

基于深度强化学习的在线道路网络异常子轨迹检测

欧阳宇豪

2023.12.10

摘要

在许多基于位置的应用中，检测异常轨迹已成为一项重要任务。虽然针对这项任务提出了许多方法，但这些方法都存在各种问题，包括：(1) 无法检测异常子轨迹（即轨迹数据中更细粒度的异常），(2) 非数据驱动，(3) 需要足够的监督标签，而收集监督标签的成本很高。在本文中，我们提出了一种名为 RL4OASD 的基于强化学习的解决方案，它避免了上述现有方法的所有问题。RL4OASD 包括两个网络，一个负责学习道路网络和轨迹的特征，另一个负责根据学习到的特征检测异常子轨迹。在两个真实数据集上进行了广泛的实验之后，结果表明，我们的方案明显优于最先进的方法（提高了 20-30%），而且在线检测效率高（处理每个新生成的数据点所需时间少于 0.1ms）。

关键词：轨迹数据；异常子轨迹检测；道路网络；深度强化学习

1 引言

随着移动计算和地理定位技术（如 GPS 设备和智能手机）的发展，各种移动物体（如人、车辆）在空间中产生了大量空间轨迹数据。这些收集到的轨迹数据记录了移动物体在不同时间戳的移动轨迹，反映了物体多样化的移动模式。准确的空间轨迹数据分析是智能交通 [1]、基于位置的推荐服务 [2]、大流行病追踪 [3] 和公共安全犯罪预防 [4] 等多种时空应用的关键技术组成部分。

在各种轨迹挖掘应用中，异常轨迹检测在许多实际场景中发挥着至关重要的作用。例如，实时车辆轨迹异常检测有利于改善交通管理 [5]。此外，通过研究人类的移动行为以发现其异常轨迹也有助于预测异常事件（如内乱和大流行病爆发）[6]。在这种情况下，异常轨迹指的是不符合正常移动模式的轨迹，它偏离了具有相同出发地和目的地（称为 SD 对）的大多数轨迹的移动模型 [7]、[8]。如图 1 中的示例。从出发地 S(e1) 到目的地 D(e10) 有三条轨迹。如果具有相同 SD 对 $\langle S, D \rangle$ 的大多数轨迹都遵循 T1（蓝色轨迹）或 T2（绿色轨迹），则轨迹 T3（红色轨迹）被视为异常轨迹。

最近，异常轨迹的检测引起了广泛关注，学术界提出各种方法（[8]、[9]、[10]、[11]、[12]、[13]）来解决这一问题。然而，目前的方法还不能很好地解决几个关键挑战，详细说明如下。

(1) 无法检测异常子轨迹。现有的大多数方法 [9]、[10]、[11]、[12]、[13] 只关注在粗粒度级别识别异常轨迹，并检测轨迹整体是否异常。然而，细粒度层面的异常轨迹检测更有利于在智慧城市应用中做出更好的决策，但遗憾的是，目前的方法对这一问题的探讨较少。例如，

当驾驶员的轨迹开始偏离正常路线时，打车公司就能立即发现异常驾驶员，这表明存在异常子轨迹。因此，在这项工作中，我们提出了及时检测细粒度的异常子轨迹。

(2) 非数据驱动。现有的一些方法是基于预先定义参数或规则，而不是数据驱动的 [8], [10]。例如，[8] 中的方法是使用一些手动定义的参数来隔离异常子轨迹。它使用一个自适应窗口来保留最新的 GPS 接收点，以便与正常轨迹进行比较，正常轨迹由 SD 对中的大多数轨迹支持。当一个新的输入点被添加到窗口中时，它会检查窗口中子轨迹的支持率。如果支持率大于预先设定的阈值，则该点被标记为正常点；否则，该点被标记为异常点，自适应窗口缩小到只包含最新点。这个过程一直持续到轨迹完成。异常子轨迹是指那些被标记为异常的点。然而，阈值很难设置得当，而且不够通用，无法涵盖所有可能的情况或适应不同的路况。[10] 中的另一种方法使用轨迹相似性（如离散弗雷谢特）来检测异常点。它为每个新进入的点计算给定正常轨迹与当前部分轨迹之间的距离，如果距离超过给定阈值，则报告异常事件；否则，继续检测。

(3) 需要足够的监督标签。虽然已经开发了多种机器学习模型来识别异常轨迹 [12]、[13]，但这些方法的有效性在很大程度上依赖于监督学习框架下的充足标记数据。然而，与整个轨迹数据集相比，实际应用场景中标注的异常轨迹可能非常有限，这是因为需要大量的人力，例如，[8] 中使用的轨迹数据仅包含 9 对 SD。

在本文中，我们提出了一种基于强化学习的新型解决方案 RL4OASD，它避免了现有方法的上述问题。首先，设计了 RL4OASD 用于以在线方式检测异常子轨迹，从而避免了第一个问题。它通过预测轨迹中每个路段的正常/异常标签，并根据路段标签检测异常子轨迹来实现这一点。其次，RL4OASD 是一种数据驱动方法，依赖于无标签的数据，因此可以避免第二和第三个问题。具体来说，RL4OASD 由三个部分组成，即数据预处理、RSRNet 网络和 ASDNet 网络（见图 2）。在数据预处理中，它对原始轨迹进行地图匹配，将其映射到道路网络上，得到地图匹配轨迹。在地图匹配轨迹的基础上，它根据各路段之间的历史过渡数据，利用一些启发式方法计算出轨迹所涉及路段的一些标签。这些有噪声的标签将用于以弱监督方式训练 RSRNet。在 RSRNet 中，它根据交通环境特征和正常路线特征来学习路段的表征。这些向量形式的表征将被输入 ASDNet，以定义用于标注异常子轨迹的马尔可夫决策过程（MDP）的状态。在 ASDNet 中，它将标注异常子轨迹的任务建模为 MDP，并通过策略梯度法学习策略 [14]。ASDNet 会输出细化的路段标签，这些标签会被进一步用于再次训练 RSRNet，然后 RSRNet 会为 ASDNet 提供更好的观察结果，以训练出更好的策略。我们要强调的是，预处理组件计算出的噪声标签只是为解决训练 RSRNet 时的冷启动问题提供了一些先验知识，但并不像监督范式那样用于训练模型。此外，异常轨迹的概念可能会随着时间的推移而发生偏移。例如，由于交通状况的变化（如一些事故），轨迹可能会逐渐偏离正常路线，“正常”和“异常”的概念也会随之改变。RL4OASD 可通过在线学习策略来处理“概念漂移”，即在新数据到来时继续完善。

我们的贡献可归纳如下

我们提出了首个基于深度强化学习的异常子轨迹检测解决方案。所提出的模型 1) 可以自然地检测异常子轨迹，2) 是数据驱动的，3) 不需要标注数据。此外，它还能通过在线学习处理异常轨迹的概念漂移。

我们在成都和西安两个真实世界的轨迹数据集上进行了广泛的实验。我们将所提出的解决方案与各种基线进行了比较，结果表明我们的解决方案是有效的（例如，与现有最佳方法

相比，我们的解决方案提高了 20-30%)，而且效率很高（例如，处理每个新生成的数据点所需的时间不到 0.1 毫秒）。

我们对成都和西安两个真实数据集的异常子轨迹进行了人工标注测试。每个标注数据集包括 200 对 SD 和 1,688 条（1,057 条）与成都数据集（西安数据集）中这些 SD 配对的地图匹配轨迹。这个标注数据集比现有的已知数据集 [8] 大 50 多倍。这个相对较大的标注数据集将有助于对异常子轨迹检测方法进行全面、可靠的评估。

2 相关工作

2.1 在线异常轨迹/子轨迹检测

在线异常轨迹检测旨在以在线方式检测正在进行的轨迹。现有研究针对该问题提出了许多方法，涉及两类：基于启发式的方法 [8]、[10] 和基于学习的方法 [9]、[11]。

在启发式方法方面，Chen 等人 [8] 通过基于隔离的方法，以在线方式研究了异常轨迹的检测，该方法旨在检查轨迹的哪些部分与具有相同出发和目的地路线的参考轨迹（即正常轨迹）相隔离。这种方法旨在从正在进行的轨迹中检测异常子轨迹，这与我们的论文类似。不过，该方法基于许多手动设置的参数对参考轨迹进行建模。此外，该方法的性能评估是在一个仅有 9 对 SD 的小型人工标注数据集上进行的，无法反映该方法的通用性。最近的一项研究 [10] 提出通过给定参考路线和当前部分路线在每个时间戳的离散弗雷谢特距离来计算轨迹相似度。如果偏差在任意时间戳超过给定的阈值，系统就会发出警报，表示检测到绕行的异常事件。这种方法涉及许多预定义参数。我们的工作与这些基于启发式的研究不同，它是基于通过强化学习学习到的策略，而不是手工制作的启发式，其中有许多预定义的参数（如偏差阈值）用于检测异常。此外，我们还在一个大型数据集上评估了我们方法的性能。

对于基于学习的方法，最近的一项研究 [11] 提出通过生成方案来检测异常轨迹，该方案利用高斯混合分布来表示不同类型的正常路线，并检测那些无法根据给定的正常路线表示生成的异常轨迹。这种方法旨在检测正在进行的轨迹是否异常。另一种基于学习的方法 [9] 提出了一种概率模型，通过对历史轨迹的驾驶行为分布建模来检测轨迹异常。该方法涉及许多与驾驶行为建模相关的潜在特征，包括道路水平和转弯角度等。该方法也只针对在线轨迹是否异常。

在我们的工作中，我们的目标是更精细的设置，即以在线方式检测异常轨迹的哪一部分（即子轨迹）是造成其异常的原因。

2.2 离线异常轨迹检测

离线异常轨迹检测是指检测异常轨迹或轨迹的异常子轨迹，其中轨迹以离线方式输入。此外，还有两类研究。一类是基于启发式的方法 [15]、[16]、[17]、[18]、[19]，另一类是基于学习的方法 [20]、[12]。对于基于启发式的方法，通常利用一些涉及距离或密度的启发式度量来检测异常轨迹。例如，早期研究 [15] 提出了一种分区和检测机制来进行异常子轨迹检测。其主要思路是对轨迹进行分区，并通过计算目标轨迹中每个轨迹段与其他轨迹段的距离来检测异常轨迹段。最近的一项研究 [18] 在挖掘正常轨迹模式的基础上，利用编辑距离度量来检测异常轨迹。另一项研究 [16] 建议根据相同的行程对轨迹进行聚类，并通过检查目标轨迹是否与

聚类隔离来进一步检测异常轨迹。此外, [21] 还提出了一种用于检测出租车欺诈驾驶的系统。该方法的主要思想是通过结合距离和密度指标来检测异常轨迹。

还有一些研究提出利用基于学习的方法来检测异常轨迹。例如, Song 等人 [12] 采用递归神经网络 (RNN) 捕捉轨迹的序列信息进行检测。不过, 所提出的模型需要在监督下进行训练, 而在实际应用中通常无法获得标注数据。另一项基于学习的工作 [20] 旨在利用对抗模型 (GAN) 检测新移动主体的异常轨迹 (新出租车司机检测)。显然, 这些研究中提出的方法无法用于在线检测场景。

2.3 其他类型的异常检测研究

一些研究 [19]、[22]、[23]、[24]、[25] 主要关注与我们的研究相关的其他类型的异常轨迹检测。我们对这些研究综述如下。Banerjee 等人 [19] 研究了时间异常检测, 该方法利用旅行时间估算并遍历目标轨迹中的一些潜在子轨迹, 如果旅行时间与预期时间有很大偏差, 就能识别出异常。Li 等人 [22] 检测了时间异常, 并研究了一种利用历史相似性趋势的方法。此外, Ge 等人 [23] 研究了 Top-K 演变异常。[24]、[25] 检测大规模轨迹流中的异常轨迹。

2.4 深度强化学习

深度强化学习的目的是在代理与特定环境交互时, 引导代理做出连续决策, 使累积奖励最大化, 而环境通常被建模为马尔可夫决策过程 (MDP) [26]。近年来, 强化学习备受研究关注。例如, Oh 等人 [27] 探索了用于连续异常检测的反强化学习, Huang 等人 [28] 设计了一种基于深度 RL 的异常检测器, 用于时间序列检测。Wang 等人 [29]、[30] 提出使用基于 RL 的算法来加速子轨迹或子棋局相似性搜索。此外, [31]、[32] 还开发了基于 RL 的解决方案来简化具有不同目标的轨迹。在本文中, 我们为在线异常子轨迹检测提出了一种新颖的基于 RL 的解决方案 (称为 RL4OASD), 并采用策略梯度法 [33] 解决该问题。据我们所知, 这是第一个基于深度强化学习的在线异常子轨迹检测解决方案。

3 本文方法

3.1 本文方法概述

判断正在进行的轨迹的子轨迹是否异常是一个马尔可夫决策过程 (MDP) [26]。我们提出了一个名为 RL4OASD 的弱监督框架 (概述见图 1)。该框架由三个部分组成, 即数据预处理、RSRNet 和 ASDNet。

在数据预处理中, 我们对原始轨迹进行地图匹配, 将其映射到道路网络上, 得到地图匹配轨迹。在地图匹配轨迹的基础上, 我们根据各路段之间的历史过渡数据, 利用一些启发式方法计算出轨迹所涉及的一些路段标签。这些标签是有噪声的, 将用于以弱监督方式训练 RSRNet。在 RSRNet 中, 我们根据交通环境特征和正常路线特征来学习路段表征。这些向量形式的表征将被输入 ASDNet, 以定义用于标注异常子轨迹的 MDP 状态。在 ASDNet 中, 我们将标注异常子轨迹的任务建模为 MDP, 并通过策略梯度法学习策略 [14]。ASDNet 会输出细化的路段标签, 并进一步用于再次训练 RSRNet, 然后 RSRNet 会为 ASDNet 提供更好的观察结

果，以训练出更好的策略。这一过程不断重复，我们将结合 RSRNet 和 ASDNet 的算法称为 RL4OASD。

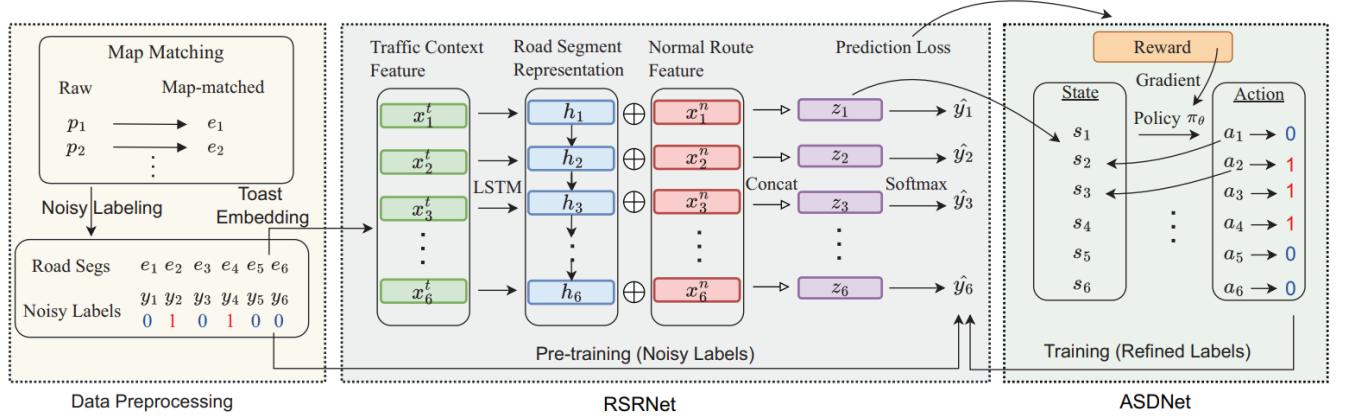


图 1. RL4OASD 架构图

RL4OASD 的有效性背后的一些解释如下。首先，在 RSRNet 中，与异常情况相关的交通环境特征（如行驶速度、行程持续时间）和正常路线特征都被很好地捕捉到模型中。其次，在 ASDNet 中，标记异常子轨迹的任务被表述为一个 MDP，其策略是以数据驱动的方式学习的，而不是像现有研究那样使用启发式方法（如一些预定义参数）。第三，首先用噪声标签训练 RSRNet，以处理冷启动问题。然后，对 RSRNet 和 ASDNet 进行迭代协作训练，其中 RSRNet 使用 ASDNet 的输出作为标签，而 ASDNet 使用 RSRNet 的输出作为特征。

3.2 先验知识

原始轨迹 (Raw Trajectory): 原始轨迹 T 由一系列 GPS 点组成, 即 $T = \langle p_1, p_2, \dots, p_n \rangle$, 其中 GPS 点 p_i 为三元组形式, 即 $p_i = (x_i, y_i, t_i)$, 表示运动物体在时间戳 t_i 位于 (x_i, y_i) 上, n 表示轨迹 T 的长度。

道路网络 (Road Network): 道路网络表示为有向图 $G(V, E)$, 其中 V 表示顶点集, 指十字路口或交叉路口, E 表示边集, 指路段, 每条边 e 连接道路网络上的两个顶点 u 和 v , 表示为 $e = (u, v)$ 。

地图匹配轨迹 (Map-matched Trajectory): 地图匹配轨迹指的是移动物体在道路网络上产生的轨迹, 它对应于一个路段序列 [34], 即 $T = \langle e_1, e_2, \dots, e_n \rangle$ 。为简单起见, 我们用 T 表示地图匹配轨迹, 在本文其他部分, 我们将地图匹配轨迹称为轨迹或路线。

子轨迹和过渡 (Subtrajectory and Transition): 子轨迹 $T[i, j]$ 与轨迹 $T = \langle e_1, e_2, \dots, e_n \rangle$ 从 e_i 到 e_j 的一部分相对应, $1 \leq i \leq j \leq n$ 。过渡段被定义为子轨迹的一种特例, 它只由相邻的两个路段组成, 即 $\langle e_{i-1}, e_i \rangle$, 其中 $1 < i \leq n$ 。

3.3 问题定义

本文研究了在线异常子轨迹检测问题。考虑源 S 和目标 D 对 (SD 对) 以及它们之间的一组轨迹 \mathcal{T} 。直观地说, 轨迹 T 可以被认为是正常的, 如果它遵循 \mathcal{T} 中大多数轨迹走过的路线。在此基础上, 我们表示了一个异常子轨迹, 表示轨迹的一部分, 它不遵循 SD 对内的正常路线。这个问题的表述如下。

问题 1(OASD): 给定一个持续的从源 S_T 到目标 D_T 以在线方式生成的轨迹 $T = \langle e_1, e_2, \dots, e_n \rangle$, 其中点 e_i 是一个一个生成的, 未来点无法提前获得。OASD 问题是检测和更新 T 的哪些部分 (即子轨迹) 是异常的, 而 T 是顺序生成的。

3.4 数据预处理

数据处理组件包括地图匹配过程 [34] 和噪声标签获取过程。后者产生一些噪声标签, 用于预训练 RSRNet 中的表示, 也为 ASDNet 中的策略学习提供了热启动。回顾一下, 我们并不假定训练时可以使用真实标签, 因为手动标注数据非常耗时。具体来说, 获取噪声标签的过程包括四个步骤。

步骤 1: 我们根据不同的 SD 对和时间段对数据集中的历史轨迹进行分组。在这里, 我们有 24 个时隙, 如果我们以一个小时的粒度划分一天, 如果一个轨迹的开始旅行时间在这个时隙内, 则其落在该时隙中。例如, 在图 1 中, 我们有三个映射匹配的轨迹 T_1, T_2 和 T_3 , 其源 e_1 和目的地 e_{10} 。假设 T_1, T_2, T_3 的出发时间分别为 9:00、9:10、9:30。然后, 所有三个轨迹都在同一组中, 因为它们在同一时间段内, 具有一个小时的粒度。

步骤 2: 然后我们计算每组中所有轨迹从一个路段到另一个路段的过渡比例。假设沿 T_1 有 5 条轨迹, 沿 T_2 有 4 条轨迹, 沿 T_3 只有 1 条轨迹, 那么跃迁 $\langle e_1, e_2 \rangle$ 的比例计算为 $5/10 = 0.5$, 因为它在所有 10 条轨迹中出现了 5 次 (沿 T_2 和 T_3)。

步骤 3: 对于每个轨迹, 我们引用它所属的组, 并将其映射到相对于每个路段的过渡分数序列。例如, 对于运行路线为 T_3 的轨迹, 其映射的过渡序列为 $\langle *, e_1 \rangle, \langle e_1, e_2 \rangle, \langle e_2, e_4 \rangle, \langle e_4, e_{11} \rangle, \langle e_{11}, e_{12} \rangle, \langle e_{12}, e_{13} \rangle, \langle e_{13}, e_{14} \rangle, \langle e_{14}, e_{15} \rangle, \langle e_{15}, e_{10} \rangle$, 其中初始过渡为 $\langle *, e_1 \rangle$, 对应的分数序列为 $\langle 1.0, 0.5, 0.5, 0.1, 0.1, 0.1, 0.1, 1.0 \rangle$ 。注意, 源 e_1 (对应 $\langle *, e_1 \rangle$) 和目的地 e_{10} (对应 $\langle e_{15}, e_{10} \rangle$) 上的分数总是设置为 1.0, 因为源和目的地路段肯定在其组内行驶。

步骤 4: 我们使用阈值参数 α 获得噪声标签, 其中 0 表示正常道路段, 其分数大于 α 表示道路段频繁行驶, 否则为 1。例如, 使用阈值 $\alpha = 0.5$, 我们得到 T_3 的噪声标签为 $\langle 0, 1, 1, 1, 1, 1, 1, 1, 0 \rangle$ 。

3.5 道路分段表示网 (RSRNet)

在 RSRNet 中, 我们采用 LSTM [35] 结构, 它接受不同长度的轨迹, 并捕获轨迹背后的顺序信息。我们将两种类型的特征嵌入到表示中, 即道路网络上的交通上下文特征和给定 SD 对的正常路线特征。

交通背景特征 (TCF)。 地图匹配轨迹对应于一连串的路段, 每个路段是一个标记 (即路段 ID)。我们将 RSRNet 嵌入层中的每个路段作为一个向量进行预训练, 以捕捉交通环境特征 (如行驶速度、行程持续时间、道路类型)。为此, 我们采用了 Toast [36], 这是一种最新的路网表示学习模型, 可支持基于路段的应用。学习到的路段表示将用于初始化 RSRNet 中的嵌入层, 这些嵌入可以通过模型训练进一步优化。其他路网表示模型 [37]、[38] 也适用于这项任务。

正常路线特征 (NRF)。 给定一个时隙中的 SD 对, 我们首先推断其中的正常路线。直观地说, 正常轨迹通常遵循相同的路线。如果一条轨迹包含一些其他人很少走过的路段, 那么它很可能包含异常路线。因此, 我们通过计算通过该路线的轨迹相对于其 SD 对中所有轨迹

的比例，来推断一条路线为正常路线。推断结果是通过比较阈值（用 α 表示）和各路段的分数得出的，也就是说，如果分数大于阈值，则推断该路线为正常路线，反之亦然。例如，在图 1 中，沿 T_1 有 5 条轨迹，沿 T_2 有 4 条，沿 T_3 只有 1 条。鉴于 $\alpha = 0.3$ ，我们推断 T_1 （以及 T_2 ）为正常路线，因为在所有 10 条轨迹中，其分数 $5/10 = 0.5$ （以及 $4/10 = 0.4$ ）都大于阈值 α 。根据推断出的正常路线，我们提取轨迹的特征。例如，给定一条遵循 T_3 的轨迹，我们提取的特征如下：如果目标轨迹中的路段上的过渡发生在推断出的正常路线上，则该路段为正常路段（即 0）；否则为 1。例如，提取的 T_3 正常特征为 $\langle 0, 0, 0, 1, 1, 1, 1, 0 \rangle$ ，其中路段 e_2 的特征为 0，因为过渡 $\langle e_1, e_2 \rangle$ 发生在正常路线 T_2 中。请注意，源路段和目的地路段的特征总是 0（即正常）。我们注意到正常路线特征和噪声标签都是 0-1 的形式。它们之间的区别在于，前者是为了捕捉正常路线的信息，相应地，它是由正常路线在路线层面上获得的。而后者则用作训练 RSRNet 的标签，是通过计算边缘级的过渡频率获得的。此外，前者在整个训练过程中使用，而后者仅用于 RSRNet 表征的预训练（这将为 ASDNet 提供一个热启动）。在获得以标记为形式的正常路线特征后，我们通过将标记嵌入为独热向量来获得该特征的向量。我们将获得的向量称为嵌入式正常路线特征。

训练 RSRNet。图 2 展示了 RSRNet 的架构。具体来说，给定一串嵌入式交通环境特征 $x_i^t (1 \leq i \leq n)$ ，其中 n 表示轨迹长度，LSTM 在每个路段获得隐藏状态 h_i 。然后，我们将 h_i 与嵌入的正常路线特征 x_i^n 连接起来，表示成 $z_i = [h_i; x_i^n]$ 。请注意， h 和 z_i 两部分分别捕捉了顺序轨迹信息和正常路线信息。需要注意的是，我们没有让 x_n 经过 LSTM，因为它保留了每个路段的正常路线特征。我们采用交叉熵损失来训练基于 z_i 的预测标签 \hat{y}_i 和噪声/改进标签 y_i 之间的 RSRNet，即

$$\mathcal{L} = \frac{1}{n} \sum_{i=1}^n \mathcal{H}(y_i, \hat{y}_i), \quad (1)$$

其中 \mathcal{H} 表示交叉熵算子。

3.6 异常子轨迹检测网络 (ASDNet)

考虑在线异常子轨迹检测的任务是顺序扫描正在进行的轨迹，并且对于每个路段，确定该位置是否发生异常。这促使我们将该过程建模为马尔可夫决策过程 (MDP)，包括状态、行动和奖励。

状态：我们用 s_i 表示扫描路段 e_i 时的状态。状态 s_i ($1 < i < n$) 由 z_i 和 $v(e_{i-1}.l)$ 连接而成，即 $s_i = [z_i; v(e_{i-1}.l)]$ ，其中 z_i 从 RSRNet 中获取， $v(e_{i-1}.l)$ 表示通过嵌入标记（即 0 或 1）获得的上一个路段 e_{i-1} 的标签向量。状态设计的基本原理是从三个方面捕捉特征，即交通环境、正常路线和上一个标签。

行动：我们用 a 表示 MDP 的一个动作，即给每个路段贴上正常或不正常的标签。请注意，当相邻两路段的标签不同时，可以识别出异常子轨迹边界。

奖励：奖励包括两个部分。一个是局部奖励，旨在捕捉路段标签的局部连续性。其原理是正常路段或异常路段的标签不会频繁变化。第二部分称为全局奖励，旨在表示精炼标签的质量（由 RSRNet 的交叉熵损失表示）。我们将损失作为一些反馈来指导 ASDNet 的训练。局部奖励是一种中间奖励，用于鼓励路段标签的连续性。具体来说，它的定义为

$$r_i^{local} = \text{sign}(e_{i-1}.l = e_i.l) \cdot \cos(z_{i-1} = z_i) \quad (2)$$

其中，如果条件 $e_{i-1}.l = e_i.l$ 为真（即标签是连续的），则 $\text{sign}(e_{i-1}.l = e_i.l)$ 返回 1，否则返回 -1； $\cos(z_{i-1}, z_i)$ 表示 z_{i-1} 和 z_i 之间的余弦相似度。我们选择余弦相似度是因为它输出的归一化值介于 0 和 1 之间，与全局奖励的输出范围（即 0 和 1 之间）一致。全局奖励旨在衡量 ASDNet 精炼标签的质量。我们将完善后的标签输入 RSRNet，并计算全局奖励，即

$$r^{global} = \frac{1}{1 + \mathcal{L}} \quad (3)$$

其中 \mathcal{L} 表示等式 1 中的交叉熵损失。我们注意到这个奖励的范围在 0 和 1 之间，这使得 RSRNet 的目标与 ASDNet 非常一致（损失 \mathcal{L} 越小，全局奖励越大）。

MDP 的策略学习。 MDP 的核心问题是学习一个策略，该策略指导智能体根据构造的状态选择行动，以使累积奖励（用 R_n 表示）最大化。我们通过策略梯度方法 [14] 学习策略，称为 REINFORCE 算法 [39]。具体来说，令 $\pi_\theta(a|s)$ 表示随机策略，用于通过神经网络对给定状态 s 的动作 a 进行采样，神经网络的参数用 θ 表示。然后，一些性能度量 $J(\theta)$ 对于网络参数 θ 的梯度估计为

$$\nabla_\theta J(\theta) = \sum_{i=2}^n R_n \nabla_\theta \ln \pi_\theta(a_i | s_i), \quad (4)$$

可以使用优化器进行优化（例如 Adam 随机梯度上升）。轨迹的预期累积奖励 R_n 定义为

$$R_n = \frac{1}{n-1} \sum_{i=2}^n r_i^{local} + r^{global}. \quad (5)$$

RSRNet 和 ASDNet 的联合训练。 我们联合训练两个网络（RSRNet 和 ASDNet）。首先，我们将原始轨迹映射到道路网络上，并生成噪声标签。然后，我们随机抽样 200 个轨迹，分别对 RSRNet 和 ASDNet 进行预训练。具体来说，对于 RSRNet，我们使用噪声标签以监督方式训练网络。

对于 ASDNet，我们将其动作指定为噪声标签，并通过等式 4 计算的梯度上升步骤训练策略网络。预训练为两个网络提供了一个热启动，在联合训练之前，两个网络的参数已经包含了从正常路径中获取的一些信息。在联合训练期间，我们随机采样 10,000 条轨迹，并为每条轨迹对 RSRNet 和 ASDNet 进行 5 次迭代训练。具体来说，我们应用从 ASDNet 中学习到的策略来获得细化标签，并使用细化标签来训练 RSRNet，以获得每个路段的较佳表示 z_i 。然后，用 z_i 构建状态，在 ASDNet 中学习更好的策略。由于学习到的策略可以进一步细化标签，而且两个网络是联合训练的，在此过程中会选择最佳模型。

3.7 RL4OASD 算法

我们的 RL4OASD 算法基于学习的策略来检测异常子轨迹。该过程如算法 1 所示。具体来说，RL4OASD 以在线方式接受持续生成的轨迹，将每个路段标记为正常（即 0）或非正常（即 1）（第 19 行）。它首先根据定义将源路段或目标路段标记为正常路段（第 2-3 行）。然后，它通过调用 RSRNet（第 5-6 行）构建一个状态，并根据 ASDNet 中学习到的策略对动作进行采样以标记路段（第 7-8 行）。它监视路段上涉及异常标签（即 1）的异常子轨迹，并在其形成时将其返回（第 9 行）。最后，如果没有检测到异常，算法将返回 NORMAL 轨迹信号（第 11 行）。我们进一步开发了两项增强功能来提高 RL4OASD 的有效性和效率。一种称为路网增强标签（RNEL），另一种称为延迟标签（DL）。

Algorithm 1: The RL4OASD algorithm

Input: A map-matched trajectory

$T = \langle e_1, e_2, \dots, e_n \rangle$ which is inputted in an online manner

```
1 for  $i = 1, 2, \dots, n$  do
2   if  $i = 1$  or  $i = n$  then
3      $e_i.l \leftarrow 0$ ;
4   else
5     Call RSRNet for obtaining a representation  $\mathbf{z}_i$ ;
6     Construct a state  $\mathbf{s}_i = [\mathbf{z}_i; \mathbf{v}(e_{i-1}.l)]$ ;
7     Sample an action (i.e., 0 or 1),  $a_i \sim \pi_\theta(a|\mathbf{s})$ ;
8      $e_i.l \leftarrow a_i$ ;
9     Monitor the anomalous subtrajectory
       consisting of the road segments with the label
       1 and return the subtrajectory when it is
       formed;
10 end
11 Return a NORMAL trajectory signal;
```

路网增强标签。对于 RNEL，我们利用道路网络的图结构来帮助标记路段，其中路段上的标签在以下三种情况之一是确定性的：(1) 如果 $e_{i-1}.out = 1$, $e_i.in = 1$ ，则 $e_i.l = e_{i-1}.l$ 。(2) 如果 $e_{i-1}.out = 1$, $e_i.in > 1$ 且 $e_{i-1}.l = 0$ ，则 $e_i.l = 0$ 。(3) 如果 $e_{i-1}.out > 1$, $e_i.in = 1$ 且 $e_{i-1}.l = 1$ ，则 $e_i.l = 1$ 。这里， $e_i.out$ 和 $e_i.in$ 分别表示路段 e_i 的出度和入度， $e_i.l$ 表示道路 e_i 的标签段。它们背后的共同原因是：(a) 异常状态从 e_{i-1} 处的正常 (0) 到 e_i 处的异常 (1) 的任何变化都意味着存在从 e_{i-1} 到其他路段的替代过渡（即， $e_{i-1}.out > 1$ ）；(b) 异常状态从 e_{i-1} 处的异常 (1) 到 e_i 处的正常 (0) 的任何变化都意味着存在从其他路段到 e_i 的替代过渡（即 $e_i.in > 1$ ）。根据规则，我们仅通过强化学习模型在其他情况下执行操作，因此可以避免一些潜在的错误决策。此外，由于通过检查规则而不是调用 RL 模型来节省某些情况下采取行动的时间，因此可以提高效率。

延迟标签。对于 DL，只要识别出边界，RL4OASD 就会形成一个异常子轨迹，即，如果 $e_{i-1}.l = 1$ 且 $e_i.l = 0$ ，则边界被识别在 e_{i-1} 处。直观上，形成一个异常子轨迹看起来有点仓促。异常的子轨迹，可能会从目标轨迹中产生许多短碎片。因此，我们将延迟技术视为后处理过程。具体来说，当形成异常子轨迹时，它会扫描 e_{i-1} 之后的 D 个路段。在 D 条路段中，我们选择最终位置 $j(i-1 < j \leq i-1 + D)$ ，其中该路段的标签为 1，然后将位置 $i-1$ 和 j 之间的一些 0 转换为 1。可以验证，延迟标记不会产生太多时间成本，并且提供了更好的连续性，避免形成太多碎片。

时间复杂度。RL4OASD 算法的时间复杂度为 $O(n)$ ，其中 n 表示目标轨迹的长度。时间由两个网络主导，即 RSRNet 和 ASDNet。我们分析如下。在 RSRNet 中，一段路段的时间成本包括 (1) 获取 TCF 和 NRF 嵌入的时间成本，均为 $O(1)$ 和 (2) 通过经典 LSTM 单元获取 \mathbf{z} 的时间成本，即 $O(1)$ 。在 ASDNet 中，一条路段的时间成本包括 (1) 构建状态的时间成本，其中 \mathbf{z} 部分已在 RSRNet 中计算， $\mathbf{v}(e.l)$ 部分通过嵌入层获得，其复杂度为 $O(1)$ (2) 通过学习的策略对动作进行采样，其时间复杂度为 $O(1)$ 。正如我们在算法 1 中看到的，两个网络最多被调用 n 次。因此，RL4OASD 的时间复杂度为 $O(n) \times O(1) = O(n)$ 。我们注意到，对于轨迹或子轨迹异常检测任务， $O(n)$ 时间复杂度保持了现有算法 [8]、[9]、[11] 当前最好的时间复杂度，并且可以很大程度上满足在线场景的实际需求正如我们的实验所示。处理异常轨迹

的概念漂移。正如第三节中所讨论的，我们检测到不遵循正常路线的异常子轨迹。然而，“正常”和“异常”的概念可能会随着时间和交通状况的变化而改变。例如，当某些受欢迎的路线拥堵时，驾驶员可能会逐渐喜欢行驶另一条不受欢迎的路线（即，以避免交通拥堵）。在这种情况下，不受欢迎的路线应被视为正常路线，而先前路线行驶的轨迹可能会变得异常。为了解决正常和异常概念漂移引起的问题，我们采用在线学习策略 [11]，模型继续使用新记录的轨迹数据进行训练，并根据当前的交通状况更新其策略。

4 复现细节

4.1 与已有开源代码对比

本文复现参考了论文作者公开的代码，代码链接 (<https://github.com/lizzyhku/OASD>)。在深入研究源代码的基础上，我进行了一些完善和改进，以进一步提升其功能和性能表现。

4.2 数据集及实验环境搭建

4.2.1 数据集

实验在滴滴出行的成都真实出租车轨迹数据集上进行。通过地图匹配算法 [34]，将所有原始轨迹预处理为地图匹配轨迹，并从 OpenStreetMap 中获得成都的道路网络。根据之前的研究 [11], [8]，我们对数据集进行预处理，过滤那些包含少于 25 条轨迹的 sd 对，使其有足够的轨迹来表示正常路线。我们从数据集中随机抽取 10000 条轨迹用于训练，剩下的用于测试。

4.2.2 实验环境搭建

操作系统: Windows 10

处理器: Intel(R) Core(TM) i7-10700K CPU @ 3.80GHz 3.79 GHz

内存: 16 GB

显卡: RTX 3090 24 GB

Python 版本: 3.6

Tensorflow 版本: 1.8.0

安装必要的 Python 库: gensim, gensim, numpy, networkx, pickle 等

4.3 源码结构分析

data_utils 文件是用于数据预处理的，主要有以下函数：get_signal 用于获取信号，判断部分轨迹是否在参考中。get_initial_labels 获取初始标签。sort_nomal_rep_SD 对正常轨迹进行排序。my_accuracy_whole_determine 计算准确率。my_fscore_whole_determine 计算 F1 分数。SD_Data 处理 SD 数据。SD_distribution 获取 SD 分布。生成中间结果，如 SD_pair_data、SD_pair_time、sample_table、transition_proba1、transition_proba2、w2v_toast 等，并保存到文件。

ASDNet 文件实现了一个强化学习网络，用于在给定环境中学习和执行动作。模型使用了 Actor-Critic 算法，其中 Actor 负责选择动作，而 Critic 负责评估动作的价值。定义了一

个类 ASDNetwork, 初始化方法, 接受强化学习环境的参数, 如动作数量、特征数量、采样轮数等; 设置神经网络的结构, 包括输入层、标签向量、全连接层和输出层; 定义损失函数和训练操作; 初始化 TensorFlow 会话和模型保存器。

ASDNet_tool 文件组成了一个强化学习网络, 该网络通过训练和探索来学习在给定环境中进行决策。其中, 一些函数用于计算奖励、损失以及模型的预测结果。similarity 函数使用点积除以向量范数的方式来计算两个向量的相似度。from_RSR_loss 函数利用模型对观察进行评估, 然后计算估计回报损失。determine 函数根据观察、标签等确定模型的奖励。使用连续性权重计算局部奖励, 并通过 RSR 损失获取全局奖励。explore 函数利用模型进行探索, 计算多轮奖励, 并使用学习方法进行训练。evaluate_asd 函数使用模型对观察和文本进行评估, 返回结果标签。

RSRNet 文件定义了一个基于 TensorFlow 的神经网络模型, 用于实现 RSR 网络。包括了嵌入层、LSTM 层还有输出层等结构。

RSRNet_tool 文件主要包含了一些用于 RSR 模型的训练、评估和推断的辅助函数。refine 函数根据输入路径 route 和输出 output 进行改进, 在路径上对于那些出度为 1 且入度为 1 的节点, 将其输出设置为前一个节点的输出, 最终得到经过改进的输出。train_rsr 函数调用 RSR 模型的训练步骤, 传递批处理数据, 计算损失并返回平均损失和批处理数据。evaluate_rsr 函数调用 RSR 模型的测试步骤, 传递批处理数据, 计算损失并返回总损失。inference_rsr 函数根据输入的文本和信号, 通过 RSR 模型进行推断, 得到隐藏状态。

train 文件主要涉及到对 RSRNet 和 ASDNet 模型进行预训练和训练的过程。在整个过程中, 通过对 RSR 模型和 ASD 模型进行交替训练, 达到对异常子轨迹检测的目标。

evaluation 文件用于加载预训练好的 RSRNet 和 ASDNet 模型, 并在测试集上进行评估, 通过计算 labelling-fscore 来衡量模型的性能。最终, 真实标签和预测标签的结果也被保存下来。

parameter 文件主要是用于设置和定义一些训练过程中需要用到的参数, 在训练代码中, 通过访问 FLAGS 对象, 可以方便地获取这些参数的值。

4.4 创新点

通过仔细观察模型的训练过程, 我发现在 ASDNet 网络模型的热启动几轮训练后, 实验结果已经表现出显著的改善趋势。使用 20% 的数据进行训练后, 模型趋于稳定, 未在结果上进一步提升。鉴于此, 我决定改变最优模型的选择标准, 引入其他指标进行评估, 以更全面地衡量模型性能。

同时, 为了更好地适应复杂的交通场景, 我对 RSRNet 进行了改进, 引入了时间维度的考虑。这是因为在不同的时间段内, 道路交通状况可能会发生变化, 导致异常轨迹概念的漂移。通过加入时间维度, 我旨在使模型更加敏感地捕捉时间对轨迹模式的影响, 从而提高模型在处理复杂交通场景时的适应性。这一改动有助于模型更准确地理解和预测轨迹数据中的动态变化, 为应对实际交通环境中的多变因素提供更为可靠的预测能力。

5 实验结果分析

本部分对实验在成都真实出租车轨迹数据集上进行，下面将展示通过超参数 `delay` 的调节对原文结果的复现、使用不同指标选择最优模型的结果对比和添加时间维度后的结果对比。

超参数 `delay` 的调节对原文结果的复现。如图 2 所示，图中揭示了当超参数 `delay` 设置为 8 时，F1-score 值是最高的，这也验证了作者将超参数 `delay` 设置为 8 的原因。

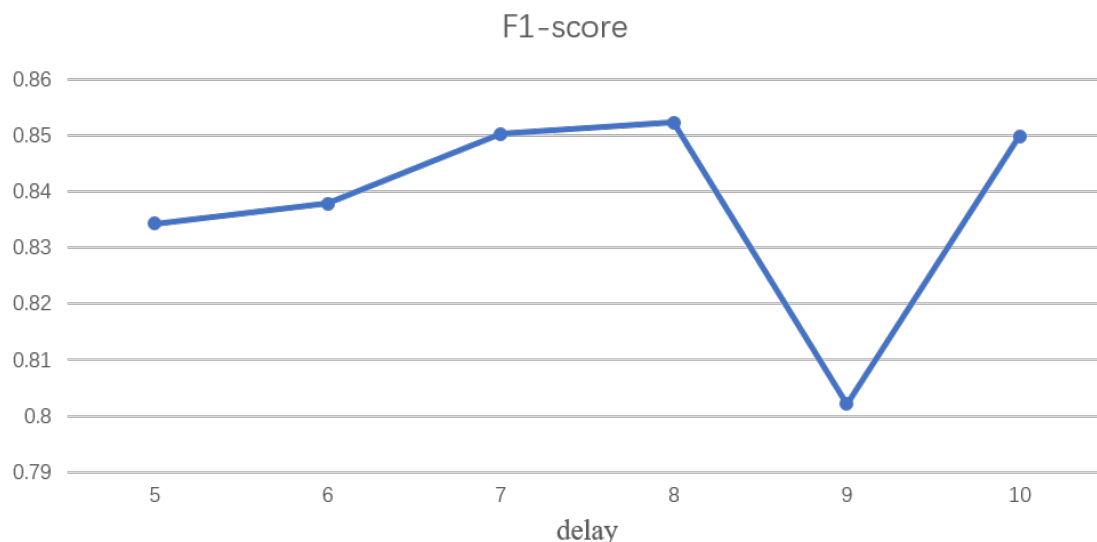


图 2. `delay` 的影响

使用不同指标选择最优模型的结果对比。由于观察训练过程发现，模型在训练前期就已经达到了最优结果，后期的训练都没有提高最终的指标。因此，我将模型选择指标由原来的 F1-score 修改为 Accuracy，希望通过这样的方法，能在模型训练的后半期找出更优的训练结果保存下来。最终的结果如表 1 所示，可以发现，使用 Accuracy 选择出来的模型相对于原文的方法可以在 F1-score 和 Accuracy 上获得略微的效果提升。

表 1. 指标修改前后结果对比

	F1-score	Accuracy
F1-score_origin	0.8523	0.793
Accuracy_our	0.8529	0.798

添加时间维度后的结果对比。在原文的基础上，添加了时间槽的时间维度特征信息，希望通过该方法来提高模型的特征提取能力，获得更多维度的信息，以此来提高强化学习模型的决策能力。在原文的方法中，将所有的轨迹通过时间的不同，分为 24 组，每小时为 1 组，然后通过同一时间槽内的轨迹信息来计算初始噪声标签，然后将所有的轨迹数据一起训练。受时间槽的启发，我将时间槽的信息经过独热编码后，在 RSRNet 中和原始信息进行 concat 操作，然后再送入 ASDNet 中训练。最终的实验结果显示，在实验指标为 F1-score 的情况下，将原文的结果 0.854 提高至 0.861。实验结果表明，时间维度信息可能是该论文模型框架里的一个重要的影响特征。

6 总结与展望

本研究专注于探讨道路网络在线异常子轨迹检测问题，并首次提出了基于深度强化学习的创新解决方案，即 RL4OASD。RL4OASD 采用了一种数据驱动的方法，其独特之处在于它能够在缺乏标记数据的情况下进行训练。我们通过广泛的实验在人工标记异常的真实出租车轨迹数据集上验证了 RL4OASD 的有效性。实验结果表明，RL4OASD 在各种交通条件下均表现出色，始终优于当前已有的算法。此外，RL4OASD 还展现出相对较快的运行速度，为实时应用提供了可行性。

未来的研究方向将聚焦于解决历史轨迹不足的情况下的冷启动问题。这一挑战性问题涉及到在缺乏足够历史数据的情况下，如何使 RL4OASD 在新的道路网络环境中仍然能够高效准确地进行异常子轨迹检测。探讨并开发适用于冷启动情境的方法，以进一步提升该解决方案的实际适用性和普适性。同时，还可以研究如何将时间维度更好的融入网络结构中，或者是寻找其它可能存在联系的特征信息等，以此提高最终的实验效果。

参考文献

- [1] Haitao Yuan, Guoliang Li, Zhifeng Bao, and Ling Feng. An effective joint prediction model for travel demands and traffic flows. In *2021 IEEE 37th International Conference on Data Engineering (ICDE)*, pages 348–359. IEEE, 2021.
- [2] Shanshan Feng, Lucas Vinh Tran, Gao Cong, Lisi Chen, Jing Li, and Fan Li. Hme: A hyperbolic metric embedding approach for next-poi recommendation. In *Proceedings of the 43rd International ACM SIGIR Conference on research and development in information retrieval*, pages 1429–1438, 2020.
- [3] Torsten Seemann, Courtney R Lane, Norelle L Sherry, Sebastian Duchene, Anders Gonçalves da Silva, Leon Caly, Michelle Sait, Susan A Ballard, Kristy Horan, Mark B Schultz, et al. Tracking the covid-19 pandemic in australia using genomics. *Nature communications*, 11(1):1–9, 2020.
- [4] Chao Huang, Junbo Zhang, Yu Zheng, and Nitesh V Chawla. Deepcrime: Attentive hierarchical recurrent networks for crime prediction. In *Proceedings of the 27th ACM international conference on information and knowledge management*, pages 1423–1432, 2018.
- [5] Zhuoning Yuan, Xun Zhou, and Tianbao Yang. Hetero-convlstm: A deep learning approach to traffic accident prediction on heterogeneous spatio-temporal data. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 984–992, 2018.
- [6] Yue Ning, Liang Zhao, Feng Chen, Chang-Tien Lu, and Huzefa Rangwala. Spatio-temporal event forecasting and precursor identification. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 3237–3238, 2019.

- [7] Chao Chen, Daqing Zhang, Pablo Samuel Castro, Nan Li, Lin Sun, and Shijian Li. Real-time detection of anomalous taxi trajectories from gps traces. In *International Conference on Mobile and Ubiquitous Systems: Computing, Networking, and Services*, pages 63–74. Springer, 2011.
- [8] Chao Chen, Daqing Zhang, Pablo Samuel Castro, Nan Li, Lin Sun, Shijian Li, and Zonghui Wang. iboat: Isolation-based online anomalous trajectory detection. *IEEE Transactions on Intelligent Transportation Systems*, 14(2):806–818, 2013.
- [9] Hao Wu, Weiwei Sun, and Baihua Zheng. A fast trajectory outlier detection approach via driving behavior modeling. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, pages 837–846, 2017.
- [10] Dongxiang Zhang, Zhihao Chang, Sai Wu, Ye Yuan, Kian-Lee Tan, and Gang Chen. Continuous trajectory similarity search for online outlier detection. *IEEE Transactions on Knowledge and Data Engineering*, 34(10):4690–4704, 2020.
- [11] Yiding Liu, Kaiqi Zhao, Gao Cong, and Zhifeng Bao. Online anomalous trajectory detection with deep generative sequence modeling. In *2020 IEEE 36th International Conference on Data Engineering (ICDE)*, pages 949–960. IEEE, 2020.
- [12] Li Song, Ruijia Wang, Ding Xiao, Xiaotian Han, Yanan Cai, and Chuan Shi. Anomalous trajectory detection using recurrent neural network. In *Advanced Data Mining and Applications: 14th International Conference, ADMA 2018, Nanjing, China, November 16–18, 2018, Proceedings 14*, pages 263–277. Springer, 2018.
- [13] Xiaolei Li, Jiawei Han, Sangkyum Kim, and Hector Gonzalez. Roam: Rule-and motif-based anomaly detection in massive moving object data sets. In *Proceedings of the 2007 SIAM International Conference on Data Mining*, pages 273–284. SIAM, 2007.
- [14] David Silver, Guy Lever, Nicolas Heess, Thomas Degris, Daan Wierstra, and Martin Riedmiller. Deterministic policy gradient algorithms. In *International conference on machine learning*, pages 387–395. Pmlr, 2014.
- [15] Jae-Gil Lee, Jiawei Han, and Xiaolei Li. Trajectory outlier detection: A partition-and-detect framework. In *2008 IEEE 24th International Conference on Data Engineering*, pages 140–149. IEEE, 2008.
- [16] Daqing Zhang, Nan Li, Zhi-Hua Zhou, Chao Chen, Lin Sun, and Shijian Li. ibat: detecting anomalous taxi trajectories from gps traces. In *Proceedings of the 13th international conference on Ubiquitous computing*, pages 99–108, 2011.
- [17] Jie Zhu, Wei Jiang, An Liu, Guanfeng Liu, and Lei Zhao. Time-dependent popular routes based trajectory outlier detection. In *Web Information Systems Engineering–WISE 2015: 16th International Conference, Miami, FL, USA, November 1-3, 2015, Proceedings, Part I 16*, pages 16–30. Springer, 2015.

- [18] Zhongjian Lv, Jiajie Xu, Pengpeng Zhao, Guanfeng Liu, Lei Zhao, and Xiaofang Zhou. Outlier trajectory detection: A trajectory analytics based approach. In *Database Systems for Advanced Applications: 22nd International Conference, DASFAA 2017, Suzhou, China, March 27-30, 2017, Proceedings, Part I 22*, pages 231–246. Springer, 2017.
- [19] Prithu Banerjee, Pranali Yawalkar, and Sayan Ranu. Mantra: a scalable approach to mining temporally anomalous sub-trajectories. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1415–1424, 2016.
- [20] Kathryn Gray, Daniel Smolyak, Sarkhan Badirli, and George Mohler. Coupled igmm-gans for deep multimodal anomaly detection in human mobility data. *arXiv preprint arXiv:1809.02728*, 2018.
- [21] Yong Ge, Hui Xiong, Chuanren Liu, and Zhi-Hua Zhou. A taxi driving fraud detection system. In *2011 IEEE 11th International conference on data mining*, pages 181–190. IEEE, 2011.
- [22] Xiaolei Li, Zhenhui Li, Jiawei Han, and Jae-Gil Lee. Temporal outlier detection in vehicle traffic data. In *2009 IEEE 25th International Conference on Data Engineering*, pages 1319–1322. IEEE, 2009.
- [23] Yong Ge, Hui Xiong, Zhi-hua Zhou, Hasan Ozdemir, Jannite Yu, and Kuo Chu Lee. Top-eye: Top-k evolving trajectory outlier detection. In *Proceedings of the 19th ACM international conference on Information and knowledge management*, pages 1733–1736, 2010.
- [24] Yanwei Yu, Lei Cao, Elke A Rundensteiner, and Qin Wang. Detecting moving object outliers in massive-scale trajectory streams. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 422–431, 2014.
- [25] Zhihua Zhu, Di Yao, Jianhui Huang, Hanqiang Li, and Jingping Bi. Sub-trajectory-and trajectory-neighbor-based outlier detection over trajectory streams. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 551–563. Springer, 2018.
- [26] Martin L Puterman. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 2014.
- [27] Min-hwan Oh and Garud Iyengar. Sequential anomaly detection using inverse reinforcement learning. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & data mining*, pages 1480–1490, 2019.
- [28] Chengqiang Huang, Yulei Wu, Yuan Zuo, Ke Pei, and Geyong Min. Towards experienced anomaly detector through reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

- [29] Zheng Wang, Cheng Long, Gao Cong, and Yiding Liu. Efficient and effective similar subtrajectory search with deep reinforcement learning. *arXiv preprint arXiv:2003.02542*, 2020.
- [30] Zheng Wang, Cheng Long, and Gao Cong. Similar sports play retrieval with deep reinforcement learning. *IEEE Transactions on Knowledge and Data Engineering*, 2021.
- [31] Zheng Wang, Cheng Long, and Gao Cong. Trajectory simplification with reinforcement learning. In *2021 IEEE 37th International Conference on Data Engineering (ICDE)*, pages 684–695. IEEE, 2021.
- [32] Zheng Wang, Cheng Long, Gao Cong, and Qianru Zhang. Error-bounded online trajectory simplification with multi-agent reinforcement learning. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pages 1758–1768, 2021.
- [33] Richard S Sutton, David McAllester, Satinder Singh, and Yishay Mansour. Policy gradient methods for reinforcement learning with function approximation. *Advances in neural information processing systems*, 12, 1999.
- [34] Can Yang and Gyoza Gidofalvi. Fast map matching, an algorithm integrating hidden markov model with precomputation. *International Journal of Geographical Information Science*, 32(3):547–570, 2018.
- [35] Long Short-Term Memory. Long short-term memory. *Neural computation*, 9(8):1735–1780, 2010.
- [36] Yile Chen, Xiucheng Li, Gao Cong, Zhifeng Bao, Cheng Long, Yiding Liu, Arun Kumar Chandran, and Richard Ellison. Robust road network representation learning: When traffic patterns meet traveling semantics. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pages 211–220, 2021.
- [37] Tobias Skovgaard Jepsen, Christian S Jensen, and Thomas Dyhre Nielsen. Graph convolutional networks for road networks. In *Proceedings of the 27th ACM SIGSPATIAL international conference on advances in geographic information systems*, pages 460–463, 2019.
- [38] Meng-xiang Wang, Wang-Chien Lee, Tao-yang Fu, and Ge Yu. Learning embeddings of intersections on road networks. In *Proceedings of the 27th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pages 309–318, 2019.
- [39] Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8:229–256, 1992.