



大学生论文检测系统

文本复制检测报告单(全文标明引文)

No:ADBD2023R_20230531161745472705103498

检测时间: 2023-05-31 16:17:45

篇名: 城市交通流的异常检测平台

作者: 马浩原 (11912919;计算机科学与工程系;计算机科学与技术)

指导教师: 宋轩

检测机构: 南方科技大学

提交论文IP: 110.***.***.***

文件名: 马浩原_毕业论文.pdf

检测系统: 大学生论文检测系统

检测类型: 大学生论文

检测范围: 中国学术期刊网络出版总库

中国博士学位论文全文数据库/中国优秀硕士学位论文全文数据库

中国重要会议论文全文数据库

中国重要报纸全文数据库

中国专利全文数据库

图书资源

优先出版文献库

大学生论文联合比对库

互联网资源(包含贴吧等论坛资源)

英文数据库(涵盖期刊、博硕、会议的英文数据以及德国Springer、英国Taylor&Francis 期刊数据库等)

港澳台学术文献库

互联网文档资源

源代码库

CNKI大成编客-原创作品库

机构自建比对库

时间范围: 1900-01-01至2023-05-31

检测结果

去除本人文献复制比: 0.7%

跨语言检测结果: -

去除引用文献复制比: 0.7%

总文字复制比: 0.7%

单篇最大文字复制比: 0.4%

重复字数: [145]

总段落数: [2]

总字数: [19449]

疑似段落数: [2]

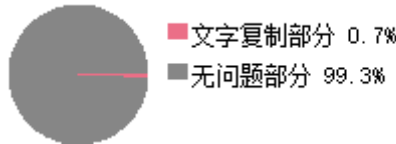
单篇最大重复字数: [79]

前部重合字数: [36]

疑似段落最大重合字数: [79]

后部重合字数: [109]

疑似段落最小重合字数: [66]



指标: ☐ 疑似剽窃观点 ☐ 疑似剽窃文字表述 ☐ 疑似整体剽窃 ☐ 过度引用

相似表格: 0

相似公式: 没有数据

疑似文字的图片: 0

0.7% (66)

0.7% (66) 城市交通流的异常检测平台 第1部分 (总9794字)

0.8% (79)

0.8% (79) 城市交通流的异常检测平台 第2部分 (总9655字)

(注释: 无问题部分 文字复制部分 引用部分)

指导教师审查结果

指导教师: 宋轩
审阅结果:
审阅意见: 指导老师未填写审阅意见

1. 城市交通流的异常检测平台_第1部分 总字数: 9794

相似文献列表

去除本人文献复制比: 0.7%(66) 文字复制比: 0.7%(66) 疑似剽窃观点 (0)

1	易雪枫_毕设论文 易雪枫 - 《大学生论文联合比对库》- 2021-06-10	0.4% (35) 是否引证: 否
2	考虑Jerk的汽车油耗预测模型与速度优化方法研究 张立成(导师: 赵祥模) - 《长安大学博士论文》- 2022-11-03	0.3% (30) 是否引证: 否

原文内容

本科生毕业设计（论文）
题目: 城市交通流的异常检测平台
姓名: 马浩原
学号: 11912919
系别: 计算机科学与工程系
专业: 计算机科学与技术
指导教师: 宋轩副教授
2023 年 5 月 30 日

诚信承诺书

1. 本人郑重承诺所呈交的毕业设计（论文），是在导师的指导下，独立进行研究工作所取得的成果，所有数据、图片资料均真实可靠。
2. 除文中已经注明引用的内容外，本论文不包含任何其他人或集体已经发表或撰写过的作品或成果。对本论文的研究作出重要贡献的个人和集体，均已在文中以明确的方式标明。

3. 本人承诺在毕业论文（设计）选题和研究内容过程中没有抄袭他人研究成果和伪造相关数据等行为。
4. 在毕业论文（设计）中对侵犯任何方面知识产权的行为，由本人承担相应的法律责任。

作者签名:

年月日城市交通流的异常检测平台马浩原
(计算机科学与工程系指导教师: 宋轩)

[摘要]: 随着物联网技术 (Internet of Things, IoT) 的迅速发展及普及，我们与城市环境互动的方式已经被彻底改变。对于构建智慧城市而言，城市交通流的预测与异常检测是智能交通系统中至关重要的一个领域。但是，交通数据的强实时性和时空大数据的复杂性使其成为了一项极具挑战性的任务。我们项目的目标是开发一个城市交通流数据分析与异常检测平台。与基于长时间间隔与时间跨度进行批处理的解决方案不同，我们将使用 Apache-Flink 流计算框架处理和分析实时数据，来监控出租车和共享自行车的流入流出指标。同时集成基于时空的深度学习交通流预测算法进行异常检测。

[关键词]: 流式系统; 深度学习; 异常检测; 交通流

I

[ABSTRACT]: With the rapid development and popularization of Internet of Things (IoT) technology, the way we interact with urban environments has

been completely changed. For building a smart city, predicting and detecting anomalies in urban traffic flow is a crucial area in intelligent transportation systems. However, the strong real-time nature of traffic data and complexity of spatiotemporal big data make it an extremely challenging task. Our project aims to develop a platform for analyzing and detecting anomalies in urban traffic flow data. Unlike previous solutions based on long timespan and time intervals, we will use Apache-Flink stream computing framework to process and analyze real-time data to monitor inflow/outflow indicators for taxis and shared bicycles. We will also integrate deep learning-based spatiotemporal traffic flow prediction algorithms for anomaly detection.

[Key words]: Streaming System, Deep Learning, Anomaly Detection, Traffic Flow

II

目录

1. 前言 1

2. 相关工作 3

2.1 时空图神经网络与相关数据集 3

2.2 异常检测方法 4

2.3 流式系统 5

2.4 小结 6

3. 方法与设计 6

3.1 数据源 6

3.2 数据采集 7

3.3 流式计算 9

3.4 异常检测 11

3.5 前端可视化 13

4. 实现与结果 13

4.1 参数设定 14

4.2 主要实现 15

4.3 结果分析 16

4.4 应用部署 19

5. 结论 20

参考文献 22

致谢 25

III

1. 前言

随着物联网技术 (Internet of Things, IoT) 的迅速发展及普及，我们与城市环境互动的方式已经被彻底改变。近年来，移动电话、汽车导航系统和交通传感器产生的时空数据急剧增长，为城市尺度上的交通建模分析和相关应用提供了许多机会。对于构建智慧城市而言，城市交通流的预测在交通调度、人流管理和公共安全等各个方面均发挥着重要的作用。准确的交通流预测与分析，可以为我们的城市规划和交通管理提供重要的洞见与理解。公共交通效率将会得到显著的提高，同时也会减少交通拥堵和碳排放，进而为居民带来更好的生活质量。

图 1 交通流预测的应用交通流的异常检测是智能交通系统中至关重要的一个领域。交通异常指的是区域内的交通流量与正常情况下出现明显差异，这些差异可能源自于交通拥堵或交通事故等问题。针对交通异常进行实时监测和预警是提高城市交通管理效率的重要手段。通过不断积累和分析交通数据，交通管理者可以及早发现和解决道路瓶颈、交通事故等问题，优化交通规划和调度，提高城市交通的运营效率和服务水平。此外，在公共安全层面的紧急情况下，成功预测城市交通可以缩短应急服务响应时间，进而提升资源分配与调度的效率，更好地保障城市居民的生命安全和财产安全。

为了实现智慧城市意义下交通流预测与异常检测等目的，我们主要采用机器学习

习和深度学习等方法。具体来说, 我们可以将交通流预测的任务建模为: 给定 N 个不重叠的地区划分 $\{r_1, r_2, \dots, r_N\}$, 我们旨在构建一个可学习的模型 F , 具有可学习参数 θ , 在过去 α 个时间段的基础上, 预测未来 β 个时间段内 N 个地区的交通流入 1

流出情况。具体的预测机制可以用如下式子表示:

$$F \rightarrow \begin{bmatrix} X_{t-\alpha+1}, \dots, X_t \\ X_{t+1}, \dots, X_{t+\beta} \end{bmatrix} \quad (1)$$

其中 $X_i \in \mathbb{R}^N \times C$, C 表示预测目标的通道数量 (不同来源的数据或交通流量)。

而对于异常检测, 我们实时获取每个区域过去 α 个时间段的流量信息, 并利用交通流预测模型得到每个区域下一个时间段的预测值 \hat{X}_{t+1}

。然后, 我们将其与观测到的真实值进行比较。如果这两个值之间的差异

$$|\hat{X}_{t+1} - X_{t+1}| > \gamma \quad (2)$$

超过了预先设定好的阈值 γ , 那么我们认为该区域可能发生异常的交通情况, 可以进行相应的响应和处理。

虽然城市交通预测与异常检测可以带来很多好处, 但交通数据的强实时性和时空大数据的复杂性使其成为了一项极具挑战性的任务。为了解决这个问题, 流式计算平台吸引了我们的关注。流式计算平台是一种逐步处理原始数据流的技术工具, 具有较强的实时性, 可以在数据流传输的过程中对其进行实时处理和分析, 为城市交通预测与异常检测提供了更加快速和高效的数据处理手段。相较于批处理方式, 数据在流

式计算平台上得到即时处理, 计算结果也几乎是实时可见的。这些数据处理技术在处

理复杂的时空大数据中具有很好的优势。因此, 在城市交通预测领域, 流式计算平台的应用已经成为了一个重要趋势。

基于以上因素, 我们项目的目标是开发一个城市交通流数据分析与异常检测平

台。与基于长时间间隔与时间跨度, 进行批处理与深度学习的解决方案不同, 我们将使用 Apache-Flink 流式计算框架处理和分析实时数据, 来监控出租车和共享单车的流入流出指标。同时集成基于时空的深度学习交通流预测算法进行异常检测, 检测安全问题, 并在需要紧急响应的情况下提高城市应对能力。这种创新的分析平台有望显著改善城市交通预测与异常检测的能力, 为实现更智能、更高效和可持续发展的城市做出贡献。2

2. 相关工作

随着交通流预测和智慧城市的快速发展, 许多深度学习模型与流式计算框架被发布用来解决城市交通预测和异常检测问题。我们对一些典型的模型与方法进行简要介绍。

2.1 时空图神经网络与相关数据集

考虑到现实世界中的非欧几里得属性, 利用图卷积网络 (GCN) 捕获数据的空间依赖性会获得更加精确的效果。在实践中, 我们会把每个节点 (即区域或传感器) 中的关系表示为邻接矩阵, 并输入到 [GCN 中。STGCN\[1\]是最早使用图神经网络预测交通流量的模型之一。](#)该模型将 GCN 和门控时间卷积网络 (TCN) 结合起来作为时空卷积单元完成此任务。然而, STGCN 只适用于对称邻接矩阵, 因此在邻接关系不对称情况下表现不佳。此外, TGCLSTM[2]、T-GCN[3]和 DCRNN[4]也应用了 GCN 来提取交通图中的空间特征, 并使用 RNN 的变体来捕获多步交通流量的时间依赖性。DCRNN

是这种类型模型中最具代表性的模型之一。它提出了扩散卷积操作, 可以捕获节点之

间双向关系。进一步地, 在 Graph WaveNet[5]、AGCRN[6]、MTGNN[7]、SLCNN[8]和 EAST-Net[9]中采用了自适应学习技术来学习动态可训练图形以替换固定图形。此外,

注意力增强 GCN 在 GaAN[10]、ASTGCN[11]和 STGNN[12]的空间建模中也表现出更好的性能。GCGA[13]将生成对抗网络 (GAN) 和自编码器与 GCN 相结合。有关深度交通模型的更多详细信息, 请参阅全面的文献综述

[14-15]。

下面是关于交通流预测常用数据集的一些介绍。对于交通速度预测,有几个广泛使用的公开数据集是从高速公路交通传感器中收集的。文献[4]发布了 METR-LA,其中包含了来自洛杉矶 207 个高速公路传感器的交通速度数据,时间跨度为 2012 年 3

月 1 日至 2012 年 6 月 30 日期间的 4 个月。PeMS-BAY[4]和 PeMSD7M[1]是两个从加利福尼亚州运输机构绩效测量系统 (PeMS) 收集的交通速度数据集。前者包含 325

个传感器,时间跨度为 2017/1/1 到 2017/5/31;后者包含 228 个传感器,时间跨度为

2012/5/1 到 2012/6/30。由于交通传感器部署在道路网络中,因此自然而然地可以利用图的数据结构来表示这些数据。因此,它们经常用于评估基于 GNN 的深度学习模型和多元的时间序列模型。另一方面,还有几种常用于运输需求预测的开放式数据集, 3

例如出租车 / 自行车需求、流入或流出等:北京出租车数据集由[16]发布,在 2013 年至 2016 年之间;纽约市 (NYC) 自行车流量数据集在[16-18]中使用;以及由[18]发布的

2015 年纽约市出租车数据。此外,[19]还发布了东京和大阪的大规模人群密度和进出流量数据集。上述数据集都基于网格结构,因此它们往往被许多基于 CNN 的模型使用。

2.2 异常检测方法

对于交通流的异常检测,存在一些基于机器学习与深度学习的方法。其中一种方法是使用如 K-means[20]和 Autoencoder[21]来分析特定时间段的交通序列,识别数据中隐藏的异常模式。K-means 是一种聚类算法,可以将交通流数据划分成若干个簇,从而识别出异常点。具体过程可以通过对正常的交通流数据进行聚类,然后将新的数据点与各簇的质心进行距离计算,如果其与某个质心的距离超过了一定的阈值,就可以判定该点为异常点。这种方法对于规律性异常点进行较好的检测,但对于一些不规律的异常点则效果较差。[22]

而 Autoencoder 则是一种无监督的深度学习方法,可以用于对数据进行特征提取

和降维,从而更好地发现异常点。在对交通流数据进行处理时,可以先将数据通过 Autoencoder 进行编码,然后通过再解码的过程来重构数据,如果某些数据点重构的误差超过了阈值,则可以判定该数据点为异常点。[23] 相比于 K-means 方法,Autoencoder 方法的效果更加稳定,可以对于不规律性异常点进行较好的检测。

另一个选择是采用监督学习的方法。与无监督学习不同,监督学习则借助预训练过的模型和历史数据来进行异常检测。主要思路是对交通流预测的模型进行预训练,根据历史数据预测未来时间戳的值。当有新数据点或事件发生时,我们将其与预测值进行比较,并判断差异是否超过一定阈值。如果是,则假设这里发生了异

常。目前最先进的方法是一种利用时空图卷积对抗网络 (STGAN) 来解决上述问题。

STGAN 包含一个时空生成器来预测正常的交通动态,并设计了一个时空鉴别器来确定输入序列是否真实存在。经过对抗训练后,生成器和鉴别器可以独立使用作为检测器,其中生成器建模正常的车流动态模式,鉴别器则根据其变化与位置信息提供检测标准。[24] 4

2.3 流式系统

在实时数据的处理中,流式系统有着举足轻重的地位。流式系统是一种计算系统,主要用于处理和分析连续生成的数据流。与传统的批处理系统不同,流式系统可以实时或接近实时地处理数据。这种系统的特点是可以接受连续不断的输入,并在处理过程中产生连续不断的输出,常用于处理高速、连续的数据流,如社交媒体数据、

金融交易数据、网络日志数据、物联网 (IoT) 数据等。这些数据流是连续生成的,有时候还有非常高的更新频率,需要在短时间内进行处理和分析。交通数据处理是流式系统的一个重要应用领域。实时交通数据,如车辆位置信息、交通信号状态、道路拥堵情况等,都是连续生成的数据流。这些数据需要实时或接近实时地处理和分析,以便提供实时的交通信息,如路况预测、交通拥堵警报、最佳路线建议等。流式系统正好可以满足这种需求,因此广泛应用于交通数据的处理中。这些系统处理连续不断的交通事件和交通流数据,提供实时的交通信息和决策支持。此类系统的核心技术包括

了许多流式计算平台,如 Hadoop[25], Spark[26]和 Apache-Flink[27]等。

Hadoop,由 Apache 基金会开发,是一个开源的分布式计算框架,它允许在物理集群中的大量计算机节点上存储和处理大量的数据。研究[28]使用了基于 Hadoop 的大数据平台来处理道路交通数据,并提出了一种新的基于时空相关性的实时预测模型,使得能够准确预测道路拥堵的发生和持续时间。然而,Hadoop 主要是设计用来进行批处理,对于实时数据流处理有一定的局限性。Spark,也是 Apache 基金会的开源项目,相较于 Hadoop,它提供了更强大的实时处理能力。Spark 的核心是一个计算引擎,它支持高级的数据处理任务,包括 SQL 查询,流处

理，机器学习和图计算等。研究[29]基于大数据和深度学习的技术开发了一个实时的交通预测模型，基于 Spark 处理大规模数据，实现了更高的预测精度和更快的预测速度。但是，尽管 Spark 支持流处理，但其设计初衷并非专门针对连续的数据流，其流处理是基于微批处理模型的。

Apache-Flink 则是当前最先进的流式计算框架之一。它的核心理念是“流优先”，意味着 Flink 在设计之初就考虑到了连续数据流的处理。Flink 不仅可以处理无限的数据流，还可以处理有限的数据流，即批处理。它提供了强大的故障恢复能力，能够确保在处理大规模实时数据流时的系统稳定性。此外，Flink 还提供了全面的工具来对 5 流处理的过程进行监视和维护，这对于实时交通流处理中的性能调优和问题诊断非常有用。研究[30]提出一种基于低秩张量分解的实时城市交通预测模型。该模型基于Flink 实现了分布式计算，可以更加高效地进行实时预测

在实际应用中，这些流式计算平台可以结合使用，比如 Spark 和 Flink 可以在Hadoop 的基础上运行，利用 Hadoop 的分布式存储能力，同时发挥各自的计算优势。

对于实时交通流处理，Flink 的流优先设计理念和强大的故障恢复能力使其成为一个非常合适的选择。

2.4 小结

总而言之，这些先进的算法和技术大大增强了我们对城市交通和实时数据处理的理解，并为实现更智能、更高效和可持续的城市计算带来了许多机遇。通过不断创新和发展，这些算法和技术有望进一步推动城市交通领域的发展，为城市居民带来更好的交通与生活质量。

3. 方法与设计

我们开发的实时城市交通流的异常检测平台的设计和实现分为以下五个部分，分别是：数据源、数据采集、流式计算、存储与中间件、异常检测和前端可视化。系统的架构示意图如图2所示。下面是各个模块的具体介绍。

3.1 数据源

在数据来源方面，我们采用了一份基于区域划分的开放数据集，其覆盖了 2019 年 1 月 1 日至 2020 年 12 月 31 日的时间跨度。该数据集来源于纽约市（NYC）的出租车和共享单车的数据，以模拟实时的交通事件。这个开放数据集基于区域划分，可以有效地整合提取现实世界中的出租车和共享单车的流入、流出、始发-到达流（Origin-Destination，简称 OD Flow）等多种运输模式。因此，该数据集非常适合于基于图神经网络的一些模型进行时空交通流预测和异常检测。表1提供该数据集原始数据的全面概述，包括 NYC 交通数据集的名称、数据描述、数据来源、空间-时间域以及列（如时间戳和 GPS 位置）。此外，所有字段的数据分布也如图3所示。

在我们的应用程序中，在对数据进行分析与处理时，我们采用了流式计算和批处理结合的计算方法。流式计算用于处理实际交通事件，利用交通流构成的天然时间序 6

图 2 系统架构设计

表 1 数据源概述

Data	Data Description / Data Source / Data Column	Spatial Domain	Temporal Domain
BikeNYC	Bike inflow, outflow, and odflow Bike trip data in New York City (https://www.citibikenyc.com/system-data) Start/end timestamp, start/end station ID/latitude/longitude, etc	69 regions	2019/1/1 - 2020/12/31
TaxiNYC	Taxi inflow, outflow, and odflow Taxi trip data in New York City (https://www1.nyc.gov/site/tlc/about/data.page) Pickup/dropoff timestamp, pickup/dropoff location ID (region ID), etc		

列来判断交通模式并识别潜在异常。另一方面，我们还使用批量计算，通过对进出流量在不同时间间隔进行聚合的数据进行分析，以检测异常情况。这种技术需要在一段时间内收集流式数据，并计算聚合统计信息，以进行全面的分析。通过流式计算和批处理计算的结合，我们的应用可以充分挖掘数据集的时空信息。

3.2 数据采集

在数据采集模块中，我们利用消息队列来模拟处理交通事件的流式输入。具体而言，我们使用 Kafka[31]作为消息队列。Kafka 是一个开源的分布式流处理平台，其因其速度、可扩展性和容错能力等特点而变得越来越受欢

迎。使用 Kafka 作为消息队列的关键优势之一是它的高容量数据处理设计，非常适合大数据应用程序。由于我们项

目的数据来源具有高容量特点，利用 Kafka 作为消息队列可以确保数据处理的有效

7

图 3 数据源的分布情况性。

对于我们的项目，数据源的格式为 CSV（逗号分隔值）。我们首先需要在 Python

中对数据进行一些预处理，过滤掉异常值，将其按照时间排序，然后作为生产者写入到 Kafka 中。通过在将数据输入到 Kafka 之前进行排序，我们可以确保按时间顺序高

效地处理数据，并在整个分析过程中维护交通流数据的实时性与一致性。

Kafka 的使用为我们的项目带来了多重好处。一方面，Kafka 实现了解耦，将我们的模块和数据源分离。软件工程中的最佳实践之一就是解耦应用程序不同组件。在我们的项目中，生产者接口负责提供纽约市自行车和出租车数据点，而消费者接口负责使用 Flink 进行数据处理。这种解耦设计提供了灵活性，可以根据项目要求或未来的需求简单地更改数据源和平台。另一方面，Kafka 支持集群，以满足高并发和大数据的需求。Kafka 集群是一组服务器，共同存储和处理数据。这种集群化提供了多个好处，例如高可用性、可扩展性和容错能力等。Kafka 集群易于扩展，非常适合处理

大量数据。在未来，我们也可以引入集群，以处理更大规模的数据。综上所述，使用 Kafka 提供了一个强大的数据平台，可以帮助我们更好地处理数据以提高项目的效率和可扩展性。

图4显示了我们应用程序中 Kafka 的示意图。8

图 4 系统中 Kafka 的架构示意

3.3 流式计算

在我们的应用程序中，关键的模块是流式计算平台，它能够处理实时交通事件。

Apache Flink 是一个先进的开源流处理系统，具有部署灵活性和可伸缩性。它可以轻松地集成到现有基础架构或云环境中，在大数据任务的应用程序中发挥强大的作用。

Flink 拥有多项与其他系统不同的关键功能。例如，它的应用程序编程接口（API）

由 DataStream 用于流计算和实时事件处理，以及 Dataset 用于批处理组成。DataStream

API 特别适用于实时数据处理和管理交通事件，而 Dataset API 则非常适合复杂数据处理，如聚合和机器学习。此外，Flink 还提供了多种针对机器学习算法的扩展库，并提供分析异常检测数据的可能性。Flink 的并行能力允许在不同的插槽之间进行多任务处理，并确保最大吞吐量和最小延迟。这使得开发人员可以轻松配置系统，并根据特定情况进行调整，以实现优化生产力。Flink 的架构如图5所示。

接下来，我们将讨论 Flink 中的数据流水线，这是我们应用程序中核心的交通数据处理步骤。Flink 的数据流架构旨在轻松处理批量和流式数据。该流程始于 Source，其中将数据摄入系统。源头可以是集合、文件、套接字和用户定义的源，而在我们的应用程序中，我们使用 Kafka 提供了无边界的交通事件流，使其非常适合依赖于流式传输的应用程序。

一旦数据被摄入，就会通过多个流操作（例如过滤和映射）进行操作。这些操作 9

图 5 Flink 的架构示意[32]

通过 filter() 操作删除不合理的交通事件，然后通过 map() 操作将事件转换为系统所需要的适当格式来清理交通数据。keyby() 是另一个基本操作，它用来标记事件的时间戳。时间戳指向事件的唯一属性（例如位置、速度和时间），从而实现对数据进行唯一性的处理。Flink 还提供聚合和窗口函数，以便于使用滚动窗口和滑动窗口等机制计算一组时间间隔内的指标值。例如，在计算一定时间间隔内的交通流入与流出情况时，聚合的窗口函数 window() 与 groupby() 将持续汇总当前时间段内所有选定区域流入流出的交通时间数量，并提供关于该时间段内净交通如何变化的综合视图。

经过处理后，处理结果将会被输出到 Sink 端以结束整个过程。Sink 操作负责将

已处理好的数据持久化，以便进行进一步的分析或在系统外进行后续处理。在我们的项目中，我们将数据保存到 MySQL 中进行长期存储，并将其加载到 Elasticsearch 进行索引，提升对于热点数据的搜索效率。

总体而言，Flink 的数据流架构为批量和流式处理之间提供了高效且平滑的转换方式，非常适合处理大量数据。该框架出色地支持各种流操作、高效窗口化以及受欢迎 source 和 sink 等特性，将交通数据方面的处理转换变得非常简单。交通数据处理

结果不仅准确而且易于检索和分析。下图6显示了我们的数据处理管道。

10

图 6 数据流式处理管道[33]

3.4 异常检测

在我们的数据经过 Kafka 输入，以及 Flink 的流式处理之后，我们将热点数据保存到 Elasticsearch 中，进行后续的异常处理等分析。

在给定纽约市 (r1, r

2, ..., r

N

) N 个非重叠区域集合中，我们的目标是构建一个具有可学习参数 θ 的模型 F。

2. 城市交通流的异常检测平台_第2部分

总字数：9655

相似文献列表

去除本人文献复制比：0.8%(79) 文字复制比：0.8%(79) 疑似剽窃观点 (0)

1	不同羽色鹌鹑Tyr基因表达差异分析 李国锐 - 《大学生论文联合比对库》 - 2021-06-03	0.8% (79) 是否引证：否
2	仙鹤图案在现代服装中的创新设计 孙敏(导师：陈文成) - 《青岛大学硕士论文》 - 2022-05-30	0.3% (33) 是否引证：否

原文内容

该模型将根据过去 α 个时间槽预测未来 β 个时间槽中每个区域自行车和出租车流入/流出情况。

$[X_{t-\alpha+1}, \dots, X_t$

$-1, X_t]$

$F \rightarrow$

θ

$[\hat{X}_t$

$+1, \hat{X}_{t+1}, \hat{X}_{t+2}, \dots, \hat{X}_{t+\beta}]$

在这个背景下， $X_i \in \mathbb{R}^N \times \mathbb{C}$ ，其中 \mathbb{C} 表示预测目标的数量（自行车流入、自行车流出、出租车流入、出租车流出），而 N 则代表空间区域的数量。一旦我们获得了每个区域的预测值 \hat{X}_{t+1}

，我们就会将其与真实值进行比较。如果这两个值之间的差异

$|\hat{X}_{t+1} - X_{t+1}|$

$> \gamma$

超过了事先设定好的阈值 γ ，那么我们将考虑可能发生异常情况。

为了实现该算法，一种有效的方法是创建一个 Python 接口，该接口可以包括各种算法或预训练深度学习模型（如 STGCN[1]、DCRNN[4] 和 Graph-WaveNet[5]），以实现精确预测。在这里，我们选择预训练的 DCRNN 模型，在我们的数据集上，他的时空预测效果已经被证明可靠。同时，基于该模型，我们采用多种性能增强方法，包括多源 (Multi-Source)，多图 (Multi-Graph)，元信息 (Meta-Information) 等策略。[34]。

因为来自不同来源的交通信息可能会相互影响并具有隐藏的关系。例如，出租车 11

和自行车的流动可以相互影响，某种车辆的流入和流出也可能存在相关性。多源策略

(Multi-Source) 的想法是合并来自不同来源的交通信息，改善输入张量通道，从而使模型输入包含更多信息以进行特征提取和依赖捕获。多图策略 (Multi-Graph) 是为了充分考虑区域间的多种相关性，将不同邻接矩阵中包含的各种信息相结合，提取不同的空间特征，并基于这些不同的空间结构进行预测。例如将包含简单邻接关系的 0-1

矩阵和包含长时间跨度内区域之间的流量 OD 矩阵相结合，作为 GCN 模型的图输入进行时空预测。最后，元信息策略 (Meta-Information) 可以进一步提高模型的有效性。众所周知，交通很容易受到一些元信息的影响，例如天气、特殊事件和一天或一

周中的时间。利用这些外部信息来辅助增强时空建模是直观的。我们将时间戳标记为元数据（即 `dayofweek`、`hourofday`、`isholiday`），并对其进行 one-hot 编码。将日期、星期和该日是否为假期编码成 32 维向量。这个想法类似于多源策略，其中元信息作为

额外来源来增强预测结果。以上三种策略的示意如图7所示。

图 7 基本预测结构与增强策略的示意图利用 DCRNN 模型以及多种增强方法，我们可以利用历史的交通数据预训练深度学习模型。在这里，我们将 2019 年 1 月 1 日到 2019 年 1 月 31 日的数据作为训练集，进行模型的预训练。在实时数据的模拟中，我们采用 2019 年 2 月 1 日的数据作为测试集，进行异常检测。我们的预训练方法接受传入的 `XS` 和 `YS` 作为对应监督学

习的特征和相应标签。其中 `XS` 是维度为 $B \times T_i$

$n \times N \times C$ 的张量，这里 B 表示训

练批数据的大小， T_i

n

表示 RNN 模型输入的时间槽数量， N 表示 GNN 模型输入的图节点个数， C 表示用于训练的数据通道数量。

由于我们需要处理出租车与共享单车 12

流入流出，取 $C = 4$ 。`YS` 是维度为 $B \times T_o$

$ut \times N \times 1$ 的张量， T

`out`

表示 RNN 模型输出的时间槽数量，我们需要预测的目标为单一交通流，所以通道数量为 1。在训练过程中，

我们将 `XS` 和 `YS` 转换成 torch 张量，并使用它们创建 `trainval_data`、`train_data`

和 `val_data` 数据集以及相应的迭代器 `train_iter` 和 `val_iter`。之后，我们进行 EPOCH

次迭代训练。对于每次迭代，计算不同的 MSE, RMSE, MAE 和 MAPE 等指标，并在

`train_iter` 上进行优化器更新操作。同时评估验证集上的损失值 `val_loss`，并保存最小的 `loss` 所对应的模型权重，作为预训练的模型。

在系统中，我们会封装运行模型进行在线推理的函数，进行模型的加载与推断，得

到预测结果。在这个过程中，会设置随机种子以实现可重复性。通过读取配置文件中的

模型超参数获取 DCRNN 模型加载到内存中，并且通过 Pytorch 的 `load_state_dict()`

函数加载此前针对出租车与共享单车数据进行预训练的模型参数。流数据数组作为输入，我们使用 Scikit-learn 库中的 `Scaler` 对输入数据进行标准归一化处理，消除数据分布的差异，并将其转换成 Pytorch 的四维张量格式 $B \times T \times N \times C$ 。载入到之前

的模型中进行预测。最后再次使用 `Scaler` 对输出结果进行反向恢复得到真实值，并将

预测结果转换为整数，堆叠在一起作为输出返回。需要注意的是，我们在函数的执

行过程中禁止梯度计算以提高效率和节省内存空间。

通过 4 个预训练的模型组合，我们可以在出租车与共享单车流入流出的实时系统中达到在线推断的效果，进行一定时空维度的预测与异常检测。

3.5 前端可视化

在上文中，我们已经概述了我们的应用的后端设计思路，介绍了从数据处理，数据检索和存储，以及算法检测的全面思路。除了后端，我们还将为数据管理和可视化开发了一个高效的前端界面。为此，我们选择了 Kibana，这是一个开源的数据可视化平台，可提供详尽的图形化表示数据。这种集成将进一步简化操作流程，并使监控和分析数据更加容易，可显著提高数据管理和可视化方面的效率。

4. 实现与结果

下面，我们将会对平台的工程实现和结果做一些讨论。

13

4.1 参数设定

我们的代码实现中采用了如表2中所列的超参数。为进行异常检测，我们应用了DCRNN 模型，并将历史观测时间步长（`TIMESTEP_IN`）设置为 12 作为输入，预测时间步长（`TIMESTEP_OUT`）设置为 3 作为输出。这意味着我们使用过去 12 个时间段的观测数据来预测未来 3 个时间段的交通需求，并对未来一个时间段的预测值与观测值进行异常检测的比较。`N_NODE` 代表 GNN 模型中图的节点个数，根据纽约曼哈顿的数据集，该参数被设置为 69。

对于 2019 年到 2020 年的数据集, 我们使用 2019 年 1 月的数据作为深度学习

模型的预训练数据集, 使用 2019 年 2 月 1 日的数据作为模拟实时数据的测试集。

INITIAL_TIMESTAMP 代表最初的时间, 这里我们使用了 2019-02-01 00:00:00。INITIAL_TIMESTEP 代表系统开始运行异常检测的时间步数。因此, 在系统的开始时间步被设置为 12, 代表我们提前加载 0-11 个时间步的数据, 在第 12 个时间步开始异常检测的分析计算。

TIME_INTERVAL 代表我们对每个时间段设置的时间长度。为了满足实时性的要求, 我们将其设置为 5 分钟。在实际层面, 我们预先加载了 2019-02-01 00:00:00 至

2019-02-01 01:00:00 这 12 个时间步的交通流量数据, 并在第 12 个时间戳, 即 2019-

02-01 01:00:00 进行第一次异常检测, 之后每隔 5 分钟进行一次这样的计算。

TIME_SCALE 代表我们在模拟系统对时间加速的倍数。在本实验中, 我们将其设置为 60 倍, 即每运行 5 秒更新一次数据。DETECT_THRESHOLD 代表我们运行异常检测的检测阈值, FLOW_TYPES 代表我们在实际中采用的数据流量, 可以根据需要进行配置。总而言之, 我们的实验基于出租车与共享单车的进出流量, 时间间隔为 5 分钟, 并以此模拟实时的数据输入与处理。 14

表 2 系统中参数的设定

Parameters Values

TIMESTEP_IN 12

TIMESTEP_OUT 3

N_NODE 69

INITIAL_TIMESTAMP 2019-02-01 00:00:00

INITIAL_TIMESTEP 12

TIME_INTERVAL 5

TIME_SCALE 60

DETECT_THRESHOLD 10

FLOW_TYPES bike_inflow, bike_outflow, taxi_inflow, taxi_outflow

4.2 主要实现

我们的系统主要运行逻辑在算法1中所示

Algorithm 1: The main logic of our system

Data: current timestep t , flow array $flow$ with shape $(T, N, C) = (12, 69, 4)$

Result: Real-time traffic flow and anomalies in Kibana

1 $t \leftarrow \text{INITIAL_TIMESTEP}$;

2 $flow \leftarrow \text{load_flow}(t)$;

3 while True do

4 $prediction \leftarrow \text{run_model}(flow)$;

5 $t \leftarrow t + 1$;

6 $\text{sleep}(\text{TIME_INTERVAL} * 60 / \text{TIME_SCALE})$;

7 $flow \leftarrow \text{load_flow}(t)$;

8 $ground_truth \leftarrow flow[-1]$;

9 $\text{detect_anomaly}(ground_truth, prediction)$;

10 end

在我们的程序中, 主要涉及到四个关键的函数, 它们分别是 `load_flow()`、`run_model()`、`detect_anomaly()`, 以及两个辅助函数 `read_from_ES()` 与 `write_to_ES()`, 用于从 Elastic-search 中读取与写入数据。

首先, `load_flow()` 函数是负责数据加载的核心部分。该函数的作用是从 Elastic-

search 中加载当前时间步之前的 12 步数据, 并更新当前的流量数据 (`flow`)。为了避免数据重复加载和计算, 提高数据读取的效率, 我们设计了一个类似于滑动窗口的机制。具体来说, 仅在初始时间步 (`INITIAL_TIME STEP`) 时, 我们会加载全部的 12 步 15

数据。然后, 在每一个更新周期中, 我们只需将滑动窗口向前移动一步, 加载一个新的时间步的数据。这种设计能够极大地提高数据读取和处理的效率。

接下来，run_model() 函数是用于运行深度学习模型，进行交通流量预测和异常检测的核心部分。在这个函数中，我们通过读取配置文件，加载预训练的深度学习模型。然后，我们用这个模型进行在线推理，获取下一个时间步的预测值。在完成预测

后，程序会暂停，模拟时间窗口的推进，等待下一轮的数据加载和预测。

detect_anomaly() 函数则负责比较预测值（prediction）与实际值（ground_truth），并计算可能的异常区域信息。在这个函数中，我们使用一些统计方法或者机器学习算法，来识别那些与预测值差距较大的实际值，这些值可能表示交通流量的异常情况，然后将识别到的异常区域信息写入到 Elasticsearch 中。

最后，我们的辅助函数用于进行 ES 的操作，例如 write_to_ES()，用来将异常检测的结果写入 Elasticsearch。这个函数会将异常区域的信息，如位置、时间、预测值、实际值等，写入 Elasticsearch。在 Elasticsearch 中，我们可以方便地对这些数据进行查询和可视化，帮助我们更好地理解和分析交通流量的异常情况。

4.3 结果分析

首先对预训练的模型的训练结果与性能表现作分析。我们给出了 DCRNN 与多种增强策略在自行车和出租车数据集上的结果，并列出了时间步数为 1、2、3 时的不同表现。性能的评估标准包括 MAPE、MAE 和 RMSE。具体如表3所示。在结果中，我们将基础的 DCRNN 的性能列为基线，三种增强方法（Multi-Source，Multi-Graph，Meta-Information）超过基线的 MAPE 的结果在此表中用粗体标出。经过计算可以发现，与基准 DCRNN 模型的 MAPE 指标相比，多源策略的 MAPE 下降了 6.89%，Multi-Graph 策略的 MAPE 下降了 4.65%，Meta-Information 策略的 MAPE 下降了 2.75%。因此，多源策略具有最好的改进效果，我们在实现中采取这种模型增强方法。

然后对平台的可视化与异常检测的结果做一些展示。我们的平台基于 Kibana 提供了丰富的可视化功能和实时的异常检测结果，主要以热力图，等值线图为主，辅以柱状图和表格说明其他相关信息。这些都是基于实时的交通数据流动态生成的。图8向我们展示了自行车流量数据的可视化结果。左下角柱状图中包含了各个区

16

表 3 模型性能表现

DCRNN

NYC Bike Inflow NYC Bike Outflow NYC Taxi Inflow NYC Taxi Outflow

Horizon Horizon Horizon Horizon

Method Metrics 1 2 3 1 2 3 1 2 3 1 2 3

Baseline

MAPE 42.6346 50.7661 58.5621 44.6016 53.8079 62.8287 20.3592 19.5046 19.9000 18.1738 17.7856 18.1

919

MAE 7.3910 9.6459 11.4638 7.6119 9.8538 11.5503 12.3003 12.0602 12.3934 10.9309 10.6898 11.0004

RMSE 12.0919 16.6117 19.8238 12.3588 16.4575 19.0149 21.9841 21.6861 22.3543 18.3649 18.3998 19.1

277

Multi

Source

MAPE 41.2595 44.6673 49.0068 40.3554 43.6320 47.6464 19.3911 18.8880 19.4047 18.0847 17.5670 18.0

066

MAE 7.6618 9.3698 11.9164 7.3707 9.5447 10.9075 11.7860 11.5029 11.8937 10.6481 10.3835 10.7161

RMSE 12.7054 16.1704 19.5833 11.9186 16.2096 18.3425 20.7566 20.3835 21.2178 17.9110 17.8361 18.5

601

Multi

Graph

MAPE 41.7348 49.9950 57.2736 44.6169 55.1529 64.8835 19.3479 18.7136 18.8574 17.2768 16.5611 16.5

363

MAE 7.6022 9.7422 11.3288 8.2095 10.5856 12.3937 11.5867 11.3287 11.5363 10.3059 9.9997 10.1416

RMSE 12.5623 16.8064 19.6936 13.5001 17.6083 20.6380 20.6944 20.3344 20.7884 17.1432 17.0001 17.5

491

Meta
Information

MAPE 43.3114 51.6457 59.4778 42.4803 50.5702 57.9207 19.5746 18.9642 19.3782 18.0071 17.5551 17.9

372

MAE 7.5207 9.5333 11.0998 7.4369 9.3567 10.8800 12.1256 11.9834 12.4457 10.8650 10.7060 11.0324
RMSE 12.1956 15.8595 18.3901 12.0224 15.3896 17.8190 21.7002 21.7728 22.6920 18.2931 18.4464 19.2

216

域的自行车流量统计数据，从中我们可以清楚地看到在不同的区域内，自行车的流量分布情况。此外，右侧图中还显示了原始的自行车数据点的流动情况，通过这种方式，我们能够直观地观察到自行车在城市中的流动模式，比如在哪些地方自行车的流动较为集中，哪些地方则相对较少。

图 8 自行车流量的可视化结果

图9则展示了出租车流量数据的可视化结果。左侧直观的显示了出租车流量数据的热力图统计。右侧的两张图包含了对各个区域的出租车流量的统计与可视化。我们可以看到在某些区域，如商业区或交通枢纽等，出租车的流量通常会比较大。 17

图 9 出租车流量的可视化结果

图10展示了可能出现流量异常的区域统计汇总。在这张图中，右侧表格将可能出现异常的区域 ID，区域名称以及相应的异常类型展示了出来。而在左侧纽约曼哈顿区域的地图中，异常的地区和流量状况也会被实时显示，使得我们能够迅速地定位到异常发生的位置。 18

图 10 异常检测结果

对于上图在曼哈顿地区进行交通流量异常检测的结果，我们分析可以得到了以下结论：系统显示在凌晨 2 点左右，曼哈顿某些区域的 taxi_outflow 出现了异常。通过对异常区域进行进一步的分析，我们可以发现这些区域主要位于曼哈顿的市中心的夜生活娱乐区，如时代广场，中心城等。另外，当前模拟数据的时间戳是 2019 年

2 月 1 日，通过网络了解到当天是星期五，前一天晚上在曼哈顿地区发生了集会庆祝

活动。[35] 因此这些区域在凌晨时段仍然有许多人员流动，导致出租车的出行需求量较大，出现了 taxi_outflow 异常的现象。

这一发现对城市交通管理部门具有一定的参考意义，可以帮助我们更好地了解城市交通流量的变化趋势，并采取相应的措施进行规划和管理。例如，可考虑增加凌晨的公共交通服务，并在夜间加强交通管制和稽查，以确保城市道路保持畅通。

4.4 应用部署

本项目的后端部署依赖于多种技术和框架，包括 Java，MySQL，Apache-Flink，Zookeeper，Kafka，Python，以及 Pytorch。我们利用这些工具和框架，构建了本系统。Java 作为一种通用的高级编程语言，是 Flink，Kafka，Zookeeper 等框架的底层依赖。MySQL 是我们选择的数据库系统，我们使用它来存储和管理数据。Apache-Flink 是强大的流式计算框架，我们使用它来处理实时的交通数据。Zookeeper 和 Kafka 都是分布式系统中的重要组件，Zookeeper 用于维护系统的配置信息和服务状态，Kafka用于实现实时数据的模拟与高效传输。

Python 是我们用来编写数据处理和分析代码的语言，它的简洁性和强大的数据处理库使得快速实现复杂的数据处理任务变得可能。Pytorch 则是我们用来构建和运行深度学习模型的框架。

为了方便部署，我们选择了 Docker 这个强大的容器平台。通过 Docker，用户可以将所有的依赖和应用打包到一个容器中，这使得在本地部署和运行本系统变得非常简单和快速。只需几个简单的命令，就可以在任何安装了 Docker 的机器上运行我们的系统。

我们还使用了 Elasticsearch 和 Kibana 这两个强大的工具，来实现数据的存储、查询和可视化。这两个工具都已经在 Elasticsearch Cloud 平台上成功部署，用户可以通过互联网轻松地访问和监控实时的交通数据。

更多关于系统的部署和代码实现的细节，都可以在我们的 GitHub 仓库中找到。

仓库地址是：<https://github.com/Evens1sen/anomaly-traffic-stream>。欢迎任何对我们项目感兴趣的人访问本仓库，参考其中的代码，或者对我们的项目提出改进建议。

5. 结论

我们的项目构建了一个实时计算平台,该平台专门用于监测城市中的出租车和共享单车的流量数据,并能够实时地检测出可能发生的异常情况。在整个过程中,我们充分结合和利用了当前最先进的时空预测技术,同时也充分考虑了城市交通数据的实时性和动态性。

首先,我们选择了 Apache Flink 这个强大的流计算框架作为我们平台的核心。

Apache Flink 能够提供低延迟和高吞吐量的流数据处理能力,使得我们能够实时地监测和处理大量的交通数据。同时,我们也采用了深度学习算法来进行异常检测。深度学习算法能够从大量的数据中学习出复杂的模式,这使得我们的系统能够及时地捕捉到可能出现的异常情况,从而为城市交通管理提供了有力的支持。20

总的来说,我们构建的这个实时计算平台在智慧城市和智能交通系统中将起到关键的作用。通过实时地处理和分析交通数据,我们的平台能够快速响应用户的需求和城市可能出现的交通安全问题。例如,我们的平台可以及时地预测出交通繁忙的区域和时间段,从而帮助城市管理者进行更有效的交通规划。同时,我们的平台也能及时地检测出可能出现的交通异常,这对于提前预防交通事故和保障城市交通的安全具有重要的意义。2参考文献

- [1] YU B, YIN H, ZHU Z. Spatio-temporal graph convolutional networks: a deep learning framework for traffic forecasting[C]//Proceedings of the 27th International Joint Conference on Artificial Intelligence. 2018: 3634-3640.
- [2] CUI Z, HENRICKSON K, KE R, et al. Traffic graph convolutional recurrent neural network: A deep learning framework for network-scale traffic learning and forecasting[J]. IEEE Transactions on Intelligent Transportation Systems, 2019, 21(11): 4883- 4894.
- [3] ZHAO L, SONG Y, ZHANG C, et al. T-gcn: A temporal graph convolutional network for traffic prediction[J]. IEEE Transactions on Intelligent Transportation Systems, 2019, 21(9): 3848-3858.
- [4] LI Y, YU R, SHAHABI C, et al. Diffusion Convolutional Recurrent Neural Network: Data-Driven Traffic Forecasting[C]//International Conference on Learning Representations. 2018.
- [5] WU Z, PAN S, LONG G, et al. Graph wavenet for deep spatial-temporal graph modeling[C]//IJCAI. 2019: 1907-1913.
- [6] BAI L, YAO L, LI C, et al. Adaptive graph convolutional recurrent network for traffic forecasting[J]. Advances in Neural Information Processing Systems, 2020, 33: 17804- 17815.
- [7] WU Z, PAN S, LONG G, et al. Connecting the dots: Multivariate time series forecasting with graph neural networks[C]//Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. 2020: 753-763.
- [8] ZHANG Q, CHANG J, MENG G, et al. Spatio-temporal graph structure learning for traffic forecasting[C]//Proceedings of the AAAI Conference on Artificial Intelligence: vol. 34: 01. 2020: 1177-1185.
- [9] WANG Z, JIANG R, XUE H, et al. Event-Aware Multimodal Mobility Nowcasting [J]. arXiv preprint arXiv:2112.08443, 2021.
- [10] ZHANG J, SHI X, XIE J, et al. Gaan: Gated attention networks for learning on large and spatiotemporal graphs[J]. arXiv preprint arXiv:1803.07294, 2018.
- [11] GUO S, LIN Y, FENG N, et al. Attention based spatial-temporal graph convolutional networks for traffic flow forecasting[C]//Proceedings of the AAAI conference on artificial intelligence: vol. 33: 01. 2019: 922-929.
- [12] WANG X, MA Y, WANG Y, et al. Traffic flow prediction via spatial temporal graph neural network[C]//Proceedings of The Web Conference 2020. 2020: 1082-1092. 22
- [13] YU J J Q, GU J. Real-time traffic speed estimation with graph convolutional generative autoencoder[J]. IEEE Transactions on Intelligent Transportation Systems, 2019, 20(10): 3940-3951.
- [14] JIANG W, LUO J. Graph neural network for traffic forecasting: A survey[J]. arXiv preprint arXiv:2101.11174, 2021.

- [15] JIANG R, YIN D, WANG Z, et al. DL-Traff: Survey and Benchmark of Deep Learning Models for Urban Traffic Prediction[C]//Proceedings of the 30th ACM International Conference on Information & Knowledge Management. 2021: 4515-4525.
- [16] ZHANG J, ZHENG Y, QI D. Deep spatio-temporal residual networks for citywide crowd flows prediction[C]//Thirty-first AAAI conference on artificial intelligence. 2017.
- [17] LIN Z, FENG J, LU Z, et al. Deepstn+: Context-aware spatial-temporal neural network for crowd flow prediction in metropolis[C]//Proceedings of the AAAI conference on artificial intelligence: vol. 33: 01. 2019: 1020-1027.
- [18] YAO H, TANG X, WEI H, et al. Revisiting spatial-temporal similarity: A deep learning framework for traffic prediction[C]//Proceedings of the AAAI conference on artificial intelligence: vol. 33: 01. 2019: 5668-5675.
- [19] JIANG R, CAI Z, WANG Z, et al. DeepCrowd: A deep model for large-scale citywide crowd density and flow prediction[J]. IEEE Transactions on Knowledge and Data Engineering, 2021.
- [20] HASTIE T, TIBSHIRANI R, FRIEDMAN J. The Elements of Statistical Learning: Data Mining, Inference, and Prediction[M]. 2nd ed. Springer, 2009.
- [21] HINTON G E, SALAKHUTDINOV R R. Reducing the Dimensionality of Data with Neural Networks[C]//Science. 2006: 504-507.
- [22] MÜNZ G, LI S, CARLE G. Traffic Anomaly Detection Using K-Means Clustering [C]//. 2007.
- [23] CHOLLET F. Building Autoencoders in Keras[EB/OL]. 2016 [2021-11-24]. <https://blog.keras.io/building-autoencoders-in-keras.html>.
- [24] DENG L, LIAN D, HUANG Z, et al. Graph Convolutional Adversarial Networks for Spatiotemporal Anomaly Detection[J]. IEEE Transactions on Neural Networks and Learning Systems, 2022, 33(6): 2416-2428. DOI: 10.1109/TNNLS.2021.3136171.
- [25] Apache Software Foundation. Apache Hadoop[EB/OL]. n.d. [2021-11-24]. <https://hadoop.apache.org/>.
- [26] Apache Software Foundation. Apache Spark[EB/OL]. n.d. [2021-11-24]. <https://spark.apache.org/>. 23
- [27] Apache Software Foundation. Apache Flink[EB/OL]. n.d. [2021-11-24]. <https://flink.apache.org/>.
- [28] WANG L, CHENG K, FU R, et al. Real-time road traffic prediction with spatio-temporal correlations using Big Data platform[J]. Transportation Research Part C: Emerging Technologies, 2017, 81: 57-77.
- [29] ZHANG S, CHENG L. Real-time traffic prediction using big data and deep learning: framework and model[J]. IEEE Transactions on Intelligent Transportation Systems, 2018, 19(5): 1396-1406.
- [30] PANG Y, HOU M, LI Y, et al. Real-time prediction of urban traffic using low rank tensor decomposition[J]. Transportation Research Part C: Emerging Technologies, 2019, 104: 23-35.
- [31] Apache Software Foundation. Apache Kafka[EB/OL]. n.d. [2021-11-24]. <https://kafka.apache.org/>.
- [32] FOUNDATION A S. General Architecture[Z]. https://nightlies.apache.org/flink/flink-k-docs-release-1.1/internals/general_arch.html. Accessed: August 20, 2021. 2016.
- [33] FLINK A. Apache Flink Documentation[EB/OL]. 2021 [2021-11-17]. <https://nightlies.apache.org/flink/flink-docs-stable/docs/learn-flink/overview/>.
- [34] MA H, ZHOU M, OUYANG X, et al. Forecasting Regional Multimodal Transportation Demand with Graph Neural Networks: An Open Dataset[C]//2022 IEEE 25th International Conference on Intelligent Transportation Systems (ITSC). 2022: 3263-3268. DOI: 10.1109/ITS C55140.2022.9922512.

[35] AmNewYork. Manhattan happenings: Week of Jan. 31, 2019[Z]. <https://www.amny.com/news/manhattan-happenings-week-of-jan-31-2019/>. 2019. 241

致谢

在毕业论文即将完成之际，也是本科阶段即将结束的时刻，我想在此向那些在我的成长历程中一直给予我帮助、指导、鼓励、支持的人表达真挚的谢意。

首先，我要感谢我的导师宋轩老师。在整个本科阶段，他对我进行了悉心的指导和关怀，总能在学习道路上给予我关键方向的建议和许多帮助。我还要感谢我一直以来的指导老师姜仁河老师。自大二以来，是他带领我进入了交通流预测的领域，为我提供了科研训练的机会，使我熟悉了科学研究开展的流程与方法论。在平日里，他也给了我很多指导和帮助，带领我们取得了一些成果。然后，我还要感谢实验室的尹渡老师。每当我们遇到困难时，他总会无私的帮助我们，并像朋友一样给出了许多建议。在本科阶段的学习和科研中，老师们的支持和鼓励对我非常重要。正是有了他们的帮助，我才能顺利地完成本科学业和毕业论文的写作，并获得了能够前往更高平台深造的机会。

其次，我想感谢我的同学和队友们。他们与我一起工作，互相帮助和支持。在各种团队协作和项目开发中，我学到了许多知识和技能，这些都是我们共同努力的结果。此外，我还要感谢我的朋友们。是他们的陪伴和帮助使我的学习和生活充满乐趣。

他们一直给我很大的鼓励和支持，让我克服一切困难，始终保持乐观开朗的心态。

当然，在我求学期间，我也要感谢我深爱的父母。一直以来，他们对我付出、关爱、尊重和信任。在我学习和生活上遇到困难时，给予我无条件的支持。正是他们的帮助和鼓励，才使我能专注于学习，顺利完成学业。

最后，我要再次感谢所有支持我、帮助我的人，谢谢你们！ 25

-
- 说明：
- 1.总文字复制比：被检测论文总重合字数在总字数中所占的比例
 - 2.去除引用文献复制比：去除系统识别为引用的文献后，计算出来的重合字数在总字数中所占的比例
 - 3.去除本人文献复制比：去除作者本人文献后，计算出来的重合字数在总字数中所占的比例
 - 4.单篇最大文字复制比：被检测文献与所有相似文献比对后，重合字数占总字数的比例最大的那一篇文献的文字复制比
 - 5.指标是由系统根据《学术论文不端行为的界定标准》自动生成的
 - 6.红色文字表示文字复制部分;绿色文字表示引用部分;棕灰色文字表示作者本人已发表文献部分
 - 7.本报告单仅对您所选择比对资源范围内检测结果负责



 amlc@cnki.net

 check.cnki.net

<http://check.cnki.net/>