

---

上海大学

SHANGHAI UNIVERSITY

毕业设计（论文）

UNDERGRADUATE PROJECT (THESIS)

题目：基于回归分析与记忆网络的交通拥堵预警系统

学院	计算机工程与科学学院
专业	计算机科学与技术
学号	15121252
学生姓名	卢广犇
指导教师	吴绍春
起讫日期	2019.02.25 – 2019.06.07

## 目录

摘要.....	III
ABSTRACT.....	IV
第 1 章 绪论.....	1
§1.1 交通拥堵预警系统构建的背景及意义.....	1
§1.2 模型研究现状及存在的问题.....	1
§1.2.1 研究现状.....	2
§1.2.2 交通流数据的特点.....	2
§1.2.3 交通流数据存在的问题.....	4
§1.2.4 模型构建的难点.....	4
§1.3 本文研究内容及目标.....	4
§1.3.1 研究内容.....	5
§1.3.2 研究目标.....	5
§1.4 交通流数据库介绍.....	6
§1.5 本文组织结构.....	8
第 2 章 基于支持向量机回归的短时预测.....	9
§2.1 支持向量机回归概述.....	9
§2.1.1 支持向量机回归的基本概念.....	9
§2.1.2 支持向量机回归的建模过程.....	11
§2.1.3 粒子群优化算法.....	12
§2.2 实例验证.....	13
§2.2.1 数据来源.....	13
§2.2.2 效果分析.....	13
§2.3 本章小结.....	16
第 3 章 基于长短期记忆网络的异地预测.....	17
§3.1 长短期记忆网络概述.....	17
§3.1.1 循环神经网络与长短期记忆网络.....	17
§3.1.2 长短期记忆网络结构.....	18
§3.1.3 长短期记忆网络的搭建.....	19
§3.2 实例验证.....	21
§3.2.1 数据来源.....	21
§3.2.2 效果分析.....	21
§3.3 本章小结.....	23
第 4 章 模糊综合评价.....	24
§4.1 评价指标选取.....	24
§4.2 模糊评价模型.....	25

§4.2.1 熵值法赋权.....	25
§4.2.2 确定评价指标隶属度.....	28
§4.2.3 拥堵级别确定.....	29
§4.3 实例验证.....	29
§4.3.1 数据来源.....	29
§4.3.2 权重计算.....	29
§4.3.3 过程举例.....	30
§4.3.4 效果分析.....	30
§4.4 本章小结.....	32
<b>第 5 章 总结与展望.....</b>	<b>33</b>
§5.1 本文总结.....	33
§5.1.1 本文的主要工作.....	33
§5.1.2 本文的主要创新点.....	34
§5.2 展望.....	34
致谢.....	36
参考文献.....	38
附录：部分源程序清单.....	40

# 基于回归分析与记忆网络的交通拥堵 预警系统

## 摘要

本文针对多岔口道路交通流数据预测困难的问题，围绕交通流短时预测与异地预测两大场景，提出了一种基于支持向量机回归与长短期记忆网络的短时异地交通流参数预测模型，并结合模糊综合评价方法构建交通拥堵预警系统。

通过对上游路段历史交通流数据的学习，支持向量机回归预测模型可以准确地预估未来时刻上游路段道路平均速度与交通流量。结合预测结果与历史数据，长短期记忆网络可以准确预估下游路段的道路平均速度与交通流量。最终预测数据经过交通流参数转化后得到包括平均速度，交通流密度，道路饱和度在内的评价指标集。通过熵值法赋权确定早晚高峰及其他时段下各指标的权重系数，并利用隶属度函数确定各指标的评价矩阵，最终根据最大隶属度原则得到道路交通拥堵等级，若达到阈值则发出警报。

通过对美国加利福尼亚 PeMS 系统数据库中标号为 2552 和 2554 的高速路交通流数据进行实验分析后证明，该系统具有较高的预测精度，正确率可达到 96%。

**关键词：**交通拥堵，支持向量机回归，长短期记忆网络，熵值法

# Traffic congestion warning system based on regression analysis and memory network

## ABSTRACT

Concerning the difficulty of traffic flow forecasting in multi-segment road, this paper proposes a short-term off-site traffic flow prediction model based on support vector regression and long short-term memory networks, combined with fuzzy synthetic evaluation method, a traffic congestion early warning system is constructed.

By learning the historical traffic flow data of the upstream road section, the support vector machine regression prediction model, which has been optimized by the particle swarm optimization algorithm, can accurately estimate the average road speed and traffic flow of the upstream road section in the future. Combined with prediction results and historical data, long short-term memory networks can accurately estimate the average road speed and traffic flow of downstream road sections. The final forecast data is transformed into traffic parameters to obtain an evaluation index set including average speed, traffic flow density, and road saturation. The weights of the evaluating indicators under the morning peak, evening peak and other periods are determined by the entropy method, and using the membership function to determine the evaluation matrix of each indicator. Finally, the road traffic congestion level is obtained according to the principle of maximum membership degree. Once the threshold is reached, an alarm is issued.

Numerical experiments are performed using the data from the No.2552 and No.2554 highway in the Caltrans Performance Measurement System of California, USA, where No.2554 is upstream and No. 2552 is downstream. It should be pointed out that the road section contains two forks, which makes the experiment practical. The results show that the system has great performance and the prediction accuracy can reach 96%.

**Keywords:** Traffic Congestion, Epsilon-Support Vector Regression, Long Short-Term Memory Network, Entropy Method

## 第1章 绪论

本章主要描述了交通拥堵预警系统的背景、意义，介绍了现有的交通流数据库，介绍了现有的交通流数据库，研究分析了国内外相关课题的研究现状，进而提出了所要研究的内容及目标。

### § 1.1 交通拥堵预警系统构建的背景及意义

近年来，随着城市经济高速发展，机动车保有量不断增加，交通拥堵现象日益严重。数据表明，城市汽车保有量逐年攀升，截至 2018 年，全国机动车保有量已达 3.25 亿辆。在类似北京，上海等一线城市中，交通拥堵已经成为了一个很普遍的现象，人们对此抱怨连连。放眼国际，交通拥堵均为大城市的通病。除了包括升级道路，修建高架桥在内的一系列现有方法外，智慧交通的概念慢慢走近人们的视野。

另一方面，随着商业 5G 部署的第一波浪潮的兴起，智慧交通的概念逐步变得清晰。作为智慧交通系统的最新发展方向，针对智能车路协同系统的研究愈发火热。智能车路协同系统通过无线通信、传感探测等技术获取车路信息，在车车、车路信息交互和共享的基础上，实现车辆和基础设施之间智能协同。这一系统保证了交通安全，提高了通行效率，为形成安全高效的道路交通系统奠定了基础。随着 5G 技术的到来，智能车路协同系统的最后一个环节将逐渐完善，结合人工智能等新兴技术，最终将实现不同等级智能车辆在同一条道路上的同时运行，从而达到车路协同。

交通拥堵预警系统是智能车路协同系统的重要组成部分，其依靠庞大的数据，高效的数据挖掘等技术，实现时间与空间上对未来道路交通情况的实时预测。对于交通拥堵的预警，不仅可以提醒即将经过拥堵路段的司机重新选择路线，从而防止拥堵加剧，而且可以帮助交警有重点地疏导交通，缓解拥堵趋势，从而预防拥堵发生。甚至可以与路网系统中的红绿灯系统融合，合理调整红绿灯状态，做到分流与疏通。

总的来说，交通拥堵预警可以帮助司机合理的规划出行路线，节省人们出行时间，有效地分配资源。也可以帮助交警出台合理的控制策略，缓解城市拥堵。

综上所述，本课题研究基于回归分析的交通拥堵预警系统具有现实意义。

### § 1.2 模型研究现状及存在的问题

近年来，有关短时交通流预测的研究得到了国内外众多学者的高度关注，下面具体阐述研究现状以及存在的问题。

### § 1.2.1 研究现状

应用于短时交通预测的模型主要可被归纳为统计模型，非线性预测模型与人工智能模型。

对于统计模型，Van Der Voort M[1]等人构建了结合自组织映射与时间序列模型的 KARIMA 模型，其通过将输入映射至最佳匹配单元，依据与该单元相连的可单独调整的 ARIMA 模型实现预测，凭借多个细分的时间序列模型，克服了单时间序列模型预测的局限性。周宏[2]等人提出差分自回归移动平均(ARIMA)和季节性差分自回归移动平均(SARIMA)的组合预测模型，结合最小二乘原理进行权重分配，使组合预测模型取得了高于单一预测模型的预测精度。对于非线性预测模型，宗春光[3]等人提出了一种基于相空间重构理论的局部预测方法。杨飞[4]提出了基于回声状态网络的交通流预测模型，并结合信号处理方法克服交通流中复杂的噪声成分的影响，取得了不错的预测效果。对于人工智能模型，谭娟[5]等人提出了基于深度学习的自编码网络，从无标签数据集中挖掘获取可表征数据深层特征的隐层参数，最后利用 Softmax 回归并进行多态预测。傅贵[6]等人深入研究了支持向量机回归，通过对算法本身的优化，使得预测误差小于基于卡尔曼滤波的预测方法。同样采用支持向量机回归，晏雨婵[7]等人通过粒子群优化算法，并结合模糊综合评价构建预测模型，取得了较高的准确度。Cheng[8]等人利用最大 Lyapunov 指数识别交通流混沌特征，结合相空间重构理论以及支持向量机回归，在预测准确性和及时性方面取得了更好的性能。

### § 1.2.2 交通流数据的特点

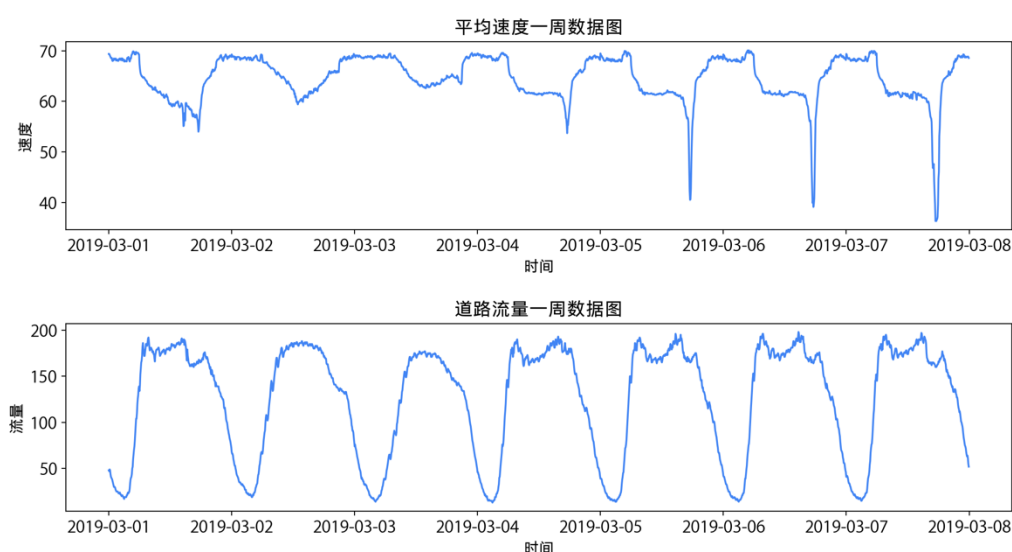


图 1-1 交通流数据

交通流数据看似杂乱无章，实际上具有一定的规律性。如图 1-1 所示，道路平均速度从 3 月 5 日起有明显的规律性，从宏观上看，5，6，7 三天的平均速度走势类似，形状相同。道路一周流量也具备明显的规律性，从宏观上看，1，4，5，6，7 五天的道路流量走势类似，形状相同，规律性有待挖掘。

时间序列的平稳性，要求经由样本时间序列所得到的拟合曲线在未来一段时间内仍能顺着现有的形态惯性地延续下去。平稳性要求序列的均值和方差不发生明显变化。平稳性又可以分为严平稳和宽平稳两种类型。严平稳指的是数据分布不随时间的改变而改变，如正态分布型白噪声，无论怎么处理，期望均为 0，方差为 1。宽平稳指的是期望与表示依赖性的相关系数不变，其中依赖性指的是未来某时刻  $t$  的值  $x_t$  依赖于它的过去信息，宽平稳要求依赖性不能有明显的变化。

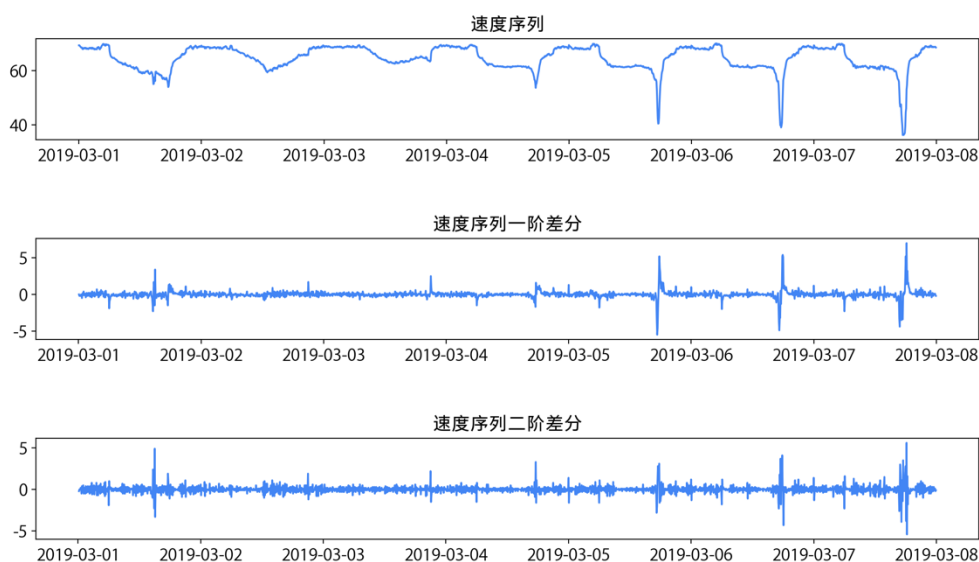


图 1-2 速度序列差分法分析图

如图 1-2 所示，速度图像通过有限次差分法后可以使得数据更为平稳，在二阶差分法后可以发现数据在一定范围内小规模波动。如图 1-3 所示，流量图像经过一阶差分法处理后，数据依旧存在较大的波动性与不确定性，经过二阶差分法处理后，波动范围过大，表明流量数据需要经过进一步有限次差分法后才能挖掘出数据潜在模式的规律。

通过对数据进行差分法分析后发现，特征向量的选取，决定了能否有效的挖掘数据背后的这一模式规律。



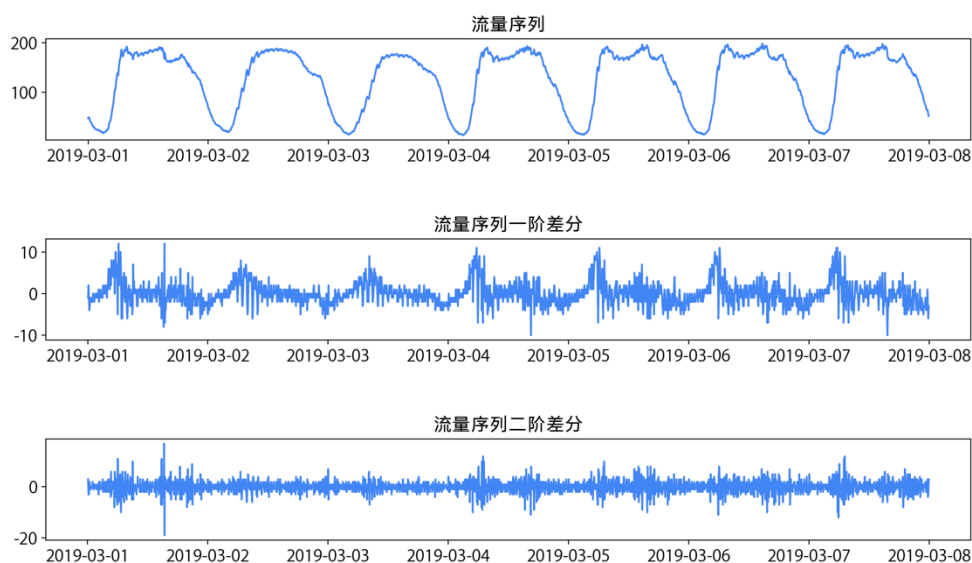


图 1-3 流量序列差分法分析图

### § 1.2.3 交通流数据存在的问题

交通流数据主要存在如下几个问题。

- (1) 数据噪声大，存在许多干扰。
- (2) 交通数据的非线性以及复杂性影响交通流预测的精度。

### § 1.2.4 模型构建的难点

虽然针对交通流的实时预测技术已较为成熟，但针对异地短时预测，仍然存在以下一些技术难点：

- (1) 如何确定交通流的特征向量，应该考虑何种交通流参数。
- (2) 如何建模挖掘异地之间交通流的相互影响。
- (3) 如何实现时间与空间上的准确预测。

## § 1.3 本文研究内容及目标

本文针对短时异地交通流预测问题入手，对历史交通流数据进行分析挖掘，提取主要参数指标，并建立短时异地预测模型，结合模糊综合评价方法，实现构建交通拥堵预警系统。

### § 1.3.1 研究内容

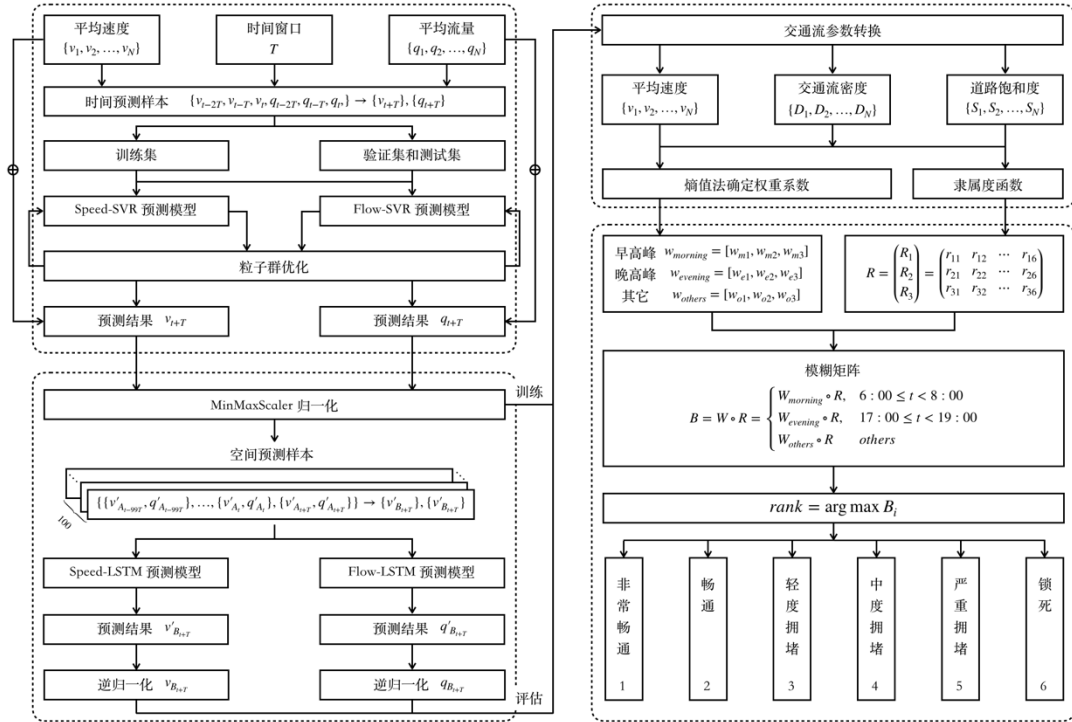


图 1-4 系统流程图

本文研究短时异地交通流预测，具体研究内容有以下几个方面。

- (1) 短时交通参数预测，即根据历史数据，构造基于时间序列的特征向量组，用于表征过去三个时间窗口的交通流情况。经过对数据的学习，支持向量机回归模型可以准确预测下一时间窗口时刻道路交通流参数，从而实现短时预测。
- (2) 异地交通参数预测，即根据历史数据以及短时预测结果，构建满足异地预测的三维特征向量，用以表征上游的交通流情况。经过对数据的学习，长短期记忆网络可以准确预测同一时刻下游路段的交通流参数，从而实现异地预测。
- (3) 模糊综合评价，即运用熵值法学习历史数据，确定在早晚高峰及其他时刻各评价指标所占权重，短时异地预测结果经过交通流参数转化，经由隶属度函数得到评价矩阵，最终根据最大隶属度原则确定交通拥堵级别。

### § 1.3.2 研究目标

针对本文的研究内容，制定了以下几项指标：模型能自主构建基于时间序列的特征向量组，自主构建满足异地预测的三维特征向量，能结合短时预测模型与

异地预测模型，综合预估未来交通流参数，并且可以根据历史交通流数据，正确计算评价指标的权重系数，最终达到合理评估交通拥堵等级。

## § 1.4 交通流数据库介绍

本节主要介绍几个交通数据集，并分析各自的优缺点。

### （1）芝加哥交通跟踪器元数据（Chicago Traffic Tracker Metadata, CTT）

芝加哥交通跟踪器系统是由芝加哥交通管理局负责维护的道路监控系统。该系统通过持续监控和分析城市公交车的 GPS 信号，实时估算芝加哥主干道（非高速公路）的交通拥堵情况。该系统目前监测 1250 段大约 300 英里的主干道的拥堵程度，并每隔十分钟发布两种类型的拥塞估计：以路段为单位或以行政区域划分为单位。由于数据集来源依靠于公交车 GPS 信号的采集，考虑到公交车数量有限，所以在非高峰时段，某些路段的道路拥塞情况因数据缺失而不予显示。

以按路段编号的数据集为例，该数据集中的每一条记录，主要包括路段编号，街道名称，车流方向，起点街道名，终点街道名，路段长度，路段朝向，起点与终点经纬度与当前速度等信息，其中当前速度是通过一定量的 GPS 信号整理计算得出，可以用以表征当前路段的车流平均速度。

另外，该数据集不仅包括车辆数据，也包括行人数据，可用作其他分析。

### （2）纽约市交通速度探测器元数据（New York City Traffic Speed Detectors Metadata, NYCSD）

纽约市交通局的交通速度检测器元数据是一项公开的数据服务，数据主要包含来自纽约市五个行政区内主干道和高速公路上道路传感器收集的实时交通信息，用于城市交通的紧急响应与管理。

该数据集中的每一条记录，主要包含车辆编号，车辆平均速度，车辆通过路段的运行时间，时间戳，路段编号，路段名称，路段多点经纬度以及街区名称等信息。

### （3）德国 HighD 数据集（The HighD Dataset, HighD）

HighD 数据集记录的是德国高速公路上的自然车辆轨迹。该数据集使用无人机通过航空视角收集道路车流信息，得益于此，该数据集克服了影响已有交通数据收集方法精度的一些限制，比如车辆遮挡等问题。

该数据集选定了德国科隆周边高速公路的六个采集点，分别记录了在不同地点超过 110500 辆车的交通信息，记录总里程达 44500 公里，总时长为 147 小时。结合最先进的计算机视觉算法，每个车辆的轨迹，包括车辆类型，尺寸等都会被自动提取，且定位误差通常小于十厘米。

该数据集中的每一条记录，包含车辆编号，车辆坐标，车辆尺寸，车辆横纵向瞬时速度，车辆横纵向瞬时加速度等总共 25 个条目。由于该数据集创建的初衷是用于自动驾驶安全验证，所以包含了许多细致的条目，如与周边车辆的车距，周边车辆编号等信息，数据集质量非常高。

#### (4) 加州交通绩效评估系统 (Caltrans Performance Measurement System, PeMS)

加州交通绩效评估系统是由加州交通管理中心 (TMC) 通过 Caltrans 环路检测器收集的统一数据库，旨在用于减少道路拥堵，提高行程可靠性。该数据库可通过浏览器访问，凭借细致入微的用户交互界面，用户可以获取丰富的加州道路交通历史数据与实时数据。

该系统覆盖了加州 12 个行政区的若干条主干道，数据收集周期为  $7 \times 24$  小时，传感器每 30 秒上传一次数据，系统数据发布周期为五分钟，提供包括道路平均速度，车流量，道路占有率等在内的交通流数据。

表 1-1 交通数据集对比表

数据集	收集工具	单条数据主体	速度含义	流量偏差	天气因素
CTT	GPS	路段	路段平均速度	有	丰富，可收集
NYCTS	摄像头	车辆	车辆平均速度	有	丰富，可收集
HighD	无人机	车辆	车辆瞬时速度	无	单一，无差异
PeMS	摄像头	路段	路段平均速度	无	丰富，可收集

表 1-1 显示了各个数据集的交叉对比情况。CTT, NYCTS, PeMS 数据集均为实时数据集，即用户可以看到当日的道路交通情况，而 HighD 为历史数据集。

从单条数据主体的角度分析，CTT 与 PeMS 数据集为路段数据，即系统提供的平均速度，流量等数据均以路段为单位。由于 CTT 数据集来源依靠于公交车 GPS 信号的采集，存在部分路段信息缺失的问题，且流量数据存在偏差，因为同一路段上的公交车数并不足以表征整条路段的汽车流量。PeMS 数据集为路段数据，数据来源于路网探头，速度，流量等数据具备足够的准确度，能真实反映道路交通情况。NYCTS 与 HighD 均以车辆为单条数据主体，所记录的交通信息均以车辆为单位。两者的区别在于，由于 HighD 采用更为先进的无人机与计算机视觉技术，突破了传统摄像头采集面临的收集障碍，数据更为准确，并能提供每一帧场景中的交通流参数，更容易捕获各参数的变化。以速度为例，HighD 数据集提供车辆每一帧的瞬时速度，而 NYCTS 中采集的数据，是车辆通过某条路段时的平均速度，即为道路长度与通过时间的比值，考虑到数据集中的路段长度并不短，该车辆平均速度并不能很好的表示汽车当前的速度，精度偏差较大。

由于无人机采集数据受天气因素影响,所以 HighD 数据集采集均在晴天或能见度较高的阴天进行,天气因素单一,而 CTT, NYCTS, PeMS 均可以收集当天的天气状况,三个数据集采集周期均为  $7 \times 24$  小时,天气因素丰富,便于研究其对交通的影响。

本文主要采用速度与流量为道路交通流的主要表征参数,考虑到 CTT 数据集流量参数不能准确表征道路全部车辆数, NYCTS 数据集速度参数不能准确表示车辆采集时刻的准确速度, HighD 数据集受限于采集时刻的间断性,不符合全天多日采集的要求,故本文选取 PeMS 数据集为最终的实验数据集。

## § 1.5 本文组织结构

整篇论文分为五章。

第一章介绍了研究背景、研究意义,分析了交通流预测模型研究现状以及交通流数据存在的问题和模型构造的难点,分析对比了当下存在的几个可靠的交通数据集,并提出了本文的研究内容以及研究目标。

第二章主要介绍了支持向量机回归的基本概念,提出了基于支持向量机回归的短时交通流预测模型,阐述了交通流数据的特征提取与模型的搭建过程,并介绍了如何采用粒子群优化算法对预测模型进行优化,最后进行了实验模拟。

第三章主要介绍了长短期记忆网络与循环神经网络的基本概念与网络结构,提出了基于长短期记忆网络的异地交通流预测模型,阐述了模型的搭建过程,最后进行了实验模拟。

第四章主要介绍了模糊综合评价方法,阐述了评价指标选取,熵值法赋权,确定评价指标隶属度以及确定拥堵级别等环节,最后运用实例,详细展示了演算过程。

第五章对全文进行了总结,归纳了本文的主要工作与创新点,并指出了进一步研究的方向。

## 第2章 基于支持向量机回归的短时预测

本章是全文的重点章节，具体描述了基于支持向量机回归的短时交通流参数预测方法：介绍支持向量机回归的基本概念，引出本文所使用的基于支持向量机回归的短时预测方法，阐述交通流数据的特征提取与模型的搭建过程，并介绍如何采用粒子群优化算法对预测模型进行优化，最后进行实验模拟。

短时预测具有其现实意义，在现实场景中，短时预测可以通过利用路段当前时刻及历史交通流数据，构建短时预测模型，从而实现针对未来时刻的交通流参数预测，对道路拥堵作提前防范。

### § 2.1 支持向量机回归概述

本节介绍了支持向量机回归的数学模型与建模过程，以及如何利用粒子群优化算法优化模型。

#### § 2.1.1 支持向量机回归的基本概念

支持向量机（SVM）通过结构风险最小化较好地解决了小样本、非线性、维数灾难、过拟合和局部极小等问题，已成为机器学习领域的研究热点之一[9]。

支持向量机的主要思想是在高维空间中构造超平面或者超平面集合，使正反样例之间的间隔最大化。类似的，支持向量机回归（SVR）是指在高维空间中找到一个回归平面，使样本中尽可能多的点到该平面的距离最近。以本文所运用的 $\varepsilon$ -支持向量机回归为例：

首先，将样本集表示为

$$(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_l, y_l) \in \mathbb{R}^d \times \mathbb{R} \quad (1)$$

定义超平面为

$$f(\mathbf{x}) = \mathbf{w}^T \mathbf{x} + b, \mathbf{w} \in \mathbb{R}^d, b \in \mathbb{R} \quad (2)$$

支持向量机回归要解决的问题为

$$\min \frac{1}{2} \|\mathbf{w}\|^2 \quad s.t. |\mathbf{w}^T \mathbf{x}_i + b - y_i| \leq \varepsilon, i = 1, 2, \dots, N. \quad (3)$$

其中， $\varepsilon$  用来表示 SVR 预测值与实际值最大的差距。由于样本集中存在着噪声，所以引入松弛变量，式(3)可写为：

$$\min \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^N (\xi_i + \xi_i^*) \quad s.t. \begin{cases} (\mathbf{w}^T \mathbf{x}_i + b) - y_i \leq \varepsilon + \xi_i \\ y_i - (\mathbf{w}^T \mathbf{x}_i + b) \leq \varepsilon + \xi_i^* \\ \xi_i, \xi_i^* \geq 0 \end{cases} \quad (4)$$

对式(4)采用拉格朗日乘子法，得到

$$\begin{aligned}
 L = \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{\{i=1\}}^N (\xi_i + \xi_i^*) - \sum_{\{i=1\}}^N (\eta_i \xi_i + \eta_i^* \xi_i^*) \\
 - \sum_{\{i=1\}}^N \alpha_i (\varepsilon + \xi_i - y_i + w \cdot x_i + b) \\
 - \sum_{\{i=1\}}^N \alpha_i^* (\varepsilon + \xi_i^* + y_i - w \cdot x_i - b)
 \end{aligned} \tag{5}$$

求解(4)式相当于求解

$$\min_{w, b, \xi, \alpha, \eta} \max_{\alpha, \eta} L \quad \text{s. t. } \alpha, \eta \geq 0 \tag{6}$$

经过对偶变换，可得到

$$\begin{aligned}
 \max_{\alpha, \eta} -\frac{1}{2} \sum_{i, j=1}^N (\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*) x_i \cdot x_j - \varepsilon \sum_{i=1}^N (\alpha_i + \alpha_i^*) \\
 + \sum_{i=1}^N y_i (\alpha_i - \alpha_i^*)
 \end{aligned} \tag{7}$$

$$\text{s. t. } \sum_{i=1}^N (\alpha_i - \alpha_i^*) = 0 \text{ and } \alpha_i, \alpha_i^* \in [0, C]$$

考虑到在样本所在的维度空间内不一定存在线性可分的超平面，于是引入核函数进行升维，将样本投射至更高维度的样本空间。本文采用高斯径向基核函数

$K(\mathbf{x}, \mathbf{x}') = \exp(-\frac{\|\mathbf{x}-\mathbf{x}'\|_2^2}{2\sigma^2})$ ，将式(7)转换如下

$$\begin{aligned}
 \max_{\alpha, \eta} -\frac{1}{2} \sum_{i, j=1}^N (\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*) K(x_i, x_j) \\
 - \varepsilon \sum_{i=1}^N (\alpha_i + \alpha_i^*) + \sum_{i=1}^N y_i (\alpha_i - \alpha_i^*)
 \end{aligned} \tag{8}$$

$$\text{s. t. } \sum_{i=1}^N (\alpha_i - \alpha_i^*) = 0 \text{ and } \alpha_i, \alpha_i^* \in [0, C]$$

通过求解式(8)即可求得参数  $\alpha_i, \alpha_i^*$ ，从而可以求得权重参数  $\mathbf{w} = \sum_{i=1}^N (\alpha_i^* - \alpha_i) \mathbf{x}_i$ ，并求得  $b$  如下

$$b = \begin{cases} y_j + \varepsilon - \sum_{i, j=1}^N (\alpha_i^* - \alpha_i) K(\mathbf{x}_i, \mathbf{x}_j), & 0 \leq \alpha_i \leq C \\ y_j - \varepsilon - \sum_{i, j=1}^N (\alpha_i^* - \alpha_i) K(\mathbf{x}_i, \mathbf{x}_j), & 0 \leq \alpha_i^* \leq C \end{cases} \tag{9}$$

### § 2.1.2 支持向量机回归的建模过程

本文选取速度与流量作为表征交通流数据的主要参数，并分别建立针对速度与流量的支持向量机回归模型。假设特征向量为  $\mathbf{x}_i (\mathbf{x}_i \in \mathbb{R}^n)$ ， $y_i$  为交通流预测值。基于支持向量机回归的短时交通流预测模型就是寻找  $\mathbf{x}_i$  与  $y_i$  之间的关系。

$$\begin{aligned} f: \mathbb{R}^N &\rightarrow \mathbb{R} \\ y_i &= f(\mathbf{x}_i) \end{aligned} \quad (10)$$

文中将当前时刻与时间窗口分别记为  $t, T$ ，其中  $T$  设置为五分钟。采用当前时刻  $t$  与过去  $t-T, t-2T$  时刻的速度，流量参数组成的特征向量作为输入值，分别建立速度回归预测模型与流量回归预测模型，对未来  $t+T$  时刻的参数进行预测。

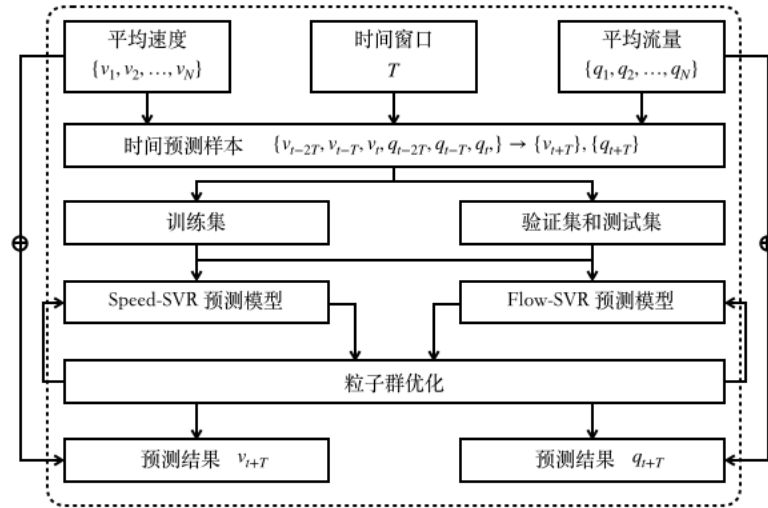


图 2-1 基于支持向量回归的建模流程图

图 2-1 为基于支持向量回归的建模流程，具体描述如下：

- (1) 构造特征向量，构造训练集。

$$\begin{aligned} \mathbf{x}_i &= (v_{t-2T}, q_{t-2T}, v_{t-T}, q_{t-T}, v_t, q_t) \\ y_i &= v_{t+T} \text{ or } q_{t+T} \end{aligned} \quad (11)$$

- (2) 对已知样本进行分析，选择合适的核函数及参数值。
- (3) 通过求解式(8)寻找最优超平面。
- (4) 粒子群算法优化参数
- (5) 根据求得的最优解构造决策函数，并利用测试样本集对未来  $t+T$  时刻的参数进行预测。



### § 2.1.3 粒子群优化算法

粒子群优化算法 (PSO) 是基于群体的演化算法, 其思想来源于人工生命和演化计算理论[10]。算法根据对环境的适应度将群体中的个体移动到更好的区域。粒子群优化算法将每一个个体看作是没有质量和体积的微粒, 在搜索空间中根据一定的速度飞行。粒子本身速度的改变不是随机的, 而是根据自身与同伴的飞行经验作动态调整。记每一个微粒为  $x_i$ , 搜索空间存在的粒子群为  $X = (x_1, x_2, \dots, x_i)$ , 每个粒子经过的最好位置, 即适应度最高的位置为  $pbest_i$ , 搜索空间中的粒子群的最优位置表示为  $Pbest = (pbest_1, pbest_2, \dots, pbest_i)$ , 群体所有微粒经历过的最好位置的索引号用  $gbest$  表示, 即  $gbest = \arg \max(Pbest)$ 。

第  $i$  个微粒的速度用  $v_i$  表示, 则第  $k$  次迭代, 粒子速度与位置的更新公式如下所示:

$$\begin{aligned} v_{id}^{k+1} &= w \cdot v_{id}^k + c_1 rand_1 \cdot (p_{id} - v_{id}) + c_2 rand_2 \cdot (p_{gd} - v_{id}) \\ x_{id}^{k+1} &= x_{id}^k + v_{id}^k \end{aligned} \quad (12)$$

其中,  $w$  称为惯性系数,  $c_1, c_2$  被称为加速常数,  $rand_1, rand_2$  为 0 到 1 之间的随机数。将算式的第一部分称为惯性行为, 即对粒子先前速度的记忆。第二部分称为认知行为,  $c_1$  成为自我因子, 表示微粒根据自己以往的经验调整自己, 即微粒的自我认知。第三部分称为社会行为,  $c_2$  成为社会因子, 表示微粒根据微粒群中最好的位置调整自己的飞行方向, 即微粒间信息共享。

惯性系数  $w$  表述了粒子先前速度对于当前速度的影响。当  $w$  较大时, 表明粒子移动速度快, 全局巡优能力强。当  $w$  较小时, 表明粒子移动速度慢, 局部巡优能力强。一般而言, 通常做法是使算法在前期有较高的全局搜索能力以得到合适的位置, 而在后期有较高的局部搜索能力以提高收敛精度。

当社会因子  $c_2 = 0$  时, 模型为自我认识型粒子群算法, 完全没有信息的共享, 即“只有自我, 没有社会”, 导致的结果是算法收敛速度缓慢; 当自我因子  $c_1 = 0$  时, 模型为无私型粒子群算法, 自我的缺失会导致群体多样性的丧失, 即“只有社会, 没有自我”, 容易陷入局部最优解而无法跳出, 当  $c_1, c_2$  均不为 0 时, 模型成为完全型粒子群算法, 该算法更容易保持收敛速度和搜索效果的均衡, 是比较好的选择。

在本文中, 优化算法初始生成众多参数不一的支持向量机回归模型, 将每一个支持向量机回归模型当作一个粒子, 待优化参数为惩罚参数  $C$  与高斯径向基核函数参数  $\gamma$ , 粒子适应度为支持向量机回归评价参数  $score$ 。更新公式如下:

$$\begin{aligned} C_{id}^{k+1} &= w \cdot C_{id}^k + c_1 rand_1 \cdot (p_{id} - C_{id}) + c_2 rand_2 \cdot (p_{gd} - C_{id}) \\ \gamma_{id}^{k+1} &= w \cdot \gamma_{id}^k + c_1 rand_1 \cdot (p_{id} - \gamma_{id}) \end{aligned} \quad (13)$$

$$+c_2rand_2 \cdot (p_{gd} - gamma_{id})$$

算法流程如下:

- (1) 根据预先设定的参数范围, 依据参数增长步长初始化微粒群, 置局部最优解, 全局最优解为 0
- (2) 根据更新公式分别调整粒子参数
- (3) 评价每个粒子的适应度
- (4) 对每个粒子, 将其适应值与其自我最优解  $p_{best}$  作比较, 如果较好, 则将其作为当前的自我最优解  $p_{best}$ ;
- (5) 对每个粒子, 将其适应值与全局最优解  $g_{best}$  作比较, 如果较好, 则将其作为当前的全局最优解  $g_{best}$ ;
- (6) 未达到结束条件则转(2)

## § 2.2 实例验证

针对上述章节所阐述的方法, 本节将进行实例验证。

### § 2.2.1 数据来源

本节选取美国加利福尼亚 PeMS 系统数据库中标号为 2554 的高速路交通流数据进行实验分析, 采样间隔为 5min, 选取时间为 2019 年 3 月 1 日至 2019 年 3 月 7 日, 共 2016 条速度, 流量数据。

对数据集进行划分, 将 2019 年 3 月 1 日至 2019 年 3 月 6 日共 1728 条数据, 用于训练支持向量机回归模型, 并用粒子群优化算法不断寻优, 找到最合适的参数对, 得到最佳模型。将 2019 年 3 月 7 日共 288 条数据作为测试样本, 用以检测模型是否具有较优的预测能力。

### § 2.2.2 效果分析

速度支持向量机回归模型预测结果如图 2-2 左图所示, 流量支持向量机回归模型预测结果图 2-2 右图所示, 从图中可知, 文中模型的预测结果与实际数据走势相同, 且吻合较好。

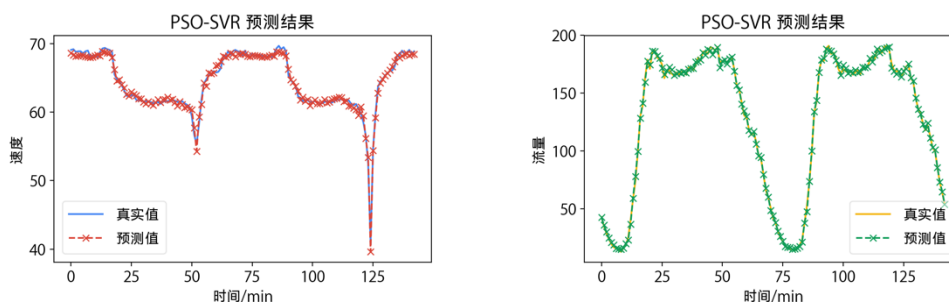


图 2-2 支持向量机回归预测结果

图 2-3 展示了粒子群优化算法结束时，找到的最佳参数值。针对速度支持向量机回归预测模型，最佳惩罚参数  $C = 246.931$ ，最佳高斯径向基核函数参数  $\gamma = 0.001$ ，针对流量支持向量机回归预测模型，最佳惩罚参数  $C = 87.221$ ，最佳高斯径向基核函数参数  $\gamma = 0.001$ 。

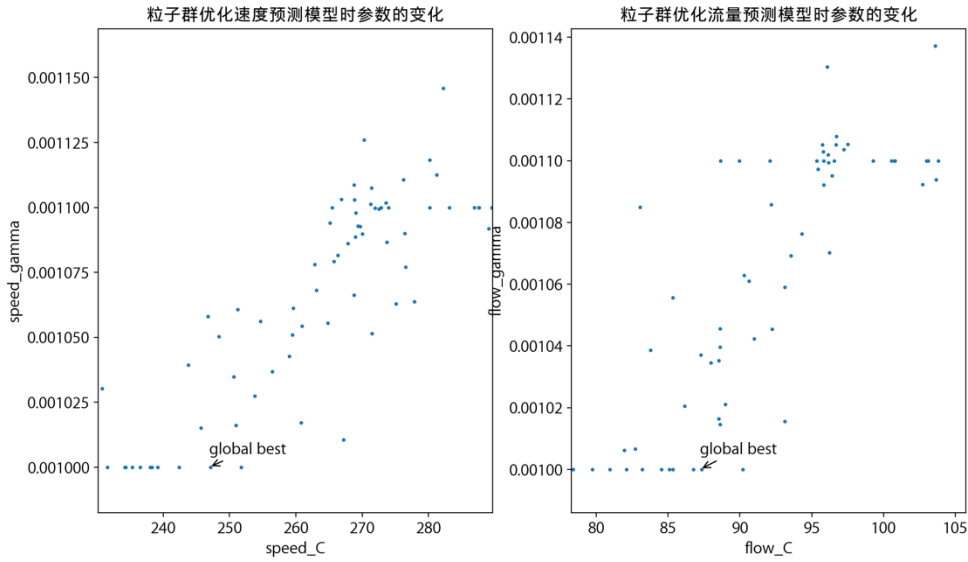


图 2-3 粒子群优化结果图

为了定量分析模型的预测效果，本文选取决定系数（coefficient of determination），均方误差（mean square error, MSE）和平均绝对百分比误差（mean absolute percent error, MAPE）作为评价指标，其公式如下

$$MSE = \frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2 \quad (14)$$

$$MAPE = \frac{1}{N} \sum_{i=1}^N \left| \frac{y_i - \hat{y}_i}{y_i} \right| \times 100\% \quad (15)$$

$$R^2 = 1 - \frac{\sum_{i=1}^N (y_i - \hat{y}_i)^2}{\sum_{i=1}^N (y_i - \bar{y})^2} \quad (16)$$

其中  $y_i$  表示真实值， $\hat{y}_i$  表示预测值。决定系数是一个评价拟合好坏的指标，用于度量应变量的总变异中可由自变量解释部分所占的比例，数值越大，自变量对因变量的解释程度越高，拟合效果越好。均方误差，平均绝对百分比误差是反映估计量与被估计量之间差异程度的一种度量，数值越小，表明误差越小，数值为零则表示完美模型。

为了探讨粒子群优化算法的优化效果，本文对比在训练集与测试集数据相同

的前提下，预测模型优化前后评价指标的差异。需要指出的是，未优化前的模型采用模型缺省参数。

表 2-1 粒子群优化算法前后决定系数差异表

	速度	流量
优化前	0.734	0.977
优化后	0.986	0.999

决定系数优化前后对比如表 2-1 所示。由表可知采用粒子群优化算法后，速度支持向量机回归预测模型决定系数提高明显，表明拟合效果增强。流量支持向量机回归预测模型决定系数改变不大，但数值非常高，可能存在过拟合现象。

表 2-2 粒子群优化算法优化前后评价指标差异表

	MSE		MAPE	
	速度	流量	速度	流量
优化前	4.824	86.153	1.785	17.883
优化后	0.257	3.724	0.635	1.828

MSE, MAPE 优化前后对比如表 2-2 所示。采用粒子群优化算法后，不论是速度支持向量机回归预测模型，还是流量支持向量机回归预测模型，评价参数 MSE 与 MAPE 下降明显，表明优化对预测模型精度的提高起积极作用。尤其是流量支持向量机模型的 MSE 参数下降最为明显，表明优化效果较优。由表可知，平均速度的 MSE 和 MAPE 分别为 0.257, 0.635%，结合模型决定系数，表明速度支持向量机回归预测模型精度高，预测效果好，能够很好的捕捉道路的交通流特点。交通流量的 MSE 和 MAPE 分别为 3.724, 1.828%，结合模型决定系数，表明虽然模型拟合效果很好，但依旧存在较高的偏差，说明流量支持向量机回归模型存在过拟合现象。

实验结果证明，粒子群优化算法优化效果明显，对预测模型的精度提升起积极促进作用。

## § 2.3 本章小结

基于支持向量机回归的交通流预测模型的构建，是短时交通流预测的基础。只有通过预测短时交通流参数，才能进行下一步同一时刻异地交通流参数预测。本章主要介绍了支持向量机回归模型的数学原理，建模过程，参数优化以及实验验证。通过对交通流数据进行特征提取，建立模型的输入输出向量，并划分训练集与测试集，实验发现影响模型预测精度的主要参数为惩罚参数  $C$  与高斯径向基核函数参数  $\gamma$ 。通过粒子群优化算法，对此参数进行自优化后得到最佳参数对，并重新训练得到最终的预测模型。实验对比发现，粒子群优化算法优化效果明显，该算法显著提升了预测模型的精度，降低了预测模型的出错概率。

最后将预测模型应用于测试集，发现拟合效果良好，实验成果符合预期。

### 第3章 基于长短期记忆网络的异地预测

本章是全文的重点章节，具体描述了基于长短期记忆网络（LSTM）的异地交通流参数预测方法：首先介绍循环神经网络与长短期记忆网络的基本概念，并引出本文所使用的基于长短期记忆网络的异地预测方法，阐述模型的搭建过程，最后进行实验模拟。

异地预测具有其现实意义，针对现实场景中下游路段因地理受限或实际因素考量而无数据采集设备，或数据采集设备损坏的状况，异地预测可以通过利用上游路段的历史交通流数据，构建异地预测模型，从而实现针对下游路段的交通流参数预测。

#### § 3.1 长短期记忆网络概述

##### § 3.1.1 循环神经网络与长短期记忆网络

介绍长短期记忆网络前，首先需要介绍循环神经网络（Recurrent Neural Network, RNN）。循环神经网络是一种用于处理序列数据的神经网络，常用于语音识别与机器翻译等方面。循环神经网络与普通的前馈神经网络相比，最突出的特点就是其具有短期记忆能力。在前馈神经网络中，信息的传递是单向向前的，而在循环神经网络中，神经元不但可以接受其他神经元的信息，还能接受自身信息，形成具有环路的网络结构。循环神经网络会对前面的信息进行记忆并应用于当前输出的计算中，这是由隐藏层之间节点的相互连接保证的。如图 3-1 所示，隐藏层的输入主要由输入层的输出与上一时刻隐藏层的输出组成。

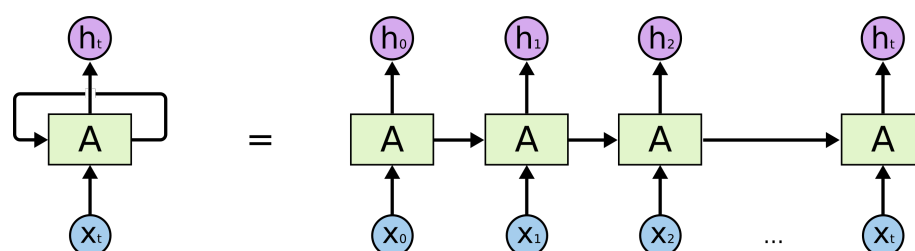


图 3-1 循环神经网络结构图及结构拆解图

循环神经网络的结构优势决定了其具有特殊的记忆能力，但因为其为每一个时间特征向量保持了对应的激活矢量，从而加大了循环神经网络的深度[12]。由于网络的参数学习主要依靠随时间方向传播算法，即按照时间的逆序将错误信息一步步地往前传递。随着深度的增加，循环神经网络会面临梯度爆炸与消失问题[13][14]，也称为长期依赖问题。虽然简单循环网络理论上可以建立长时间间隔的

状态之间的依赖关系，但是由于梯度爆炸与消失问题，实际上只能学习到短期的依赖关系。正由于循环神经网络不能有效的处理长期依赖问题，从而使得循环神经网络难以训练。

考虑到循环神经网络难以训练的性质，研究人员提出了许多改进形式，如双向循环神经网络(BiRNN)，深度循环神经网络(DRNN)，门控循环单元网络(GRU)与长短期记忆网络(LSTM)等，其中最成功的变体便是长短期记忆网络。长短期记忆网络(Long Short Term Memory networks, LSTM)是一种特殊的循环神经网络，它原则上可以通过显式门控机制和内置的恒定误差流在长时间内存储和检索信息[15]，从而有效解决了简单循环神经网络面临的梯度爆炸与消失问题。

### § 3.1.2 长短期记忆网络结构

长短期神经网络最重要的改进便是加入了门控机制和恒定误差流，如图 3-2 所示，恒定误差流指的是图片最上方的部分，在恒定误差流中进行线性循环信息传递的是内部状态 $\mathbf{c}_t$ ，其记录了到当前时刻 $t$ 为止网络记忆的历史信息。

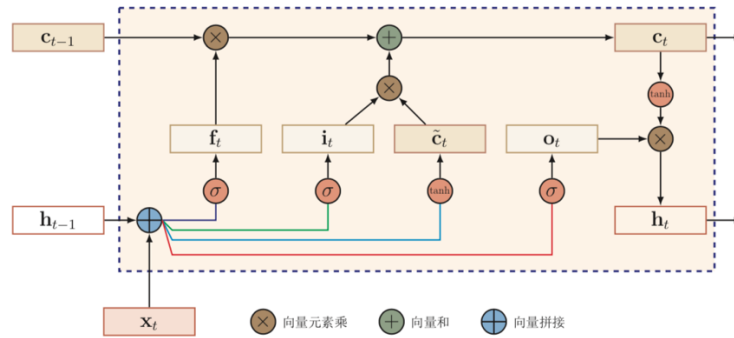


图 3-2 LSTM 单元内部结构图

门控机制主要分成三部分，分别为遗忘门 $\mathbf{f}_t$ ，输入门 $\mathbf{i}_t$ 与输出门 $\mathbf{o}_t$ ，遗忘门 $\mathbf{f}_t$ 控制上一个时刻的内部状态 $\mathbf{c}_{t-1}$ 需要遗忘哪些部分；输入门 $\mathbf{i}_t$ 控制当前时刻的候选状态 $\tilde{\mathbf{c}}_t$ 有多少信息需要保留；输出门 $\mathbf{o}_t$ 控制当前时刻的内部状态 $\mathbf{c}_t$ 有多少信息需要输出给外部状态 $\mathbf{h}_t$ 。三个门的计算公式如(17-19)所示

$$\mathbf{i}_t = \sigma(\mathbf{W}_i \cdot \mathbf{x}_t + \mathbf{U}_i \cdot \mathbf{h}_{t-1} + \mathbf{b}_i) \quad (17)$$

$$\mathbf{f}_t = \sigma(\mathbf{W}_f \cdot \mathbf{x}_t + \mathbf{U}_f \cdot \mathbf{h}_{t-1} + \mathbf{b}_f) \quad (18)$$

$$\mathbf{o}_t = \sigma(\mathbf{W}_o \cdot \mathbf{x}_t + \mathbf{U}_o \cdot \mathbf{h}_{t-1} + \mathbf{b}_o) \quad (19)$$

举例来说，当 $\mathbf{f}_t = 0$ 时，记忆单元将历史信息清空，反之 $\mathbf{f}_t = 1$ 时全部保留，否则保留部分历史信息。

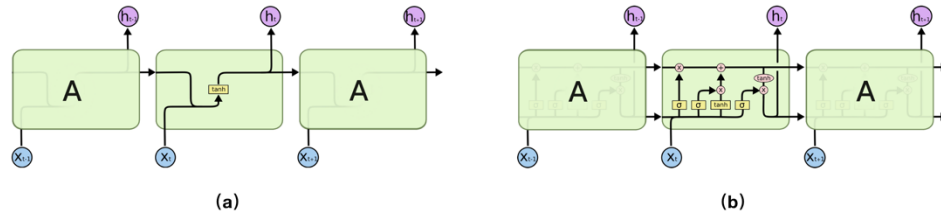


图 3-3 RNN 与 LSTM 内部结构对比图，图(a)为 RNN 内部结构，(b)为 LSTM 内部结构

如图 3-3(a) 所示，简单循环神经网络结构中，隐藏态  $h$  被完全重写，而图 3-3 (b) 长短期记忆网络结构中，内部状态  $c$  会被选择性保留。通过这一改变，长短期记忆网络解决了简单循环神经网络不能应付的长期依赖问题。

交通流数据是一种典型的序列数据，根据上文描述的长短期记忆网络的优点，本文将借助该网络模型，挖掘历史交通流数据的特点，进行异地预测。

### § 3.1.3 长短期记忆网络的搭建

本文选取速度与流量作为表征交通流数据的主要参数，并分别建立针对速度和流量的长短期记忆网络模型。

长短期记忆网络的输入为一个长度为  $\xi$  的时序序列  $\mathbf{x}_{1:\xi} = (\mathbf{s}_1, \mathbf{s}_2, \dots, \mathbf{s}_\xi)$ ，其保存了训练集中的所有数据。 $\mathbf{s}_i$  为一个长度为  $\eta$  的序列  $\mathbf{s}_{1:\eta} = (\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_\eta)$ ，称为观测周期序列，本文中采用  $\eta = 100$ ，即观测周期序列为 100 个时间采样点。 $\mathbf{z}_i$  为  $i$  时刻观测点的交通流数据，称为特征序列，在本文中为速度与流量组合，即  $\mathbf{z}_i = (v_i, q_i)$ 。基于长短期记忆网络的交通流预测模型就是寻找  $\mathbf{x}_{1:\xi}$  与交通流预测值  $y_i$  之间的关系。

$$y_i = f(\mathbf{x}_{1:\xi}) \quad (20)$$

图 3-4 为基于长短期记忆网络的建模流程图，具体描述如下：

- (1) 结合序列最大值与最小值，将数据归一化
- (2) 将上游交通流序列重构，作为长短期记忆网络的输入序列，将下游交通流序列重构，作为长短期记忆网络的标签序列
- (3) 基于上下游历史交通流数据，分别训练速度长短期记忆网络预测模型与流量长短期记忆网络预测模型
- (4) 调参优化模型
- (5) 预测结果逆归一化，得到最终异地预测结果



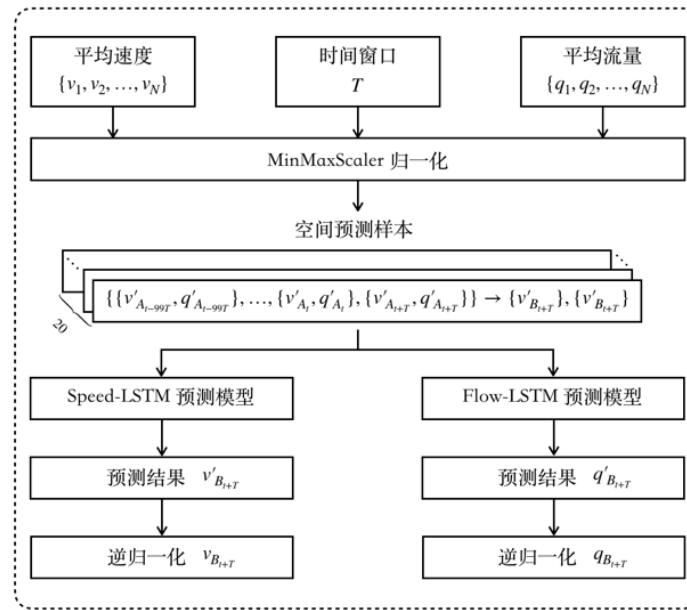


图 3-4 基于长短期记忆网络的建模流程图

本文所采用的速度长短期记忆网络的结构参数如表 3-1 所示，总共由 3 层 LSTM 循环结构与 1 层全连接层组成。流量长短期记忆网络的结构参数如表 3-2 所示，总共由 2 层 LSTM 循环结构与 1 层 dropout 层以及 1 层全连接层组成。需要指出的是，在实验过程中，本文对长短期记忆网络模型进行了多次调参优化，综合考虑运行环境限制，以下给出的网络结构，均为众多实验结果中最优的网络结构。

表 3-1 速度长短期记忆网络结构参数表

层（类型）	输出结构	变量数
lstm_1 (LSTM)	(32, 100, 32)	4480
lstm_2 (LSTM)	(32, 100, 32)	8320
lstm_3 (LSTM)	(32, 32)	8320
dense_1 (Dense)	(32, 1)	33
总参数: 21, 153	可训练参数: 21, 153	不可训练参数: 0

表 3-2 流量长短期记忆网络结构参数表

层（类型）	输出结构	变量数
lstm_1 (LSTM)	(32, 100, 64)	17152
lstm_2 (LSTM)	(32, 64)	33024
dropout_1 (Dropout)	(32, 64)	0
dense_1 (Dense)	(32, 1)	65
总参数: 50, 241	可训练参数: 50, 241	不可训练参数: 0

## § 3.2 实例验证

针对上述章节所阐述的方法，本节将进行实例验证。

### § 3.2.1 数据来源

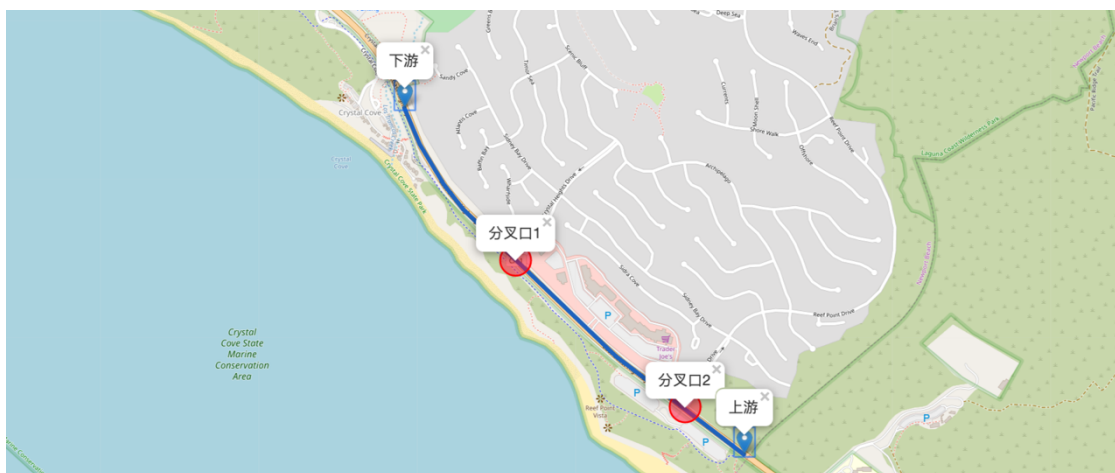


图 3-5 PeMS 第 2552 号公路段（下游）与第 2554 号公路段（上游）

本节选取美国加利福尼亚 PeMS 系统数据库中标号为 2552 和 2554 的高速路交通流数据进行实验分析，其中 2554 号路段为上游路段，2552 号路段为下游路段。采样间隔为 5min，选取时间为 2019 年 3 月 1 日至 2019 年 3 月 7 日，每个路段均为 2016 条速度，流量数据。

对数据集进行划分，将 2019 年 3 月 1 日至 2019 年 3 月 6 日共 3456 条数据，用于训练长短期记忆网络。将 2019 年 3 月 7 日共 576 条数据作为测试样本，用以检测模型是否具有较优的预测能力。

需要指出的是，本文选取的上下游路段之间存在两条分岔路段，分岔口车流的交汇对整条道路的交通流具有很强的干扰作用。正由于分岔路的存在，使得本文的异地预测具有现实意义。

### § 3.2.2 效果分析

长短期记忆网络可分为多种类型，针对本文的问题，可以根据输出向量类型的不同，划分为多对一网络和多对多网络。多对一网络的输出向量为单个数值，而多对多网络的输出向量为多值序列。

实验主要探讨了这两类网络类型的预测效果，具体划分为：

- (1) 多对一速度长短期记忆网络
- (2) 多对一流速长短期记忆网络
- (3) 多对多长短期记忆网络

针对多对一网络的实验结果，速度长短期记忆网络模型预测结果如图 3-6 左

所示，流量长短期记忆网络模型预测结果如图 3-6 右所示。从图中可知，网络模型的预测结果基本与实际数据相吻合，大致走势相同，表明模型性能良好。

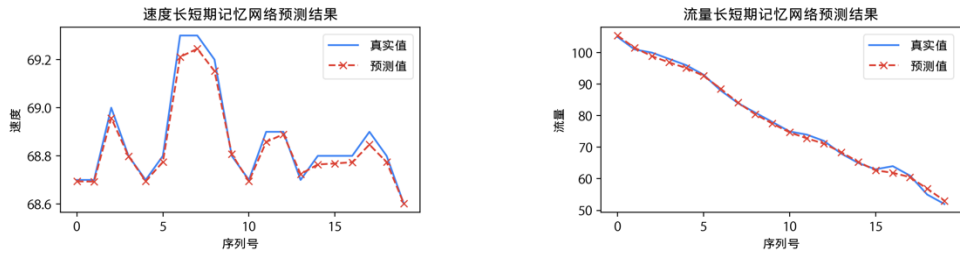


图 3-6 多对一长短期记忆网络预测结果图

针对多对多网络的实验结果如图 3-7 所示，从图中可知，多对多长短期记忆网络对于流量的预测结果与实际数据吻合较好，但速度预测偏离实际值较大，实验结果不理想。

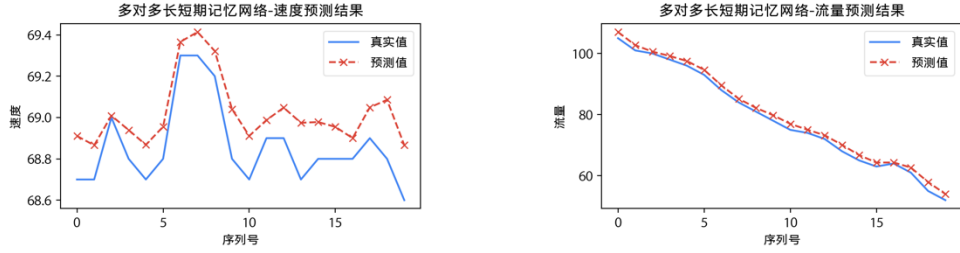


图 3-7 多对多长短期记忆网络预测结果图

为了定量分析模型的预测效果，选取决定系数（coefficient of determination），均方误差（mean square error, MSE）和平均绝对百分比误差（mean absolute percent error, MAPE）作为评价指标。

表 3-3 长短期记忆网络评价指标表

	决定系数	MSE	MAPE
多对一 LSTM 速度预测	0.967	0.0012	0.277
多对多 LSTM 速度预测	0.167	0.031	0.237
多对一 LSTM 流量预测	0.997	0.821	24.207
多对多 LSTM 流量预测	0.989	2.668	2.088

据表 3-3 发现，多对一网络中，速度长短期记忆网络预测模型表现优异，综合三个系数可以发现，模型并未出现过拟合现象。流量长短期记忆网络预测模型的 MAPE 数值较高，结合决定系数来看，存在过拟合现象。针对多对多网络，从决定系数来看，其速度预测精度低。模型的 MSE 值均比对应的多对一网络的 MSE 值要高，表明其预估偏差较大，考虑到多对多网络的输出为速度与流量序列组，速度预测的偏差将影响到最终输出精度，所以实验最终选取多对一长短期记忆网

络模型，并划分为独立的速度与流量模型进行训练。

实验结果证明，长短期记忆网络可以较好的挖掘交通流数据中的序列信息，比较有效的预测异地交通流参数。

### § 3.3 本章小结

基于长短期记忆网络的交通流预测模型的构建，是异地交通流预测的基础。通过上一步支持向量机回归预测模型预测得到的上游交通流参数，结合上游历史交通流数据，便可以依靠长短期记忆网络预估同一时刻下游交通流参数，从而最终完成异地短时交通流参数预测任务。

本章主要介绍了长短期记忆网络的主要原理，演变过程，并进行了实验验证。通过对上游历史交通流数据的特征提取，得到长短期记忆网络的输入向量，下游交通流数据则为标签集。训练长短期记忆网络是一个比较困难的过程，由于其训练成本较高，受限于实验环境，并不适合采用类似粒子群优化算法的辅助手段。另一方面，通过实验发现，多对多的长短期记忆网络预测偏差较大，分别针对速度与流量训练多对一长短期记忆网络更具备可行性。

通过实验发现，本文训练的速度长短期记忆网络及流量长短期记忆网络，拟合效果良好，结合评价参数分析，模型具备良好的预测能力，实验结果符合预期。

## 第4章 模糊综合评价

前两章已经实现了短时异地预测,即已知上游 A 地  $t$  时刻的流量 $q_A^t$ 与速度 $v_A^t$ ,可以估计 T 时间间隔后,下游 B 地的速度 $v_B^{t+T}$ 与流量 $q_B^{t+T}$ 。本章主要介绍如何运用模糊综合评价方法构建交通流拥堵等级评价体系,并在获得预测参数后实现实时评价。

在交通拥堵模糊综合评价中,需确定影响交通流拥堵情况的主要因素,确定评价指标集,评价集,隶属函数,然后通过计算各因素的权重和隶属度,得到综合隶属度,确定拥堵级别[16],具体过程如图 4-1 所示。

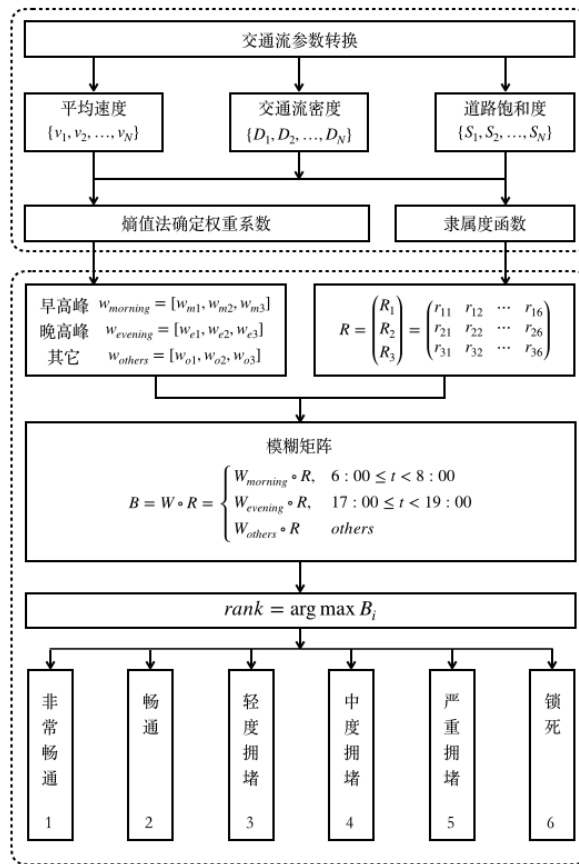


图 4-1 模糊综合评价建模过程

### § 4.1 评价指标选取

选取能够准确有效地表征交通拥堵状态的指标,对交通拥堵评估系统的科学性至关重要。为了避免单一因素评估的缺陷,本文选取交通流平均速度、交通流密度、路段饱和度三个交通流参数作为因素指标[7]。

交通流平均速度指的是某一时刻固定路段上所有车辆行驶速度的平均值。该

指标可以直观地反映道路当前的拥堵状态。平均速度与道路拥堵状态呈负相关,一般而言,速度越小,道路越拥堵,速度越大,道路越通畅。具体计算公式如下:

$$\bar{v}_t = \frac{1}{N} \sum_{t=1}^N v_t \quad (21)$$

其中  $N$  为当前时刻道路中的车辆总数,  $v_t$  为每辆车的瞬时速度,  $\bar{v}_t$  为当前时刻路段交通流的平均速度, 单位为英里每小时 (mph)。

交通流密度是指在某一时刻, 单位长度道路上存在的车辆数。它直接地反映了交通需求量。交通流密度与道路拥堵状态呈正相关, 一般而言, 交通流密度越大, 道路越拥堵, 交通流密度越小, 道路越畅通。交通流密度还可以近似地用来衡量驾驶员操纵车辆的舒适性和灵活性。具体计算公式如下:

$$D = \frac{N}{L} = \frac{f}{v} \quad (22)$$

其中  $N$  为当前时刻道路中的车辆总数,  $L$  为观测路段长度, 在已知每小时车流量  $f$  与交通流平均速度  $v$  的情况下, 密度等于两者的比值。

路段饱和度是描述道路交通负荷程度的指标, 指的是某一路段实际交通流量与该路段的通行能力的比值。具体计算公式如下:

$$S = \frac{Q}{C} \quad (23)$$

其中  $Q$  是道路当前时刻的车流量,  $C$  为路段的最大通行能力, 结合美国《道路通行能力手册》, 本文中  $C$  选取为 183。

## § 4.2 模糊评价模型

由于对于道路拥堵情况判别的边界并不十分明显, 评价时很难将其归于某个具体类别。本文采用先对单个因素进行评价, 然后对所有因素进行模糊综合评价的方法, 防止遗漏任何统计信息和信息的中途损失。该方法有助于解决评估对客观真实的偏离问题, 避免了主观性判断对实验的影响。

### § 4.2.1 熵值法赋权

在信息论中, 熵是系统无序程度的度量, 它还可以度量数据所提供的有效信息量[17]。熵值法赋权是指通过计算熵值大小, 来确定赋予评价指标的权重系数。当某项指标的值相差较大时, 即指标的离散程度较高, 则熵值较小, 表明该指标提供的有效信息量较大, 应该赋予高权重。反之, 当某项指标的值相差较小时, 即指标的离散程度较低, 则熵值较大, 表明该指标提供的有效信息量较小, 应该赋予低权重。所以熵值法赋权是一种比较客观的赋权方法, 避免了人为主观因素的干扰, 保证了系统的科学性。

以本文为例，熵值法赋权用于确定在早高峰，晚高峰以及其他时段时，交通流平均速度、交通流密度、路段饱和度三个交通流参数各自所占的权重，权重越高，说明对当前道路拥堵级别的确定贡献越大，反之贡献越小。

通过对 PeMS 数据中各个交通流参数进行预分析发现，交通流数据大致上存在一定的往复规律性，早晚高峰时段较固定。以 2019 年 4 月 8 日第 2554 号检测路段当日数据为例，如图 4-2 所示，在早高峰期间，道路平均速度下降明显，早高峰结束后，速度较稳定，在某一数值周围小幅波动，道路流量在早高峰期间急剧上升并达到峰值，早高峰后流量下降，并小幅波动上升。在晚高峰期间，道路平均速度陡然下降并触及最低值，表明晚高峰期间拥堵最为严重，道路流量小幅下降，变化不如速度明显。

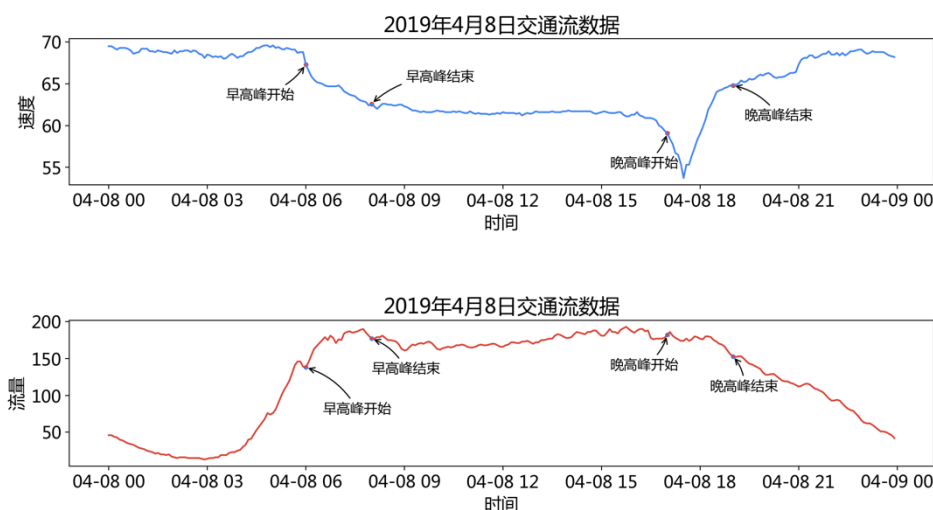


图 4-2 某日道路交通流数据

经过数据预处理，可以发现交通拥堵状态基本发生在早晚高峰时段，符合常规预期。所以在熵值法赋权时，采用三个权重系数分别表征不同时段的道路交通特征，即早高峰，晚高峰与其他时段。

熵值法赋权的主要步骤如下：

- (1) 将历史交通流数据按照早高峰(6:00 到 8:00)，晚高峰(17:00 到 19:00)以及其他时段划分数据集。
- (2) 设有  $m$  个评价指标，共  $n$  个评价对象，则形成原始数据矩阵  $X$ ，其中  $x_{ij}$  表示第  $i$  个评价对象的第  $j$  个评价指标。

$$X = \begin{bmatrix} x_{11} & x_{21} & \cdots & x_{m1} \\ x_{12} & x_{22} & \cdots & x_{m2} \\ \vdots & \vdots & \ddots & \vdots \\ x_{1n} & x_{2n} & \cdots & x_{mn} \end{bmatrix} \quad (24)$$

- (3) 原始数据矩阵  $X$  进行标准化, 评价指标可分为正向指标和负向指标。在本文选取的三个评价指标中, 交通流平均速度属于负向指标, 速度越大, 道路拥堵等级越低, 速度越小, 道路拥堵等级越高。针对负向指标, 标准化计算公式如下:

$$x'_{ij} = \frac{\max\{x_{1j}, \dots, x_{nj}\} - x_{ij}}{\max\{x_{1j}, \dots, x_{nj}\} - \min\{x_{1j}, \dots, x_{nj}\}} \quad (25)$$

交通流密度、路段饱和度属于正向指标, 数值越大, 道路拥堵等级越高, 数值越小, 道路拥堵等级越低。针对正向指标, 标准化计算公式如下:

$$x'_{ij} = \frac{x_{ij} - \min\{x_{1j}, \dots, x_{nj}\}}{\max\{x_{1j}, \dots, x_{nj}\} - \min\{x_{1j}, \dots, x_{nj}\}} \quad (26)$$

最终生成标准化数据矩阵  $X'$ :

$$X' = \begin{bmatrix} x'_{11} & x'_{21} & \cdots & x'_{m1} \\ x'_{12} & x'_{22} & \cdots & x'_{m2} \\ \vdots & \vdots & \ddots & \vdots \\ x'_{1n} & x'_{2n} & \cdots & x'_{mn} \end{bmatrix} \quad (27)$$

其中  $x'_{ij}$  表示第  $i$  个评价对象的第  $j$  个评价指标的标准值。

- (4) 计算熵值。

$$H_i = -k \sum_{j=1}^N p_{ij} \ln(p_{ij}) \quad (28)$$

$$p_{ij} = \frac{x'_{ij}}{\sum_{i=1}^N x'_{ij}} \quad (29)$$

$$k = \frac{1}{\ln N} \quad (30)$$

其中, 式(29)为第  $i$  评价对象的第  $j$  个评价指标所占该指标的比重。式(28)表示第  $j$  项评价指标的熵值。

- (5) 根据熵值计算权重。计算公式如下:

$$g_j = 1 - H_i \quad (31)$$

$$w_j = \frac{g_j}{\sum_{j=1}^m g_j} \quad (32)$$

根据熵的概念, 熵值越小, 该指标提供的有效信息量较大, 应该赋予高权重, 如式(31)所示。式(32)表示各项因素所占的权重, 并且保证权重之和为 1。

- (6) 运用相同的方法, 计算晚高峰时段及其他时段三种评价指标的权重系数。



需要说明的是，熵权并不是表示决策评估问题中某指标实际意义上的重要性系数，而是在给定评价对象集后各种评价指标值确定的情况下，各指标在竞争意义上的相对激烈程度。从信息角度考虑，它代表该指标在该问题中，提供有效信息量的多寡程度[16]。

### § 4. 2. 2 确定评价指标隶属度

结合本文所研究的交通流数据，确定本文采用的三个评价指标对应拥堵等级的划分范围，如表 4-1 所示。

表 4-1 交通拥堵等级划分范围

各项指标	拥堵等级					
	非常畅通	畅通	轻度拥堵	中度拥堵	严重拥堵	锁死
	1	2	3	4	5	6
平均速度/mph	$(80, +\infty)$	$(70, 80]$	$(60, 70]$	$(50, 60]$	$(40, 50]$	$[0, 40]$
密度/辆/英里	$(0, 10]$	$(10, 20]$	$(20, 30]$	$(30, 40]$	$(40, 50]$	$(50, +\infty)$
路段饱和度	$(0, 0.25]$	$(0.25, 0.4]$	$(0.4, 0.55]$	$(0.55, 0.7]$	$(0.7, 0.85]$	$(0.85, 1]$

隶属度函数用于计算某一个评价指标隶属于某个等级的概率，即对每个评价指标  $j$  求得其隶属于第  $k$  个等级的隶属度  $b_{jk}$ 。为了减少主观性，本文采用的隶属度函数具体形态如图 4-3 所示：

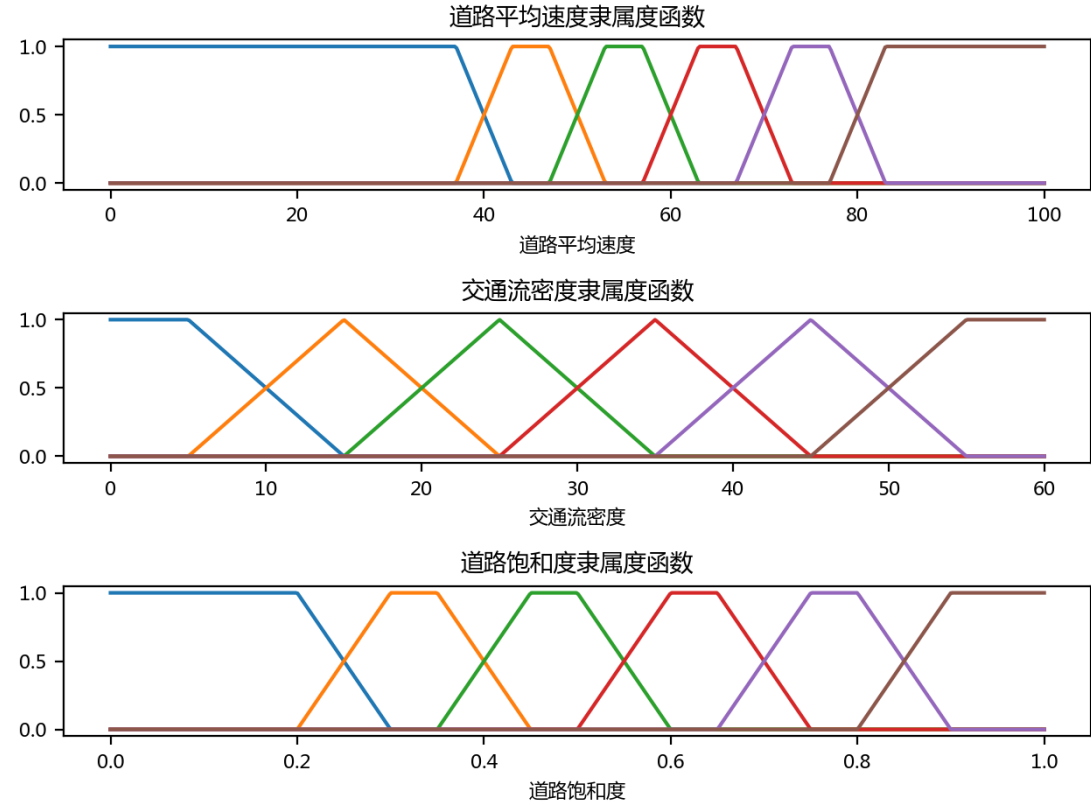


图 4-3 隶属度函数

其中横轴表示评价指标的值,纵轴表示隶属度,  $y = 0$  表示不属于该等级,  $y = 1$  表示绝对属于该等级。由于评价指标分为正向指标与负向指标,对于正向指标,函数从左到右拥堵等级递增,对于负向指标,函数从左到右拥堵等级递减。

通过计算,最终得到评价矩阵如下:

$$R = \begin{bmatrix} r_{11} & r_{21} & \cdots & r_{m1} \\ r_{12} & r_{22} & \cdots & r_{m2} \\ \vdots & \vdots & \ddots & \vdots \\ r_{1n} & r_{2n} & \cdots & r_{mn} \end{bmatrix} \quad (33)$$

### § 4. 2. 3 拥堵级别确定

将评价指标权重与评价矩阵作模糊合成运算,最终得到模糊综合评价矩阵,计算公式如下:

$$B = W \circ R = \begin{cases} W_m \circ R, & 6:00 \leq t < 8:00 \\ W_e \circ R, & 17:00 \leq t < 19:00 \\ W_o \circ R, & \text{其他} \end{cases} = (b_1, b_2, \dots, b_m) \quad (34)$$

其中,  $W_m, W_e, W_o$  分别表示一天中早高峰,晚高峰,其他时段的权重系数。根据最大隶属度原则,选取模糊综合评价矩阵中最大值所属的级别为最终的拥堵级别,即:

$$\text{rank} = \arg \max(b_1, b_2, \dots, b_m) \quad (35)$$

## § 4. 3 实例验证

本节将运用真实数据,具体展示模糊综合评价。

### § 4. 3. 1 数据来源

本节选取美国加利福尼亚 PeMS 系统数据库中标号为 2554 的高速路交通流数据进行权重实验分析。采样间隔为 5min,选取时间为 2019 年 3 月 1 日至 2019 年 3 月 7 日,共 2016 条速度,流量数据。选取系统数据库中标号为 2552 的高速路交通流数据进行拥堵等级评估实验分析。采样间隔为 5min,选取时间为 2019 年 4 月 9 日,共 288 条速度,流量数据。

### § 4. 3. 2 权重计算

针对训练数据进行时段划分,经过熵值法赋权计算,计算结果如表 4-2 所示。

表 4-2 评价指标权重

	平均速度权重	交通流密度权重	道路饱和度权重
早高峰	0.452	0.290	0.258
晚高峰	0.921	0.075	0.005
其他时段	0.384	0.320	0.296

从表中可以发现，早高峰时段，平均速度权重较大，即表明平均速度很大程度上决定了当前时刻道路的拥堵等级所属；晚高峰时段，平均速度占比 0.92，表明平均速度完全决定了拥堵等级所属，观察图 4-2 某日道路交通流数据可以发现，晚高峰时段平均速度变化十分明显。其他时段，三种评价指标权重相近，共同影响拥堵等级的判定。

### § 4. 3. 3 过程举例

以下游路段 2019 年 4 月 9 日 19:05:00 时刻为例，通过短时异地预测模型，预估该时刻平均速度为 63.424，流量为 157.065。运用模糊综合评价方法评估拥堵级别过程如表 4-3 所示：

表 4-3 模糊综合评价运算举例						
	预测值					真实值
时间	19:05:00					19:05:00
速度	63.424					64.8
流量	157.065					159
评价指标向量	[63.424, 0.206, 0.858]					[64.8, 0.204, 0.869]
评价矩阵	$\begin{bmatrix} 0 & 0 & 0.842 & 0.158 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.417 & 0.583 \end{bmatrix}$					$\begin{bmatrix} 0 & 0 & 0.98 & 0.02 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.311 & 0.689 \end{bmatrix}$
时段类型	其他时段					其他时段
权重系数	[0.384, 0.320, 0.296]					[0.384, 0.320, 0.296]
模糊综合评价矩阵	[0.32, 0, 0.323, 0.0607, 0.123, 0.173]					[0.32, 0, 0.376, 0.008, 0.092, 0.204]
拥堵等级	3					3

### § 4. 3. 4 效果分析

通过如上方法，模糊综合评价方法可以估计每一个时间点的拥堵等级。实验过程中采用短时异地预测模型所得到的交通流参数为实验组，同一时刻的交通流参数真实值为对照组，进行拥堵等级评估，具体结构如图 4-4 所示。

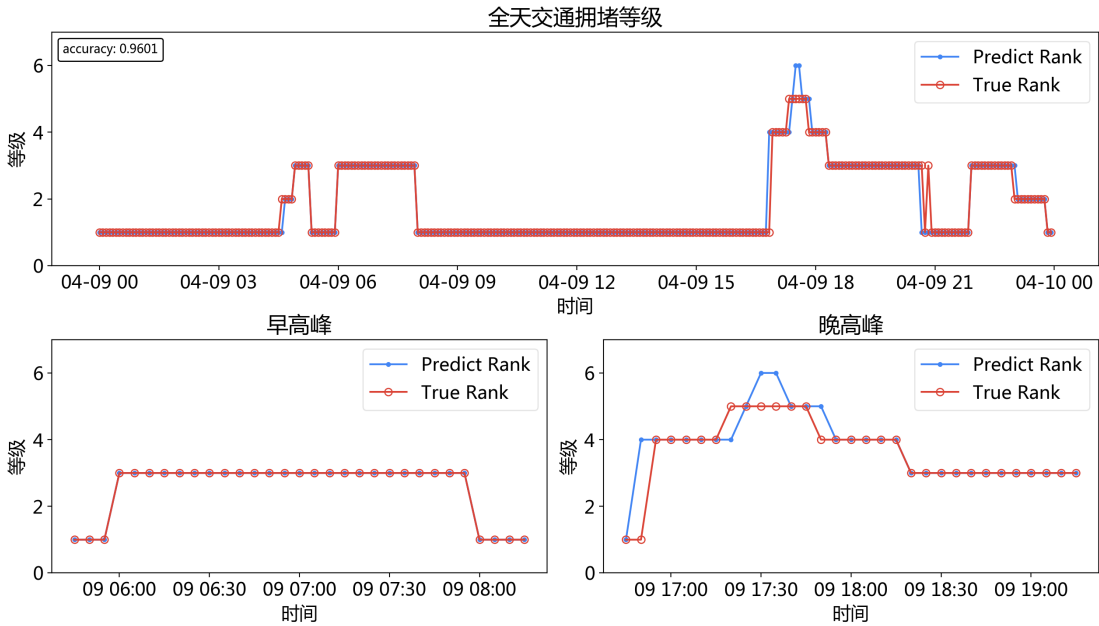


图 4-4 全天拥堵等级表

实验结果表明，2552 号路段道路通行总体畅通。在非高峰时段，拥堵等级为 1，即非常畅通的时刻占大多数。早高峰时段拥堵等级明显上升，为等级 3，即轻度拥堵。晚高峰时段，拥堵等级变化明显，且存在峰值状态，依次经过中度拥堵，严重拥堵，轻度拥堵状态，符合现实逻辑。

为了定量分析模型的预测效果，选取正确率（accuracy），均方误差（mean square error, MSE），和平均绝对百分比误差（mean absolute percent error, MAPE）作为评价指标。

表 4-4 模糊综合评价模型评价指标表

	正确率	MSE	MAPE
模糊综合评价	0.96	0.1134	2.9272

由表 4-4 可知，模糊综合评价模型正确率达 0.96，MSE 值为 0.1134，表明评价误差较小。考虑到模型实际运用中，当预估等级小于真实等级时，会出现无法及时防范拥堵的问题，而当预估等级高于真实等级时，虽然并未发生拥堵，但依旧属于有效警报，通过计算，有效警报占比为 0.977，表明系统具备要求的预警能力，且误差在允许范围内，系统表现优异。

## § 4.4 本章小结

本章主要介绍了运用模糊综合评价方法评价当前道路交通拥堵级别。该方法的运用减少了对拥堵等级分类的主观性判断。熵值法赋权可以有效分析历史交通流数据中各评价指标的权重系数，是模糊综合评价中至关重要的一步。隶属度函数的确定，客观地划分了各评价指标的等级范围。模糊综合评价矩阵表示了该评价对象隶属于各个级别的概率，最终根据最大隶属度原则确定交通拥堵级别。

实验表明，该方法可以准确地评价当前道路拥堵级别。

## 第5章 总结与展望

本章对全文的主要工作和创新点作了总结，并提出需要进一步研究和改进之处。

### § 5.1 本文总结

本文主要目的为构造交通拥堵预警模型，围绕交通流短时预测与异地预测两大场景，提出了一种基于支持向量机回归与长短期记忆网络的短时异地预测模型，运用模糊综合评价方法评估最终拥堵等级，并设置拥堵预警阈值，为交通拥堵预警系统的开发提供技术支持。

具体为，通过对上游路段历史交通流数据的学习，支持向量机回归预测模型可以准确地预估未来时刻上游路段道路平均速度与交通流量。结合预测结果与历史数据，长短期记忆网络可以准确预估下游路段的道路平均速度与交通流量。最终预测数据经过交通流参数转化后得到包括平均速度，交通流密度，道路饱和度在内的评价指标集。通过熵值法赋权确定早晚高峰及其他时段下各指标的权重系数，并利用隶属度函数确定各指标的隶属度，最终根据最大隶属度原则得到道路交通拥堵等级，若达到阈值则发出警报。

本文通过对美国加利福尼亚 PeMS 系统数据库中标号为 2552 和 2554 的高速路交通流数据进行实验分析后证明，该系统具有较高的预测精度，正确率可达到 96%。

#### § 5.1.1 本文的主要工作

本文的主要工作可分为以下三大块：

##### (1) 短时预测

- a) 分别构建了速度支持向量机回归预测模型和流量支持向量机回归预测模型，通过对以当前时刻为基准，包含过去三个时间窗口的交通流数据的学习分析，预测同一地点下一个时间窗口的交通流参数。
- b) 运用粒子群优化算法，综合优化支持向量机回归模型的模型参数，通过模拟粒子的迭代寻找，不断逼近全局最优参数值。

##### (2) 异地预测

- a) 分别构建了速度长短期记忆网络预测模型和流量长短期记忆网络预测模型，通过对上游与下游历史交通流数据的综合学习，预测同一时刻，不同地点的交通流参数。

- b) 根据经验对模型进行参数调整优化。
- (3) 模糊评价
  - a) 提出了交通拥堵等级综合模糊评价方法。运用熵值法赋权学习历史交通流数据，确定不同时段评价指标的权重系数。针对每个评价对象，采用隶属度函数确定评价矩阵，结合权重系数及最大隶属度原则，确定最终拥堵等级。

### § 5.1.2 本文的主要创新点

本文基于回归的交通拥堵预警系统构建方法创新点主要有以下两项：

- (1) 实现了运用粒子群优化算法优化短时预测模型，避免了人工调参的局限性，极大优化了预测模型的准确度。
- (2) 实现了基于长短期记忆网络的异地预测，为缺乏样本数据的现实场景提供了解决方案。
- (3) 采用综合模糊评价评估拥堵等级，区分不同时段并根据熵值法赋予不同的权重，客观评估拥堵等级，与传统按照等级表评估相比更为灵活。

### § 5.2 展望

虽然本文实现了基于回归的交通拥堵预警系统，但仍存在不少需要改进的地方：

- (1) 现有的预测模型建立在时间窗口为五分钟的基础上，此时间窗口的设定使得交通流数据之间存在较高的关联度，便于预测。随着时间窗口的扩大，数据关联度下降，如何训练时间窗口较长的模型，以及如何优化，是未来系统需要考虑的一个方向。
- (2) 异地预测模型内包含了分岔口信息，虽然长短期记忆网络模型表现优异，但缺乏对于分岔路口的解释性，即无法知晓网络内部是如何表征分岔路口信息的。在多岔路，甚至立交桥等实际模型中，每个分岔口的权重可能是不一样的，而现有模型因为缺乏可解释性，所以难以针对分岔口设置参数调参。网络模型的可解释性，路网的相关性分析，是一个难点也是一个可以研究的方向。
- (3) 本文对交通流信息中速度部分的预测取得了很好的成果，但在流量预测方面，存在过拟合现象。虽然经过参数调整，依旧不能很好的避免这一现象，说明还需要进一步考虑特征提取，从而避免模型过拟合，这是本文需要改进的一个地方。
- (4) 本文受限于实验环境，未采用优化算法对长短期记忆网络模型调参优

化，而是根据他人经验设置参数值，并未采用类似于粒子群优化算法等自优化算法。探寻合适的优化算法可以更好的发挥长短期记忆网络的优势。

- (5) 本文只考虑平均速度以及道路流量作为表征交通流数据的特征值，并未考虑其他交通流参数。交通参数复杂多样，综合考虑其他因素可以更有效的表征当前交通流特征。
- (6) 天气因素是影响交通拥堵状况的一个强因子，比如雨天不确定因素增加，导致道路更容易拥堵；大雾天因能见度差，车辆行车缓慢，但并未发生拥堵，按照原先等级划分效果可能不佳。本文因缺乏样本数据，并未考虑天气因素对于交通的影响，将天气因素纳入建模过程是下一步可以尝试的方向。
- (7) 模型迁移也是可以考虑的下一个尝试方向。文中模型针对 PeMS 数据库表现优异，但出现了过拟合现象，如何有效地将模型迁移至其他数据库，是一个值得探究的方向。
- (8) 预警系统由多个独立的模块组成，每个模块都是一个独立的任务，其结果的好坏会影响到下一步骤，从而影响整个系统的精度。如何精简模块，或模块的融合是本次实验最困难的地方，需要较长的时间改进，是一个非常值得研究的方向。



## 致谢

时光荏苒，四年的本科生活很快就要结束了，还记得当年暑假收到上海大学录取通知书时候的欣喜与激动，也还记得初到上海大学时候的兴奋与茫然。在这里的四年，充满了许多美好的回忆，也经历了许多的困难与磨练。四年的时间，看似很长，实则弹指一挥间。在我的毕业设计论文即将完成之际，我要衷心地感谢很多人。

首先，要特别感谢我的父母和家人。我亲爱的父母不仅向我提供了经济上的支持，更是给予了我精神上的鼓励。每当我生活上遇到烦心事，或者学习上遭遇挫败的时候，父母总是能在电话的另一端鼓励我。特别是在我考研期间，面对精神上巨大的压力，正是来自父母和家人的鼓励，使我顺利地度过这一人生的磨炼期。

其次，我要非常感谢我的指导老师吴绍春老师。在整个毕设进行过程中，不论是前期的问题确定，还是中期的模型调整，抑或是后期的论文修改，吴老师都给予了我很多宝贵的建议。每周一次的组会我总能得到对自己毕设的精进意见，这对我毕设的完成起到了决定性作用。感谢岳晓冬老师，非常荣幸能在大二阶段获得老师为期一年的辛勤指导。正是因为岳老师对我的影响，使我对于机器学习有着浓厚的兴趣，并决定在研究生阶段继续钻研学习。感谢方昱春导师在大三阶段的辛勤指导。正是由于方老师的鼓励，我们发表了自己的第一篇论文，方老师不仅指导我们实验，更是与我们一同修改论文的不足之处，其平易近人的性格给我留下了十分深刻的印象。感谢郑宇老师，郑老师为人亲和，一直是学生们最喜欢的导师，当年早上五点起床去教务处加课才有幸能加入郑老师的课堂。特别感谢郑老师在我考研期间给予我的巨大支持，老师对我大量疑惑的专门解答使我对数据结构有了更为透彻的理解。非常感谢以上三位老师为我写的推荐信，能得到老师们的认可，实乃我之荣幸。还有许多其他老师需要感谢，各位老师都给了我很多帮助，启发了我对人生的思考，真的万分感谢。

感谢黄加新学长对本次实验给予的支持，每当遇到瓶颈期时，学长总是能和我一同探讨模型的问题所在，使我总能及时调整模型，最终顺利完成毕业设计。

我也非常感谢的我同学和朋友。首先最为感谢顾晨屹同学。我与顾同学结缘高数课堂，他是我认识最早的同学，也是我大学的挚友，也是这个世界上唯一能用脸解开我手机的人。尤其感谢顾同学考研期间的陪伴，一起考研的经历必将永生难忘。感谢李如鹏，陈航天同学三年来给予我的无私帮助与关怀，感谢你们带我吃遍上海的大小餐馆。感谢王宇吉同学总是能在我低落，焦躁，怀疑自己的时

候时给予我最温暖的安慰。感谢远在延长不常见面的倪云川同学平日里给予我的鼓励与坦诚。感谢孙旭日同学给予我的支持，信任与包容，我天生不合群，有过自我质疑，也有到过崩坏的边缘。总觉得合也无味，孤也无味，本以为来这世间一趟，我将会匆匆走过，但他的存在告诉我，我并不孤独，亦有陪伴，希望他一直都是我的好朋友。感谢黄正章、常文昊、赵子衡、陈阳、周豪磊考研六人组以及苏文知同学的陪伴，使我在前进的路上不孤独，尤其感谢黄正章同学考研一整个暑假的陪伴，让我搭了一个暑假的顺风车，最重要的是因为他，我终于可以不用一个人吃午饭了。感谢远在东北的俞建聪同学与我隔空一起考研，讨论专业课，互相鼓励互相帮助，感谢他的乌鸡汤总能让我认清自己不再矫情。感谢王子豪、谭杰文、赵鹏伟、冯熙来、张鸿元、倪加成、王微、杨铭韬同学四年来的照顾。感谢陈弘毅同学给予本实验可视化方面的技术支持。还有许多其他的同学和朋友需要感谢，是你们陪伴我度过了人生中最珍贵的四年，是你们给我平日的生活带来很多的乐趣，同时也感谢你们让我收获到最珍贵的友谊。

最后，感谢在百忙之中抽空参与论文审阅与答辩的各位老师，谢谢你们！

## 参考文献

- [1] Van Der Voort M, Dougherty M, Watson S. Combining Kohonen maps with ARIMA time series models to forecast traffic flow[J]. Transportation Research Part C: Emerging Technologies, 1996, 4(5): 307-318.
- [2] 周宏,刘钊.基于时间序列的高速公路短时交通流最优组合预测[J].工程与建设,2018,32(06):812-814.
- [3] 宗春光,宋靖雁,任江涛,胡坚明.基于相空间重构的短时交通流预测研究[J].公路交通科技,2003(04):71-75.
- [4] 杨飞. 基于回声状态网络的交通流预测模型及其相关研究[D].北京邮电大学,2012.
- [5] 谭娟,王胜春.基于深度学习的交通拥堵预测模型研究[J].计算机应用研究,2015,32(10):2951-2954.
- [6] 傅贵,韩国强,逯峰,许子鑫.基于支持向量机回归的短时交通流预测模型[J].华南理工大学学报(自然科学版),2013,41(09):71-76.
- [7] 晏雨婵,白璘,武奇生,叶珍.基于多指标模糊综合评价的交通拥堵预测与评估[J/OL].计算机应用研究:1-7[2019-05-07].<http://kns.cnki.net/kcms/detail/51.1196.TP.20181011.1259.032.html>.
- [8] Cheng A, Jiang X, Li Y, et al. Multiple sources and multiple measures based traffic flow prediction using the chaos theory and support vector regression method[J]. Physica A: Statistical Mechanics and its Applications, 2017, 466: 422-434.
- [9] 邵春福,熊志华, and 姚智胜. "道路网短时交通需求预测理论, 方法及应用." (2011): 78-82.
- [10] 杨维,李歧强.粒子群优化算法综述[J].中国工程科学,2004(05):87-94.
- [11] Olah C. Understanding lstm networks[J]. 2015.
- [12] Jozefowicz, Rafal, Wojciech Zaremba, and Ilya Sutskever. "An empirical exploration of recurrent network architectures." International Conference on Machine Learning. 2015.
- [13] Hochreiter, Sepp. Untersuchungen zu dynamischen neuronalen netzen. Master's thesis, Institut für Informatik, Technische Universität München, 1991.
- [14] Bengio, Yoshua, Simard, Patrice, and Frasconi, Paolo. Learning long-term dependencies with gradient descent is difficult. Neural Networks, IEEE Transactions on, 5 (2):157-166, 1994.
- [15] Karpathy A, Johnson J, Fei-Fei L. Visualizing and understanding recurrent networks[J]. arXiv preprint arXiv:1506.02078, 2015.
- [16] 邹志红, 孙靖南, 任广平. 模糊评价因子的熵权法赋权及其在水质评价中的应用[D]. , 2005.

- [17] 孟庆生. 信息论[M]. 西安: 西安交通大学出版社, 1989: 19 —36

## 附录：部分源程序清单

算法：粒子群算法 + 支持向量机回归

```
import configparser
import matplotlib.pyplot as plt
import numpy as np
import pickle
import sys
import json
from sklearn.metrics import mean_squared_error
from sklearn.metrics import
mean_absolute_error, r2_score, mean_squared_error
from sklearn.model_selection import train_test_split
from sklearn.pipeline import Pipeline
from sklearn.preprocessing import StandardScaler
from sklearn.preprocessing import MinMaxScaler
from sklearn.svm import SVR
from utils import get_SVR_input
import time
c_pair_plot=[]
gamma_pair_plot=[]
#-----PSO_SVR
class-----#

class PSO_SVR:
    """
    在给定数据集上训练一组 SVR，获取 SVR 的最佳超参数 C 和 gamma
    """

    def __init__(self, config_file, type):
        # 读取配置文件.
        config = configparser.ConfigParser()
        config.read(config_file)

        # PSO 参数.
        pso_config = config["PSO"]
        self.n_iterations = int(pso_config.get("n_iterations",
100))
        self.inertia_wt = float(pso_config.get("inertia_weight",
1))
        self.c1 = float(pso_config.get("c1", 2))
        self.c2 = float(pso_config.get("c2", 2))
```

```

# 验证环节 参数.
val_config          = config["Validation"]
self.validation_type = val_config.get("type", "random-split")
self.n_validations   = int(val_config.get("n_validations",
10))

self.validation_size =
float(val_config.get("validation_size", 0.1))

# SVR 参数.
svr_config          = config["SVR"]
self.kernel         = svr_config.get("kernel", "rbf")
self.epsilon        = float(svr_config.get("epsilon", 0.1))
self.C_min          = float(svr_config.get("C_min", 1))
self.C_max          = float(svr_config.get("C_max", 100))
self.C_step         = float(svr_config.get("C_step", 10))
self.gamma_min      = float(svr_config.get("gamma_min", 0.01))
self.gamma_max      = float(svr_config.get("gamma_max", 2))
self.gamma_step     = float(svr_config.get("gamma_step",
0.1))

# 数据路径
data_config          = config["Data"]

# 读取数据
self.X, self.Y, _, _ = get_SVR_input('train_up', type)
# 初始化所有粒子 (SVRs)
self.init_particles()
print("Number of SVRs :", self.n_svrs)

def init_particles(self):
    """
    每个 SVR 就是一个粒子
    初始化所有粒子参数，局部最优和全局最优。
    对于 SVR 初始化，C 值从 C_min (包括) 到 C_max (不包括)，步长为 C_step，
    对于 gamma 初始化，从 eps_min (包括) 到 eps_max (不包括)，步长为 eps_step
    经过如上初始化，所有可能的 (C, gamma) 均已考虑
    """
    # 所有的 SVRs (particles).
    self.svrs = []
    # 每个 svr 的局部最优 score
    self.p_best_score = []
    # 每个 svr 获得局部最优时的参数值
    self.p_best_params = []

```

```

# 全局最优
self.g_best_score = 0
# 取得全局最优时的 (C, gamma) 参数值
self.g_best_params = {}

ctemp=[]
gammatep=[]

for C in np.arange(self.C_min, self.C_max, self.C_step):
    for gamma in np.arange(self.gamma_min, self.gamma_max,
self.gamma_step):
        ctemp.append(C)
        gammatep.append(gamma)
        svr = SVR(kernel = self.kernel,
                    C = C,
                    epsilon = self.epsilon,
                    gamma = gamma)

        self.svrs.append(svr)
        self.p_best_score.append(0)
        self.p_best_params.append({"C":0, "gamma": 0})

self.n_svrs = len(self.svrs)
c_pair_plot.append(ctemp)
gamma_pair_plot.append(gammatep)

def get_train_val_data(self, validation_step):
    """
    1. 将 data 划分为训练集与测试集
    2. 将 x_train 标准化为标准正态分布
    """

    if self.validation_type == "random-split":
        x_train, x_test, y_train, y_test = train_test_split(
            self.X, self.Y, test_size = self.validation_size)
    elif self.validation_type == "k-fold":
        # n_validations 就是 k.
        k = self.n_validations
        n_samples = self.Y.shape[0]
        subset_size = np.ceil(n_samples/k)

        #选取第几个 subset 作为验证集
        idx_min = int((validation_step - 1) * subset_size)

```

```

        idx_max = validation_step * subset_size
        idx_max = int(min(idx_max, n_samples))

        # 验证集
        x_test = self.X[idx_min:idx_max]
        y_test = self.Y[idx_min:idx_max]
        # 测试集
        x_train = self.X[np.r_[0:idx_min, idx_max:n_samples]]
        y_train = self.Y[np.r_[0:idx_min, idx_max:n_samples]]

    return x_train, x_test, y_train, y_test

def run_optimizer(self):
    """
    优化器
    """
    start = time.time()

    for i in range(self.n_iterations):
        print("\nOptimizer step: ", i+1)

        # 训练 SVR, 获得验证分数
        self.train_svrs()
        #根据验证分数, 更新粒子(C and gamma)的状态
        self.update_particle_state()
        print ('training takes {:.2f}
minutes'.format((time.time()-start)/60))
        print("Step best score: ", max(self.val_score))
        print("Global best score: ", self.g_best_score)

def train_svrs(self):
    """
    函数用于训练所有的 SVR, 并获得验证分数
    """
    # 存储每个 SVR 的验证分数的列表的列表。
    # 每个子列表对应一个 svr, 子列表中的每个元素将是对应于每一个验证步骤的验证分
    数。

    val_score_list = [[] for i in range(self.n_svrs)]

    #多次训练
    for val_step in range(self.n_validations):

```



```

        # 获取训练数据以及得分
        print ("val_step + 1:", val_step + 1)
        x_train, x_test, y_train, y_test = self.get_train_val_data(
            validation_step = val_step + 1)

        # 训练每一个 svr
        for j in range(self.n_svrs):
            self.svrs[j].fit(x_train, y_train.ravel())

        # 获取评分.
        val_score = self.svrs[j].score(x_test, y_test)
        val_score_list[j].append(val_score)

    # 将最终验证分数设置为每步验证分数的平均值。
    # 每一行都是一个 svr 的训练多次后得到的平均验证分数
    self.val_score = np.mean(val_score_list, axis = 1)

def update_particle_state(self):
    """
    更新每个粒子的局部和全局最佳值，然后更新每个粒子的参数
    """

    for i in range(self.n_svrs):
        # 更新 p_best
        if self.val_score[i] > self.p_best_score[i]:
            params = self.svrs[i].get_params()
            self.p_best_score[i] = self.val_score[i]
            self.p_best_params[i] = {'C': params['C'],
                                     'gamma': params['gamma']}

        # 更新 g_best
        if self.p_best_score[i] > self.g_best_score:
            self.g_best_score = self.p_best_score[i]
            self.g_best_params = params

    ctemp=[]
    gammatemp=[]
    # 更新 SVRs 的参数值
    for i in range(self.n_svrs):
        params = self.svrs[i].get_params()

        r1 = np.random.random()
        r2 = np.random.random()

```

```

        # 更新 C 的值
        C = params['C']
        C_new = (self.inertia_wt * C \
                  + r1*self.c1*(self.p_best_params[i]['C'] - C) \
                  + r2*self.c2*(self.g_best_params['C'] - C))
        C_new = max(0.01, C_new)

        # Find new gamma value.
        gamma = params["gamma"]
        gamma_new = (self.inertia_wt * gamma \
                     + r1*self.c1*(self.p_best_params[i]["gamma"] - gamma) \
                     + r2*self.c2*(self.g_best_params["gamma"] - gamma))
        gamma_new = max(0.001, gamma_new)
        ctemp.append(C_new)
        gammatemp.append(gamma_new)
        # 更新 SVR 的参数.
        self.svrs[i].set_params(**{'C': C_new, 'gamma': gamma_new})

    c_pair_plot.append(ctemp)
    gamma_pair_plot.append(gammatemp)

def get_best_values(self):
    """
    返回获得的全局最优和获得全局最优的 SVR 的参数
    """
    return self.g_best_score, self.g_best_params

def train_svr(config_file, svr_params, type):
    """
    用最终获取的参数训练 svr, 并保存 svr 模型
    """
    # 加载数据.
    config = configparser.ConfigParser()
    config.read(config_file)
    data_config = config["Data"]
    X, Y, _, scaler = get_SVR_input('train_up', type)

    # # 新建 pipeline (集合 fit transform train) .
    # # x_scaler 为标准正态分布应用器

```

```
# x_scaler = StandardScaler()
# # x_scaler = MinMaxScaler()
# 设定好参数的 svr
svr = SVR(**svr_params)

pipeline = Pipeline(steps = [('SVR', svr)])
pipeline.fit(X,Y)

# Save the pipeline.
if type=='speed':
    save_path = config["Model"]["svr_s_save_path"]
    scaler_path = config["Model"]["scaler_s_save_path"]
    with open(save_path, 'wb') as save_file:
        pickle.dump(pipeline, save_file)
    with open(scaler_path, 'wb') as scaler_file:
        pickle.dump(scaler, scaler_file)

if type=='flow':
    save_path = config["Model"]["svr_f_save_path"]
    scaler_path = config["Model"]["scaler_f_save_path"]
    with open(save_path, 'wb') as save_file:
        pickle.dump(pipeline, save_file)
    with open(scaler_path, 'wb') as scaler_file:
        pickle.dump(scaler, scaler_file)

if __name__ == "__main__":

    config_file = sys.argv[1]

    # Use PSO to find best hyper parameters for SVR.
    train_type = 'flow'
    print("Running PSO")
    pso_optimizer = PSO_SVR(type = train_type, config_file = config_file)

    start = time.time()
    pso_optimizer.run_optimizer()

    print ('The whole training takes {:.2f}
minutes'.format((time.time()-start)/60))

    best_score, best_params = pso_optimizer.get_best_values()
    print("\nBest score: ", best_score)
    print("Params of best SVR : ", best_params)
```

```
with open(train_type + '_best_params.json','w') as f:
    json.dump(best_params,f)

# Use the best parameters to train an SVR using the whole data.
print("\nTraining final model")
train_svr(config_file, best_params,type=train_type)

with open('json/c_' + train_type + '.json','w') as f:
    json.dump(c_pair_plot,f)
with open('json/gamma_'+ train_type + '.json','w') as f:
    json.dump(gamma_pair_plot,f)
```