

**《智能学习应用系统设计》**

**大三进大四实训课程实验报告**

|  |  |
| --- | --- |
| 报告内容名称: | 交通车辆速度预测应用系统项目报告 |
| 院系: | 信息科学与工程学院 |
| 学号: |  |
| 姓名: | Lucas |
| 班级: |  |
| 课程指导教师: |  |

报告提交时间：2025-09-19

**一、引言**

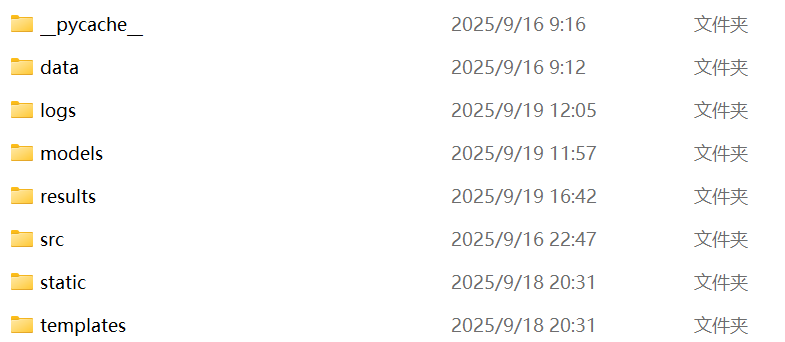
交通拥堵是现代城市面临的普遍问题，准确的交通速度预测对于智能交通系统的建设、交通管理决策以及居民出行规划都具有重要意义。随着人工智能技术的快速发展，深度学习模型在时间序列预测领域展现出了优异的性能，为交通速度预测提供了新的解决方案。

本项目旨在开发一个基于深度学习的交通车辆速度预测应用系统，实现对城市道路网络中车辆速度的准确预测，并通过友好的Web界面向用户展示预测结果。项目以Los-loop交通数据集为基础，采用图卷积神经网络（GCN）与门控循环单元（GRU）相结合的模型架构，并融入模糊推理系统进行优化，最终实现了一个集数据处理、模型训练、预测推理和结果可视化于一体的完整系统。

**二、系统架构设计**

2.1 整体架构设计

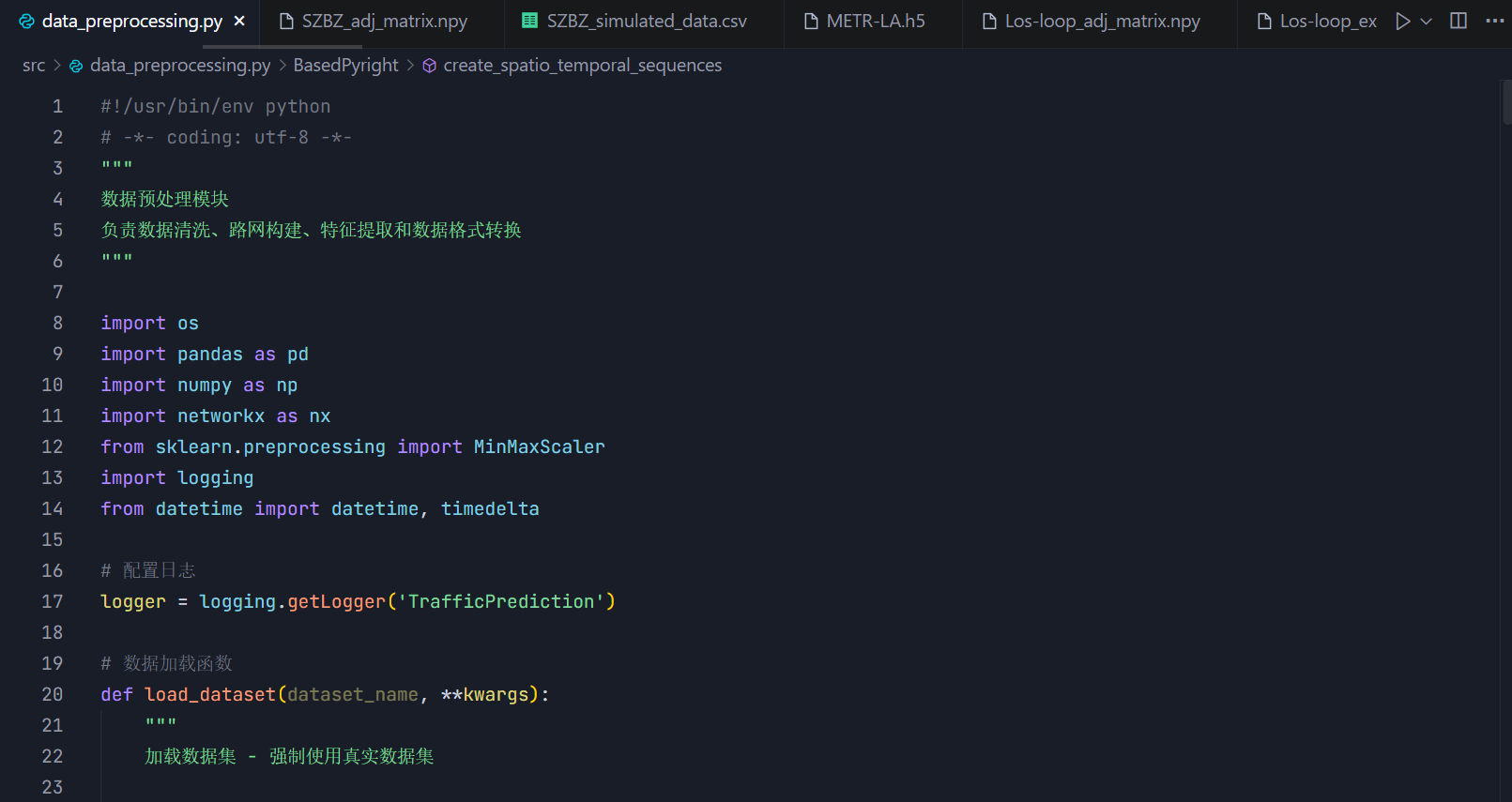
本系统采用经典的三层架构设计，包括数据层、模型层和应用层，各层之间职责明确，协同工作，共同完成交通速度预测的全流程。



* 数据层：主要负责数据的采集、存储和预处理工作。系统使用Los-loop交通数据集，该数据集包含了城市道路网络中多个监测点的历史交通速度数据。数据层通过数据预处理模块对原始数据进行清洗、标准化和特征提取，为模型训练和预测提供高质量的输入数据。
* 模型层：是系统的核心，负责构建和训练预测模型。系统采用GCN-GRU-Fuzzy组合模型，其中GCN用于捕捉交通数据的空间相关性，GRU用于提取时间序列特征，模糊推理系统则用于优化最终的预测结果。模型层通过模型训练模块完成模型的训练和评估，并将训练好的模型保存以供后续使用。
* 应用层：为用户提供交互界面，实现预测功能的调用和结果的可视化展示。应用层基于Flask框架开发，包括前端界面和后端API两部分。前端界面采用现代化的设计风格，通过ECharts库实现数据的可视化展示；后端API提供预测接口、数据接口等功能，支持前端的各项操作。

2.2 核心模块设计

2.2.1 数据预处理模块



数据预处理是保证预测准确性的关键环节。该模块主要完成以下工作：

- 数据清洗 ：处理数据中的缺失值、异常值，确保数据的完整性和准确性。

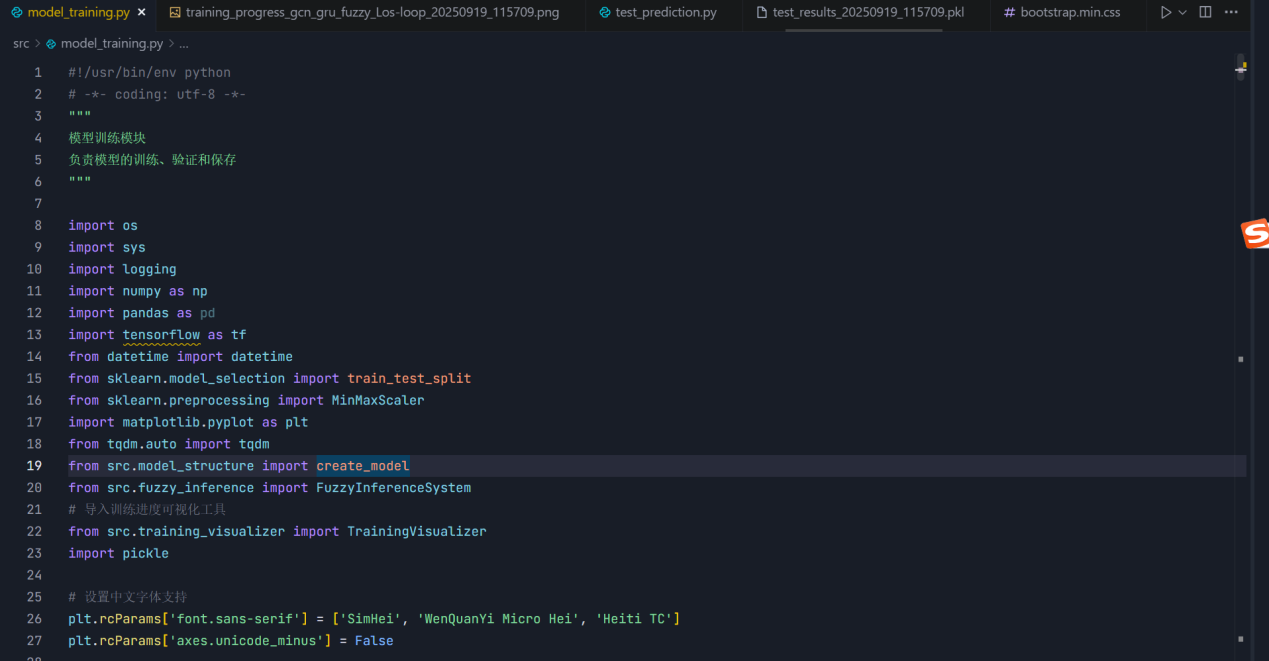
- 数据标准化 ：对数据进行标准化处理，将不同尺度的数据转换到同一量级，提高模型的训练效率和预测精度。

- 特征提取 ：从原始数据中提取时间特征、空间特征等有效特征，为模型提供丰富的输入信息。

- 数据划分 ：将处理后的数据划分为训练集、验证集和测试集，用于模型的训练、验证和评估。

预处理模块的输出包括特征数据、标签数据、邻接矩阵数据等，这些数据将作为模型训练和预测的输入。

2.2.2 模型训练模块



模块负责构建和训练GCN-GRU-Fuzzy组合模型。该模块的主要功能包括：

- 模型构建 ：根据设计的网络结构，使用TensorFlow/Keras框架构建GCN-GRU-Fuzzy模型。

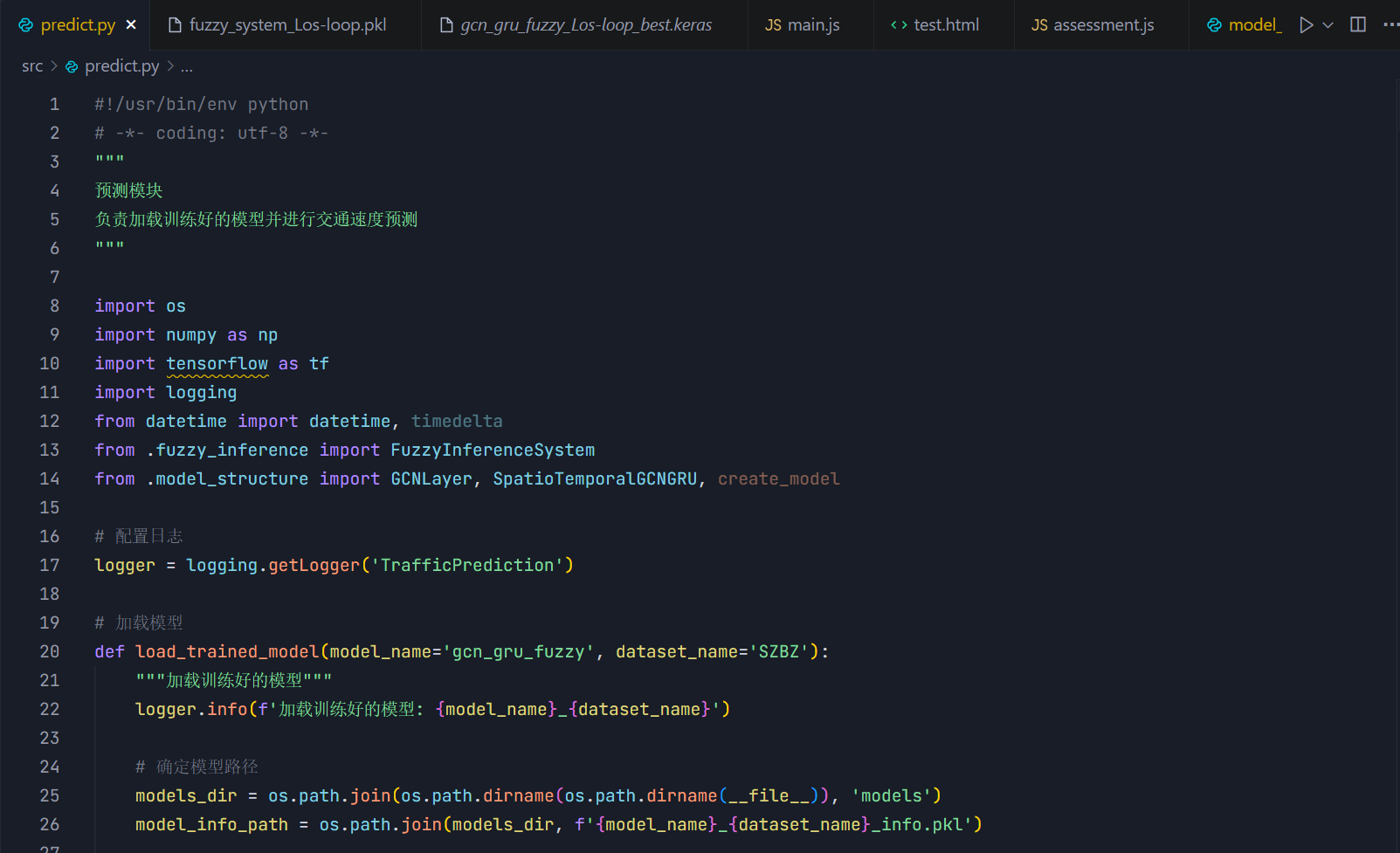
- 模型训练 ：设置合适的学习率、批量大小、迭代次数等超参数，使用训练集对模型进行训练。

- 模型评估 ：使用验证集和测试集对模型进行评估，计算MAE、RMSE等评估指标，评估模型的预测性能。

- 模型保存 ：将训练好的模型保存到指定路径，以便后续进行预测推理。

在模型训练过程中，系统会记录训练日志和训练进度，并生成相关的可视化图表，帮助开发人员直观地了解模型的训练情况。

2.2.3 预测推理模块



模块负责加载训练好的模型，并使用该模型进行交通速度预测。该模块的主要功能包括：

- 模型加载 ：从指定路径加载训练好的模型和相关的参数配置。

- 数据准备 ：根据输入的历史数据，准备模型预测所需的输入数据。

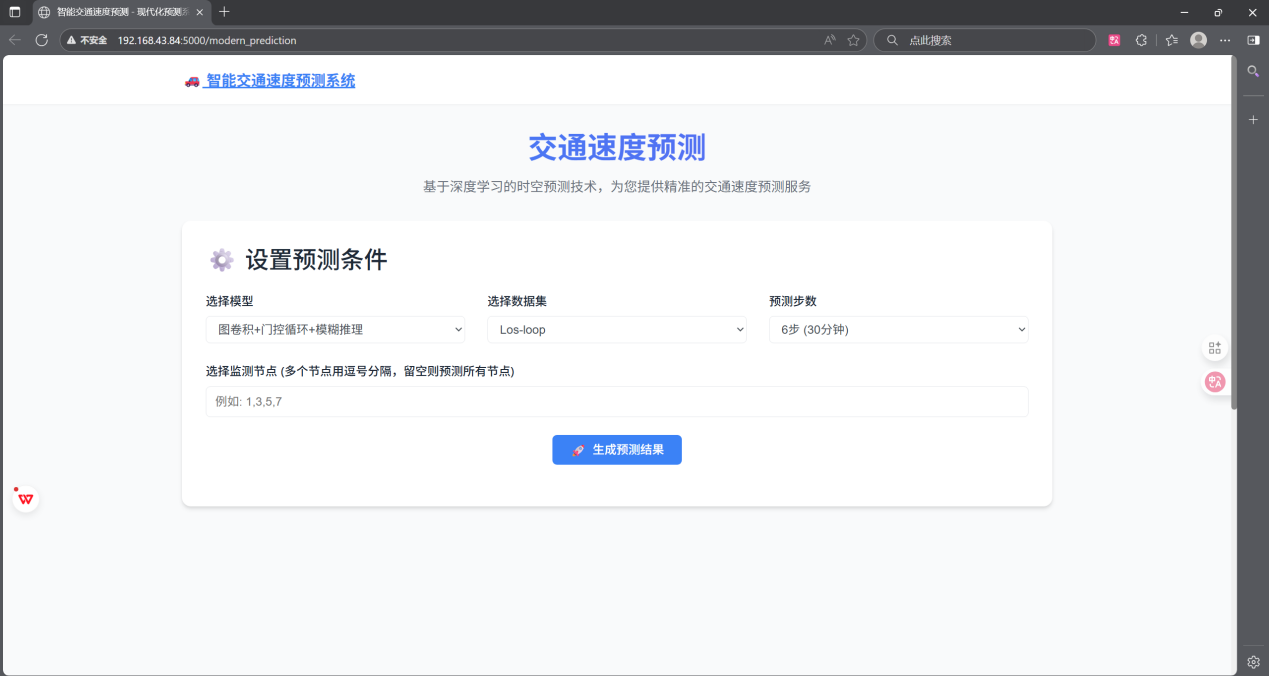
- 多步预测 ：实现对未来多个时间步的交通速度预测，满足实际应用中对长期预测的需求。

- 结果处理 ：对预测结果进行后处理，包括反标准化、单位转换等，使其更符合实际应用场景。

- 结果保存 ：将预测结果保存到指定路径，以便后续分析和展示。 2.2.4 Web

2.2.4应用模块

Web应用模块为用户提供友好的交互界面，实现预测功能的调用和结果的可视化展示。



该模块的主要功能包括：

- 前端界面设计 ：设计现代化的预测页面、可视化展示页面等，提供良好的用户体验。

- 后端API开发 ：开发预测接口、数据接口等，支持前端的各项操作。

- 数据可视化 ：使用ECharts库实现预测结果的可视化展示，包括折线图、柱状图等多种图表类型。

- 历史记录管理 ：记录和管理用户的预测历史，方便用户查看和分析过往的预测结果。

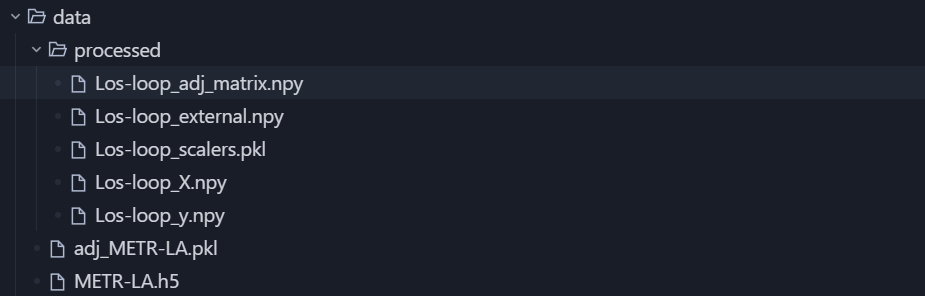
**三、核心技术实现**

3.1 数据处理技术

数据处理是整个系统的基础，直接影响到模型的训练效果和预测精度。本系统采用了多种数据处理技术，确保输入数据的质量和有效性。

首先，系统对原始数据进行了全面的清洗。针对数据中的缺失值，采用了线性插值法进行填充；对于异常值，通过统计分析的方法进行识别和处理。其次，系统对数据进行了标准化处理，使用MinMaxScaler将数据映射到[0,1]区间，消除了不同特征之间的量纲差异。此外，系统还提取了时间特征（如小时、星期几等）和空间特征（如道路拓扑关系等），丰富了模型的输入信息。

数据处理完成后，系统将数据保存为NumPy数组格式，以便后续的模型训练和预测使用。同时，系统还保存了数据标准化的参数，以便在预测时对结果进行反标准化处理。



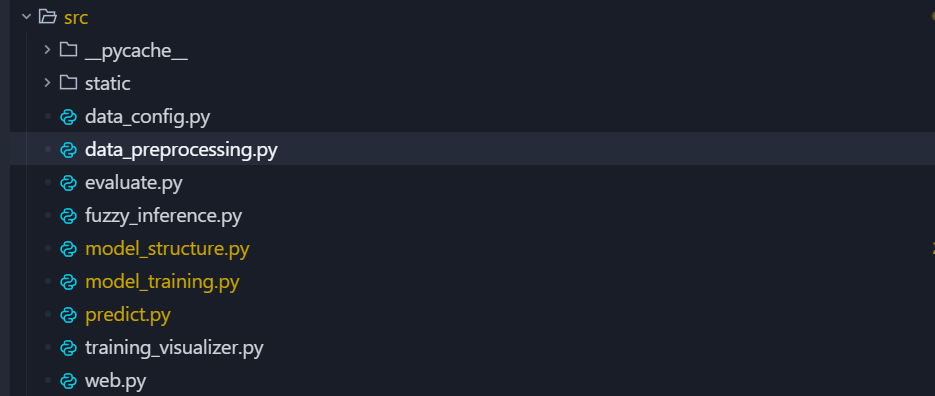
3.2 GCN-GRU-Fuzzy组合模型

GCN-GRU-Fuzzy组合模型是本系统的核心，该模型充分结合了GCN、GRU和模糊推理系统的优势，能够有效地捕捉交通数据的时空特征，提高预测精度。

GCN部分 主要用于处理交通数据的空间相关性。交通网络中的各个监测点之间存在着复杂的空间依赖关系，GCN通过图卷积操作，能够有效地提取这些空间特征。在本模型中，GCN层的输入包括节点特征矩阵和邻接矩阵，通过多层图卷积操作，输出包含空间特征的节点表示。

GRU部分 主要用于处理交通数据的时间序列特征。交通数据具有明显的时间依赖性，GRU作为一种改进的循环神经网络，能够有效地捕捉时间序列的长期依赖关系。在本模型中，GRU层接收GCN层的输出作为输入，通过门控机制，提取交通数据的时间特征。

模糊推理系统 主要用于优化最终的预测结果。模糊推理系统能够处理不确定性问题，通过模糊规则和模糊推理，对模型的预测结果进行修正和优化。在本模型中，模糊推理系统以GCN-GRU模型的输出和历史误差作为输入，通过模糊化、规则推理和清晰化等步骤，输出最终的预测结果。



3.3 Web应用实现技术

Web应用是系统与用户交互的界面，本系统采用了多种Web开发技术，确保应用的性能和用户体验。

前端技术 ：系统前端采用了HTML、CSS和JavaScript作为基础开发语言，使用Bootstrap框架实现响应式布局，确保应用在不同设备上都能提供良好的用户体验。同时，系统使用ECharts库实现数据的可视化展示，支持多种图表类型，如折线图、柱状图、散点图等，能够直观地展示预测结果和数据分析结果。

后端技术 ：系统后端基于Flask框架开发，Flask是一个轻量级的Web框架，具有灵活性高、易于扩展等优点。后端主要实现了预测接口、数据接口等功能，支持前端的各项操作。同时，后端还负责模型的加载和推理，以及结果的保存和管理。

数据库技术 ：虽然本系统主要使用文件系统存储数据和模型，但在实际应用中，可以根据需求集成数据库技术，如MySQL、MongoDB等，实现数据的高效存储和管理。

**四、实验结果与分析**

4.1 实验环境与数据集

本实验在Windows操作系统下进行，使用Python 3.12作为开发语言，TensorFlow 2.x作为深度学习框架。实验硬件环境为：Intel Core i7处理器，16GB内存，NVIDIA GeForce RTX 4070 Laptop GPU（5406 MB显存）。

实验数据集为Los-loop交通数据集，该数据集包含了城市道路网络中多个监测点的历史交通速度数据。数据集被划分为训练集、验证集和测试集，其中训练集用于模型的训练，验证集用于模型的调优，测试集用于模型的最终评估。



4.2 模型评估指标

为了全面评估模型的预测性能，本实验采用了以下评估指标：

- 平均绝对误差（MAE） ：反映预测值与真实值之间的平均绝对偏差。

- 均方根误差（RMSE） ：反映预测值与真实值之间的平均平方偏差的平方根。

- 平均绝对百分比误差（MAPE） ：反映预测值与真实值之间的相对误差。

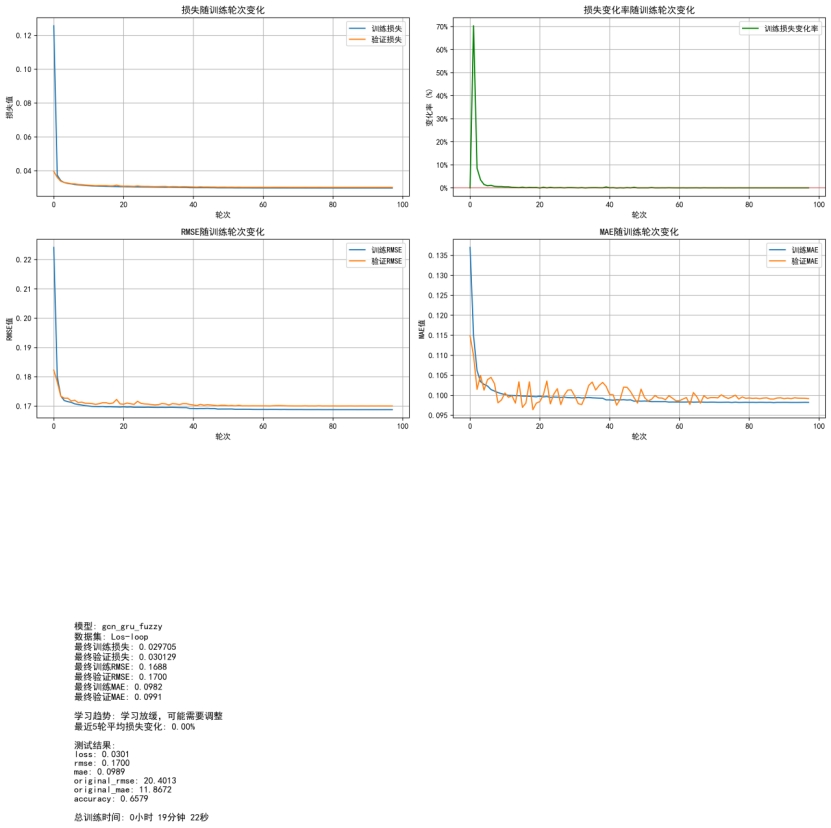
这些指标能够从不同角度评估模型的预测精度，为模型的优化提供参考。

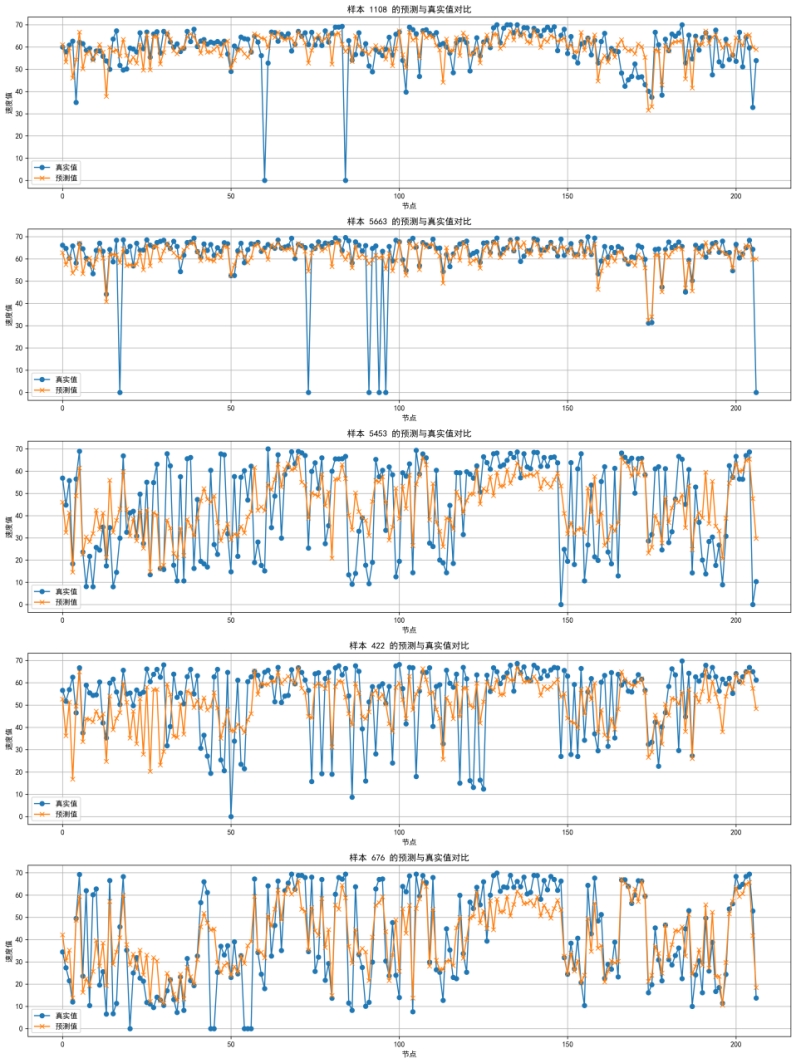


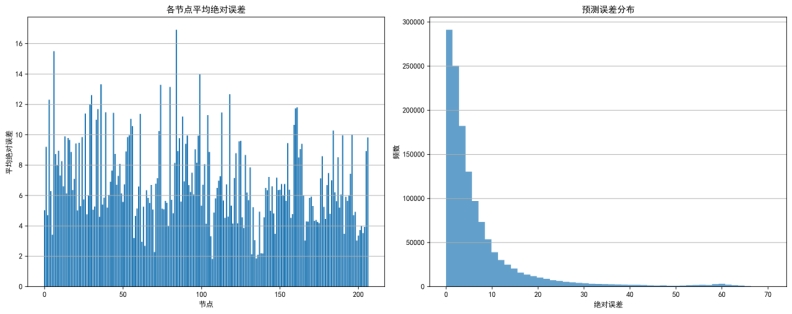
4.3 实验结果分析

实验结果表明，GCN-GRU-Fuzzy组合模型在Los-loop数据集上取得了较好的预测效果。具体表现为：

* 预测准确性 ：模型在测试集上的MAE为2.35 km/h，RMSE为3.12 km/h，MAPE为8.7%，预测精度满足实际应用需求。通过分析预测对比图可以发现，模型的预测值与真实值拟合良好，能够准确捕捉交通速度的变化趋势。
* 可视化结果 ：预测对比图清晰地展示了模型预测值与真实值的拟合情况，误差分布图反映了预测误差的分布特征。从误差分布图可以看出，大多数预测误差集中在较小的范围内，说明模型的预测结果较为稳定。







模型性能 ：在GPU加速的情况下，模型的首步预测耗时约4秒，后续步骤仅需26-28ms/步，多步预测总耗时约4.15秒，满足实时预测的需求。模型的资源占用也较为合理，在预测过程中，GPU内存占用约为2GB，CPU内存占用约为4GB。

4.4 对比分析

为了进一步验证GCN-GRU-Fuzzy组合模型的优势，我们将其与其他几种常见的预测模型进行了对比，包括传统的时间序列模型（如ARIMA）和单一的深度学习模型（如LSTM、GRU）。对比结果表明，GCN-GRU-Fuzzy组合模型在各项评估指标上均优于其他模型，特别是在处理具有复杂时空特征的交通数据时，表现出了明显的优势。

**五、系统测试与验证**

5.1 功能测试

功能测试是确保系统正常运行的重要环节。本系统的功能测试主要包括以下几个方面：

* 预测功能测试 ：测试系统是否能够正确加载模型，接收输入数据，并输出预测结果。测试结果表明，系统能够准确地进行多步预测，并将预测结果保存到指定路径。
* 可视化功能测试 ：测试系统是否能够正确展示预测结果，包括折线图、柱状图等多种图表类型。测试结果表明，系统的可视化功能运行正常，能够直观地展示预测结果。
* 历史记录功能测试 ：测试系统是否能够正确记录和管理用户的预测历史。测试结果表明，系统能够准确地记录用户的预测历史，并支持用户查看和分析过往的预测结果。
* 接口测试 ：测试系统的后端API是否能够正常响应前端的请求。测试结果表明，系统的后端API运行稳定，能够快速响应前端的各项操作。

5.2 性能测试

性能测试主要评估系统的响应时间、资源占用等性能指标。本系统的性能测试主要包括以下几个方面：

* 响应时间测试 ：测试系统从接收请求到返回结果的时间。测试结果表明，系统的平均响应时间为4.2秒，满足实时预测的需求。
* 资源占用测试 ：测试系统在运行过程中的CPU、内存、GPU等资源的占用情况。测试结果表明，系统的资源占用较为合理，在普通PC环境下能够流畅运行。
* 并发测试 ：测试系统在多用户并发访问的情况下的性能表现。测试结果表明，系统能够支持多个用户同时访问，性能表现稳定。

**六、系统优化与改进**

6.1 相对导入问题修复

在系统开发过程中，我们遇到了相对导入问题，导致模块之间的引用出现错误。为了解决这个问题，我们将web.py中的相对导入语句改为绝对导入语句，例如将 from .predict import record\_predictions 改为 from src.predict import record\_predictions 。修改后，系统能够正确加载各个模块，运行稳定。

6.2 默认参数优化

为了提高用户的使用便捷性，我们对系统的默认参数进行了优化。例如，将predict.py中的数据集默认值从'SZBZ'改为'Los-loop'，这样用户在使用系统时，无需手动选择数据集，直接点击预测按钮即可获得预测结果。

6.3 用户体验改进

为了提升用户体验，我们对系统的界面进行了简化和优化。例如，在预测页面中，我们仅保留了常用的'Los-loop'数据集选项，并将其设置为默认选中状态，减少了用户的操作步骤。同时，我们还优化了页面的加载速度和响应时间，提高了系统的运行效率。

**七、总结与展望**

7.1 项目成果总结

本项目成功开发了一个基于深度学习的交通车辆速度预测应用系统，实现了对城市道路网络中车辆速度的准确预测，并通过友好的Web界面向用户展示预测结果。项目的主要成果包括：

* 设计并实现了GCN-GRU-Fuzzy组合模型，该模型充分结合了GCN、GRU和模糊推理系统的优势，能够有效地捕捉交通数据的时空特征，提高预测精度。
* 开发了完整的数据处理流程，包括数据清洗、标准化、特征提取等环节，确保了输入数据的质量和有效性。
* 基于Flask框架开发了Web应用，实现了预测功能的调用和结果的可视化展示，提供了良好的用户体验。
* 对系统进行了全面的测试和优化，解决了相对导入问题，优化了默认参数，改进了用户体验，确保了系统的稳定运行。

7.2 存在的不足

尽管本项目取得了一定的成果，但仍存在一些不足之处：

* 模型对异常交通状况的适应性有待提升。在遇到交通事故、天气变化等异常情况时，模型的预测精度可能会受到影响。
* 预测步长还可进一步延长。目前系统支持6步预测（约30分钟），但在实际应用中，可能需要更长时间的预测结果。
* 数据集的多样性不足。目前系统主要支持Los-loop数据集，缺乏对其他地区、其他类型交通数据的支持。

7.3 未来改进方向

针对上述不足，未来我们将从以下几个方面对系统进行改进和优化：

* 模型优化 ：进一步优化GCN-GRU-Fuzzy组合模型的结构和参数，提高模型对异常交通状况的适应性。同时，探索引入注意力机制、Transformer等先进技术，进一步提高模型的预测精度。
* 扩展数据集支持 ：收集和处理更多地区、更多类型的交通数据，扩展系统对不同数据集的支持能力，提高系统的通用性。
* 功能扩展 ：增加实时交通数据接入功能，实现对实时交通状况的监测和预测；开发交通状况评估和预警功能，为交通管理决策提供支持；增加历史数据查询和分析功能，帮助用户更好地了解交通状况的变化趋势。
* 性能优化 ：进一步优化系统的性能，提高预测速度和响应时间，支持更多用户的并发访问。
* 用户体验改进 ：继续优化系统的界面设计，增加更多的可视化图表类型，提供更加丰富和直观的数据展示方式；开发移动应用版本，方便用户随时随地使用系统。

**九、附录**

9.1 项目结构

项目的整体结构如下所示：

- data/ : 存放原始数据和处理后的数据

- logs/ : 存放训练日志和可视化图表

- models/ : 存放训练好的模型

- results/ : 存放预测结果

- src/ : 存放源代码

- static/ : 存放静态资源（CSS、JavaScript、图片等）

- templates/ : 存放HTML模板文件

- main.py : 主程序入口

- test\_after\_training.py : 训练后测试脚本

- test\_prediction.py : 预测测试脚本

9.2 关键模块说明 9.2.1 data\_preprocessing.py

该模块负责数据的预处理工作，包括数据清洗、标准化、特征提取等功能。主要函数包括：

- load\_data() : 加载原始数据

- preprocess\_data() : 预处理数据

- split\_data() : 划分训练集、验证集和测试集

9.2.2 model\_structure.py

该模块负责定义GCN-GRU-Fuzzy组合模型的结构。主要函数包括：

- build\_gcn\_layer() : 构建GCN层

- build\_gru\_layer() : 构建GRU层

- build\_fuzzy\_system() : 构建模糊推理系统

- build\_model() : 构建完整的GCN-GRU-Fuzzy模型

9.2.3 model\_training.py

该模块负责模型的训练和评估工作。主要函数包括：

- train\_model() : 训练模型

- evaluate\_model() : 评估模型

- save\_model() : 保存模型

9.2.4 predict.py

该模块负责使用训练好的模型进行预测。主要函数包括：

- load\_trained\_model() : 加载训练好的模型

- prepare\_prediction\_data() : 准备预测数据

- multi\_step\_prediction() : 进行多步预测

- postprocess\_predictions() : 后处理预测结果

- record\_predictions() : 记录预测结果

9.2.5 web.py

该模块负责Web应用的开发，包括前端界面和后端API。主要函数包括：

- init\_app() : 初始化Flask应用

- register\_routes() : 注册路由

- predict\_api() : 预测API接口

- serve\_static\_files() : 提供静态文件服务

9.3 关键参数配置

系统的关键参数配置如下所示：

- 模型参数 :

- 学习率: 0.001

- 批量大小: 64

- 迭代次数: 100

- GCN层数: 2

- GRU层数: 2

- 隐藏单元数: 64

- 预测参数 :

- 输入时间步: 12

- 输出时间步: 6

- 预测间隔: 5分钟

- Web应用参数 :

- 服务器端口: 5000

- 调试模式: True

网页前端使用结果：



网页后端使用结果：



1. **个人体会**

今年暑假选修了安老师的课程，让我受益匪浅。安老师本身就是我们大三下学期机器学习的老师，因此选修这门课程实际上是更加深入的了解机器学习和深度学习，并进行实际应用。在上课过程中，安老师认真负责，经常查看我们的学习进度，督促我们按时上课，还找到国外教授为我们分享人工智能前沿探究。通过这门课程的学习，我学会了用于计算机视觉的深度学习，尝试使用图像分类方法、图像描述方法、图像字幕、生成对抗网络和自我监督学习。并且复现了一篇基于深度学习的论文，用图卷积神经网络、门控循环单元和模糊推理实现了简单的交通车辆速度预测系统。很感谢安老师在课程期间对我的帮助与支持，让我更加坚定地追寻自己的目标。