杭州電子科技大學

本 科 毕 业 设 计

（2022 届）

题 目	基于大数据的天然气用气预测算法实现
学 院	计算机学院
专 业	软件工程
班 级	18052712
学 号	18052221
学生姓名	牛振洋
指导教师	孙笑笑
完成日期	2022 年 6 月

摘　要

天然气主要由烷烃组成，是一种相对清洁的气态化石燃料。实验表明，天然气可以作为我国煤炭的替代能源，并能缓解石油的消耗，可以减少细小颗粒物的排放，并有效抑制大气中的雾霾和其它有毒气体的排放。天然气已经是实现“双碳”目标和“美丽中国”的重要力量。

天燃气用气受到温度、价格、季节等多种复杂因素的影响并且会受到短期相关政策的干扰，导致区域的用气需求、市场是不断变化的。天然气一般由当地燃气公司供给，因此为了帮助燃气公司进行有效的供应管理和规划，需要找到关于短日天然气消费量的正确预测模型。

本文在船舶制造、材料制造、物业管理、学校和生物制药五大行业尝试对比了 GRA_LSTM、LSTM、ARIMA、SARIMA、ARIMAX、ARIMAX 等各种预测模型，并探究了 LSTM 与 ARIMA 串行和并行组合模型,以求得到在本数据集下更好的预测模型，以满足全行业的天然气用气预测需求。对于串联组合模型，本文将 ARIMA 输出的预测序列作为 LSTM 的输入序列。对于并联组合模型，用 SARIMA 模型和 LSTM 模型分别对时序序列进行预测得到预测序列 S 和 L。如果预测序列具有很好的季节周期性，那么 SARIMA 模型得到的预测序列 S 就能提取出良好的季节周期性，使得其在局部上比 L 有更高的吻合度。同时利用 LSTM 本身相对于 SARIMA 的预测精度高的优点，可知预测序列 L 在整体上比 S 具有更高的吻合度。最后，对 SARIMA 进行整体上的比例修正，以达到保留局部优势扩大全局优势的目的。实验表明，各个模型在各行业中都有较好的预测拟合效果（R-Squared>0.8）且差距不大。

关键词：天然气用气；预测；LSTM; ARIMA; GRA

ABSTRACT

Natural gas, consisting mainly of alkanes, is a relatively clean gaseous fossil fuel. Experiments show that natural gas can be used as an alternative to coal in China, and can alleviate the consumption of oil, can reduce the emission of fine particles, and effectively inhibit the emission of haze and other toxic gases in the atmosphere. Natural gas is already an important force in achieving the "dual carbon" goal and the "beautiful China".

Natural gas consumption is affected by a variety of complex factors such as temperature, price and season, and will be interfered by short-term relevant policies, resulting in the constant change of regional gas demand and market. Natural gas is usually supplied by local gas companies, so to help gas companies with effective supply management and planning, it is necessary to find the correct forecast model for short-term gas consumption.

This paper tries to compare various prediction models such as GRA_LSTM, LSTM, ARIMA, SARIMA, ARIMAX and ARIMAX in five major industries of shipbuilding, material manufacturing, property management, school and biopharmacology. The serial and parallel combination models of LSTM and ARIMA are explored in order to obtain a better prediction model under this data set, so as to meet the prediction demand of natural gas consumption in the whole industry. For the series combination model, the prediction sequence of ARIMA output is used as the input sequence of LSTM. For the parallel combination model, SARIMA model and LSTM model are used to predict the sequence respectively to obtain the prediction sequence S and L. If the prediction series has good seasonal periodicity, the prediction series S obtained by SARIMA model can extract good seasonal periodicity, which makes it have a higher local coincidence than L. At the same time, by taking advantage of LSTM's high prediction accuracy compared with SARIMA, we can know that the prediction sequence L has a higher coincidence degree than S on the whole. Finally, the overall proportion of SARIMA is modified to retain local advantages and expand global advantages. The experiment shows that each model has a good prediction and fitting effect in various industries (R-Squared>0.8) and the difference is not big.

Keywords: natural gas; To predict; LSTM; ARIMA; GRA

目　　录

1 绪论

1.1 研究背景及意义

当前中国社会经济结构正在面临着巨大调整，积极进行生态文明建设，不断向资源节约型和环境友好型社会转变。目前，温室效应引起的全球变暖等一系列重大环境问题，也在不断说明搞好生态文明建设的必要性和紧迫性。众所周知，二氧化碳的大量排放是温室效应的罪魁祸首，给人类的可持续健康发展构成了巨大威胁。数据表明，二氧化碳的排放量近九成来自煤炭燃烧，而中国是煤炭使用大国，对煤炭有极强的依赖性，这就意味着高效、清洁的新能源在这次社会经济结构调整的重大意义。而，天然气作为一种可清洁能源是实现“双碳”目标和“美丽中国”的重要力量。实验说明，天然气可以作为我国煤炭的替代能源，并能缓解石油的消耗，可以减少细小颗粒物的排放，并有效抑制大气中的雾霾和其它有毒气体的排放。

天然气用途十分广泛[1]。一是作为高效、优质、清洁的能源供居民消费或工业使用，如居民日常能源消费、天然气代替煤炭发电、天然气燃料电池和电动汽车等。 其次，天然气主要成分甲烷可以和二氧化碳共同作为化工原料能够代替石油，生成甲醛、汽油、煤油、柴油、氮肥、液氮等化工产品。有行业内人士表明，到2050年,天然气在能源中的使用占比将从21世纪初期的四分之一上升到五分之二，而石油的使用占比将从34%下降到五分之一。

目前，我国大力推进了居民和工商业领域煤改气措施的实施，城市燃气需求急剧增加，供需形势更加严峻。再加上燃气用气量易受到温度、价格、季节和短期相关政策等多种复杂因素的干扰，这都给当地燃气公司采购、储气和供给工作带来了挑战。因此，为了帮助燃气公司进行有效的供应管理和规划，促进天然气资源高效利用，需要找到关于短时天然气消费量的正确预测模型。

1.2 国内外研究动态

时间序列预测是一种应用场景十分广泛的方法，可以被应用在交通流量、生物迁徙、水文气象、能源优化等众多领域。建立时间序列预测模型，通过挖掘历史观测数据发现数据波动规律，进而预测未来走势。时序数据有平稳和非平稳之分。平稳时序数据的预测已经相当成熟，尤其是线性模型的建模。然而，在实际生活场景中，我们所接触到的时序数据大多数非平稳的和非线性的，而且它们所服务的实际场景不同，会受到各种因素的影响并在短时间内会有无规律的随机波动，以致目前还没有绝对完美的方法或技术来分析和处理这类时序数据。目前，

应用广泛的时序预测模型有：LSTM 时序预测模型(Long Short-Term Memory networks, LSTM)[2]、GRU 时序预测模型(Gated Recurrent Unit networks, GRU)[3]、Transformer 时序预测模型（Attention Is All You Need)[4]和 ARIMA 时序预测模型(Auto regressive Integrated Moving Average model, ARIMA)[5]。

在天然气需求预测中，对短期、中期和长期需求预测的研究较多，主要采用统计方法、人工智能（AI）方法以及两者的混合组合[6]。如，小波分析、神经网络方法、神经模糊、机器学习、灰色理论预测[7]。值得注意的相关研究有：基于自适应网络的模糊推理算法[8][9]、基于优化遗传算法和改进 BP 神经网络的天然气短期负荷预测模型[10]、及默克尔等人率先研究了利用 deep_learning[11]进行天然气短期负荷预测[12]。

在摩尔定律下，计算机算力性能不断突破。与此同时，内存、磁盘等数据存储设备的价格不断下降，计算机设备的在各种各样的场景中的应用中产生了庞大的数据，促使了大数据时代的到来。再加上人工智能在特定领域和场景中异于常人的喜人表现，吸引着大批研究人员融入进来，相关算法不断优化和突破。在算力、数据、算法的三驾马车的拉动下，人工智能不断发展和日趋成熟。神经网络、支持向量机（Support Vector Machine, SVM）等模型在时序数据预测中有不错的表现。其中长短期记忆网络(Long Short—term Memory，LSTM)是最流行的预测方法之一，已经被广泛应用在医疗、教育、生产、金融等多个领域。LSTM 同 RNN 一样是一种递归神经网络，能够处理时序信息，但，LSTM 可以选择的进行记忆和遗忘，不会出现在时序过长的情况下梯度爆炸或梯度消失的情况，因而具有更高的预测精度。而且 LSTM 模型可以借助其它外生辅助变量对历史时序数据进行预测，基于灰色关联度分析(G11ey Relation Analysis，GRA)是在外生变量中筛选出辅助变量的常用方法。GRA_LSTM 模型虽然可以得到更高的预测精度，但在面对具有明显季节性的时序数据时，和 LSTM 一样并不能很好的提取时序数据中的季节因子。

RNN 不同于传统前馈神经网络，它的隐藏层之间存在时序依赖，具有记忆性，有着处理时序序列的优点。但 RNN 神经网络在反向传播过程中存在梯度消失或梯度爆炸的问题，这就限制了它的记忆功能，无法将更远的时序信息影响当前的输出。这是由长短期记忆人工神经网络解决算法 LSTM 优于原版本 RNN 的地方[13]。由于 LSTM 具有特定的单元结构，这使得算法有能力通过称为门的结构改变来自这个单元的信息量。这些“门”通过让模型知道哪些信息要储存在长记忆中，哪些要丢弃，来控制这个记忆过程。LSTM 在时间序列的动态时间行为问题中发现了很好的应用，比如时间序列分析或单词预测。如，应用 LSTM 算法预测 G 地区次日天然气消费需求[14]。

对于 SARIMA，在面对具有良好季节周期性和趋势性变化的时间序列时有着不错的效果。如，应用 SARIMA 算法预测次日电价[15]。

1.3 本文研究内容及章节安排

1.3.1 本文研究内容

根据上文的国内外研究动态，SARIMA 模型和 LSTM 模型都是十分流行的时间序列预测模型，并且有各自的优点和适应场景，SARIMA 能够提取季节周期性，LSTM 能够根据历史信息学习非线性时序序列，所以如果能将 SARIMA 与 LSTM 的各自的优点结合起来或许可以对时间序列有更精确预测效果。针对 SARIMA 模型和 LSTM 模型的组合，本文探究了两种模型进行串联组合和并联组合的预测效果。对于串联组合模型，本文将 ARIMA 输出的预测序列作为 LSTM 的输入序列。对于并联组合模型，用 SARIMA 模型和 LSTM 模型分别对时序序列进行预测得到预测序列 S 和 L。如果预测序列具有很好的季节周期性，那么 SARIMA 模型得到的预测序列 S 就能提取出良好的季节周期性，使得其在局部上比 L 有更高的吻合度。同时利用 LSTM 本身相对于 SARIMA 的预测精度高的优点，可知预测序列 L 在整体上比 S 具有更高的吻合度。最后，对 SARIMA 进行整体上的比例修正，以达到保留局部优势扩大全局优势的目的。对于 GRA_LSTM 模型[16]，最终输入为原生变量和利用 GRA 法在外生变量中筛选出的辅助变量的融合。

本文旨在找到更好的预测算法，以满足全行业的天然气用气预测需求。本文对比了 GRA_LSTM、LSTM、ARIMA、SARIMA、ARIMAX、ARIMAX 等各种预测模型。并探究了 LSTM 与 ARIMA 串行和并行组合模型。

1.3.2 本文章节安排

按照章节划分，本文共分为 5 个章节，各章的具体结构如下：

第 1 章为绪论，介绍课题研究背景以及国内外天然气用气量预测分析现状。

第 2 章为相关理论介绍，将介绍相关算法理论，以及分析各类算法在天然气用气量分析领域的适用性。

第 3 章为基于第 2 章相关理论的天然气用气量预测算法的实现，包括数据预处理、模型设计流程和相应的代码实现。

第 4 章对第 3 章实现的各种算法进行性能评估和对比。

第 5 章是总结与展望，作为全文的最后一章，对全文做总结，并说明本文实验得出的结论和自身的局限，从而引出对未来的展望。

2 相关理论概述

本章节介绍了各类机器学习算法的基本原理以及在天然气用气量预测中的适用性。ARIMA 算法、LSTM 算法的基本原理及实现流程进行介绍，再进一步根据其各自特点分别分析其在天然气用气量预测中的适用性。

2.1 ARIMA 算法介绍

2.1.1 ARIMA 算法原理

2.1.1.1 预备知识

（1）白噪声

白噪声

由一组均值为 0、方差为常数且相互独立的元素组成。

（2）平稳性

指的是时间序列样本及拟合曲线在未来时间点上按照现有的某种趋势延续下去。均值与方差无明显的变化。平稳性分为严平稳性和弱平稳性。

①严平稳：严平稳代表的分布不随时间的推移而改变。例如白噪声，其分布特征不随时间改变为正态分布，期望和方差恒定不变，分别为 0 和一个常数 c。

②弱平稳：期望与相关系数（依赖性）不改变。某时刻 t 的值 X_t 需要依赖于其历史信息，所以具有时序依赖性。时间序列模型主要用来预测，若平稳性都无法保证，何来预测？

因为严平稳性数据不太可能出现，实际场景中，数据基本都是弱平稳，需要进行差分。

（3）差分

举例来说：一阶差分就是 X_t 与 X_{t-1} 时刻的差分值。n 阶差分就是在 n-1 阶差分完成后，再进行一次一阶差分操作。

2.1.1.2 AR(自回归)

AR 表示自回归(Auto Regression)，表示这是对自身变量的回归。在自回归模型中，我们使用过去变量的线性组合来预测。

p 阶自回归如下定义：

$$y_t = \mu + \sum_{i=1}^{p} \gamma_i y_{t-i} + \varepsilon_t \tag{2.1}$$

其中，y_t 是当前自身值，μ 表示常数项，p 是阶数，γ_i 是自相关系数，ε_t 是误差。

2.1.1.3 I(单整阶数)

I 表示单整阶数(Integration)

时间序列必须具有平稳性，这样才能输入计量模型，用于预测。通常要对时间序列进行单位根检验，以判断其是否具有平稳性。如果时间序列数据是非平稳的，需要在此基础上进行一阶数据差分。重复上述操作直到将时序序列转化为平稳序列，时序序列从平稳性到非平稳性所经过的差分次数，就称为几阶单整阶数；

一阶差分：对时间序列在 t 与 t-1 时刻函数值进行差值操作以提高时序数据的平稳性。

2.1.1.4 MA(移动平均)

MA 表示移动平均(Moving Average)，移动平均模型关注 AR 模型中 ε_t 的累积，能有效地消除预测中的随机波动。

q 阶 MA 模型的公式定义：

$$y_t = \mu + \sum_{i=1}^{q} \beta_i \varepsilon_{t-i} + \varepsilon_t. \tag{2.2}$$

2.1.1.5 ARMA 模型

将 AR(p)与 MA(q)结合，得到一个一般的自回归移动平均模型 ARMA(p，q),该序列可以由其自身的历史值以及随机扰动项来表示。

$$y_t = \mu + \sum_{i=1}^{p} \gamma_i y_{t-i} + \sum_{i=1}^{q} \beta_i \varepsilon_{t-i} + \varepsilon_t. \tag{2.3}$$

2.1.1.6 ARIMA 模型

ARIMA 模型(Auto regressive Integrated Moving Average model)包含 3 个部分，即自回归(AR)、差分(I)和移动平均(MA)，是流行的时间序列预测分析方法之一，全称叫做自回归差分移动平均模型。

ARIMA 模型相对于 ARMA 模型多了数据差分操作，这就表明其不同于 ARIMA 模型，可以对非平稳时间序列进行预测。换句话说，如果要预测的时间序列是非平稳的，则 ARMA 模型不使用，首先需要经过差分操作，然后建立 ARMA 模型

ARIMA 模型记作 ARIMA(p，d，q)，p 为自回归项数；q 为滑动平均项数，d 为使之成为平稳序列所需要的单整阶数。当 $p == q == d == 0$ 时，ARIMA 学不到任何顺序信息，即退化为白噪声序列。

ARIMA(p,d,q)阶数确定，参照表 2-1

表 2-1 ARIMA(p,d,q)阶数确定

模型	ACF	PACF
AR(p)	衰减趋于 0（几何或震荡型）	p 阶后截尾
MA (q)	q 阶后截尾	衰减趋于 0（几何或震荡型）
ARMA	q 阶后衰减趋于 0（几何或震荡型）	p 阶后衰减趋于 0（几何或震荡型）

2.1.1.7 SARIMA 模型

SARIMA 模型又称季节差自回归移动平均模型，是在 ARIMA 模型的基础上发展起来的一种预测模型，由于考虑了季节因素更适用于季节性和周期性变化的数据，是时间序列领域的主要预测模型之一。该模型的一般表达式为：

$$\mathrm{SARIMA}(p,d,q)(P,D,Q)_s \tag{2.4}$$

其中，P、Q、D 分别为季节求和自回归移动平均模型中的自回归、移动平均和差分的值；S 为季节周期和循环长度。当 $\mathrm{P}==D==Q==S==0$ 时，SARIMA 模型退化成 ARIMA 模型。

2.1.1.8 ARIMAX 模型

实际情况中很多序列的变化规律会受到其它序列的影响，针对这种情况需要建立多元时间序列的 ARIMAX 模型。ARIMAX 模型是指带回归项的 ARIMA 模型，又称扩展的 ARIMA 模型。回归项的引入有利于提高模型的预测效果。引入的回归项一般是与预测对象（即被解释变量）相关程度较高的变量。比如在本案例，分析天然气使用量序列时，当天的使用量会受到当日温度的影响，如果将当日的温度也纳入到研究范围，就能够得到更精确的天然气使用量预测。

2.1.2 在天然气用气量预测中的适用性

ARMA 模型预测只考虑预测序列本身历史数据，不直接考虑其他相互因素的变动，这就表明 ARIMA 算法易于建模。但要注意，ARMA 模型要求时序数据是平稳的，否则需要使用 ARIMA 扩展的差分功能。若时间序列具有良好的季节周期性，SARIMA 具有其自身的优势。若要预测的时序序列受外生变量影响较大，ARIMAX 具有其自身的优势。

2.2 灰度关联分析算法介绍

2.2.1 灰度关联分析算法原理

GRA 法是一种通过用灰色关联度来分析各个外生变量与内生变量的关联度，进而确定辅助变量的方法。GRA 法计算简单，使用方便快捷，用于分析两个变量的关联度，在本案例中，可被用于分析各种外生变量与天然气使用量自身的内生变量之间关联程度，从而删选出影响天然气使用量的关键外生变量，最终模型的输入为原生变量和利用 GRA 法在外生变量中筛选出的辅助变量的融合。

该方法使用步骤如下：

Step1:确定内生变量序列 X_0；外生变量序列 X_1，X_2，...，X_n；

其中 $\mathrm{X}_i = \{\mathrm{X}_i(k) \mid k=1,2,...,m\}$。

Step2:对所有序列进行无量纲化处理：

$$x_i(k)=\frac{\mathrm{X}_i(k)}{\mathrm{X}_i(l)}.(i=0,1,...,n)(k=1,2,...,m). \tag{2.5}$$

Step3:计算 $x_0(k)$ 和 $x_i(k)$ 的关联系数：

$$\xi_j(k)=\frac{\min\limits_i\min\limits_k|y(k)-x_i(k)|+\rho\max\limits_i\max\limits_k|y(k)-x_i(k)|}{|y(k)-x_i(k)|+\rho\max\limits_i\max\limits_k|y(k)-x_i(k)|}. \tag{2.6}$$

其中，ρ 为分辨系数可以自定义。

Step4：计算 X_0 和 X_i 的 r_i：

$$r_i=\frac{1}{n}\sum_{k=1}^{n}\xi_i(i=1,2,...,n)(k=1,2,...,m). \tag{2.7}$$

基于第 4 步的关联度分析结果，将所以外生变量与内生变量的相关度 $r_i=1,2,...,p$，按从大到小进行排序。我们可以设定一个固定的阈值 σ，取 $\sigma\in(0,1)$，对所以满足 $r_i>\sigma$ 的外生变量设定为内生变量的辅助变量，那么 LSTM 模型的最终输入为原生变量和利用 GRA 法在外生变量中筛选出的辅助变量的融合。

2.2.2 在天然气使用量预测中的适用性

实际情况中很多序列的变化规律会受到其它序列的影响，GRA 考虑了这种情况。比如在本案例，分析天然气使用量序列时，当天的使用量会受到当日温度的影响，如果将当日的温度也纳入到研究范围，就能够得到更精确的天然气使用量预测。这里，GRA 算法配合 LSTM 模型一起使用。

2.3 LSTM 算法介绍

2.3.1 LSTM 算法原理

LSTM 在 RNN 基础上引入了细胞状态这种特殊的数据结构，每个隐藏层都有一个这样的数据结构。在处理时序信息时，每个细胞结构按照时间顺序组成一条信息传送带，并且在传送期间，每个细胞能够借助输入门、遗忘门和输出门来更新自身状态、从后有选择地接收和向前有选择地传送信息。这样，在处理长时序数据时，LSTM 通过这种特殊的数据结构避免了 RNN 那样在时序累乘大于 1 的数之后会梯度爆炸或在时序累乘小于 1 的数之后梯度消失的问题。其内部结构如图 2-3 所示。

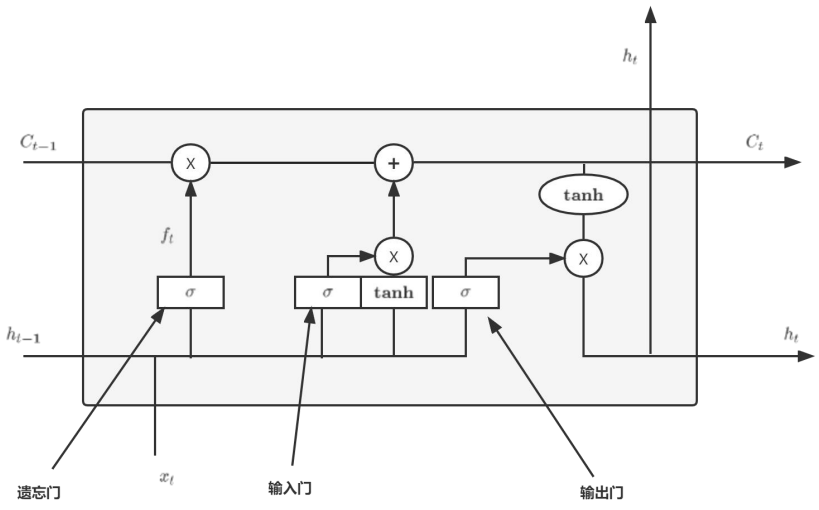

图 2-3 LSTM 神经网络结构图

图 2-3 中，整个矩阵表示的是一个神经网络层；圆点表示是一次矩阵或向量操作，比如矩阵相乘、向量相加；一个有向线段表示从一个节点传输一整个向量到另一个节点；合在一起的线表示向量连接；分开的线表示复制内容，然后分发到不同的位置。

各种门对细胞状态的工作机制，如下：

① 遗忘门通过 *sigmoid* 函数将输入信息映射到 0 到 1，得到遗忘系数 f_t。

$$f_t = \sigma\left(w_1^f .x_t + w_h^f .h_{t-1} + b_f\right) \tag{2.8}$$

其中，x_t 为当前层的输入，h_{t-1} 为前一隐藏层的输出。

② 输入门通过 *sigmoid* 函数将输入信息映射到 0 到 1，得到记忆系数 i_t。

$$\mathrm{i}_t = \sigma\left(w_1^i .x_t + w_h^i .h_{t-1} + b_i\right), \tag{2.9}$$

③ ta*nh* 层 RNN 机制一样，输入信息经过 ta*nh* 存储到候选细胞 $\overline{C_t}$ 中。

$$\overline{\mathrm{C}_t} = \tanh\left(w_1^C .x_t + w_h^C .h_{t-1} + b_c\right). \tag{2.10}$$

④ 更新当前层的细胞状态，通过遗忘系数 f_t 来遗忘旧的细胞状态 C_{t-1}、记忆系数 i_t 来记忆 $\overline{C_t}$，得到细胞状态 C_t。

$$\mathrm{C}_t = f_t .C_{t-1} + i_t .\overline{C_t}, \tag{2.11}$$

⑤ 输出门通过 *sigmoid* 函数将输入信息映射到 0 到 1，得到输出系数 o_t。

$$\mathrm{o}_t = \sigma\left(w_1^o .x_t + w_h^o .h_{t-1} + b_0\right), \tag{2.12}$$

⑥ 通过输出系数 o_t 控制当前层的输出，并作为下一层的输入 h_t

$$\mathrm{h}_t = \mathrm{o}_t .\tanh\left(C_t\right). \tag{2.13}$$

2.3.2 在天然气用气量预测中的适用性

LSTM 算法在天然气用气量预测方面主要有以下几点适用性：

（1）天然气用气会受各种因素的影响，短期波动较大。LSTM 通过自身强大的非线性学习能力，可以胜任复杂的天然气用气时序数据。

（2）在处理长时序数据时，LSTM 依靠其内部特殊的结构避免了 RNN 那样在时序累乘大于 1 的数之后会梯度爆炸或在时序累乘小于 1 的数之后梯度消失的问题。

2.4 小结

本章从算法基本原理介绍及算法在天然气用气预测中的适用性两方面介绍 ARIMA 算法、灰度关联分析算法 GRA 以及 LSTM 算法，为下一章节各类算法的基本实现打下了理论基础。

3 天然气用气预测算法实现

本章基于上一章的相关理论，在天然气用气量的各种行业的大数据中进行算法实现。包括 ARIMA、ARIMAX、SARIMA、SARIMAX、LSTM、GAR_LSTM、ARIMA 与 LSTM 并联、ARIMA 与 LSTM 串联等各种模型。最终在第 4 章对比考量各类算法的性能。

3.1 数据集的说明及预处理

3.1.1 数据集说明

本数据集存储在 SQL Server 的两个数据库 A，B 中，其中 A 数据库中的每个表记录了该表具在一段时间内的天然气用气量数据，B 数据库说明了 A 数据库中每个表对于的行业信息。有大致五个行业，船舶制造、材料制造、物业管理、学校和生物制药。通过行业类别和表具信息，可以得到每个行业在一段时间内的天然气用气量数据，进而对各个模型在每个行业中进行预测分析。

3.1.2 数据预览

本文所选取的天然气用气数据集中包含多个变量——日期（CreateDate）、工况累计流量(GTotal)、标况累计流量(BTotal)、工况瞬时流量(GTotal)、标况瞬时流量(BTotal)、温度（T）。

G 是工况，也就是实测，B 是标况，计算转化得到。我这里只使用了工况。

图 3-1 为天然气用气表的一部分记录的展示及各个变量的具体取值情况概览：

	CreateDate	GTotal	BTotal	GFlow	BFlow	T	Pa	Err	Alarm
0	2018-08-20 12:21:00	435501.056612	443744.214674	0.0	0.0	23.922302	88.624054	NaN	NaN
1	2018-08-20 13:21:00	435501.056612	443744.214674	0.0	0.0	23.922302	88.624054	NaN	NaN
2	2018-08-20 14:21:00	435501.056612	443744.214674	0.0	0.0	23.922302	88.624054	NaN	NaN
3	2018-08-20 15:21:00	435501.056612	443744.214674	0.0	0.0	23.922302	88.624054	NaN	NaN
4	2018-08-20 16:21:00	435501.056612	443744.214674	0.0	0.0	23.922302	88.624054	NaN	NaN
5	2018-08-20 17:21:00	435501.056612	443744.214674	0.0	0.0	23.922302	88.624054	NaN	NaN

图 3-1 表数据

图 3-2、图 3-3 和图 3-4 可以看出天然气用气量和温度之间有一定的相关性，这就为下面 GRA_LSTM 模型奠定了基础：

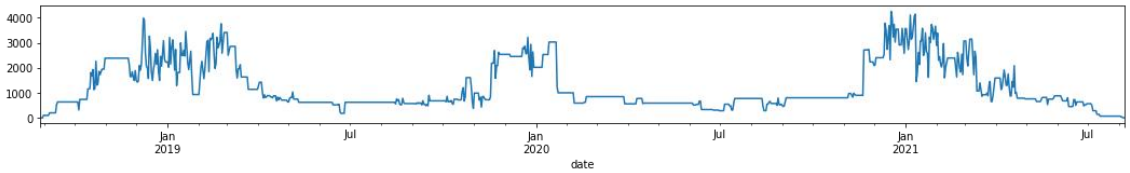

图 3-2 天然气用气量波动曲线图

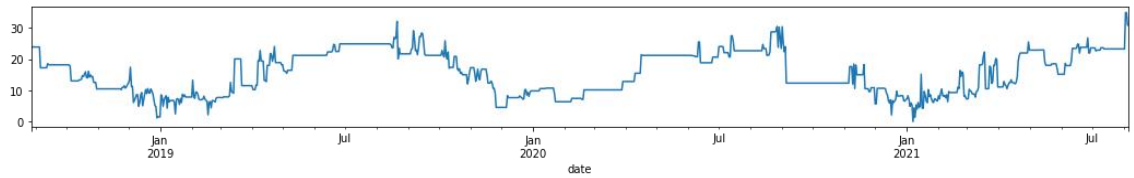

图 3-3 温度波动曲线图

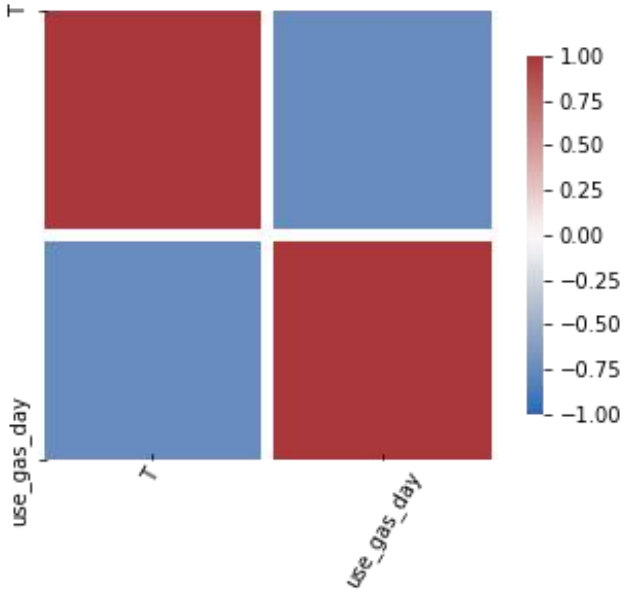

图 3-4 天然气用气量与温度 T 的关联图

3.1.3 数据预处理

机器学习模型本质上是一种能够通过有效识别、提取信息特征并加以挖掘得到规律后做出决策的算法表达。所以，机器学习的天花板由好的数据特征决定，而后面的模型和算法只是逼近这个上限罢了。数据预处理是一种数据挖掘技术，就是为了能够给机器学习模型更好的学习资料，以求达到更好的学习效果。

可是，在现实场景中，数据通常是不完整的，缺少了我们感兴趣的属性值或数据值、极易受各种噪声的影响，导致错误和异常值、存在数据冗余或数据不一致。数据预处理就是解决上面所提到的数据问题的可靠技术。包括一些常用的方法：数据集成、数据清洗、数据归约和数据变换等。下面是对本数据集进行数据预处理的步骤：

Step1:合并每个行业下的数据表，得到该行业的整个数据表

```
#encoding=utf-8
import numpy as np
import pandas as pd
import pymssql

conn = pymssql.connect(host=".",user = "sa",password="",database="CNMCSP_Test",charset="GB18030")
Pos_pd = pd.read_sql("select Code,Post from G_Meter",con = conn)
pos_dict = dict()
pos_set = set()
for i in range(len(Pos_pd)):
```

```
        if(Pos_pd.at[i,'Post'] not in pos_set):
            pos_dict[Pos_pd.at[i,'Post']] = set()
            pos_set.add(Pos_pd.at[i,'Post'])
        pos_dict[Pos_pd.at[i,'Post']].add(Pos_pd.at[i,'Code'])
    conn.close()

    def get_pos_df(pos):
        conn = pymssql.connect(host=".",user = "sa",password="",database="CNMCSPData20210603",charset="GB18030")
        str_select = "select CreateDate,GTotal,T from "
        tableList = []
        for Code in pos_dict[pos]:
            tableList.append(pd.read_sql(str_select+'G_Data'+Code,con = conn))
        sql_source = pd.concat(tableList,ignore_index = True)
        conn.close()
        return sql_source
```

Step2:得到该行业的每天的天然气用气量

```
target = pd.DataFrame(columns = ['date','T','use_gas_day'])
i = 0
j = 0
while i<len(sql_source)-1:
    while(i+1<len(sql_source) and sql_source.at[i,'CreateDate'].date()==sql_source.at[i+1,'CreateDate'].date()):
        i = i+1
    j = i+1
    while(j+1<len(sql_source) and sql_source.at[j,'CreateDate'].date()==sql_source.at[j+1,'CreateDate'].date()):
        j = j+1
    if(sql_source.at[j,'GTotal']-sql_source.at[i,'GTotal']<0):
        break
    if(sql_source.at[j,'CreateDate'].date()-sql_source.at[i,'CreateDate'].date()).days==1
       and sql_source.at[i,'CreateDate'].hour==sql_source.at[j,'CreateDate'].hour
       and 30>=abs(sql_source.at[i,'CreateDate'].minute-sql_source.at[j,'CreateDate'].minute)>=0):
        target = target.append([{
            'date':sql_source.at[j,'CreateDate'].date(),
            'T':sql_source.at[j,'T'],
            'use_gas_day':sql_source.at[j,'GTotal']-sql_source.at[i,'GTotal']}],
            ignore_index = True)
    i = j
```

Step3:处理异常值

```
i = 1
while i<len(target)-1:
    a = target.at[i-1,'use_gas_day']
    b = target.at[i,'use_gas_day']
    c = target.at[i+1,'use_gas_day']
    A = abs(b-a)/max(a,b)
    B = abs(c-b)/max(c,b)
    if(A>0.5 and B>0.5):
        target.at[i,'use_gas_day'] = (a+c)/2
    i = i+1
```

Step4:处理缺省值，得到连续完整的日期序列

```
target = target.reset_index(drop=True).set_index('date')
target.index = pd.DatetimeIndex(target.index)
target = target.groupby('date')['T','use_gas_day'].sum().reset_index()
target = target.set_index('date')
y = target['use_gas_day'].resample('D').mean()
#用下一个非缺失值填充该缺失值
y = y.fillna(y.bfill())

y_T = target['T'].resample('D').mean()
#用下一个非缺失值填充该缺失值
y_T = y_T.fillna(y_T.bfill())
```

Step5:数据归一化

```
#归一化
from sklearn.preprocessing import MinMaxScaler
scaler = MinMaxScaler(feature_range = (0,1))
sca_target['T'] = scaler.fit_transform(sca_target.iloc[:, 0:1])
sca_target['use_gas_day'] = scaler.fit_transform(sca_target.iloc[:, 1:2])
```

Step6:得到训练数据和标签数据并划分训练集和测试集

```
def use_gas_day_LSTM_Data_Precesing(df,timestep):
    features = []
    labels = []
    df = np.array(df)
    date_list = []
    for i in range(timestep, len(sca_target)):
        features.append(df[i-timestep:i,:])
        labels.append(df[i, 1])
        date_list.append(target.index[i])
    features, labels = np.array(features), np.array(labels)
    features = np.reshape(features, (features.shape[0], features.shape[1], -1))
    x_train,x_test,y_train,y_test= features[:900, ],features[900:, ],labels[:900],labels[900:]
    return x_train,x_test,y_train,y_test,date_list
```

3.2 SARIMA 算法实现

3.2.1 SARIMA 模型设计流程

搭建 SARIMA 模型的关键是确定 p,P,q,Q,d和S 6 个参数,其中 d 是原始的非平稳时序数据达到平稳性要求所需要经过一阶趋势差分操作的最少次数，p,P,q,Q 的值是通过观察已达到平稳的时序数据的 ACF (自相关函数)和 PACF (偏自相关函数)的图形是截尾还是拖尾来确定，参数 S 默认为 12，表明是季节性差分自回归滑动平均模型。下面是 SARIMA 模型参数的确定方法和模型设计步骤：

Step1:判断原始时序数据是否具有季节周期性

通过对原始时序数据进行时间序列趋势分解来分析季节周期性。如果其具有良好的季节周期性，适合 SARIMA 模型，否则选用 ARIMA 模型。

面对两个不同的数据序列，进行时间序列趋势分解，分解方法有两种，一种是 additive，另一种是 multiplicative。下面进行 additive 和 multiplicative 分解，如

图 3-8 和图 3-9。

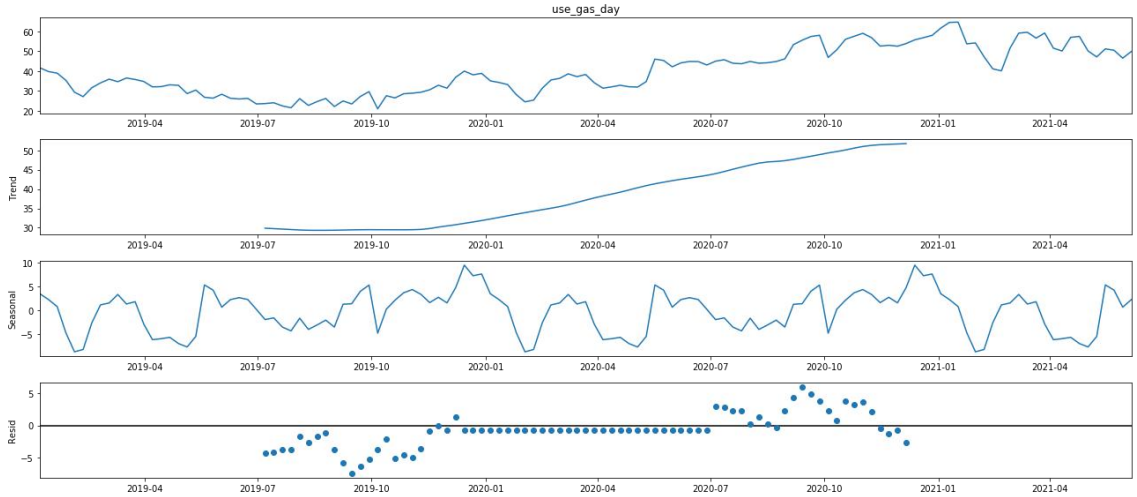

图 3-8 additive 分解图

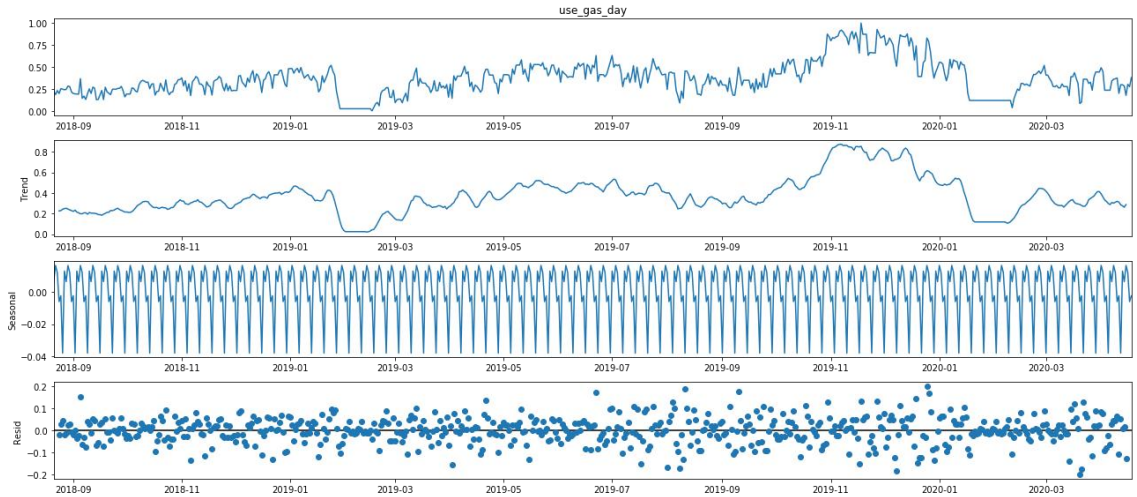

图 3-9multiplicative 分解图

图 3-8 和图 3-9 说明，对于上面每张图，包含四张子图。第一张图是初始序列 $\mathrm{L}_i(i=1,2,...,n)$，第二张图是剔除季节周期性后的趋势图 $\mathrm{T}_i(i=1,2,...,n)$，第三张子图是季节周期图， $\mathrm{S}_i(i=1,2,...,n)$ 第四张子图是残差图 $\mathrm{R}_i(i=1,2,...,n)$。

①如果是 additive 分解，那么满足， $\mathrm{L}_i = T_i + S_i + R_i.$

②如果是 multiplicative 分解，那么满足， $\mathrm{L}_i = T_i * S_i * R_i.$

对于第一张图的第三张子图，也即季节周期图，最大值是 5，相对于原始序列平均值 40，有较大的影响。说明第一张图，对于的数据序列，有良好的季节周期性。

③对于第二张图的第三张子图，也即季节周期图，最大值是 0.04，相对于原始序列平均值 1，几乎没有影响。说明第二张图，对于的数据序列，没有良好的季节周期性。

Step2: 判断数据平稳性

在经验上，可以画出原始数据的时序图进行初步判断，如图 3-5。在数学上，

可以使用单位根检验，来严格验证是否平稳。

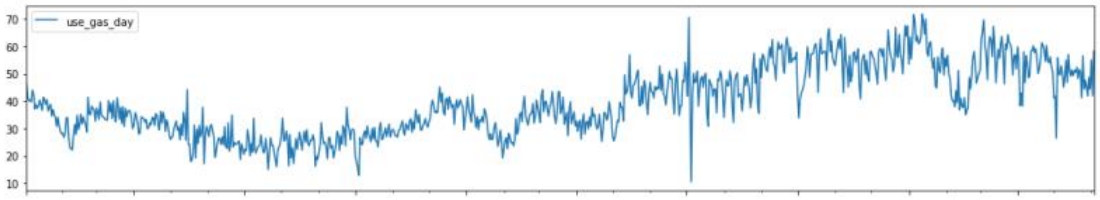

图 3-5 天然气用气量时间序列曲线图

以下是对上图进行单位根检验的结果：

①没做差分的初始数据的单位根检验：

(-1.3519814855118628, 0.6050682572944464, 21, 862, {'1%': -3.437958909 7679975, '5%': -2.86489877693665, '10%': -2.568558467170181}, 5209.262 880783307)

---

②一阶差分的单位根检验:

(-7.401900442793181, 7.51689402612888e-11, 21, 861, {'1%': -3.43796777 36185514, '5%': -2.8649026847264074, '10%': -2.568560548763626}, 5196. 093673880159)

参数 i，是τ统计量的值；参数 ii，是 p_value 的值；参数 iii，是结果使用的延迟阶数；参数 iv，是 ADF 回归和计算临界值所使用的观察次数；参数 v，是临界值；参数 vi，是最大信息准则的值，也就是 AIC 或许 BIC 的值。

当我们看序列是否平稳的结果时，一般首先看第二部分的 p_value 值。如果 p_value 值比 0.05 小，证明有单位根，也就是说序列平稳。如果 p_value 比 0.05 大则证明非平稳。

如果 p_value 接近于 0.05 时，则要通过临界值进行判断。也就是说如果 p_value 接近于 0.05 就要通过第一部分τ统计量的值和第五部分的临界值进行对比。τ统计量的值比临界值小，就证明平稳，反之就是非平稳。这里的 1%，5%，10%对应的是 99%，95%，90%置信区间。

第一个单位根检验中，p_value 大于 0.05。则原始数据是非平稳的。说明要对其进行一次或多次差分操作，以保证数据是平稳的。

第二个单位根检验中，p_value 小于 0.05。则对原始非平稳的数据序列进行一阶差分过后，平稳了。

Step3: 数据差分

预测的原始时序数据一般是不平稳的，具有趋势性和周期性。如果对原始时序数据进行单位根检验发现其是非平稳的，则需要对其进行数据差分来消除趋势性和周期性达到平稳性要求。如果当前序列已经平稳，无需在进行差分。数据差分分为趋势差分和周期差分，趋势差分是消除原始时序数据随时间的变化趋势性，

周期差分是消除原始时序数据的周期性。参数 d 是最少经过几阶趋势差分才能消除原始时序数据的趋势性，参数 S 是经过几阶季节差分才能消除原始时序数据的季节周期性，最终参数 d 和 S 共同使得原始时间序列达到平稳性的要求。

对原始数据进行差分后的可视化，如图 3-6。其中，第一张子图，是初始数据序列；第二张子图，是经过一阶差分过后的数据序列；第三张子图，是经过二阶差分过后的数据序列。

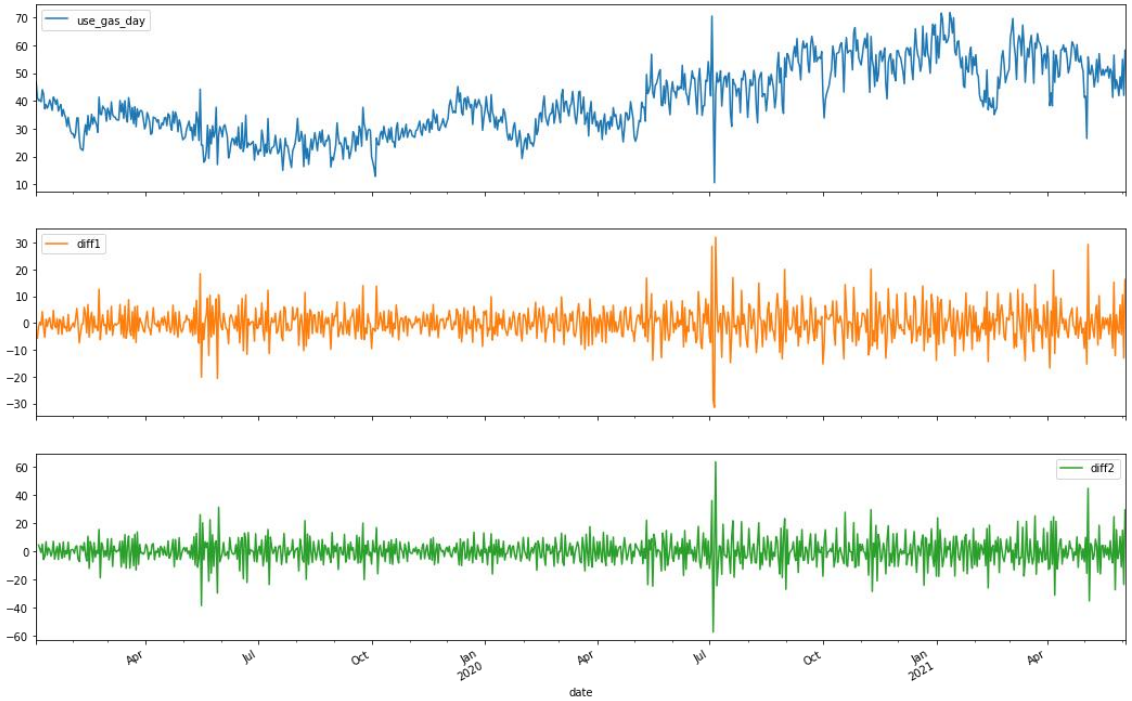

图 3-6 原始时间序列、一阶差分和二阶差分后的序列曲线图

Step4: 确定 p,P,q,Q 的值

通过 ACF 和 PACF 图并结合表 2-1 来确定 p 和 q 。

画出经过参数 d 消除趋势性后的数据序列的 ACF 和 PACF 图，如图 3-7：

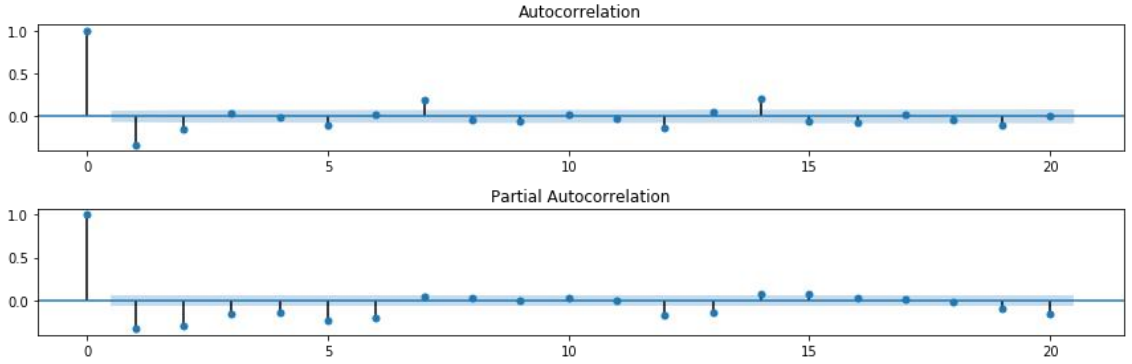

图 3-7 ACF 和 PACF 图

可以，看出 ACF 和 PACF 都是拖尾。对于 PACF,当下标是 7 时，衰减到了置信区间内，所以可设 p 的值为 7。对于 ACF,当下标是 2 时，衰减到了置信区间内，所以可设 q 的值为 2。

同理，画出经过参数S消除季节周期性后的数据序列的ACF和PACF图，并结合表2-1来确定P和Q。

Step5: 模型确定

得到p,P,q,Q,d和S 6个参数后，也就确定最终模型。

3.2.1 SARIMA模型代码实现

```
mod = sm.tsa.statespace.SARIMAX(y,
                                exog = y_T,
                                order=(1,1,1),
                                seasonal_order=(0,1,1,12),
                                enforce_stationarity=False,
                                enforce_invertibility=False)
results = mod.fit()

print(results.summary().tables[1])

results.plot_diagnostics(figsize=(15, 12))
plt.show()
```

3.3 GRA_LSTM算法实现

3.3.1 GRA_LSTM模型设计流程

GRA_LSTM模型是LSTM模型优化版本，它考虑到了其它外生变量（如，温度T）对天然气使用量的影响。在本案例中，如果选取的外生变量与天然气使用量这个自变量有较大的关联度，那么考虑这个外生变量将会辅助LSTM做出更好的预测。其中，GRA技术就是一种通过用灰色关联度来分析各个外生变量与内生变量的关联度，进而确定辅助变量的方法。可被用于分析各种外生变量与天然气使用量自身的内生变量之间关联程度，从而删选出影响天然气使用量的关键外生变量，最终模型的输入为原生变量和利用GRA法在外生变量中筛选出的辅助变量的融合。

天然气用气量预测的具体步骤如下:

（1）补全缺失值，排除异常值，对数据集进行归一化。

（2）利用GRA，计算天然气用气与其他外生变量（如，温度T）的相关性。

（3）选取一组外生变量作为辅助变量集合U，其相关系数绝对值较大。然后，构造训练集D和测试集T。

（4）建立基于LSTM的天然气用气量预测模型，用D训练模型直到模型的损失函数是收敛的。

（5）将测试集T输入到模型中，得到该时间段下的预测值。

（6）根据测试结果，评估模型，并不断调整模型参数，直至模型最优化。

最后，基于GRA和LSTM的天然气用气量预测方法流程如图3-10所示。首先对原始数据进行补全缺失值、排除异常值等一系列数据预处理得到要预测的时

序数据 $y_i(i=1,2,...,n)$。通过 GRA 技术，计算其它外生变量（如，温度 T）与天然气用气量的相关性，将相关度高于特定阈值 σ 的外生变量筛选为辅助变量，得到天然气用气量的辅助变量集合 U 。然后将辅助变量 u 的时序数据 $u_i(i=1,2,...,n)$ 与 y_i 进行合并融合得到新的时序数据 $y_i(i=1,2,...,n)$。将 y_i 进行数据归一化，并根据时间段划分出训练集 D 和测试集 T 。用训练集 D 训练 LSTM 模型直至模型收敛，然后验证测试集 T 得到预测结果，最后通过分析预测结果来调整模型参数，直至模型最优化。

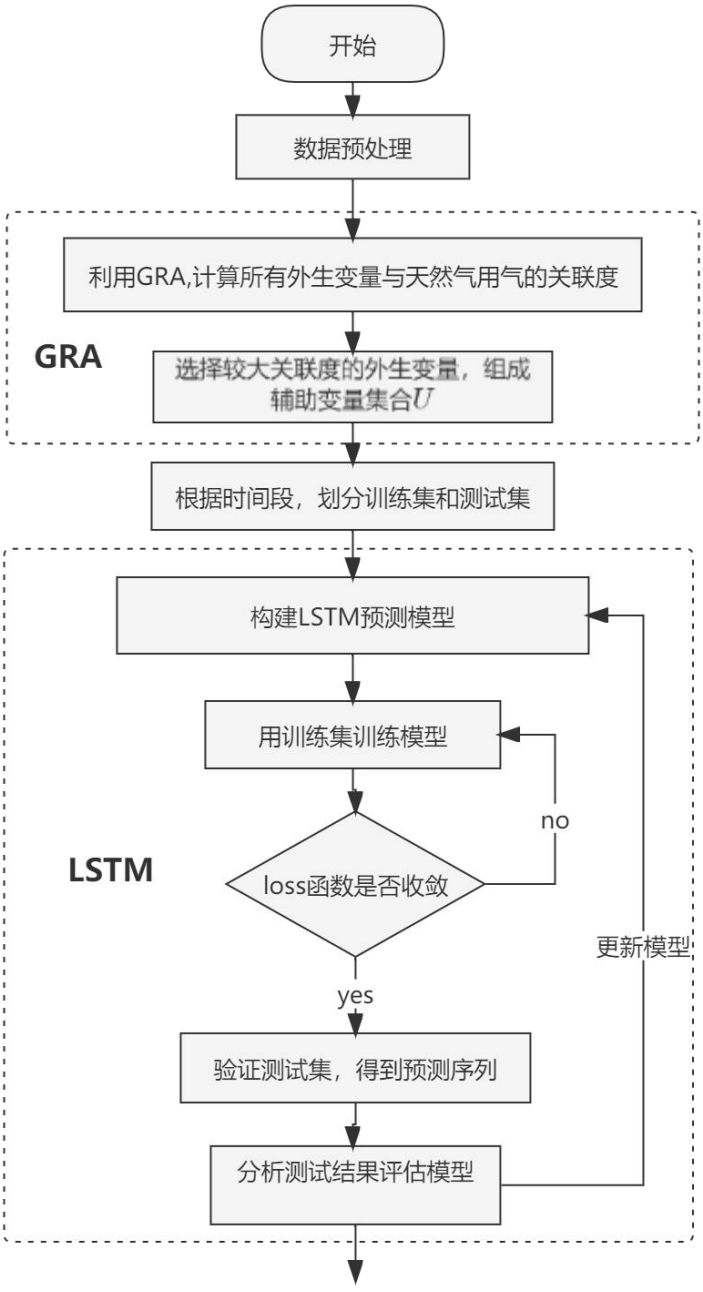

图 3-10 GRA_LSTM 预测方法流程图

3.3.2 GRA_LSTM 模型代码实现

```
mem_days = [4]
lstm_layers = [0]
dense_layers = [1]
units = [52]
from tensorflow.keras.callbacks import ModelCheckpoint
for the_mem_days in mem_days:
    for the_lstm_layers in lstm_layers:
        for the_dense_layers in dense_layers:
            for the_units in units:
                filepath = './mdel_lstm_T/{val_mape:.2f}_{epoch:02d}_'+
                f'men_{the_mem_days}_lstm_{the_lstm_layers}_dense_{the_dense_layers}_unit_{the_units}'
                checkpoint = ModelCheckpoint(
                    filepath=filepath,
                    save_weights_only=False,
                    monitor='val_mape',
                    mode='min',
                    save_best_only=True
                )
                history = LossHistory()
                x_train,x_test,y_train,y_test,date_list = use_gas_day_LSTM_Data_Precesing(sca_target,the_mem_days)
                from tensorflow.keras.models import Sequential
                from tensorflow.keras.layers import LSTM,Dense,Dropout
                model = Sequential()
                model.add(LSTM(the_units,input_shape = (x_train.shape[1], x_train.shape[2]),activation='relu'))
                model.add(Dropout(0.1))
                for i in range(the_lstm_layers):
                    model.add(LSTM(the_units,activation='relu',return_sequences=True))
                    model.add(Dropout(0.1))
                model.add(LSTM(the_units,activation='relu'))
                model.add(Dropout(0.1))
                for i in range(the_dense_layers):
                    model.add(Dense(the_units))
                    model.add(Dropout(0.1))
                model.add(Dense(1))
                model.compile(optimizer='adam',loss='mse',metrics=['mape'])
                model.fit(x_train,y_train,batch_size = 32,epochs = 300,shuffle = True,
                validation_data=(x_test,y_test),callbacks = [checkpoint,history])
```

3.4 SARIMA_LSTM 并联算法实现

3.4.1 并联组合模型设计流程

并联组合模型设计思路为，在 SARIMA 和 LSTM 单模型后额外添加一个组合层。首先分别用 SARIMA 模型和 LSTM 模型对时序序列进行预测得到预测序列 S 和 L。如果预测序列具有很好的季节周期性，那么 SARIMA 模型得到的预测序列 S 就能提取出良好的季节周期性，使得其在局部上比 L 有更高的吻合度。同时利用 LSTM 本身相对于 SARIMA 的预测精度高的优点，可知预测序列 L 在整体上比 S 具有更高的吻合度。最后，对 SARIMA 进行整体上的比例修正，以达到保留局部优势扩大全局优势的目的。设计思路为:

（1）LSTM 单模型预测

将内生变量的序列数据以及根据 GRA 法确定的辅助变量的序列数据融合后输入到 LSTM 模型，得到 LSTM 单模型预测结果 $L_i(i=1,2,...,n)$。其中，n 为预测时长。

（2）SARIMA 单模型预测

将预测时序数据输入到 SARIMA 模型，得到 SARIMA 单模型预测结果 $S_i(i=1,2,...,n)$。其中，n 为预测时长。

（3）组合设计

组合设计步骤如下：

①首先根据 LSTM 模型初步预测结果，得到总时间段预测结果数值总合 $L_{总}$：

$$L_{总}=\sum_{i=1}^{n}L_i. \tag{3.1}$$

②对于 SARIMA 单模型预测结果 $S_i(i=1,2,...,n)$，也得到总时间段预测结果数值总合 $S_{总}$：

$$S_{总}=\sum_{i=1}^{n}S_i. \tag{3.2}$$

③保留 SARIMA 单模型预测结果 $S_i(i=1,2,...,n)$ 的比例信息，用 $L_{总}/S_{总}$ 对 SARIMA 进行整体上的比例修正，以达到保留局部优势扩大全局优势的目的。f_i 的完整表达式为：

$$f_i=\frac{L_{总}}{S_{总}}S_i \tag{3.3}$$

最后并联组合模型预测方法的处理步骤如 3-11 图所示。在输入层中，首先经过数据预处理得到满足 LSTM 模型和 SARIMA 模型输入要求的内生变量的时序数据，并直接将其作为 SARIMA 的输入。而后，需要通过 GRA 技术在外生变量中筛选出辅助变量，将辅助变量时序数据和内生变量时序数据融合，一同作为 LSTM 的输入。在模型层中，LSTM 和 SARIMA 分别拿到自己的时序数据后，各自进行工作并得到初步的预测结果 $L_i(i=1,2,...,n)$ 和 $S_i(i=1,2,...,n)$。在组合层中，分别对 LSTM 和 SARIMA 的预测结果时序数据进行每个节点的求和得到总时间段预测结果数值总合 $L_{总}$ 和总时间段预测结果数值总合 $S_{总}$。然后，用 $L_{总}/S_{总}$ 对 SARIMA 单模型预测结果 $S_i(i=1,2,...,n)$ 的比例信息进行整体上的比例修正得到 $f_i(i=1,2,...,n)$。在输出层中，输出最终的预测数据 $f_i(i=1,2,...,n)$。

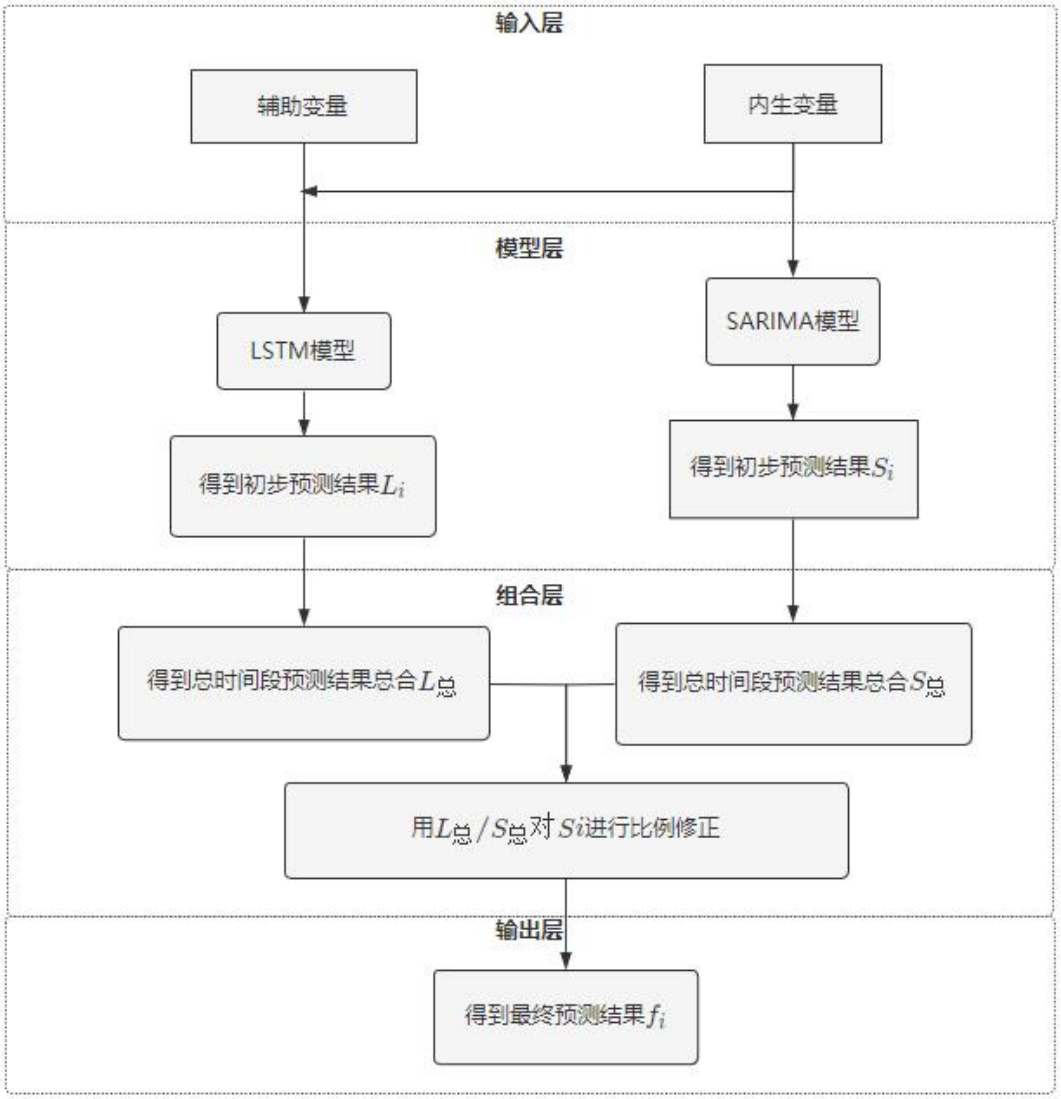

图 3-11 并联模型预测方法流程图

3.4.2 并联组合模型代码实现

```
total_val = 0
for one in y_forecasted:
    total_val=total_val+one
for i in range(len(y_forecasted)):
    y_forecasted[i] = y_forecasted[i]*total_val/156720.08
pred = results.get_prediction(start = pd.to_datetime('2021-2-17'), dynamic=False)
pred_ci = pred.conf_int()
ax = y['2020':].plot(label = 'Observed age_use')
y_forecasted.plot(ax=ax, label='Forecast age_use', alpha=.7, figsize=(15, 4))
ax.fill_between(pred_ci.index,
                pred_ci.iloc[:, 0],
                pred_ci.iloc[:, 1], color='k', alpha= 0.15)

ax.set_xlabel('Date')
ax.set_ylabel('Day Consumption')
plt.legend()
```

```
plt.show()
```

3.5 ARIMA_LSTM 串联联算法实现

3.5.1 串联组合模型设计流程

串联组合模型,即将 ARIMA 的初步预测结果作为 GRA_LSTM 的输入。通过 ARIMA 模型将初始预测序列的周期、趋势信息做初步的预提取，然后在经过 LSTM 强大的学习能力得到最终预测结果。

（1）补全缺失值，排除异常值

（2）ARIMA 单模型预测

将整个时间段天然气用气量序列数据输入到 ARIMA 模型中，并初步得到整个时间段的预测序列 $S_i(i=1,2,...,n)$。其中，n 为整个时间段，最小时间节点个数。

（3）利用 GRA, 计算天然气用气与其它因素（如，温度 T）的相关性。并选取一组因素 U，其相关系数绝对值较大。

（4）ARIMA 得到的初始预测序列替代原始天然气用气量数据序列。

（5）将 S_i 与 U 合并，得到 LSTM 的待输入序列 $y_i(i=1,2,...,n)$。

（6）数据归一化

（7）构造训练集 D 和测试集 T。

（8）建立基于 LSTM 的天然气用气量预测模型，用 D 训练模型直到模型的损失函数是收敛的。

（9）将测试集 T 输入到模型中，得到该时间段下的预测值。

（10） 根据测试结果，评估模型。

基于 ARIMA 与 GRA_LSTM 的串联模型的天然气用气量预测方法流程如图 3-12 所示。首先原始数据经过补全缺失值、排除异常值等一系列数据预处理，得到要预测的时序数据 $L_i(i=1,2,...,n)$。将时序数据 L_i 作为 ARIMA 模型的输入得到初步的预测结果 $S_i(i=1,2,...,n)$，并取代 L_i。通过 GRA 技术，计算其它外生变量（如，温度 T）与天然气用气量的相关性，将相关度高于特定阈值 σ 的外生变量筛选为辅助变量，得到天然气用气量的辅助变量集合 U。然后将辅助变量 u 的时序数据 $u_i(i=1,2,...,n)$ 与更新后的时序数据 $L_i(i=1,2,...,n)$ 进行合并融合得到新的时序数据 $y_i(i=1,2,...,n)$。将 y_i 进行数据归一化，并根据时间段划分出训练集 D 和测试集 T。用训练集 D 训练 LSTM 模型直至模型收敛，然后验证测试集 T 得到预测结果，最后通过分析预测结果来调整模型参数，直至模型最优化。

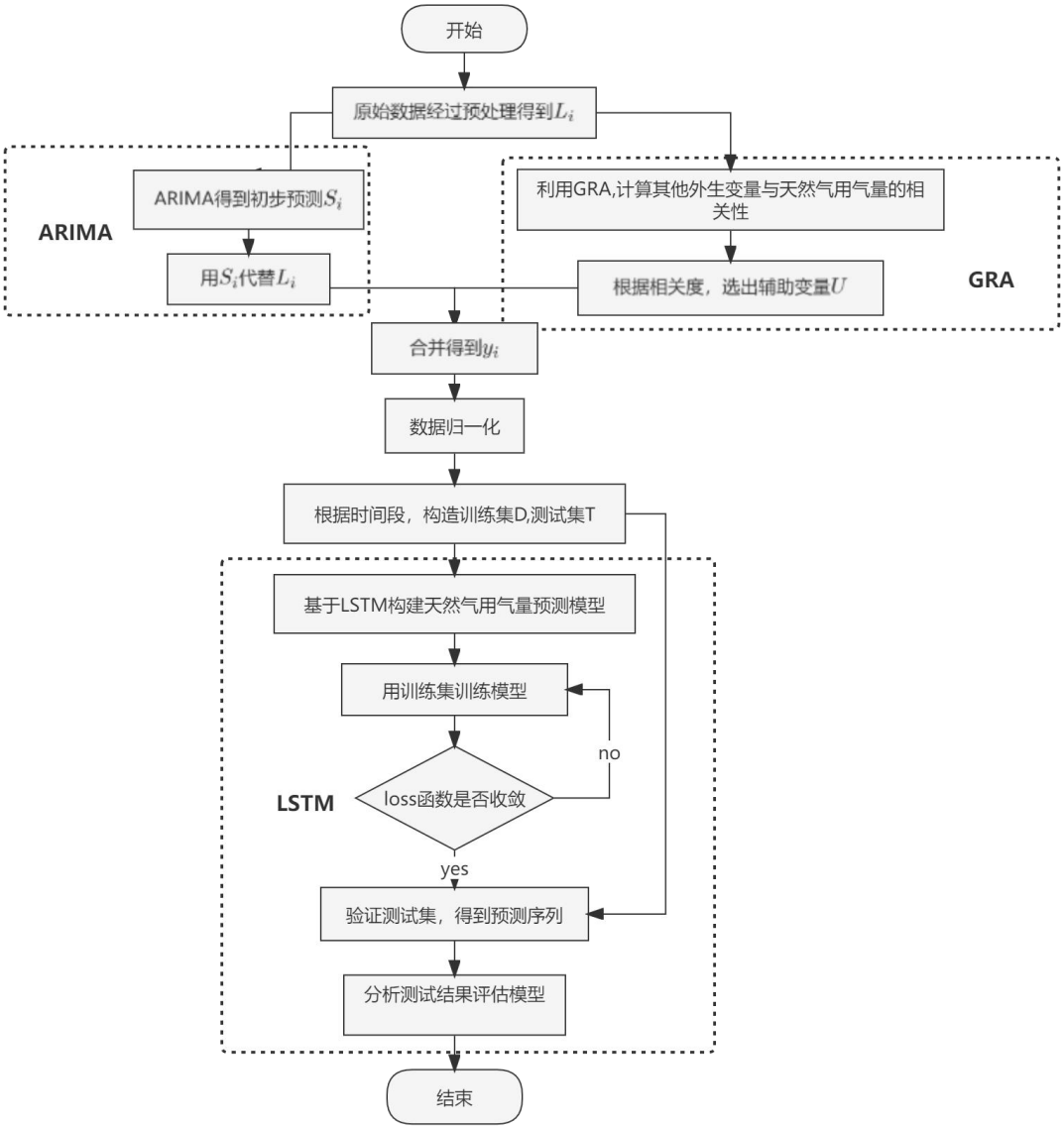

图 3-12 串联模型预测方法流程图

3.5.2 串联组合模型代码实现

```
###ARIMA 模型处理
mod = sm.tsa.statespace.SARIMAX(y,
#                                 exog = y_T,
                                order=(1,1,1),
                                seasonal_order=(0,0,0,0),
                                enforce_stationarity=False,
                                enforce_invertibility=False)
results = mod.fit()
### 将 ATRIMA 预测结果作为 LSTM 的输入
pred = results.get_prediction(start = pd.to_datetime('2018-8-28'), dynamic=False)
y_forecasted = pred.predicted_mean
target['y_forecasted'] = y_forecasted

def use_gas_day_LSTM_Data_Precesing(df,timestep):
    features = []
    labels = []
```

```
    df = np.array(df)
    date_list = []
    for i in range(timestep, len(target)):
        features.append(df[i-timestep:i,1:])
        labels.append(df[i, 0])
        date_list.append(target.index[i])
    features, labels = np.array(features), np.array(labels)
    features = np.reshape(features, (features.shape[0], features.shape[1], -1))
    x_train,x_test,y_train,y_test= features[:900, ],features[900:, ],labels[:900],labels[900:]
    return x_train,x_test,y_train,y_test,date_list
###LSTM 处理
mem_days = [4]
lstm_layers = [2]
dense_layers = [1]
units = [52]
from tensorflow.keras.callbacks import ModelCheckpoint
for the_mem_days in mem_days:
    for the_lstm_layers in lstm_layers:
        for the_dense_layers in dense_layers:
            for the_units in units:

                filepath = './mdel_arima_lstm_T_/{val_mape:.2f}_{epoch:02d}_'
                +f'men_{the_mem_days}_lstm_{the_lstm_layers}_dense_{the_dense_layers}_unit_{the_units}'
                checkpoint = ModelCheckpoint(
                    filepath=filepath,
                    save_weights_only=False,
                    monitor='val_mape',
                    mode='min',
                    save_best_only=True
                )
                history = LossHistory()
                x_train,x_test,y_train,y_test,date_list = use_gas_day_LSTM_Data_Precesing(target,the_mem_days)

                from tensorflow.keras.models import Sequential
                from tensorflow.keras.layers import LSTM,Dense,Dropout
                model = Sequential()
                model.add(LSTM(the_units,input_shape = (x_train.shape[1], x_train.shape[2]),activation='relu')
)
                model.add(Dropout(0.1))
                for i in range(the_lstm_layers):
                    model.add(LSTM(the_units,activation='relu',return_sequences=True))
                    model.add(Dropout(0.1))
                model.add(LSTM(the_units,activation='relu'))
                model.add(Dropout(0.1))
                for i in range(the_dense_layers):
                    model.add(Dense(the_units))
                    model.add(Dropout(0.1))
                model.add(Dense(1))
                model.compile(optimizer='adam',loss='mse',metrics=['mape'])
                model.fit(x_train,y_train,batch_size = 32,epochs = 300,
                shuffle = True,validation_data=(x_test,y_test),callbacks = [checkpoint,history])
```

4 模型性能对比

本章在上一章预测算法实现的基础之上，对各个模型分别在船舶制造、材料制造、物业管理、学校和生物制药等五大行业的数据中进行充分的性能测试，以对比各类模型的性能。

4.1 回归评价指标

各个模型在各个行业的预测效果需要经过回归评价指标进行判定，常见的回归评价指标有：均方误差、均方根误差、平均绝对值误差和 R 平方等。

（1）MSE(Mean Squared Error) 叫做均方误差。公式如下：

$$\frac{1}{m}\sum_{i=1}^{m}\left(y_{test}^{(i)}-\hat{y}_{test}^{(i)}\right)^2. \tag{4.1}$$

这里，$y_{test}^{(i)}$ 表示测试集上的真实值，$\hat{y}_{test}^{(i)}$ 表示测试集上的预测值。

（2）RMSE(Root Mean Squared Error)叫做均方根误差。公式如下：

$$\sqrt{\frac{1}{m}\sum_{i=1}^{m}\left(y_{test}^{(i)}-\hat{y}_{test}^{(i)}\right)^2}=\sqrt{\mathrm{MSE}_{test}}. \tag{4.2}$$

这里，$y_{test}^{(i)}$ 表示测试集上的真实值，$\hat{y}_{test}^{(i)}$ 表示测试集上的预测值。

（3）MAE(Mean Absolute Error)叫做平均绝对值误差。公式如下：

$$\frac{1}{m}\sum_{i=1}^{m}\left|y_{test}^{(i)}-\hat{y}_{test}^{(i)}\right|. \tag{4.3}$$

这里，$y_{test}^{(i)}$ 表示测试集上的真实值，$\hat{y}_{test}^{(i)}$ 表示测试集上的预测值。

（4）R-Squared 叫做 R 平方。公式如下：

$$R^2=1-\frac{\sum_i\left(\hat{y}^{(i)}-y^{(i)}\right)^2}{\sum_i\left(\bar{y}-y^{(i)}\right)^2}=1-\frac{\left(\sum_{i=1}^{m}\left(\hat{y}^{(i)}-y^{(i)}\right)^2\right)\div m}{\left(\sum_{i=1}^{m}\left(y^{(i)}-\bar{y}\right)^2\right)\div m}=1-\frac{MSE\left(\hat{y},\ y\right)}{\mathrm{var}(y)} \tag{4.4}$$

这里，$y_{test}^{(i)}$ 表示测试集上的真实值，$\hat{y}_{test}^{(i)}$ 表示测试集上的预测值。

根据 R^2 的取值，来判断模型的好坏，其值越大表明模型的拟合效果越好。一般来说，$R^2>0.8$，表示模型有良好的拟合效果。

4.2 各模型评估结果

对各个模型在各个行业进行预测分析，并用各种回归评价指标对各个预测结果进行性能评估。见表 4-1、表 4-2、表 4-3、表 4-4 和表 4-5。

（1）对各个模型在船舶制造行业进行预测分析，评估结果见表 4-1。通过对比观察可以看出：

① 类 LSTM 模型明显优于类 ARIMA 模型，表明了 LSTM 在本数据集下有更强的学习能力。

② 各个模型的 R-Squared 值大致为 0.7 低于但十分接近 0.8，表明预测拟合效果略差。

③ 在本数据集下，GRA_LSTM 模型的性能最好，表明外生变量温度与天然气用气量有很好的相关性，且借助于 LSTM 的强大学习能力，达到了不错的效果。

④ 但是类 ARIMA 模型在加上辅助变量后，效果却普遍略差。

⑤ 不带季节周期性的 ARIMA 和 ARIMAX 都比带了季节周期性的 SARIMA 和 SARIMAX 效果要好，表明本数据集没有良好的季节周期性。

⑥ 由于本数据集没有良好的的季节周期性，导致 SARIMA+LSTM 并联模型表现最差。

表 4-1 各模型在船舶制造行业中的评估结果

船舶制造行业	MSE	RMSE	MAE	R-Squared	Mean-Val
ARIMA	33676.49	183.51	131.50	0.709	862.83
ARIMAX	34136.25	184.75	132.09	0.705	862.83
SARIMA	35577.99	188.62	136.02	0.693	862.83
SARIMAX	36039.85	189.84	136.63	0.689	862.83
LSTM	35563.10	188.58	126.77	0.693	862.83
GRA_LSTM	**31142.18**	**176.47**	**114.36**	**0.731**	**862.83**
SARIMA+GRA_LST 并联	39489.59	198.71	142.87	0.659	862.83
ARIMA+GRA_LSTM 串联	32744.58	180.95	136.30	0.717	862.83

（2）对各个模型在材料制造行业进行预测分析，评估结果见表 4-2。通过对比观察可以看出：

① SARIMA 模型在本数据集下表现最好。

② 各个模型的 R-Squared 值大致为 0.7 低于但十分接近 0.8，表明预测拟合效果略差。

③ 带季节周期性的 SARIMA 和 SARIMAX 都比带了不带季节周期性的 ARIMA 和 ARIMAX 效果要好，表明本数据集有良好的季节周期性。

④ 由于本数据集有良好的的季节周期性，使得并联和串联模型表现都要好于没有季节周期性的 LSTM 模型和 GRA_LSTM 模型。

表 4-2 各模型在材料制造行业中的评估结果

材料制造行业	MSE	RMSE	MAE	R-Squared	Mean-Val
ARIMA	487.45	22.078	5.13	0.774	473.44
ARIMAX	508.17	22.54	5.38	0.765	473.44
SARIMA	**468.57**	**21.64**	**9.97**	**0.783**	473.44
SARIMAX	484.17	22.00	10.10	0.776	473.44
LSTM	701.13	26.47	14.26	0.676	473.44
GRA_LSTM	540.69	23.25	12.52	0.750	473.44
SARIMA+GRA_LSTM 并联	512.41	22.63	10.83	0.763	473.44
ARIMA+GRA_LSTM 串联	532.94	23.08	11.03	0.753	473.44

（3）对各个模型在物业管理行业进行预测分析，评估结果见表 4-3。通过对比观察可以看出：

① 类 ARIMA 模型明显优于类 LSTM 模型，表明了 ARIMA 在本数据集下有更强的预测拟合能力。

② 在本数据集下，SARIMAX 模型的性能最好。其 R-Squared 数值达到 0.916，表明达到了接近完美的预测拟合度。

③ 对比发现，表明外生变量温度与天然气用气量有很好的相关性，使得 ARIMAX 和 SARIMAX 的预测效果优于 ARIMA 和 SARIMA;GRA_LSTM 的预测效果优于 LSTM。

表 4-3 各模型在物业管理行业中的评估结果

物业管理行业	MSE	RMSE	MAE	R-Squared	Mean-Val
ARIMA	4915554	2217.10	420.25	0.901	3164.14
ARIMAX	4688520	2154.29	422.28	0.905	3164.14
SARIMA	4284437	2069.88	400.96	0.913	3164.14
SARIMAX	**4170980**	**2042.29**	**412.66**	**0.916**	**3164.14**
LSTM	13442580	3666.41	1054.96	0.729	3164.14
GRA_LSTM	11207978	3347.83	978.41	0.774	3164.14
SARIMA+GRA_LSTM 并联	7512247	2740.84	1039.29	0.848	3164.14
ARIMA+GRA_LSTM 串联	16576581	4071.43	1231.06	0.666	3164.14

（4）对各个模型在学校行业进行预测分析，评估结果见表 4-4。通过对比观察可以看出：

① 各个模型的 R-Squared 值大致相当且都大于 0.8，表明各个模型在本数据集下性能相当，都具有良好的预测效果。

② SARIMA 模型在本数据集下表现最好。

③ 在本数据集下，GRA_LSTM 模型的性能优于 LSTM，表明外生变量温度与天然气用气量有很好的相关性。

④ 不带季节周期性的 ARIMA 和 ARIMAX 和带了季节周期性的 SARIMA 和 SARIMAX 预测效果相当，表明本数据集没有明显的季节周期性。

表 4-4 各模型在学校行业中的评估结果

学校行业	MSE	RMSE	MAE	R-Squared	Mean-Val
ARIMA	65.35	8.08	6.03	0.862	50.15
ARIMAX	64.97	8.06	6.01	0.862	50.15
SARIMA	**64.33**	**8.02**	**6.00**	**0.864**	**50.15**
SARIMAX	66.04	8.12	6.14	0.860	50.15
LSTM	78.75	8.87	6.57	0.833	50.15
GRA_LSTM	67.74	8.23	5.96	0.857	50.15
SARIMA+GRA_LSTM 并联	66.56	8.15	6.17	0.859	50.15
ARIMA+GRA_LSTM 串联	86.36	9.29	6.84	0.817	50.15

（5）对各个模型在生物制药行业进行预测分析，评估结果见表 4-5。通过对比观察可以看出：

① 各个模型的 R-Squared 值大致相当且都大于 0.8，表明各个模型在本数据集下性能相当，都具有良好的预测效果。

② ARIMAX 和 SARIMAX 的预测效果优于 ARIMA 和 SARIMA，且 GRA_LSTM 的预测效果和 LSTM 相当（可能是模型学习的偏差），表明外生变量温度与天然气用气量有很好的相关性。

③ 在本数据集下，ARIMAX 模型的性能最好

④ 不带季节周期性的 ARIMA 和 ARIMAX 都比带了季节周期性的 SARIMA 和 SARIMAX 效果要好，表明本数据集没有良好的季节周期性。

⑤ 由于本数据集没有良好的的季节周期性，导致 SARIMA+LSTM 并联模型表现最差。

表 4-5 各模型在生物制药行业中的评估结果

生物制药行业	MSE	RMSE	MAE	R-Squared	Mean-Val
ARIMA	77056	277.59	157.72	0.862	959.82
ARIMAX	**69758**	**264.11**	**152.09**	**0.875**	**959.82**
SARIMA	78825	280.75	169.42	0.859	959.82
SARIMAX	72500	269.25	165.43	0.870	959.82
LSTM	77897	279.10	173.57	0.861	959.82
GRA_LSTM	79145	281.32	167.78	0.859	959.82
SARIMA+GRA_LSTM 并联	96906	311.29	195.25	0.827	959.82
ARIMA+GRA_LSTM 串联	79605	282.14	169.86	0.858	959.82

4.3 部分模型在生物制药行业的部分预测趋势图

以生物制药行业为例，通过可视化，直观上看各模型拟合效果。

⑴ 图 4-1 为 ARIMA 模型在测试集上的预测趋势图，其中蓝色线条代表真实数据、橙色线条代表预测数据。

① 可以看出，在生物制药行业中，ARIMA 模型可以测试集中取得不错的拟合效果。

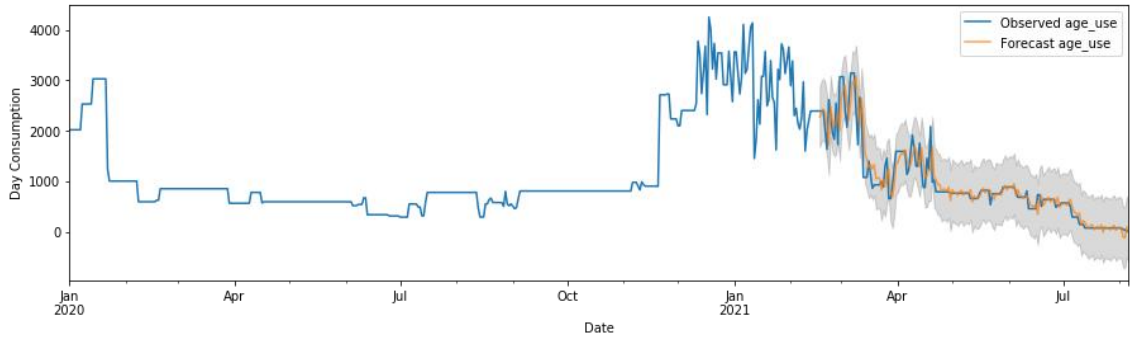

图 4-1 ARIMA 预测图

⑵ 图 4-2 和图 4-3 分别为 LSTM 模型在训练集和测试集上的预测趋势图，其中红色线条代表真实数据、蓝色线条代表预测数据。

① 可以看出，在生物制药行业中，LSTM 模型可以训练集中取得不错的拟合效果，但是在数据特别突兀的小时间段内无法完全拟合，并且预测值普遍偏小。

② 可以看出，在生物制药行业中，LSTM 模型可以测试集中取得不错的拟合效果，但是在整体上的预测数值偏低于真实数值且但总体会有滞后。

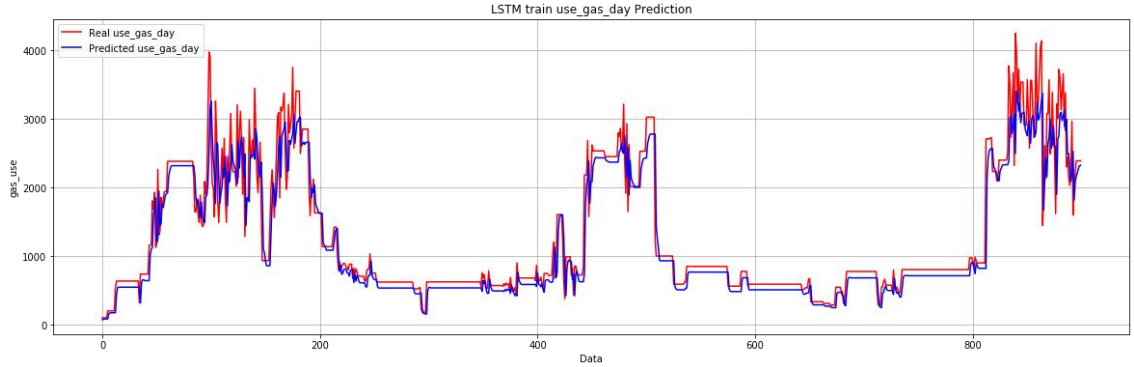

图 4.2 LSTM 在训练集上的预测趋势图

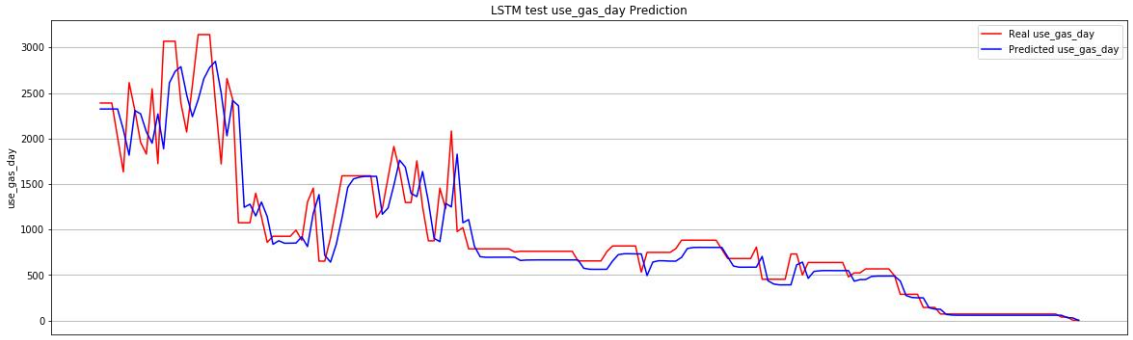

图 4.3 LSTM 在测试集上的预测趋势图

⑶ 图 4-4 和图 4-5 分别为 GRA_LSTM 模型在训练集和测试集上的预测趋势图，其中红色线条代表真实数据、蓝色线条代表预测数据。

① 可以看出，在生物制药行业中，GRA_LSTM 模型可以训练集中取得非常不错的拟合效果，在细节上比 LSTM 有更好的拟合效果，而且没有图 4-2 中 LSTM 的预测结果普遍偏小的情况，但是在数据特别突兀的小时间段内也是无法完全拟合。

② 可以看出，在生物制药行业中，GRA_LSTM 模型可以测试集中取得不错的拟合效果，但是在整体上的预测数值偏低于真实数值且但总体会有滞后。

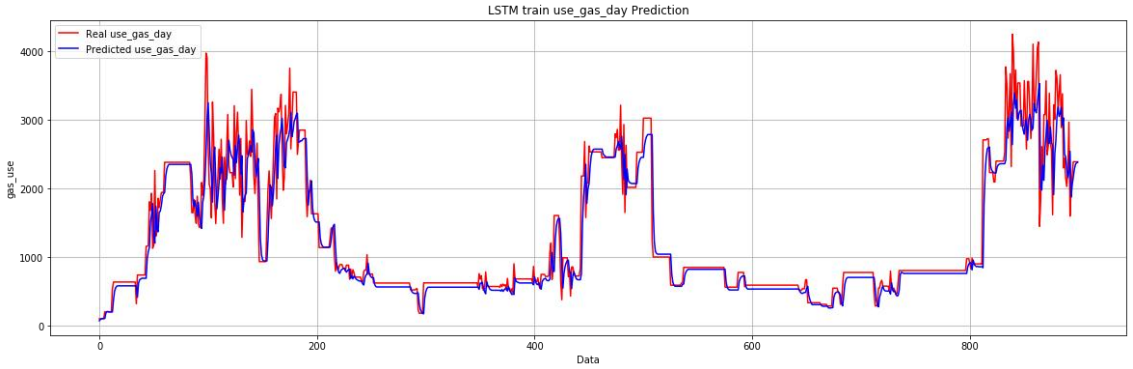

图 4.4 GRA_LSTM 在训练集上的预测趋势图

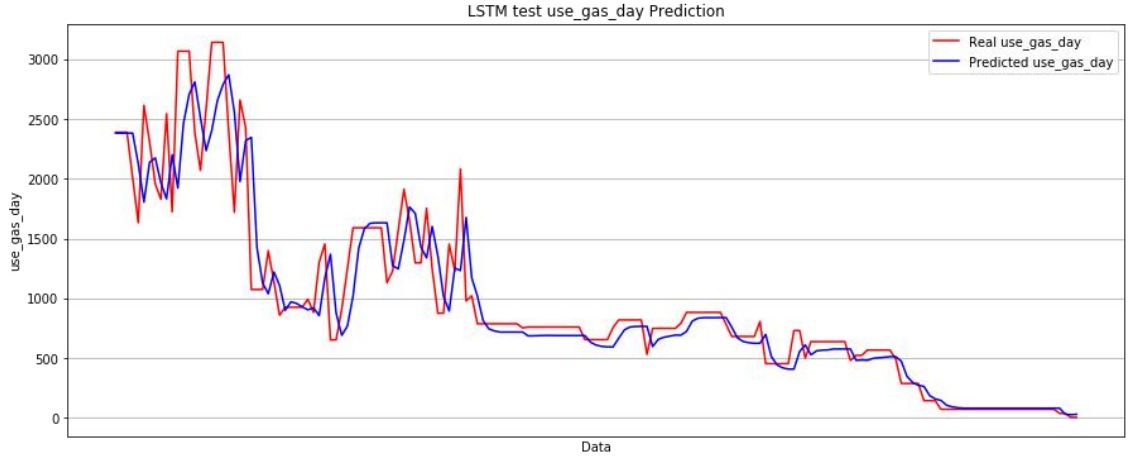

图 4.5 GRA_LSTM 在测试集上的预测趋势图

⑷ 图 4-6 和图 4-7 分别为串联模型模型在训练集和测试集上的预测趋势图，其中红色线条代表真实数据、蓝色线条代表预测数据。

① 可以看出，在生物制药行业中，串联模型可以训练集中取得不错的拟合效果，但是在拟合效果上没有 LSTM、GRA_LSTM 好。

② 可以看出，在生物制药行业中，串联模型可以测试集中取得不错的拟合效果，但是在细节上拟合效果上没有 LSTM、GRA_LSTM 好。

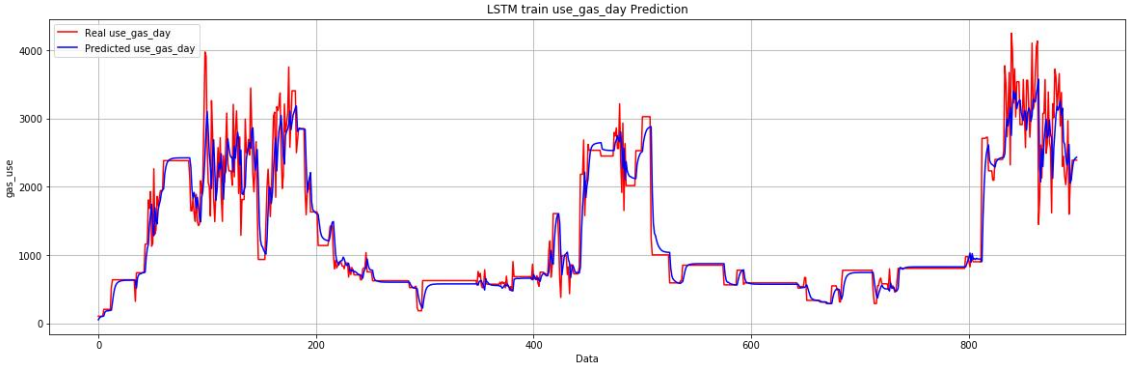

图 4.6 串联模型在训练集上的预测趋势图

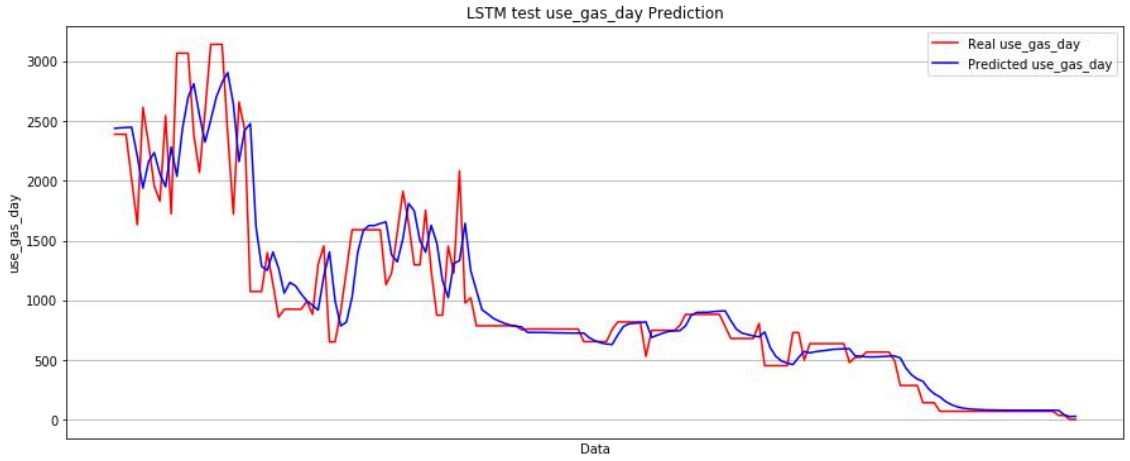

图 4.7 串联模型在测试集上的预测趋势图

⑸ 图 4-1 为并联模型在测试集上的预测趋势图，其中蓝色线条代表真实数据、橙色线条代表预测数据。

① 可以看出，在该行业数据下并联模型可以测试集中取得不错的拟合效果。

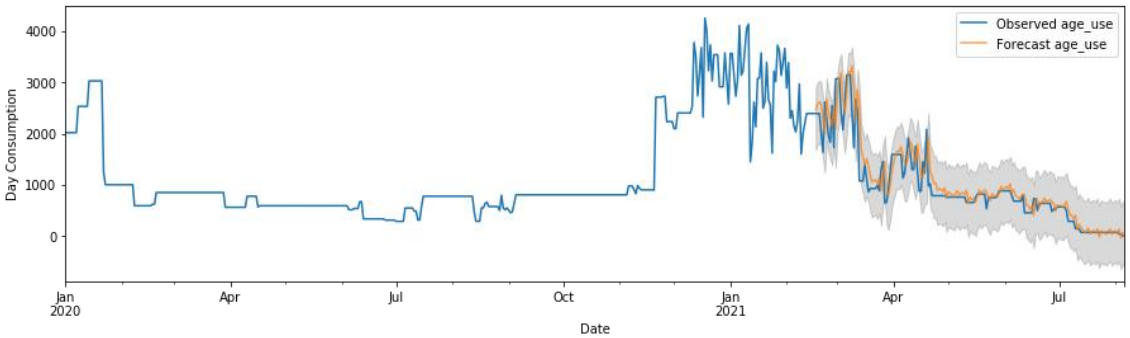

图 4.8 并联模型在测试集上的预测趋势图

5 总结与展望

5.1 总结

基于大数据和机器学习等人工智能算法进行时序数据分析预测是目前比较热门的研究方向。本实验在五大行业：船舶制造、材料制造、物业管理、学校和生物制药尝试并设计了 ARIMA、ARIMAX、SARIMA、SARIMAX、LSTM、GAR_LSTM、ARIMA 与 LSTM 并联、ARIMA 与 LSTM 串联等各种模型，以求得到在本数据集下更好的预测模型。

（1）实验表明，各个模型在各行业中几乎都有较好的预测拟合效果（R-Squared>0.8）

（2）通过各模型在各行业中的性能对比发现，差距不大。

（3）从整体数据上看，GRA_LSTM 模型的效果最好，ARIMA+GRA_LSTM 串联模型效果其次。SARIMA+LSTM 并联模型的效果最差。

（4）对于 ARIMA 和 SARIMA 来说，加上额外辅助信息（如温度 T）成为 ARIMA 和 SARIMAX 后，效果都略微变好。说明，额外辅助信息对 ARIMA 模型有一定的改进。

（5）对于 LSTM 来说，加上额外辅助信息（如温度 T）成为 GRA_LSTM 模型后，效果最好。说明额外辅助信息对 LSTM 的改进很大。

（6）如果输入数据序列没有明显的季节周期性，那么 ARIMA 模型的准确率比 SARIMA 模型的准确率要好。如果输入数据序列有明显的季节周期性，那么 SARIMA 模型的准确率比 ARIMA 模型的准确率要好。

5.2 展望

由于时间、水平和经验有限，上述结论只在本数据集下有效，如若更进一步深究，还需要经受更多的数据测试。再加上数据集没有很明显的季节周期性，无法精确验证 SARIMA 与 ARIMA 的预测效果上的优劣，也无法探讨 SARIMA+LSTM 并联模型的效果究竟是模型本身就差，还是数据集没有季节周期性的缘故。

总之，随着大数据时代的到来，计算机算力的不断提升，各种机器学习算法的层出不穷，未来会有更出色的预测算法。但，当前预测数据的场景是多种多样的，为每个场景适配出合适的预测算法还要做不少的工作，希望这篇文章可以提供一些帮助。

参考文献

[1]汪朝阳,陈曼升,章伟光,钱扬义,俞英.天然气的组成与应用[J].化学教育,2007(04):1-3+12.

[2]Hochreiter, S. and Schmidhuber, J., 1997. Long short-term memory. Neural com.

[3]Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H. and Bengio, Y., 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. arXiv preprint arXiv:1406.1078.

[4]Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need [C]//Advances in Neural Information Processing Systems. 2017: 5998-6008.

[5]G.E.P. Box, G.M. Jenkins, Time Series Analysis Forecasting and Control, Holden-Day, 1976.

[6]M. Q. Raza and A. Khosravi, “A review on artificial intelligence based load demand forecasting techniques for smart grid and buildings,” Renewable and Sustainable Energy Reviews. 2015.

[7]S. N. Fallah, M. Ganjkhani, S. Shamshirband, and K. wing Chau, “Computational intelligence on short-term load forecasting: A methodological overview,” Energies. 2019.

[8]A. Azadeh, S. M. Asadzadeh, and A. Ghanbari, “An adaptive network-based fuzzy inference system for short-term natural gas demand estimation: Uncertain and complex environments,” Energy Policy, 2010.

[9]A. Behrouznia, M. Saberi, A. Azadeh, S. M. Asadzadeh, and P. Pazhoheshfar, “An adaptive network based fuzzy inference system-fuzzy data envelopment analysis for gas consumption forecasting and analysis: The case of South America,” in 2010 International Conference on Intelligent and Advanced Systems, ICIAS 2010, 2010.

[10]F. Yu and X. Xu, “A short-term load forecasting model of natural gas based on optimized genetic algorithm and improved BP neural network,” Appl. Energy, 2014.

[11]G. D. Merkel, R. J. Povinelli, and R. H. Brown, “Deep neural network regression as a component of a forecast ensemble,” in International Symposium on Forecasting, 2017.

[12]G. D. Merkel, “Deep Neural Networks As Time Series Forecasters of Energy Demand,” 37th Annu. Int. Symp. Forecast., 2017.

[13] Hochreiter and J. Schmidhuber, “Long Short-Term Memory,” Neural Comput., 1997.

[14]A. Anagnostis, E. Papageorgiou, V. Dafopoulos and D. Bochtis, Applying Long Short-Term Memory Networks for natural gas demand prediction, 2019 10th International Conference on Information, Intelligence, Systems and Applications (IISA), 2019, pp. 1-7, doi 10.1109IISA.2019.8900746.

[15]J. Contreras, R. Espinola, F. J. Nogales and A. J. Conejo, "ARIMA models to predict next-day electricity prices," in IEEE Transactions on Power Systems, vol. 18, no. 3, pp. 1014-1020, Aug. 2003, doi: 10.1109/TPWRS.2002.804943.

[16]Jian Z， Wang Y， X Fu, et al. Water quality prediction method based on IGRA and LSTM[J]. Water, 2018, 10(9):1148.

[17]张洪刚，李焕．基于双向长短时记忆模型的中文分词方法[J]．华南理工大学学报(自然科学版)，2017(3)