中国研究生创新实践系列大赛

“华为杯”第十八届中国研究生数学建模竞赛

学　　校	华东师范大学
参赛队号	21102690002
队员姓名	1.王君悦
	2.黄家祺
	3.王昭芸

中国研究生创新实践系列大赛
“华为杯”第十八届中国研究生数学建模竞赛

题　目　　空气质量预报二次建模

摘　　要：

大气污染对生态环境及人体健康有严重影响，因此准确预估未来大气污染物浓度具有十分重要的科学意义，预测未来污染物浓度情况有利于提前采取污染防治措施。目前常用的污染物预报模型 WRF-CMAQ 还存在一定缺陷，如受气象场模拟结果的制约等问题。为了提高现有预报模拟的准确性，本文首先根据六种常见大气污染物的实际观测浓度，计算空气质量指数及相应的首要污染物；然后针对不同的气象条件，分别讨论不同情况下对污染物浓度和空气质量指数的影响情况，进行气象条件的合理分类。此外，本文以 WRF-CMAQ 模型模拟结果作为一次预报数据，结合实际观测数据，建立二次预报模型，预报未来三天三个监测站点 A、B、C 的常规污染物浓度、AQI 以及首要污染物；最后，考虑相邻区域污染物浓度的相关性，建立区域协同预报，结合反距离权重法建立 LSTM 模型，预测未来三天的污染物浓度、空气质量指数以及首要污染物。

针对问题一：空气质量指数常用于描述空气污染程度，通过六种常规大气污染物建立空气质量指数计算模型，求得给点监测点 A 未来四天的每日实测 AQI 和首要污染物。结果显示，这四天每日实测 AQI 先减小后增大，在 8 月 28 日达到 138，除 8 月 26 日外，其余三日的首要污染物均为 O3。

针对问题二：通过阅读相关参考文献和相关分析，发现空气质量指数及各污染物浓度主要受温度、湿度、压强和风速这些气象要素的影响。利用多元线性回归模型分别建立 AQI 与各气象条件、各污染物与各气象条件之间的定量表达式。除 PM2.5 的拟合优度较小外，其他模型具有较好的解释能力。根据相关分析以及多元线性归回模型，将气象条件分为了高影响类气象条件及低影响类气象条件，湿度与风速属于高影响类气象条件，温度与压强属于低影响类气象条件。

针对问题三：WRF-CMAQ 模型输出的污染物浓度与真实值偏差较大，利用实际观测数据对一次预报模拟数据进行修正。本文将一次模拟的气象要素以及对应时间的实际观测污染物浓度数据进行数据预处理、归一化等操作，利用长短期记忆人工神经网络 LSTM 对时间序列数据集的高可预报性，建立污染物浓度的二次预报模型，结果表明该模型模拟效果良好。对三个监测站点的日 AQI 预测结果在 40-100 之间，首要污染物主要为 O3。

针对问题四：相邻区域污染物浓度具有一定的相关性，基于地理学第一定律，本文根据各监测点的地理位置进行反距离权重插值，结合问题三中的 LSTM 模型建立区域协同预报模型。结果表明模拟精度有所提升，并解决了部分在利用 lSTM 模型模拟问题三数据时出现的极大极小值问题。

关键词：空气质量指数，多元线性回归模型，深度学习模型，LSTM，反距离权重插值

目录

1 问题重述

1.1 问题背景

随着经济社会的发展，科技进步的同时也带来了一系列大气污染问题。大气污染是指由于人类活动或自然过程引起某些物质进入大气中，呈现足够的浓度，达到了足够的时间，并因此危害了人体的舒适、健康和福利或危害了生态环境[1]。根据《环境空气质量标准》（GB3095-2012），用于衡量空气质量的常规大气污染物共有六种，分别为二氧化硫（SO2）、二氧化氮（NO2）、粒径小于 10μm的颗粒物（PM10）、粒径小于 2.5μm的颗粒物（PM2.5）、臭氧（O3）、一氧化碳（CO）[2]。防治大气污染需要每个人参与，共同努力。根据污染防治的实践经验，建立空气质量预报模型是改善空气质量的有效方法之一。通过空气质量预报模型，可以预先知道可能发生的大气污染过程，根据预报的大气污染过程制定相应的污染控制方案，采取有关措施，从而减少人体健康和环境遭到大气污染的影响。因此建立准确度高的空气质量预报模型对控制大气污染，提高环境空气质量具有重要意义。

目前在各国实际应用的空气质量预报模式包括 NAQPMS 模式、WRF-CMAQ、WRF-Chem、GATOR-GCMOM 模式等，包括第三第四代的空气质量数值预报模式，较之前两代预报模式，模式中的物理化学过程与机制更为复杂。在众多空气质量预报模型中， WRF-CMAQ 模型是一种很常用的模型。其中 WRF 模型是中尺度数值天气预报系统，CMAQ（Community Mutiscale Air Quality Model）是美国国家环境保护局开发的第三代空气质量模型，与传统模式不同，CMAQ 并非对单一物种进行模拟，而是将复杂的污染问题，例如不同污染物的复合污染考虑在内，进行综合处理。通过 WRF 提供的气象数据和场域内污染排放清单模拟污染物的变化过程，从而得到各时间的预报结果[3]。但是模拟的气象场具有不确定性，排放清单的分配也无法达到非常精细且与实际排放情况完全相符，也存在一定的不确定性，并且，WRF-CMAQ 模型对一些具有复杂化学反应机制的污染物的生成机理不太清晰，都容易导致预报结果的不准确。我国六大常规污染物中的臭氧，就是一种化学机制暂时还未揭示明确的污染物。国内外学界目前能达到的共识是臭氧对其前体物 NOx 和 VOCs 具有非线性的响应关系。但是这种非线性关系暂时难以用具体的公式阐明。因此 WRF-CMAQ 模式对臭氧浓度的模拟有时就会出现结果不准确的情况，而近年来以跃居我国许多重要城市的首要污染物的臭氧防治问题迫在眉睫，提高臭氧预报精度对我国空气污染防治十分重要。

为了提高 WRF-CMAQ 模型的预报准确性，本文提出二次建模的概念。结合实际观测数据与 WRF-CMAQ 模型的预报数据进行建模。实际气象条件对空气污染物浓度有很大影响，气象条件模拟的误差也会导致模型预报污染浓度的误差；而实际的污染物浓度结合实测气象数据可以对模型的一次预报数据进行订正，建立订正模型，提高预报精度。因此，利用实测数据对模型预报数据结果进行二次建模，达到优化预报模型，提高预报准确性的效果。

1.2 问题重述

综合以上背景，为了提高空气质量预报的准确性，建立二次模型以对给定监测点未来

的空气质量情况进行预测，本文主要回答以下问题：

（1）给定监测点 A，根据该点逐日污染物浓度实测数据（监测点 A 空气质量预报基础数据.xlsx），结合《环境空气质量指数（AQI）技术规定（试行）》（HJ633-2012）中 AQI 的定义，根据已给出的臭氧每日最大 8 小时滑动平均监测浓度和其他污染物每日 24 小时平均监测浓度分别计算 IAQI 指数，再根据 IAQI 指数计算每日 AQI 值。依据格式要求整理 2020 年 8 月 25 日到 8 月 28 日监测点 A 的每日实测 AQI 和首要污染物；

（2）根据题目所提供的监测点 A 的污染物浓度与气象数据（监测点 A 空气质量预报基础数据.xlsx），建立模型，分析污染物浓度与气象要素之间的相关关系，并依据相关分析结果对气象条件进行分类，阐明各类气象条件对污染物浓度的影响程度。问题关键在于判断气象条件会对污染物浓度（AQI）造成什么影响，分别使 AQI 上升时和使 AQI 下降的两类气象条件，再阐明影响程度；

（3）假设监测点之间不存在相互影响，根据题目所提供的三个监测点 A、B、C 的一次预报数据和实际观测数据，建立同时适用于三者的二次预报数据模型。该模型的关键在于，在对未来常规污染物浓度预测的同时，还需要保证 AQI 的预报精度要尽可能的高，首要污染物的精度尽可能的高。根据上述要求建立二次预报数学模型，然后对监测点 A、B、C 在未来三天，即 2021 年 7 月 13 日至 7 月 15 日的 6 种污染物日浓度进行预测，同时还需要计算各站点这三天对于的 AQI 以及首要污染物；

（4）考虑邻近区域内污染物浓度的相互影响，建立区域协同预报模型。根据题目所给的相邻区域内的四个监测点 A、A1、A2、A3，建立同时包含这四个点的二次预报模型。预报未来三天（2021 年 7 月 13 日至 7 月 15 日）各站点的污染物浓度、AQI 结果以及首要污染物，通过与问题 3 中模型对比，研究分析区域协同预报模型在污染物浓度预报上是否更加精准。

2　模型假设

（1）假设各监测站点位于同一海拔；
（2）假设各监测站点位于平原；
（3）假设各监测站点污染物浓度不受风向影响。

3　符号说明

符号名称	符号含义	符号说明
SO2	二氧化硫	
NO2	二氧化氮	
PM10	粒径小于 10μm的颗粒物	常规大气污染物
PM2.5	粒径小于 2.5μm的颗粒物	
O3	臭氧	
CO	一氧化碳	
AQI	空气质量指数	
IAQI	空气质量分指数	
T	温度	单位为℃
RH	湿度	单位为%
P	气压	单位为 MBar
V	风速	单位为 m/s
D	风向	单位为°
LSTM	长短期记忆	深度学习模型
RNN	循环神经网络	深度学习模型

4　问题一的建模与求解

4.1　问题分析

本题需要我们计算监测点 A 从 2020 年 8 月 25 日到 8 月 28 日每日实测的 AQI 和首要污染物。监测点 A 逐日污染物浓度实测数据包括从 2019 年 4 月 16 日到 2021 年 7 月 12 日各污染物浓度的逐日监测浓度，共 819 条数据。由于本题只需求解四日的实测 AQI 和首要污染物，在实测数据中这四天不存在数据异常情况，因此不需要进行数据预处理操作。

针对附件 1 给出的主要污染物：二氧化硫（SO2）、二氧化氮（NO2）、粒径小于 10μm 的颗粒物（PM10）、粒径小于 2.5μm的颗粒物（PM2.5）、臭氧（O3）、一氧化碳（CO），有实测的逐小时浓度数据和实测的逐日浓度数据。根据 AQI 的计算方法，我们可以直接采用已经把逐小时浓度数据平均了的 SO2 的 24 小时平均浓度、NO2 的 24 小时平均浓度、CO 的 24 小时平均浓度、PM2.5 的 24 小时平均浓度、PM10 的 24 小时平均浓度和 O3 的最大 8 小时滑动平均浓度。利用以上数据计算各污染物的空气质量分指数（IAQI），再将 IAQI 的最大值作为当日的 AQI 值，且 IAQI 最大的污染物即为当日的首要污染物，若 AQI 小于 50，则空气质量为优，当日无首要污染物。依据上述思路建立模型进行求解。

4.2　模型建立与求解

根据《环境空气质量指数（AQI）技术规定（试行）》（HJ633-2012），空气质量指数（Air Quality Index, AQI）可以对空气质量等级进行划分。其具体的计算方法如下。

空气质量指数（AQI）由各污染物项目的空气质量分指数（individual air quality index, IAQI）计算得到：

$$AQI = \max\{IAQI_1, IAQI_2, IAQI_3, \dots, IAQI_n\} \qquad (4-1)$$

其中 n 为污染物项目数量，$IAQI_1, IAQI_2, IAQI_3, \dots, IAQI_n$为各污染物项目的空气质量分指数，AQI 即为各污染物项目的 IAQI 中的最大值。

根据本题要求，主要的污染物项目有六种，因此得到 AQI 的计算公式如下：

$$AQI = \max\{IAQI_{SO_2}, IAQI_{NO_2}, IAQI_{PM_{10}}, IAQI_{PM_{2.5}}, IAQI_{O_3}, IAQI_{CO}\} \qquad (4-2)$$

对于某一污染物项目 P，它的空气质量分指数（$IAQI_P$）按公式（3）计算：

$$IAQI_P = \frac{IAQI_{Hi} - IAQI_{Lo}}{BP_{Hi} - BP_{Lo}} \cdot (C_P - BP_{Lo}) + IAQI_{Lo} \qquad (4-3)$$

其中，$IAQI_P$即为污染物项目 P 的空气质量分指数；C_P为污染物项目 P 的质量浓度值；BP_{Hi}为表 1 与C_P相近的污染物浓度限值的高位值；BP_{Lo}为表 1 与C_P相近的污染物浓度限值的低位值；$IAQI_{Hi}$为表 1 中与BP_{Hi}对应的空气质量分指数；$IAQI_{Lo}$为表 1 中与BP_{Lo}对应的空气质量分指数。

表 1 空气质量分指数（IAQI）及对应的污染物项目浓度限值

序号	指数或污染物项目	空气质量分指数及对应污染物浓度限值								单位
0	空气质量分指数（IAQI）	0	50	100	150	200	300	400	500	-
1	一氧化碳（CO）24 小时平均	0	2	4	14	24	36	48	60	mg / m³
2	二氧化硫（SO2）24 小时平均	0	50	150	475	800	1600	2100	2620	
3	二氧化氮（NO2）24 小时平均	0	40	80	180	280	565	750	940	
4	臭氧（O3）最大 8 小时滑动平均	0	100	160	215	265	800	-	-	
5	粒径小于等于 10μm颗粒物（PM10）24 小时平均	0	50	150	250	350	420	500	600	μg / m³
6	粒径小于等于 2.5μm颗粒物（PM2.5）24 小时平均	0	35	75	115	150	250	350	500	

在计算空气质量分指数（IAQI）时，要对使用的数据进行异常值的剔除。例如当 PM2.5 浓度普遍小于 200μg／m3 时，出现一个大于 400 的值，则该值为异常值，需要将它去除在外不进行 IAQI 的计算。

使用空气质量分指数（IAQI）计算得到空气质量指数（AQI）后，再根据计算得到的空气质量指数（AQI）划分空气质量等级，具体的等级及对应空气质量指数（AQI）范围如下表 2 所示。

表 2 空气质量等级及对应空气质量指数（AQI）范围

空气质量等级	优	良	轻度污染	中度污染	重度污染	严重污染
空气质量指数（AQI）范围	[0,50]	[51,100]	[101,150]	[151,200]	[201,300]	[301,+∞)

由表 2 可知，当 AQI 小于或等于 50 时，空气质量等级为优，当天无首要污染物；而当 AQI 大于 50 时，IAQI 最大的污染物项目为首要污染物。IAQI 最大的污染物为首要污染物。若 IAQI 最大的污染物为两项或两项以上时，并列为首要污染物。IAQI 大于 100 的污染物为超标污染物。

4.3 结果分析

根据由上述 AQI 及首要污染物的定义，在 MATLAB 软件中建立计算模型。计算得到 2019 年 4 月 16 日到 2021 年 7 月 12 日监测点 A 每天实测的 AQI 和首要污染物。根据问题一要求，提取出 2020 年 8 月 25 日到 8 月 28 日的数据，结果如下表 3 所示：

表 3 AQI 计算结果表

监测日期	地点	AQI 计算	
		AQI	首要污染物
2020/8/25	监测点 A	60	O3
2020/8/26	监测点 A	46	无
2020/8/27	监测点 A	109	O3
2020/8/28	监测点 A	138	O3

根据计算结果可知，除 2020 年 8 月 26 日 AQI 小于 50，无首要污染物外，其余三天 AQI 均大于 50，且首要污染物均为 O3。其中 2020 年 8 月 25 日空气质量等级为良，2020 年 8 月 27 日空气质量等级为轻度污染，2020 年 8 月 28 日空气质量也为轻度污染。

5 问题二的建模与求解

5.1 问题分析

除受到污染物排放变化的影响，当地的气象条件也会导致 AQI 的变化。气象条件通过使污染物发生扩散或者沉降从而导致该地区 AQI 的下降，反之则会引起 AQI 的上升。

附件 1（监测点 A 空气质量预报基础数据.xlsx）提供了逐日污染物浓度实测数据、逐小时的五项气象指标的实测数据，结合问题一计算得到的逐日 AQI 实测数据，将时期统一为 2019 年 4 月 16 日到 2021 年 7 月 12 日，共 819 条数据。经过异常数据处理等工作后得到本题所需数据。

为确定不同气象条件对 AQI 的影响，首先建立相关分析模型检验各气象条件与实测 AQI 之间的关系，结合显著性分析情况，建立各气象条件与 AQI 之间的多元回归模型。同时，对六种主要污染物也进行上述操作，分析各气象条件与各污染物浓度的相关性，通过多元线性回归模型建立实测 AQI 与各气象条件之间的回归方程。结合以上两部分的模型结果，对气象条件进行分类，从而分析各类气象条件的特征。

5.2 模型建立与求解

5.2.1 气象条件相关分析模型

相关分析是研究两个变量之间线性相关程度，皮尔逊相关系数常用来衡量两个变量的相关情况。皮尔逊相关系数用 r 表示，具体的计算方法如下公式 4 所示：

$$r=\frac{\sum_{i=1}^{n}(X_i-\bar{X})(Y_i-\bar{Y})}{\sqrt{\sum_{i=1}^{n}(X_i-\bar{X})^2}\sqrt{\sum_{i=1}^{n}(Y_i-\bar{Y})^2}} \qquad (5-1)$$

上式中 X、Y 为两个变量；n 为样本数量；X_i、Y_i为变量 X、Y 对应 i 时的观测值；$\bar{X}$、$\bar{Y}$分别为 X、Y 的样本平均值。计算得到的相关系数 r 值位于-1 到 1 之间。当 r 大于 0 时，两个变量呈正相关；r 小于 0 时，两个变量呈负相关；r 为 1 时表明两个变量之间呈完全正

相关关系，r 为-1 时呈完全负相关关系，r 为 0 则表示两个变量线性无关。

根据附件 1 中相关数据可知，气象条件有温度（T）、湿度（RH）、气压（P）、风速（V）、风向（D）。由于附件 1 数据所提供监测点 A 气象实测数据为逐小时数据，时间为 2019 年 4 月 16 日至 2021 年 7 月 13 日。我们将一日内 24 小时的数据平均得到监测点 A 气象实测数据的逐日结果。考虑到风向 24 小时平均无明显物理意义，结合其他相关文献对气象条件的选取情况，我们只选取了温度（T）、湿度（RH）、气压（P）、风速（V）四个气象要素作为本题的分析条件。

实测气象数据在 2021 年 7 月 23 日的数据截止于上午 7 时，且问题一计算得到的 AQI 逐日实测数据时期为 2019 年 4 月 16 日至 2021 年 7 月 12 日。为了将数据日期匹配，我们不考虑监测点 A2021 年 7 月 13 日的实测气象数据。只保留与 AQI 对应的时期，即 2019 年 4 月 16 日至 2021 年 7 月 12 日。同时由于缺测等情况引起的异常数据直接剔除，保留气象实测数据和 AQI 实测数据完整的数据项。

在进行相关分析前，首先将整理好后的逐日实测 AQI 数据与逐日实测气象数据绘制散点图。各数据散点图的分布情况如图 1 所示。图 1 中对角线代表对应行或列 AQI 及各个气象要素，如对角线第一个为 AQI，表示第一行或第一列为 AQI 与其他要素之间的关系。图中对角线下半部分为 AQI 及各个气象要素之间的散点图。根据散点图可以看出大部分 AQI 与各气象要素之间基本存在线性关系。利用公式 5-1 计算得到的相关系数如图中对角线上半部分所示。其结果为所在行列元素之间的相关系数。

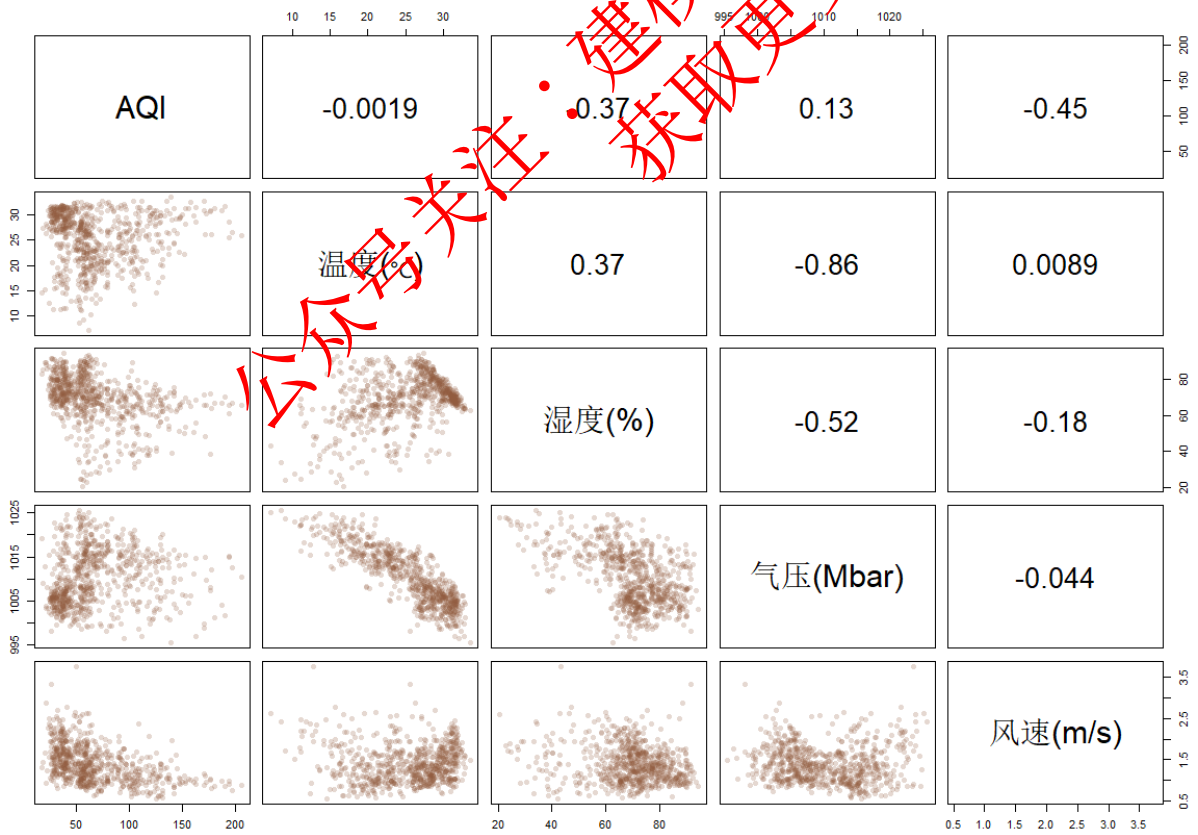

图 1 AQI 及各个气象要素散点图

第二行第一列图为 AQI 与温度对应的散点图，由图可知两者几乎无相关性，相关系数只有-0.0019，且没有通过显著性检验；第三行第一列图为 AQI 与湿度对应的散点图，可以看出当湿度增加时，AQI 有减小的趋势，两者呈中相关性；第四行第一列图为 AQI 与气压

对应的散点图，AQI 与气压为弱相关性，但是正相关，即 AQI 有随气压增大而增大的趋势。第五行第一列图为 AQI 与风速对应的散点图，AQI 与风速的相关性更明显，相关系数为-0.46，呈负的中相关性，风速越大，AQI 越小。

在四种气象影响因素中，湿度与风速对 AQI 的影响较为明显，呈负的中相关性。这也与通常认知相符。当风速较大时，空气流动使污染物更容易扩散，污染物浓度降低，AQI 也会相应减小。同样，湿度较大时，对污染物的扩散有增强作用，因而对 AQI 会有降低的作用，AQI 与空气湿度呈负相关性。气压可能受到其他气候因素的影响，可以从上表中看到，气压与温度和湿度都存在明显的相关性。因此气压可能会受到冷空气过境时风速大的影响，间接影响污染物的浓度；也会由于气压高导致高压中心风速小以及下沉气流等因素，不利于污染物的扩散，只能累积，从而导致 AQI 值增大[4]。在多种气象条件下，气压本身与 AQI 之间的相关性较弱。而温度在上图中显示的与 AQI 之间的相关关系接近于 0，表面上看是与 AQI 之间不存在相关关系的，但是由于每天的首要污染物不同，不同污染物受到温度的影响可能也是不同的，如果今天的污染物浓度随温度升高而降低，而昨天的首要污染物浓度随温度降低而降低，这样在整体相关性上就可能显示不出来明显的相关关系，需要进一步分析。

接下来我们进一步分析了温度、湿度、压强和风速与各物种浓度的相关关系（见图 2）。图 2 的横坐标从左到右分别为日平均温度、日平均相对湿度、日平均气压和日平均风速（由于风向的数字代表的是角度不同，数字的增长并非上升下降的物理意义，因此不纳入相互关系的考虑）。图 2 的纵坐标从上到下分别为 SO2 的 24 小时平均浓度、NO2 的 24 小时平均浓度、PM10 的 24 小时平均浓度、PM2.5 的 24 小时平均浓度、O3 的最大 8 小时滑动平均浓度和 CO 的 24 小时平均浓度。更细化地分析了各个气象条件对不同污染物浓度变化的影响。可以看到，这些散点图上都显示了一定的线性变化趋势。

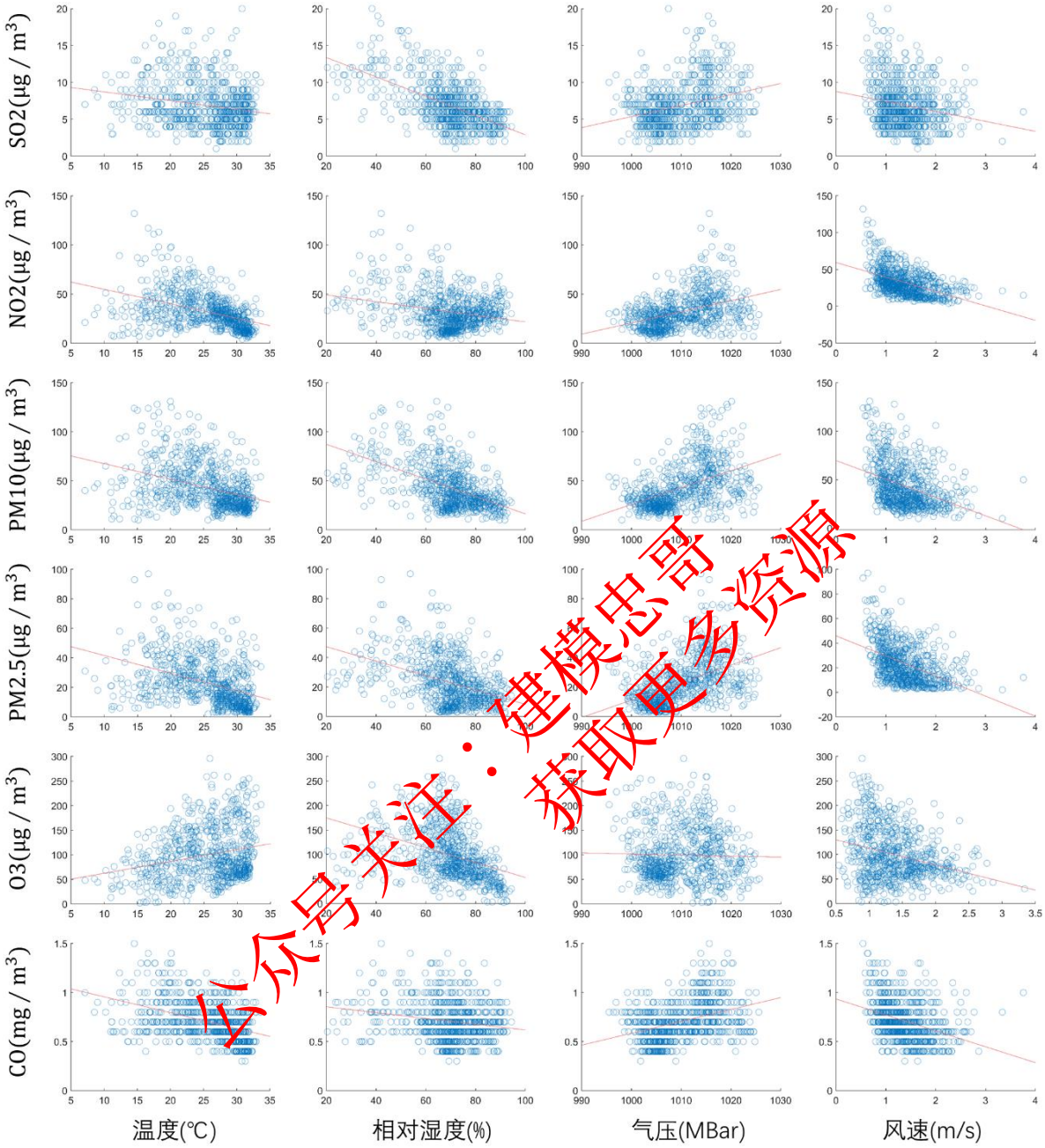

图 2 气象参数-污染物浓度散点图及趋势线

对于 SO2 来说，浓度有随着温度、相对湿度和风速下降而升高的趋势，而随着压强的增长，SO2 也有增长的趋势。相对来说 SO2 浓度随气压的变化而变化的幅度较大。当 SO2 浓度上升时，温度湿度都较低、风速也不高；当 SO2 浓度降低时，温度湿度都较高，风速偏高。对于 NO2 而言，浓度有随温度、湿度和风速下降而升高的趋势，且随着压强的增长而增长。根据拟合直线，相对而言浓度随相对湿度变化而变化的幅度小。PM10 和 PM2.5 都是空气中的细颗粒物，它们对应气象条件变化而变化的关系也较为一致。浓度随温度湿度风速上升而下降，并随压强增大二增大，浓度对各气象要素的响应程度较为一致。臭氧浓度与气象条件的关系稍有不同，它随湿度和风速下降而升高，但温度越高，臭氧浓度也越高，与气压之间的线性关系并不明显，略有负相关关系。而 CO 的变化关系与前四种污染物较为类似，有随温度、相对湿度和风速增大而减小的趋势，也有着随气压上升而上升

的趋势。相对来说，CO 浓度对相对湿度变化而变化的幅度最小。

可以看到温度升高的时候，SO2、NO2、PM10、PM2.5 和 CO 浓度都是有降低趋势的。气温升高可以降低污染物浓度的原因可能在于当温度较高时，大气层结稳定度会受到影响[5]，大气层结不稳定了，空气中的污染物就容易向上运动，近地面污染物浓度就降低了。温度较低时，大气污染物向上运动的倾向就小，污染物浓度就也较低。而臭氧浓度随温度上升而上升，这是由臭氧的化学机制所决定的，首先高温下阳光可能较好，生成臭氧的光化学反应容易发生，其次温度越高，高温催化反应越容易发生，即同等条件下臭氧容易在高温条件下生成，因此温度越高，臭氧浓度越高。在开放环境下，气压与气温有明显的负相关关系，气温越高，气压越低，气压的增大不利于污染物的扩散，因而污染物浓度与气压呈现了与气温相反的关系。而空气湿度与污染物浓度都呈负相关关系，这一关系可能是由于高湿度有利于污染物的扩散[6]，不同污染物在相同空气湿度下的污染系数有所不同。增大风速对污染物浓度也有降低作用，风速增大，有利于大气扩散，大气扩散时，污染物浓度就会下降。因此可以得出规律，当首要污染物不是臭氧时，污染物浓度上升时的天气条件是：气温下降，湿度下降，气压上升和风速降低；污染物浓度下降时的天气条件是：气温升高，湿度上升，气压下降和风速增大。当首要污染物是臭氧时，污染物浓度上升时的天气条件是：气温上升，湿度下降和风速降低；污染物浓度下降时的天气条件是：气温下降，湿度上升和风速增大。

由于根据图 2 可以了解到污染物与气象条件这两是一定线性关系的，因此可以使用皮尔逊相关系数来表征这些污染物和气象条件的相关关系。为了更加定量地了解污染物浓度和气象条件之间的相关关系，我们计算得到 6 种重要污染物（SO2、NO2、PM10、PM2.5、O3、CO）和四种气象条件（温度、湿度、气压、风速）间的相关系数。通过皮尔逊相关系数的计算公式计算，计算结果均通过了显著性检验，计算结果见表 4。

表 4 6 种污染物和 4 种气象条件间相关系数

	温度（℃）	相对湿度（%）	气压（MBar）	风速（m/s）
SO2（μg / m³）	-0.2086	-0.582	0.3127	-0.2014
NO2（μg / m³）	-0.4329	-0.2549	0.3974	-0.4873
PM10（μg / m³）	-0.3662	-0.5244	0.4759	-0.3762
PM2.5（μg / m³）	-0.4004	-0.4132	0.4723	-0.4636
O3（μg / m³）	0.2406	-0.3851	-0.0249	-0.2862
CO（mg / m³）	-0.4209	-0.1947	0.3763	-0.3594

根据表 4 可知，相对湿度与 SO2 浓度最具有相关性，相关系数高达-0.582；其他气象条件的相关性从高到低是气压、温度、风速，温度、风速、湿度相关系数为负，而气压与 SO2 呈正相关，相关性最低的温度与风速相关系数都约等于-0.2。风速和温度对 NO2 的影响显然要高于对 SO2 的影响，可能的原因是 NO2 更容易扩散；这两者之外的负相关项相对湿度对 NO2 的相关系数约为-0.25；气压与 NO2 的正相关性也较高，约为 0.4。各个气象条件对 PM10 和 PM2.5 这两种细颗粒污染物的影响比较一致，各项的相关系数都不小，但与 PM10 最相关的是相对湿度，相关系数高达-0.5244；与 PM2.5 最相关的气象条件是唯一呈正相关的气压，但是相关系数略小于 PM10 与气压的相关系数；温度和风速与细颗粒物污染物的相关系数都在-0.4 左右。臭氧与温度呈正相关关系，但相关性并不是很高，相关系数为 0.24；与 O3 相关性最强的气象条件是相对湿度，相关系数达到约-0.39；O3 浓度与气压几乎没有相关关系，相关系数约为 0；与风速的相关系数约为-0.29。CO 与空气湿度的

相关性是这 6 种污染物中最低的，负相关系数不到-0.2；和气温的负相关关系最强，相关系数-0.42；气压与风速和 CO 的相关性差的不多，但气压为正相关，风速负相关，相关系数分别约为 0.38 和-0.36。

5.2.2 气象条件多元回归模型

为了进一步研究各气象条件对 AQI 及污染物浓度的影响大小问题，需要建立 AQI（及 SO2、NO2、PM10、PM2.5、O3、CO）与各气象条件（温度、湿度、气压和风速）之间的数学模型。我们在忽略风向的的条件下（即假设风向对于各个污染物浓度的变化影响不大），利用监测点 A 的逐日实测气象数据（温度、湿度、气压、风速）与监测点 A 的逐日 AQI 数据与各个逐日污染物浓度数据，通过最小二乘法来建立线性方程，对温度、湿度、气压、风速当日 AQI 进行多元回归分析，定量地得到各个气象要素与当日空气质量指数间具体的变化关系。又由于每日的首要污染物并非固定的同一物种，也需要我们定量地了解气象条件和六种重要污染物间的具体影响关系。因此我们在得到 AQI 与气象条件的数学模型的基础上，计算六种污染物分别与四项气象条件的数学模型，用以进一步分析气象条件对不同大气污染物的影响关系和影响大小。

由于实测数据在客观上会发生缺测和误测的情况，在建模之前，首先需要对实测数据进行预处理。首先但对于实测数据在某一时次对某污染物发生缺测的情况，不应简单地将空缺值处理成上一时次值或下一时次值或者两者的平均，因为污染物浓度可能会发生较为突然敏感的变化，应当保留这些空值，在多元线性回归中并不要求数据的时间次序是连续的，因此在编程中直接将有空值的时次不加入分析即可。其次对于实测数据本身在时次上的缺失，同样也是因为多元线性回归不要求数据的连续性，在本问的解答中可以不需要对缺失的时次进行补齐。最后，实际测量的数据也有可能发生测量误差与计算误差，可能会有一些极端值远远偏离“大部队”，针对这些极端数据，应当予以剔除，避免因此可能产生的误差。另一方面的数据预处理是数据的归一化处理。对于作为自变量的气象要素值，由于各气象要素的量纲不同，为避免在拟合过程中由于不同变量间数量级相差过大而影响拟合的效果。我们需要对监测点 A 的实测气象数据，温度、湿度、气压和风速统一进行归一化，使得不同指标之间具有可比性。我们使用的线性函数归一化公式具体计算方法如下：

$$X_{nor} = \frac{X - X_{min}}{X_{max} - X_{min}} \qquad (5-2)$$

其中，X_{nor}为归一化后的结果，X为需要进行归一化处理的各类实测数据，X_{max}、X_{min}分别为各类数据的最大值和最小值。

整理上述预处理后的数据，利用最小二乘法（Ordinary Least Square, OLS）建立多元回归模型。最小二乘法是通过使误差平方和最小来拟合一个最优的函数与数据匹配。这里分别建立 AQI 与各类气象要素的多元线性回归方程、各个污染物与各类气象要素的多元线性回归模型：

$$Y_\alpha = x0_\alpha + x1_\alpha T + x2_\alpha RH + x3_\alpha P + x4_\alpha V + \varepsilon \qquad (5-3)$$

公式 6 中，α分别为 AQI、SO2、NO2、PM10、PM2.5、O3 以及 CO；Y_α为各要素对应的逐日数据；T、RH、P、V 分别代表气象条件：温度、湿度、气压以及风速；$x1_\alpha$、$x2_\alpha$、$x3_\alpha$、$x4_\alpha$为各气象条件对应的回归系数，$x0_\alpha$为常数项，ε为误差项。

利用 Python 建立各多元线性回归模型，得到各模型的拟合优度 R^2、F 统计量、P 值等参数。通过 R^2 可以来说明线性回归模型的拟合程度，其他参数也可以反映模型的质量。

5.3 结果分析

对实测逐日气象数据、AQI、各气象要素建立的多元回归方程模型结果如下表 5 所示。根据模型参数，如 P 值、标准估算误差等，各模型均通过了显著性检验。

表 5 各气象要素与 AQI 及污染物浓度的多元回归分析

	温度	湿度	气压	风速	常数	R^2
AQI	35.446	-103.767	-0.253	-122.754	149.021	0.445
SO2	-0.211	-11.046	-0.918	-6.067	16.662	0.434
NO2	-44.424	-26.884	-16.457	-61.306	107.227	0.468
PM10	-4.825	-64.076	15.699	-66.047	101.441	0.531
PM2.5	-9.504	-32.456	11.304	-51.246	61.814	0.245
O3	163.468	-167.843	46.935	-128.751	114.105	0.478
CO	-0.474	-0.1776	-0.126	-0.5124	1.3676	0.328

从上表中可以看出，AQI 对应的 R^2 值大约为 0.445，拟合效果较好，可以解释一部分气象要素变化与 AQI 的变化的关系，并且 P 值约等于 0，说明接受原假设的概率很低，推翻了原假设，有总体显著性差异，可以接受各气象要素与对应参数组成的回归曲线与 AQI 存在线性关系。

计算结果表明，压强对应的回归系数的 P 值大于 0.05，说明该值没有通过显著性检验，即气压对于空气质量指数的改变可能没有明显的线性关系，所以气压对于空气质量指数的变化没有明确的作用。

而剩下的温度、湿度、风速的都通过显著性检验，表明温度湿度和风速对 AQI 有明显的线性关系，对 AQI 的影响大小排序为湿度＞风速＞温度，其中，温度与 AQI 呈正相关，但影响较小；风速和湿度与 AQI 呈负相关关系，对 AQI 的影响也较大。

对污染物 SO2 的浓度而言，对应的 R^2 为 0.434，这些气象条件可以解释近一半的 SO2 浓度。各自变量项的 P 值都小于 0.01，结果显著，具有明显的线性关系。对比湿度与风速的回归系数，温度和压强的回归系数都很小，表明于 SO2 而言，湿度和风速的影响较大，且都呈负相关关系，湿度对 SO2 浓度的影响最大；而温度和压强对 SO2 的影响较小。

对污染物 NO2 来说，对应 R^2 为 0.468，气象条件对它的解释程度略高于对 AQI 值的解释程度。所有气象条件都通过了显著性检验，且所有气象条件都与 NO2 呈负相关关系，其中，各参数影响大小排序为风速＞湿度＞温度＞气压。

对污染物 PM10 而言，它的对应 R^2 为 0.53，有超过一半的污染物浓度可以通过这四个气象参数来解释，是这些组里解释程度最高的，整体显著性水平较高。所有气象参数中，温度没有通过显著性检验，在此不做讨论。气压、湿度与风速对 PM10 浓度的影响大小为风速＞湿度＞气压，风速与湿度的回归系数大致相同，属于对 PM10 影响较大的条件，且都为负影响；气压对 PM10 浓度影响较小，为正影响。

PM2.5 的 R^2 值只有 0.245，接近四分之一的该值可以被这些气象条件解释到，说明还有很多其他的因素会影响到 PM2.5 的变化。整体显著性水平高，但温度和气压的没有通过显著性检验，从回归系数上看也对 PM2.5 影响较小。湿度和气压对 PM2.5 的影响较大，且都是负影响，风速对 PM2.5 浓度的影响最大。

O3 的回归方程和其他物种较为不同。它的 R^2 为 0.478，解释程度大致处于平均水平

上。所有气象参数均通过了显著性检验。对 O3 的影响大小排列为湿度＞温度＞风速＞气压。湿度和温度的影响程度基本一致，但温度对 O3 的影响为明显的正影响，而湿度和风速是负影响。气压也呈正影响，但影响较小。

对 CO 而言，R^2 为 0.328，约三分之一的该浓度值可以被气象条件解释。气压项没有通过显著性检验。温度与风速对 CO 浓度的变化有较大影响，湿度对 CO 浓度也略有影响，都呈负相关关系。

综上所述，在上面各气象要素对于六种污染物的影响的统计下，发现：

无论是六种污染物中的哪一种，湿度与风速都对于污染物浓度变化有明显的负相关关系，并且在一些污染物中，只有这两个气象要素通过了显著性检验。说明湿度变化与风速变化对于污染物浓度变化影响极大，属于高影响类气象条件，即湿度与风速增大，污染物浓度极有可能下降，而对应两因素减少时，污染物浓度极有可能上升。

而对于气压和温度来说，在对于六种污染物的影响的统计下，观察到，两个气象要素各有三次在总体显著性较高的情况下，而该要素的显著性水平没有通过显著性检验，并且通过显著性检验的情况下，对于一些污染物的浓度的变化存在负影响，而另一些则为正影响，说明气压变化与温度变化对于污染物浓度变化影响较小，属于低影响类气象条件。

并且发现高影响类气象要素具有对所有污染物变化有一致的影响作用，而低影响类污染物对特定的污染物的浓度变化可能有不同的影响作用，可能为正也可能为负，并且有些情况可能都无法通过显著性检验，而无法判定对该污染物的浓度变化有影响。

6 问题三的建模与求解

6.1 问题分析

由于一次预报模型存在不确定性等问题，加入实测数据可以对一次预报模型进行修正从而建立二次预报模型，以提高预报精度。本题给出 A、B、C 三个监测点，在此不考虑三个监测点之间的相互影响，利用三个监测点的一次预报模型数据和实际观测数据，建立一个能同时满足三个站点的二次预报模型。通过该二次预报数学模型来预报未来三天（2021 年 7 月 13 日至 2021 年 7 月 15 日）各站点 6 种污染物的单日浓度值。根据各站点的一次预报的污染物浓度数据和气象数据和各站点的实测数据，我们建立二次预报数学模型，通过模型模拟未来三天各污染物的逐小时浓度，经过平均得到单日浓度值，再根据 AQI 计算公式计算得到监测点 A、B、C 未来三日的 AQI 及首要污染物。

需要注意的是，根据题目要求，模型结果应使单日 AQI 预报值尽可能小而首要污染物预测浓度尽可能准确。为此我们利用附件材料中提供的实测污染物浓度数据与一次预报的气象要素数据进行模拟，建立一种基于长短期记忆人工神经网络的预测模型（LSTM），预测未来逐小时的各站点 6 种污染物浓度值，再计算得到每日 AQI 及首要污染物。

6.2 模型建立与求解

6.2.1 LSTM 模型简介

为对空气质量进行二次建模，我们使用基于 LSTM（Long short-term memory）的递归神经网络模型。LSTM 神经网络最早由 Hochreiter 和 Schmidhuber 提出，能较好地发现长期依赖关系而被广泛用于处理序列信息。它是一种递归神经网络（Recurrent Neural Network，RNN）的一种，与传统的前馈神经网络不同，RNN 是基于时间序列的模型，能够建立先前信息和当前环境之间的时间相关性。而 LSTM 则解决了 RNN 可能会出现的梯度弥散和爆炸问题，也就是当网络层数增加的时候，会出现随着时间推移忘记前面信息的现象[7]。

基于以上的优势，LSTM 模型也被广泛用于股票，天气条件，污染物排放等预测情况，适合于处理和预测时间序列中间隔和延迟非常长的重要事件。本题题目要求建立一个二次预报模型，并且这个模型应该是建立于一次模型基础之上，而从本题所给出的附件中可以看出，各个气象站点都有大量的以时间为序的一次模型预测出的气象要素和污染物浓度，并且由于数据缺省等特殊情况，时间存在一定间隔，这些都恰好符合 LSTM 对于初始数据的需求。

LSTM 添加了一个专门用于保存历史信息的记忆单元，如图 3 所示。历史信息通过三个门：输入门（Input Gate）、遗忘门（Forget Gate）和输出门（Output Gate）的控制进行更新。图 3 中 Cell 有一参数为 state，该参数用来表示神经元的状态。输入门用于接受参数和接受修正参数。遗忘门根据上一次神经元的参数，对修正参数进行选择性遗忘。输出门用于输出参数或者输出修正参数。这对于传统的 RNN 模型中一些缺点，会有很大的弥补作用，将极大的提升模型的可复用性，以及推广价值。

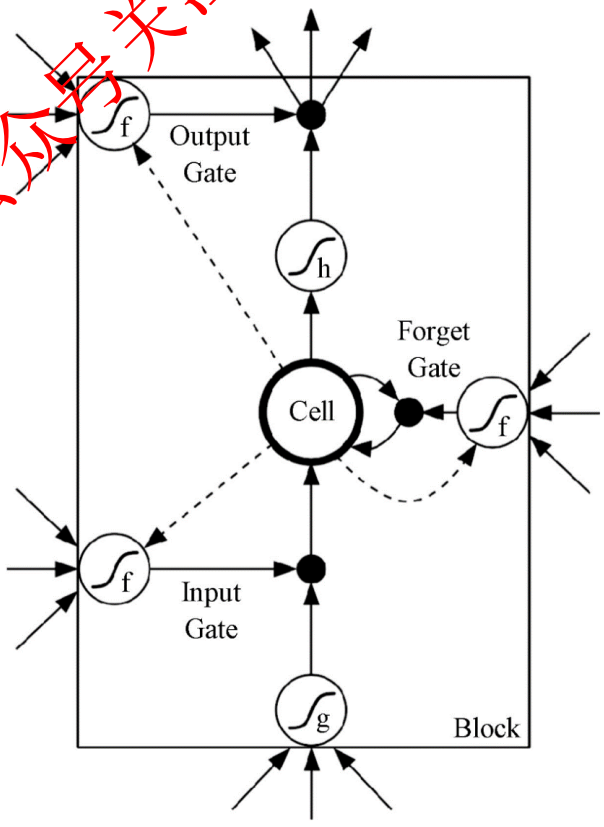

图 3 LSTM 神经元结构示意图[8]

遗忘门 ft、输入门 it、输入节点 gt、输出门 ot、本单元状态 st 和本单元输出 ht 计算公式为：

$$f_t = \sigma\left(W_{fx}x_t + W_{fh}h_{t-1} + b_f\right) \quad (6-1)$$

$$i_t = \sigma\left(W_{ix}x_t + W_{ih}h_{t-1} + b_i\right) \quad (6-2)$$

$$g_t = \sigma\left(W_{gx}x_t + W_{gh}h_{t-1} + b_g\right) \quad (6-3)$$

$$o_t = \sigma\left(W_{gx}x_t + W_{oh}h_{t-1} + b_o\right) \quad (6-4)$$

$$st = gt \cdot it + s_{t-1} \cdot ft \quad (6-5)$$

$$ht = \emptyset(st) \cdot ot \quad (6-6)$$

其中：$x_t \in R^k$表示时刻的向量的输入；W_{fx}、W_{fh}、W_{ix}、W_{ih}、W_{gx}、W_{gh}、W_{gx}、W_{oh}是权重矩阵；b_f、b_i、b_g、b_o是对应权重的偏向置。σ是 sigmoid 函数；是 tanh 函数；·表示将点积。ht 是 t 时刻以及之前时刻存储了所有有用信息的隐状态向量。

上述原理陈述论证了 LSTM 在本题中的可用性，以及对于结果的高可预测性，图 4 为二次预报深度学习模型的流程图，我们将按以下流程进行建模。

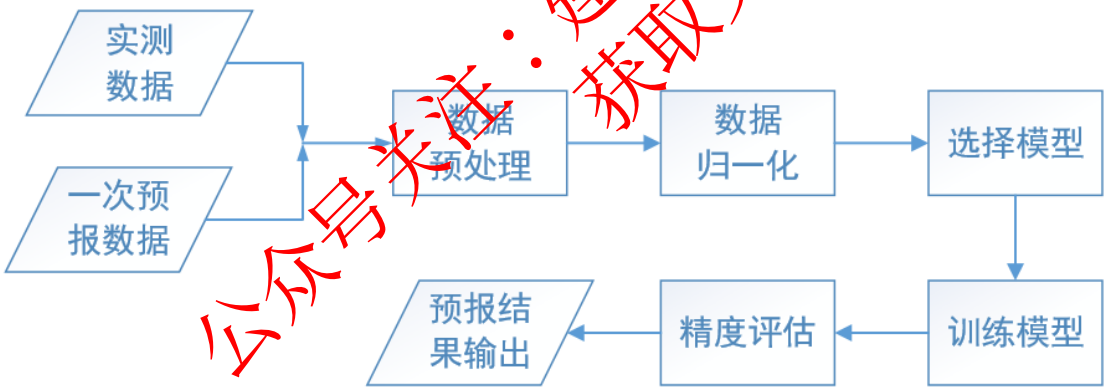

图 4 二次预报深度学习模型流程图

6.2.2 LSTM 模型建立与求解

首先是将本题附件中所给出的 A、B、C 三个监测站的一次预报数据中的逐小时精细化气象数据，与三个站点各自所对应的时间的实际观测污染物值进行组合。组合成为 A、B、C 三个站点，各自的一次模拟下的气象要素值与真实污染物的合成数据集，再按照当天日期预测当天数据，和当天预测第二日以及当天预测第三日，分成这三组数据集。根据所给数据，在对未来的预报数据中，只有 7 月 13 号可以预报 7 月 15 号的结果，因此我们统一选取当天预测第三日的数据集来完成后面的建模工作。

在上面工作基础上，由于一次模型中输出了 15 个气象要素值，这些要素值各自的取值

范围，以及单位有很大的差别，故为了后续建模的结果的准确性，进行归一化操作。并且由于 LSTM 模型对于时间序列的要求可以存在一定的间隔，故可以通过直接去除掉一些缺省值来加强模型的鲁棒性。

在对数据进行采集、预处理、归一化之后，将数据带入 LSTM 模型之中，进行训练。同时在此过程之中，逐渐调整模型的序列步长为 15 步，来保证原有时间序列对于数据所带来的稳定性，又不至于步数太长而导致产生数据异常。在经过大量样本训练，大约 8000 至 10000 次训练之后，二次预报模型的损失率可以达到稳定的 0.02 左右。将之前分好测试集带入模型之中，进行检验，发现模拟的效果良好，因此完全可以利用 LSTM 模型进行未来三天的预报检测。

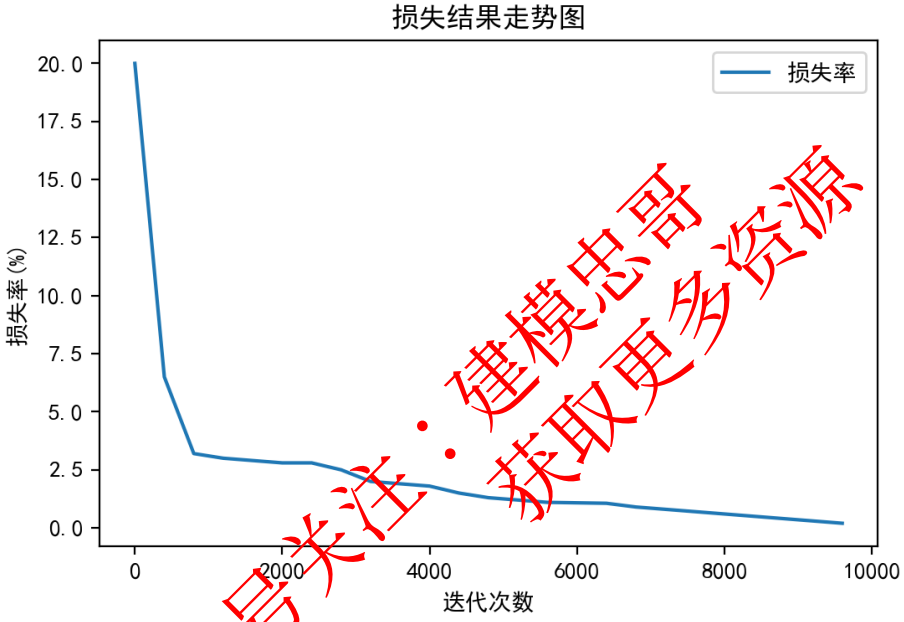

图 5 模型精度

6.3 结果分析

6.3.1 LSTM 模型精度分析

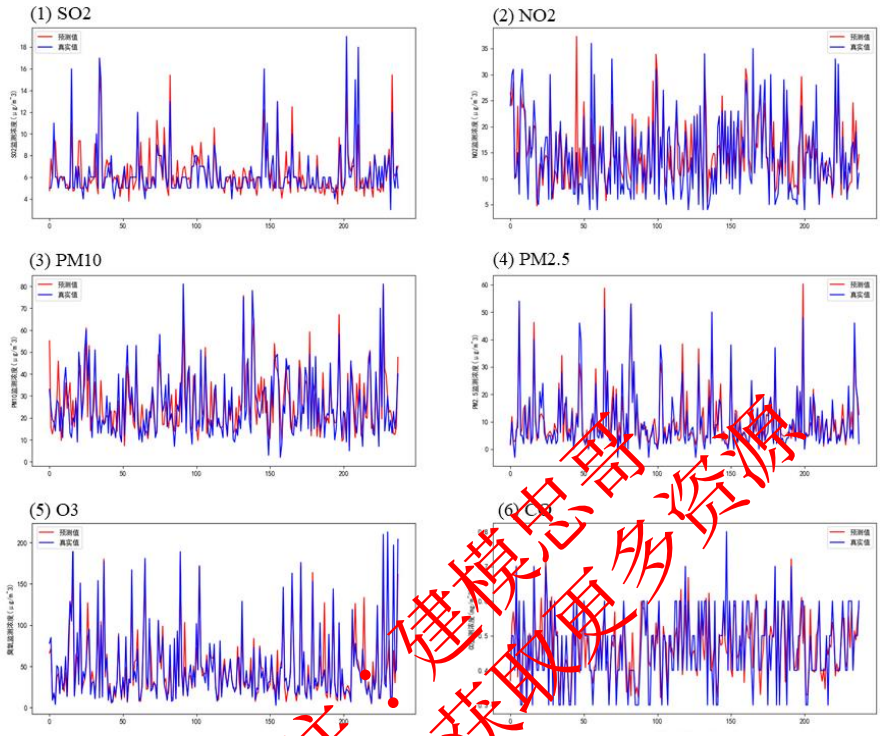

图 6 各污染物浓度实际值与 LSTM 估计值拟合曲线

随机选择 1%的数据进行测试，每种污染物对应的模型估计值与实际值对比情况如图 6 所示。由图 6（1）可知，对于 SO2 浓度的二次预估结果变化情况与实际观测浓度值大致相同。在 5-8μg/m³区域，LSTM 模型预估的 SO2 浓度值最接近实际值。而在 10μg/m³以上的区域，模型预估的结果相对较差。图 6（2）为 LSTM 模型对 NO2 浓度的预估结果与实际观测数据的拟合曲线图。对比图 6（1）可以发现，该模型对 NO2 的预估结果明显不如 SO2。在 NO2 浓度大于 20μg/m³的区域模型预估结果较好。LSTM 模型对 PM10、PM2.5、O3 的浓度的二次预报结果比较好，如图 6（3）、6（4）、6（5）所示。在测试集各污染物对应的浓度范围内，模型预报结果也基本与实际观测值像吻合。其中较难预报的臭氧浓度也相对较好。CO 浓度的模型预报结果在 0.4-0.6μg/m³范围内预估较好，其他区域存在一定的误差。

6.3.2 预报结果分析

根据上述模型，我们为未来三日监测站 A、B、C 各污染物浓度、AQI 及首要污染物的结果如下表所示：

表 6 A 站污染物浓度及 AQI 预测表

预报日期	地点	二次模型日值预测							
		SO_2 ($\mu g/m^3$)	NO_2 ($\mu g/m^3$)	PM_{10} ($\mu g/m^3$)	$PM_{2.5}$ ($\mu g/m^3$)	O_3 最大八小时滑动平均 ($\mu g/m^3$)	CO (mg/m^3)	AQI	首要污染物
2021/7/13	监测点 A	6	11	23	8	87	0.4	44	无
2021/7/14	监测点 A	6	18	24	8	124	0.4	70	O3
2021/7/15	监测点 A	6	15	19	7	99	0.4	50	O3

表 7 B 站污染物浓度及 AQI 预测表

预报日期	地点	二次模型日值预测							
		SO_2 ($\mu g/m^3$)	NO_2 ($\mu g/m^3$)	PM_{10} ($\mu g/m^3$)	$PM_{2.5}$ ($\mu g/m^3$)	O_3 最大八小时滑动平均 ($\mu g/m^3$)	CO (mg/m^3)	AQI	首要污染物
2021/7/13	监测点 B	5	11	15	5	89	0.3	45	无
2021/7/14	监测点 B	5	9	13	5	89	0.4	45	无
2021/7/15	监测点 B	6	10	16	5	70	0.4	35	无

表 8 C 站污染物浓度及 AQI 预测表

预报日期	地点	二次模型日值预测							
		SO_2 ($\mu g/m^3$)	NO_2 ($\mu g/m^3$)	PM_{10} ($\mu g/m^3$)	$PM_{2.5}$ ($\mu g/m^3$)	O_3 最大八小时滑动平均 ($\mu g/m^3$)	CO (mg/m^3)	AQI	首要污染物
2021/7/13	监测点 C	7	20	31	13	150	0.5	92	O3
2021/7/14	监测点 C	7	15	34	15	136	0.6	80	O3
2021/7/15	监测点 C	7	19	25	10	134	0.5	79	O3

根据以上三个表，对于 A 站未来三天的预估结果，预估的 SO2 浓度没有发生变化，均为 6μg/m³，NO2 在 7 月 14 日最大，PM10 的浓度基本在 20 左右，而 O3 变化比较大，其中在 7 月 14 日高达 124μg/m³，CO 的未来三天的预估结果也不变。根据各污染物浓度计算出来的 A 站 2021 年 7 月 13 日的 AQI 为 44，IAQI 最大的污染物为 O3，但因为 AQI 小于 50，空气质量为优，因为模拟结果当日无首要污染物；2021 年 7 月 14 日二次模拟 AQI 为 70，首要污染物为 O3，空气质量良；7 月 15 日二次模拟的 AQI 值为 50，首要污染物也是 O3，空气质量良。

对于 B 站各污染物在这三天的预估浓度变化不大。2021 年 7 月 13 日至 7 月 15 日的日 AQI 模拟结果分别为 45、45、35，空气质量均为优，因此都没有首要污染物，但 IAQI 最高的污染物仍是 O3。C 站未来三天的二次预报结果显示，首要污染物都是 O3，AQI 值分别为 92、80 和 79，都在 50 到 100 的区间内，空气质量均为良。

7　问题四的建模与求解

7.1　问题分析

在问题三中，我们没有考虑到 A、B、C 三个监测站点之间的相互影响。而在实际情况下，地理事物之间是存在空间关联性的，根据地理学第一定律可知，任何事物之间都是相互关联的，距离较近的事物之间的联系更紧密。因此本题中，在对空气质量预报进行二次建模时，还需要考虑各站点之间的相互影响，建立区域协同预报模型，可能会提高预报结果的准确性。

本题给出了监测点 A 附近的三个监测点 A1、A2、A3 的一次预报气象数据与污染物浓度数据，实测气象数据与污染物浓度数据。在此需考虑 A、A1、A2、A3 之间的相互影响，建立包含四个监测站点的协同预报模型，因此我们基于反距离权重法对各站点的预报数据重新补充，再利用预报数据和实测数据建立适用于该站点二次预报模型。

通过建立的模型，我们需要对四个监测点未来三天的常规污染物单日浓度进行预测，同时计算单日 AQI 以及首要污染物。将该模型得到的预报结果与问题 3 中的没有考虑相互影响的二次预报模型的预报结果进行对比，根据对比结果，研究分析协同预报模型是否能起到提高未来污染物浓度预报精度的作用。

7.2　模型建立与求解

A、A1、A2、A3 四个站点在邻近地区内，可以对这些站点进行区域协同预报，用三个站点的数据根据插值方法可以插值到第四个站点数据，用这样的插值法可以有效地避免某一站点数据的数据异常，用邻近区域中其他站的数据对区域中某一站点的数据进行修正。由于是根据地理位置进行的插值，不能仅使用简单的数学插值方法，需要使用有实际地理意义的插值。地理学中常用的插值方法有克里金插值法、反距离权重插值法等。我们使用反距离权重插值法对站点数据进行插值。

反距离权重插值法是基于相似相近的假设：彼此距离较近的事物要比彼此距离较远的事物更相近，两个物体离得越近，它们的性质就越相似，而两个物体离得越远，他们的性质相似性就越小。反距离权重插值法的原理是针对一个被插值点，给周围离散点赋予一个权重，按照这个权重将周围离散点的数据插入到这个插值点中。反距离权重插值法中的权重通常依赖于插值点和离散点距离幂次方的倒数，最常用的幂次方为 2 次方，即权重与距离的平方成反比。

某个离散点与被插值点间的距离d_i为：

$$d_i = \sqrt{(x - x_i)^2 + (y - y_i)^2} \tag{7-1}$$

其中，插值点的坐标为（x，y），离散点的坐标为（x_i，y_i）。

在本问中，我们对 A、A1、A2、A3 站点数据的权重考虑为，本站点自身占 50%，另外三个站点的占 50%。即当 A 站点为插值点时，A 站权重为 1，A1、A2、A3 权重相加等于 1。在这一情况下，权重计算公式为：

$$W_i = \frac{\frac{1}{d_i^2}}{\sum_{i=1}^{n}\frac{1}{d_i^2}} \qquad (7-2)$$

其中，W_i为离散站点对插值站点的权重，n 为离散站点的总数。

要预测 A、A1、A2、A3 四个站点的污染物数据，需要将四个站点分别作为插值点计算权重，计算结果如表 6。

表 9 站点权重表

站点权重 / 插值站点	A	A1	A2	A3
A	0.5	0.06	0.12	0.32
A1	0.12	0.5	0.18	0.2
A2	0.22	0.15	0.5	0.13
A3	0.34	0.09	0.07	0.5

根据反距离权重插值得到权重计算各站点加权后的新数据，作为输入层带入 LSTM 模型中进行二次预报模拟，LSTM 神经网络的具体配置问题三。

7.3 结果分析

根据上述模型进行区域协同预报，对监测站 A、A1、A2、A3 未来三日各污染物浓度、AQI 及首要污染物的结果如下表所示：

表 10 A 站污染物浓度及 AQI 预测表

预报日期	地点	二次模型日值预测							
		SO_2 (μg/m³)	NO_2 (μg/m³)	PM_{10} (μg/m³)	$PM_{2.5}$ (μg/m³)	O_3 最大八小时滑动平均 (μg/m³)	CO (mg/m³)	AQI	首要污染物
2021/7/13	监测点 A	6	11	25	7	98	0.4	49	无
2021/7/14	监测点 A	6	16	23	7	154	0.4	95	O3
2021/7/15	监测点 A	6	16	20	7	139	0.4	83	O3

表 11 A1 站污染物浓度及 AQI 预测表

预报日期	地点	二次模型日值预测							
		SO_2 ($\mu g/m^3$)	NO_2 ($\mu g/m^3$)	PM_{10} ($\mu g/m^3$)	$PM_{2.5}$ ($\mu g/m^3$)	O_3 最大八小时滑动平均 ($\mu g/m^3$)	CO (mg/m^3)	AQI	首要污染物
2021/7/13	监测点 A1	8	13	24	10	109	0.3	58	O3
2021/7/14	监测点 A1	8	19	30	8	160	0.4	100	O3
2021/7/15	监测点 A1	9	13	29	7	111	0.4	60	O3

表 12 A2 站污染物浓度及 AQI 预测表

预报日期	地点	二次模型日值预测							
		SO_2 ($\mu g/m^3$)	NO_2 ($\mu g/m^3$)	PM_{10} ($\mu g/m^3$)	$PM_{2.5}$ ($\mu g/m^3$)	O_3 最大八小时滑动平均 ($\mu g/m^3$)	CO (mg/m^3)	AQI	首要污染物
2021/7/13	监测点 A2	7	11	24	7	93	0.4	47	无
2021/7/14	监测点 A2	6	19	33	9	142	0.4	85	O3
2021/7/15	监测点 A2	6	16	16	7	137	0.4	81	O3

表 13 A3 站污染物浓度及 AQI 预测表

预报日期	地点	二次模型日值预测							
		SO_2 ($\mu g/m^3$)	NO_2 ($\mu g/m^3$)	PM_{10} ($\mu g/m^3$)	$PM_{2.5}$ ($\mu g/m^3$)	O_3 最大八小时滑动平均 ($\mu g/m^3$)	CO (mg/m^3)	AQI	首要污染物
2021/7/13	监测点 A3	6	11	25	8	97	0.4	49	无
2021/7/14	监测点 A3	6	8	33	7	139	0.4	83	O3
2021/7/15	监测点 A3	6	16	29	7	145	0.4	88	O3

表 10-表 13 是 A 站、A1 站、A2 站和 A3 站区域协同二次预报模型的 2021 年 7 月 13 日到 7 月 15 日的模拟结果。A 站 7 月 13 日模拟日 AQI 值为 49，O3 浓度占主导地位，但由于天气质量为优，当日无首要污染物；7 月 14 日模拟的臭氧最大 8 小时滑动平均较高，日 AQI 值达到 95，首要污染物是 O3，天气质量为良；7 月 15 日 A 站模拟 AQI 为 83，首要污染物是 O3；在首要污染物的预测上，本文与上一问的结果是一致的。A1、A2、A3 站的模拟结果和 A 站在量级没有很大区别，首要污染物也主要是 O3。A1 站的每日预测空气质量都是良，AQI 分别为 58、100 和 60，每天的首要污染物都是 O3；其中，7 月 14 日预测 AQI 为 100，即将到达轻度污染的程度，预报发生的是 O3 轻度污染。A2 站 7 月 13 日预测 AQI 为 47，无首要污染物；7 月 14 日和 7 月 15 日的预报 AQI 值为 85 和 81，首要污染物是 O3，空气质量良。A3 站的预报第一日 AQI 为 49，空气质量优，没有首要污染物；

7 月 14 日和 7 月 15 日的空气质量都预报为良，AQI 预测为 83 和 88，首要污染物是 O3。

8 模型评价与推广

针对于问题二，我们首先使用相关分析来检验了实测数据下的各个气象要素与各个污染物之间的关系，发现了很多有价值的内容，如湿度与风速与空气质量指数的关系较为明显，这成为了后面创建多元线性回归分析模型的基础。紧接着又依照合理的假设，在假设各个实测点的条件基本一致下，利用多元线性回归模型，建立温度，湿度，气压以及风速和各个污染物以及 AQI 之间的关系，在有限的数据下，模型很好的表现出了温度，气压对于污染物的浓度影响较小，且在不同情况下，对于污染物浓度变化的影响会出现完全相反的效果，而湿度与风速则对于任何一种数据中的污染物都有很强的负相关关系，虽然相对而言，R^2 的值不是很大，但是在有限的数据量以及有限的自变量下，依然可以较为明确指出各个气象要素与污染物浓度之间的关系，有效分出了高影响类气象条件以及低影响类气象条件。后期如果有更多层次的气象要素以及以时间为序的更多的数据量，相信模型的运行效果会更佳。

针对于问题三，我们在对问题的需要进行明确后，选用了 LSTM 模型，该模型对于以时间序列为序的数据组有很大的优势，除了 LSTM 自带的“遗忘”功能来增加传统 RNN 神经网络模型的稳定性以外，还有手动调节时间序列步长的功能以及对于缺省值的包容性，这些都对于本次比赛中数据的缺测情况，或部分存在空值的情况，有着良好的稳健性。在模型的建立中，进行 5000 迭代之后，损失率将下降到 1%，并且逐步趋于稳定，在测试环节，模型的测试结果显示，预测较小值与真实较小值的之间差值较小，而较大值之间的差值较大，这可能是在极端值的影响之下，模型对于一些极大极小值的预测存在一定问题，这将是模型的重点改进方向，但是值得注意的是，模型对于以时间序列为序的数据之间的波动情况预测良好，误差极小，这说明模型在对于各个污染物未来的变化的趋势预测上有着非常显著的优势，如果 LSTM 模型中输入的自变量要素值种类增加，并且增加数据量，将极大地提升模型的稳健性，并且对于输入数据中的极大极小值进行二次预处理，将会对于模型地预测结果有很大地提升。但是也需要指出的是，对于模型输入量也应该有所控制，以 LSTM 模型对于时间序列的要求来看，输入的气象要素值，前后之间的变化最好也应该要有一定的次序，这样的变量将对于结果的提升有所帮助，在后面的工作中，应该对于气象要素的筛查加大力度，重点去筛查一些本身变化就有一定次序关系的气象要素来作为自变量。

针对于问题四，区域协同预报无疑将对于整体区域的污染物预测精确度有明显的提升，利用各个监测点的地理位置，尝试性地引入地理学第一定律，即各个事物之间的联系与各个事物之间的空间距离有着密切的联系，依次为基础，我们引进了反距离权重插值法，来标定 A,A1,A2,A3 四个监测点之间的各自位置，并且通过各自之间的距离为标准，建立一个拟合模型来重新设定各个监测点的气象要素，这将有以下好处：1、对于单个测站出现的极端值有很好的去极端作用，也可以相对于问题三中出现的极大极小问题提供一种解决思路。2、将要预测点的权重设为 50%，而剩下的测站共同承担剩下的 50%权重，这既保证了以主站数据为主，不被外围测站的数据过度干扰，也留下充足的权值值来让外围的测站来平衡主站中的一些极端值，或弥补一部分缺省值。在对数据进行以上的模拟操作之后，再次利用 LSTM 模型来完成建模，在测试阶段，尤其是对于极大极小值问题有很明显的改善，

以及对于模型预测的准确度也有很了很大的提升。但需要注意的是，外围测站与主测站之间建立关系的过程，还可以用更加精确的方法，在计算实际的各个测站之间的影响时，应该考虑更多的因素，如高度，地形等，这些都是后面要重点研究的方向。

9 参考文献

[1] 郝吉明, 马广大, 王书肖. 大气污染控制工程 [M]. 北京: 高等教育出版社, 2010.
[2] 中国环境科学研究院, 中国环境监测总站. 环境空气质量标准, 2012: 12P.;A4.
[3] 伯鑫等. 空气质量模型（SMOKE、WRF、CMAQ 等）操作指南及案例研究 [M]. 北京: 中国环境出版集团, 2019.
[4] 王景云, 张红日, 赵相伟, 等.2012-2015 年北京市空气质量指数变化及其与气象要素的关系[J]. 气象与环境科学, 2017, 40(04): 35-41.
[5] 李猛,宋晓奎,国慧.气象条件对大气污染物影响的地区差异性研究——以河北四地市为例[J].大众标准化,2021(12):248-250.
[6] 梅宁, 尹凤, 陆虹涛. 湿度变化对气体污染物扩散影响的研究[J]. 中国海洋大学学报(自然科学版), 2006(06): 987-990+994.
[7] 白盛楠, 申晓留. 基于 LSTM 循环神经网络的 PM2.5 预测[J]. 计算机应用与软件, 2019, 36(01): 67-70+104.
[8] Graves A. Supervised Sequence Labelling with Recurrent Neural Networks[M]. Springer Berlin Heidelberg, 2012:5-13.

10 附录

10.1问题一部分代码

```
% SO2  NO2  PM10 PM2.5 O3 CO
clear
load('F:\数模\data1.mat')

so2=aqdata(:,1); %max=20
no2=aqdata(:,2); %max=132
pm10=aqdata(:,3); %max=143
pm25=aqdata(:,4); %max=465 应该是异常值 只有465这个 剔除
o3=aqdata(:,5); %max=296
co=aqdata(:,6); %max=1.5

pm25(pm25>400)=nan;



%SO2 max=20 IAQI=浓度
iaqi_so2=so2;


%NO2 max=132  0 40 80 180
% [0,40]
no2(no2<=40)=no2(no2<=40)*50/40;
% (40,80]
no2(no2>40&no2<=80)=(no2(no2>40&no2<=80)-40)*50/40+50;
% (80,180]
no2(no2>80&no2<=180)=(no2(no2>80&no2<=180)-80)*50/100+100;
iaqi_no2=no2;

%pm10 max=143 0 50 150
% [0,50]
pm10(pm10<=50)=pm10(pm10<=50)*50/50;
% (50,150]
pm10(pm10>50&pm10<=150)=(pm10(pm10>50&pm10<=150)-50)*50/100+50;
iaqi_pm10=pm10;


%pm25  max=97 0 35 75 115
% [0,35]
pm25(pm25<=35)=pm25(pm25<=35)*50/35;
% (35,75]
pm25(pm25>35&pm25<=75)=(pm25(pm25>35&pm25<=75)-35)*50/40+50;
% (75,115]
pm25(pm25>75&pm25<=115)=(pm25(pm25>75&pm25<=115)-75)*50/40+100;
iaqi_pm25=pm25;


%o3  max=296  0 100 160 215 265 800
% [0,100]
o3(o3<=100)=o3(o3<=100)*50/100;
% (100,160]
o3(o3>100&o3<=160)=(o3(o3>100&o3<=160)-100)*50/60+50;
% (160,215]
o3(o3>160&o3<=215)=(o3(o3>160&o3<=215)-160)*50/55+100;
% (215,265]
o3(o3>215&o3<=265)=(o3(o3>215&o3<=265)-215)*50/50+150;
% (265,800]
o3(o3>265&o3<=800)=(o3(o3>265&o3<=800)-265)*100/535+200;
iaqi_o3=o3;

%co max=1.5 0 2
iaqi_co=co*50/2;
```

```
iaqi=[iaqi_so2 iaqi_no2 iaqi_pm10 iaqi_pm25 iaqi_o3 iaqi_co];
iaqi=ceil(iaqi);
[aqi,specid]=max(iaqi,[],2);
firstspec=cell(size(aqdata,2),1);
spec={'SO2','NO2','PM10','PM2.5','O3','CO'};
for i=1:length(specid)
    firstspec{i}=spec{specid(i)};
end
```

10.2 问题二部分代码

```python
import sklearn as sk
import pandas as pd
data = pd.read_excel(r'C:\Users\98178\Desktop\2021年中国研究生数学建模竞赛赛题\2021年B题\问题2 数据 -v2.xlsx'
                     ,sheet_name=6)
data = data.dropna()
data.rename(columns={'aqi差值':'下一时刻aqi变化量'},inplace=True)
feature = data[['温度', '湿度', '气压','风速', '风向',]] ## 设置自变量
labels = data[['SO2']] ## 这里因变量设置为SO2
labels = labels.values.squeeze()
feature = feature.values
sc = MinMaxScaler(feature_range=(0, 1))
#转换
X = feature
X = sc.fit_transform(X)
x = sm.add_constant(X) #生成自变量
y = labels#生成因变量
model = sm.OLS(y, x) #生成模型
result = model.fit() #模型拟合
result.summary() #模型描述
```

10.3 问题三、四部分代码

数据处理

```python
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
simulate = pd.read_excel(r'C:\Users\98178\Desktop\2021年中国研究生数学建模竞赛赛题\2021年B题\附件3 监测点A1、A2、A3空气质量预报基础
simulate['class'] = ''
var = 0
for year in ["2020","2021"]:
    for month in ["01","02","03","04","05",'06','07','08','09','10','11','12']:
        for day in range(1,32):
            indexs = simulate.loc[(simulate['模型运行日期'] == str(year)+'-'+str(month)+'-'+ str(day))].index
            indexs = indexs - var
            if len(indexs):
                for i in indexs:
                    index = np.floor(i / 24)
                    if index == 0: # 0 3 6 9
                        simulate['class'].loc[i + var] = 'a'
                    elif index == 1: # 1 4 7 11
                        simulate['class'].loc[i + var] = 'b'
                    else: # 2 5 8 12
                        simulate['class'].loc[i + var] = 'c'
                var = var + 72
simulate.to_excel('e:/shumoa3.xlsx')
```

```python
import pandas as pd
import numpy as np
shichi=  pd.read_excel(r'C:\Users\98178\Desktop\2021年中国研究生数学建模竞赛赛题\2021年B题\附件1 监测点C一次模拟数据分开.xlsx'
                       ,sheet_name=4)
shichi = shichi.replace('—',np.NAN)
shichi = shichi.dropna()
shichi.rename(columns={'监测时间': '预测时间'}, inplace=True)
yuci_1 = pd.read_excel(r'C:\Users\98178\Desktop\2021年中国研究生数学建模竞赛赛题\2021年B题\附件1 监测点C一次模拟数据分开.xlsx'
                       ,sheet_name=3)data_merge = pd.merge(yuci_1, shichi, on='预测时间', how='outer')
data_merge = data_merge.dropna()
newdata = data_merge[['预测时间', '地点_x', '近地2米温度（℃）', '地表温度（K）', '比湿（kg/kg）',
       '湿度（%）', '近地10米风速（m/s）', '近地10米风向（°）', '雨量（mm）', '云量', '边界层高度（m）',
       '大气压（Kpa）', '感热通量（W/m²）', '潜热通量（W/m²）', '长波辐射（W/m²）', '短波辐射（W/m²）',
       '地面太阳能辐射（W/m²）','SO2监测浓度(μg/m³)', 'NO2监测浓度(μg/m³)',
       'PM10监测浓度(μg/m³)', 'PM2.5监测浓度(μg/m³)', 'O3监测浓度(μg/m³)', 'CO监测浓度(mg/m³)',]]
newdata.to_excel(r'C:\Users\98178\Desktop\2021年中国研究生数学建模竞赛赛题\2021年B题\问题三\c站预测第三日的气象要素值与实测污染物数据.
```

LSTM 模型

```python
import torch.nn as nn


class lstm(nn.Module):

    def __init__(self, input_size=15, hidden_size=32, num_layers=1
                 , output_size=1 , dropout=0, batch_first=True):
        super(lstm, self).__init__()
        # lstm的输入 #batch,seq_len, input_size
        self.hidden_size = hidden_size
        self.input_size = input_size
        self.num_layers = num_layers
        self.output_size = output_size
        self.dropout = dropout
        self.batch_first = batch_first
        self.rnn = nn.LSTM(input_size=self.input_size, hidden_size=self.hidden_size, num_layers=self.num_layers
                           , batch_first=self.batch_first, dropout=self.dropout )
        self.linear = nn.Linear(self.hidden_size, self.output_size)

    def forward(self, x):
        out, (hidden, cell) = self.rnn(x)
        # a, b, c = hidden.shape
        # out = self.linear(hidden.reshape(a * b, c))
        out = self.linear(hidden)
        return out
```

```python
from torch.autograd import Variable
import torch.nn as nn
import torch
from LSTMModel import lstm
from parser_my import args
from dataset import getData

def train():

    model = lstm(input_size=args.input_size, hidden_size=args.hidden_size, num_layers=args.layers , output_size=1, dropo
    model.to(args.device)
    criterion = nn.MSELoss()  # 定义损失函数
    optimizer = torch.optim.Adam(model.parameters(), lr=args.lr)  # Adam梯度下降  学习率=0.001

    close_max, close_min, train_loader, test_loader = getData(args.corpusFile,args.sequence_length,args.batch_size )
    for i in range(args.epochs):
        total_loss = 0
        for idx, (data, label) in enumerate(train_loader):
            if args.useGPU:
                data1 = data.squeeze(1).cuda()
                pred = model(Variable(data1).cuda())
                # print(pred.shape)
                pred = pred[1,:,:]
                label = label.unsqueeze(1).cuda()
                # print(label.shape)
```

```
            else:
                data1 = data.squeeze(1)
                pred = model(Variable(data1))
                pred = pred[1, :, :]
                label = label.unsqueeze(1)
            loss = criterion(pred, label)
            optimizer.zero_grad()
            loss.backward()
            optimizer.step()
            total_loss += loss.item()
        print('LOSS: ' + str(total_loss))
        if i % 10 == 0:
            # torch.save(model, args.save_file)
            torch.save({'state_dict': model.state_dict()}, args.save_file)
            print('第%d epoch，保存模型' % i)
    # torch.save(model, args.save_file)
    torch.save({'state_dict': model.state_dict()}, args.save_file)

train()

```

```
from LSTMModel import lstm
from dataset import getData
from parser_my import args
import torch
import matplotlib.pyplot as plt
#解决中文显示问题
plt.rcParams['font.sans-serif']=['SimHei']
plt.rcParams['axes.unicode_minus'] = False


def eval():
    # model = torch.load(args.save_file)
    model = lstm(input_size=args.input_size, hidden_size=args.hidden_size, num_layers=args.layers , output_size=1)
    model.to(args.device)
    checkpoint = torch.load(args.save_file)
    model.load_state_dict(checkpoint['state_dict'])
    preds = []
    labels = []
    yuci = [] # hjq
    zhenshi = [] # hjq
    close_max, close_min, train_loader, test_loader = getData(args.corpusFile, args.sequence_length, args.batch_size)
    for idx, (x, label) in enumerate(test_loader):
        if args.useGPU:
            x = x.squeeze(1).cuda() # batch_size,seq_len,input_size
```

```
        else:
            x = x.squeeze(1)
        print(x)
        pred = model(x)
        list = pred.data.squeeze(1).tolist()
        preds.extend(list[-1])
        labels.extend(label.tolist())

    for i in range(len(preds)):
        print('预测值是%.2f,预测值取整是%d,真实值是%.2f,输出概率是%.2f' % (
        preds[i][0] * (close_max - close_min) + close_min, round(preds[i][0] * (close_max - close_min) + close_min), lab
        yuci.append(preds[i][0] * (close_max - close_min) + close_min)
        zhenshi.append(labels[i] * (close_max - close_min) + close_min)
    print(yuci)
    print(zhenshi)
    plt.figure(figsize=(10, 5))
    plt.plot(yuci, 'r', label='预测值')
    plt.plot(zhenshi, 'b', label='真实值')
    plt.ylabel('O3监测浓度(μg/m^3)')
    plt.xlabel('相对于起始的天数(d)')
    plt.legend(loc='best')
    plt.show()
eval()
```

```python
from LSTMModel import lstm
from parser_my import args
import torch
from pandas import read_csv
import numpy as np
from torch.utils.data import DataLoader
from dataset import Mydataset
from torch.autograd import Variable
import matplotlib.pyplot as plt
import pandas as pd
#解决中文显示问题
plt.rcParams['font.sans-serif']=['SimHei']
plt.rcParams['axes.unicode_minus'] = False

model = lstm(input_size=args.input_size, hidden_size=args.hidden_size, num_layers=args.layers , output_size=1)
model.to(args.device)
checkpoint = torch.load(args.save_file)
model.load_state_dict(checkpoint['state_dict'])

data = read_csv('./data/predict.csv')
df = data.apply(lambda x: (x - min(x)) / (max(x) - min(x)))
X = []
Y = []
for i in range(df.shape[0] - args.sequence_length):
    X.append(np.array(df.iloc[i:(i + args.sequence_length), ].values, dtype=np.float32))
    Y.append(np.array(df.iloc[(i + args.sequence_length), 0], dtype=np.float32))
```

```python
dl = DataLoader(dataset=Mydataset(X, Y), batch_size=args.batch_size, shuffle=True)
preds = []
labels = []
yuci = []
zhenshi = []
close_max = 2.5
close_min = 0.2
for idx, (x, label) in enumerate(dl):
    if args.useGPU:
        x = x.squeeze(1).cuda()  # batch_size,seq_len,input_size
    else:
        x = x.squeeze(1)
    pred = model(x)
    list = pred.data.squeeze(1).tolist()
    preds.extend(list[-1])
for i in range(len(preds)):
    #print('预测值是%.2f,预测值取整是%d' % (
     #   preds[i][0] * (close_max - close_min) + close_min, round(preds[i][0] * (close_max - close_min) + close_min)))
    yuci.append(round(preds[i][0] * (close_max - close_min) + close_min))
data = pd.DataFrame(yuci,columns=['co'])
data.to_excel(r'C:\Users\98178\Desktop\2021年中国研究生数学建模竞赛赛题\2021年B题\问题三\第三问预测数据\anew\a_co.xlsx')
print(data)
```