·绿色智能交通运输工程·

基于高德地图 API 的路段车速预测研究

陈国强[1], 张道文[2*], 周　凯[3]

(1. 西南交通大学交通运输与物流学院, 四川 成都 610031; 2. 西华大学汽车与交通学院, 四川 成都 610039;
3. 大连工业大学外国语学院, 辽宁 大连 116000)

摘　要：为了更容易地获取交通数据, 实现车速预测, 利用 Python 语言开发一套操作简单、界面友好的程序, 实现路段车速数据采集、处理、分析、预测和发布过程的简捷和集成化。根据高德地图开发平台的操作指南, 利用 Python 编写爬虫程序, 完成数据采集; 将采集的数据进行清洗、修复, 提取出指定路段的时间序列车速数据; 将时间序列进行分解, 使用 ARIMA 模型进行预测; 利用 Qt Designer 生成界面代码, 将逻辑代码与界面代码合并, 完成数据采集、处理、分析和预测过程的可视化设计; 利用 Django 框架, 完成发布预测结果的 Web 页面。本文的研究结果可供研究人员快速获取指定路段车速数据, 为出行者提供“拥堵预报”。

关键词：车速预测; 高德地图; PyQt5; Django; ARIMA; Python

中图分类号：U495　　**文献标志码：**A　　**文章编号：**1673–159X(2021)05 − 0035 − 07

doi:10.12198/j.issn.1673 − 159X.3759

Research on the Prediction of Road Speed Based on Amap API

CHEN Guoqiang[1], ZHANG Daowen[2*], ZHOU Kai[3]

(1. *School of Transportation & Logistics*, *Southwest Jiaotong University*, *Chengdu* 610031 *China*;
2. *School of Automotive & Transportation Engineering*, *Xihua University*, *Chengdu* 610039 *China*;
3. *School of Foreign Languages*, *Dalian Polytechnic University*, *Dalian* 116000 *China*)

Abstract: In order to obtain traffic data more easily and put the prediction of vehicle speed, a simple-to-use and user-friendly program is developed based on Python, which makes the collection, processing, analysis, prediction and release of road speed data more accessible and integrated. Such goal can be achieved, according to the operation guide of Amap development platform, by taking the following steps: Firstly, through python, a crawler program is designed to collet road speed data. Secondly, these collected data are purified and restored to extract the time series speed data of designated road section. Thirdly, the data are decomposed and used for prediction based on ARIMA model. Fourthly, through Qt Designer, the interface code is devloped and merged with logic code to complete the visual design process including data collection, processing, analysis and prediction. At last, the Django framework is used to complete the Web

收稿日期:2020 − 07 − 02

基金项目:成都市科技惠民项目(2015-HM01-00369-SF)。

第一作者:陈国强(1998—), 男, 硕士研究生, 主要研究方向为智慧交通。
ORCID: 0000 − 0001 − 6819 − 492X　　E-mail: 13458873575@163.com

*** 通信作者:**张道文(1968—), 男, 教授, 主要研究方向为智慧交通。
ORCID: 0000 − 0002 − 3098 − 857X　　E-mail: zhangdaowen@mail.xhu.edu.cn

引用格式:陈国强, 张道文, 周凯. 基于高德地图 API 的路段车速预测研究[J]. 西华大学学报(自然科学版), 2021, 40(5): 35 − 41.
CHEN Guoqiang, ZHANG Daowen, ZHOU Kai. Research on the Prediction of Road Speed Based on Amap API[J]. Journal of Xihua University(Natural Science Edition), 2021, 40(5): 35 − 41.

page of publishing prediction. The program can quickly obtain the speed data of designated road section and provide "congestion forecast" for travelers.

Keywords: vehicle speed prediction; Amap; PyQt5; Django; ARIMA; Python

在城市快速发展过程中,交通拥堵问题日益严重,仅通过“粗放式”修建基础设施来缓解拥堵问题存在诸多局限,而利用有效地组织路网交通流来缓解拥堵是智能交通系统研究的主要内容。交通诱导是智能交通系统的关键组成部分,而准确的交通状态预测是有效地进行交通诱导的前提和依据。大量的研究人员对交通状态预测模型进行了研究,使预测精度得到了极大的提升,如 KNN 模型[1]、神经网络模型[2-4]、支持向量回归[5-6]、旋转网络模型[7]等。但是如何让这些预测模型能够供非专业人员使用(如驾驶员选择路线),这方面的应用研究较少。

此外,交通数据获取也是一大难点,不仅数据源之间存在“信息孤岛”,而且数据源和需求方也不能有效连接,许多交通研究者有心无力,望而兴叹。即使有文献中提出采用爬虫方法获取交通数据[8-9],但也只是直接给出数据获取结果,没有给出具体实施步骤,导致后续研究人员需要用大量的时间研究怎么获取数据。

本研究拟基于高德地图 API,开发一套操作简单、界面友好的程序,实现路段车速数据采集、处理、分析、预测和发布过程的集成化、简便化。该方法让研究人员可以根据需求快速获取指定路段车速数据,给出 ARIMA 模型预测案例,为出行者提供“拥堵预报”,为之后的应用研究做铺垫。

1 车速数据获取

1.1 数据采集流程

进入高德开发平台[10],查看态势数据开发指南,通过该平台能够获取矩形区域、圆形区域和指定线路的交通态势数据。根据“使用说明”可知,获取数据分为 3 个步骤:1)申请“Web 服务 API 接口”密钥(Key);2)拼接 HTTP 请求 URL;3)接收 HTTP 请求返回的数据(JSON 或 XML 格式),解析数据。

高德地图态势数据 2 min 更新一次。若采集时间间隔太短,则数据量太大;若采集时间间隔过长,则可能错过一些重要信息。综合考虑后,本文将采集时间间隔定为 5 min,采集时间为 2019 年 12 月 21 日—2020 年 1 月 10 日,采集流程如图 1 所示。

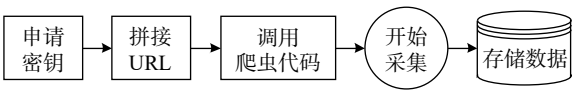


图 1 数据采集流程图

1.2 采集区域划分

为确定采集区域的坐标参数,可以利用“高德地图坐标拾取器”获取[11]。根据开发指南给出的“使用限制”可知,矩形对角线不能超过 10 km,故只能采用网格切分的思想,先将采集区域进行切割,划分为几个较小的采集区域,最后根据实际需求合并为较大的采集区域。因为普通用户调用量上限为 2 000 次/日,并发量上限为 20 次/秒,所以每次采集最多构造 6 个小格,才能不超过每天的调用量(以 5 min 为时间间隔,一个区域每天需要调用 288 次,每次采集由于区域融合需要调用 6 次,共计 1 728 次)。为了观察横贯成都市的道路情况,本研究采用如图 2 所示的分割方法。将数据采集结果通过 arcgis 绘制出来,如图 3 所示。此外,为解决爬取实时数据及代码运行时间过长的问题,可以采用云服务器托管运行的方法。

图 2 矩形区域分割示例

图 3　矩形区域分割示例

2　车速数据处理

2.1　数据转换

由于高德地图态势数据为实时数据而非历史数据，所以编写爬虫代码将不同时间点的采集结果放在不同的 CSV 文件中，利用 Python 编程实现文件的批量处理。首先，利用 groupby() 函数将每个文件内的同一路段车速数据求平均并存储到另一个文件中，再利用 Pandas 库从这些文件中提取出需要的路段数据，并赋予对应的时间，然后存储归类到一个文件中，形成指定路段平均速度的时间序列数据。

2.2　数据清洗

打开存储采集数据的文件，发现有部分路段车速数据缺失，有以下两方面的原因：1）交通态势数据的来源是用户和浮动车的 GPS 点数据，在定位、传输、采集的过程中，难免会出现一些错误；2）由于长时间、密频次调用高德地图 API，会造成服务器排名 ID 靠后，导致采集出来的数据不完整。交通态势数据中最关键的部分是路段平均行程速度，如果一条道路长时间缺失平均行程速度数据，则应该放弃此条道路的研究[8]。

2.3　数据修复

如果发现一条道路的车速数据并非长时间缺失，而是在某个时间点缺失，则可以采用数据填充的方法修复。这类情况发生的原因是没有车辆在道路上行驶，导致在该时刻无法采集到该路段内的速度信息。数据少量缺失的情况与道路等级有关，如快速路和主干道上，很少出现数据缺失，而在支路上较为常见。这是因为支路的车流量较少，且行驶的车辆不一定使用高德地图或者该车辆不一定是浮动车，不能向高德地图发送数据。

3　ARIMA 模型预测

3.1　分解数据

以“213 国道”为例，将前 20 天的数据作为训练集，最后一天的数据作为测试集，分解数据如图 4 所示。

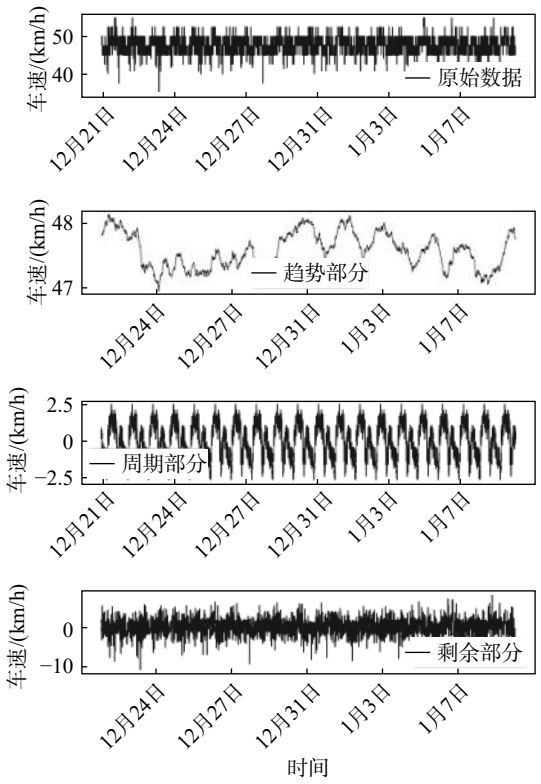


图 4　“213 国道”数据分解

3.2　平稳性检验

将分解后的“剩余部分”进行 adfuller 检验，如图 5(a)所示。结果显示：P-value<0.05，拒绝原假设，数据稳定；Test Statistic=−26.742 366<Critical Value(1%)=−3.431 547，表明 99% 的置信区间下都满足稳定性需求，数据稳定。

将分解后的“趋势部分”进行 adfuller 检验，如图 5(b)所示。结果显示：Test Statistic=−3.204 048>Critical Value(1%)=−3.431 553，Test Statistic=−3.204 048<Critical Value(5%)=−2.862 072，表明仅 95% 的置信区间满足稳定性需求，为保证预测精度，对数据进行差分运算。

Test statistic	−26.742 366
p-value	0.000 000
#Lags used	4.000 000
Number of observations used	5 467.000 000
Critical value (1%)	−3.431 547
Critical value (5%)	−2. 862 069
Critical value (10%)	−2.567 051

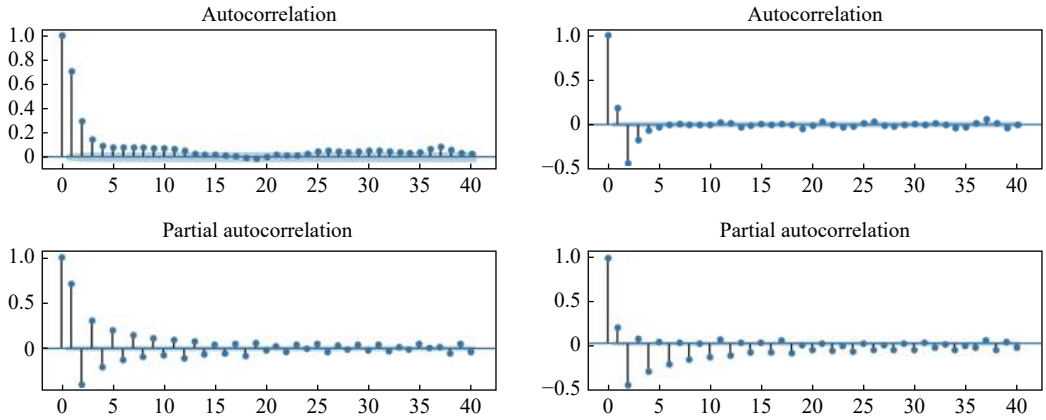
(a) 剩余部分

Test statistic	−3.204 048
p-value	0.019 752
#Lags used	33.000 000
Number of observations used	5 438.000 000
Critical value (1%)	−3.431 553
Critical value (5%)	−2.862 072
Critical value (10%)	−2.567 053

(b) 趋势部分

图 5 平稳性检验

图 6 的(a)、(b)分别是一阶差分和二阶差分的 ACF/PACF 图, 都呈现出拖尾的现象, 数据都稳定, 但二阶差分数据的稳定性明显优于一阶差分数据。为保证精度, 选择二阶差分数据作为预测数据, 确定 ARIMA 模型中的 d=2。



(a) 一阶差分 (b) 二阶差分

图 6 ACF/PACF 图

3.3 “趋势部分”白噪声检验

利用Python 的statsmodels 库中的acorr_ljungbox () 函数进行白噪声检验, 结果显示延迟 6 阶的 P 值为 6.08*10-298<0.05, 因此可以拒绝原假设, 认为该序列是非白噪声序列, 可以进行 ARIMA 模型预测。

3.4 模型定阶

绘制 AIC 的热力图, 如图 7 所示。颜色由浅到深, AIC 逐渐减小, MA4、AR2 对应的 AIC 最小, 确定 ARIMA 模型的参数 p=5, q=3。

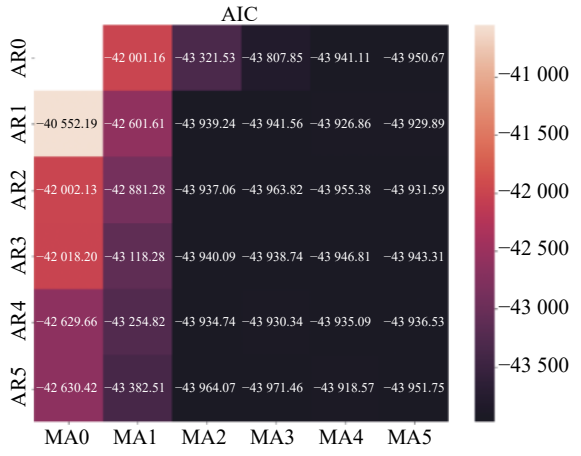


图 7 AIC 热力图

3.5 模型检验

用残差来检验模型的好坏。从图 8 中可以看出残差基本满足正态分布。再进行 D-W 检验, 发现 D-W 检验值接近于 2, 表明不存在自相关性, 说明模型较好。

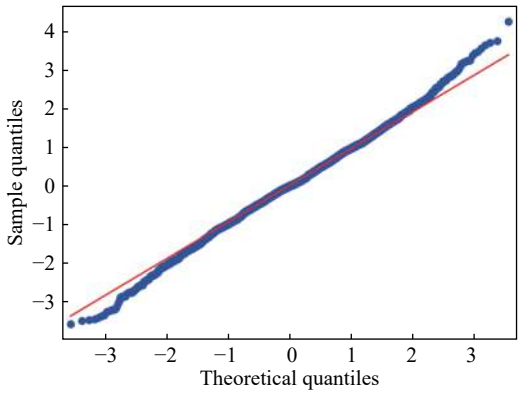


图 8 qq 图

3.6 数据预测

对“趋势部分”进行预测, 再将“季节部分”和“剩余部分”按平均的方式融合在一起得到车速数据的预测结果, 进行 MAE 计算, 结果为 1.6564, 绘制预测结果如图 9 所示, 其中黑色实线表示预测值, 灰色虚线表示实际值。可以看出预测值较实际值更为保守, 车速变化不明显, 而实际值的尖峰(异常值)较多, 相邻时间点的车速变化较大。

3.7 分时段预测

根据车速平均值, 将时段划分为高峰期和非高峰期, 如图 10 所示。其中灰色虚线为平均值 1, 是

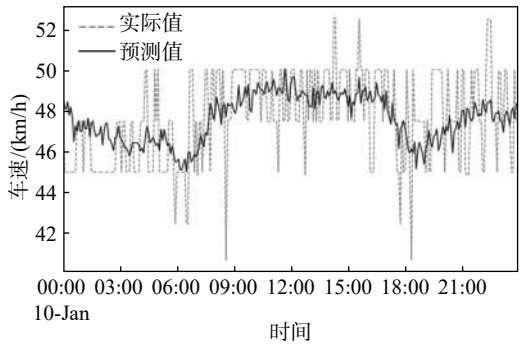


图 9　ARIMA 预测结果

对所有天数按照每天的时间点进行分组再求车速平均值;黑色实线为平均值 2,是所有车速的平均值。对于“213 国道”(一个远离城区的高速路),小于车速均值的时间为 00:20—07:40 和 17:15—23:30,这个是高峰期,7:40—17:15 是非高峰期。不同道路的高峰期和非高峰期不同,不能以平常上下班时间作为划分典型时段的依据,故以平均值作为划分依据。图 11 为非高峰期(7:40—17:15)的预测结果,MAE=1.373 2,有显著提升。图 12 为高峰期 1(00:20—07:40)的预测结果,MAE=1.544 2,MAE 提升。图 13 为高峰期 2(17:15—23:30)的预测结果,MAE=1.851 5,MAE 降低。

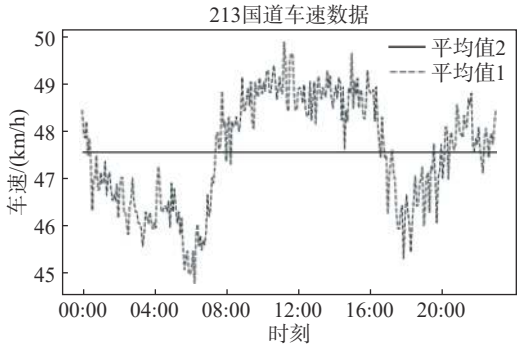


图 10　“213 国道”时段划分

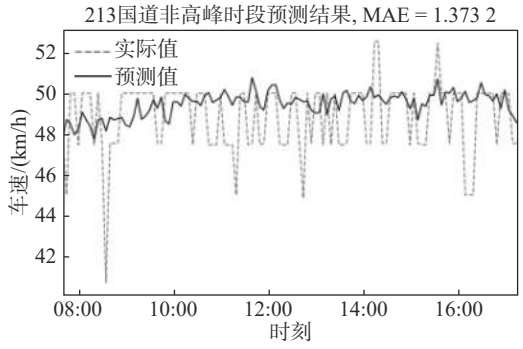


图 11　“213 国道”非高峰期预测结果

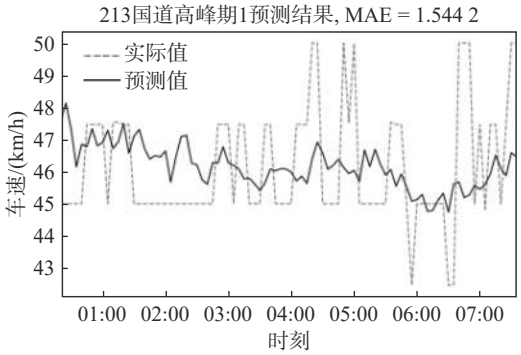


图 12　“213 国道”高峰期 1 预测结果

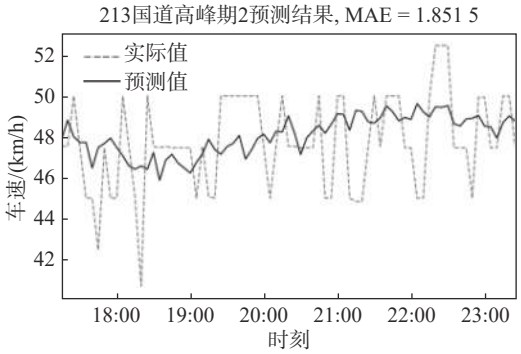


图 13　“213 国道”高峰期 2 预测结果

3.8　方法改进

上述步骤需人工建立 ARIMA 模型,过程复杂。为实现 ARIMA 模型预测的自动化,拟采用以结果(MAE)为导向的运算方法,将“模型定阶”“模型检验”的部分去掉,利用计算机的优势,将 6 以内的 p、q 值都计算一遍,找出 MAE 最小的 ARIMA 模型。

4　软件设计

4.1　技术实现工具

以 Python3.6 作为开发语言,PyCharm 作为集成开发工具,PyQt5 作为界面开发框架。用 PyInstaller 打包,使程序能在 Windows 操作环境下运行。

4.2　效果展示

打开软件,弹出主界面,如图 14 所示,共有 4 个模块。点击模块 3 的“数据处理”按钮,按需求选择,点击“运行”,如图 15 所示。

返回主界面,点击模块 4 的“数据预测”按钮,在弹出的窗口中选择“3 周”的数据量,在“指定目标路段”后的文本框中输入“人民南路三段”,点击“相关性可视化”按钮,软件绘制出“人民南路三段”在

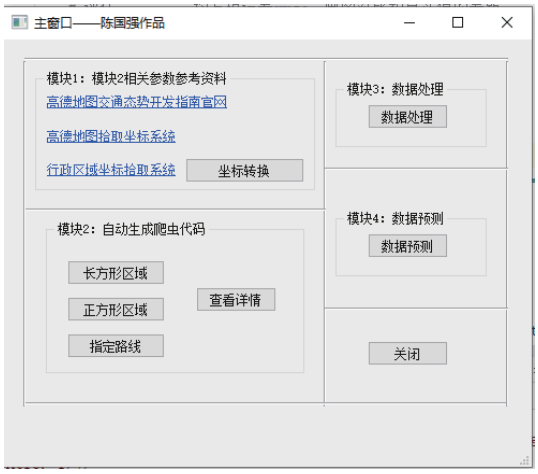


图 14 主界面

图 15 数据处理示例

工作日和休息日的日均车速数据分布情况，如图 16 所示。其中，灰色虚线表示工作日，黑色实线表示休息日，工作日与休息日有明显的不同，工作日存在两个峰值，而休息日的峰值只有一个。这表明工作日的出行是刚需，大部分居民必须在指定时间出行(朝九晚六的上下班时间)，而休息日居民可以按照自己的意愿选择出行时间。

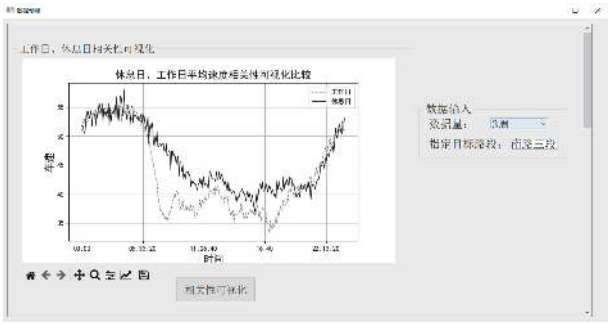


图 16 “人民南路三段”工作日与休息日的日均车速分布情况

以“213 国道”为例，滚动“数据预测”窗口右边的滑轮，直到出现 ARIMA 预测版面，点击“预测按钮”，将出现预测结果和计算过程，如图 17 所示。其中，灰色虚线表示实际值，黑色实线表示预测值，实际值的变化幅度较大，预测值比较保守，预测结果 MAE=1.6394，相较于图 9(以 AIC 值最小作为判定模型参数的方法)的预测结果 1.6564 有所提升。

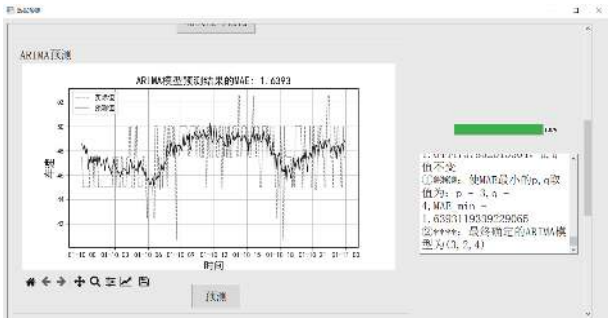


图 17 “213 国道”预测结果

5 拥堵预报的 Web 开发

以 Django 作为开发框架，设计了一个能够发布预测结果的网页。其中“输入页面”如图 18 所示，默认值为“大件路”。输入“成都绕城高速”，点击“预测”按钮，出现“输出页面”，如图 19 所示。图 19 的右下角是预测报告，粉红色虚线表示预测值，蓝色实线表示平均值。报告显示 1 月 10 日的预测值总是高于平均值，表明 1 月 10 日(星期五)的“成都绕城高速”较平常而言更通畅。

图 18 输入页面

图 19 输出页面

6 结论

1)以高德地图态势数据为数据来源，运用 Python 语言编写爬虫代码，通过云服务器托管运

行,经过数据转换、清洗和修复,获得成都市部分区域 3 周的车速数据。

2)采用 ARIMA 模型,完成了时间序列预测工作。

3)以 Python 语言为基础,通过 PyCharm 和 Qt Designer 等工具,实现了指定路段车速数据采集、处理、分析和预测的自动化、程序化。

4)以 Django 为 Web 开发框架,实现了指定路段车速预测结果的发布功能。

论文的研究成果可以帮助交通研究者快速获取车速数据,为出行者提供指定路段的拥堵预测结果。在以后的工作中,可以尝试完善“拥堵预报”的 Web 开发,为导航软件添加“预报”功能,当输入驾驶目的地后,系统将为用户提供途经路段的预测报告,包括途经路段的最通畅出行时间和最拥堵出行时间。

参　考　文　献

[1] RYU U, WANG J, KIM T, et al. Construction of traffic state vector using mutual information for short-term traffic flow prediction[J]. Transportation Research Part C: Emerging Technologies, 2018, 96(3): 55 − 71.

[2] LANA I, LOBO J L, CAPECCI E, et al. Adaptive long-term traffic state estimation with evolving spiking neural networks[J]. Transportation Research Part C: Emerging Technologies, 2019, 101: 126 − 144.

[3] 姚志洪, 蒋阳升, 韩鹏, 等. 基于神经网络的小时间粒度交通流预测模型[J]. 交通运输系统工程与信息, 2017, 17(1): 67 − 73.

[4] 钟晨昊, 王蒙, 赵威, 等. 基于 LSTM 的平面交叉口短时交通流预测[J]. 公路交通技术, 2019, 35(5): 119 − 123.

[5] LIU Y, WANG R X. Study on network traffic forecast model of SVR optimized by GAFSA[J]. Chaos, Solitons and Fractals, 2016, 89(8): 153 − 159.

[6] 傅成红, 杨书敏, 张阳. 改进支持向量回归机的短时交通流预测[J]. 交通运输系统工程与信息, 2019, 19(4): 130 − 148.

[7] HUANG S, SADEK A W. A novel forecasting approach inspired by human memory: The example of short-term traffic volume forecasting[J]. Transportation Research Part C: Emerging Technologies, 2009, 17(5): 510 − 525.

[8] 鲁武当. 基于交通态势数据的城市道路平均行程速度时空分布特征及预测研究[D]. 西安: 长安大学, 2016.

[9] 刘旭. 基于时间序列的经典模型和 LSTM 模型的城市宏观行程速度预测研究[D]. 北京: 北京交通大学, 2016.

[10] 大厂_08_全栈开发. python 高德地图交通态势数据爬取[EB/OL]. (2020-05-09)[2020-05-25]. https://blog.csdn.net/yan88888888888888888/article/details/106017713.

[11] Vicent_9920.高德地图之拾取地点[EB/OL]. (2017-04-30)[2020-05-25]. https://blog.csdn.net/vicent_9920/article/details/70991629.

(编校:夏书林)