



研究报告

基于TOPSIS环境评价及 $PM_{2.5}$ 含
量LSTM神经网络预测

2022年4月16日

摘要

随着“绿水青山就是金山银山”战略以及“碳达峰和碳中和”目标的提出，表明中国将以新发展理念为引领，在推动高质量发展中促进经济社会全面绿色转型开启绿色低碳时代。近来环境监测设备已经不是问题，而如何对检测数据进行良好的挖掘成为了有效治理环境污染的重要环节。为了研究环境的发展趋势，首先必须对环境质量做出科学的评价，预测其发展趋势，科学评价和预测对于实现环境后续保护措施有重要意义。

本文的主要研究内容和创新点如下：

(1)由于环境质量的评价具有主观性，本文从环境评价模型的普适性推广性出发，利用熵权法解决了传统评价模型各评价指标权重一致、无区分度的问题，并结合TOPSIS综合评价模型在很大程度上改进了传统评价方法对指标和数据量要求严苛的局限性。以塞罕坝三十年来的数据为例，可视化地展示了当地环境质量的客观发展趋势。

(2)本文分析了全国各省份的各种空气指标，并利用热力图将全国各省的环境质量进行可视化。分析结果直观，与实际各省份实际空气质量情况相一致。能为环保部门的后续措施提供参考。

(3)本文利用LSTM神经网络对北京市朝阳区从2015年7月17日中午12点至2022年4月12日中午12点每个小时的 $PM_{2.5}$ 的含量，共计50764条 $PM_{2.5}$ 的数据，对接下来一周每小时的空气质量进行预测。

关键字：环境评价空气质量预测 TOPSIS综合评价模型 LSTM神经网络

目录

1 问题描述与解决办法	4
2 模型适用性检验	6
2.1 数据来源	6
2.2 数据分析和处理	6
2.3 异常值处理	8
3 综合评价方法介绍	9
3.1 熵权法	9
3.2 TOPSIS综合评价法	10
4 大气环境综合评价	12
4.1 $PM_{2.5}$ 浓度和 PM_{10}	12
4.2 AQI(空气质量指数)	12
5 空气质量变化预测分析	16
5.1 RNN神经网络介绍	17
5.2 LSTM神经网络介绍	18
5.3 模型评价	20
5.4 数据清洗	20
5.5 LSTM神经网络预测	22
5.5.1 LSTM神经网络实时预测	22
5.5.2 LSTM神经网络72小时预测	24
6 北京市朝阳区空气污染治理状况及对策	27
7 研究结论总结	29
8 参考文献	30
A 附录	31
A.1 完整的预测表格	31
A.1.1 24小时 $PM_{2.5}$ 实时预测表格	31
A.1.2 72小时 $PM_{2.5}$ 预测表格	32

目录	3
----	---

A.2 相关代码	33
A.2.1 <i>TOSIS</i> 综合评价代码	33
A.2.2 热力图生成代码	38
A.2.3 $PM_{2.5}$ 实时预测LSTM神经网络代码	39
A.2.4 $PM_{2.5}$ 72小时预测LSTM神经网络代码	51

1 问题描述与解决办法

2020年9月，习近平总书记提出“中国二氧化碳排放力争于2030年前达到峰值，努力争取2060年前实现碳中和”的发展目标，自然环境、大气环境越来越受到人们的关注。近年来，国内已经出现了许多环境空气质量实时自动检测系统和重点大气污染源的检测系统。

获取监测数据的系统工程已经日趋完善，现代城市环境空气质量管理面临的主要问题是如何有效管理数据资源并挖掘出数据中蕴含的丰富信息，充分发挥信息潜力及价值^[1]。基于数据信息的挖掘可以为大气环境的评价以及后续的环境指标预测打下坚实的基础。

针对目前研究的局限性，同时为了建立一个具有普适性的环境评价预测系统，引入了熵权法定权模型。可以消除模糊评价、TOPSIS综合评价法各个权重重要性无区分度的局限性。熵权法^[7]利用熵值来判断某个指标的离散程度，其信息熵值越小，指标的离散程度越大，该指标对综合评价的影响越大，相应的该指标的权重越大。TOPSIS综合评价法^[8]避免了数据的主观性，不需要目标函数，能够很好的刻画多个影响指标的影响程度；同时TOPSIS综合评价法对数据分布量、指标多少无限制，适用于大样本也适用于小样本，可以适用于多评价单元、多指标的大系统。同时由于TOPSIS模型的普适性，也可以用于其他自然环境指标系统的评价。

目前用于大数据预测时间序列的算法有很多种，诸如AR、MA、ARMA模型等。本文使用的数据集高达五万多，数据量很大，一般用于时序分析的都是小样本算法，不适合进行大数据时间序列预测。而长短期记忆神经网络作为循环神经网络的变种，能够比一般的循环神经网络和其他种类的神经网络在更长的时间序列中有更好的表现。

表1是本文使用的算法的优缺点。

表 1: Advantages and disadvantages

模型	优点	缺点
模糊综合评价	1.精确的数字手段处理模糊的评价对象对信息呈现模糊性数据作出比较科学的量化评价 2.评价结果是一个矢量，包含的信息比较丰富,可比较准确的刻画被评价对象。	1.计算复杂，对指标权重矢量的确定主观性较强。 2.指标集个数较大时,会出现超模糊现象。
RNN神经网络	具有良好的非线性衍射逼近性能，循环传递神经网络的数据具有一定的记忆力。	1.在处理长序列时经常出现“梯度消失”。 2.隐藏层结构难以确定
LSTM神经网络	可以拟合序列数据(LSTM)通过遗忘门和输出门忘记部分信息来解决梯度消失的问题。	1.LSTM虽然部分解决了rnn梯度消失问题，但是信息在过远的距离传播中损失很厉害 2.无法很好的并行

2 模型适用性检验

2.1 数据来源

本文所有的数据均来自国家统计局、河北省统计局和开源空气质量组织(OpenAQ)的网站。

2.2 数据分析和处理

由于塞罕坝在过去的几十年间环境改造取得了举世瞩目的成果，本文通过选取塞罕坝环境数据来检验模型的适用性，通过查询国家统计局数据库得到河北塞罕坝森林覆盖率、森林覆盖面积、林木蓄积、涵养水量、二氧化碳吸收量、氧气释放量6个环境评价指标自1962年至2021年间的完整数据。首先对获取的塞罕坝数据利用SPSS软件进行描述统计：

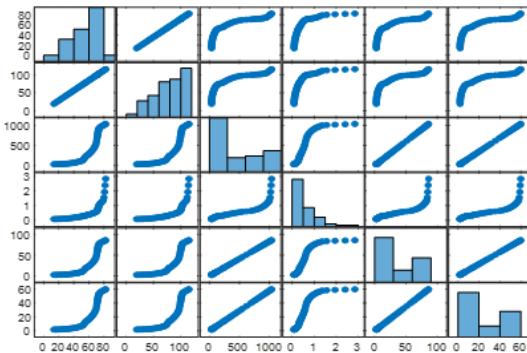


图 1: 数据描述统计图

变异系数CV:

$$CV = \frac{\delta}{\mu} \quad (1)$$

其中为样本数据的标准差，为样本数据的平均值。

当需要比较数据离散程度大小的时候，如果数据的测量尺度相差太大，或者数据量纲的不同，直接使用标准差来进行比较不合适，此时就应当消除测量尺度和量纲的影响，而变异系数可以做到这一点，标准差与其平均数的比。CV虽然没有量纲，同时又按照其均数大小进行了标准化，这样就可以进行客观比较了。因此，可以认为变异系数和极差、标准差和方差一

样，都是反映数据离散程度的绝对值。其数据大小不仅受变量值离散程度的影响，而且还受变量值平均水平大小的影响。

变量名	样本量	最大值	最小值	平均值	标准差	中位数	方差	峰度	偏度	变异系数 (CV)
二氧化碳吸收量/万吨	60	86.03	2.74	35.263	30.87	23.71	952.98	-1.448	0.469	0.875
森林覆盖率	60	82.21	13.57	55.986	19.232	59.84	369.859	-0.739	-0.611	0.344
氧气释放量/万吨	60	59.84	1.9	24.527	21.473	16.495	461.069	-1.447	0.469	0.875
涵养水量/亿立方米	60	2.84	0.09	0.591	0.565	0.395	0.32	4.634	1.984	0.956
覆盖面积/万亩	60	115.1	19	78.38	26.924	83.775	724.926	-0.739	-0.61	0.344
林木蓄积/万立方米	60	1036.8	33	424.961372.038285.74138412.227	-1.447	0.469	0.875			

图 2: 变异系数

当变异系数 $CV > 0.15$ 时，可认为数据中有异常值：

1. 基于二氧化碳吸收量/万吨，变异系数 (CV) 为 0.875，大于 0.15，当前数据中可能存在异常值，建议对异常的或者表现得较为突出的指标进行分析。
2. 基于森林覆盖率，变异系数 (CV) 为 0.344，大于 0.15，当前数据中可能存在异常值，建议对异常的或者表现得较为突出的指标进行分析。
3. 基于氧气释放量/万吨，变异系数 (CV) 为 0.875，大于 0.15，当前数据中可能存在异常值，建议对异常的或者表现得较为突出的指标进行分析。
4. 基于涵养水量/亿立方米，变异系数 (CV) 为 0.956，大于 0.15，当前数据中可能存在异常值，建议对异常的或者表现得较为突出的指标进行分析。
5. 基于覆盖面积/万亩，变异系数 (CV) 为 0.344，大于 0.15，当前数据中可能存在异常值，建议对异常的或者表现得较为突出的指标进行分析。
6. 基于林木蓄积/万立方米，变异系数 (CV) 为 0.875，大于 0.15，当前数据中可能存在异常值，建议对异常的或者表现得较为突出的指标进行分析。

2.3 异常值处理

一组数据只含有随机误差，对其进行计算处理得到标准偏差，按一定概率确定一个区间，认为凡超过这个区间的误差，就不属于随机误差而是粗大误差，含有该误差的数据应予以剔除。这种判别处理原理及方法仅局限于对正态或近似正态分布的样本数据处理，它是以测量次数充分大为前提。根据 3σ 原则。

$$\text{原则} = \begin{cases} \text{数值分布在 } (\mu - \delta, \mu + \delta) \text{ 中的概率为 } 0.6826 \\ \text{数值分布在 } (\mu - 2\delta, \mu + 2\delta) \text{ 中的概率为 } 0.9544 \\ \text{数值分布在 } (\mu - 3\delta, \mu + 3\delta) \text{ 中的概率为 } 0.9974 \end{cases} \quad (2)$$

对样本进行等精度测量，独立得到 x_1, x_2, \dots, x_n 算出其算术平均值，以及剩余误差，并计算标准差，当样本值满足：

$$|v_i| = |x_i - \bar{x}| > 3\delta \quad (3)$$

则 x_i 应该剔除。据此，筛选出样本中的异常值。对于异常值的处理采用相邻两数据的平均值进行替换。



图 3: 塞罕坝环境评价指标

建立塞罕坝对生态环境的影响评价模型是对评价方法的应用，目前综合评价方法多种多样，如模糊评价法、加权平均法、层次分析法、主成分分析法、熵权评价法、灰色关联评价法、TOPSIS方法和数据包络法等。本文结合熵权法和TOPSIS方法，分析塞罕坝地区改造对环境影响问题。

3 综合评价方法介绍

3.1 熵权法

熵权法是一个客观的赋权方法,可以最大程度上避免主观性赋权对于环境指标量化结果的影响。熵权法依据的原理是指标的变异程度,即变异程度越高则对应的权值也就越高。

首先本文需要对环境指标数据进行正向化和归一化处理,保证数据的统一性:

$$z_{ij} = \frac{x_{ij} - x_{min}}{x_{max} - x_{min}} \quad (4)$$

其中为归一化处理后的变量, x_{min} 和 x_{max} 分别为每个指标的最大值和最值。计算第j个环境指标下第i个年份所占权重,将其看作计算信息熵时的概率

$$p_{ij} = \frac{z_{ij}}{\sum_{i=1}^n z_{ij}} \quad (5)$$

计算第j个环境指标的信息熵,并计算对应信息效用值,此处进行转换的原因是因为信息熵越大代表该环境指标的信息越少,引入就可以正向衡量信息量。

$$e_j = -\frac{1}{ln n} \sum_{i=1}^n \ln(p_{ij}) \quad (6)$$

$$d_j = 1 - e_j \quad (7)$$

最终归一化得到每个环境评价指标的熵权 w_j

$$w_j = \frac{d_j}{\sum_{j=1}^m d_j} \quad (8)$$

得到6个权重分别为:

	森林覆盖率	森林覆盖面积	林木面积	涵养水量
Weights	0.0561	0.0561	0.2198	0.2309

	二氧化碳吸收量	氧气释放量
Weights	0.2198	0.2198

表 2: 权重表

塞罕坝环境综合评价一项较为复杂的系统工程，在研究此问题时不能忽视指标本身所蕴含的信息。采用了熵权分析法来处理指标权重。按照信息论最基本的原理，信息作为系统有序程度的度量，熵则是系统无序程度的度量，熵权法利用熵值来判断某个指标的离散程度，其信息熵值越小，指标的离散程度越大，该指标对综合评价的影响越大，相应的该指标的权重越大。

3.2 TOPSIS综合评价法

TOPSIS法是用来处理指标决策问题的多方案排序和选择的方法，它的基本思想是：依据理想点的理论原理，找寻距离理想点最近的方案。并通过计算对象与最优解、最劣解的距离大小，确定顺序。即先设定一个虚拟的最优解（又称正理想解）和一个最劣解（又称负理想解），将各备选方案与正负理想解相互比较，若方案最靠近最优解即又距最劣解最远为最好。另外，TOPSIS方法需要的评价指标决策矩阵和指标权重，由上文中的熵权法计算给出。

所选指标均为效益性指标，故不做另外处理。找出每列也就是每个环境指标的最大值，记为 $z_i^+(i=1, 2, \dots, m)$ ，组成向量

$$Z^+ = z_1^+, z_2^+, \dots, z_m^+ \quad (9)$$

该向量代表了环境最好的年份。同样的，找出每列也就是每个指标的最小值，记为 $z_i^-(i=1, 2, \dots, m)$ ，组成向量

$$Z^- = z_1^-, z_2^-, \dots, z_m^- \quad (10)$$

该向量代表了环境最差的年份。

定义第*i*个年份与理想目标距离为 D_i^+ ，计算公式为

$$D_i^+ = \sqrt{\sum_{j=1}^m w_j (z_j^+ - z_{ij})^2} \quad (11)$$

定义第*i*个年份与不理想目标距离为 D_i^- ，计算公式为

$$D_i^- = \sqrt{\sum_{j=1}^m w_j (z_j^- - z_{ij})^2} \quad (12)$$

定义第*i*个年份的得分为 S_i , 计算公式为

$$S_i = \frac{D_i^-}{D_i^+ + D_i^-} \quad (13)$$

显然, $S_i \in [0, 1]$ 当 S_i 越接近于1, 说明此年份*i*距离理想化目标越近, 该年份塞罕坝环境就越好。反之, 当越接近于0, 说明年份*i*距离理想化目标越远, 该年份塞罕坝环境就越差。

计算数据得到1962-2021年间塞罕坝环境的量化得分:

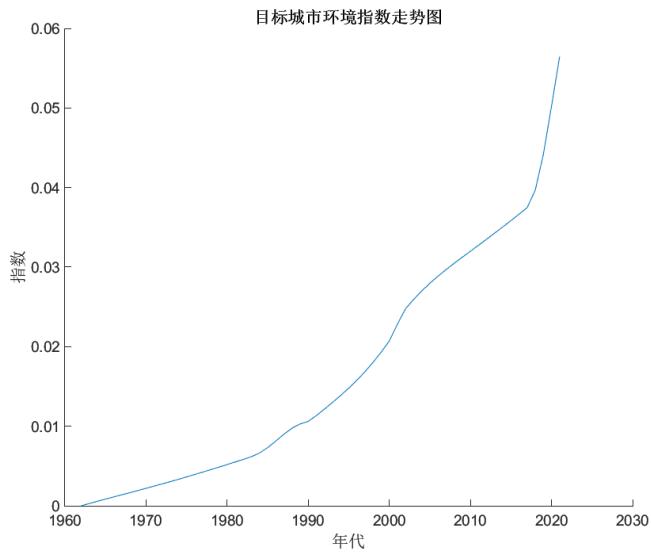


图 4: 塞罕坝评价历年环境指数

根据评价模型得出的1962年到2021年塞罕坝环境得分呈上升趋势, 可知塞罕坝恢复对环境改造有积极作用。塞罕坝几代人的防沙治沙卓有成效, 不仅留下了巨大精神财富, 也为后来环境改造工程提供了范本。

4 大气环境综合评价

目前全国的环境保护监测站已经建立起一套十分完备的城市环境质量监测系统，包括空气质量实时监测系统和重点污染源在线监控系，使得全国各地的空气质量监测数据都能通过系统上传到城市环境空气质量数据库^[9]最终被国家统计局收录，本文从国家统计局官网下载2021年12月各个市份 $PM_{2.5}$ 、 PM_{10} 、AQI(空气质量指数)作为评价指标数据来研究本文中模型的适用性。

4.1 $PM_{2.5}$ 浓度和 PM_{10}

大气污染物因子有很多种，当前我国环境保护部门监测环境空气污染物时采用的是 PM_{10} 这个指标。其定义是空气中当量直径为10 μm 的尘埃或飘尘的浓度。由此，可知 $PM_{2.5}$ 是直径小于或等于2.5 μm 的尘埃或飘尘在环境空气中的浓度 $PM_{2.5}$ 。 $PM_{2.5}$ 和 PM_{10} 的主要成分包括：含碳颗粒（包括元素碳和有机碳，元素碳主要产生于高温燃烧过程，有机碳则主要来自相对低温过程的不完全燃烧产物）、硫酸盐、硝酸盐、铵盐、重金属等。 $PM_{2.5}$ 和 PM_{10} 在空气悬浮过程中还会进一步吸附空气中存在的有机和金属等化学成分、细菌、病毒、真菌等微生物成分。对空气质量破坏明显，对人体危害极大。

4.2 AQI(空气质量指数)

空气质量指数 (Air Quality Index, 简称AQI)，是一个用来定量描述空气质量水平的数值。AQI的取值范围位于0-500 之间。环境空气污染物的种类有很多，参与AQI指数评价的有二氧化硫(SO₂)、二氧化氮(NO₂)、一氧化碳(CO)、臭氧(O₃)、 $PM_{2.5}$ 、 PM_{10} 。

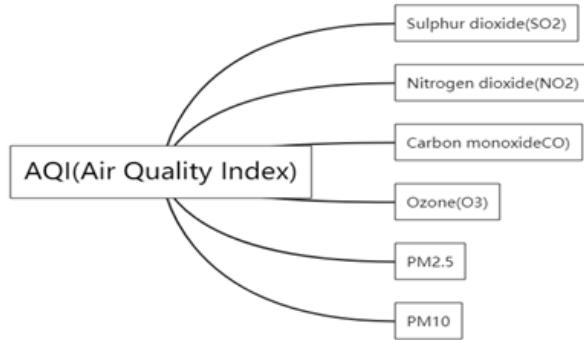


图 5: AQI评价指标内容

环境监测部门每天发布的空气质量报告中，会包含各种污染物的浓度值。很难从这么多个抽象的浓度数据中判断出到底当前的空气质量处在什么水平。于是将各种不同污染物含量折算成一个统一的指数，这就是空气质量指数。空气质量指数的值在不同的区间，就代表了不同的空气质量水平。比如0-50之间，代表良好；51-100之间，代表中等；101-150之间代表轻度污染等等。为了更直观起见，每个区间都有一个固定的颜色值与它对应：

Air Quality Index (AQI)	Air Quality Classification	
0-50	Good	
51-100	Moderate	
101-150	Lightly Polluted	
151-200	Moderately Polluted	
201-300	Heavily Polluted	
>300	Severely Polluted	

图 6: AQI数据对应表

AQI的折算公式如下：

$$I = \frac{I_{high} - I_{low}}{C_{high} - C_{low}}(C - C_{low}) + I_{low} \quad (14)$$

其中I等于空气质量指数，即AQI，输出值；C为污染物浓度，输入值；
 C_{low} 为小于或等于C的浓度限值 C_{high} 为大于或等于C的浓度限值； I_{low} 对应

于 C_{low} 的指数限值; I_{high} 为对应于 C_{high} 的指数限值。将计算好的数据按照从低到高进行排序并使用matlab绘制出条形图。

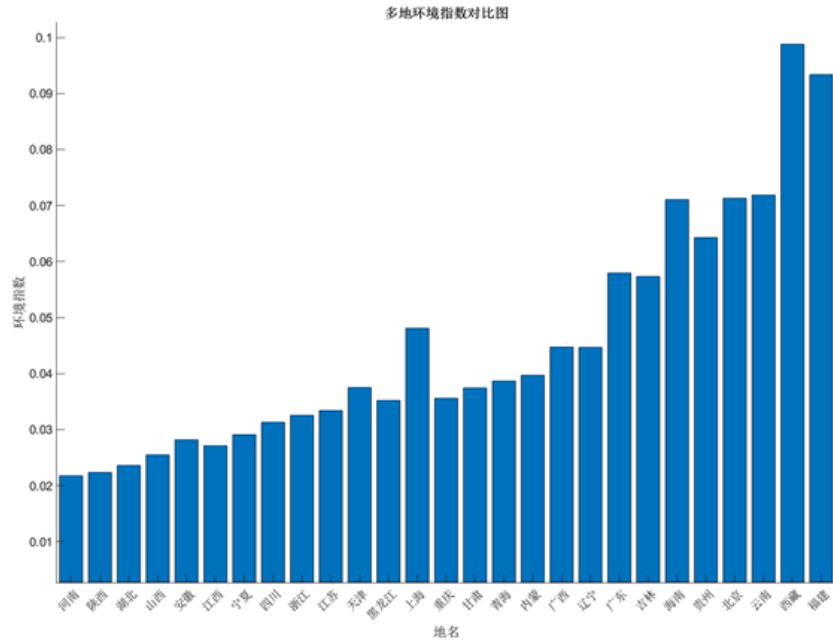


图 7: 各省环境质量排名图

利用Matlab编程得到环境质量指数热力图，将各个省份的数据可视化后，可以看出我国南方地区普遍空气质量较好。除开新疆沙漠的原因导致的气候异常，华北平原和华中平原目前属于环境空气污染较为严重的地区。



图 8: 全国环境质量指数热力图

5 空气质量变化预测分析

为了预测北京市朝阳区后续的空气质量变化，本文使用了China - OpenAQ组织的Beijing US Embassy项目中北京市朝阳区2015年7月17日中午12点-2022年4月12日中午十二点每个小时的 $PM_{2.5}$ 含量，共50764条数据，清洗后的数据如下图所示。

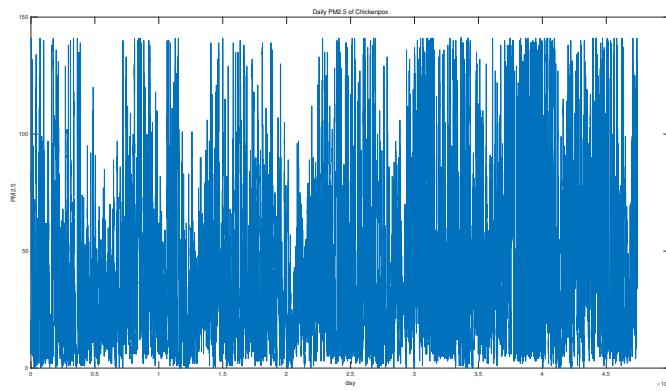


图 9: 清洗后 $PM_{2.5}$ 分布图

将清洗后的数据使用中国气象局的标准^[4]转换为空气质量指标，北京市朝阳区近2115天中的优的空气质量占52%，良占%29,轻度污染占10%，中度污染占4%，重度污染占1%，严重污染占1%。

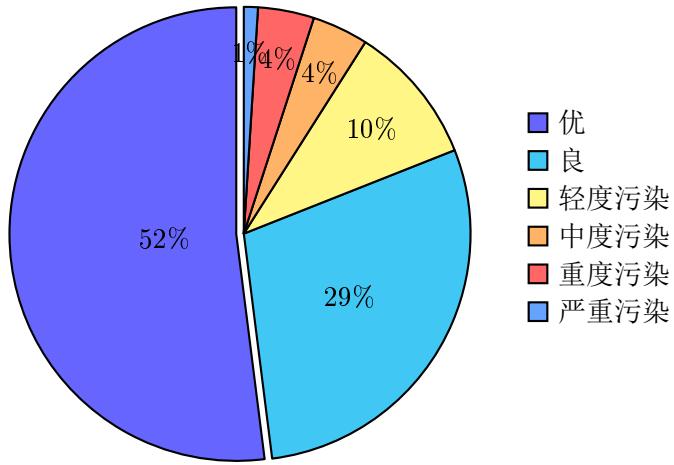


图 10: 北京市朝阳区空气质量饼图

5.1 RNN神经网络介绍

处理与事件发生的时间轴有关系的问题时，比如自然语言处理，文本处理，文字的上下文是一定的关联性的；时间序列数据，如连续几天的天气状况，当日的天气情况与过去的几天有某些联系；又比如语音识别，机器翻译等。在考虑这些和时间轴相关的问题时，传统的神经网络就无能为力了，因此就有了RNN（recurrent neural network，循环神经网络）。

RNN之所以被称为循环神经网络是因为一个序列的输出与前一时刻的输出有关，前面数据信息会影响后一个输出，隐含层的节点是相互关联的。

RNN网络结构如下图所示，RNN模型结构展开图见图1，其中 a 表示输入样本， y 表示训练后的输出样本， $t-1, t, t+1$ 分别表示时间序列， U, W, V 是在每一刻都共享的网络权重， U 为某一时刻输入样本的权重、 W 为隐含层的权重、 V 表示输出的样本权重。表示当前 t 时刻的隐藏状态，由当前时刻的输入样本 x ，和 $t-1$ 时刻的隐藏状态共同决定，表达式为：

$$s_t = f(Ux_t + Ws_{t-1}) \quad (15)$$

$$y_t = g(Vs_t) \quad (16)$$

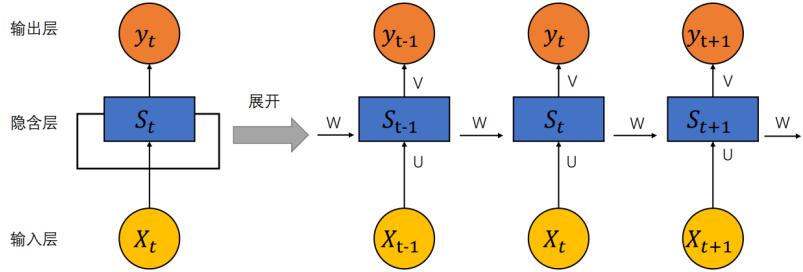


图 11: RNN网络结构图

5.2 LSTM神经网络介绍

长短期记忆（Long short-term memory, LSTM）是一种改进的RNN，主要是为了解决长序列训练过程中的梯度消失和梯度爆炸问题。长短期记忆网络的结构如下图所示，在一般的循环神经网络中，记忆单元没有衡量信息的价值量的能力，因此，记忆单元对于每个时刻的状态信息等同视之，这就导致了记忆单元中往往存储了一些无用的信息，而真正有用的信息却被这些无用的信息弱化。LSTM正是从这一点出发做了相应改进，和一般结构的循环神经网络只有一种网络状态不同，LSTM中将网络的状态分为内部状态和外部状态两种。LSTM的外部状态类似于一般结构的循环神经网络中的状态，即该状态既是当前时刻隐藏层的输出，也是下一时刻隐藏层的输入。这里的内部状态则是LSTM特有的。

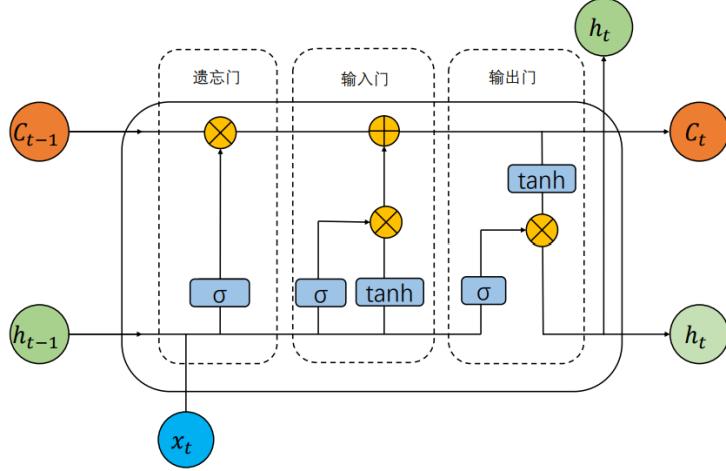


图 12: RNN网络结构图

在LSTM中有三个称之为“门”的控制单元，分别是输入门（input gate）、输出门（output gate）和遗忘门（forget gate），其中输入门和遗忘门是LSTM能够记忆长期依赖的关键。输入门决定了当前时刻网络的状态有多少信息需要保存到内部状态中，而遗忘门则决定了过去的状态信息有多少需要丢弃。最后，由输出门决定当前时刻的内部状态有多少信息需要输出给外部状态。遗忘门：

$$n_t = \delta(W_n[h_{t-1}, x_t] + bn) \quad (17)$$

输入门：

$$l_t = \delta(W_l[h_{t-1}, x_t] + bi) \quad (18)$$

$$r_t = \tanh(W_r[h_{t-1}, x_t] + br) \quad (19)$$

$$c_t = n_t * c_{t-1} + l_t * r_t \quad (20)$$

输出门：

$$m_t = \delta(W_m[h_{t-1}, x_t] + bm) \quad (21)$$

$$h_t = m_t * \tanh(c_t) \quad (22)$$

其中为sigmoid激活函数将数据映射到0-1之间：

$$f(x) = \frac{1}{1 + e^{-x}} \quad (23)$$

\tanh 是双曲正切激活函数，表示当前时刻的样本输入,其中， W 为权值矩阵, B 表示为参数矩阵, C 表示遗忘门输出, I 为输入门输入, O 为输出门输出, S 表示当前时刻保存的信息。

本文使用两个不同隐藏层结构的LSTM神经网络，分别用于实时预测和72小时预测。

5.3 模型评价

模型误差大多采用如下三种公式进行计算。

均方误差 (MSE):

$$MSE = \frac{1}{n} \sum_{i=1}^n (\hat{y} - y)^2 \quad (24)$$

平均相对误差 (MAE):

$$MAE = \frac{1}{n} \sum_{i=1}^n |\hat{y} - y| \quad (25)$$

根均方误差 (RMSE):

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (\hat{y} - y)^2} \quad (26)$$

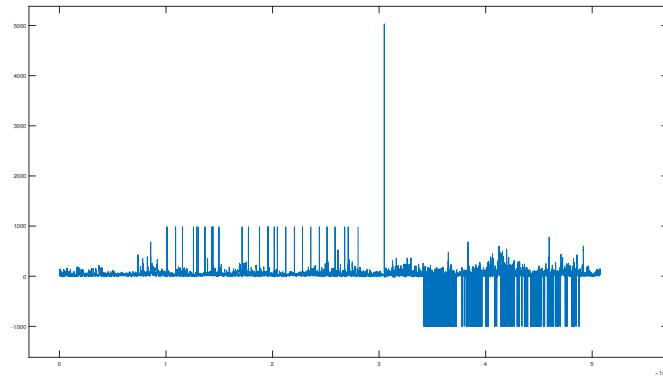
而本文使用第三种方式进行误差评估，理由是根均方误差度量模型所预测的值与实际值之间的平均差值，表示了相对于真实值的离散程度,根均方误差的值越小，模型的质量越好。

5.4 数据清洗

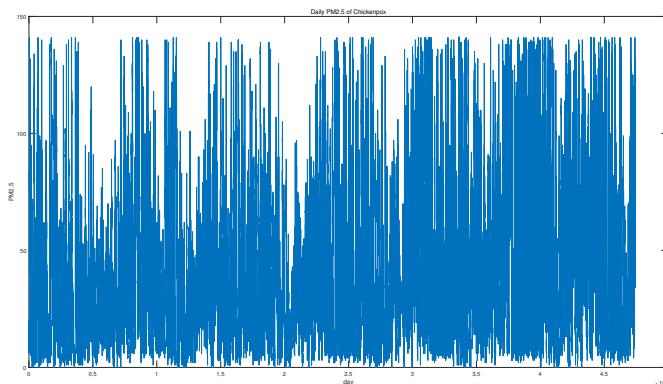
利用Matlab进行LSTM神经网络训练时，首先需要对数据进行清洗。本文将比上四分位数 (75%) 大 1.5 个四分位差以上或比下四分位数 (25%) 小 1.5 个四分位差以上的元素和低于0值的元素作为离群值进行删除。将含量低于0微克/立方公尺的元素清除的理由是 $PM_{2.5}$ 的含量不可能低于0。

空缺值使用相邻非缺失值的线性插值。

清洗前数据分布为：

图 13: 清洗前 $PM_{2.5}$ 分布图

清洗后数据分布为:

图 14: 清洗后 $PM_{2.5}$ 分布图

5.5 LSTM神经网络预测

5.5.1 LSTM神经网络实时预测

本文用于实施预测的LSTM神经网络的网络结构如表3所示

LSTM	Input Layer(1)
	Fully Connected Layer(200)
	LSTM Layer(200)
	Dropout Layaer(0.2)
	Fully Connected Layer(200)
	Fully Connected Layer(1)
	Regression Layer

表 3: 实时预测LSTM神经网络结构

使用的隐藏层为五层，第一层是有200个神经元的全连接层，第二层有200个神经元的LSTM层，第三层抛弃层遗忘率设置为0.2，第四层是有200个神经元的全连接层，第五层是有1个神经元的全连接层用于输出。前125次的学习率设置为0.0005，后125次的学习率设置为0.25。



图 15: 根均方误差图和损失函数图

从图15中可以看到模型在第50次训练后，以后基本上使得根均方差保持在了0.3左右，同时损失值也相对降低到了最低值。前期训练使用0.0005的

学习率是为了更精确找到损失函数的最小值，在后面125次训练中使用较大的学习率是为了防止梯度爆炸而陷入局部优解，错过最优解。

处理后的数据有50764条数据，将其前90%的数据作为训练集，将剩余的10%的数据作为测试集。设置好神经网络的结构后，使用CPU进行计算并使用实时更新数据集的方式进行预测训练。

训练结束后，将预测数据与测试数据进行对比，计算根均方差，结果如下。

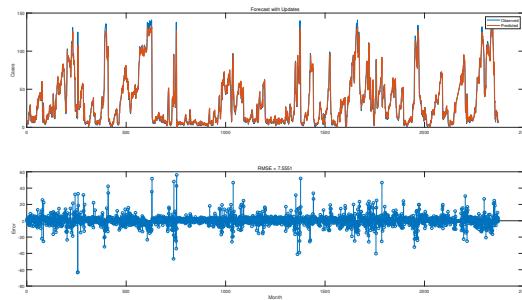


图 16: 实时预测和根均方误差 (RMSE)

其中，根均方差为7.409。

考虑到只进行数值上的预测不够直观，于是将预测结果和真实值换算成气象局^[4]规定的空气质量指标。由于图16中测试数据较多，附录中存取了24小时的预测数据用于参考。表4只选取了24小时实时预测的部分数据作为文章内容中的演示，完整的预测表格表7可以在附录找到。

时间	预测等级	实际等级
第1个小时	优	优
第2个小时	优	优
...
第23个小时	优	优
第24个小时	优	优

表 4: 24小时预测 (部分)

换算后，根均方差为0.2889。基本能够准确地实现实时预测。

5.5.2 LSTM神经网络72小时预测

本文用于实施预测的LSTM神经网络的网络结构如表5所示

LSTM	Input Layer(240)
	Fully Connected Layer(200)
	LSTM Layer(200)
	Drop Out Layer(0.2)
	Fully Connected Layer(200)
	Fully Connected Layer(1)
	Regression Layer

表 5: 实时预测LSTM神经网络结构

除神经网络的结构不一样之外，本文用于预测72小时的LSTM神经网络的方式区别与本文用于实时预测的LSTM的最主要的特征是使用了240个时间节点的数据去预测第241个时间节点的数据，而用于实时预测的神经网络只使用了预测节点的上一个节点的数据。

结构上，如表5所述，输入层有240个神经元，隐藏层的第一层使用了200个神经元的全连接层，隐藏层的第二层使用了200个神经元的LSTM层，隐藏层的第三层使用了抛弃值为0.2的遗忘层，隐藏层的第四层使用200个神经元的全连接层，隐藏层的最后一层使用一个神经元的全连接层。最大训练次数设置为100次，其中前75次的学习率设置为0.0005，后25次的学习率设置为0.2。

从图17中可以看到模型在第50次训练后，以后基本上使得根均方差保持在了0.025左右，同时损失值也相对降低到了最低值。前期训练使用0.0005的学习率是为了更精确找到损失函数的最小值，在后面125次训练中使用较大的学习率是为了防止梯度爆炸而陷入局部优解，错过最优解。

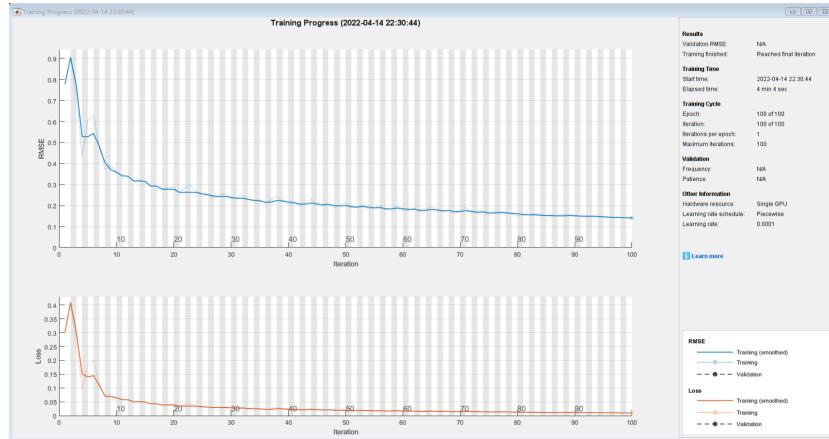


图 17: 根均方误差图和损失函数图

其中，整体的根均方差为37.431488。之所产生较大的整体误差是由于需要进行长期预测，不能使用观察值更新的方式更新神经网络的记忆，导致随着时间的增加误累积越来越大，最后直至失效。但本文只预测了72小时之内的数据，远没有到达失效的时间节点(如图18所示，越靠近0时刻预测数据越准确)。将测试集中72小时内的预测结果和真实值换算成气象局^[4]规定的空气质量指标，得到误差0.8862；基本上可以预测72小时内的数据。

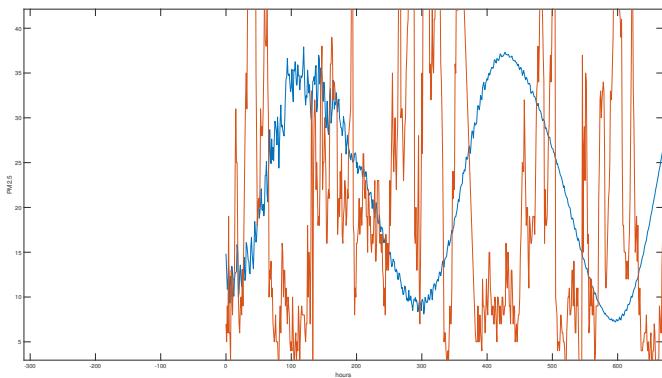


图 18: 根均方误差图和损失函数图

表6只选取了2022年4月12日中午12点后72小时预测的部分数据，并且与2022年4月12日之后72小时的数据进行对比。需要注意的是，本文写作的

时间尚未到预测节点后72小时，同时因为此时刻Open-AQ在2022年4月15日10点之前只收集到了2022年4月12日中午12点至2022年4月14日中午12点的数据，共计48个小时的数据，因此能够对比的数据只有40个。完整的对比表格表8可以在附录中找到。

时间	预测等级	实际等级
第1个小时	优	优
第2个小时	优	优
...
第71个小时	轻度污染	尚未发生
第72个小时	轻度污染	尚未发生

表 6: 24实时预测（部分）

6 北京市朝阳区空气污染治理状况及对策

北京历史上曾在北京和环北京区域的相邻省份采取过极端的减排措施，但是在"APEC"会议时期 $PM_{2.5}$ 的浓度依然高达51.5微克/立方米^[4]。大气污染具有时间关系和周期关系，应当利用周期规律进行合理的减排手段，实现逐年提高空气质量。

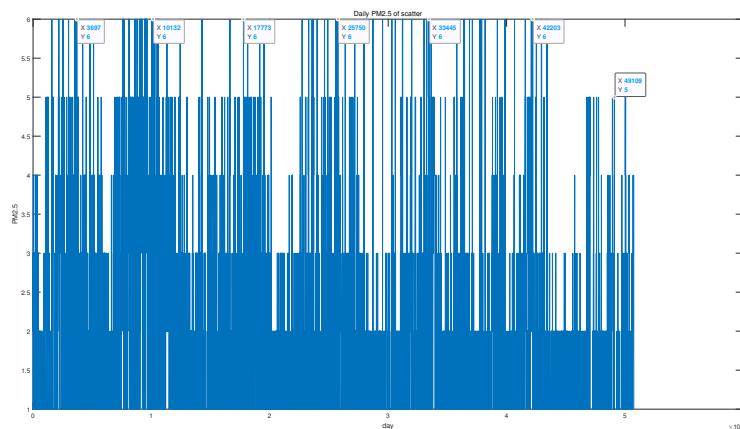


图 19: $PM_{2.5}$ 高峰周期

根据 $PM_{2.5}$ 高峰周期图可以得出， $PM_{2.5}$ 的周期基本上是以一年为一个周期，在每年从入冬到开春是空气质量最差的时期，由于供暖和更多的私家车上路，造成空气污染更为严重。应对此种现象，可以考虑采用清洁能源取暖，减少燃煤取暖；同时在冬季来临之前的三个月内提前适当限行车辆，减少尾气排放。为了遏制北京空气质量的恶化，后续应重点控制电力、热力生产企业、黑色金属、有色金属行业的废气排放量。

其中电力行业产生的废气多的原因主要是由于我国的新能源装机容量不足，截至2020年底，全国电源总装机容量超过22亿千瓦，火电水电仍为第一第二大电源，分别占比57%，17%。当前电力行业二氧化碳排放约占中国能源活动二氧化碳排放的40%。2020年全社会用电量为7.5万亿千瓦时，十三五期间全社会用电量年均增速为5.7%。利用新能源代替传统发电仍需继续研究，逐步推进。拉闸限电势在必行。对于相关部门而言，应加大重点行业排放的监督力度，引导企业做好生产后的废气净化处理，加大投入。减少火力发电，引进新能源发电技术迫在眉睫。

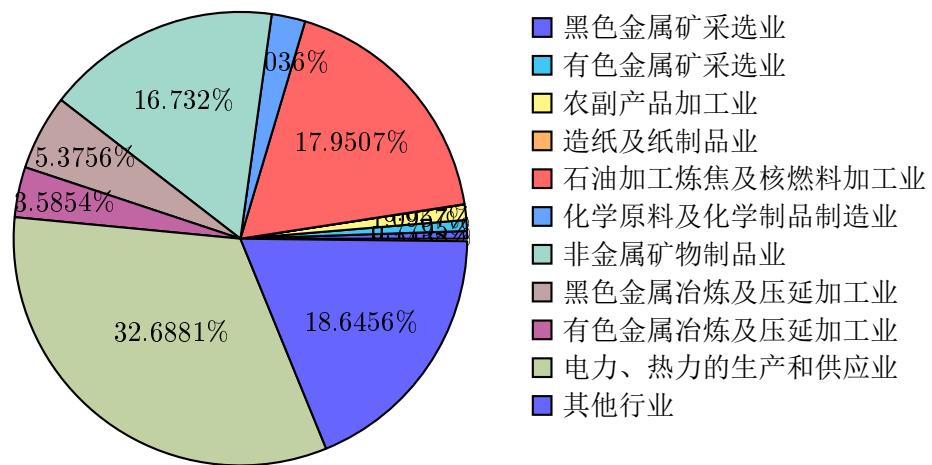


图 20: 各行用电占比

7 研究结论总结

本文以Matlab2019、SPSSPro等软件作为编程语言和分析工具。通过使用基于TOPSIS的环境评价模型，可以分析一个地区历年的环境发展变化，也可以分析多个地区同一年的环境发展情况，为环境政策的有效性提供了客观的参考的指标。本文以塞罕坝三十多年的数据进行分析，得出了每个时期塞罕坝的环境发展质量的评价水平。同时利用该模型对全国各大省、直辖市(除台湾省以及香港特别行政区和澳门特别行动区)进行了分析和排名，并绘制了全国环境质量发展热力图，得出了我国目前环北京地区，华北平原以及黄河中下游省份环境质量较差，我国越往南的地区环境质量越好的结论。

其次使用北京市朝阳区从2015年7月7日中午12点到2022年4月12日中午12点每个小时的 $PM_{2.5}$ 的含量训练了两个LSTM神经网络模型，分别用于实时预测和72小时预测北京市朝阳区今后 $PM_{2.5}$ 的含量，并分析了北京市朝阳区 $PM_{2.5}$ 出现的周期现象，提出了针对 $PM_{2.5}$ 周期治理的政策建议。

近年“绿水青山就是金山银山”政策以及“碳达峰、碳中和”目标的提出，越来越多的学者、专家将目光投入环境评价和预测中去。随着计算机技术的发展，待更多的优秀的评价预测算法的出现，可把大气环境评价预测推进到一个更高的水平。

8 参考文献

- [1]李丽. 基于数据挖掘的城市环境空气质量决策支持系统设计与实现[D].山东师范大学,2006.
- [2]马媛媛,孙世群.模糊综合评价在合肥市大气环境评价中的应用[J].环境科学与管理,2012,37(05):188-191.
- [3]金文彪,姚永杰,金哲植.基于主成分分析的大气环境预测研究[J].科教导刊(中旬刊),2016(32):148-150.DOI:10.16400/j.cnki.kjdz.2016.11.071.
- [4]北京大学统计科学中心, 北京大学光华管理学院.北京城区2010-2014年 $PM_{2.5}$ 污染状况研究[J].2015.3.
- [5]Endah Kristiani, Hao Lin, Jwu-Rong Lin, Yen-Hsun Chuang, Chin-Yin Huang and Chao-Tung Yang.5Short-Term Prediction of $PM_{2.5}$ Using LSTM Deep Learning Methods[J].Sustainability 2022, 14, 2068.
- [6]袁冲.基于熵权法的江苏省各市经济高质量发展评价分析[J].商业经济,2022(04):19-20+42.DOI:10.19905/j.cnki.syjj1982.2022.04.061.
- [7]徐政华,曹延明.基于熵权TOPSIS模型的长春市水资源承载力评价[J/OL].安全与环境学报:1-10[2022-03-27].DOI:10.13637/j.issn.1009-6094.2021.1448

A 附录

A.1 完整的预测表格

A.1.1 24小时PM_{2.5}实时预测表格

时间	预测等级	实际等级
第1小时	优	优
第2小时	优	优
第3小时	优	优
第4小时	优	优
第5小时	优	优
第6小时	优	优
第7小时	优	优
第8小时	优	优
第9小时	优	优
第10小时	优	优
第11小时	优	优
第12小时	优	优
第13小时	优	优
第14小时	优	优
第15小时	优	优
第16小时	优	优
第17小时	优	优
第18小时	优	优
第19小时	优	优
第20小时	优	优
第21小时	优	优
第22小时	优	优
第23小时	优	优
第24小时	优	优

表 7: 24实时预测

A.1.2 72小时PM_{2.5}预测表格

时间	预测等级	实际等级	时间	预测等级	实际等级
第1小时	优	优	第37小时	良	优
第2小时	优	优	第38小时	良	良
第3小时	优	优	第39小时	良	良
第4小时	优	优	第40小时	良	优
第5小时	优	优	第41小时	良	优
第6小时	优	优	第42小时	良	优
第7小时	优	优	第43小时	良	优
第8小时	优	优	第44小时	良	优
第9小时	优	优	第45小时	良	优
第10小时	优	优	第46小时	良	优
第11小时	优	优	第47小时	良	优
第12小时	优	优	第48小时	良	优
第13小时	优	优	第49小时	良	暂无数据
第14小时	优	优	第50小时	良	暂无数据
第15小时	优	优	第51小时	良	暂无数据
第16小时	优	优	第52小时	良	暂无数据
第17小时	优	优	第53小时	良	暂无数据
第18小时	优	优	第54小时	良	暂无数据
第19小时	优	优	第55小时	良	暂无数据
第20小时	优	优	第56小时	良	暂无数据
第21小时	优	优	第57小时	良	暂无数据
第22小时	优	优	第58小时	良	暂无数据
第23小时	优	优	第59小时	良	暂无数据
第24小时	优	优	第60小时	良	暂无数据
第25小时	优	优	第61小时	良	暂无数据
第26小时	优	优	第62小时	良	暂无数据
第27小时	优	优	第63小时	良	暂无数据
第28小时	优	优	第64小时	轻度污染	暂无数据
第29小时	优	优	第65小时	轻度污染	暂无数据
第30小时	优	优	第66小时	轻度污染	暂无数据
第31小时	良	优	第67小时	轻度污染	暂无数据
第32小时	良	优	第68小时	轻度污染	暂无数据
第33小时	良	优	第69小时	轻度污染	暂无数据
第34小时	良	优	第70小时	轻度污染	暂无数据
第35小时	良	优	第71小时	轻度污染	暂无数据
第36小时	良	良	第72小时	轻度污染	暂无数据

表 8: 24实时预测

A.2 相关代码

A.2.1 TOSIS综合评价代码

```
[file,path] = uigetfile({'*.xlsx','*.xls'},'File Selector');
filepath=strcat(path,file);
Data = xlsread(filepath,'sheet1');
Datasize=size(Data);
X=Data;

msgbox("导入数据成功，正在计算，请稍后");
%% 第二步：判断是否需要正向化
[n,~] = size(X);
%% 第三步：对正向化的矩阵进行标准化
Z = X ./ repmat(sum(X.*X) .^ 0.5, n, 1);
%% 让用户判断是否需要增加权重
weight = Entropy_Method(Z);
%% 第四步：计算与最大值的距离和最小值的距离，并算出得分
D_P = sum([(Z - repmat(max(Z),n,1)) .^ 2 ] .* repmat(weight,n,1) ,2) .^ 0.5; % D+ 与最大值的距离
D_N = sum([(Z - repmat(min(Z),n,1)) .^ 2 ] .* repmat(weight,n,1) ,2) .^ 0.5; % D- 与最小值的距离
S = D_N ./ (D_P+D_N); % 未归一化的得分
% disp('最后的得分为：')
stand_S = S / sum(S);
% stand_S = 1./stand_S;

pathname = '.';
filename = 'stand_S.mat';
filepath2=strcat(pathname,filename);
save(filepath2,'stand_S');

[~,index] = sort(stand_S , 'descend');

dates=Data(:,1);
result = [dates index stand_S];
result = sortrows(result,2);
A=result(:,1);
B=result(:,2);
C=result(:,3);

plot(app.UIAxes,dates,stand_S);

R=[A B C];
app.Output.Data=R;
```

```
[file,path] = uigetfile({'*.xlsx','*.xls'},'File Selector');
filepath=strcat(path,file);
Data = readtable(filepath);
```

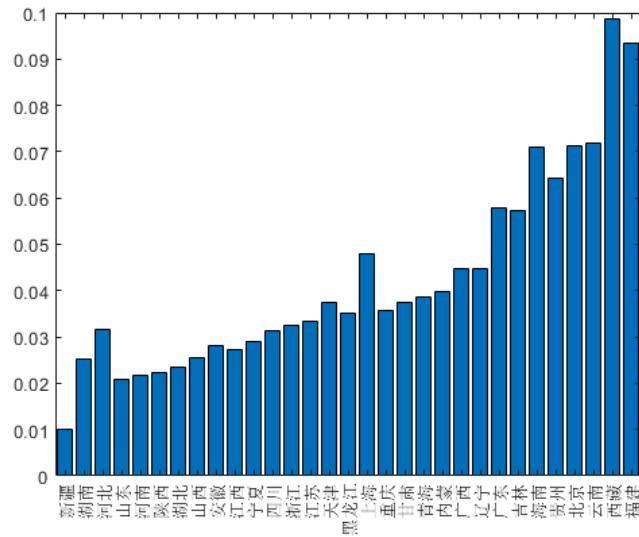
警告: 表变量名称已修改为有效的 MATLAB 标识符。原始名称保存在 `VariableDescriptions` 属性中。

```
Place = Data(:,1);
Place =table2array(Place);
LY = Place;
Y = categorical(Place);
Y = reordercats(Y,Place);
Datasize=size(Data);
Data =Data(:,2:Datasize(2));
Data = table2array(Data);
X=Data;
msgbox("导入数据成功, 正在计算, 请稍后");
```

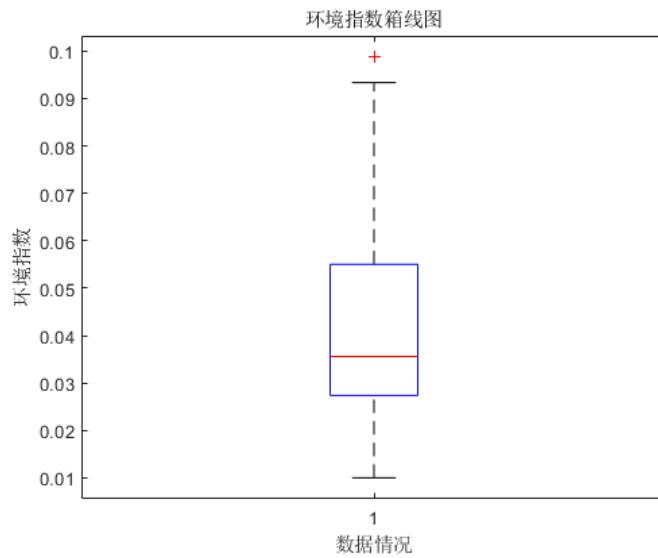


```
%% 第二步: 判断是否需要正向化
[n,~] = size(X);
X = Min2Max(X);
%% 第三步: 对正向化的矩阵进行标准化
Z = X ./ repmat(sum(X.*X) .^ 0.5, n, 1);
%% 让用户判断是否需要增加权重
weight = Entropy_Method(Z);
%% 第四步: 计算与最大值的距离和最小值的距离, 并算出得分
D_P = sum([(Z - repmat(max(Z),n,1)) .^ 2] .* repmat(weight,n,1) ,2) .^ 0.5; % D+
D_N = sum([(Z - repmat(min(Z),n,1)) .^ 2] .* repmat(weight,n,1) ,2) .^ 0.5; % D-
S = D_N ./ (D_P+D_N); % 未归一化的得分
% disp('最后的得分为: ')
stand_S = S / sum(S)+0.01;

figure;
bar(Y,stand_S);
```



```
boxplot(stand_S);
title('环境指数箱线图')
xlabel('数据情况')
ylabel('环境指数')
```



```
[~,index] = sort(stand_S);
meanstand_s = mean(stand_S);
mean.Value = meanstand_s;

lownumber = 0;
for i = 1:n
    if stand_S(i)<=meanstand_s
        lownumber =lownumber + 1;
    end
end

Value = num2str(lownumber);

order = 1:n;
order = flip(order);
order =order';
order = num2cell(order);
nX=stand_S(index);
nX = num2cell(nX);
nY=lY(index);
nY = cell(nY);

XY = [nY nX order];
% XY = str2cell(XY);
%
% [~,index] = sort(stand_S);
```

```
Data=XY;  
%  
% [~,index] = sort(stand_S);
```

A.2.2 热力图生成代码

```

clc
clear
close all

sheng=shaperead('bou2_4p.shp', 'UseGeoCoords', true); % 省
load F:\matlab\bin\App_Desgin\china_map1\chinese_name.mat % 省, 省会, 主要城市的正确中文
for i=1:length(sheng)
    sheng(i).NAME=sheng_chinese_name{i}; % 纠正中文显示错误
end
unique(sheng_chinese_name) % 含有34个省(直辖市)的数据
length(sheng) % 共分为925个区块

d=importdata('F:\matlab\bin\App_Desgin\EnvironmentRange.txt');
data=d.data; % 人口数目
textdata=d.textdata; % 相对应的省的名称

k=128;
mycolormap=summer(k);
% 生成不同区域按大小的颜色, 按照人口数目多少分别指定不同的颜色
% 人口越多, 颜色越突出
geoname={sheng.NAME};
max_data = max(data);
n=length(data);
mysymbolspec=cell(1,n);% 预定义变量可以加快处理速度
for i=1:n
    count=data(i);
    mycoloridx=floor( k * count / max_data );
    mycoloridx(mycoloridx<1)=1;
    myprovince=textdata{i};
    geoidx=strmatch(myprovince, geoname);
    if numel(geoidx) > 0
        province_name=geoname( geoidx(1) );
        mysymbolspec{i} = {'NAME', char(province_name), 'FaceColor', mycolormap( mycoloridx, :)};
    end
end

figure
ax=worldmap('china'); % 使用worldmap的坐标轴作图
setm(ax,'grid','off') % 关闭grid
setm(ax,'frame','off') % 关闭边框
setm(ax,'parallellabel','off') % 关闭坐标轴标记
setm(ax,'meridianlabel','off') % 关闭坐标轴标记

% 最关键的两个语句
symbols=makesymbolspec('Polygon',{'default','FaceColor',[0.9 0.9 0.8],...
    'LineStyle','--','LineWidth',0.2,...%
    'EdgeColor',[0.8 0.9 0.9]},...
    mysymbolspec{:}...
);
geoshow(sheng,'SymbolSpec',symbols); % 此处用mapshow投影会不正确

% 在图像右侧显示bar
colormap(summer(k))

```

A.2.3 $PM_{2.5}$ 实时预测LSTM神经网络代码

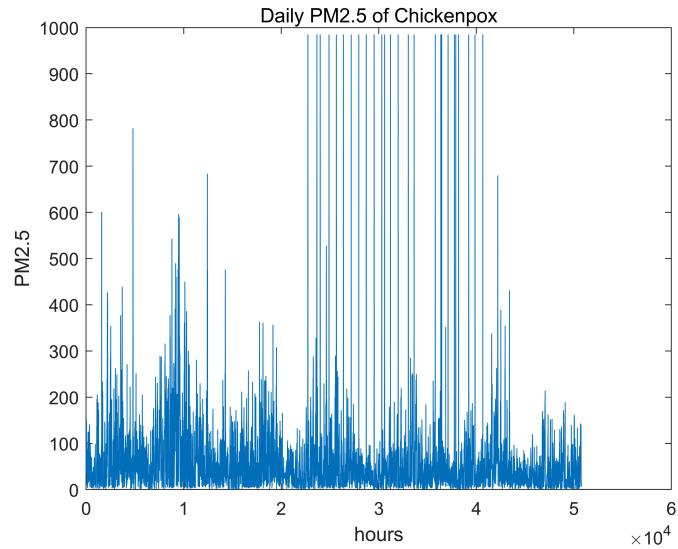
```
clc
clear
close all

load F:\matlab\bin\App_Desgin\beijingData.mat

data = beijingdatas;
data = data';
PM25 = flip(data);
m = numel(PM25);

for j=1:m
    if PM25(j)<0 || PM25(j)> 1500
        PM25(j)=nan;
    end
end
% PM25(30488)=nan;
PM25 = fillmissing(PM25,"linear");
% PM25 = fillmissing(PM25,"linear");
% PM25 = rmoutliers(PM25,'median');

% PM10 = data(2,:);
% PM10 = fillmissing(PM10,"linear");
% PM10 = rmoutliers(PM10,'quartiles');
%
% AQI = data(3,:);
% AQI = fillmissing(AQI,'linear');
% AQI = rmoutliers(AQI,'quartiles');
PM = PM25;
PM = rmoutliers(PM,'quartiles');
figure
m = numel(PM25);
y = 1:m;
plot(y,PM25)
xlabel("hours")
ylabel("PM2.5")
title("Daily PM2.5 of Chickenpox")
```



```

x1 = 0;
x2 = 0;
x3 = 0;
x4 = 0;
x5 = 0;
x6 = 0;
m = numel(PM25);
for j=1:m
    if PM25(j)<=35
        PM25(j)=1; % 优
        x1 = x1+1;
    end
    if 35<PM25(j) && PM25(j)<=75 % 良
        PM25(j)=2;
        x2 = x2+1;
    end
    if 75<PM25(j) && PM25(j)<=115 % 轻度污染
        PM25(j)=3;
        x3 = x3+1;
    end
    if 115<PM25(j) && PM25(j)<=150 % 中度污染
        PM25(j)=4;
        x4 = x4+1;
    end
    if 150<PM25(j) && PM25(j)<=250 % 重度污染
        PM25(j)=5;
        x5 = x5+1;
    end

```

```
if 250<PM25(j) && PM25(j)<=350 % 严重污染
    PM25(j)=6;
    x6 = x6+1;
end
if 350<PM25(j) % 严重污染
    PM25(j)=6;
    x6 = x6+1;
end
end

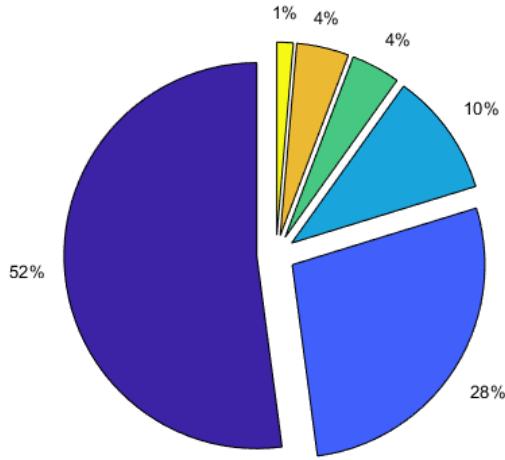
m

m =
50764

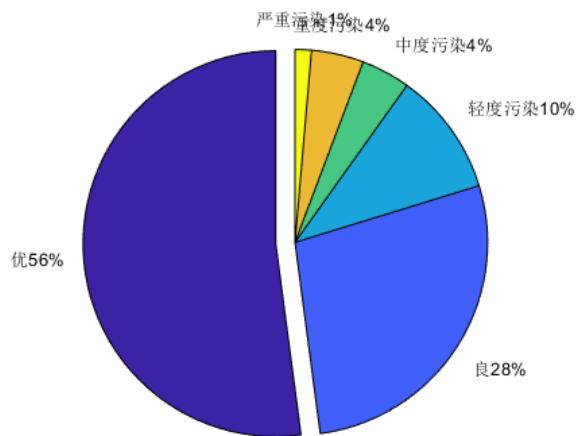
x = [x1/m x2/m x3/m x4/m x5/m x6/m]

x =
1×6
0.520959735245450 0.276554251044047 0.103774328264124 0.041151209518556 ...

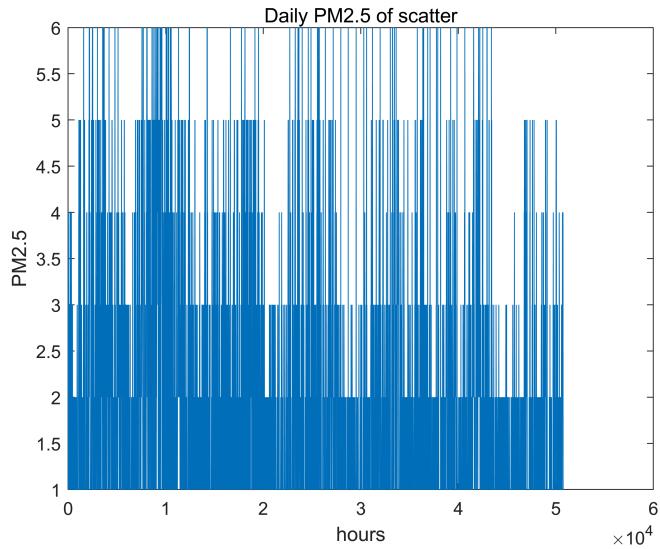
labels = {'优56%', '良28%', '轻度污染10%', '中度污染4%', '重度污染4%', '严重污染1%'};
pie(x, '%.3f%');
```



```
explode = [1 0 0 0 0 0];
pie(x,explode,labels)
```



```
y = 1:m;
figure
plot(y,PM25)
xlabel("hours")
ylabel("PM2.5")
title("Daily PM2.5 of scatter")
```



```
% LABEL = PM25;
% index = 1;
% DATA = zeros(index,m-index+1);
% for i=1:index
%     DATA(i,:) = PM(i:end-index+i);
% end
% DATAlabel = PM25(index+1:end);
DATA = PM;

[c,1] = size(DATA);
numTimeStepsTrain = floor(0.95*c);

dataTrain = DATA(1:numTimeStepsTrain+1);
dataTrain = dataTrain;

dataTest = DATA(numTimeStepsTrain+1:end);
dataTest = dataTest;
```

```
mu = mean(dataTrain);
sig = std(dataTrain);
dataTrainStandardized = (dataTrain - mu) / sig;
```

```
XTrain = dataTrainStandardized(1:end-1);
YTrain = dataTrainStandardized(2:end);
```

```
numFeatures = 1;
numResponses = 1;
numHiddenUnits1 = 200;
% numHiddenUnits2 = 200;

layers = [ ...
    sequenceInputLayer(numFeatures)
    fullyConnectedLayer(200)
    lstmLayer(numHiddenUnits1)
%    lstmLayer(numHiddenUnits2)
    dropoutLayer(0.2)
    fullyConnectedLayer(200)
    fullyConnectedLayer(numResponses)
    regressionLayer];

options = trainingOptions('adam', ...
    'MaxEpochs',250, ...
    'GradientThreshold',1, ...
    'InitialLearnRate',0.0005, ...
    'LearnRateSchedule','piecewise', ...
    'LearnRateDropPeriod',125, ...
    'LearnRateDropFactor',0.2, ...
    'Verbose',0, ...
    'Plots','training-progress');
```

```
net = trainNetwork(XTrain,YTrain,layers,options);
```



```
dataTestStandardized = (dataTest - mu) / sig;
XTest = dataTestStandardized(1:end-1);

net = predictAndUpdateState(net,XTrain);
[net,YPred] = predictAndUpdateState(net,YTrain(end));

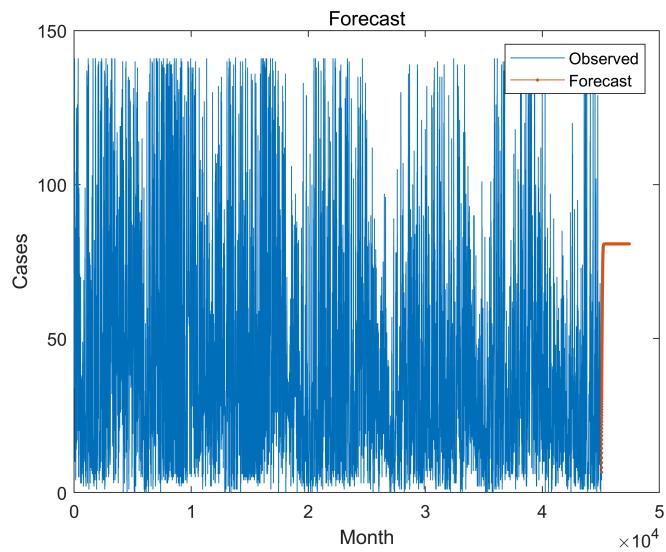
numTimeStepsTest = numel(XTest);
for i = 2:numTimeStepsTest
    [net,YPred(:,i)] = predictAndUpdateState(net,YPred(:,i-1),'ExecutionEnvironment','cpu');
end

YPred = sig*YPred + mu;

YTest = dataTest(2:end);
rmse = sqrt(mean((YPred-YTest).^2))

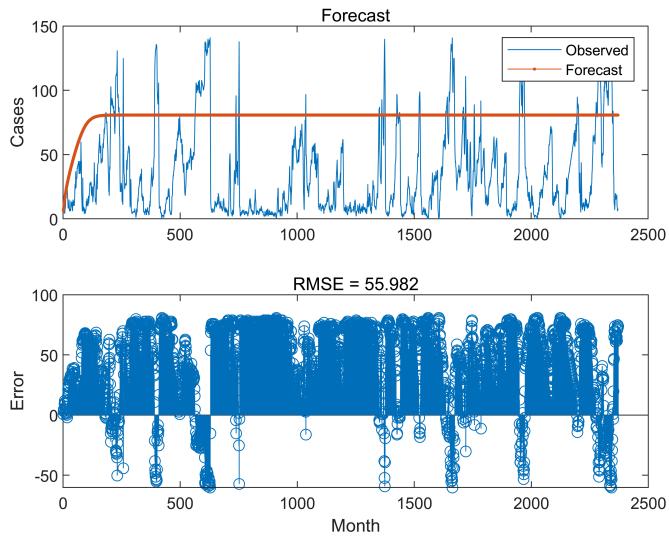
rmse = single
55.9820366

figure
plot(dataTrain(1:end-1))
hold on
idx = numTimeStepsTrain:(numTimeStepsTrain+numTimeStepsTest);
plot(idx,[data(numTimeStepsTrain) YPred],'.-')
hold off
xlabel("Month")
ylabel("Cases")
title("Forecast")
legend(["Observed" "Forecast"])
```



```
figure
subplot(2,1,1)
plot(YTest)
hold on
plot(YPred, '.-')
hold off
legend(["Observed" "Forecast"])
ylabel("Cases")
title("Forecast")

subplot(2,1,2)
stem(YPred - YTest)
xlabel("Month")
ylabel("Error")
title("RMSE = " + rmse)
```



```

net = resetState(net);
net = predictAndUpdateState(net,XTrain);

YPred = [];
numTimeStepsTest = numel(XTest);
for i = 1:numTimeStepsTest
    [net,YPred(:,i)] = predictAndUpdateState(net,XTest(:,i),'ExecutionEnvironment','cpu');
end

YPred = sig*YPred + mu;

rmse = sqrt(mean((YPred-YTest).^2))

rmse =
7.543113068488156

figure
subplot(2,1,1)
plot(YTest)
hold on
plot(YPred,'.-')
hold off
legend(["Observed" "Predicted"])

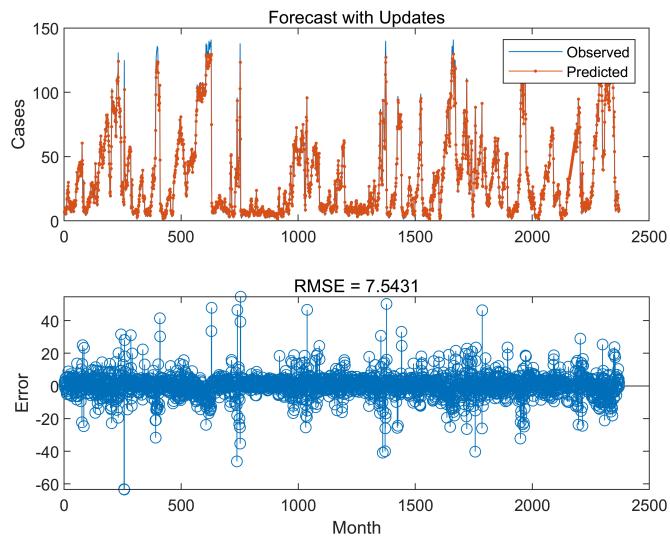
```

```

ylabel("Cases")
title("Forecast with Updates")

subplot(2,1,2)
stem(YPred - YTest)
xlabel("Month")
ylabel("Error")
title("RMSE = " + rmse)

```



```

m = numel(YPred);
YPredLabel = YPred;
for j=1:m
    if YPredLabel(j)<=35
        YPredLabel(j)=1; % 优
    end
    if 35<YPredLabel(j) && YPredLabel(j)<=75 % 良
        YPredLabel(j)=2;
    end
    if 75<YPredLabel(j) && YPredLabel(j)<=115 % 轻度污染
        YPredLabel(j)=3;
    end
    if 115<YPredLabel(j) && YPredLabel(j)<=150 % 中度污染
        YPredLabel(j)=4;
    end
    if 150<YPredLabel(j) && YPredLabel(j)<=250 % 重度污染
        YPredLabel(j)=5;
    end

```

```

if 250<YPredLabel(j) && YPredLabel(j)<=350 % 严重污染
    YPredLabel(j)=6;
end
if 350<YPredLabel(j) % 严重污染
    YPredLabel(j)=6;
end
end

YTextLabel = YTest;
for j=1:m
    if YTextLabel(j)<=35
        YTextLabel(j)=1; % 优
    end
    if 35<YTextLabel(j) && YTextLabel(j)<=75 % 良
        YTextLabel(j)=2;
    end
    if 75<YTextLabel(j) && YTextLabel(j)<=115 % 轻度污染
        YTextLabel(j)=3;
    end
    if 115<YTextLabel(j) && YTextLabel(j)<=150 % 中度污染
        YTextLabel(j)=4;
    end
    if 150<YTextLabel(j) && YTextLabel(j)<=250 % 重度污染
        YTextLabel(j)=5;
    end
    if 250<YTextLabel(j) && YTextLabel(j)<=350 % 严重污染
        YTextLabel(j)=6;
    end
    if 350<YTextLabel(j) % 严重污染
        YTextLabel(j)=6;
    end
end
rmse = sqrt(mean((YPredLabel-YTextLabel).^2))

```

rmse =
0.290435180104375

```

net = predictAndUpdateState(net,XTest);
[net,YPred] = predictAndUpdateState(net,XTest(end));

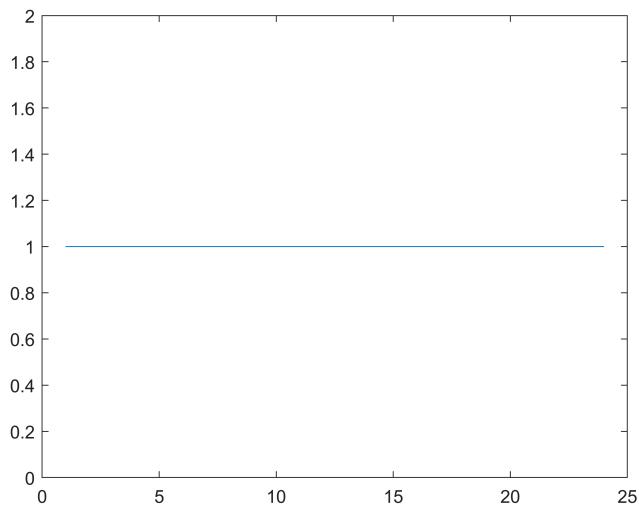
for i = 2:24
    [net,YPred(:,i)] = predictAndUpdateState(net,YPred(:,i-1), 'ExecutionEnvironment', 'cpu');
end

for j=1:24
    if YPred(j)<=35
        YPred(j)=1; % 优
    end
    if 35<YPred(j) && YPred(j)<=75 % 良
        YPred(j)=2;
    end
    if 75<YPred(j) && YPred(j)<=115 % 轻度污染

```

```
    YTextLabel(j)=3;
end
if 115<YPred(j) && YPred(j)<=150 % 中度污染
    YPred(j)=4;
end
if 150<YPred(j) && YPred(j)<=250 % 重度污染
    YPred(j)=5;
end
if 250<YPred(j) && YPred(j)<=350 % 严重污染
    YPred(j)=6;
end
if 350<YPred(j) % 极重污染
    YPred(j)=6;
end
end

figure
plot(YPred)
```



```
save ../bin/App_Desgin/BeijingNet.mat net
```

```
save .\App_Desgin\Beijing24hourseP.mat YPred
save .\App_Desgin\Beijing24hourseTruth.mat YTextLabel
```

A.2.4 $PM_{2.5}$ 72小时预测LSTM神经网络代码

```
clc
clear
close all

load F:\matlab\bin\App_Desgin\beijingData.mat

data = beijingdatas;
data = data';
PM25 = flip(data);
m = numel(PM25);

for j=1:m
    if PM25(j)<0 || PM25(j)> 1500
        PM25(j)=nan;
    end
end

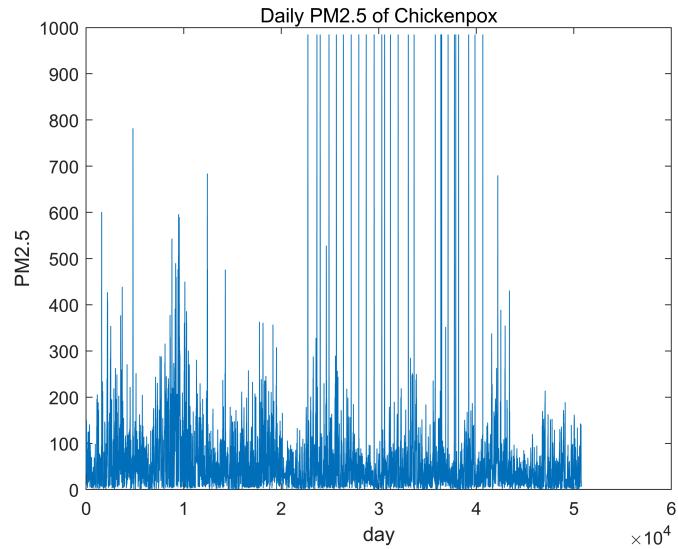
PM25 = fillmissing(PM25,"linear");

PM = PM25;
PM = rmoutliers(PM,'quartiles');
PMM = [];

numHouser = 240;
for i = 1:numHouser
    PMM(:,i) = PM(i:end-numHouser+i);
end
PMM = PMM';
```



```
figure
m = numel(PM25);
y = 1:m;
plot(y,PM25)
xlabel("day")
ylabel("PM2.5")
title("Daily PM2.5 of Chickenpox")
```



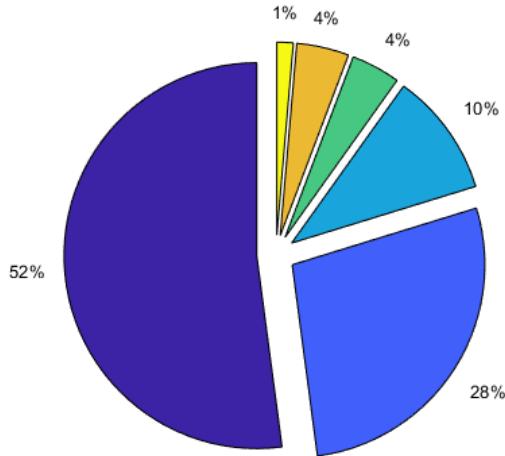
```

x1 = 0;
x2 = 0;
x3 = 0;
x4 = 0;
x5 = 0;
x6 = 0;
m = numel(PM25);
for j=1:m
    if PM25(j)<=35
        PM25(j)=1; % 优
        x1 = x1+1;
    end
    if 35<PM25(j) && PM25(j)<=75 % 良
        PM25(j)=2;
        x2 = x2+1;
    end
    if 75<PM25(j) && PM25(j)<=115 % 轻度污染
        PM25(j)=3;
        x3 = x3+1;
    end
    if 115<PM25(j) && PM25(j)<=150 % 中度污染
        PM25(j)=4;
        x4 = x4+1;
    end
    if 150<PM25(j) && PM25(j)<=250 % 重度污染
        PM25(j)=5;
        x5 = x5+1;
    end
end

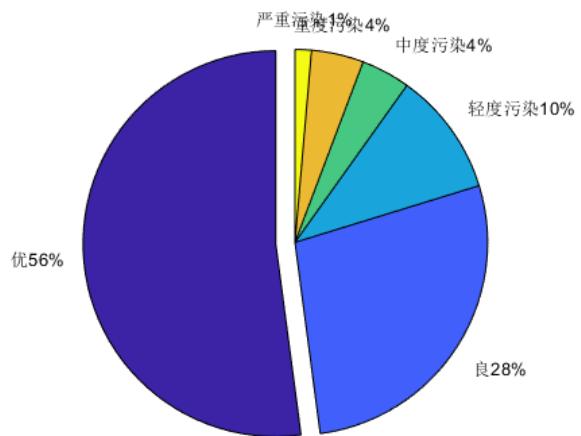
```

```
if 250<PM25(j) && PM25(j)<=350 % 严重污染
    PM25(j)=6;
    x6 = x6+1;
end
if 350<PM25(j) % 严重污染
    PM25(j)=6;
    x6 = x6+1;
end
end
```

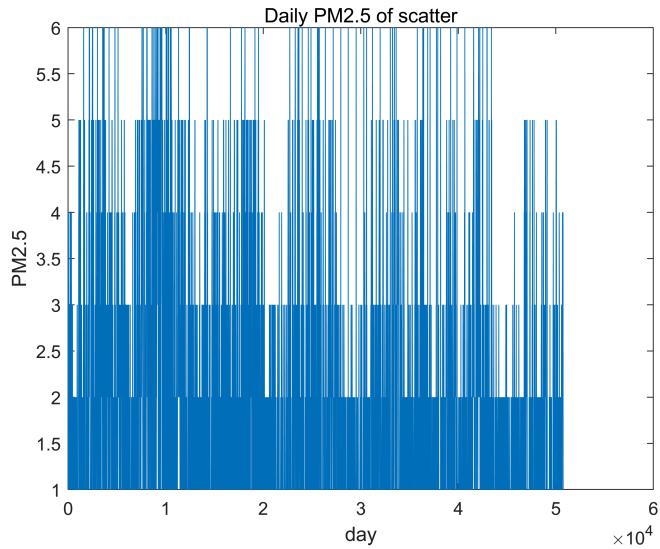
```
x = [x1/m x2/m x3/m x4/m x5/m x6/m]
x = 1×6
0.520959735245450 0.276554251044047 0.103774328264124 0.041151209518556 ...
labels = {'优56%', '良28%', '轻度污染10%', '中度污染4%', '重度污染4%', '严重污染1%'};
pie(x, '%.3f%');
```



```
explode = [1 0 0 0 0 0];
pie(x,explode,labels)
```



```
y = 1:m;
figure
plot(y,PM25)
xlabel("day")
ylabel("PM2.5")
title("Daily PM2.5 of scatter")
```



```
% LABEL = PM25;
% index = 1;
% DATA = zeros(index,m-index+1);
% for i=1:index
%     DATA(i,:) = PM(i:end-index+i);
% end
% DATAlabel = PM25(index+1:end);
DATA = PMM;

[c,1] = size(DATA);
numTimeStepsTrain = floor(0.90*c);

for i = 1:numHouser
    dataTrain(i,:) = DATA(i,1:numTimeStepsTrain+1);
end

for i = 1:numHouser
    dataTest(i,:) = DATA(i,numTimeStepsTrain+1:end);
end
```

```
[~,Xrule] = mapminmax(dataTrain);
```

```
dataTrainStandardized = mapminmax('apply',dataTrain,Xrule);
XTrain = dataTrainStandardized(:,1:end-numHouser+1);
```

```
YTrain = dataTrain(1,numHouser:end);

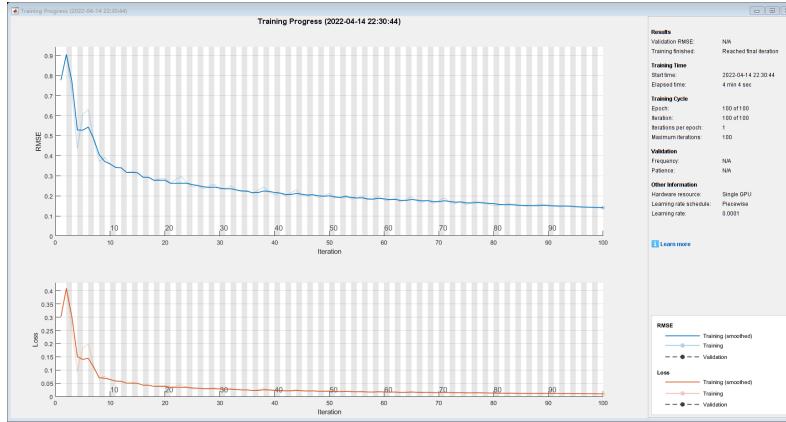
[~,Yrule] = mapminmax(YTrain);
YTrain = mapminmax('apply',YTrain,Yrule);
```

```
numFeatures = numHouser;
numResponses = 1;
numHiddenUnits1 = 200;
% numHiddenUnits2 = 200;

layers = [ ...
    sequenceInputLayer(numFeatures)
    fullyConnectedLayer(200)
    lstmLayer(numHiddenUnits1)
%    lstmLayer(numHiddenUnits2)
    dropoutLayer(0.2)
    fullyConnectedLayer(200)
    fullyConnectedLayer(numResponses)
    regressionLayer];

options = trainingOptions('adam', ...
    'MaxEpochs',100, ...
    'GradientThreshold',1, ...
    'InitialLearnRate',0.0005, ...
    'LearnRateSchedule','piecewise', ...
    'LearnRateDropPeriod',75, ...
    'LearnRateDropFactor',0.2, ...
    'Verbose',0, ...
    'Plots','training-progress');
```

```
net = trainNetwork(XTrain,YTrain,layers,options);
```



```
load ..\bin\App_Desgin\BeijingNet2.mat;
```

```
dataTestStandardized = mapminmax('apply',dataTest,Xrule);
XTest = dataTestStandardized(:,1:end-numHouser);

Ytest = dataTest(1,numHouser:end-1);

Ytest = mapminmax('apply',Ytest,Yrule);
```

```
net = predictAndUpdateState(net,XTrain);
[net,YPred] = predictAndUpdateState(net,XTrain(:,end));

mid = XTrain(:,end);
for i = 1:numHouser-1
    mid(i) = mid(i+1);
end
```

```
mid(end) = YPred;
YPred = mid;
[c,l] = size(XTest);

for i = 1:l
    [net,rs(i)] = predictAndUpdateState(net,YPred(:,1),'ExecutionEnvironment','cpu');
    for j = 1:numHouser-1
        YPred(j)=YPred(j+1);
    end
```

```

YPred(end) = rs(i);
end

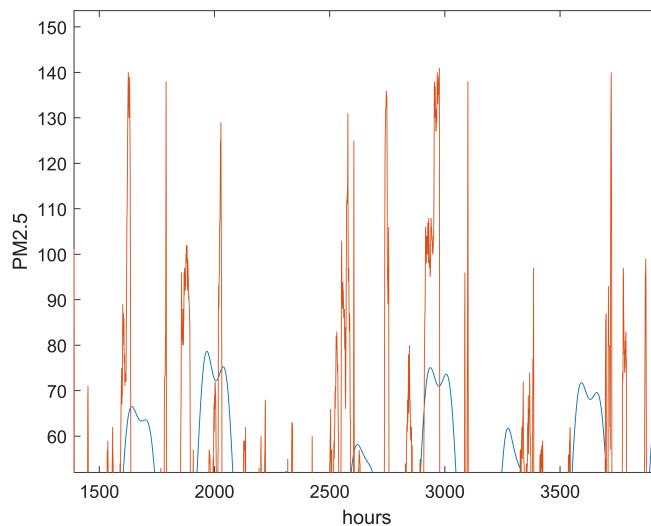
endrs = mapminmax('reverse',rs,Yrule);
Ytest = mapminmax('reverse',Ytest,Yrule);
rmse = sqrt(mean((endrs-Ytest).^2))

rmse = single
37.4313164

index = 0:numel(endrs)-1;

plot(index,endrs);hold on;
plot(index,Ytest);hold off

```



```

xlabel('hours')
ylabel('PM2.5')

```

```

m = numel(endrs);
for j=1:m
    if endrs(j)<=35
        endrs(j)=1; % 优
    else

```

```
end
if 35<endrs(j) && endrs(j)<=75 % 良
    endrs(j)=2;
end
if 75<endrs(j) && endrs(j)<=115 % 轻度污染
    endrs(j)=3;
end
if 115<endrs(j) && endrs(j)<=150 % 中度污染
    endrs(j)=4;
end
if 150<endrs(j) && endrs(j)<=250 % 重度污染
    endrs(j)=5;
end
if 250<endrs(j) && endrs(j)<=350 % 严重污染
    endrs(j)=6;
end
if 350<endrs(j) % 严重污染
    endrs(j)=6;
end
end

m = numel(Ytest);
for j=1:m
    if Ytest(j)<=35
        Ytest(j)=1; % 优
    end
    if 35<Ytest(j) && Ytest(j)<=75 % 良
        Ytest(j)=2;
    end
    if 75<Ytest(j) && Ytest(j)<=115 % 轻度污染
        Ytest(j)=3;
    end
    if 115<Ytest(j) && Ytest(j)<=150 % 中度污染
        Ytest(j)=4;
    end
    if 150<Ytest(j) && Ytest(j)<=250 % 重度污染
        Ytest(j)=5;
    end
    if 250<Ytest(j) && Ytest(j)<=350 % 严重污染
        Ytest(j)=6;
    end
    if 350<Ytest(j) % 严重污染
        Ytest(j)=6;
    end
end
Pre72 = endrs(1:73);
Test72 = Ytest(1:73);
PreLabelloss = sqrt(mean((Pre72-Test72).^2))
figure
plot(Pre72)
hold on;

PreLabelloss = single
0.8862793
```

```
plot(Test72)
```

预测

```
PreData = DATA(:,end-numFeatures:end)
```

```
PreData = 240x241
102 ×
0.520000000000000 0.510000000000000 0.280000000000000 0.070000000000000 ...
0.510000000000000 0.280000000000000 0.070000000000000 0.040000000000000
0.280000000000000 0.070000000000000 0.040000000000000 0.030000000000000
0.070000000000000 0.040000000000000 0.030000000000000 0.020000000000000
0.040000000000000 0.030000000000000 0.020000000000000 0.070000000000000
0.030000000000000 0.020000000000000 0.070000000000000 0.070000000000000
0.020000000000000 0.070000000000000 0.070000000000000 0.080000000000000
0.070000000000000 0.070000000000000 0.080000000000000 0.060000000000000
0.070000000000000 0.080000000000000 0.060000000000000 0.050000000000000
0.080000000000000 0.060000000000000 0.050000000000000 0.070000000000000
:
```

```
dataPreStandardized = mapminmax('apply',PreData,Xrule);
PreData = dataPreStandardized;
```

```
net = predictAndUpdateState(net,XTest);
```

```
for i = 1:72
    [net,prs(i)] = predictAndUpdateState(net,PreData(:,i), 'ExecutionEnvironment', 'cpu');
end
```

```
pendrs = mapminmax('reverse',prs,Yrule);
```

```
m = numel(pendrs);
for j=1:m
    if pendrs(j)<=35
        pendrs(j)=1; % 优
    end
    if 35<pendrs(j) && pendrs(j)<=75 % 良
        pendrs(j)=2;
    end
    if 75<pendrs(j) && pendrs(j)<=115 % 轻度污染
        pendrs(j)=3;
    end
    if 115<pendrs(j) && pendrs(j)<=150 % 中度污染
        pendrs(j)=4;
    end
    if 150<pendrs(j) && pendrs(j)<=250 % 重度污染
        pendrs(j)=5;
    end
    if 250<pendrs(j) && pendrs(j)<=350 % 严重污染
        pendrs(j)=6;
    end
```

```
end
if 350<pendrs(j) % 严重污染
    pendrs(j)=6;
end
end

save ..\bin\App_Desgin\BeijingNet2.mat net

save ..\bin\App_Desgin\PendRs.mat pendrs

save ..\bin\App_Desgin\D1215.mat D1215

m = numel(D1215);
for j=1:m
    if D1215(j)<=35
        D1215(j)=1; % 优
    end
    if 35<D1215(j) && D1215(j)<=75 % 良
        D1215(j)=2;
    end
    if 75<D1215(j) && D1215(j)<=115 % 轻度污染
        D1215(j)=3;
    end
    if 115<D1215(j) && D1215(j)<=150 % 中度污染
        D1215(j)=4;
    end
    if 150<D1215(j) && D1215(j)<=250 % 重度污染
        D1215(j)=5;
    end
    if 250<D1215(j) && D1215(j)<=350 % 严重污染
        D1215(j)=6;
    end
end
```