目录

表格与插图清单

摘要

天津市位于中国北方沿海地区，具有典型的渤海湾城市气候并且它是中国重要的经济中心和人口密集区之一，所以以它为空气质量的时空分布分析和预测的研究对象具有重要意义。本文利用天津市 2014-2024 年的空气质量指数月统计数据，建立基于 LSTM(长短期记忆网络)的随机森林优化模型，来预测未来的空气质量；同时收集了天津市的边界数据和 23 个监测点的地理位置进行 kriging 插值法进行天津市空间上的大气质量预测。并对得到的结果进行分析,也提出了针对性的建议。

首先本文先对收集到的数据进行预处理，通过查阅论文发现有 23 个特征会影响空气质量。为了选取对空气质量具有显著影响和共线性最小的特征,我们使用灰色关联度分析和逐步回归法,最终选取了 NO_2，SO_2，$PM_{2.5}$，PM_{10}，CO，降水量，平均风速，最高气温这八种特征。

其次，我们使用收集到的数据构建多元有序逻辑回归模型对天津市的大气质量进行分类预测来判断天津市的空气质量等级。但发现该模型的预测分类的准确度为 77%并不是很理想,所以我们选择直接预测 AQI 的值(空气质量指数),使用随机森林、XGBoost、LightGBM 三种算法进行预测, 评估了模型的性能和准确度以后发现随机森林拟合效果最好,同时观察数据发现天津市的空气质量指数月统计数据发现其在时间序列上存在季节性特征,每年的 10 月份至次年的 3 月份的 AQI 值会明显增大,所以引入 LSTM（长短期记忆网络）对随机森林算法进行优化。

然后，使用 kriging 插值法来预测天津市空间上的大气质量,并使用 ArcGIS 绘制了空气污染物浓度分布图，得到了 $PM_{2.5}$ 和 PM_{10} 的浓度呈现"西高东低"的趋势。而 SO_2 和 CO 的分布格局变化较大等结果。

最后，我们综合以上方法得到的天津市大气质量的时空分布的预测结果，为决策者提供针对性的改善建议。这对于天津市及其他相似城市的大气质量改善具有重要意义。

本研究的创新点在于使用了 LSTM 和随机森林相结合的优化算法，使得得到的预测结果更符合实际；并且使用了 Kriging 插值法来拟合空气质量的空间分

布特征，使得研究更加完整，全面。

关键词：多元有序逻辑回归；随机森林和 LSTM 优化算法；Kriging 插值法

一、绪论

（一）研究背景及意义

天津市作为中国北方的重要城市，面临着严重的大气污染问题。随着城市化进程和工业化的快速发展，大量的工业排放、机动车尾气、燃煤和扬尘等因素导致了大气污染的严重积累。根据《天津市生态环境质量报告（2020）》的数据显示，天津市 $PM_{2.5}$ 年平均浓度超过国家空气质量标准限值，空气质量状况较差。这对居民的健康和环境的可持续发展带来了严重威胁。

此外，天津市大气污染研究的意义还体现在可持续发展方面。大气污染不仅威胁人类健康，还对生态环境和社会经济发展产生负面影响。空气质量的下降会影响旅游业、经济发展和外来投资等方面，阻碍城市的可持续发展。通过深入研究天津市大气污染的形成机制和影响因素，可以为制定绿色发展战略和环境保护政策提供科学依据。

最后，天津市大气污染的研究对于全国范围内的大气污染治理具有示范和引领作用。天津作为人口众多、经济发达的城市，其大气污染治理经验和技术应用可供其他城市借鉴和推广。通过研究天津市大气污染问题，可以为其他地区提供重要的参考和指导，促进全国大气污染治理工作的进展。

（二）研究现状

近年来，生态环境部门以及众多学者基于现阶段公开数据对大气质量预测模型进行了大量的研究，但大多模型都面临将时空规律分开进行研究、影响因素考虑片面化、预测精度较低等问题。经查阅相关研究资料后，我们发现如今的大气质量预测研究模型主要分为机理性大气质量预测模型和非机理性大气质量预测模型。前者能够考虑更多的物理和化学机制，但需要较多的输入数据和计算资源。后者通过数据分析和统计建模，能够快速预测大气质量，但对于复杂的污染物传输和化学反应过程的理解相对较少。后者适用范围广，并且建模和预测过程相对简单，不需要大量的专业知识和复杂的参数设置。这使得非专业人员也能够使用这些模型进行大气质量预测，从而在实践中提供快速、有效的决策支持。常用的非机理性大气质量预测模型和方法有灰色系统预测模型[1][2][3]、时间序列预测模型[4]、最优化权值组合法[5]等，这些模型和方法已经在大气质量预测领域得到广泛应用。

二、数据描述及数据预处理

（一）研究区域概况

天津市位于中国的北方沿海地区，东临渤海，北濒黄海。它紧邻北京市，距离北京市约 120 公里。天津市地处华北平原，地势相对平坦。市区内有海河流经，形成了独特的城市景观。

根据 2020 年的统计数据，天津市的人口约为 15.6 万人。天津市是中国人口密度较高的城市之一，人口集中在市区和周边地区。由于地理位置的优势和经济发展的吸引力，天津市吸引了大量的人口流入。

天津市是中国重要的工业中心之一，拥有发达的制造业和重工业。天津市的化工产业较为发达，涵盖了石油化工、化学制品生产等领域。化工生产会排放大量的有害气体和颗粒物，如挥发性有机物、硫化物、氮氧化物等，对大气环境造成污染和风险。天津市的燃煤电厂是主要的能源供应来源之一。燃煤电厂排放二氧化硫、氮氧化物、颗粒物等大气污染物，对空气质量产生显著影响，尤其是在燃煤过程中未经有效净化处理的情况下。汽车制造业和交通运输业是天津市的重要工业部门。汽车尾气排放是大气污染的重要来源，包括颗粒物、氮氧化物和挥发性有机物。交通拥堵也会导致排放物积累，进一步影响空气质量。

天津市政府在工业发展和环境保护方面采取了一系列措施来减少工业项目对大气环境的影响。这包括加强排放控制和净化设施的建设、推广清洁生产技术、限制高污染和高能耗产业的发展等。

（二）数据来源

本文所使用的的数据为从由天津市统计局提供的每一年的统计年鉴中的天津市各月份气象资料以及天津市气象局提供的各区气象资料直接或间接获得。

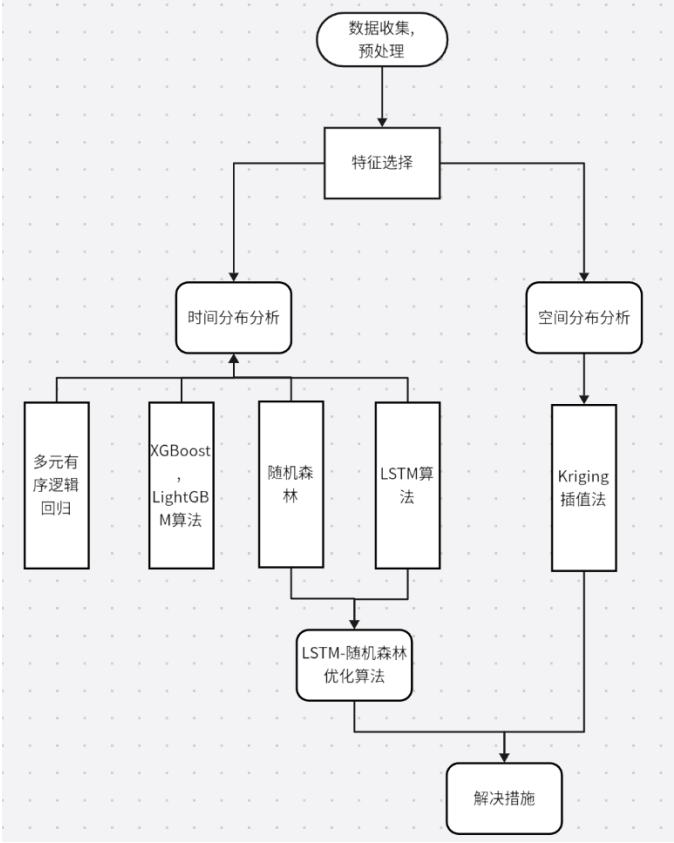

图 1 问题研究流程图

（三）数据预处理

1.数据的获取与整理

为了使大气质量的预测更具现实意义，在选取数据时选择以月为单位进行，共提取了 2013 年 12 月——2024 年 4 月各数据指标。利用 Python 对数据作预处理，删除无用列数据。

2.缺失值、异常值处理

本文对数值型数据使用均值进行填补,对分类型数据使用众数补全。本文使用 Z-score 方法来识别和处理异常值。

3.数据标准化

不同种类的的数据单位不统一将导致比较失去意义，由此我们需要借助数据标准化的方式是两者居于同一比较地位，相对于零均值标准化，我们倾向于采用 min-max 标准化对数据进行归一化处理，将表格中的数据经过变换转化成没有量纲的表达式，缩放到 0 和 1 之间；目的是使各个特征维度对目标函数的影响权重是一致的；改变了原始数据的一个分布，

Min-max 标准化公式如下:

$$x'_{ij} = \frac{x_{ij} - minx_{ij}}{maxx_{ij} - minx_{ij}}$$

对收集的数据进行可视化得到天津市空气质量指数（AQI）等级月变化趋势图，如图一：

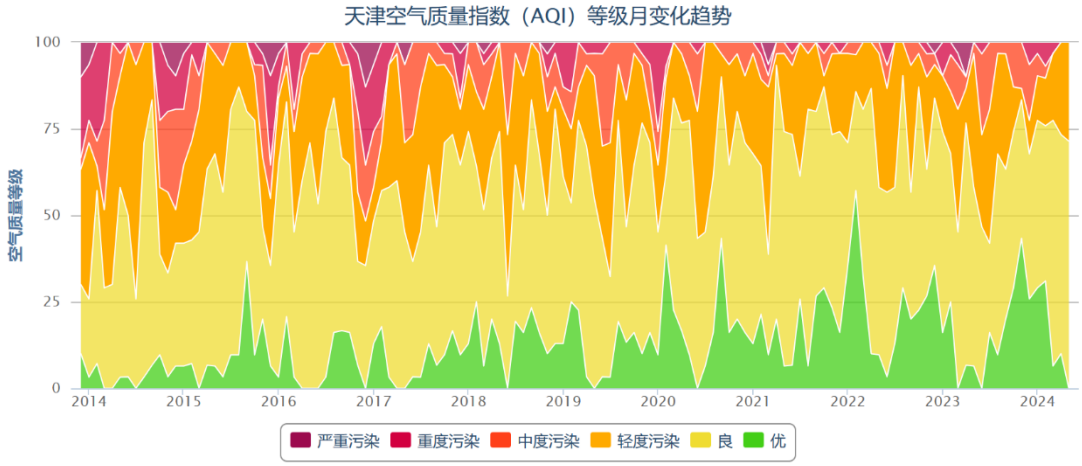

图 2 天津空气质量指数（AQI）等级月变化趋势图

三、相关性分析

（一）灰色关联度分析

我们将 AQI 的值作为重要的评判指标,也就是因变量。自变量作为参考序列以反映空气质量等级的评判标准，对因变量有影响的各指标即 21 个自变量作为比较序列，分别用x_0和x_i（i=1,2,⋯,21）

由于分析序列过分冗长，在此不做具体展示，对比子母序列并结合下列计算：

$$母序列: x_0 = [x_0(1), x_0(2), \cdots, x_0(n)]T$$

$$子序列: x_1 = [x_1(1), x_1(2), \cdots, x_1(n)]T$$

$$x_m = [x_m(1), x_m(2), \cdots, x_m(n)]T$$

为了考虑序列的整体变化趋势和局部变化特征，

$$a = \min_i \min_k |x_0(k) - x_i(k)|$$

$$b = \max_i \max_k |x_0(k) - x_i(k)|$$

通过上述公式，我们可以计算出两级最小差值（a）和两级最大差值（b）。通过综合考虑最大值和最小值，我们能够更全面地描述序列的特征和变化趋势，

并进而准确计算关联度。使用这种方法，可以在一定程度上平衡整体趋势和局部特征之间的关系，从而提高灰色关联模型的准确性和可靠性。

(因此，我们将其代入最终的关联系数计算公式):

$$Y\left(x_0(k),x_i(k)\right)=\frac{a+\rho b}{|x_0(k)-x_i(k)|+\rho b}$$

代入分辨系数$\rho_{=0.5}$，求出关联系数$Y\left(x_0(k),x_i(k)\right)$的值

灰色关联度计算：

$$Y(x_0,x_i)=\frac{1}{n}\sum_{k=1}^{n}y\left(x_0(k),x_i(k)\right)$$

通过除以样本量n，可以对关联系数进行归一化。由于元素组成和粒径所反应的每个数据点上的关联系数取值范围可能存在差异，并且样本量的大小也可能会影响计算结果的尺度。为了使得不同样本量和不同数据范围的序列能够进行比较和对比，我们需要对关联系数进行归一化处理，将其限制在[0,1]的范围内。

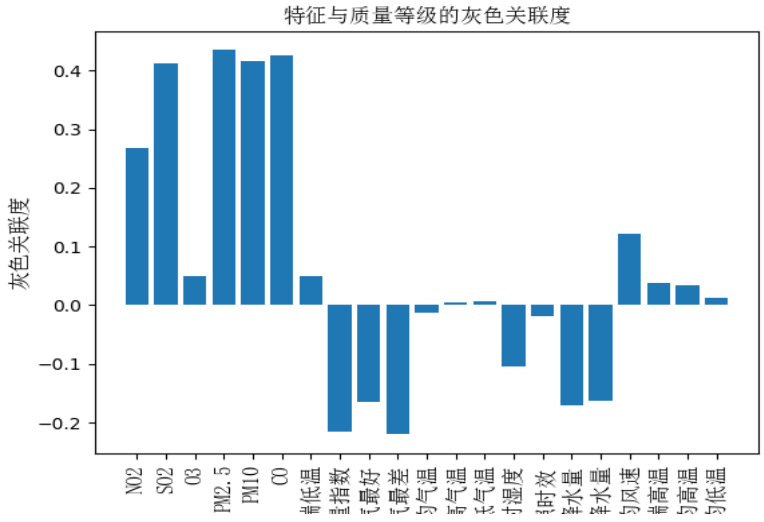

图 3 特征与质量等级的灰色关联度图

从结果可知，NO_2，SO_2，$PM_{2.5}$，PM_{10}，CO，降水量，平均风速，最高气温等特征与空气质量指数的关联度较高，意味着它们可能对于预测空气质量指数具有较强的影响力。

（二）逐步回归分析

逐步回归分析方法的是自动从 21 种可供选择的变量中选取最重要的变量，

首先将自变量逐个引入，引入的条件是其偏回归平方和经检验后是显著的。同时每引入一个新的自变量后，要对旧的自变量逐个检验，剔除偏回归平方和不显著的自变量。这样一直边引入边剔除，直到既无新变量引入也无旧变量删除为止。它的实质是建立“最优”的多元回归方程。本文使用的是向前法建立回归分析的预测或者解释模型。首先分析模型拟合情况R^2,以及可对 VIF 值或者容忍度值。容忍度=1/VIF 值进行分析判断，VIF>5 一般说明特征间有共线性。

表 1 逐步回归分析部分结果

	非标准化系数		标准化系数	*t*	*p*	共线性诊断	
	B	标准差	Beta			VIF	容忍度
常数	-0.161	0.040	-	-4.050	0.000**	-	-
PM2.5	1.053	0.059	1.031	17.962	0.000**	1.508	0.663
03	0.625	0.083	0.797	7.569	0.000**	5.078	0.197
最高气温	-0.233	0.072	-0.340	-3.241	0.002**	5.030	0.199

分析X的显著性,如果显著,则说明X对Y有影响关系,接着具体分析影响关系方向；将 $PM_{2.5}$,PM_{10},CO,NO_2,SO_2,O_3,平均气温,最高气温,最低气温,平均相对湿度,日照时效,降水量,空气最差,空气最好，一日最大降水量,平均风速,平均空气质量指数,平均高温,平均低温,极端高温,极端低温作为自变量，而将 AQI 作为因变量进行逐步回归分析，经过模型自动识别，最终余下 NO_2,SO_2,$PM_{2.5}$,PM_{10},CO,降水量,平均风速,最高气温作为自一共 8 种特征,最后得到模型的R^2值为 0.790，意味着这几种特征可以解释 AQI 的 79.0%变化原因。并且模型通过 F 检验(F=120.716，p=0.000<0.05)，说明模型有效。

灰色关联度分析和逐步回归分析,我们决定将 NO_2，SO_2，$PM_{2.5}$，PM_{10}，CO，降水量，平均风速，最高气温这 8 种特征作为自变量。

四、基于随机森林创新模型的构建与评价

（一）多元有序逻辑回归原理及步骤

通过前面的相关性分析,我们决定将 NO_2，SO_2，$PM_{2.5}$，PM_{10}，CO，降水量，平均风速，最高气温这 8 种特征作为自变量，而将质量等级作为因变量进行有序 logistic 回归分析[6][12]并且使用 Logit 连接函数来预测当地某一时刻的天气状况属于哪一种质量等级。

1.连接函数选择

因为选取的 8 种变量的分布较为均匀,所以我们选用 Logit 函数作为连接函

数：

$$P(y \leqslant j) = \frac{e^{\alpha_j - \sum_{k=1}^{K} \beta_i x_{ij} + \varepsilon_i}}{1 + e^{\alpha_j - \sum_{k=1}^{K} \beta_k x_{ij} + \varepsilon_i}}$$

式中j为感知变化等级；i为自变量下标；$P(y \leqslant j)$代表质量等级为j这一事件发生概率；X_{ij}为 8 个自变量；β_i为系数；α_j为截距；ε_i为误差。

$$O_i = \frac{P_i}{1 - P_i}$$

式中O_i为比数,指相对于不发生的可能性而言,发生的可能性。可以得到有意义的解释,并且除去上限的限制。

步骤二：取(2)式的对数：

$$Y_i = \mathrm{Ln}\left[\frac{P_i}{1 - P_i}\right]$$

可知(3)式无上下限,并且以 0.5 为中点对称,概率上很小的改变会引起更大的变化,使得数据更易观测。

步骤三：在没有了上限和下限的情况下,Y相对于X上的改变可以说是线性相关的:

$$\mathrm{Ln}\left(\frac{P_i}{1 - P_i}\right) = b_0 + b_1 X_i = \log itY_i$$

则有：

$$P_i = \frac{e^{Y_i}}{1 + e^{Y_i}}$$

由此可知，(5)式就是 logistic 回归模型的一般表达式。

2.平行性检验

平行性是有序 Logit 回归的前提条件，如果不满足平行性就无法使用该模型,因此进行平行性检验

H_0:有序 Logestic 回归满足平行性

H_A:有序 Logestic 回归不满足平行性

从结果可知 p 值大于 0.05,说明模型接受原假设，即模型满足平行性检验,接受原假设。

3.似然比检验

首先对模型整体有效性进行分析(模型似然比检验)

H_0:是否放入自变量两种情况时模型质量相等

H_A:是否放入自变量两种情况时模型质量不相等

表 2 有序 Logistic 回归模型似然比检验

模型	-2 倍对数似然值	卡方值	*df*	*p*	AIC 值	BIC 值
仅截距	188.770					
最终模型	136.352	52.418	8	0.000	156.352	184.715

从上表可知：chi=52.418,p=0.000<0.05 此处拒绝原假设,且说明本次构建模型时，放入的自变量具有有效性，本次模型构建有意义。

4.模型建立和参数假设

使用最大数似然估计和梯度下降算法进行参数估计

表 3 有序 Logistic 回归模型分析结果汇总

项	项	回归系数	标准误	z 值	Wald x^2	p 值	OR 值	OR 值 95% CI
因变量阈值	2.0	5.745	1.545	3.718	13.826	0.000	0.003	0.000~0.066
	3.0	10.388	1.937	5.364	28.776	0.000	0.000	0.000~0.001
自变量	NO2	-1.899	1.976	-0.961	0.924	0.337	0.150	0.003~7.198
	SO2	-0.415	2.460	-0.169	0.028	0.866	0.660	0.005~81.985
	PM2.5	7.787	3.701	2.104	4.426	0.035	2408.475	1.704~3405104.396
	PM10	1.518	2.648	0.573	0.328	0.567	4.562	0.025~818.955
	CO	2.860	3.245	0.881	0.777	0.378	17.459	0.030~10093.606
	降水量	0.169	1.365	0.124	0.015	0.901	1.185	0.082~17.191
	平均风速	1.087	1.255	0.866	0.750	0.386	2.965	0.253~34.694
	最高气温	2.554	1.218	2.097	4.398	0.036	12.861	1.182~139.980

并得到有序逻辑回归模型：

$$Logit(odds) = 5.745 - 1.899 * NO_2 - 0.415 * SO_2 + 7.787 * PM_{2.5} + 1.518 * PM_{10}$$

$$+ 2.860 * CO + 0.169 * \text{降水量} + 1.087 * \text{平均风速} + 2.554 * \text{最高气温}$$

5.基于机器学习的有序逻辑回归

最后使用有序逻辑回归算法进行预测,通过模型预测准确率去判断模型拟合质量，由结果可知：研究模型的整体预测准确率为 77%，模型拟合情况可以接受。

建立模型之后,使用有序逻辑回归算法对已有的数据进行分类,可以判断在某一时刻该地区的空气质量等级， 通过模型预测准确率去判断模型拟合效果，由结果可知：

	precision	recall	f1-score
2	0.79	0.88	0.83
3	0.71	0.71	0.71
4	0.00	0.00	0.00
accuracy			0.77
macro avg	0.50	0.53	0.52
weighted avg	0.71	0.77	0.74

图 4 多元有序逻辑回归结果

从结果可知模型对于类别 2 的性能表现较好。精确度为 0.79 表示在所有被模型预测为类别 2 的样本中，有 79%是真正属于类别 2 的。召回率为 0.88 表示在所有真实属于类别 2 的样本中，有 88%被模型成功预测为类别 2。F1-score 为 0.83 是精确度和召回率的调和平均值，综合考虑了精确度和召回率的表现。

对于类别 3，模型的性能较为一般。精确度为 0.71 表示在所有被模型预测为类别 3 的样本中，有 71%是真正属于类别 3 的。召回率为 0.71 表示在所有真实属于类别 3 的样本中，有 71%被模型成功预测为类别 3。

模型的整体准确率为 0.77，模型在所有样本上的正确分类比例为 77%。

综合来看，该有序逻辑回归模型在类别 2 上表现良好，但在类别 3 和类别 4 上性能较为一般。所以我们选择其他的模型进行优化。

（二）随机森林、XGBoost、LightGBM、LSTM

在数据分析的初始阶段，决策树对于探索多个因素与目标变量（即 AQI 值）之间的关系非常有用且易于解释。为了进一步确保结果更加准确可靠，我们首先介绍三种基于决策树的集合方法，包括随机森林、XGBoost 和轻梯度提升机（Light Gradient Boosting Machine，简称轻 GBM）。同时，为了更好地对月份这一时间序列数据进行分析预测，我们使用 LSTM 这一长短期记忆递归神经网络进行时间序列上的分析和预测。

1.随机森林的原理

随机森林[7]是一种使用多棵决策树对样本进行训练、分类和预测的方法。在数据分类过程中，可以通过每个变量的重要性来衡量其在分类中的地位。随机森林中的 ″随机 ″有两层含义：首先，样本选择是基于带撤回的抽样，这意味着每个样本都有可能被多次选中或不被选中。这种抽样方法可以有效地增加决策树之间的差异，进一步降低过度筛选的风险。其次，在构建决策树的过程中，随机森林并不是使用所有特征来构建每一棵决策树，而是从所有特征中随机抽取一部分特征来构建决策树。这种方法可以有效降低特征之间的相关性，提高模型的性能。在随机森林中，每棵决策树都是通过对数据进行迭代分区来构建的。在构建决策树的过程中，数据会根据特定的指标进行分割，直到达到预定的停止条件。为了避免过度分裂问题，随机森林还可以通过控制决策树的深度和节点停止分裂的最小样本数等参数来限制决策树的生长。

2.随机森林的算法步骤

本模型的预测目标是空气质量指数 AQI 的值。随机森林预测过程主要包括三部分内容：数据拆分、训练和测试。

数据拆分环节，数据集被分成两个子集，分别是训练集和测试集，各占总数据集的 50%。随机森林回归（RFR）是一类基于决策树的机器学习算法，在使用训练集进行模型训练的过程中，大部分数据将被直接打包，并通过不同的决策树进行拟合，另外有 10%的数据用于检测模型是否存在过拟合。

决策树评价回归质量的标准选用的是均方误差（Mean Squared Error, MSE），计算公式为

$$MSE = \frac{1}{m}\sum_{1}^{m}(y_{f_i} - y_{c_i})^2 \ (i = 1,2,\cdots,n)$$

式中：m是一棵决策树上的节点数量，y_{f_i}是父节点的数值，y_{c_i}是子节点的数值。在决策树每一个节点的选择中，MSE更小的节点将被视为回归质量高的节点。

每棵树都是随机从 3 个特征中选择固定数量的特征子集，且都尽最大可能地生长，并且没有剪枝的过程。最终每棵树都会获得对应的预测结果，根据预测结果占比，可以获取最合理的结果，并作为最终预测结果。在这一过程中，根据学习曲线，不断整理参数，以获取最佳结果。

测试集的主要作用是验证模型的效果。在测试环节所用的决策树不需要再进行训练，直接使用在训练环节已经确定的树。同样，打包所用的参数也是训练过程中保存下来的，当获取到测试数据集的最终预测结果后，可以计算本模型的精度。本文选用确定系数R^2来对模型进行评估，计算公式为

$$R^2 = 1 - \frac{\sum_1^n (\hat{y}_i - y_i)^2}{\sum_1^n (\overline{y}_i - y_i)^2} (i = 1,2,\cdots,n)$$

式中：n数据集的数据量，y_i是原始数据，$\hat{y}_i$是预测数据，$\overline{y}_i$是原始数据均值。R^2的取值范围为 [0，1]，如果结果是 0，说明模型拟合效果很差，如果结果是 1，说明模型无错误。

3.XGBoost、LightGBM 的原理和结果

XGBoost 算法属于 Boosting 框架，其本质区别在于优化残差树所需的增益不同，XGBoost 使用的增益是分割前后的结构得分之差。XGBoost 的一个重要特点是引入了新的分割标准，在最佳分割点将分割损失降至最低。XGBoost 算法的核心思想分为三个步骤。首先，采用特征分割法不断增加树，每增加一棵树实质上都是学习一个新函数来修正上次预测的残差。其次，在完成训练并获得 k 棵树后，应预测样本的得分。第三，样本的预测值是每棵树对应分数相加的结果。XGBoost 模型迭代m次后，目标函数的定义如下式所示。

$$O^{(m)} = \sum_{i=1}^{n} l(y_i, \hat{y}_i^{(m-1)} + f_t(x_i)) + \Omega(f_i)$$

式中，l和Ω分别为损失函数和正则项，y_i和f_i分别代表结果对应的真实值和模型的预测值。

LightGBM 旨在解决 GBDT 在处理大规模数据时所面临的挑战，使 GBDT 能更好、更快地应用于工业实践。LightGBM 通过引入直方图算法和采用限制性分叶策略，克服了 XGBoost 算法内存消耗大、训练时间长等缺点。直方图算法利用直方图找出最佳分割点，处理连续变量，减少特征中的特征值数量，并减少叶节点分割时需要处理的特征值数量。其基本思想分为三个步骤。首先，将连续波动点特征值离散为 k 个整数，并构建宽度为 k 的直方图。然后，在遍历数据时，将离散值作为指数并累积到直方图中。在此基础上，进行遍历并找到最佳

分割点。如下图所示，XGBoost 算法采用 Level-wise 作为增长策略。这种策略对数据进行一次遍历，可以同时分割同一层的叶子，有利于控制模型的复杂度，达到控制迭代的效果。但在实际应用中，大多数叶子的分割增益相对较小，因此没有必要对叶子进行搜索和分割，从而避免了不必要的计算。

LightGBM 算法使用"Leaf-wise"作为生长策略。如图 4 所示，Leaf-wise 算法每次从当前的所有叶子中进行分割，找出分割增益最大的叶子，然后重复这一过程。与分层算法相比，分叶算法具有以下优点。在分割数量相同的情况下，叶式算法能有效减少误差，提高算法的准确性。但叶式算法的缺点是会形成一棵更深的决策树，从而导致算法过度。因此，LightGBM 算法在"Leaf-wise"算法的基础上增加了最大深度限制，以避免过密，提高计算效率。

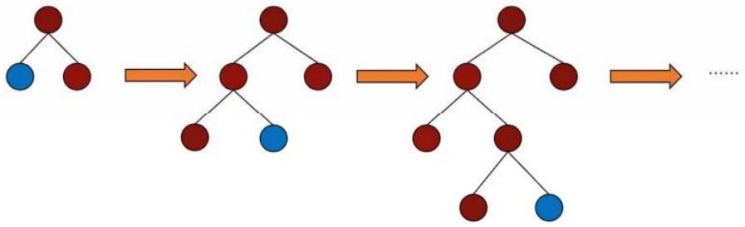

图 5 LightGBM 算法原理图

使用三种算法进行分析预测，R^2可以用来评估统计模型的拟合程度,得到R^2值：

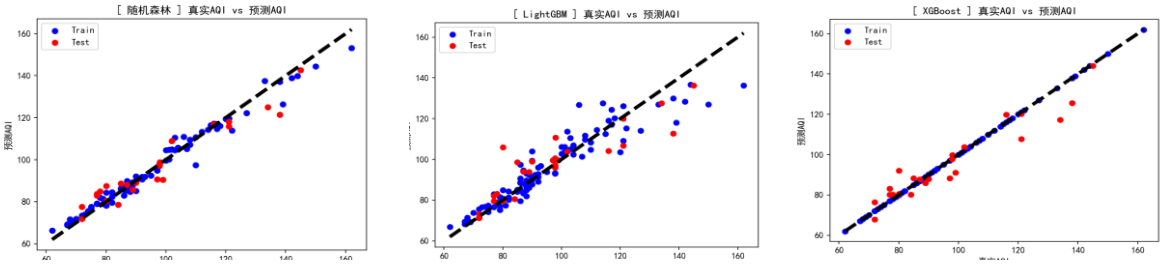

图 6 随机森林、XGBoost、LightGBM 拟合效果图

并绘制拟合效果图，R^2衡量了因变量的变异程度中，由模型解释的部分所占的比例。R^2越接近 1，表示模型能够很好地解释因变量的变异；而R^2越接近 0，表示模型对因变量的变异解释能力较弱。

其中绘制出三个算法的R^2值如下：

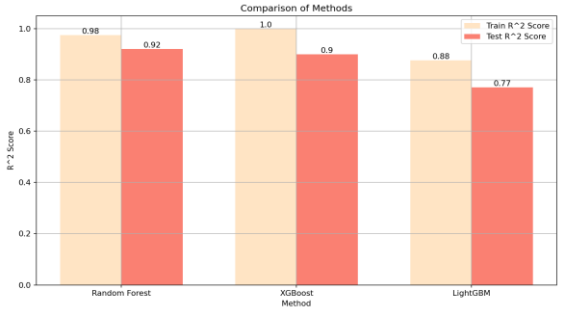

图 7 随机森林、XGBoost、LightGBM R^2值可视化图

从图中可知，随机森林的训练和预测效果均较好，LightGBM 效果相对较差，XGBoost 对于训练集有很好的拟合度，但是对于预测集的效果不好。

本文基于随机森林良好和稳定的分析预测效果，我们选择其进行进一步的优化算法。因为我们观察到收集到的数据在时间序列上具有季节性特征，所以我们选择 LSTM 算法进行优化。

4.LSTM 的原理

LSTM（Long Short-Term Memory）是一种循环神经网络[8][9]（RNN）的变种。RNN 是一类用于处理序列数据的神经网络，它在每个时间步接受一个输入，并在下一个时间步产生一个输出，同时还会保存一些内部状态以处理序列信息。

LSTM 是 RNN 的一种特殊类型，旨在解决传统 RNN 在处理长期依赖性时容易出现的梯度消失或梯度爆炸问题。它通过引入门控机制（如输入门、遗忘门、输出门）来控制信息的流动，有效地捕捉和利用长期依赖关系。

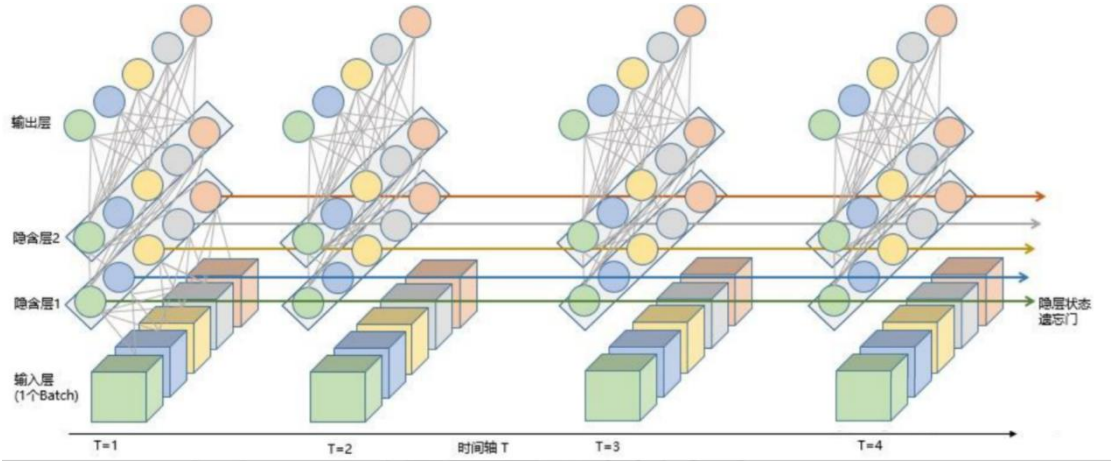

图 8 LSTM 原理图

5.LSTM 的算法步骤

①门机制

现实中的“门”通常解释为出入口，在 LSTM 网络的门也是一种出入口，但是是控制信息的出入口。门的状态通常有三种状态，分别为全开（信息通过概率为 1），全闭（信息通过概率为 0）以及半开（信息通过概率介于 0 和 1 之间）。在这里，我们发现对于全开，全闭以及半开三种状态下的信息通过可以通过概率来表示，在神经网络中，sigmoid 函数也是一个介于 0 和 1 之间的表示，可以应用到 LSTM 中门的计算中。

②LSTM 的计算过程

如下是 LSTM 的网络结构的具体形态，如下所示

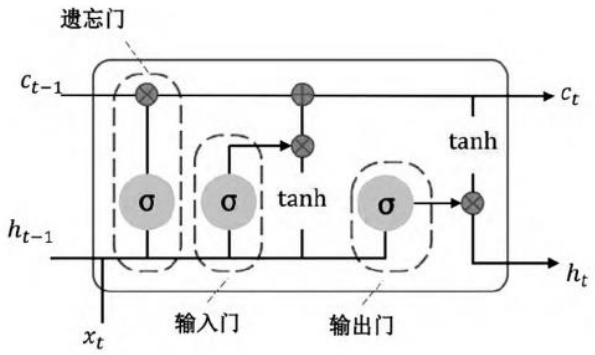

图 9 LSTM 结构图

其中，c_{t-1}表示的是$t-1$时刻的 cell state（注：关于 cell state，查了多个版本的中文翻译，有翻译为“细胞状态”，有翻译成“单元状态”。因为没有一个明确的中文翻译，此处使用英文），h_{t-1}表示的是$t-1$时刻的 hidden state(注：与前面的 cell state 对应)，x_t表示的是t时刻的输入，f_t表示的是遗忘门，i_t表示的是输入门，$\tilde{c}_t$表示的是候选值（candidate values），o_t表示的是输出门。从图中的数据流向得到的计算流程如下所示:

1.利用$t-1$时刻的 hidden state h_{t-1}计算遗忘门f_t的结果，f_t的计算公式如下所示

$$f_t = \sigma(W_f x_t + U_f h_{t-1} + b_f)$$

2.利用$t-1$时刻的 hidden state h_{t-1}计算输入门i_t的结果，i_t的计算公式如下所示

$$i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i)$$

3. 利用$t-1$时刻的 hidden state h_{t-1}计算候选值$\tilde{c}_t$的结果，$\tilde{c}_t$的计算公式如下所示

$$\tilde{c}_t = tanh(W_c x_t + U_c h_{t-1} + b_c)$$

4. 利用$t-1$时刻的 hidden state h_{t-1}计算输出门o_t的结果，o_t的计算公式如下所示

$$o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o)$$

5. 根据t时刻的 cell state c_t，这里会使用到$t-1$时刻的 cell state c_{t-1}，遗忘门f_t，输入门i_t和候选值$\tilde{c}_t$，c_t的计算公式如下所示

$$c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t$$

上述的公式是由前面的 1，2，3 部分的公式组成，也是 LSTM 网络中的关键的部分，对该公式，我们从如下的几个部分来理解:

$f_t \odot c_{t-1}$,使用遗忘门f_t对$t-1$时刻下的 cell state c_{t-1}遗忘;

$i_t \odot \tilde{c}_t$，首先是$\tilde{c}_t$表示的是通过t时刻的输入和$t-1$时刻的 hidden state h_{t-1}需要增加的信息，与输入门i_t结合起来就表示整体需要增加的信息;

两部分结合表示的是t时刻下的 cell state 下需要从$t-1$时刻下的 cell state 中保留的部分信息以及t时刻下新增信息的总和。

6. 根据输出门o_t和 cell state c_t计算外部状态h_t，h_t的计算公式如下所示

$$h_t = o_t \odot tanh(c_t)$$

6. LSTM 的算法结果

通过 LSTM 算法进行时间序列预测得到下图效果：

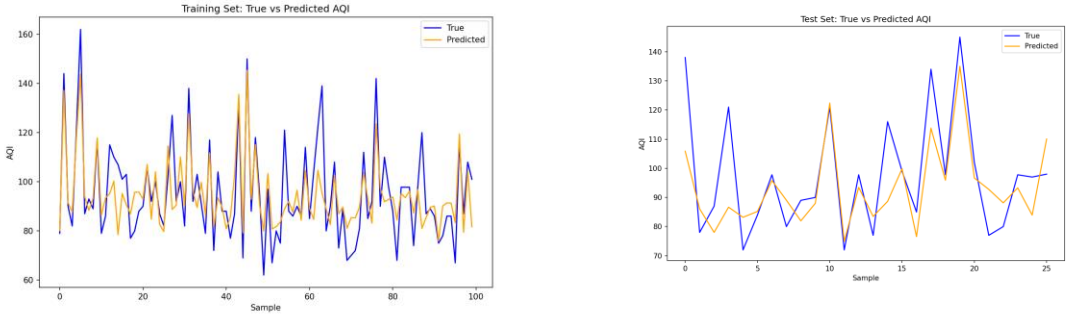

图 10 LSTM 训练集和预测集拟合图

其中：

LSTM Train MSE Score: 202.85757446289062

LSTM Test MSE Score: 198.930908203125

由图和数据可知：LSTM 能进行基本的拟合和预测，但是不能达到最优的效果，我们基于 LSTM 和随机森林进行下一步的模型优化。

（三）基于随机森林的 LSTM 时间序列分析

为了保留随机森林的良好预测结果和 LSTM 的时间预测特性，我们采取构建二者的集成算法[10]，首先使用随机森林模型对数据进行训练和预测，然后将随机森林的预测结果作为特征加入到 LSTM 模型中，最后输出一个能进行时间预测的效果较好的模型。

绘制出分析和预测的拟合图如下：

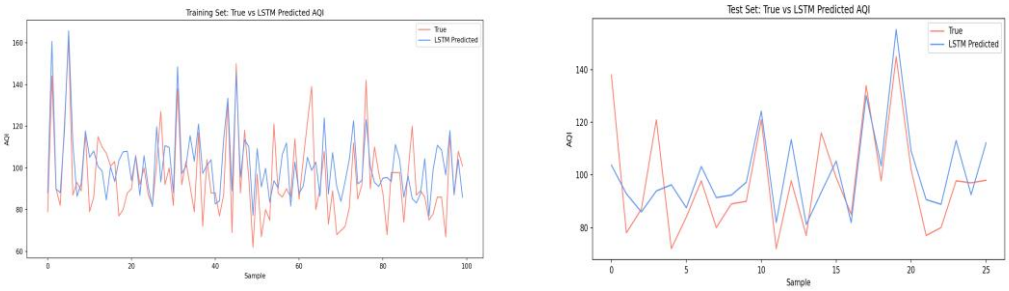

图 11 基于随机森林的 LSTM 训练集和预测集拟合图

对最终模型进行评估：

学习曲线：Training Score 逐渐减小且趋于稳定，可见随着训练集大小的增加，模型在训练集上的性能逐渐提高，但随着数据量的增加，提高的幅度逐渐减小，最终达到一个稳定的水平。这表示模型能够很好地拟合训练数据。

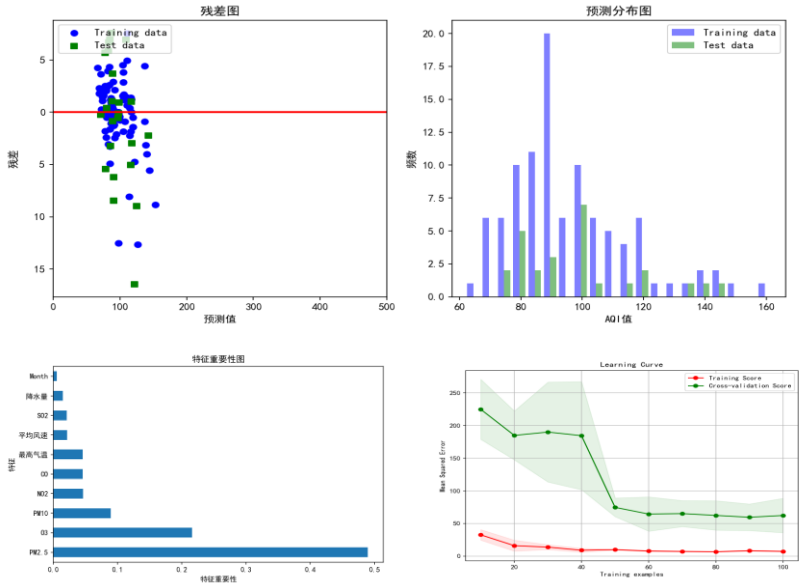

图 12 模型评估可视化

至此构建和完整得到时间序列上的预测模型。

五、空间预测

本文以天津市 23 个监测点的经纬度和 2023 年的 6 种大气污染物年浓度均值为基础数据，进行 Kriging 插值分析[11]。因为 Kriging 插值结果与站点监测值存在一定误差，故需要对插值结果进行可信度验证。Kriging 插值分析是一种基于地理空间数据的插值方法，它可以通过已知数据点的空间分布和属性值来估计未知位置的属性值.本文选取 80%的监测点数据作为训练集来进行空间插值，剩余 20%的监测点数据中山北路和西四道等四个监测点数据作为验证集来进行可信度验证。将模型预测值与站点实际监测值进行对比，发现平均绝对偏差为 8.2%。由此可知，使用 Kriging 插值法对天津市大气污染物浓度的插值预测偏差较小，可信度高。

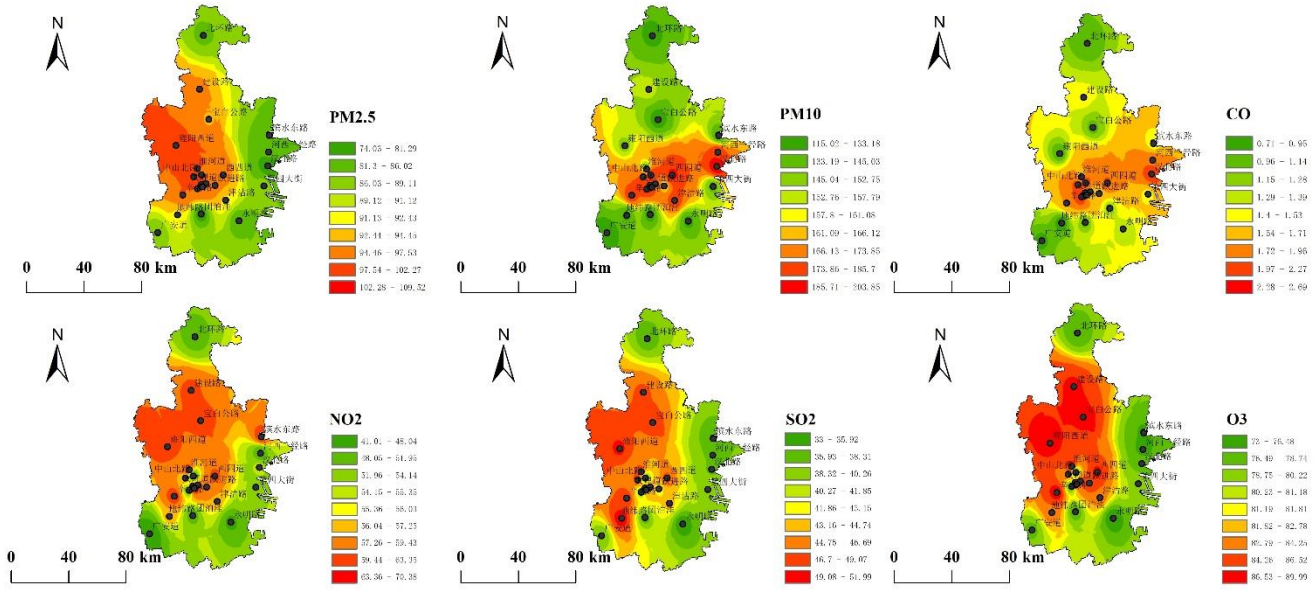

图 13 空气污染物浓度空间分布图

通过 kriging 插值得到的分析结果显示，天津市中心城区的空气污染物浓度分布图中，$PM_{2.5}$和 PM_{10}的浓度呈现"西高东低"的趋势。而 SO_2和 CO 的分布格局变化较大，但年均浓度最高的地区仍然是天津市中心。重度污染区逐渐向城市郊区转移。NO_2 的空间分布呈现"城镇高、乡村低"的格局，同时随着时间推移，城市区域的污染有明显缓解的趋势，滨海新区核心区成为污染重点区域。O_3 的

污染主要集中在西部和北部地区，并且随着时间的推移，O_3的污染程度逐年加剧。

六、结论与展望

（一）结论

在时间层面上,使用 LSTM-随机森林优化模型对天津市空气质量进行分析预测天气污染有季节性变化趋势,在每年的 10 月份到次年的 3 月份,AQI 会处于一个较高的水平。因为暖季节污染物反应速度较快：在暖季节，气温较高，大气中的化学反应速率加快，导致污染物的生成和转化速度加快。例如，光化学反应会促使NO_2和挥发性有机化合物生成O_3，而暖季的高温和日照条件有利于这些反应的进行。另外天津市在暖季节通常有较高的风速。强风有助于将污染物稀释和扩散，减少空气中的污染物浓度。而在冷季节，风速较低，使得污染物在空气中停留时间更长，导致污染物的累积和积聚。最后冬季是供暖季节，天津市的能源消耗会大幅增加。燃煤、燃油和天然气等能源的燃烧会释放大量的污染物，如二氧化硫（SO_2）、颗粒物（$PM_{2.5}$和 PM_{10}）和一氧化碳（CO），从而导致冬季污染加重。

空间层面上,天津市部分大气污染物浓度分布有明显差异，但总体上都呈现出西高东低,市区高郊区低的趋势。天津市污染物浓度分布态势与本市的社会经济要素有关，城市中心人口密集，其污染主要与机动车尾气和居民生活源排放有关，而新区和郊区的污染主要与工业污染和机动车尾气有关目前，天津市的大气污染得到了有效的缓解，尽管 NO_2的质量浓度波动较大，但 2019 年的年均质量浓度与 2015 年相比变化不大。机动车尾气排放仍然是天津市大气污染的重要贡献源，而NO_2等污染物经光化学反应又能促进O_3的生成。因此控制机动车辆的污染排放和解决工业污染是天津市当前大气污染治理中急需解决的问题。

（二）可行措施

大气污染是环境保护中很棘手的问题,针对天津市的大气污染问题，以下是一些具体的解决办法。

天津市的大气污染在空间上呈现中心高,四周低的分布特征,因为它是以前的工业基地,工业污染是造成大气质量不高的主要原因之一,所以我们针对这一特点给出以下建议。

1.控制机动车辆尾气排放：推广清洁能源汽车，如电动汽车或混合动力汽车，并提供相应的充电基础设施。

2.强化车辆尾气排放标准，限制高排放车辆的进入和使用，并加强尾气排放监管和执法。并鼓励公共交通工具的使用，提供高效、便捷的公共交通网络，以减少私人汽车的使用量。

天津市的大气污染在空间上呈现中心高,四周低的分布特征,因为它是以前的工业基地,工业污染是造成大气质量不高的主要原因之一,所以我们针对这一特点给出以下建议。

1.加强工业污染治理：严格执行和监督工业企业的污染排放标准，加强对污染物排放的监测和处罚力度。并鼓励工业企业采用清洁生产技术和设备，减少污染物的排放。

2.加强工业园区的环境管理，建设和改造污水处理设施，防止工业废水的直接排放。并提倡节能减排和可持续发展：推广能源高效利用技术，鼓励企业和居民使用节能设备和产品。

另外在政府层面上,也可以从以下方面进行改进：

1.加强环境监测和数据共享：建立健全的大气污染监测网络，覆盖城市中心、新区和郊区等不同区域，实时监测和报告污染物浓度。并加强数据共享和信息公开，提高公众对大气污染状况的认识，促进公众参与和监督。

2.加强跨部门合作和政策协调：建立跨部门的大气污染治理机制，加强政府各部门之间的协调合作，形成合力。制定综合性的大气污染治理政策，包括法规、经济手段和激励措施，并加强政策的执行和监督。

这些解决办法需要政府、企业和公众的共同努力，通过改变行为习惯、技术创新和政策支持，逐步改善天津市的大气质量，减少污染物的排放，实现可持续发展和生态环境的保护。同时，需要持续监测和评估措施的效果，并不断优化和调整治理策略。

（三）展望

本文根据建立的大气质量的时空预测模型，对天津市等渤海地区城市的大气污染治理和改善具有实际参考价值。但本文仍存在不足之处，有待进一步探讨和改进。

1.本文开始所选取的 21 种可能影响空气质量等级的特征上是否准确和完整有待商榷。

2.由于收集到的数据有较多缺失值，可能会对得到的结果产生一定影响。随着大气质量监测公开数据的发展，将有助于对水质时空分布特征分析的改进。

参考文献

[1] 邓聚龙.灰色系统基本方法[M].武汉：华中理工大学出版社,1987.141-145.

[2] 王学萌,张继中,王荣.灰色系统分析及实用计算程序[M].武汉：华中科技大学出版社,2001.109-118

[3] 邓聚龙.灰色预测与决策[M].武汉：华中理工大学出版社,1987.96-120.

[4] 常学将,陈敏,王明生.时间序列分析[M].北京：高等教育出版社,1993.54-295.

[5] 唐小我.最优组合预测方法及其应用[Ｊ].数理统计与管理,1992,11(1)：31-35

[6] 何晓群，刘文卿.应用回归分析[M].北京:中国人民大学出版社，2015.

[7] 方匡南,吴见彬,朱建平,等.随机森林方法研究综述[J].统计与信息论坛,2011,26(03):32-38.

[8] 韩力群，人工神经网络理论、设计及应用，北京：化学工业出版社，2002，17～23

[9] 刘豹等，神经网络在预测中的一些应用研究，系统工程学报，1999，14（4）：338～343

[10] 陈华友. 组合预测方法有效性理论及其应用 [M]. 北京:科学出版社，2008.

[11] 王占山,李志刚,钱岩,等.基于监测及 Kriging 方法的京津冀地区大气污染物暴露分布研究[J].环境科学研究,2021,34(1):185-193.

[12] CHENINI I,MSADDEK H M. Groundwater recharge susceptibility mapping using logistic regression model and bivariate statistical analysis[J]. Quarterly Journal of Engineering Geology and Hydrogeology,2020,53（2）,167-175.

附录

基于随机森林的 LSTM 时间序列算法代码：

```
import pandas as pd
import numpy as np
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import LSTM, Dense
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import MinMaxScaler
import matplotlib.pyplot as plt
from sklearn.ensemble import RandomForestRegressor

# 读取数据
data = pd.read_excel("D:统计建模.xlsx")

# 将"Month"列转换为日期类型
data['Month'] = pd.to_datetime(data['Month'], format='%b-%y')

# 将"Month"列拆分为年和月，并添加这两列作为特征
data['Year'] = data['Month'].dt.year
data['Month'] = data['Month'].dt.month

# 确定特征和目标列
X = data.drop(["AQI"], axis=1)
y = data["AQI"]

# 划分训练集和测试集
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=42)

# 创建随机森林模型
model_rf = RandomForestRegressor(n_estimators=100, random_state=42)

# 拟合随机森林模型
model_rf.fit(X_train, y_train)

# 使用随机森林模型对训练集和测试集进行预测
y_train_pred_rf = model_rf.predict(X_train)
y_test_pred_rf                =                model_rf.predict(X_test)

# 使用随机森林模型预测训练集和测试集的 AQI 值
y_train_pred_rf = model_rf.predict(X_train)
y_test_pred_rf = model_rf.predict(X_test)
```

```
# 将随机森林的预测结果添加到原始数据集中
X_train['RF_Prediction'] = y_train_pred_rf
X_test['RF_Prediction'] = y_test_pred_rf

# 数据缩放
scaler = MinMaxScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)

# 将数据重塑为 LSTM 所需的形状 (samples, time steps, features)
X_train_reshaped = np.reshape(X_train_scaled, (X_train_scaled.shape[0],
X_train_scaled.shape[1], 1))
X_test_reshaped  =  np.reshape(X_test_scaled,  (X_test_scaled.shape[0],
X_test_scaled.shape[1], 1))

# LSTM 模型
model = Sequential()
model.add(LSTM(50,                                    activation='relu',
input_shape=(X_train_reshaped.shape[1], X_train_reshaped.shape[2])))
model.add(Dense(1))
model.compile(optimizer='adam', loss='mse')

# 训练 LSTM 模型
model.fit(X_train_reshaped, y_train, epochs=100, verbose=0)

# 使用 LSTM 模型进行预测
y_train_pred_lstm = model.predict(X_train_reshaped)
y_test_pred_lstm = model.predict(X_test_reshaped)
```