. 99–108. [doi: [10.1145/2030112.2030127](https://doi.org/10.1145/2030112.2030127)]
- [36] Qian SY, Cheng B, Cao J, Xue GT, Zhu YM, Yu JD, Li ML, Zhang T. Detecting taxi trajectory anomaly based on spatio-temporal relations. *IEEE Trans. on Intelligent Transportation Systems*, 2022, 23(7): 6883–6894. [doi: [10.1109/TITS.2021.3063199](https://doi.org/10.1109/TITS.2021.3063199)]
- [37] Chen SY, Wang W, van Zuylen H. A comparison of outlier detection algorithms for ITS data. *Expert Systems with Applications*, 2010, 37(2): 1169–1178. [doi: [10.1016/j.eswa.2009.06.008](https://doi.org/10.1016/j.eswa.2009.06.008)]
- [38] Zheng Y. Trajectory data mining: An overview. *ACM Trans. on Intelligent Systems and Technology*, 2015, 6(3): 29. [doi: [10.1145/](https://doi.org/10.1145/)]

- 2743025]
- [39] Bhowmick K, Narvekar M. Trajectory outlier detection for traffic events: A survey. In: Bhalla S, Bhateja V, Chandavale AA, Hiwale AA, Satapathy SS, eds. *Intelligent Computing and Information and Communication*. Singapore: Springer, 2018. 37–46. [doi: [10.1007/978-981-10-7245-1_5](https://doi.org/10.1007/978-981-10-7245-1_5)]
 - [40] Meng FR, Yuan G, Lv SQ, Wang ZX, Xia SX. An overview on trajectory outlier detection. *Artificial Intelligence Review*, 2019, 52(4): 2437–2456. [doi: [10.1007/s10462-018-9619-1](https://doi.org/10.1007/s10462-018-9619-1)]
 - [41] Zhang MY, Li T, Yu Y, Li Y, Hui P, Zheng Y. Urban anomaly analytics: Description, detection, and prediction. *IEEE Trans. on Big Data*, 2022, 8(3): 809–826. [doi: [10.1109/TBDATA.2020.2991008](https://doi.org/10.1109/TBDATA.2020.2991008)]
 - [42] Belhadi A, Djenouri Y, Lin JCW, Cano A. Trajectory outlier detection: Algorithms, taxonomies, evaluation, and open challenges. *ACM Trans. on Management Information Systems*, 2020, 11(3): 16. [doi: [10.1145/3399631](https://doi.org/10.1145/3399631)]
 - [43] Chen C, Zhang DQ, Castro PS, Li N, Sun L, Li SJ, Wang ZH. iBOAT: Isolation-based online anomalous trajectory detection. *IEEE Trans. on Intelligent Transportation Systems*, 2013, 14(2): 806–818. [doi: [10.1109/TITS.2013.2238531](https://doi.org/10.1109/TITS.2013.2238531)]
 - [44] Lv ZJ, Xu JJ, Zhao PP, Liu GF, Zhao L, Zhou XF. Outlier trajectory detection: A trajectory analytics based approach. In: *Proc. of the 22nd Int'l Conf. on Database Systems for Advanced Applications*. Suzhou: Springer, 2017. 231–246. [doi: [10.1007/978-3-319-55753-3_15](https://doi.org/10.1007/978-3-319-55753-3_15)]
 - [45] Wu H, Sun WW, Zheng BH. A fast trajectory outlier detection approach via driving behavior modeling. In: *Proc. of the 2017 ACM on Conf. on Information and Knowledge Management*. Singapore: ACM, 2017. 837–846. [doi: [10.1145/3132847.3132933](https://doi.org/10.1145/3132847.3132933)]
 - [46] Zheng GJ, Brantley SL, Lauvaux T, Li ZH. Contextual spatial outlier detection with metric learning. In: *Proc. of the 23rd ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining*. Halifax: ACM, 2017. 2161–2170. [doi: [10.1145/3097983.3098143](https://doi.org/10.1145/3097983.3098143)]
 - [47] Campos GO, Zimek A, Sander J, Campello RJGB, Micenkova B, Schubert E, Assent I, Houle ME. On the evaluation of unsupervised outlier detection: Measures, datasets, and an empirical study. *Data Mining and Knowledge Discovery*, 2016, 30(4): 891–927. [doi: [10.1007/s10618-015-0444-8](https://doi.org/10.1007/s10618-015-0444-8)]
 - [48] Pang GS, Shen CH, van den Hengel A. Deep anomaly detection with deviation networks. In: *Proc. of the 25th ACM SIGKDD Int'l Conf. on Knowledge Discovery & Data Mining*. Anchorage: ACM, 2019. 353–362. [doi: [10.1145/3292500.3330871](https://doi.org/10.1145/3292500.3330871)]
 - [49] Zhou C, Paffenroth RC. Anomaly detection with robust deep autoencoders. In: *Proc. of the 23rd ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining*. Halifax: ACM, 2017. 665–674. [doi: [10.1145/3097983.3098052](https://doi.org/10.1145/3097983.3098052)]
 - [50] Wei LY, Zheng Y, Peng WC. Constructing popular routes from uncertain trajectories. In: *Proc. of the 18th ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining*. Beijing: ACM, 2012. 195–203. [doi: [10.1145/2339530.2339562](https://doi.org/10.1145/2339530.2339562)]
 - [51] Molnar C. *Interpretable Machine Learning*. 2nd ed., Leanpub, 2022.
 - [52] Du MN, Liu NH, Hu X. Techniques for interpretable machine learning. *Communications of the ACM*, 2020, 63(1): 68–77. [doi: [10.1145/3359786](https://doi.org/10.1145/3359786)]
 - [53] Mondal S, Roy A, Mandal S. A supervised trajectory anomaly detection using velocity and path deviation. In: Bhattacharjee D, Kole DK, Dey N, Basu S, Plewczynski D, eds. *Proc. of the 2021 Int'l Conf. on Frontiers in Computing and Systems*. Singapore: Springer, 2021. 777–784. [doi: [10.1007/978-981-15-7834-2_72](https://doi.org/10.1007/978-981-15-7834-2_72)]
 - [54] Laxhammar R, Falkman G. Online learning and sequential anomaly detection in trajectories. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 2014, 36(6): 1158–1173. [doi: [10.1109/TPAMI.2013.172](https://doi.org/10.1109/TPAMI.2013.172)]
 - [55] Laxhammar R, Falkman G. Sequential conformal anomaly detection in trajectories based on hausdorff distance. In: *Proc. of the 14th Int'l Conf. on Information Fusion*. Chicago: IEEE, 2011. 1–8.
 - [56] Laxhammar R, Falkman G. Inductive conformal anomaly detection for sequential detection of anomalous sub-trajectories. *Annals of Mathematics and Artificial Intelligence*, 2015, 74(1): 67–94. [doi: [10.1007/s10472-013-9381-7](https://doi.org/10.1007/s10472-013-9381-7)]
 - [57] Vovk V, Gammernan A, Shafer G. *Algorithmic Learning in A Random World*. 2nd ed., New York: Springer, 2022. [doi: [10.1007/978-3-031-06649-8](https://doi.org/10.1007/978-3-031-06649-8)]
 - [58] Ma C, Miao ZJ, Li M, Song SY, Yang MH. Detecting anomalous trajectories via recurrent neural networks. In: *Proc. of the 14th Asian Conf. on Computer Vision*. Perth: Springer, 2019. 370–382. [doi: [10.1007/978-3-030-20870-7_23](https://doi.org/10.1007/978-3-030-20870-7_23)]
 - [59] Zhou ZH. A brief introduction to weakly supervised learning. *National Science Review*, 2018, 5(1): 44–53. [doi: [10.1093/nsr/nwx106](https://doi.org/10.1093/nsr/nwx106)]
 - [60] Sultani W, Chen C, Shah M. Real-world anomaly detection in surveillance videos. In: *Proc. of the 2018 IEEE/CVF Conf. on Computer Vision and Pattern Recognition*. Salt Lake City: IEEE, 2018. 6479–6488. [doi: [10.1109/CVPR.2018.00678](https://doi.org/10.1109/CVPR.2018.00678)]
 - [61] Zhong JX, Li NN, Kong WJ, Liu S, Li TH, Li G. Graph convolutional label noise cleaner: Train a plug-and-play action classifier for anomaly detection. In: *Proc. of the 2019 IEEE/CVF Conf. on Computer Vision and Pattern Recognition*. Long Beach: IEEE, 2019. 1237–1246. [doi: [10.1109/CVPR.2019.00133](https://doi.org/10.1109/CVPR.2019.00133)]

- [62] Tian Y, Pang GS, Chen YH, Singh R, Verjans JW, Carneiro G. Weakly-supervised video anomaly detection with robust temporal feature magnitude learning. In: Proc. of the 2021 IEEE/CVF Int'l Conf. on Computer Vision. Montreal: IEEE, 2021. 4975–4986. [doi: [10.1109/ICCV48922.2021.00493](https://doi.org/10.1109/ICCV48922.2021.00493)]
- [63] Sillito RR, Fisher RB. Semi-supervised learning for anomalous trajectory detection. In: Proc. of the 2008 British Machine Vision Conf. 2008. [doi: [10.5244/C.22.103](https://doi.org/10.5244/C.22.103)]
- [64] Huang TQ, Yu YQ, Guo GD, Qin XL. Trajectory outlier detection based on semi-supervised technology. Journal of Computer Research and Development, 2011, 48(11): 2074–2082 (in Chinese with English abstract).
- [65] Wang J, Xia LM, Hu XJ, Xiao YL. Abnormal event detection with semi-supervised sparse topic model. Neural Computing and Applications, 2019, 31(5): 1607–1617. [doi: [10.1007/s00521-018-3417-1](https://doi.org/10.1007/s00521-018-3417-1)]
- [66] Aljaloud AS, Ullah H. IA-SSLM: Irregularity-aware semi-supervised deep learning model for analyzing unusual events in crowds. IEEE Access, 2021, 9: 73327–73334. [doi: [10.1109/ACCESS.2021.3081050](https://doi.org/10.1109/ACCESS.2021.3081050)]
- [67] Hassner T, Itcher Y, Kliper-Gross O. Violent flows: Real-time detection of violent crowd behavior. In: Proc. of the 2012 IEEE Computer Society Conf. on Computer Vision and Pattern Recognition Workshops. Providence: IEEE, 2012. 1–6. [doi: [10.1109/CVPRW.2012.6239348](https://doi.org/10.1109/CVPRW.2012.6239348)]
- [68] Mehran R, Oyama A, Shah M. Abnormal crowd behavior detection using social force model. In: Proc. of the 2009 IEEE Conf. on Computer Vision and Pattern Recognition. Miami: IEEE, 2009. 935–942. [doi: [10.1109/CVPR.2009.5206641](https://doi.org/10.1109/CVPR.2009.5206641)]
- [69] University of Minnesota. Department of Computer Science and Engineering. Monitoring human activity—A project of the Artificial Intelligence, Robotics and Vision Laboratory. 2006. <http://mha.cs.umn.edu/>
- [70] Liu W, Luo WX, Lian DZ, Gao SH. Future frame prediction for anomaly detection—A new baseline. In: Proc. of the 2018 IEEE/CVF Conf. on Computer Vision and Pattern Recognition. Salt Lake City: IEEE, 2018. 6536–6545. [doi: [10.1109/CVPR.2018.00684](https://doi.org/10.1109/CVPR.2018.00684)]
- [71] Wu P, Liu J, Shi YJ, Sun YJ, Shao FT, Wu ZY, Yang ZW. Not only look, but also listen: Learning multimodal violence detection under weak supervision. In: Proc. of the 16th European Conf. on Computer Vision. Glasgow: Springer, 2020. 322–339. [doi: [10.1007/978-3-030-58577-8_20](https://doi.org/10.1007/978-3-030-58577-8_20)]
- [72] Chan AB, Vasconcelos N. Modeling, clustering, and segmenting video with mixtures of dynamic textures. IEEE Trans. on Pattern Analysis and Machine Intelligence, 2008, 30(5): 909–926. [doi: [10.1109/TPAMI.2007.70738](https://doi.org/10.1109/TPAMI.2007.70738)]
- [73] Lu CW, Shi JP, Jia JY. Abnormal event detection at 150 FPS in Matlab. In: Proc. of the 2013 IEEE Int'l Conf. on Computer Vision. Sydney: IEEE, 2013. 2720–2727. [doi: [10.1109/ICCV.2013.338](https://doi.org/10.1109/ICCV.2013.338)]
- [74] Das D, Mishra D. Unsupervised anomalous trajectory detection for crowded scenes. In: Proc. of the 13th Int'l Conf. on Industrial and Information Systems. Rupnagar: IEEE, 2018. 27–31. [doi: [10.1109/ICIINFS.2018.8721320](https://doi.org/10.1109/ICIINFS.2018.8721320)]
- [75] Zhao K, Liu B, Li WH, Yu NH, Liu ZQ. Anomaly detection and localization: A novel two-phase framework based on trajectory-level characteristics. In: Proc. of the 2018 IEEE Int'l Conf. on Multimedia & Expo Workshops. San Diego: IEEE, 2018. 1–6. [doi: [10.1109/ICMEW.2018.8551517](https://doi.org/10.1109/ICMEW.2018.8551517)]
- [76] Zhao JF, Yi ZT, Pan SY, Zhao YY, Zhao ZC, Su F, Zhuang BJ. Unsupervised traffic anomaly detection using trajectories. In: Proc. of the 2019 IEEE/CVF Conf. on Computer Vision and Pattern Recognition. Long Beach: IEEE, 2019. 133–140.
- [77] Wang ZQ, Yuan G, Pei HR, Zhang YM, Liu X. Unsupervised learning trajectory anomaly detection algorithm based on deep representation. Int'l Journal of Distributed Sensor Networks, 2020. [doi: [10.1177/1550147720971504](https://doi.org/10.1177/1550147720971504)]
- [78] Ester M, Kriegel HP, Sander J, Xu XW. A density-based algorithm for discovering clusters in large spatial databases with noise. In: Proc. of the 2nd Int'l Conf. on Knowledge Discovery and Data Mining. Portland: AAAI Press, 1996. 226–231.
- [79] Zhang Z, Huang KQ, Tan TN. Comparison of similarity measures for trajectory clustering in outdoor surveillance scenes. In: Proc. of the 18th Int'l Conf. on Pattern Recognition. Hong Kong: IEEE, 2006. 1135–1138. [doi: [10.1109/ICPR.2006.392](https://doi.org/10.1109/ICPR.2006.392)]
- [80] Berndt DJ, Clifford J. Using dynamic time warping to find patterns in time series. In: Proc. of the 3rd Int'l Conf. on Knowledge Discovery and Data Mining. Seattle: AAAI Press, 1994. 359–370.
- [81] Chen JY, Wang RD, Liu LX, Song JT. Clustering of trajectories based on Hausdorff distance. In: Proc. of the 2011 Int'l Conf. on Electronics, Communications and Control. Ningbo: IEEE, 2011. 1940–1944. [doi: [10.1109/ICECC.2011.6066483](https://doi.org/10.1109/ICECC.2011.6066483)]
- [82] Vlachos M, Kollios G, Gunopulos D. Discovering similar multidimensional trajectories. In: Proc. of the 18th Int'l Conf. on Data Engineering. San Jose: IEEE, 2002. 673–684. [doi: [10.1109/ICDE.2002.994784](https://doi.org/10.1109/ICDE.2002.994784)]
- [83] Porikli F. Trajectory distance metric using hidden Markov model based representation. In: Proc. of the 2004 IEEE European Conf. on Computer Vision, PETS Workshop. New York: IEEE, 2004.
- [84] Chen L, Özsü MT, Oria V. Robust and fast similarity search for moving object trajectories. In: Proc. of the 2005 ACM SIGMOD Int'l Conf. on Management of Data. Baltimore: ACM, 2005. 491–502. [doi: [10.1145/1066157.1066213](https://doi.org/10.1145/1066157.1066213)]

- [85] Chen L, Ng R. On the marriage of L_p-norms and edit distance. In: Proc. of the 30th Int'l Conf. on Very Large Data Bases. Toronto: VLDB Endowment Inc., 2004. 792–803.
- [86] Alt H, Godau M. Computing the Fréchet distance between two polygonal curves. *Int'l Journal of Computational Geometry & Applications*, 1995, 5(1–2): 75–91. [doi: [10.1142/S0218195995000064](https://doi.org/10.1142/S0218195995000064)]
- [87] Knorr EM, Ng RT. Algorithms for mining distance-based outliers in large datasets. In: Proc. of the 24th Int'l Conf. on Very Large Data Bases. San Francisco: Morgan Kaufmann Publishers Inc., 1998. 392–403.
- [88] Knorr EM, Ng RT, Tucakov V. Distance-based outliers: Algorithms and applications. *The VLDB Journal*, 2000, 8(3): 237–253. [doi: [10.1007/s007780050006](https://doi.org/10.1007/s007780050006)]
- [89] Lee JG, Han JW, Li XL. Trajectory outlier detection: A partition-and-detect framework. In: Proc. of the 24th IEEE Int'l Conf. on Data Engineering. Cancun: IEEE, 2008. 140–149. [doi: [10.1109/ICDE.2008.4497422](https://doi.org/10.1109/ICDE.2008.4497422)]
- [90] Liu LX, Qiao SJ, Liu B, Le JJ, Tang CJ. Efficient trajectory outlier detection algorithm based on R-tree. *Ruan Jian Xue Bao/Journal of Software*, 2009, 20(9): 2426–2435 (in Chinese with English abstract). <http://www.jos.org.cn/1000-9825/3580.htm> [doi: [10.3724/SP.J.1001.2009.03580](https://doi.org/10.3724/SP.J.1001.2009.03580)]
- [91] Liu LX, Qiao SJ, Zhang YP, Hu JS. An efficient outlying trajectories mining approach based on relative distance. *Int'l Journal of Geographical Information Science*, 2012, 26(10): 1789–1810. [doi: [10.1080/13658816.2012.654792](https://doi.org/10.1080/13658816.2012.654792)]
- [92] Zhu J, Jiang W, Liu A, Liu GF, Zhao L. Time-dependent popular routes based trajectory outlier detection. In: Proc. of the 16th Int'l Conf. on Web Information Systems Engineering. Miami: Springer, 2015. 16–30. [doi: [10.1007/978-3-319-26190-4_2](https://doi.org/10.1007/978-3-319-26190-4_2)]
- [93] Wang YL, Qin K, Chen YX, Zhao PX. Detecting anomalous trajectories and behavior patterns using hierarchical clustering from taxi GPS data. *ISPRS Int'l Journal of Geo-Information*, 2018, 7(1): 25. [doi: [10.3390/ijgi7010025](https://doi.org/10.3390/ijgi7010025)]
- [94] San Román I, de Diego IM, Conde C, Cabello E. Outlier trajectory detection through a context-aware distance. *Pattern Analysis and Applications*, 2019, 22(3): 831–839. [doi: [10.1007/s10044-018-0732-1](https://doi.org/10.1007/s10044-018-0732-1)]
- [95] Breunig MM, Kriegel HP, Ng RT, Sander J. LOF: Identifying density-based local outliers. In: Proc. of the 2000 ACM SIGMOD Int'l Conf. on Management of Data. Dallas: ACM, 2000. 93–104. [doi: [10.1145/342009.335388](https://doi.org/10.1145/342009.335388)]
- [96] Tang J, Chen ZX, Fu AWC, Cheung DW. Enhancing effectiveness of outlier detections for low density patterns. In: Proc. of the 6th Pacific-Asia Conf. on Knowledge Discovery and Data Mining. Springer, 2002. 535–548. [doi: [10.1007/3-540-47887-6_53](https://doi.org/10.1007/3-540-47887-6_53)]
- [97] Goldstein M, Uchida S. A comparative evaluation of unsupervised anomaly detection algorithms for multivariate data. *PLoS One*, 2016, 11(4): e0152173. [doi: [10.1371/journal.pone.0152173](https://doi.org/10.1371/journal.pone.0152173)]
- [98] Tang JL, Ngan HYT. Traffic outlier detection by density-based bounded local outlier factors. *Information Technology in Industry*, 2016, 4(1): 6–18. [doi: [10.17762/iti.v4i1.38](https://doi.org/10.17762/iti.v4i1.38)]
- [99] Ge Y, Xiong H, Zhou ZH, Ozdemir H, Yu J, Lee KC. TOP-EYE: Top-k evolving trajectory outlier detection. In: Proc. of the 19th ACM Int'l Conf. on Information and Knowledge Management. Toronto: ACM, 2010. 1733–1736. [doi: [10.1145/1871437.1871716](https://doi.org/10.1145/1871437.1871716)]
- [100] Liu ZP, Pi DC, Jiang JF. Density-based trajectory outlier detection algorithm. *Journal of Systems Engineering and Electronics*, 2013, 24(2): 335–340. [doi: [10.1109/JSEE.2013.00042](https://doi.org/10.1109/JSEE.2013.00042)]
- [101] Luan FJ, Zhang YT, Cao KY, Li Q. Based local density trajectory outlier detection with partition-and-detect framework. In: Proc. of the 13th Int'l Conf. on Natural Computation, Fuzzy Systems and Knowledge Discovery. Guilin: IEEE, 2017. 1708–1714. [doi: [10.1109/FSKD.2017.8393023](https://doi.org/10.1109/FSKD.2017.8393023)]
- [102] Lan DT, Yoon S. Detecting anomalous trajectories of workers using density method. *Int'l Journal of Internet, Broadcasting and Communication*, 2022, 14(2): 109–118. [doi: [10.7236/IJIBC.2022.14.2.109](https://doi.org/10.7236/IJIBC.2022.14.2.109)]
- [103] Kong XJ, Song XM, Xia F, Guo HC, Wang JZ, Tolba A. LoTAD: Long-term traffic anomaly detection based on crowdsourced bus trajectory data. *World Wide Web*, 2018, 21(3): 825–847. [doi: [10.1007/s11280-017-0487-4](https://doi.org/10.1007/s11280-017-0487-4)]
- [104] MacQueen JB. Some methods for classification and analysis of multivariate observations. In: Proc. of the 5th Berkeley Symp. on Mathematical Statistics and Probability. Berkeley: University of California Press, 1967. 281–297.
- [105] Huang YF, Zhang QR. Identification of anomaly behavior of ships based on KNN and LOF combination algorithm. *AIP Conf. Proc.*, 2019, 2073(1): 020090. [doi: [10.1063/1.5090744](https://doi.org/10.1063/1.5090744)]
- [106] Li XL, Han JW, Kim S. Motion-alert: Automatic anomaly detection in massive moving objects. In: Proc. of the 2006 Int'l Conf. on Intelligence and Security Informatics. San Diego: Springer, 2006. 166–177. [doi: [10.1007/11760146_15](https://doi.org/10.1007/11760146_15)]
- [107] Li XL, Han JW, Kim S, Gonzalez H. ROAM: Rule- and motif-based anomaly detection in massive moving object data sets. In: Proc. of the 2007 SIAM Int'l Conf. on Data Mining. SIAM, 2007. 273–284. [doi: [10.1137/1.9781611972771.25](https://doi.org/10.1137/1.9781611972771.25)]
- [108] Piciarelli C, Micheloni C, Foresti GL. Trajectory-based anomalous event detection. *IEEE Trans. on Circuits and Systems for Video Technology*, 2008, 18(11): 1544–1554. [doi: [10.1109/TCSVT.2008.2005599](https://doi.org/10.1109/TCSVT.2008.2005599)]

- [109] Yang WQ, Gao Y, Cao BL. TRASMIL: A local anomaly detection framework based on trajectory segmentation and multi-instance learning. *Computer Vision and Image Understanding*, 2013, 117(10): 1273–1286. [doi: [10.1016/j.cviu.2012.08.010](https://doi.org/10.1016/j.cviu.2012.08.010)]
- [110] Guha S, Rastogi R, Shim K. CURE: An efficient clustering algorithm for large databases. *ACM SIGMOD Record*, 1998, 27(2): 73–84. [doi: [10.1145/276305.276312](https://doi.org/10.1145/276305.276312)]
- [111] Wang W, Yang J, Muntz RR. STING: A statistical information grid approach to spatial data mining. In: *Proc. of the 23rd Int'l Conf. on Very Large Data Bases*. San Francisco: Morgan Kaufmann Publishers Inc., 1997. 186–195.
- [112] Sheikholeslami G, Chatterjee S, Zhang AD. WaveCluster: A wavelet-based clustering approach for spatial data in very large databases. *The VLDB Journal*, 2000, 8(3): 289–304. [doi: [10.1007/s007780050009](https://doi.org/10.1007/s007780050009)]
- [113] Piciarelli C, Foresti GL. On-line trajectory clustering for anomalous events detection. *Pattern Recognition Letters*, 2006, 27(15): 1835–1842. [doi: [10.1016/j.patrec.2006.02.004](https://doi.org/10.1016/j.patrec.2006.02.004)]
- [114] Ying X, Xu Z, Yin WG. Cluster-based congestion outlier detection method on trajectory data. In: *Proc. of the 6th Int'l Conf. on Fuzzy Systems and Knowledge Discovery*. Tianjin: IEEE, 2009. 243–247. [doi: [10.1109/FSKD.2009.504](https://doi.org/10.1109/FSKD.2009.504)]
- [115] Lee JG, Han JW, Whang KY. Trajectory clustering: A partition-and-group framework. In: *Proc. of the 2007 ACM SIGMOD Int'l Conf. on Management of Data*. Beijing: ACM, 2007. 593–604. [doi: [10.1145/1247480.1247546](https://doi.org/10.1145/1247480.1247546)]
- [116] Lee JG, Han JW, Li XL, Gonzalez H. TraClass: Trajectory classification using hierarchical region-based and trajectory-based clustering. *Proc. of the VLDB Endowment*, 2008, 1(1): 1081–1094. [doi: [10.14778/1453856.1453972](https://doi.org/10.14778/1453856.1453972)]
- [117] Zhu J, Jiang W, Liu AF, Liu GF, Zhao L. Effective and efficient trajectory outlier detection based on time-dependent popular route. *World Wide Web*, 2017, 20(1): 111–134. [doi: [10.1007/s11280-016-0400-6](https://doi.org/10.1007/s11280-016-0400-6)]
- [118] Park HS, Jun CH. A simple and fast algorithm for K-medoids clustering. *Expert Systems with Applications*, 2009, 36(2): 3336–3341. [doi: [10.1016/j.eswa.2008.01.039](https://doi.org/10.1016/j.eswa.2008.01.039)]
- [119] Kumar D, Bezdek JC, Rajasegarar S, Leckie C, Palaniswami M. A visual-numeric approach to clustering and anomaly detection for trajectory data. *The Visual Computer*, 2017, 33(3): 265–281. [doi: [10.1007/s00371-015-1192-x](https://doi.org/10.1007/s00371-015-1192-x)]
- [120] Han JW, Pei J, Tong HH. *Data Mining: Concepts and Techniques*. 4th ed., San Francisco: Morgan Kaufmann, 2022.
- [121] Boukerche A, Zheng LN, Alfandi O. Outlier detection: Methods, models, and classification. *ACM Computing Surveys*, 2021, 53(3): 55. [doi: [10.1145/3381028](https://doi.org/10.1145/3381028)]
- [122] Aitkin M, Wilson GT. Mixture models, outliers, and the EM algorithm. *Technometrics*, 1980, 22(3): 325–331. [doi: [10.1080/00401706.1980.10486163](https://doi.org/10.1080/00401706.1980.10486163)]
- [123] Schoenfeld D. Partial residuals for the proportional hazards regression model. *Biometrika*, 1982, 69(1): 239–241. [doi: [10.1093/biomet/69.1.239](https://doi.org/10.1093/biomet/69.1.239)]
- [124] Firth D. Bias reduction of maximum likelihood estimates. *Biometrika*, 1993, 80(1): 27–38. [doi: [10.1093/biomet/80.1.27](https://doi.org/10.1093/biomet/80.1.27)]
- [125] Yang XW, Latecki LJ, Pokrajac D. Outlier detection with globally optimal exemplar-based GMM. In: *Proc. of the 2009 SIAM Int'l Conf. on Data Mining*. SIAM, 2009. 145–154. [doi: [10.1137/1.9781611972795.13](https://doi.org/10.1137/1.9781611972795.13)]
- [126] Tang XM, Yuan RX, Chen J. Outlier detection in energy disaggregation using subspace learning and Gaussian mixture model. *Int'l Journal of Control and Automation*, 2015, 8(8): 161–170. [doi: [10.14257/ijca.2015.8.8.17](https://doi.org/10.14257/ijca.2015.8.8.17)]
- [127] Lam P, Wang LL, Ngan HYT, Yung NHC, Yeh AGO. Outlier detection in large-scale traffic data by naïve Bayes method and Gaussian mixture model method. In: *Proc. of the 2017 IS&T Int'l Symp. on Electronic Imaging: Intelligent Robotics and Industrial Applications using Computer Vision*. 2017. 73–78. [doi: [10.2352/ISSN.2470-1173.2017.9.IRIACV-272](https://doi.org/10.2352/ISSN.2470-1173.2017.9.IRIACV-272)]
- [128] Dalatu PI, Fitrianto A, Mustapha A. A comparative study of linear and nonlinear regression models for outlier detection. In: *Proc. of the 2nd Int'l Conf. on Soft Computing and Data Mining*. Bandung: Springer, 2017. 316–326. [doi: [10.1007/978-3-319-51281-5_32](https://doi.org/10.1007/978-3-319-51281-5_32)]
- [129] Zhao HY, Yu D, Wang Y, Wang B. Enhancing the prediction of mach number in wind tunnel with a regression-based outlier detection framework. *IEEE Access*, 2021, 9: 27668–27677. [doi: [10.1109/ACCESS.2021.3058096](https://doi.org/10.1109/ACCESS.2021.3058096)]
- [130] Pearson RK. Outliers in process modeling and identification. *IEEE Trans. on Control Systems Technology*, 2002, 10(1): 55–63. [doi: [10.1109/87.974338](https://doi.org/10.1109/87.974338)]
- [131] Kind A, Stoecklin MP, Dimitropoulos X. Histogram-based traffic anomaly detection. *IEEE Trans. on Network and Service Management*, 2009, 6(2): 110–121. [doi: [10.1109/TNSM.2009.090604](https://doi.org/10.1109/TNSM.2009.090604)]
- [132] Becirovic S, Zunic E, Donko D. A case study of cluster-based and histogram-based multivariate anomaly detection approach in general ledgers. In: *Proc. of the 19th Int'l Symp. INFOTEH-JAHORINA*. East Sarajevo: IEEE, 2020. 1–6. [doi: [10.1109/INFOTEH48170.2020.9066333](https://doi.org/10.1109/INFOTEH48170.2020.9066333)]
- [133] Pavlidou M, Zioutas G. Kernel density outlier detector. In: Akritas MG, Lahiri SN, Politis DN, eds. *Topics in Nonparametric Statistics*. New York: Springer, 2014. 241–250. [doi: [10.1007/978-1-4939-0569-0_22](https://doi.org/10.1007/978-1-4939-0569-0_22)]

- [134] Latecki LJ, Lazarevic A, Pokrajac D. Outlier detection with kernel density functions. In: Proc. of the 5th Int'l Workshop on Machine Learning and Data Mining in Pattern Recognition. Leipzig: Springer, 2007. 61–75. [doi: [10.1007/978-3-540-73499-4_6](https://doi.org/10.1007/978-3-540-73499-4_6)]
- [135] Schubert E, Zimek A, Kriegel HP. Generalized outlier detection with flexible kernel density estimates. In: Proc. of the 2014 SIAM Int'l Conf. on Data Mining. SIAM, 2014. 542–550. [doi: [10.1137/1.9781611973440.63](https://doi.org/10.1137/1.9781611973440.63)]
- [136] Xu YW, Xu NB, Feng XX. A new outlier detection algorithm based on kernel density estimation for ITS. In: Proc. of the 2016 IEEE Int'l Conf. on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData). Chengdu: IEEE, 2016. 258–262. [doi: [10.1109/iThings-GreenCom-CPSCom-SmartData.2016.67](https://doi.org/10.1109/iThings-GreenCom-CPSCom-SmartData.2016.67)]
- [137] Zhang JT. Smarter outlier detection and deeper understanding of large-scale taxi trip records: A case study of NYC. In: Proc. of the 2012 ACM SIGKDD Int'l Workshop on Urban Computing. Beijing: ACM, 2012. 157–162. [doi: [10.1145/2346496.2346521](https://doi.org/10.1145/2346496.2346521)]
- [138] Batz GV, Geisberger R, Sanders P, Vetter C. Minimum time-dependent travel times with contraction hierarchies. ACM Journal of Experimental Algorithmics, 2013, 18: 1.4. [doi: [10.1145/2444016.2444020](https://doi.org/10.1145/2444016.2444020)]
- [139] Boniol P, Palpanas T. Series2Graph: Graph-based subsequence anomaly detection for time series. Proc. of the VLDB Endowment, 2020, 13(12): 1821–1834. [doi: [10.14778/3407790.3407792](https://doi.org/10.14778/3407790.3407792)]
- [140] Akoglu L, Tong HH, Koutra D. Graph based anomaly detection and description: A survey. Data Mining and Knowledge Discovery, 2015, 29(3): 626–688. [doi: [10.1007/s10618-014-0365-y](https://doi.org/10.1007/s10618-014-0365-y)]
- [141] Das Bhattacharjee S, Yuan JS, Zhang JQ, Tan YP. Context-aware graph-based analysis for detecting anomalous activities. In: Proc. of the 2017 IEEE Int'l Conf. on Multimedia and Expo. Hong Kong: IEEE, 2017. 1021–1026. [doi: [10.1109/ICME.2017.8019421](https://doi.org/10.1109/ICME.2017.8019421)]
- [142] Wang C, Gao H, Liu Z, Fu Y. Outlier detection using diverse neighborhood graphs. In: Proc. of the 15th Int'l Computer Conf. on Wavelet Active Media Technology and Information Processing. Chengdu: IEEE, 2018. 58–62. [doi: [10.1109/ICCWAMTIP.2018.8632604](https://doi.org/10.1109/ICCWAMTIP.2018.8632604)]
- [143] Ahmed U, Srivastava G, Djenouri Y, Lin JCW. Knowledge graph based trajectory outlier detection in sustainable smart cities. Sustainable Cities and Society, 2022, 78: 103580. [doi: [10.1016/j.scs.2021.103580](https://doi.org/10.1016/j.scs.2021.103580)]
- [144] Hinton GE, Zemel RS. Autoencoders, minimum description length and Helmholtz free energy. In: Proc. of the 6th Int'l Conf. on Neural Information Processing Systems. Denver: Morgan Kaufmann Publishers Inc., 1993. 3–10.
- [145] Goodfellow IJ, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, Courville A, Bengio Y. Generative adversarial nets. In: Proc. of the 27th Int'l Conf. on Neural Information Processing Systems. Montreal: MIT Press, 2014. 2672–2680.
- [146] Chalapathy R, Chawla S. Deep learning for anomaly detection: A survey. arXiv:1901.03407, 2019.
- [147] Di Mattia F, Galeone P, De Simoni M, Ghelfi E. A survey on GANs for anomaly detection. arXiv:1906.11632, 2019.
- [148] Pang GS, Shen CH, Cao LB, Van Den Hengel A. Deep learning for anomaly detection: A review. ACM Computing Surveys, 2022, 54(2): 38. [doi: [10.1145/3439950](https://doi.org/10.1145/3439950)]
- [149] Ruff L, Kauffmann JR, Vandermeulen RA, Montavon G, Samek W, Kloft M, Dietterich TG, Muller KR. A unifying review of deep and shallow anomaly detection. Proc. of the IEEE, 2021, 109(5): 756–795. [doi: [10.1109/JPROC.2021.3052449](https://doi.org/10.1109/JPROC.2021.3052449)]
- [150] Zhao YR, Deng B, Shen C, Liu Y, Lu HT, Hua XS. Spatio-temporal autoencoder for video anomaly detection. In: Proc. of the 25th ACM Int'l Conf. on Multimedia. Mountain: ACM, 2017. 1933–1941. [doi: [10.1145/3123266.3123451](https://doi.org/10.1145/3123266.3123451)]
- [151] Chong YS, Tay YH. Abnormal event detection in videos using spatiotemporal autoencoder. In: Proc. of the 14th Int'l Symp. on Neural Networks. Hokkaido: Springer, 2017. 189–196. [doi: [10.1007/978-3-319-59081-3_23](https://doi.org/10.1007/978-3-319-59081-3_23)]
- [152] Bouritsas G, Daveas S, Danelakis A, Thomopoulos SCA. Automated real-time anomaly detection in human trajectories using sequence to sequence networks. In: Proc. of the 16th IEEE Int'l Conf. on Advanced Video and Signal Based Surveillance. IEEE, 2019. 1–8. [doi: [10.1109/AVSS.2019.8909844](https://doi.org/10.1109/AVSS.2019.8909844)]
- [153] Fan YX, Wen GJ, Li DR, Qiu SH, Levine MD, Xiao F. Video anomaly detection and localization via Gaussian mixture fully convolutional variational autoencoder. Computer Vision and Image Understanding, 2020, 195: 102920. [doi: [10.1016/j.cviu.2020.102920](https://doi.org/10.1016/j.cviu.2020.102920)]
- [154] Li NJ, Chang FL, Liu CS. Spatial-temporal cascade autoencoder for video anomaly detection in crowded scenes. IEEE Trans. on Multimedia, 2021, 23: 203–215. [doi: [10.1109/TMM.2020.2984093](https://doi.org/10.1109/TMM.2020.2984093)]
- [155] Santhosh KK, Dogra DP, Roy PP, Mitra A. Vehicular trajectory classification and traffic anomaly detection in videos using a hybrid CNN-VAE Architecture. IEEE Trans. on Intelligent Transportation Systems, 2022, 23(8): 11891–11902. [doi: [10.1109/TITS.2021.3108504](https://doi.org/10.1109/TITS.2021.3108504)]
- [156] Hu J, Kaur K, Lin H, Wang XD, Hassan MM, Razzak I, Hammoudeh M. Intelligent anomaly detection of trajectories for IoT empowered maritime transportation systems. IEEE Trans. on Intelligent Transportation Systems, 2023, 24(2): 2382–2391. [doi: [10.1109/TITS.2022.3162491](https://doi.org/10.1109/TITS.2022.3162491)]
- [157] Wiederer J, Bouazizi A, Troina M, Kressel U, Belagiannis V. Anomaly detection in multi-agent trajectories for automated driving. In:

- Proc. of the 5th Conf. on Robot Learning. London: PMLR, 2022. 1223–1233.
- [158] Islam M, Dukyil AS, Alyahya S, Habib S. An IoT enable anomaly detection system for smart city surveillance. *Sensors*, 2023, 23(4): 2358. [doi: [10.3390/s23042358](https://doi.org/10.3390/s23042358)]
- [159] Song L, Wang RJ, Xiao D, Han XT, Cai YN, Shi C. Anomalous trajectory detection using recurrent neural network. In: Proc. of the 14th Int'l Conf. on Advanced Data Mining and Applications. Nanjing: Springer, 2018. 263–277. [doi: [10.1007/978-3-030-05090-0_23](https://doi.org/10.1007/978-3-030-05090-0_23)]
- [160] Zhang MY, Li T, Shi HZ, Li Y, Hui P. A decomposition approach for urban anomaly detection across spatiotemporal data. In: Proc. of the 28th Int'l Joint Conf. on Artificial Intelligence. Macao: IJCAI.org, 2019. 6043–6049. [doi: [10.24963/ijcai.2019/837](https://doi.org/10.24963/ijcai.2019/837)]
- [161] Nawaratne R, Alahakoon D, De Silva D, Yu XH. Spatiotemporal anomaly detection using deep learning for real-time video surveillance. *IEEE Trans. on Industrial Informatics*, 2020, 16(1): 393–402. [doi: [10.1109/TII.2019.2938527](https://doi.org/10.1109/TII.2019.2938527)]
- [162] Ji YF, Wang LW, Wu WL, Shao H, Feng YQ. A method for LSTM-based trajectory modeling and abnormal trajectory detection. *IEEE Access*, 2020, 8: 104063–104073. [doi: [10.1109/ACCESS.2020.2997967](https://doi.org/10.1109/ACCESS.2020.2997967)]
- [163] Smolyak D, Gray K, Badirli S, Mohler G. Coupled IGMM-GANs with applications to anomaly detection in human mobility data. *ACM Trans. on Spatial Algorithms and Systems*, 2020, 6(4): 24. [doi: [10.1145/3385809](https://doi.org/10.1145/3385809)]
- [164] Rasmussen CE. The infinite Gaussian mixture model. In: Proc. of the 12th Int'l Conf. on Neural Information Processing System. Denver: MIT Press, 1999. 554–560.
- [165] Donahue J, Krähenbühl P, Darrell T. Adversarial feature learning. In: Proc. of the 2017 Int'l Conf. on Learning Representations. 2017.
- [166] Dias MLD, Mattos CLC, da Silva TLC, de Macêd JAF, Silva WCP. Anomaly detection in trajectory data with normalizing flows. In: Proc. of the 2020 Int'l Joint Conf. on Neural Networks. Glasgow: IEEE, 2020. 1–8. [doi: [10.1109/IJCNN48605.2020.9206939](https://doi.org/10.1109/IJCNN48605.2020.9206939)]
- [167] Madan N, Farkhondeh A, Nasrollahi K, Escalera S, Moeslund TB. Temporal cues from socially unacceptable trajectories for anomaly detection. In: Proc. of the 2021 IEEE/CVF Int'l Conf. on Computer Vision. Montreal: IEEE, 2021. 2150–2158. [doi: [10.1109/ICCVW54120.2021.00244](https://doi.org/10.1109/ICCVW54120.2021.00244)]
- [168] Aksoy M, Ozdemir O, Guner G, Baspinar B, Koyuncu E. Flight trajectory pattern generalization and abnormal flight detection with generative adversarial network. In: Proc. of the 2021 AIAA Scitech Forum. AIAA, 2021. [doi: [10.2514/6.2021-0775](https://doi.org/10.2514/6.2021-0775)]
- [169] Fu QW, Zhu JH, Chen YP, Wan JT, He B. An automatic learning model for trajectory outlier detection. In: Proc. of the 3rd Int'l Conf. on Data Science and Information Technology. Xiamen: ACM, 2020. 220–226. [doi: [10.1145/3414274.3414505](https://doi.org/10.1145/3414274.3414505)]
- [170] Doshi K, Yilmaz Y. An efficient approach for anomaly detection in traffic videos. In: Proc. of the 2021 IEEE/CVF Conf. on Computer Vision and Pattern Recognition. Nashville: IEEE, 2021. 4236–4244. [doi: [10.1109/CVPRW53098.2021.00479](https://doi.org/10.1109/CVPRW53098.2021.00479)]
- [171] Liu FT, Ting KM, Zhou ZH. Isolation forest. In: Proc. of the 8th IEEE Int'l Conf. on Data Mining. Pisa: IEEE, 2008. 413–422. [doi: [10.1109/ICDM.2008.17](https://doi.org/10.1109/ICDM.2008.17)]
- [172] Zhou XB, Ding Y, Peng FC, Luo Q, Ni LM. Detecting unmetered taxi rides from trajectory data. In: Proc. of the 2017 IEEE Int'l Conf. on Big Data. Boston: IEEE, 2017. 530–535. [doi: [10.1109/BigData.2017.8257968](https://doi.org/10.1109/BigData.2017.8257968)]
- [173] Han XL, Cheng R, Ma CH, Grubenmann T. DeepTEA: Effective and efficient online time-dependent trajectory outlier detection. *Proc. of the VLDB Endowment*, 2022, 15(7): 1493–1505. [doi: [10.14778/3523210.3523225](https://doi.org/10.14778/3523210.3523225)]
- [174] ECML/PKDD 15: Taxi trajectory prediction (I). 2015. <https://www.kaggle.com/competitions/pkdd-15-predict-taxi-service-trajectory-i/data>
- [175] FHWA. Next generation simulation (NGSIM) vehicle trajectories and supporting data. 2023. <https://ops.fhwa.dot.gov/trafficanalysis/tools/ngsim.htm>
- [176] NYC. TLC trip record data. 2023. <https://www1.nyc.gov/site/tlc/about/tlc-trip-record-data.page>
- [177] Yuan J, Zheng Y, Zhang CY, Xie WL, Xie X, Sun GZ. T-drive: Driving directions based on taxi trajectories. In: Proc. of the 18th SIGSPATIAL Int'l Conf. on Advances in Geographic Information Systems. San Jose: ACM, 2010. 99–108. [doi: [10.1145/1869790.1869807](https://doi.org/10.1145/1869790.1869807)]
- [178] Chorochronos.org. Greek trucks. 2013. <http://www.chorochronos.org/>
- [179] Krajewski R, Bock J, Kloeker L, Eckstein L. The highD dataset: A drone dataset of naturalistic vehicle trajectories on German highways for validation of highly automated driving systems. In: Proc. of the 21st Int'l Conf. on Intelligent Transportation Systems. Maui: IEEE, 2018. 2118–2125. [doi: [10.1109/ITSC.2018.8569552](https://doi.org/10.1109/ITSC.2018.8569552)]
- [180] Bock J, Krajewski R, Moers T, Runde S, Vater L, Eckstein L. The inD dataset: A drone dataset of naturalistic road user trajectories at German intersections. In: Proc. of the 2020 IEEE Intelligent Vehicles Symp. Las Vegas: IEEE, 2020. 1929–1934. [doi: [10.1109/IV47402.2020.9304839](https://doi.org/10.1109/IV47402.2020.9304839)]
- [181] Krajewski R, Moers T, Bock J, Vater L, Eckstein L. The round dataset: A drone dataset of road user trajectories at roundabouts in Germany. In: Proc. of the 23rd IEEE Int'l Conf. on Intelligent Transportation Systems. Rhodes: IEEE, 2020. 1–6. [doi: [10.1109/ITSC45102.2020.9294728](https://doi.org/10.1109/ITSC45102.2020.9294728)]

- [182] Tsinghua University Suzhou Automotive Research Institute. Mirror-traffic—Holographic intelligent scenarios derived from real traffic data. 2020 (in Chinese). <http://www.scenarios.cn/html/dataset.html>
- [183] Chang MF, Lambert J, Sangkloy P, Singh J, Bak S, Hartnett A, Wang D, Carr P, Lucey S, Ramanan D, Hays J. Argoverse: 3D tracking and forecasting with rich maps. In: Proc. of the 2019 IEEE/CVF Conf. on Computer Vision and Pattern Recognition. Long Beach: IEEE, 2019. 8748–8757. [doi: 10.1109/CVPR.2019.00895]
- [184] Huang XY, Wang P, Cheng XJ, Zhou DF, Geng QC, Yang RG. The ApolloScape open dataset for autonomous driving and its application. IEEE Trans. on Pattern Analysis and Machine Intelligence, 2020, 42(10): 2702–2719. [doi: 10.1109/TPAMI.2019.2926463]
- [185] Zhan W, Sun LT, Wang D, Shi HJ, Clausse A, Naumann M, Kummerle J, Konigshof H, Stiller C, de La Fortelle A, Tomizuka M. Interaction dataset: An international, adversarial and cooperative motion dataset in interactive driving scenarios with semantic maps. arXiv:1910.03088, 2019.
- [186] Sun P, Kretschmar H, Dotiwala X, *et al.* Scalability in perception for autonomous driving: Waymo open dataset. In: Proc. of the 2020 IEEE/CVF Conf. on Computer Vision and Pattern Recognition. Seattle: IEEE, 2020. 2446–2454. [doi: 10.1109/CVPR42600.2020.00252]
- [187] Cordts M, Omran M, Ramos S, Rehfeld T, Enzweiler M, Benenson R, Franke U, Roth S, Schiele B. The Cityscapes dataset for semantic urban scene understanding. In: Proc. of the 2016 IEEE Conf. on Computer Vision and Pattern Recognition. Las Vegas: IEEE, 2016. 3213–3223. [doi: 10.1109/CVPR.2016.350]
- [188] Geiger A, Lenz P, Urtasun R. Are we ready for autonomous driving? The KITTI vision benchmark suite. In: Proc. of the 2012 IEEE Conf. on Computer Vision and Pattern Recognition. Providence: IEEE, 2012. 3354–3361. [doi: 10.1109/CVPR.2012.6248074]
- [189] Caesar H, Bankiti V, Lang AH, Vora S, Liong VE, Xu Q, Krishnan A, Pan Y, Baldan G, Beijbom O. nuScenes: A multimodal dataset for autonomous driving. In: Proc. of the 2020 IEEE/CVF Conf. on Computer Vision and Pattern Recognition. Seattle: IEEE, 2020. 11621–11631. [doi: 10.1109/CVPR42600.2020.01164]
- [190] Chandra R, Bhattacharya U, Bera A, Manocha D. TraPHic: Trajectory prediction in dense and heterogeneous traffic using weighted interactions. In: Proc. of the 2019 IEEE/CVF Conf. on Computer Vision and Pattern Recognition. Long Beach: IEEE, 2019. 8483–8492. [doi: 10.1109/CVPR.2019.00868]
- [191] Houston J, Zuidhof G, Bergamini L, Ye YW, Chen L, Jain A, Omari S, Iglovikov V, Ondruska P. One thousand and one hours: Self-driving motion prediction dataset. In: Proc. of the 4th Conf. on Robot Learning. Cambridge: PMLR, 2020. 409–418.
- [192] Chandra R, Wang XJ, Mahajan M, Kala R, Palugulla R, Naidu C, Jain A, Manocha D. METEOR: A dense, heterogeneous, and unstructured traffic dataset with rare behaviors. In: Proc. of the 2023 IEEE Int'l Conf. on Robotics and Automation. London: IEEE, 2023. 9169–9175. [doi: 10.1109/ICRA48891.2023.10161281]
- [193] Che ZP, Jiang B, Meng YP, Li GY, Li T, Dong K, Zhang XS, Shi XF, Lyu Y, Wu GB, Liu Y, Tang J, Ye JP. D²-city: A large-scale dashcam video dataset of diverse traffic scenarios. arXiv:1904.01975, 2019. [doi: 10.11922/sciencedb.00603]
- [194] Zhang WS, Qi GD, Pan G, Lu H, Li SJ, Wu ZH. City-scale social event detection and evaluation with taxi traces. ACM Trans. on Intelligent Systems and Technology, 2015, 6(3): 40. [doi: 10.1145/2700478]
- [195] Pan B, Zheng Y, Wilkie D, Shahabi C. Crowd sensing of traffic anomalies based on human mobility and social media. In: Proc. of the 21st ACM SIGSPATIAL Int'l Conf. on Advances in Geographic Information Systems. Orlando: ACM, 2013. 344–353. [doi: 10.1145/2525314.2525343]
- [196] Hu WM, Li X, Tian GD, Maybank S, Zhang ZF. An incremental DPMM-based method for trajectory clustering, modeling, and retrieval. IEEE Trans. on Pattern Analysis and Machine Intelligence, 2013, 35(5): 1051–1065. [doi: 10.1109/TPAMI.2012.188]
- [197] Lin WY, Zhou Y, Xu HT, Yan JC, Xu ML, Wu JX, Liu ZC. A tube-and-droplet-based approach for representing and analyzing motion trajectories. IEEE Trans. on Pattern Analysis and Machine Intelligence, 2017, 39(8): 1489–1503. [doi: 10.1109/TPAMI.2016.2608884]
- [198] Morris BT, Trivedi MM. Trajectory learning for activity understanding: Unsupervised, multilevel, and long-term adaptive approach. IEEE Trans. on Pattern Analysis and Machine Intelligence, 2011, 33(11): 2287–2301. [doi: 10.1109/TPAMI.2011.64]
- [199] Zheng Y, Zhang LZ, Xie X, Ma WY. Mining interesting locations and travel sequences from GPS trajectories. In: Proc. of the 18th Int'l Conf. on World Wide Web. Madrid: ACM, 2009. 791–800. [doi: 10.1145/1526709.1526816]
- [200] Lerner A, Chrysanthou Y, Lischinski D. Crowds by example. Computer Graphics Forum, 2007, 26(3): 655–664. [doi: 10.1111/j.1467-8659.2007.01089.x]
- [201] Pellegrini S, Ess A, Schindler K, van Gool L. You'll never walk alone: Modeling social behavior for multi-target tracking. In: Proc. of the 12th IEEE Int'l Conf. on Computer Vision. Kyoto: IEEE, 2009. 261–268. [doi: 10.1109/ICCV.2009.5459260]
- [202] Robicquet A, Sadeghian A, Alahi A, Savarese S. Learning social etiquette: Human trajectory understanding in crowded scenes. In: Proc. of the 14th European Conf. on Computer Vision. Amsterdam: Springer, 2016. 549–565. [doi: 10.1007/978-3-319-46484-8_33]
- [203] Becker S, Hug R, Hübner W, Arens M. An evaluation of trajectory prediction approaches and notes on the TrajNet benchmark.

- arXiv:1805.07663, 2018.
- [204] Benfold B, Reid I. Stable multi-target tracking in real-time surveillance video. In: Proc. of the 2011 CVPR. Colorado Springs: IEEE, 2011. 3457–3464. [doi: [10.1109/CVPR.2011.5995667](https://doi.org/10.1109/CVPR.2011.5995667)]
- [205] Rasouli A, Kotseruba I, Kunic T, Tsotsos J. PIE: A large-scale dataset and models for pedestrian intention estimation and trajectory prediction. In: Proc. of the 2019 IEEE/CVF Int'l Conf. on Computer Vision. Seoul: IEEE, 2019. 6262–6271. [doi: [10.1109/ICCV.2019.00636](https://doi.org/10.1109/ICCV.2019.00636)]
- [206] Rasouli A, Kotseruba I, Tsotsos JK. Are they going to cross? A benchmark dataset and baseline for pedestrian crosswalk behavior. In: Proc. of the 2017 IEEE Int'l Conf. on Computer Vision Workshops. Venice: IEEE, 2017. 206–213. [doi: [10.1109/ICCVW.2017.33](https://doi.org/10.1109/ICCVW.2017.33)]
- [207] DS4C: Data science for COVID-19 in South Korea. 2020. <https://www.kaggle.com/datasets/kimjihoo/coronavirusdataset>
- [208] Beijing Advanced Innovation Center for Big Data and Brain Computing, Beihang University. 2020 (in Chinese). <https://github.com/BDBC-KG-NLP/COVID-19-tracker>
- [209] Schäfer M, Strohmeier M, Leonardi M, Lenders V. LocaRDS: A localization reference data set. *Sensors*, 2021, 21(16): 5516. [doi: [10.3390/s21165516](https://doi.org/10.3390/s21165516)]
- [210] Majdik AL, Till C, Scaramuzza D. The Zurich urban micro aerial vehicle dataset. *The Int'l Journal of Robotics Research*, 2017, 36(3): 269–273. [doi: [10.1177/0278364917702237](https://doi.org/10.1177/0278364917702237)]
- [211] Gloaguen P, Chapel L, Friguet C, Tavenard R. Scalable clustering of segmented trajectories within a continuous time framework: Application to maritime traffic data. *Machine Learning*, 2023, 112(6): 1975–2001. [doi: [10.1007/s10994-021-06004-8](https://doi.org/10.1007/s10994-021-06004-8)]
- [212] Varlamis I, Kontopoulos I, Tserpes K, Etemad M, Soares A, Matwin S. Building navigation networks from multi-vessel trajectory data. *GeoInformatica*, 2021, 25(1): 69–97. [doi: [10.1007/s10707-020-00421-y](https://doi.org/10.1007/s10707-020-00421-y)]
- [213] Landsea CW, Franklin JL. Atlantic hurricane database uncertainty and presentation of a new database format. *Monthly Weather Review*, 2013, 141(10): 3576–3592. [doi: [10.1175/MWR-D-12-00254.1](https://doi.org/10.1175/MWR-D-12-00254.1)]
- [214] Kranstauber B, Cameron A, Weinzerl R, Fountain T, Tilak S, Wikelski M, Kays R. The Movebank data model for animal tracking. *Environmental Modelling & Software*, 2011, 26(6): 834–835. [doi: [10.1016/j.envsoft.2010.12.005](https://doi.org/10.1016/j.envsoft.2010.12.005)]
- [215] Domingues R, Filippone M, Michiardi P, Zouaoui J. A comparative evaluation of outlier detection algorithms: Experiments and analyses. *Pattern Recognition*, 2018, 74: 406–421. [doi: [10.1016/j.patcog.2017.09.037](https://doi.org/10.1016/j.patcog.2017.09.037)]
- [216] Zhao Y, Nasrullah Z, Li Z. PyOD: A Python toolbox for scalable outlier detection. *Journal of Machine Learning Research*, 2019, 20(96): 1–7.
- [217] Yilmaz SF, Kozat SS. PySAD: A streaming anomaly detection framework in Python. arXiv:2009.02572, 2020.
- [218] Gao YF, Li QZ, Li XS, Lin J, Rangwala H. TrajViz: A tool for visualizing patterns and anomalies in trajectory. In: Proc. of the 2017 Joint European Conf. on Machine Learning and Knowledge Discovery in Databases. Skopje: Springer, 2017. 428–431. [doi: [10.1007/978-3-319-71273-4_45](https://doi.org/10.1007/978-3-319-71273-4_45)]
- [219] Correa-Baena JP, Hippalgaonkar K, van Duren J, Jaffer S, Chandrasekhar VR, Stevanovic V, Wadia C, Guha S, Buonassisi T. Accelerating materials development via automation, machine learning, and high-performance computing. *Joule*, 2018, 2(8): 1410–1420. [doi: [10.1016/j.joule.2018.05.009](https://doi.org/10.1016/j.joule.2018.05.009)]
- [220] Brodtkorb AR, Hagen TR, Sætra ML. Graphics processing unit (GPU) programming strategies and trends in GPU computing. *Journal of Parallel and Distributed Computing*, 2013, 73(1): 4–13. [doi: [10.1016/j.jpdc.2012.04.003](https://doi.org/10.1016/j.jpdc.2012.04.003)]
- [221] Shazeer N, Cheng YL, Parmar N, Tran D, Vaswani A, Koanantakool P, Hawkins P, Lee H, Hong MS, Young C, Sepassi R, Hechtman B. Mesh-TensorFlow: Deep learning for supercomputers. In: Proc. of the 32nd Int'l Conf. on Neural Information Processing Systems. Montréal: Curran Associates Inc., 2018. 10435–10444.
- [222] Dean J, Ghemawat S. MapReduce: Simplified data processing on large clusters. *Communications of the ACM*, 2008, 51(1): 107–113. [doi: [10.1145/1327452.1327492](https://doi.org/10.1145/1327452.1327492)]
- [223] Yu J, Zhang ZS, Sarwat M. Spatial data management in Apache Spark: The GeoSpark perspective and beyond. *GeoInformatica*, 2019, 23(1): 37–78. [doi: [10.1007/s10707-018-0330-9](https://doi.org/10.1007/s10707-018-0330-9)]
- [224] Wu ZH, Pan SR, Chen FW, Long GD, Zhang CQ, Yu PS. A comprehensive survey on graph neural networks. *IEEE Trans. on Neural Networks and Learning Systems*, 2021, 32(1): 4–24. [doi: [10.1109/TNNLS.2020.2978386](https://doi.org/10.1109/TNNLS.2020.2978386)]
- [225] Veličković P, Cucurull G, Casanova A, Romero A, Liò P, Bengio Y. Graph attention networks. arXiv:1710.10903, 2017.
- [226] Vaswani A, Shazeer N, Parmar N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser Ł, Polosukhin I. Attention is all you need. In: Proc. of the 31st Int'l Conf. on Neural Information Processing Systems. Long Beach: Curran Associates Inc., 2017. 6000–6010.
- [227] Chen T, Kornblith S, Norouzi M, Hinton G. A simple framework for contrastive learning of visual representations. In: Proc. of the 37th Int'l Conf. on Machine Learning. JMLR.org, 2020. 149.
- [228] Chen T, Kornblith S, Swersky K, Norouzi M, Hinton GE. Big self-supervised models are strong semi-supervised learners. In: Proc. of

- the 34th Int'l Conf. on Neural Information Processing Systems. 2020. 22243–22255.
- [229] 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: ACM, 2016. 1135–1144. [doi: 10.1145/2939672.2939778]
- [230] Kaur H, Nori H, Jenkins S, Caruana R, Wallach H, Vaughan JW. Interpreting interpretability: Understanding data scientists' use of interpretability tools for machine learning. In: Proc. of the 2020 CHI Conf. on Human Factors in Computing Systems. Honolulu: ACM, 2020. 1–14. [doi: 10.1145/3313831.3376219]

附中文参考文献:

- [7] 曹翰林, 唐海娜, 王飞, 徐勇军. 轨迹表示学习技术研究进展. 软件学报, 2021, 32(5): 1461–1479. <http://www.jos.org.cn/1000-9825/6210.htm> [doi: 10.13328/j.cnki.jos.006210]
- [12] 高强, 张凤荔, 王瑞锦, 周帆. 轨迹大数据: 数据处理关键技术研究综述. 软件学报, 2017, 28(4): 959–992. <http://www.jos.org.cn/1000-9825/5143.htm> [doi: 10.13328/j.cnki.jos.005143]
- [15] 毛嘉莉, 金澈清, 章志刚, 周傲英. 轨迹大数据异常检测: 研究进展及系统框架. 软件学报, 2017, 28(1): 17–34. <http://www.jos.org.cn/1000-9825/5151.htm> [doi: 10.13328/j.cnki.jos.005151]
- [64] 黄添强, 余养强, 郭躬德, 秦小麟. 半监督的移动对象离群轨迹检测算法. 计算机研究与发展, 2011, 48(11): 2074–2082.
- [90] 刘良旭, 乔少杰, 刘宾, 乐嘉锦, 唐常杰. 基于R-Tree的高效异常轨迹检测算法. 软件学报, 2009, 20(9): 2426–2435. <http://www.jos.org.cn/1000-9825/3580.htm> [doi: 10.3724/SP.J.1001.2009.03580]
- [182] 清华大学苏州汽车研究院. 镜——源于真实交通数据的全息式智能场景库. <http://www.scenarios.cn/html/dataset.html>
- [208] 北京航空航天大学大数据科学与脑机智能高精尖创新中心. 2020. <https://github.com/BDBC-KG-NLP/COVID-19-tracker>



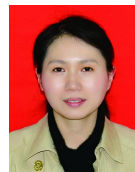
李超能(1995—), 男, 博士生, 主要研究领域为轨迹数据挖掘, 机器学习, 计算机视觉.



李宇楠(1992—), 男, 博士, 讲师, CCF 专业会员, 主要研究领域为计算机视觉, 机器学习.



冯冠文(1998—), 男, 博士生, CCF 学生会员, 主要研究领域为数据挖掘, 模式识别.



谢琨(1980—), 女, 博士, 教授, CCF 专业会员, 主要研究领域为计算机视觉, 机器学习, 模式识别.



姚航(1998—), 男, 博士生, 主要研究领域为图像分类, 机器学习.



苗启广(1972—), 男, 博士, 教授, 博士生导师, CCF 杰出会员, 主要研究领域为智能图像处理, 机器学习, 模式识别, 高性能计算.



刘如意(1989—), 女, 博士, 副教授, CCF 高级会员, 主要研究领域为遥感图像分析和处理, 计算机视觉, 深度学习.