数据科学导论大作业
基于深度学习的出租车流量预测

姓名：杨云天

学号：10245501405

摘要

为缓解城市出租车运力分布不均与乘客打车难的问题，本项目以 Kaggle 上葡萄牙波尔图市出租车 GPS 轨迹数据集 (Porto Taxi Trajectory Dataset) 为基础, 构建基于 CNN-LSTM 的深度学习模型的出租车净流量预测系统。首先通过数据清洗与预处理，筛选有效特征并提取时间维度信息，同时采用网格化方法将城市核心区域离散为规则网格，将 GPS 轨迹的起点、终点坐标映射至网格单元，形成（时段数，通道数，网格行数，网格列数）的四维时空张量。该张量可同时表征流量的时间周期性与空间分布特征。模型训练阶段，先利用 CNN 提取每个时间步的网格空间特征，再通过 LSTM 捕捉特征序列的时序依赖关系，最终实现对未来 1 小时内各网格流出量与流入量的预测，并计算相应的净流量（流入-流出）。该结果可为出租车运营平台提供运力调度依据，助力从“被动调度”向“主动预测”转型。

关键词：出租车流量预测；深度学习；CNN-LSTM；空间网格化；时空依赖

目 录

1. 项目背景

1.1. 现实困境

出租车是城市公共交通体系的重要组成部分。从早高峰写字楼群的集中用车需求，到节假日枢纽站点的客流潮汐，出租车流量呈现着显著的时空波动特征。传统依赖经验的调度模式往往导致出租车辆易扎堆于已饱和区域，出租车空载现象加剧，司机营收效益受损；而需求旺盛处却因出租车调度存在滞后性，乘客打车难问题频发。

1.2. 项目意义

在此现实困境背景下，依托深度学习技术挖掘出租车流量的时空规律，推动出租车公司从“被动调度”向“主动预测”转变。并提供未来具体时刻各区域的流入流出量预测，方便运营平台提前向司机精准推送热点需求区域信息，引导运力向需求旺盛处前置集结，缓解车辆空驶与乘客打车难问题。

1.3. 技术路线

本项目采用空间网格化方法将城市区域离散化，结合 CNN+LSTM 神经网络模型学习出租车流量的时空特征，实现短期流量预测。

2. 数据集介绍

2.1. 数据来源

数据集：Porto Taxi Trajectory Dataset

来源：Kaggle (https://www.kaggle.com/datasets/crailtap/taxi-trajectory)

2.2. 数据集字段

该数据集包含葡萄牙波尔图市 2013.7.1~2014.6.30 的出租车 GPS 轨迹数据

字段名	数据类型	描述
TRIP_ID	字符串	行程唯一标识符
CALL_TYPE	字符	叫车方式 • A - 通过调度中心派单 • B - 在指定候车点直接叫车 • C - 路边随机叫车
ORIGIN_CALL	整数	呼叫者电话号码的唯一标识符 • 当且仅当 CALL_TYPE 为 A 时有效 • 其他情况下为 NULL
ORIGIN_STAND	整数	出租车站的唯一标识符 • 当且仅当 CALL_TYPE 为 B 时有效 • 其他情况下为 NULL
TAXI_ID	整数	出租车司机唯一标识
TIMESTAMP	整数	行程开始的 Unix 时间戳（单位：秒）

DAYTYPE	字符	行程开始日的类型 • A - 普通日（工作日或周末） • B - 节假日或特殊日 • C - 节假日的前一天
MISSING_DATA	布尔值	数据是否缺失标志 • FALSE - GPS 数据流完整 • TRUE - 存在一个或多个位置点缺失
POLYLINE	字符串	行程的 GPS 坐标序列（WGS84 格式） • 每 15 秒记录一个坐标点 [经度，纬度] • 行程总时间 = (坐标点数量 - 1) × 15 秒

表 2-1

3. 数据清洗与预处理

3.1. 筛选适用的特征

本项目提取以下三个特征进行数据集构建：

字段名	数据类型	适用原因
TIMESTAMP	整数	该字段是揭示流量规律的时间维度核心。依据 Unix 时间戳，可进一步提取日期、小时、工作日/周末，是组织数据并输入深度学习模型的前提，使模型能够学习历史流量如何随时间变化。
MISSING_DATA	布尔值	识别具有缺失 GPS 点的不可靠样本，将其舍弃。若用于训练，会向模型注入噪声，导致预测偏差。
POLYLINE	字符串	该字段是定义流量起点终点的空间维度核心。通过提取轨迹的起点和终点坐标，并将其映射到空间网格中，精确统计出每个网格在特定时间段的车辆流入量和流出量。

表 3-1

3.2. 时间特征提取

从 TIMESTAMP 提取以下特征：

字段名	数据类型	描述	意义
DATETIME	datetime64	转换后的标准日期时间对象 如：2013-07-01 00:00:58	时间对象，为进一步提取时间特征做准备
DATE	字符串	行程的日期 如：2013-07-01	记录训练集与预测的准确日期
HOUR	整数	行程开始的一天内的小时数 如：10（表示 10:00:00-10:59:59）	捕捉日内周期性的关键特征
WEEKDAY	整数	行程发生的一周内的星期几 取值范围为 0（星期一）到 6（星期六）	捕捉周周期性的关键特征（工作日/周末）

表 3-2

3.3. 数据过滤

- 删除 MISSING_DATA == “True” 的行，保障数据质量与模型可靠性，从而提升预测的准确性
- 删除 POLYLINE 字段轨迹点数小于阈值 MIN_POINTS（本报告中定义为 5）的数据，原因：轨迹点过少的行程可能代表极短行程或严重数据缺失，会为模型引入噪声。

4. 模型数据集搭建

4.1. 空间特征提取

4.1.1. 提取起点和终点坐标

- 解析 POLYLINE 字符串
    - 使用 `ast.literal_eval()` 将字符串转换为坐标列表
    - 格式：`[[-8.618643, 41.141412], [-8.618499, 41.141376], ...]`
- 提取首尾坐标
    - 起点：`coords[0]`（轨迹第一个点）
    - 终点：`coords[-1]`（轨迹最后一个点）

4.1.2. 数据转换

将提取结果存储为两个新列：

- `START_POINT`: 起点坐标 [经度, 纬度]
- `END_POINT`: 终点坐标 [经度, 纬度]

4.1.3. 起点终点坐标可视化

通过 folium 库可视化分析，保存为 html/start&end_points.html，将坐标映射到 OpenStreetMap 地图上，观察起点终点分布，为后续手动选取轨迹密集区域、定义网格边界奠定基础。

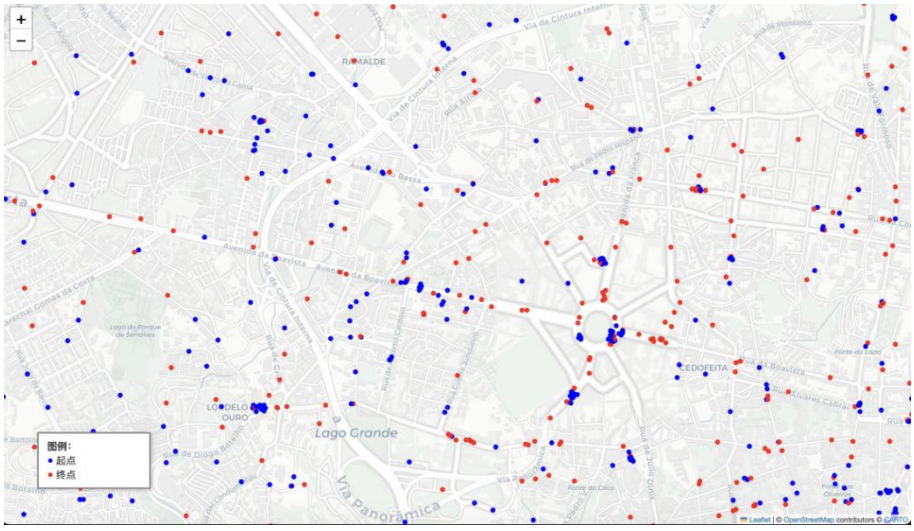

图 4-1

4.2. 空间网格化方案

将连续的地理空间离散化为规则网格，每个网格单元代表一个独立的地理区域，便于统计局部流量密度。

4.2.1. 网格参数设计

定义四个全局变量，分别表示网格左上角和右下角的经纬度坐标，以及网格的边长(grid_size_km)。使轨迹高度密集的地带主要集中在该矩形区间内。

————以下为本报告中定义的数据范围————

```
TOPLEFT_LON = -8.701859（左上角经度）
TOPLEFT_LAT = 41.200615（左上角维度）
BOTTOMRIGHT_LON = -8.543243（右下角经度）
BOTTOMRIGHT_LAT = 41.124366（右下角维度）
grid_size_km = 0.5（每个网格为边长等于 0.5km 的正方形网格）
```

- 网格跨度范围：
  - 南北距离 ≈ 8.46 km
  - 东西距离 ≈ 13.25 km
- 网格数量：
  - 行数 = 南北距离 ÷ 0.5km/格 ≈ 16 行
  - 列数 = 东西距离 ÷ 0.5km/格 ≈ 26 列
  - 总网格数 = 16 × 26 = 416 个

经验证，该区域覆盖了 94.18% 的轨迹点，既包含了核心活动区域，又避免了因边缘稀疏区域导致的网格浪费。

4.2.2. 坐标到网格的映射

通过边界检查与计算相对偏移，将起点终点的经纬度坐标转换为相应的网格索引。

```
- 函数: coord_to_grid(lon, lat) -> (row, col)
    row: 网格行索引（int）
    col: 网格列索引（int）
```

4.3. 数据张量构建

构建四维张量表示时空流量数据：

data.shape =（时段数，通道数，网格行数，网格列数）

- 第一维：时间维度：按（日期，小时）分组聚合，每个时段代表 1 小时的数据
- 第二维：通道维度（3 个通道）

  通道 0：流出量（Outflow）：统计该时段内，以各网格为起点的行程数量

  通道 1：流入量（Inflow）：统计该时段内，以各网格为终点的行程数量

  通道 2：是否周末（IS_WEEKEND）：0/1，帮助模型学习工作日/周末的流量差异模式
- 第三维：空间维度 - 网格的行数
- 第四维：空间维度 - 网格的列数

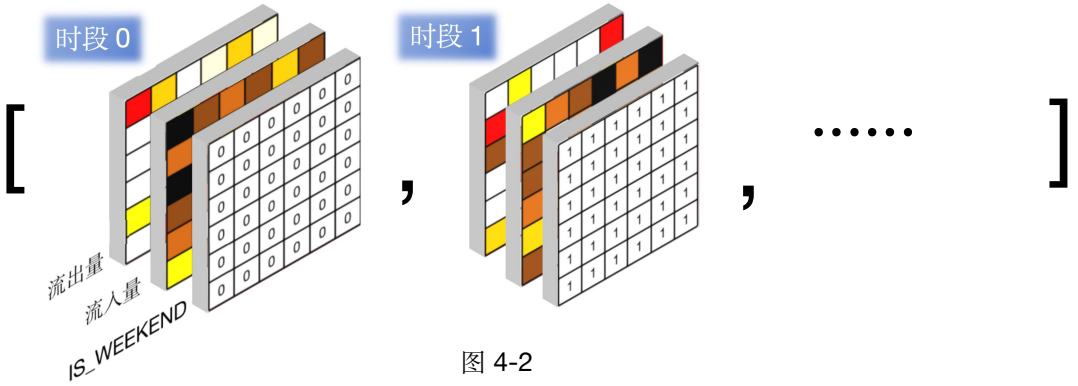

图 4-2

4.4. 时间序列数据创建

本报告的data 形状以（292, 3, 16, 26）为例

- 滑动窗口 seq_length=12，表示用连续 12 个时间步的历史数据作为输入
- 生成输入序列 X

  X_i 的时间范围: [i, i+1, ..., i+11]

  X_i.shape() = (12, 3, 16, 26)

  将 X_1, X_2, X_3...等 280 个样本堆叠后，X.shape() = (280, 12, 3, 16, 26)

  其中，

  第一个维度：样本数（280）

  第二个维度：序列长度（12）

  后三个维度：与原始数据一致（3 通道，16×26 网格）
- 生成目标序列 y

  y_i 的时间: i+12 （X_i 窗口结束后的第 1 个时刻）

  y_i.shape() = (2, 16, 26)

  将 y_1, y_2, y_3...等 280 个样本堆叠后，y.shape() = (280, 2, 16, 26)

  其中，

  第一个维度：样本数（280）

  后三个维度：与原始数据一致（2 通道：流出/流入，16×26 网格）

4.5. 数据归一化

- 通道 0/1（流出/流入）分别归一化，避免相互影响
- 通道 2（周末标记）不归一化，保持 0/1 取值

4.6. 训练/测试集划分

采用时序划分方法：

- 训练集：前 70%

```
X_train = (196, 12, 3, 16, 26),
y_train = (196, 2, 16, 26)
```

- 验证集：中间 10%

```
X_val = (28, 12, 3, 16, 26),
y_val = (28, 2, 16, 26)
```

- 测试集：后 20%

```
X_test = (56, 12, 3, 16, 26),
y_test = (56, 2, 16, 26)
```

5. 模型搭建与训练

5.1. 模型优势

5.1.1. CNN 可提取网格的空间特征

data 数据具有明确的空间维度（网格行数 H、网格列数 W）和通道信息，类似图像的空间分布特性。CNN 通过卷积核在局部空间范围内提取特征（比如相邻区域存在交互等关联特征），配合池化层压缩空间维度同时保留关键空间信息，非常适合处理这种网格状的空间结构数据。

5.1.2. LSTM 可捕捉时间序列特征

data 数据的第一维度是“时段数”，即同一空间网格在不同时间步的观测值，构成了时序序列。净流量未来的变化依赖于历史状态，LSTM 作为专门处理时序数据的模型，能长期记忆关键历史信息，适合捕捉这种跨时间步的依赖关系。

5.1.3. CNN 与 LSTM 结合，同时建模“空间-时序”双重依赖

CNN-LSTM 的“先空间后时序”流程实现优势互补——先以 CNN 提取每个时间步的空间特征，通过卷积核捕捉网格内相邻区域关联，池化层压缩并保留关键特征，将高维网格数据 (batch, C, H, W) 转化为低维特征向量 (batch, 1536) ；再由 LSTM 处理这些特征构成的时序序列，捕捉时间特征，既避免高维数据直接输入时序模型的冗余，又不丢失空间信息，协同建模时空双重属性。

5.2. CNN-LSTM 模型架构

5.2.1. 主要逻辑

操作	输入形状	输出形状	说明
对每个时间步执行 CNN			
取单时间数据	(batch=4, seq_len=12, C=3, H=16, W=26)	(batch=4, C=3, H=16, W=26)	遍历每个时间步
Conv1	(batch, 3, 16, 26)	(batch, 32, 16, 26)	channal: 3 -> 32 kernal = 3×3 stride = 1 padding=1
BatchNorm1	(batch, 32, 16, 26)	(batch, 32, 16, 26)	对32个通道分别做归一化，加速训练
ReLU+dropout	(batch, 32, 16, 26)	(batch, 32, 16, 26)	引入非线性特征，防止过拟合
MaxPool	(batch, 32, 16, 26)	(batch, 32, 8, 13)	kernal = 2×2 H, W 减半
Conv2	(batch, 32, 8, 13)	(batch, 64, 8, 13)	channal: 32 -> 64 kernal = 3×3 stride = 1 padding=1
BatchNorm2	(batch, 64, 8, 13)	(batch, 64, 8, 13)	对64个通道做归一化，加速训练
ReLU+dropout	(batch, 64, 8, 13)	(batch, 64, 8, 13)	引入非线性特征，防止过拟合
MaxPool	(batch, 64, 8, 13)	(batch, 64, 4, 6)	kernal = 2×2 H, W 减半
Flatten (展平)	(batch, 64, 4, 6)	(batch, 64×4×6=1536)	将空间维度展平为一维向量
堆叠时间步执行 LSTM			
堆叠时间步	[seq_len=12 个	(batch=4, seq_len, 1536)	将seq_len=12个时

	(batch, 1536)]		间步的 CNN 输出堆叠为 LSTM 输入
LSTM	(batch, seq_len, 1536)	(batch, seq_len, 64)	2 层 LSTM, hidden_size=64
取最后时间步	(batch, seq_len, 64)	(batch, 64)	提取 LSTM 最后一个时间步的输出作为序列特征总结
Fc (全连接层)	(batch, 64)	(batch, 2×H×W=832)	输出至目标维度
Reshape (重塑)	(batch, 832)	(batch, 2, H, W)	恢复为网格形状，2 个通道对应预测目标（流出/流入）

表 5-1

5.2.2. 模型架构图

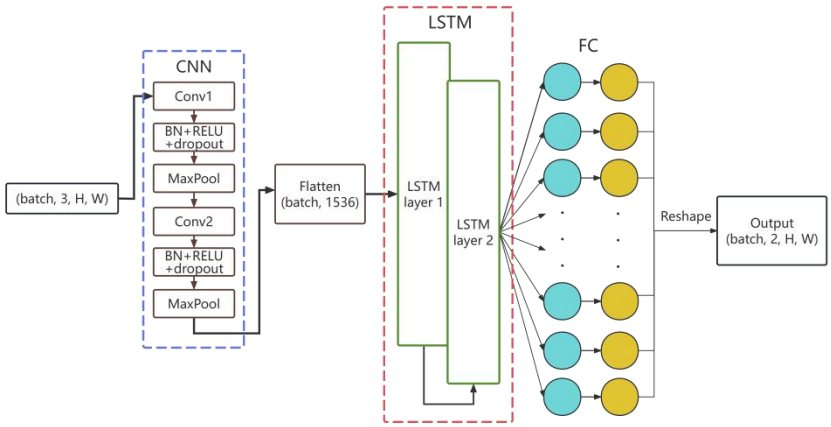

图 5-1

5.3. 模型训练

5.3.1. 训练配置

- 损失函数：采用均方误差损失 MSELoss
- 数据加载：训练集与验证集的 batch_size = 4，为保持时序连续性，数据加载时不打乱顺序（shuffle=False）。

5.3.2. 训练过程

- 训练：epochs = 50 轮，同时引入早停和学习率下降策略以避免过拟合和无效训练：
    - 早停：patience = 10；学习率下降：factor=0.5, patience=3, min_lr=1e-5
- 记录：每轮通过训练集更新模型参数，再在验证集上评估损失，并记录（下图）：

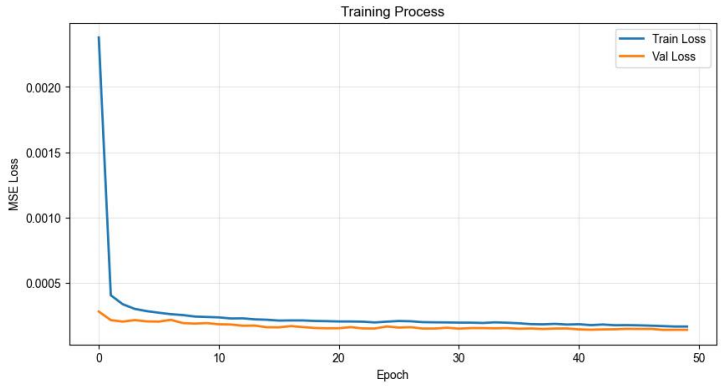

图 5-2

5.3.3. 模型保存
训练完成后，保存文件至 model/flow_model.pth，包含以下关键信息：
- 模型权重（model_state_dict）；
- 优化器状态（optimizer_state_dict）；
- 训练历史（train_losses、test_losses）；
- 模型结构参数（通道数、网格尺寸、CNN 滤波器数量、LSTM 隐藏层维度与层数等），用于模型重构与复用。

6. 净流量预测

基于最近的历史时段序列，预测下一个时段各网格的流出量与流入量，并计算净流量（流入−流出），同时在地图上可视化。

6.1. 模型加载
- 加载模型文件 flow_model.pth，包含模型权重、优化器状态、训练历史及结构参数
- 将模型设置为评估模式，关闭训练相关的正则化机制，确保预测一致性

6.2. 模型预测
- 输入准备：选取测试集的最后 12 个时段的观测值（测试集是时间上最新的数据，包含最新模式和趋势），shape: (12, 3, 16, 26)，作为模型的输入序列
- 维度调整：为输入数据添加 batch 维度，shape: (1, 12, 3, 16, 26)
- 模型预测：在关闭梯度计算的情况下，通过模型输出预测结果，shape: (1, 2, 16, 26)，包含流出量和流入量两个通道
- 维度调整：移除 batch 维度，shape: (2, 16, 26)
- 数据处理：对流出量和流入量分别进行反归一化处理，恢复至原始数据尺度，并将模型预测出的接近 0 的负数值置为 0（符合流量数据含义）
- 计算净流量：通过"流入量 - 流出量"得到 16×26 网格的净流量矩阵

6.3. 预测结果与可视化
6.3.1. 生成净流量矩阵
生成下一个时段的按网格划分的净流量矩阵（净流量保留到整数），对应 16×26 网格的每个单元。

6.3.2. 热力图可视化

- 预测下一个时段所有网格的流出/流入量

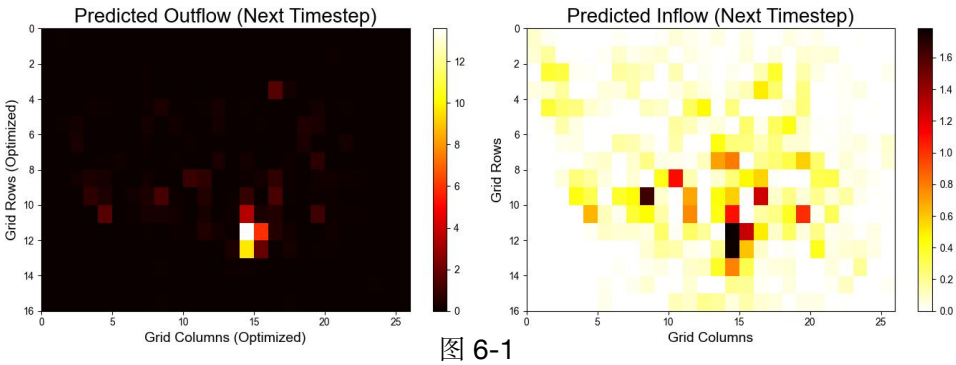

图 6-1

- 计算每个网格的净流量 = 流入量 - 流出量

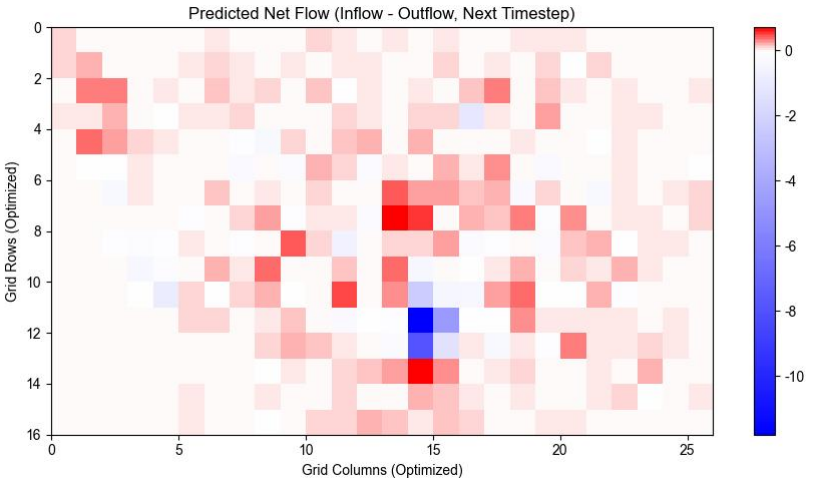

图 6-2

6.3.3. OpenStreetMap 可视化

- 使用 folium 库叠加网格，将网格热力图可视化到真实地图中。

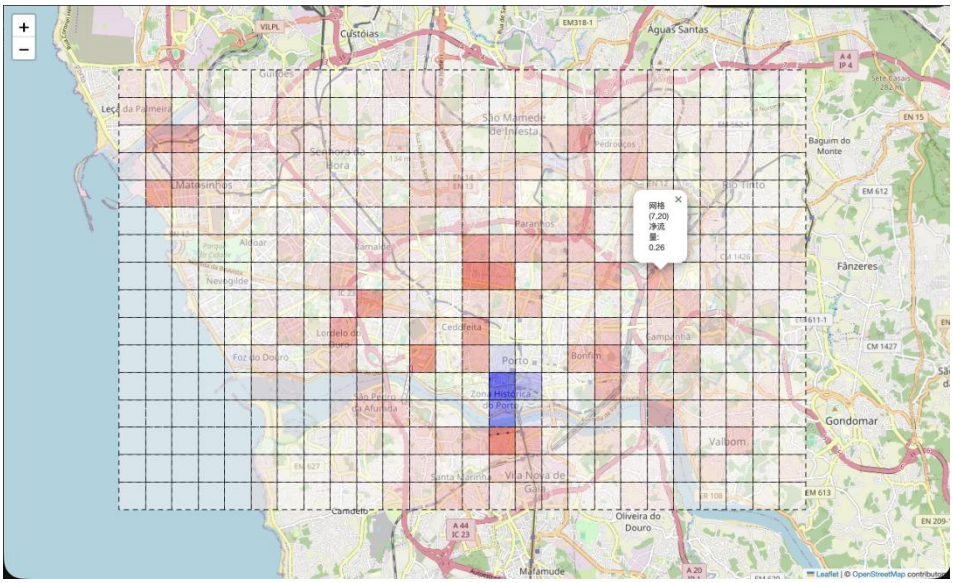

图 6-4

7. 项目总结与展望

7.1. 项目成果总结

本项目围绕出租车流量时空预测需求，完成了从数据集搭建、模型训练与预测。

- **数据标准化处理体系搭建**：基于 Porto Taxi Trajectory Dataset，建立了“特征筛选-时间提取-数据过滤-空间映射”的数据处理流程，剔除噪声样本，提取出包含日期、小时、星期几的时间特征，并通过网格划分，将绝大多数的轨迹点映射到像对较少的网格单元中，为后续模型训练提供了高质量时空数据基础。

- **时空数据张量构建**：将出租车流量数据转化为四维张量（时段数×通道数×网格行数×网格列数），既保留了流量的空间分布特征，又融入了时间周期性信息，满足深度学习模型对结构化输入的需求。

- **预测任务与可视化**：通过 CNN+LSTM 的模型架构，进行时空数据的训练，并对未来 1 小时时段内的净流量进行预测，可支撑提前 30 分钟至 1 小时的出租车调度需求。

7.2. 项目不足点分析

7.2.1. 数据维度与场景适配性局限

数据仅依赖出租车 GPS 轨迹，未融入外部影响因素（如天气数据、节假日、交通拥堵情况等），导致模型无法捕捉非流量本身的干扰因素，例如雨天打车需求激增、大型活动导致的局部流量突变等场景，预测结果的泛化性受限。

7.2.2. 空间网格灵活性不足

网格划分采用固定尺寸的矩形设计，未结合城市区域功能差异而动态调整——核心商圈、交通枢纽等区域道路密度高、人口与用车需求集中，固定尺寸网格难以满足该类区域对流量统计精度的更高要求。

同时，划分过程未参考实际地理分界（如河流、山地等自然屏障会阻断车辆流动）与行政区划分（调度通常以行政单元为基础协同管理），导致部分网格跨越功能边界或自然阻隔，增加了运营平台依据预测进行区域运力调配的操作难度。（例如，右图下方红色网格与蓝色网格距离最近，但由于河流阻隔，路程遥远，调度缺乏合理性。）

图 7-1

7.2.3. 模型结构不足

- 缺乏空间注意力机制。

现有 CNN 对网格数据的空间特征提取是“无差别”的，未考虑不同网格区域对预测目标的重要性差异。卷积核在全局范围内均匀滑动，可能忽略关键空间位置的细粒度特征，导致空间特征提取的针对性不足。

- 时序依赖捕捉的局限性。

本模型采用 CNN-LSTM 的固定流程，LSTM 处理的是经 CNN 压缩后的全局空间特征向量序列，未区分不同时间步中空间区域之间的关联（例如，t 时刻 A 区域的流量变化与 t+1 时刻 B 区域的流量变化可能存在依赖关系）。

7.3. 未来优化方向

7.3.1. 数据维度拓展与特征增强

补充外部数据集，包括实时天气数据（降雨、温度）、城市活动日历（演唱会、展会、体育赛事）、交通路况数据（道路拥堵指数、临时管制信息），并将这些特征融入四维张量的通道维度中，让模型学习多因素协同影响下的流量规律。

7.3.2. 空间网格划分优化与模型选择

尝试通过 DBSCAN 聚类算法对城市区域进行智能分簇，实现核心商圈等流量密集区域的精细划分与郊区等流量稀疏区域的粗放划分，同时融合自然地理边界与行政管辖边界优化簇群划分逻辑。

在此基础上，引入图卷积神经网络（GCN），利用其对非欧几里得空间数据的建模优势，充分挖掘簇群间的空间关联与依赖关系。其中，有权图可以刻画空间距离与连通性：依据道路通达性、跨江绕行成本等因素定义边权，从而更精准地建模簇群之间的空间依赖关系。基于此开展模型训练与流量预测，进一步提升划分合理性、实用性及预测精度。

7.3.3. 引入注意力机制

- 引入空间注意力机制

在 CNN 模块中加入注意力层，让模型自动聚焦于对预测更重要的空间区域（如高流量区域），加权增强关键区域特征。

- 加入“空间-时序”交叉注意力

不同时间步中空间特征存在关联，引入交叉注意力机制以提取这些时空数据关联。