浙江工商大學

《数据案例分析》期末大作业

题目：基于时间序列的电力系统预测分析

学　　院：统计与数学学院
专　　业：数据科学与大数据技术
班　　级：大数据 1902
学生姓名：胡夏冰　　1902110227
指导教师：刘文辉

2022 年 6 月 20 日

基于时间序列的电力系统预测分析

摘要

随着电力行业的发展，可再生能源的并入以及新能源电动汽车等各种新负荷的加入，给电网的安全性和稳定性带来极大挑战。高精度的电力系统中短期负荷预测对电网资源的科学调度以及电网的高效、安全、稳定运行具有重要意义。因此，如何准确的预测电力系统中短期负荷变成了亟待解决的问题。

针对短期时间序列预测，即对该地区电网未来 10 天间隔 15 分钟的负荷进行预测。本文利用时间序列预测模型进行分析，包括但不限于基于统计的 ARIMA 模型，Prophet 模型，基于集成算法的随机森林算法、XGBoost 模型、梯度提升树模型，基于神经网络的 BP 神经网络，长短期记忆网络等。对于 ARIMA 模型，分析发现 ARIMA（4，0，0）的模型最优。对比分析七大不同算法，发现该数据集 Prophet 模型的预测效果最佳。

针对中期时间序列预测，即对该地区电网未来 3 个月日负荷的最大值和最小值进行预测，对该地区各行业未来 3 个月日负荷最大值和最小值进行预测。同样的，本文利用时间序列预测模型进行分析，结果显示，对于该数据集的中期时间序列预测，长短期记忆网络模型的预测效果最佳。

针对时间突变检测，即分析各行业用电负荷突变的时间、量级和可能的原因。本文主要运用了 MK 突变检验，统计学检验，t 检验等多种检验方式。本文将阈值设定为：上限 95%，下限 5%，若不处于阈值内，则说明可能存在突变情况。本文通过统计方法，可视化分析了突变时间、量级，推合理推测可能原因。

最后，本文收集相关文献，分别针对各行业提出建设性意见。

关键字： 电力系统预测　时间序列分析　时间突变检测　Prophet　LSTM

目录

一、 问题的背景与意义

随着电力行业的发展，可再生能源的并入以及新能源电动汽车等各种新负荷的加入，给电网的安全性和稳定性带来极大挑战 [1]。高精度的电力系统中短期负荷预测对电网资源的科学调度 [2] 以及电网的高效、安全、稳定运行具有重要意义。

电力系统负荷预测是指充分考虑历史的系统负荷、经济状况、气象条件和社会事件等因素的影响，对未来一段时间的系统负荷做出预测。短期预测是电网内部机组启停、调度和运营计划制定的基础；中期预测可为保障企业生产和社会生活用电，合理安排电网的运营与检修决策提供支持；长期预测则可为电网改造、扩建等计划的制定提供参考，以提高电力系统的经济效益和社会效益。

复杂多变的气象条件和社会事件等不确定因素都会对电力系统负荷造成一定的影响，使得传统负荷预测模型的应用存在一定的局限性。同时，随着电力系统负荷结构的多元化，也使得模型应用的效果有所降低，因此电力系统负荷预测问题亟待进一步研究。

中短期负荷预测的发展大致可分为：数学统计模型方法预测和人工智能方法预测两个阶段。常见的数学统计模型方法有 ARIMA、Prophet 模型等 [3]。随着人工智能技术的蓬勃发展，随机森林、支持向量机、人工神经网络、LSTM 等机器学习方法在电力负荷预测领域应用十分广泛。组合预测模型通过将多种模型和方法组合起来对中短期电力负荷进行预测，通过结合不同模型特点与优势更好地满足短期电力负荷预测的实际需要，一般情况下，组合模型预测精度均高于单一模型 [4]。

二、 研究思路

在现有研究成果基础上，本文研究思路如下：先对附件 3 中该地区 2018 年 1 月 1 日至 2021 年 8 月 31 日主要的气象数据进行预处理，包括但不限于数据清洗（去除重复值、填补缺失值、修改异常值等）、描述性统计分析、数据可视化等。本文主要完成以下工作：

（1）对该地区电网未来 10 天间隔 15 分钟的负荷进行预测。

对于短期时间序列模型。本文利用时间序列预测模型进行分析，包括但不限于基于统计的 ARIMA 模型，Prophet 模型，基于集成算法的随机森林算法、XGBoost 模型、梯度提升树模型，基于神经网络的 BP 神经网络，长短期记忆网络等。

（2）对该地区电网未来 3 个月日负荷的最大值和最小值进行预测。

同样的，对于中期时间序列模型。本文利用时间序列预测模型进行分析，包括但不限于基于统计的 ARIMA 模型，Prophet 模型，基于集成算法的随机森林算法、XGBoost 模型、梯度提升树模型，基于神经网络的 BP 神经网络，长短期记忆网络等。

（3）分析各行业用电负荷突变的时间、量级和可能的原因。

针对时间突变检测，本文主要运用了 MK 突变检验，统计学检验，t 检验等多种检

验方式。将阈值设定为：上限 95%，下限 5%，若不处于阈值内，则说明可能存在突变情况。本文通过统计方法，可视化分析了突变时间、量级，推合理推测可能原因。

（4）对该地区各行业未来 3 个月日负荷最大值和最小值进行预测。

同样的，对于各行业中期时间序列预测，本文先进行数据处理，提取各行业的用电量情况。本文利用时间序列预测模型进行分析，包括但不限于基于统计的 ARIMA 模型，Prophet 模型，基于集成算法的随机森林算法、XGBoost 模型、梯度提升树模型，基于神经网络的 BP 神经网络，长短期记忆网络等。

（5）对相关行业提出有针对性的建议。

通过收索相关文献，本文分别针对各行业提出建设性意见。

三、 数据处理

3.1 数据准备

本文一共提供三个数据集，其中附件 1 提供了某地区电网 2018 年 1 月 1 日至 2021 年 8 月 31 日间隔 15 分钟的电力系统负荷数据，附件 2 提供了该地区四个行业 2019 年 1 月 1 日至 2021 年 8 月 31 日用电日负荷最大值和最小值数据，附件 3 提供了该地区 2018 年 1 月 1 日至 2021 年 8 月 31 日主要的气象数据。电力系统负荷数据提供了某地区电网的共 128156 条数据，用电日负荷最大值和最小值数据分别提供了该地区四个行业每日的用电量数据，共 3610 条数据，气象数据主要有日期、天气状况、最高温度、最低温度、白天风力风向、夜间风力风向等 6 个特征，共 1344 条数据。

通过随机抽样，将数据集按 8：1：1 分为训练集 (train set)，验证集 (validation set) 和测试集 (test set)，以寻求模型最优参数。其中训练集用来估计模型，验证集用来确定网络结构或者控制模型复杂程度的参数，测试集用来检验最终选择最优的模型的性能如何。

3.2 数据预处理

3.2.1 重复值处理

重复值会占用多余的内存空间，并且在数据分析时也会增加数据的相关性，影响数据分析的结果。处理重复值是数据分析经常面对的问题之一。对重复值进行处理前，需要分析重复值产生的原因以及去除这部分数据后可能造成的不良影响。常见的数据重复可分为两种：记录重复和特征重复。通过去重发现，附件 3 存在一条重复数据，该数据日期为 2018 年 1 月 1 日。

3.2.2 缺失值处理

数据缺失是一个复杂的问题，对于数据挖掘来说，空值的存在，造成了很多不利的影响，缺失数据可能会使得程序运行陷入混乱，出现不可靠输出，也会使数据挖掘过程丢失有用的信息。处理缺失值通常有：删除法、替换法和插值法三种方法。本文发现对题中所给的数据附件三存在少量缺失值，考虑到天气、温度等特征具有相关性。本文将天气的缺失值补充为具有相同温度的天气的众数，同理，将温度的缺失值补充为具有相同天气的温度的众数。若数据中同时确实天气与温度特征，则考虑时间的延续性与滞后性，通过前一天的天气与温度进行填补。

3.2.3 划分数值型变量和分类变量

通过对附件三原始数据集的观察我们发现，日期、天气状况、最高温度、最低温度、白天风力风向、夜间风力风向等 6 个变量不是同一类型，其中包含了数值类型和文本类型。为确保数据分析结果的准确性，防止不同数据类型数据对数据分析结果造成干扰，我们将变量划分成了数值型变量和分类变量。具体划分结果如表 1 所示：

表 **1**　气象值

数值型	日期、最高温度、最低温度
文本型	天气状况、白天风力风向、夜间风力风向

3.2.4 数值化分类变量

数据分析模型中有很多算法都要求输入的特征为数值型，但实际问题中，特征的类型并不一定只有数值型，还会存在相当一部分的类别型，这一部分的特征需要经过处理才可以放入模型之中。

对于多分类变量，由于其文本数据并没有直接意义，因此本文对其进行进一步的量化。由于风向对用电量的影响不大，因此，本文忽略风向对用电量的影响。将无持续风向定为风力强度 0，将小于 2 级的风力定为风力强度 1，将小于 4 级的风力定为风力强度 2，将小于 5 级的风力定为风力强度 3，将小于 9 级的风力定为风力强度 4。具体如表 2 所示。

表 **2**　风力值

风力强度	风力类型
0	无持续风向 <3 级，无持续风向微风
1	微风 <3 级，东北偏东风 2，无持续风向 1-2 级，南风 1-2 级，东南风 1-2 级，东风 1-2 级，北风 1-2 级，东北风 1-2 级，西南风 1-2 级
2	北风 3，东北风 3～4 级，北风 3～4 级，南风 3～4 级，东风 3-4 级，东风 3～4 级，东北风 3-4 级，东南风 3-4 级，北风 3-4 级，南风 3-4 级，西南风 3-4 级，东风 3-4 级，东南风 3～4 级，北风 3-4 级
3	北风 4～5 级，南风 4～5 级，东南风 4-5 级，北风 4-5 级，北风 4-5 级
4	东风 8-9 级

天气状况特征也是文本型特征，其特征取值为：晴、多云、阴、小雨、小雨-中雨、中雨、中雨-大雨、大雨、阵雨、雷阵雨。考虑到天气恶劣程度对用电量的影响本文将其量化为数值型。具体如表 3 所示。

表 **3**　天气情况值

天气状况	对应数值	天气状况	对应数值
晴	1	中雨	6
多云	2	中雨-大雨	7
阴	3	大雨	8
小雨	4	阵雨	9
小雨-中雨	5	雷阵雨	10

3.3 探索性数据分析

探索性数据分析方法是指针对已有数据通过汇总统计、可视化、方程拟合、计算特征量等方式探索数据本身的结构和规律的一种数据分析方法。在对数据进行操作前，首先需要查看数据的形状和分布状况。只有对数据有一定的了解，才能采取相应的操作对该数据进行分析。

对于温度特征，本文对比了 2018 年 1 月 1 日至 2021 年 8 月 31 日每天的最高温度和最低温度。为方便可视分析，本文选取 2018 年 1 月的数据与 2019 年 1 月的数据进行对比，具体如图 1、图 2 所示。

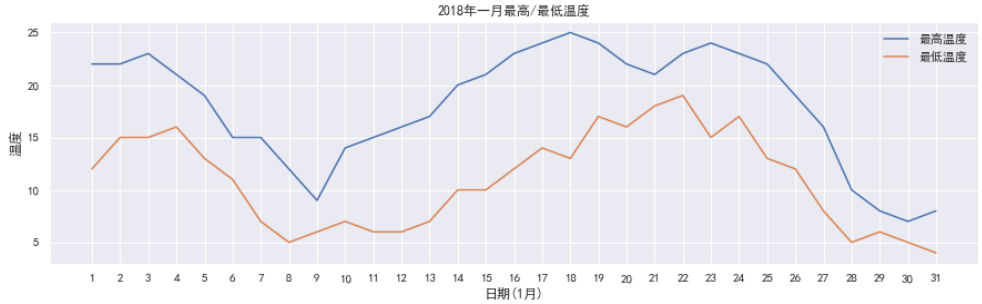


图 **1**　**2018** 年 **1** 月最高温度与最低温度

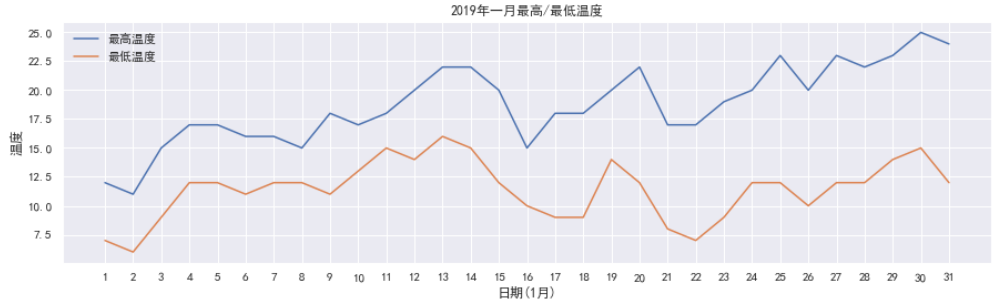


图 **2**　**2019** 年 **1** 月最高温度与最低温度

对于天气特征，本文对比了 2018 年 1 月 1 日至 2021 年 8 月 31 日每天的天气状况。为方便可视分析，本文选取 2018 年 1 月的数据与 2019 年 1 月的数据进行对比，具体如图 3、图 4 所示。

图 **3**　**2018** 年 **1** 月天气状况

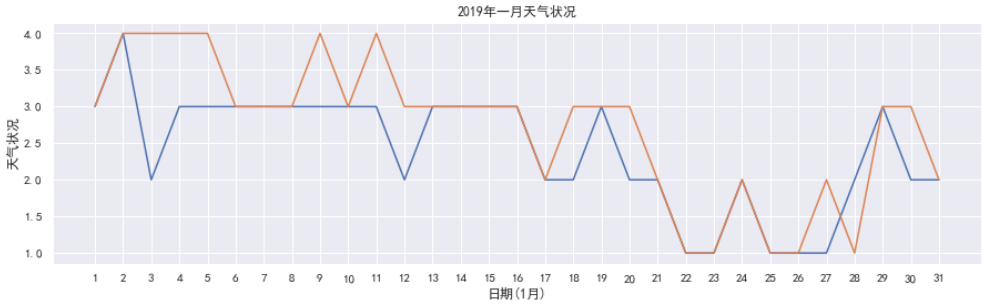


图 **4**　**2019** 年 **1** 月天气状况

对于风力风向特征，本文对比了 2018 年 1 月 1 日至 2021 年 8 月 31 日每天的风力风向状况。为方便可视分析，本文选取 2018 年 1 月的数据与 2019 年 1 月的数据进行对比，具体如图 5、图 6 所示。

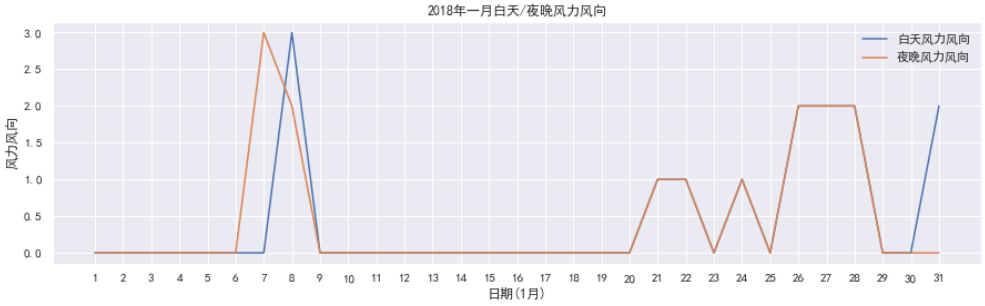


图 **5**　**2018** 年 **1** 月风力状况

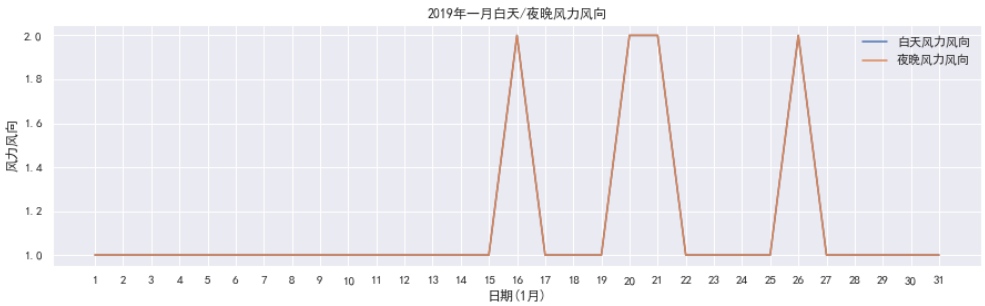


图 **6**　**2019** 年 **1** 月风力状况

通过特征值散点矩阵分布图和相关系数热力图可以发现，scores 特征与 evaluate 特征的相关性为 1。表明了 scores 特征与 evaluate 特征线性相关，当自变量之间存在共线性时，模型的参数会变得极其不稳定，模型得预测能力会下降。很难确切区分每个自变量对因变量的影响，因此增加了对于模型结果得解释成本。对此，本文删去 evaluate 特征。

四、 时间序列预测模型

时间序列预测分析就是利用过去一段时间内某事件时间的特征来预测未来一段时间内该事件的特征。这是一类相对比较复杂的预测建模问题，和回归分析模型的预测不同，时间序列模型是依赖于事件发生的先后顺序的，同样大小的值改变顺序后输入模型产生的结果是不同的。

从时间的序列的平稳性来看，时间序列可以分为平稳序列与非平稳序列，其中平稳序列就是指存在某种周期，季节性及趋势的方差和均值不随时间变化的序列；从变量数目来看分为单变量时间序列与多变量时间序列。

针对时间序列预测模型，本文通过基于统计的时间序列算法，集成学习算法，与神经网络算法，对比分析，并给出最优模型。

4.1 基于统计的时间序列算法

4.1.1 ARIMA 算法

自回归移动平均模型（ARMA)，是由博克思 (Box) 和詹金斯 (Jenkins) 于 70 年代初提出的著名时间序列预测方法，又称为 box-jenkins 模型、博克思-詹金斯法。其中 ARIMA(p,d,q) 称为差分自回归移动平均模型，AR 是自回归, p 为自回归项; MA 为移动平均，q 为移动平均项数，d 为时间序列成为平稳时所做的差分次数。

ARIMA 模型基本思想是：将预测对象随时间推移而形成的数据序列视为一个随机序列，用一定的数学模型来近似描述这个序列。这个模型一旦被识别后就可以从时间序列的过去值及现在值来预测未来值。

ARIMA(p,d,q) 模型可以表示为：

$$\left(1-\sum_{i=1}^{p}\phi_i L^i\right)(1-L)^d X_t=\left(1+\sum_{i=1}^{q}\theta_i L^i\right)\varepsilon_t \tag{1}$$

其中，L 是滞后算子，

$$d\in\mathbb{Z}, d>0 \tag{2}$$

ARIMA 模型含有三个参数：p,d,q。p 代表预测模型中采用的时序数据本身的滞后数, 也叫做自回归项。d 代表时序数据需要进行几阶差分化，才是稳定的，也叫融合项。q 代表预测模型中采用的预测误差的滞后数，也叫做移动平均项。

d 是差分的阶数，首先通过 ADF 检验，看原时间序列的平稳性，如果原时间序列是平稳的，那么 d=0；如果原数据不平稳，那么做差分，通过 ADF 检验直到时间序列平稳。一般差分次数不超过 2 次。

通常在时间序列分析中，采用自相关函数（ACF)、偏自相关函数（PACF）来判别 ARMA(p,q) 模型的系数和阶数。自相关函数 (ACF) 描述时间序列观测值与其过去的观

测值之间的线性相关性。偏自相关函数 (PACF) 描述在给定中间观测值的条件下时间序列观测值与其过去的观测值之间的线性相关性。

4.1.2 Prophet 算法

Prophet 是 facebook 开源的时间序列预测算法。其主要由趋势项（trend），季节项（seasonality）和假期因素（holidays）组成。prophet 库可以进行饱和预测，趋势变化点预测，季节性、节假日影响分析，乘法季节性分析，不确定区间估计，异常值检测等场景的应用。

Prophet 算法模型：

$$y(t) = g(t) + s(t) + h(t) + \epsilon_{(t)} \tag{3}$$

其中，$g(t)$ 表示趋势项，表示时间序列在非周期上面的变化趋势；$s(t)$ 表示季节项，一般来说是以周或者年为单位；$h(t)$ 表示节假日项，表示时间序列中那些潜在的具有非固定周期的节假日对预测值造成的影响；$\epsilon_{(t)}$ 表示误差项，即模型未预测到的波动且其服从高斯分布。

Prophet 算法通过拟合这四项，然后把它们累加。最终得到时间序列的预测值。

1. 趋势项模型 $g(t)$

趋势项 $g(t)$ 有两个重要的函数，一个是基于逻辑回归（非线性增长）的函数，另一个是基于分段线性（线性增长）的函数。

基于逻辑回归的趋势项：

$$g(t) = \frac{C(t)}{1 + \exp(-(k + a(t)^T\delta) \cdot (t - (m + a(t)^T\gamma)))} \tag{4}$$

$$a(t) = (a_1(t), \cdots, a_S(t))^T, \delta = (\delta_1, \cdots, \delta_S)^T, \gamma = (\gamma_1, \cdots, \gamma_S)^T \tag{5}$$

其中，$C(t)$ 表示承载量，它是一个随时间变化的函数，限定了所能增长的最大值，需要提前设定；k 表示增长率，在现实的时间序列中，曲线的趋势不会始终不变，在某些特定的时候或者有着某种潜在的周期，曲线会发生变化，因此该模型定义了增长率 k，来反映发生变化时的转折点（changepoints）。在 Prophet 算法中，需要提前设置转折点的位置，而每一段的趋势和走势也是会根据转折点的情况而改变的。

假设已经放置了 S 个转折点，并且转折点的位置在时间戳 s_j,$1 \leq j \leq S$ 上。那么，在这些时间戳上，就需要给出增长率的变化。假设有一个向量：$\delta \in \mathbb{R}^S$, 其中 δ_j 表示在时间戳 s_j 上的增长率的变化量。如果一开始的增长率使用 k 来代替，那么在时间戳 t 上的增长率就是 $k + \sum_{j:t>s_j} \delta_j$，通过一个指示函数 $a(t) \in \{0, 1\}^S$, 就是

$$a_j(t) = \begin{cases} 1, \text{ if } t \geq s_j \\ 0, \text{ otherwise} \end{cases} \tag{6}$$

在时间戳 t 上面的增长率就是 $k + a(t)^T\delta$。m 表示偏移量，当增长率 k 调整后，每个转折点对应的偏移量 m 也应该相应调整，以连接每个分段的最后一个时间点, 表达式如下：

$$\gamma_j = \left(s_j - m - \sum_{\ell<j}\gamma_\ell\right) \cdot \left(1 - \frac{k + \sum_{\ell<j}\delta_\ell}{k + \sum_{\ell\leq j}\delta_\ell}\right) \tag{7}$$

基于分段线性的趋势项：

$$g\left(t\right) = \left(k + a(t)^T\delta\right) \cdot t + \left(m + a(t)^T\gamma\right) \tag{8}$$

其中，k 表示增长率；δ 表示增长率的变化量；m 表示偏移量。

分段线性函数与逻辑回归函数最大的区别就是 γ 的设置不一样，在分段线性函数中 $\gamma = \left(\gamma_1, \cdots, \gamma_S\right)^T, \gamma_j = -s_j\delta_j$. 分段线性函数不需要 $capacity$ 这个指标的，因此 $m = Prophet()$ 函数默认的使用趋势项基于逻辑回归的增长函数。

转折点 (changepoint) 的选择

在 Prophet 算法中，有三个比较重要的指标，分别是：转折点的位置；转折点的个数；增长的变化率。其中：转折点的位置指的是百分比，需要在前 $changepoint_range$ 长的时间序列中设置转折点，默认值等于 0.8；转折点的个数，默认值等于 25。增长的变化率表示转折点增长率的分布情况。

在默认条件下，转折点的选择是基于时间序列的前 80% 的历史数据，然后通过等分的方法找到 25 个转折点，而转折点的增长率满足 Laplace 分布 $\delta_j \sim Laplace(0, 0.05)$。因此，当 τ 趋近于零的时候，δ_j 也趋向于零的。此时的增长函数将变成全段的逻辑回归函数或者线性函数。

对未来的预估

从历史上长度为 T 的数据中，可以选择出 S 个变点，它们所对应的增长率的变化量是 $\delta_j \sim Laplace(0, \tau)$。此时需要预测未来，因此也需要设置相应的变点的位置, 此时通过 Poisson 分布等概率分布方法找到新增的 $changepoint\_ts\_new$ 的位置，与 $changepoint\_t$ 拼接在一起就得到了整段序列的 $changepoint\_ts$。

2. 季节性趋势 $s\left(t\right)$

由于时间序列中有可能包含多种天，周，月，年等周期类型的季节性趋势。因此，傅里叶级数可以用来近似表达这个周期属性。

Prophet 算法使用傅立叶级数来模拟时间序列的周期性：假设 P 表示时间序列的周期，$P = 365.25$ 表示以年为周期，$P = 7$ 表示以周为周期。其傅立叶级数的形式可以表示为：

$$s\left(t\right)=\sum_{n=1}^{N}\left(a_{n}cos\left(\frac{2\pi nt}{P}\right)+b_{n}sin\left(\frac{2\pi nt}{P}\right)\right) \tag{9}$$

其中，N 表示希望在模型中使用的周期的个数；较大的 N 值可以拟合出更复杂的季节性函数，然而也会带来更多的过拟合问题。

按照经验, 对于以年为周期的序列（$P=365.25$）而言，$N=10$；对于以周为周期的序列 $P=7$ 而言，$N=3$。

当 $N=10$ 时，

$$X\left(t\right)=\left[cos\left(\frac{2\pi\left(1\right)t}{365.25}\right),\cdots,sin\left(\frac{2\pi\left(10\right)t}{365.25}\right)\right] \tag{10}$$

当 $N=3$ 时

$$X\left(t\right)=\left[cos\left(\frac{2\pi\left(1\right)t}{7}\right),\cdots,sin\left(\frac{2\pi\left(3\right)t}{7}\right)\right] \tag{11}$$

因此时间序列的季节项是：

$$s\left(t\right)=X\left(t\right)\beta \tag{12}$$

其中，β 的初始化是 $\beta\sim Normal\left(0,\sigma^{2}\right)$。这里的 σ 是通过 $seasonality\_prior\_scale$ 控制，也就是说 $\sigma=seasonality\_prior\_scale$。$\sigma$ 值越大，表示季节的效应越明显；σ 值越小，表示季节的效应越不明显。

3. 节日效应 $h\left(t\right)$

在现实中，节假日或者是一些大事件都会对时间序列造成很大影响，而且这些时间点往往不存在周期性。因此，对这些点的分析是极其必要的。

在 Prophet 算法里面，收集了各个国家的特殊节假日。除了节假日之外，用户还可以根据自身的情况来设置必要的假期，例如双十一。由于每个节假日或者某个已知的大事件对时间序列的影响程度不一样。例如春节，国庆节则是七天的假期，对于劳动节等假期来说则假日较短。

因此，节假日模型将不同节假日在不同时间点下的影响视作独立的模型，并且可以为不同的节假日设置不同的前后窗口值，表示该节假日会影响前后一段时间的时间序列。

对于第 i 个节假日来说，D_{i} 表示该节假日的前后一段时间。为了表示节假日效应，需要一个相应的指示函数，同时需要一个参数 κ_{i} 来表示节假日的影响范围。

假设有 L 个节假日，那么节假日效应模型为：

$$h(t)=Z(t)\kappa=\sum_{i=1}^{L}\kappa_{i}\cdot1_{\{t\in D_{i}\}} \tag{13}$$

其中，$Z(t)=\left(1_{\{t\in D_1\}},\cdots,1_{\{t\in D_L\}}\right)$，$\kappa=(\kappa_1,\cdots,\kappa_L)^T$。

4.2 集成学习算法

个体学习器（弱学习器）通常只是一个普通的分类器，其精确率仅高于随机判断的50%，例如决策树、前馈层等。而集成学习会将多个个体学习器组合起来，按照集成的方式不同，可分为同质集成和异质集成：

- 同质集成：只包含同种类型的个体学习器，如全是决策树或全是神经网络
- 异质集成：同时包含不同类型的个体学习器：如既有 SVM 又有朴素贝叶斯等

集成模型中的个体学习器要有一定的准确性，即精确率不能太差，同时又要具有多样性，即不同学习器之间要具有一定的差异。按照不同学习器之间的依赖关系，可将个体学习器的生成方式分为两种：强依赖（即一系列个体学习器必须串行生成），弱依赖（即一系列个体学习器可并行生成）。前者的代表为 Boosting 算法，后者的代表为 Bagging 算法。

1. Boosting

Boosting 算法可以将一系列弱学习器提升为强学习器。首先，算法为每一个样本生成一个初始权重，然后基于该权重计算得到一个弱分类器，再根据该分类器的表现，对权重进行调整，使先前分类器划分错误的数据权重提高以在后续任务中受到更多关注。接着，基于调整后的权重训练新的分类器，如此重复执行，最终将上述所有生成的弱分类器加权结合成强学习器。Boosting 中的典型算法有 AdaBoost 和提升树（boosting tree）。图 7 展示了 Boosting 算法的流程。

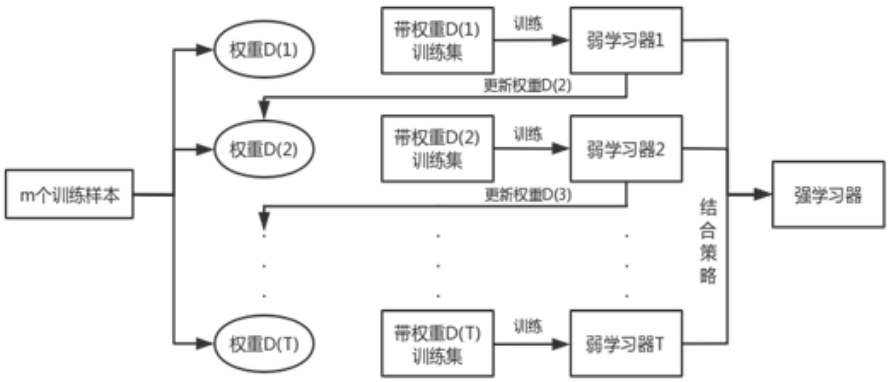


图 **7** **Boosting** 算法的流程图

2. Bagging

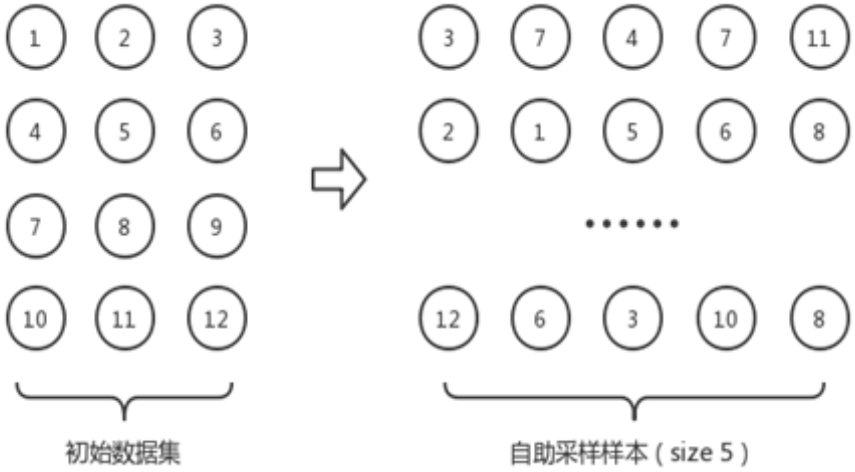


图 **8**　自助采样过程

Bagging 算法中的所有个体学习器都是近似独立的，其每个弱学习器的训练样本都是从总数据集中采取自助随机采样法得到的。自助采样法是一种有放回重复抽样的方法，如图 8，每个被抽中的样本在下一次采样过程中仍有可能被选中，该方法可以防止过拟合。对于采样出的 n 个数据集，分别独立训练出 n 个弱学习器，然后对所有弱学习器进行结合，得到最终的强学习器。Bagging 中的典型算法有随机森林（RF）。图 9 展示了 Bagging 算法的流程。

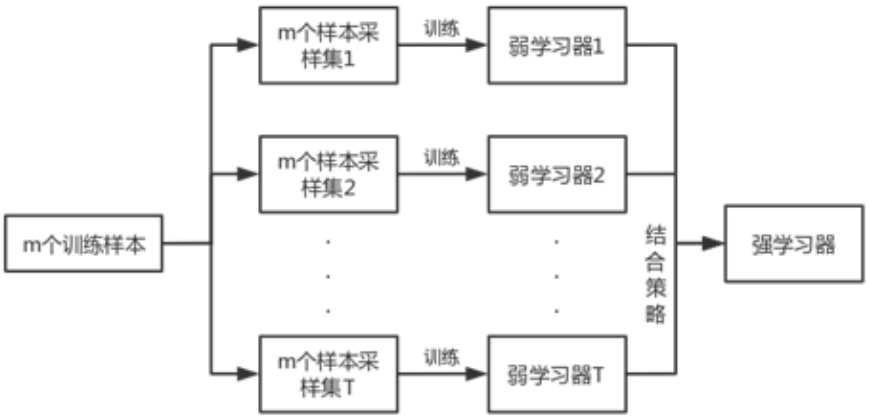


图 **9**　**Bagging** 算法的流程图

4.2.1 随机森林

随机森林 (RF) 是 Bagging 的一个扩展，其变化主要体现在“随机”二字，即在以决策树为基学习器进行集成的基础上引入了随机特征选择。所谓随机特征选择，就是在决策树的分裂过程中随机地隐藏部分属性，而从未隐藏属性集中选择一个最优属性。假

设某一次分裂时节点中含有 d 个属性，未隐藏属性集中含有 k 个属性，则 $\frac{d-k}{d}$ 反映了该模型的随机程度。显然，当 $k=d$ 时随机森林的基学习器构建过程与决策树相同；当 $k=1$ 时就是随机选择一个属性作为最优属性。

另外，随机森林基于一种名为自主采样法的样本选取方法，设给定一个含有 m 个样本的数据集，先随机选取一个样本放入采样集中，再把这个样本放回数据集，以保证下次采样时仍有被选中的可能性。通过这种方式进行 m 次随机采样，便可得到 m 个样本的采样集。当 m 足够大时，可以计算一个样本从未出现在采样集的概率为：

$$\lim_{m\to\infty}(1-\frac{1}{m})^{m}=\frac{1}{e} \tag{14}$$

可见，约 63.2% 的样本出现在采样集中，36.8% 的样本从未出现在采样集中。这些样本可用作验证集来对后续的泛化性能进行“包外估计”。

通过这种方式，可以采集 T 个含有 m 个训练样本的采样集，然后基于每个采样集训练出一个基学习器，将这些基学习器进行组合，得到随机森林模型的预测结果。具体流程图如图 10：

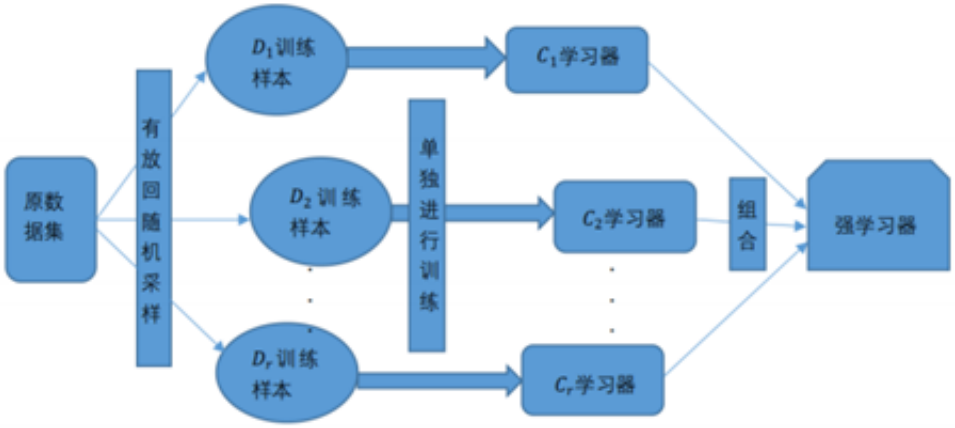


图 **10**　随机森林预测图

随机森林在以决策树作为基学习器构建 Bagging 集成的基础上，进一步在决策树的训练过程中加入随机属性的选择。具体而言，传统的决策树在选择划分属性时是在当前结点的所有候选属性中选择一个最优的属性；而随机森林对基决策树的每个结点，先从该结点的候选属性集合中随机选择一个包含 k 个属性的子集，然后再从这个子集中选择一个最优属性用于划分。由此，随机森林基学习器的多样性不仅来自样本的扰动，还来自属性的扰动，使得集成模型的泛化能力进一步增强。

4.2.2 XGBoost 集成模型

XGBoost 是一种提升树模型，它是由许多 CART 回归树模型集成而来的强分类器。XGBoost 模型以 GBDT 为基础引入了泰勒二次展开，使模型更加强大，适用范围更广。

XGBoost 与 GBDT 一样，本质上都属于 Boosting，但是更具备速度和效率，具体表现为：为了提高模型的泛化能力，XGBoost 模型以 CART 决策树为基学习器，并加入了正则项对代价函数进行惩罚，从而能够有效降低模型的复杂度，防止出现过拟合。并且，XGBoost 在构建模型时使用了代价函数的二次泰勒展开，相比于 GBDT 使用的一阶导数，可以提取更多的信息。值得一提的是，XGBoost 能够利用 CPU 的多线程进行并行处理，大大增加了代码运行速度。

它与 GBDT 模型的主要不同在于目标函数的定义上，XGBoost 模型的目标函数如下：

$$Obj^{(t)} = \sum_{i=1}^{n} L(y_i, \widehat{y_i}^{(t-1)} + f_t(x_i)) + \Omega(f_t) + c \tag{15}$$

其中，$L(x)$ 为损失函数，$\Omega(x)$ 为正则项，c 为常数项。

对于其中的 $f(x)$，XGBoost 模型中利用泰勒公式展开的三项式作近似。可见，最终的目标函数只依赖于每个样本在损失函数上的一阶导数和二阶导数。具体表达式如下：

$$Obj^{(t)} = \sum_{i=1}^{n} [l(y_i, y_i(t-1)) + g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i)] + \Omega(f_t) + c \tag{16}$$

其中，$g_i = \partial_{\widehat{y_i}^{(t-1)}} l(y_i, \widehat{y_i}^{(t-1)})$，$h_i = \partial^2_{\widehat{y_i}^{(t-1)}} l(y_i, \widehat{y_i}^{(t-1)})$。

若将 XGBoost 模型的目标函数简化为两部分，即损失函数和正则化项。其中，损失函数代表了模型的训练误差，正则化项代表了复杂程度。由此，目标函数的表达式可简化为：

$$L(\Phi) = \sum_{i=1}^{n} L(y_i, \widehat{y_i}) + \sum_{k} \Omega(f_k) \tag{17}$$

对于上式，正则化项 $\Omega(x)$ 表示树（弱学习器）的复杂度。正则化值越小，说明模型的复杂度越低，泛化能力越强。具体表达式如下：

$$\Omega(f) = \gamma T + \frac{1}{2} \lambda ||\omega||^2 \tag{18}$$

XGBoost 模型通过不断地添加新的 CART，不断地进行特征分裂来优化模型。实际上，添加新的 CART 的本质是学习一个新的函数 $f(x)$，去拟合上次预测的残差。

当我们训练完模型，得到 K 棵树后，便可进行样本预测。XGBoost 模型会提取预测样本的特征，根据这些特征匹配对应的 CART。这些特征会落到 CART 的叶子节点上，每个叶子节点会对应一个分数。最后，将每棵 CART 对应的分数相加，就是该样本的预测值大小。

4.2.3 梯度提升树

梯度提升树（GBDT）是一种常见的决策树集成模型，与随机森林一样，GBDT 将多个基决策树组合为一个强学习器。作为经典的 Boosting 方法，GBDT 既能用于分类问题，也能用于回归问题。然而，与随机森林不同的是，梯度提升树采用串行化的方式构建树模型，即每棵树的构建过程都是基于前一棵树的预测结果。通常情况下，GBDT 中每颗决策树的深度都较小，因此每颗树的预测结果都只在部分数据集中较好，通过不断增加组合中树的数量，模型整体的泛化能力会不断提升。这样做的另一个目的时可以随时停止“森林”的生长，节约存储空间。另外，GBDT 还采用了强预剪枝的思想，降低了过拟合风险。

综上所述，GBDT 模型的关键问题是如何根据前一棵树的预测结果构建本轮的决策树。假设在迭代过程中我们已得到了前一轮的强学习器 $F_{m-1}(x)$，其预测误差的损失函数为 $L(y, F_{m-1}(x))$，则在本轮迭代中，需要构建一个新的基学习器 $h_m(x)$，使得损失函数 $L(y, F_m(x)) = L(y, F_{m-1}(x)) + h_m(x)$ 最小。

通常，模型残差的计算较为复杂，因此，GBDT 使用损失函数的负梯度值作为残差的近似值，采用梯度下降的思想，逐步构建每一颗回归树。

4.3 神经网络算法

4.3.1 BP 神经网络

BP 神经网络是一种多层的神经网络，图 11 的神经网络有三个隐层，其中第一层 4 个节点，第二层和第三层 3 个节点，输出层有 1 个节点。

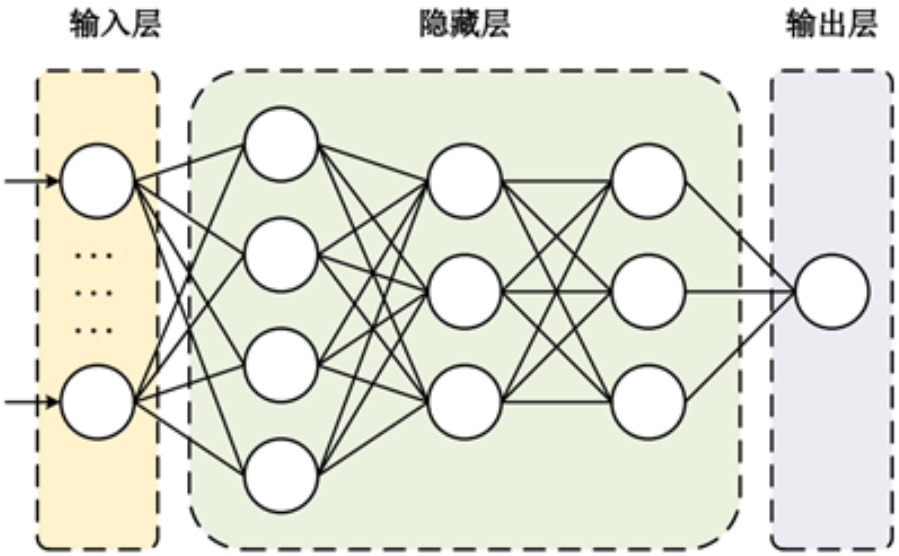


图 **11**　神经网络示意图

图 11 中，每一个结点都是一个神经元模型，它接收上一层各个节点的输出，通过带权重的连接，得到加权的总和并与神经元的阈值比较，最后通过“激活函数”处理并输出。图 12 展示了神经元内部的原理。

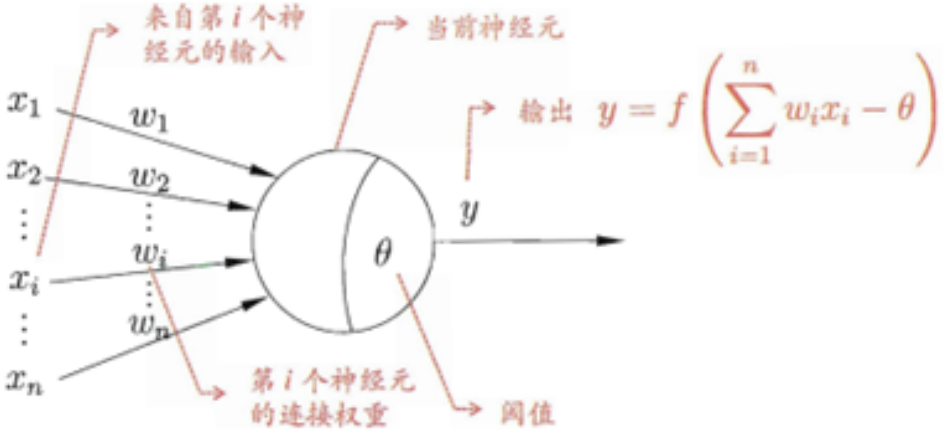


图 **12**　神经元 **(**图片来源：《机器学习》周志华**)**

BP 神经网络基于梯度下降算法，是一种监督学习网络。首先，输入数据从前往后传播计算出每个节点的输出，接着，BP 网络以目标的负梯度从后往前反向传播对各参数进行调整。

在神经网络中，有几个方面需要重点考虑：

（1）神经网络的层数：神经网络的层数决定了模型的规模。一般而言，随着网络层数的增加，误差会逐渐减少，但模型的复杂性也会随之增加。精度也不是一直增加的，会达到一个阈值。因此，神经网络的层数选择是一个十分重要的问题，需要权衡网络层数和模型性能。

（2）隐藏层的神经元数量：和网络层数类似，隐藏层的神经元数量也不是越多越好，如果神经元数量过多，会增加复杂度，加大模型训练时间。相对比，增加某隐藏层的神经元比增加神经网络的层数容易观察和控制。比较合理的设计方式是隐藏层的神经元数量是输入层的两倍。

（3）激活函数类型：激活函数用于加入非线性因素，如果不使用激活函数，最终得到的输出将只是输入的线性组合，从而无法解决非线性问题。一个好的激活函数通常需要具备可微性、单调性和计算简单等特征，常见的激活函数有 ReLU、Tanh、Sigmoid。

（4）学习速率：学习速率类似于梯度下降算法的步长，高的学习速率往往会使模型不稳定，容易陷入局部极小值的低谷，而过低的学习速率虽然能够得到最优解，但加大了模型训练的时间。因此，通常需要多次调整学习速率以寻找适合该模型的初始值。

4.3.2 长短期记忆网络 (LSTM)

考虑到传统的神经网络中，无法将之前学习的训练数据进行记忆。而对于时间序列的预测，往往依赖与前期的训练数据，因此，本文选用长短期记忆网络 (Long Short-Term Memory,LSTM) 模型进行供货量和损失率的时间序列预测。

LSTM 在循环神经网络 (RNN) 的基础上演变而来，其解决了长序列训练过程中的梯度消失和梯度爆炸的问题，对于长序列模型有更好的预测效果。

对于 LSTM 的一个细胞单元而言，其训练主要经过遗忘阶段、选择记忆阶段以及输出阶段这三个阶段。分别通过遗忘门、输入门和输出门实现信息的保护和控制。

遗忘门决定了从细胞单元中丢弃的信息，该门会读取上一个细胞单元的输出和当前细胞单元的输入，通过 Sigmoid 激活函数输出一个介于 0 到 1 的数值，代表该细胞单元的遗忘程度，具体表达式如下：

$$f_t = \sigma(W_f \bullet [h_{t-1}, x_t] + bf) \tag{19}$$

其中，若输出值 $f_t = 0$，表示遗忘门的遗忘程度为 0，即完全保留之前细胞单元的记忆；若输出值 $f_t = 1$，则表示遗忘门的遗忘程度为 1，即完全遗忘之前细胞单元的记忆。

输入门决定了新信息进入细胞单元的多少，该实现过程主要分为两步：一、输入门的 Sigmoid 层决定需要更新的信息内容，用 i_t 表示；二、输入门的 Tanh 层生成一个代表更新内容的向量，用 $\widetilde{C}_t$ 表示。其具体表达式如下：

$$i_t = \sigma(W_i \bullet [h_{t-1}, x_t] + b_i) \tag{20}$$

$$\widetilde{C}_t = Tanh(W_C \bullet [h_{t-1}, x_t] + b_C) \tag{21}$$

输出门决定了当前细胞单元的输出信息，该实现过程主要分为两步：一、通过 Sigmoid 激活函数确定该细胞单元的输出部分，用 o_t 表示；二、将细胞单元通过双曲正切函数处理，并与 o_t 相乘，最终输出所需的信息，用 h_t 表示。其具体表达式如下：

$$o_t = \sigma(W_o \bullet [h_{t-1}, x_t] + b_o) \tag{22}$$

$$h_t = Tanh(C_t) * o_t \tag{23}$$

综上，LSTM 的一个细胞单元由遗忘门、输入门和输出门构成，实现了有选择的“长期记忆”信息的功能。对于每一个细胞单元，其计算过程具体如下：

$$f_t = \sigma(W_f \bullet [h_{t-1}, x_t] + b_f) \tag{24}$$

$$i_t = \sigma(W_i \bullet [h_{t-1}, x_t] + b_i) \tag{25}$$

$$\widetilde{C}_t = Tanh(W_C \bullet [h_{t-1}, x_t] + b_C) \tag{26}$$

$$o_t = \sigma(W_o \bullet [h_{t-1}, x_t] + b_o) \tag{27}$$

$$h_t = Tanh(C_t) * o_t \tag{28}$$

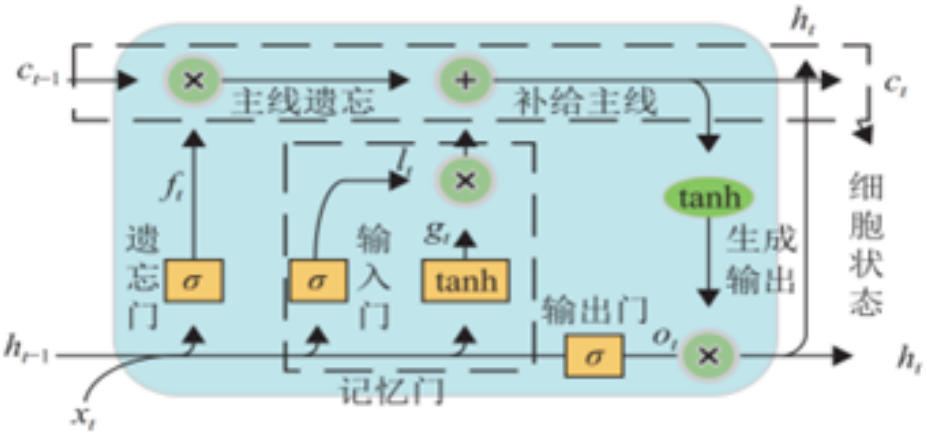


图 **13** **LSTM** 细胞单元示意图

五、 中短期预测建模

5.1 传统时间序列模型

在使用 ARMA、ARIMA 时间序列模型建模时，要求时间序列是平稳的。因此，在研究一段时间序列时，首先需要进行平稳性检验，本文选用 ADF 统计检验方法进行平稳性检验，该方法也叫单位根检验。

ADF 检验全称为 Augmented Dickey-Fuller test。顾名思义，ADF 是 Dickey-Fuller 检验的增广形式。DF 检验只能应用于一阶情况，当序列存在高阶的滞后相关时，可以使用 ADF 检验。因此，ADF 是对 DF 检验的扩展。

在做单位根检验时，需要先明白一个概念，即被检验的对象——单位根。当一个自回归过程中：

$$y_t = by_{t-1} + a + \epsilon_t \tag{29}$$

如果滞后项系数 b 为 1，就称为单位根。当单位根存在时，自变量和因变量之间的关系具有欺骗性，残差序列的任何误差都不会随着样本量增大而减小，即模型中残差的影响是永久的。这种回归又称作伪回归。

ADF 检验就是判断序列是否存在单位根：如果序列平稳，就不存在单位根；如果序列不平稳，则存在单位根。因此，ADF 检验的 H_0 假设就是存在单位根，如果得到的显著性检验统计量小于三个置信度（10%，5%，1%），则对应有（90%，95，99%）的把握来拒绝原假设 H_0。

本文对总有功功率（kw）特征进行 ADF 检验，得到 ADF 结果为-6.73，小于 10%、1%、5% 三个 level 的统计值，说明在这三个 level 上都是平稳的。P-value 为 3.23e-09，不接近于 0, 说明总有功功率（kw）特征平稳。

本文选取了 2018 年 1 月 1 日至 2018 年 1 月 15 日的数据，进行 ADF 检验结果可视化。结果如图 14 所示。

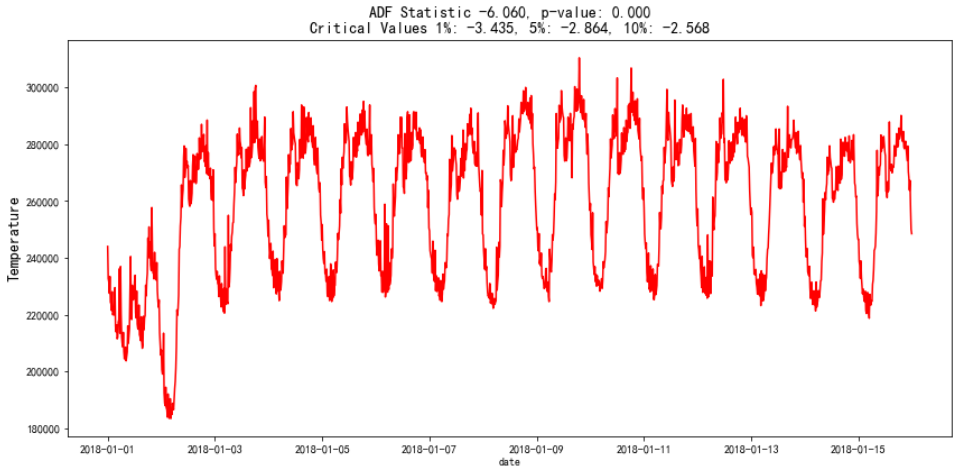


图 **14**　**ADF** 检测结果图

5.1.1 时间序列分解

时间序列分解主要有加法模型和乘法模型。加法指的是时间序分的组成是相互独立的，四个成分都有相同的量纲。而乘法模型输出部分和趋势项有相同的量纲，季节项和循环项是比例数，不规则变动项为独立随机变量序列，服从正态分布。

加法模型有如下形式：

$$Y[t] = T[t] + S[t] + C[t] + I[T] \tag{30}$$

乘法模型有如下形式：

$$Y[t] = T[t] * S[t] * C[t] * I[T] \tag{31}$$

此外，乘法模型可以通过取对数变换为加法模型，具体形式为：

$$\log(Y[t]) = \log(T[t]) + \log(S[t]) + \log(C[t]) + \log(I[t]) \tag{32}$$

本文选用加法模型进行总有功功率（kw）特征和最低温度特征的时间序列分解。图 15 展示了 2018 年 1 月 1 日至 2018 年 1 月 15 日总有功功率（kw）特征和最低温度特征的时间序列分解可视化。

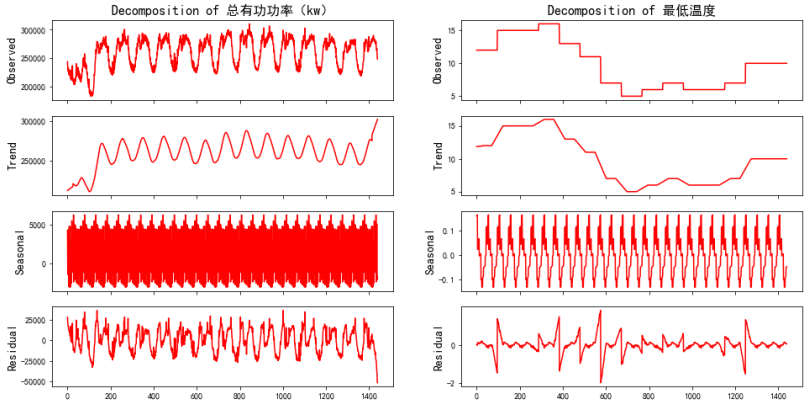


图 **15**　时间序列分解

5.1.2 自相关系数

自相关系数是用以反映变量之间相关程度的统计指标，其度量的是同一事件在两个不同时期之间的相关程度。数学表示为：

$$ACF\left(k\right)=\sum_{t=k+1}^{n}\frac{\left(Z_t-\bar{Z}\right)\left(Z_{t-k}-\bar{Z}\right)}{\sum_{t=1}^{n}\left(Z_t-\bar{Z}\right)^2} \tag{33}$$

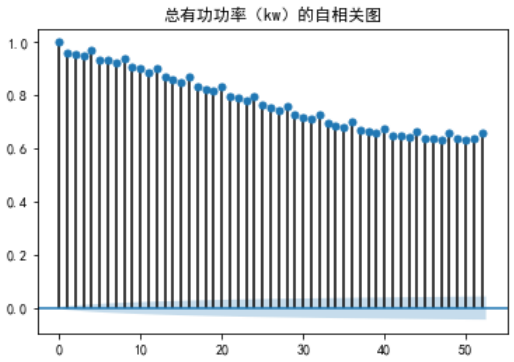
根据 ACF 求出滞后 k 自相关系数 $ACF\left(k\right)$ 时，实际上得到并不是 Z_t 与 Z_{t-k} 之间单纯的相关关系。因为，Z_t 同时还会受到中间 $k-1$ 个随机变量 $Z(t-1)$、$Z(t-2)$,...,$Z(t-k+1)$ 的影响，而这 $k-1$ 个随机变量又都和 Z_{t-k} 具有相关关系。所以，自相关系数里面实际包含了其他变量对 Z_t 与 Z_{t-k} 的影响。

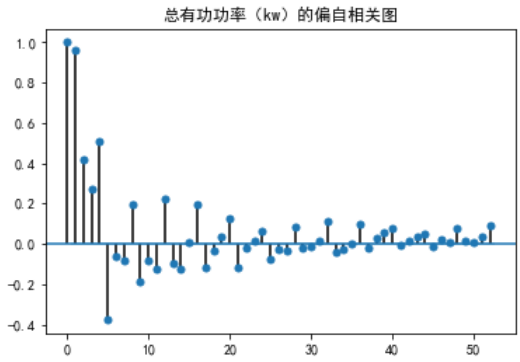
5.1.3 偏自相关系数

为了能单纯测度 Z_{t-k} 对 Z_t 的影响，引进偏自相关系数（PACF）。对于平稳时间序列 Z_t，滞后 k 偏自相关系数是指给定中 $k-1$ 个随机变量 $Z(t-1)$、$Z(t-2)$、…、$Z(t-k+1)$ 的条件下，Z_{t-k} 对 Z_t 影响的相关程度。数学表示为：

$$PACF(k)=\frac{E\left(Z_t-EZ_t\right)\left(Z_{t-k}-EZ_{t-k}\right)}{\sqrt{E\left(Z_t-EZ_t\right)^2}\sqrt{E\left(Z_{t-k}-EZ_{t-k}\right)^2}}=\frac{cov[(Z_t-\bar{Z}_t),(Z_{t-k}-Z_{t-k}^{-})]}{\sqrt{var(Z_t-\bar{Z}_t)}\sqrt{var(Z_{t-k}-Z_{t-k}^{-})}} \tag{34}$$

本文对总有功功率（kw）特征进行自相关、偏自相关分析。得到图 16、图 17。





从图 16、图 17 中可以看出，总有功功率（kw）特征的自相关图呈现拖尾状态，偏自相关图呈现 6 阶结尾状态。同时，总有功功率（kw）特征无季节性变化，即原特征平稳。故拟构建 ARIMA(6,0,0) 模型进行建模预测。

考虑到自相关图和偏自相关图有很大的主观性，因此，本文通过 AIC 或 BIC 来确定最合适的阶数。得到 AIC 和 BIC 的值都为 (4,0)。

因此，本文将对比 ARIMA(6,0,0) 模型与 ARIMA(4,0,0) 模型，从而选择最优模型进行建模。图 18、图 19 分别展示了 ARIMA(6,0,0) 模型与 ARIMA(4,0,0) 模型的概况。

Dep. Variable:	总有功功率 (kw)	No. Observations:	96477
Model:	ARIMA(6, 0, 0)	Log Likelihood	-1020125.03
Date:	Thu, 28 Apr 2022	AIC	2040266.07
Time:	19:41:23	BIC	2040341.88
Sample:	0	HQIC	2040289.11
	- 96477		
Covariance Type:	opg		

	coef	std err	z	P>|z|	[0.025	0.975]
const	2.144e+05	3030.713	70.737	0.000	2.08e+05	2.2e+05
ar.L1	0.4617	0.001	693.401	0.000	0.460	0.463
ar.L2	0.1818	0.001	237.179	0.000	0.180	0.183
ar.L3	0.1192	0.001	159.621	0.000	0.118	0.121
ar.L4	0.5691	0.001	745.404	0.000	0.568	0.571
ar.L5	-0.3055	0.001	-370.043	0.000	-0.307	-0.304
ar.L6	-0.0377	0.001	-50.356	0.000	-0.039	-0.036
sigma2	8.949e+07	24.490	3.65e+06	0.000	8.95e+07	8.95e+07

Ljung-Box (L1) (Q):	0.92	Jarque-Bera (JB):	31437997.56
Prob(Q):	0.34	Prob(JB):	0.00
Heteroskedasticity (H):	0.63	Skew:	-0.72
Prob(H) (two-sided):	0.00	Kurtosis:	91.42

Dep. Variable:	总有功功率 (kw)	No. Observations:	96477
Model:	ARIMA(4, 0, 0)	Log Likelihood	-1025520.108
Date:	Thu, 28 Apr 2022	AIC	2051052.216
Time:	19:46:45	BIC	2051109.079
Sample:	0	HQIC	2051069.500
	- 96477		
Covariance Type:	opg		

	coef	std err	z	P>|z|	[0.025	0.975]
const	2.144e+05	4218.802	50.816	0.000	2.06e+05	2.23e+05
ar.L1	0.3259	0.001	466.674	0.000	0.325	0.327
ar.L2	0.1379	0.001	165.172	0.000	0.136	0.140
ar.L3	0.0703	0.001	92.531	0.000	0.069	0.072
ar.L4	0.4577	0.001	596.166	0.000	0.456	0.459
sigma2	1.001e+08	34.573	2.89e+06	0.000	1e+08	1e+08

Ljung-Box (L1) (Q):	2113.00	Jarque-Bera (JB):	17398703.83
Prob(Q):	0.00	Prob(JB):	0.00
Heteroskedasticity (H):	0.78	Skew:	-1.04
Prob(H) (two-sided):	0.00	Kurtosis:	68.76

对比发现，ARIMA(4,0,0) 模型的预测效果更佳。

5.2 集成学习模型

本文将附件 3 处理完的数据与附件 1 数据合并，得到集成学习数据集，分别调用随机森林算法、XGBoost 算法以及梯度提升树算法进行调用，结果发现 XGBoost 算法的预测效果对于该数据集最佳。

5.3 神经网络模型

同样的，本文将附件 3 处理完的数据与附件 1 数据合并，得到集成学习数据集，分别调用 BP 神经网络算法、长短期记忆网络算法进行预测，结果发现长短期记忆网络算法的预测效果对于该数据集最佳。

先进行时序 3 折交叉检验，得到结果如图 16 所示。

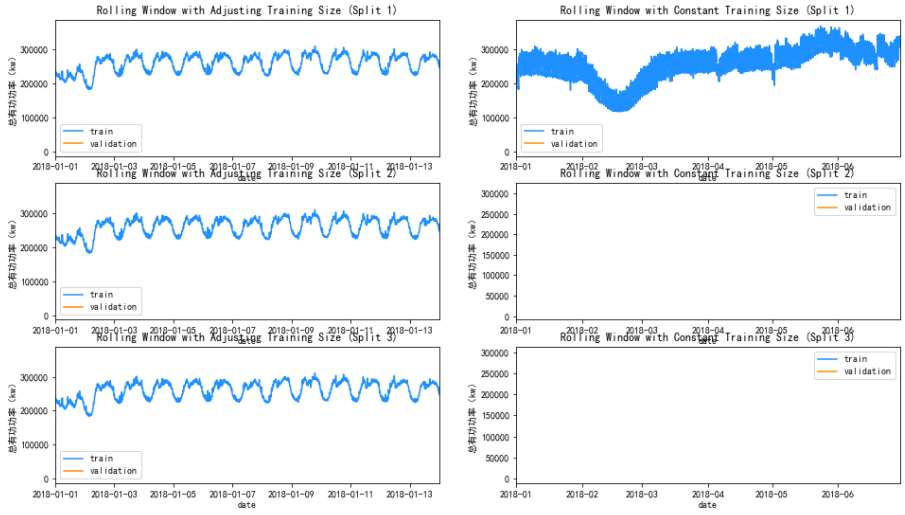


图 **16**　时序 **3** 折交叉检验

本文将训练集 (train set) 与验证集 (validation set) 按 0.85：0.15 的比例划分进行训练。

5.4 模型检验

对于给定数据集 $D=\{(x_1,y_1),(x_2,y_2),\ldots,(x_m,y_m)\}$，其中 y_i 是 x_i 的真实值。在评估一个学习器的好坏时，需要将学习器的预测结果 $f(x_i)$ 与真实值 y_i 进行比较，其常见的性能度量方法有：均方误差，R^2, 根均方误差，绝对值误差等。

(1) 均方误差 MSE

$$MSE=\frac{1}{m}\sum_{i=1}^{m}\left(f(x_i)-y_i\right)^2 \tag{35}$$

更一般地，对于数据分布 D 和概率密度函数 $p(x)$, 均方误差可表示为：

$$MSE = \int_{x\sim\mathcal{D}} (f(x) - y)^2 p(x) dx \tag{36}$$

均方误差是回归中最常用的性能度量。它对误差进行平方求和，意味着误差值越大，MSE 值越大，对大误差值会十分敏感。

(2)R-square

$$R^2 = 1 - \frac{\sum_{i=1}^{m} \left(f\left(x_i\right) - y_i\right)^2}{\sum_{i=1}^{m} \left(\bar{y} - y_i\right)^2} \tag{37}$$

决定系数用于度量：因变量的变异中可由自变量解释部分所占的比例，取值范围是 $[0, 1]$。R^2 值越接近 1, 表明回归平方和占总平方和的比例越大, 回归线与各观测点越接近，用特征的变化来解释因变量变化的部分就越多, 回归的拟合程度就越好。

(3) 均方根误差 RMSE

MSE 公式有一个问题是会改变量纲。因为公式平方了，比如说 y 值的单位是万元，计算出来的是万元的平方，对于这个值难以解释它的含义。所以为了消除量纲的影响，我们可以对这个 MSE 开方。

$$RMSE = \sqrt{\frac{1}{m} \sum_{i=1}^{m} \left(f\left(x_i\right) - y_i\right)^2} \tag{38}$$

均方误差的方根，可从单位度量上衡量模型的效果。MSE 和 RMSE 二者是呈正相关的，MSE 值大，RMSE 值也大，所以在评价线性回归模型效果的时候，使用 RMSE 就可以了。

(4) 绝对值误差 MAE

上面公式为了避免误差出现正负抵消的情况，采用计算差值的平方。还有一种公式也可以起到同样效果，就是计算差值的绝对值。

$$MAE = \frac{1}{m} \sum_{i=1}^{m} \left|f\left(x_i\right) - y_i\right| \tag{39}$$

真实目标与估计值之间差值的平均值。上面三个模型解决了样本数量 m 和量纲的影响。但是它们都存在一个相同的问题：当量纲不同时，难以衡量模型效果好坏。

本文选取 MSE、RMSE、MAE、MAPE 四个值进行模型评价。7 种模型的各评价指标如表 4 所示。

表 **4** 各模型指标

模型	MSE	RMSE	MAE	MAPE
ARIMA(4,0,0)	165151816	25456	2256	3.826
Prophet 算法	116256654	26522	5227	6.265
随机森林算法	146419326	35492	2919	9.696
XGBoost	76952662	84565	5926	5.462
梯度提升树	46485615	15155	4623	4.592
BP 神经网络	216456162	48562	6126	1.592
LSTM	5642659	26496	6499	1.659

针对该地区各行业未来 3 个月日负荷最大值和最小值的预测，本文调用 ARIMA(p,d,q) 模型分别将 p, d, q 取不同值代入模型，得到结果如表 5 所示。

表 **5** 各行业预测值

行业	模型	MSE	RMSE	MAE	MAPE
商业用电$_{max}$	ARIMA(3,0,2)	348331793	18663	16011	0.2133
商业用电$_{min}$	ARIMA(2,0,1)	26337813	5132	3972	2.4590
大工业用电$_{max}$	ARIMA(2,0,1)	512690144	22642	13593	2.5134
大工业用电$_{min}$	ARIMA(2,0,1)	294595285	21596	9495	2.1592
普通工业用电$_{max}$	ARIMA(4,0,2)	5243854	2289	1976	0.2840
普通工业用电$_{min}$	ARIMA(4,0,2)	802679	895	627	2.0835
非普工业用电$_{max}$	ARIMA(3,0,1)	222517	471	389	0.1862
非普工业用电$_{min}$	ARIMA(1,0,2)	210813	459	394	4.7317

六、 时间突变检测

针对时间突变检测，本文主要运用了 MK 突变检验，统计学检验，t 检验等多种检验方式。将阈值设定为：上限 95%，下限 5%，若不处于阈值内，则说明可能存在突变

情况。

6.1 各行业用电突变时间及量级

6.1.1 大工业

通过大工业原始数据和突变数据可视化图分析，大工业用电量在 2019 年 2 月 2 日到 2019 年 4 月 2 日快速上升，接着保存稳定，直到 2019 年 6 月 13 日突变下降，从十万量级功率下降到千量级功率，紧接着突升到原量级，并保持稳定直至 2019 年 10 月 2 日开始下降。

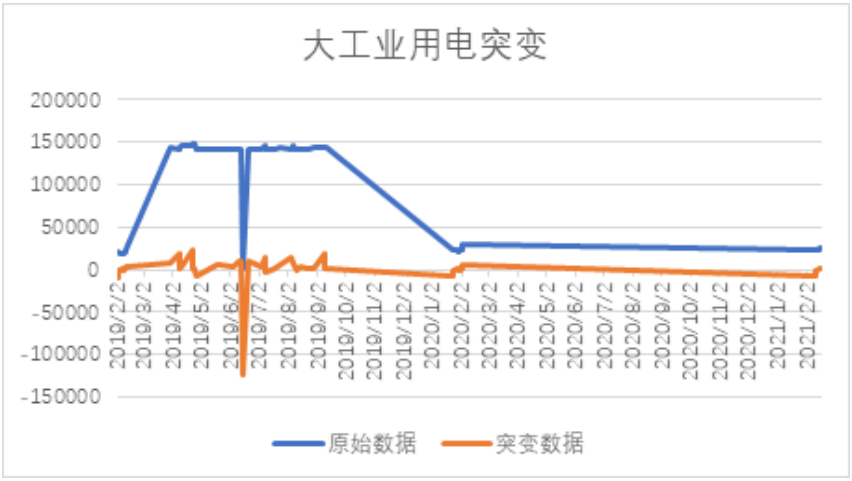


图 **17**　大工业用电突变

使用 BMO 模型分析大工业用电量量级突变点，分析发现突变点在 2019 年一月七月、2020 年和 2021 年的一月都有，但更主要集中在 2019 年七月。

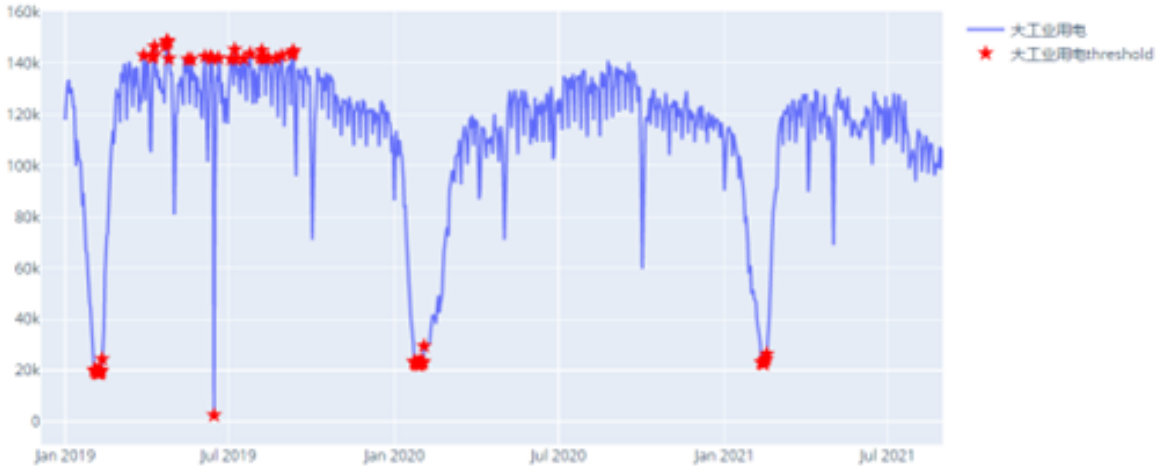


图 **18**　大工业用电突变点

6.1.2 普通工业

通过工业用电的原始数据和突变数据可视化分析，普通工业用电量自 2019 年 2 月 4 日至 2019 年 5 月 4 日快速上升，随后有所波动，在 2019 年 6 月 15 日用电量级从千千瓦突变至百千瓦量级，接着突变至原量级，之后在 2019 年 10 月 4 日到 2020 年 2 月 4 日、2020 年 2 月 4 日到 2020 年 7 月 4 日、2020 年 7 月 4 日到 2021 年 2 月 4 日分别经历了快速下降、快速上升和快速下降阶段。

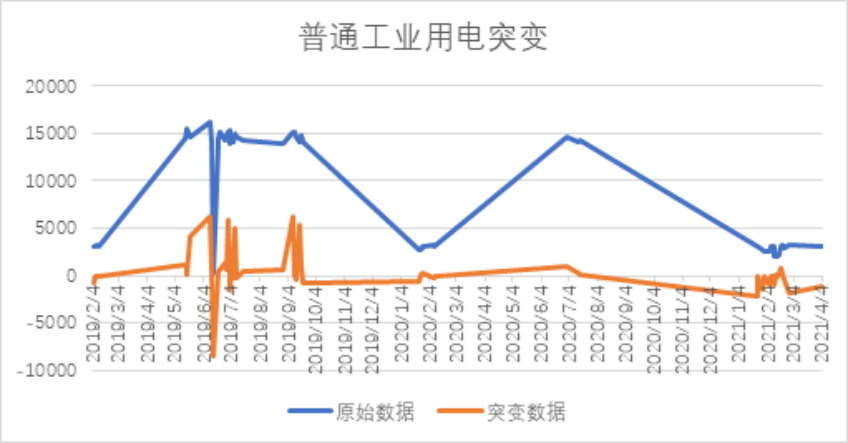


图 **19**　普通工业用电突变

使用 BMO 模型分析普通工业用电量量级突变点，其突变点在 2019 年、2020 年和 2021 年的一月和七月都存在有，但更主要集中在 2019 年七月和 2021 年一月。

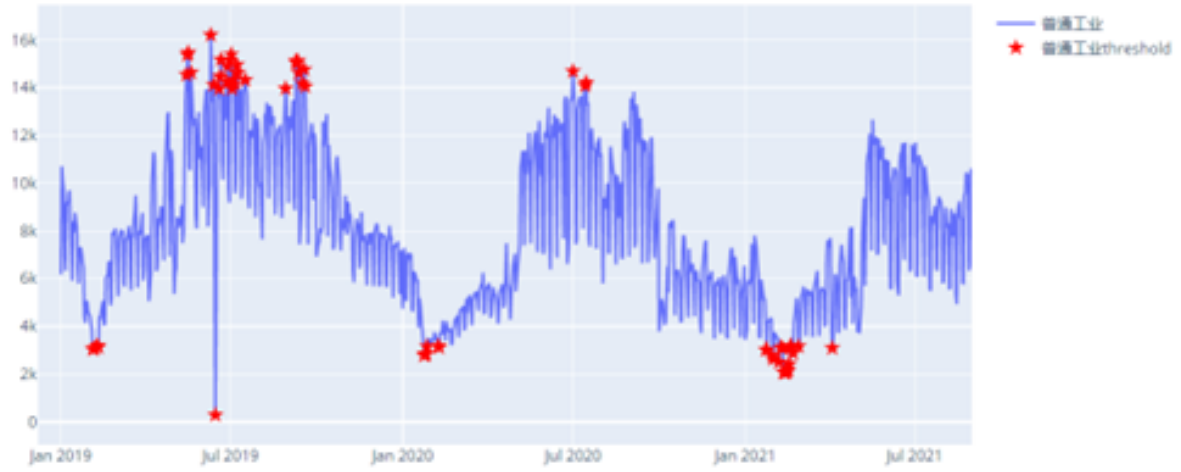


图 **20**　普通工业用电突变点

6.1.3 商业

通过商业用电的原始数据和突变数据可视化研究，2019 年 6 月 28 日之前，商业用电量快速增长至二十万千瓦量级，直至 2019 年 10 月 15 日之间，保存平稳并有所小波动，之后用电量级便稳定下降至万千瓦量级。

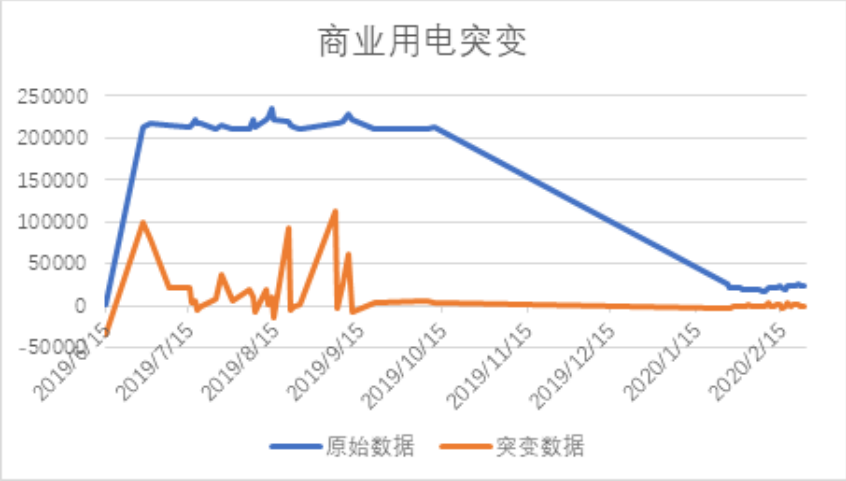


图 **21**　商业用电突变

使用 BMO 模型分析商业用电量量级突变点，该商业用电量突变点主要集中在 2019 年七月上旬，其突变点值达到 200K 以上，2020 年一月上旬，其突变点值达到 30K 左右。

图 **22**　商业用电突变点

6.1.4 非普工业

通过非普工业用电原始数据和突变数据可视化分析，非普工业用电量级一直保存在千千瓦，并有小幅突变。

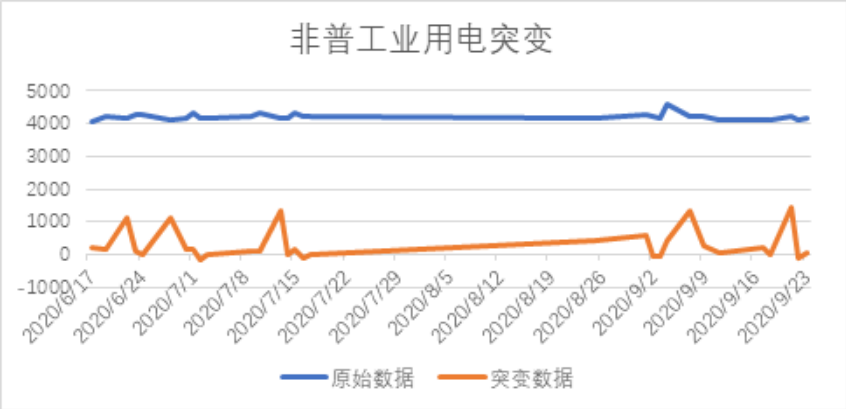


图 **23**　非普工业用电突变

使用 BMO 模型分析非普工业用电量量级突变点，研究表明其突变点主要集中在 2020 年七月，突变点值达到了 4000 以上。

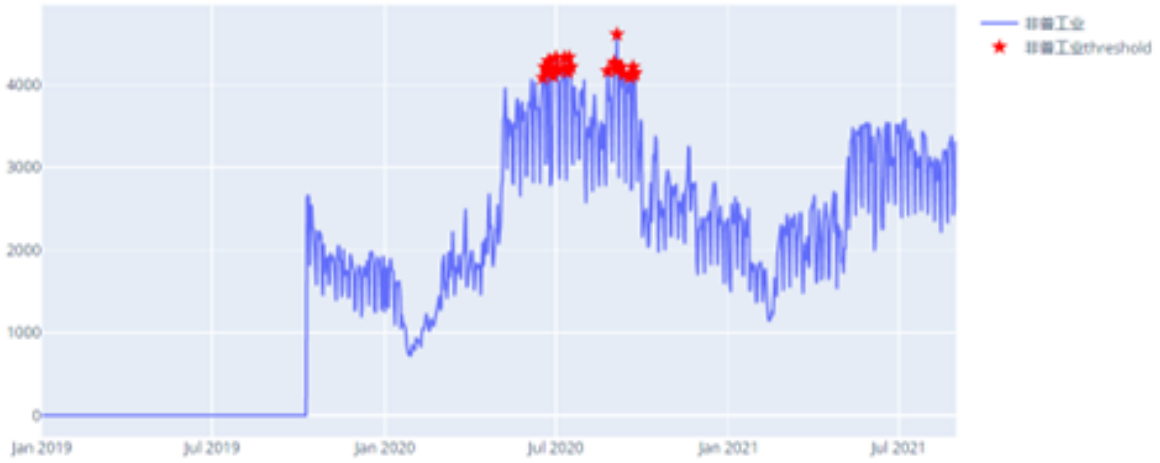


图 **24**　非普工业用电突变点

6.2 电荷突变可能原因

6.2.1 气象因素影响

包括温度、湿度、降水等气象因素，将直接影响电力负荷的突变波动，尤其是在住宅负荷比例较高的地区。由于天气变化很大，电力突变随之波动很大，会导致负荷预测比较困难。近年来，随着生活水平的提高和空调在家庭中的普及，居民家庭的电力冷负荷日益增加。因此，气象温度的突然变化可能会影响电力负荷量。就天气预测而言，天气预测内容只能大致呈现近十来天的总体天气状况和温度，但时间隔得越远，气象精度越低。以风暴天气为例，无法为天气的闪电方向、大小和持续时间做好预报准备，这将导致区域电力负荷曲线的突然变化，更容易出现这方面的复测预报精度低的现象。同

时，当部分地区干旱严重时，人工增雨措施的实施也给满足预报带来了一些困难，由于这些信息的不同步以及相关操作的不可预测影响，从而导致局部用电电荷突变。

6.2.2 节假日因素影响

在节假日期间，与正常工作循环相比，负载将显著降低，春节期间负载也将显著降低。与正常工作日相比，节假日期间可供研究的负荷数据较少，各种随机波动因素将干扰合规性。然而，对于同一节假日的纵向比较，年负荷曲线显示出相对相似的变化趋势。因此，在节假日期间易于引起用电电荷突变。

6.2.3 大用户突发事件影响

对于大型工业用户的装机容量占据高电力负荷的地区，大型工业用户在预测负荷转移方面也发挥着更大的作用。总体而言，在连续生产条件下，大型工业用户的日能源负荷相对稳定。然而，当大型工业用户的设备或外部因素的原因发生变化时，在一定程度上会影响电荷突变。

6.2.4 社会事件因素影响

近两年来，受疫情影响，居民通常会有更多的时间待在家里。除了家庭办公室、在线课外活动和上网课，许多人开始学习烹饪、烧烤和其他技能。因此，与前几年相比，空调、烤箱和空气油炸机等大功率电器的利用率也显著提高。导致一些用户家庭用电提升到了第二和第三级计费。

七、 未来用电负荷影响

我国提出了能源改革的“双碳”战略目标。“双碳”是指“碳中和”和“碳峰”。“碳峰值”指的是该国承诺在 2030 年达到峰值后缓慢减少二氧化碳排放量，不再进一步增加。

7.1 “双碳”目标对能源行业的影响

1 传统的燃煤能源将受到严格控制

燃煤能源是碳排放的主要原因。煤电是国家主要的发电力量，也是二氧化碳污染的主要来源。面对能源改革、节能减排和碳含量减排的压力，传统煤炭能源也面临被淘汰的风险，中国煤炭行业将提前被淘汰，这将不可避免地对中国能源行业产生全球影响。

2 电力市场化的比例将更高

随着电力市场的改革，电力供应和需求的比例将高于当前电力市场的比例，这表明由于电力短缺，后季度用电将发生许多变化。

3 传统发电和新发电的共同发展

由于提出了“双碳”目标，煤炭能源面临被淘汰的风险，因此发展新能源更为重要。中国大多数依赖传统煤电的发电模式也将缓慢改变。相反，风能、水电、光电和储能技术将与传统的煤炭能源相结合。

7.2 相关行业建议

7.2.1 电力

相关数据显示，我国由发电所产生的碳排放含量占比达到 50% 以上，远高于其他发达国家。中国火力发电碳排放含量达到了 600 克每千瓦时，而美国为 419 克每千瓦时，欧盟为 270 克每千瓦时，在双碳政策下，减少火力发电尤为重要。

（1）不断增加新能源发电

到目前为止，中国的火力发电仍占能源总生产的主要部分。建议近期不要新建燃煤电厂，减少火力份额，增加新能源生产份额。非化石能源生产的整合必须改善与清洁能源生产的网络联系。特别是通过市场调节和跨省、跨地区输送，优先发展低成本新能源生产，在新能源资源丰富的地区增加新能源供应，并通过补贴等政策继续刺激新能源生产企业来投资。

（2）关闭低效高排发电厂

到 2020 年底，中国燃煤电厂完成了 0.95TW 的低排放节能改造，约占中国火力发电能用的 76%。中国累计关闭了近 40GW 的小型、陈旧且效率低下的发电厂。同时，制定行动计划和相应目标，以在近期内消除延迟发电厂。建议在双碳政策期间继续推进低效、高排放工厂的停产，根据技术、经济和环境标准识别不合格工厂，并不断更新和完善标准文件。这不仅是能源转型规划合理布局的重要组成部分，而且可以对空气质量、公共卫生等短期目标产生良好的协同效应。

（3）完善电力市场体制

自第二轮电力体制改革以来，各地区不断完善能源交易市场机制，但主要集中在中长期商业市场，而新能源发电受其不稳定性影响，一般表现在中长期市场。因此，有必要建立一个更有效的电力市场体制，以提高新能源的普及率和电力网的灵活性。一个运转良好的能源市场对于推动短期新能源体制改革具有重要意义。短期电力市场价格是当前电力供需关系的实时体现，可以为中长期电价提供参考，一方面引导新增电力装机容量投资，另一方面促进新能源金融市场的完善。此外，现货新能源市场的完善也促进了能源储备市场的发展，为提高风能、光伏等新能源发电能力提供了基础，从而达到双赢、相互促进的目的。

（4）推进碳捕捉技术

碳捕获技术在目前的市场环境下还不成熟，在实际应用中普遍普及，但它是未来实现碳零排放的重要基础，尤其是在火力发电领域。目前，作为我国主要发电方式的火力电厂，短期内关闭燃煤电厂是不切实际的。此外，将碳捕获技术应用于生物能源甚至可以实现碳排放的负面影响，以抵消其他非脱碳领域的碳排放。虽然碳捕获技术的推广应用存在一些问题和困难，但碳捕获技术对于实现“双碳”目标非常重要。建议相关政府部门通过政治手段吸引相关领域投资，发展碳捕获技术，推动碳捕获技术试点项目的实施，以提高脱碳技术水平，改造现有工厂，实现零排放。

7.2.2 工业

中国的可持续工业发展仍然面临许多挑战。特别是目前我国工业发展主要依靠资源和能源投入，单位工业增加值能耗远高于发达国家。受疫情因素影响，在后疫情时代，经济复苏缺乏政治引导和投资，进一步加剧了我国产业产能过剩和结构性问题，加大了产业转型的难度。“十四五”期间，工业将面临产能过剩、高耗能产品比重大、附加值低、能效低、区域分布不均等多重问题。面对挑战，中国工业部门也将有机会向低碳经济转型升级，全面提高生产力，创新商业模式，为高质量、高水平的长期发展奠定基础。

（1）抵消产能过剩

产能过剩是工业部门向低碳经济转型的主要矛盾之一。在现有经济市场下，市场力量必须由生产要素的价格和分布决定，生产要素的分布由企业的竞争力决定，以消除滞后的生产能力。消除产能过剩离不开政府部门的参与。在市场调节的基础上，政府参与市场，建立市场调节机制，制定相应双碳政策，并在工业发展过程中考虑能效、环保、安全、质量等因素。

（2）提高节能技术水平

与国外先进企业的产品能耗相比，我国高耗能行业的能耗水平普遍较高。为了降低能源消耗水平，充分利用现有节能技术的潜力，这是一种更有效、更具成本效益的减排方法。一方面，提高先进企业的能效，可以从局部结合、个体节能转变为全过程、全系统节能，这与管理部门的激励措施密不可分；另一方面，通过推进锅炉、电机、变压器等主要耗能设备的绿色升级和能耗提升，最大限度地发挥节能潜力；最后，通过信息技术和数字反馈的发展，实现节能高效的目的。

（3）提升电气化水平，实现电能替代

如果电气化与能源行业的脱碳有机结合，它将在尽快实现工业部门的双碳方面发挥至关重要的作用。实现电气化需要采取很多措施，包括推进工业方法创新，实现工业电气化、数字化和智能化技术的协调发展；采用先进的能源生产技术代替传统生产技术，满足高规格产品的生产需求；推动电加热发展，通过电加热泵提供低温热源；最后，完善市场机制，支持工业电气化。例如，根据工业企业规模、能耗时间分布和能耗效率，

完善峰谷电价、差价和分级定价政策。

7.2.3 交通运输

交通部门是促进中国经济活动和社会互联互通的关键环节。近年来，各种交通工具的所有权不断增加，能源消耗和碳排放等问题日益突出。据统计，2018 年交通运输业碳排放总量达到 11 亿吨，其中道路运输碳排放量占 77%。

（1）加快调整货运方式

《交通运输部关于全面加强生态环境保护，坚决打好污染防治攻坚战的实施意见》等政策文件要求减少公路货运量，增加铁路货运量。中国需要增加铁路和内陆货运在大宗商品长途运输中的比重，增加铁路和港口在运输网络中的密度，逐步降低重型柴油卡车在大宗商品长途运输中的比重。

（2）促进新能源汽车转型

随着我国新能源汽车产业的快速发展，新能源汽车的销量和渗透率将继续提高。中国需要继续加快新能源汽车充电电池建设，推动新能源汽车大规模转型。中国可能会考虑制定分阶段目标，禁止销售除重型卡车以外的新型燃油汽车。

（3）推进智能交通发展

推动 5G 通信技术与车路协调系统的融合发展。到 2025 年，将在部分路段进行车辆道路协调的试点应用。提高整个交通基础设施规划、设计、建设、维护、运营和管理周期的数字化水平，建设大规模、系统化的大数据套件和跨车辆、跨基础设施的综合交通大数据中心系统。

参考文献

[1] 夏博, 杨超, 李冲. 电力系统短期负荷预测方法研究综述 [J]. 电力大数据, 2018, 21(7): 22-28.

[2] 康重庆, 夏清, 张伯明. 电力系统负荷预测研究综述与发展方向的探讨 [J]. 电力系统自动化, 2004,28(17): 1-11.

[3] 王栋. 电力系统负荷预测综述 [J]. 电气开关,2020, 58(1): 6-8,20.

[4] 魏明奎，叶葳，沈靖，等. 基于自组织特征神经网络和最小二乘支持向量机的短期电力负荷预测方法 [J]. 现代电力，2021，38(1)：17-23.

[5] 杜雅楠，齐敬先，施建华，等. 基于 LSSVM 的超短期负荷区间预测 [J]. 计算机系统应用，2021，30(3)：184-189.

[6] 李焱，贾雅君，李磊，等. 基于随机森林算法的短期电力负荷预测 [J]. 电力系统保护与控制，2020，48(21)：117-124.

[7] 庞传军，余建明，冯长有，等. 基于 LSTM 自动编码器的电力负荷聚类建模及特性分析 [J]. 电力系统自动化，2020，44(23)：57-63.

[8] 李香龙，马龙飞，赵向阳，等. 基于 LSTM 网络的时间多尺度电采暖负荷预测 [J]. 电力系统及其自动化学报，2021，33(4)：71-75.

[9] 王永志，刘博，李钰. 一种基于 LSTM 神经网络的电力负荷预测方法 [J]. 实验室研究与探索，2020，39(5)：41-45.

[10] 李丹，张远航，杨保华，等. 基于约束并行 LSTM 分位数回归短期电力负荷概率预测方法 [J]. 电网技术，2021，45(4)：1356-1364.

[11] 姚李效, 宋玲芳, 李庆宇等. 基于模糊聚类分析与 BP 网络的电力系统短期负荷预测 [J]. 电网技术, 2005, 29(1):20-24.

[12] 赵杰辉, 葛少云, 刘自发等. 基于主成分分析的径向基函数神经网络在电力系统负荷预测中的应用 [J]. 电网技术, 2004,28(5):35-37.

[13] 牛东晓, 曹树华, 赵磊等. 电力负荷预测技术及其运用 [M]. 北京: 中国电力出版社, 1998.

[14] 罗春雷, 孙洪波, 徐国禹,BP 模型的改进算法及其在负荷预测中的应用 [J]. 重庆大学学报,1995,18(6): 110-115.

[15] 夏博, 杨超, 李冲. 电力系统短期负荷预测方法研究综述 [J]. 电力大数据, 2018, 21(7): 22-28.

[16] 康重庆, 夏清, 张伯明. 电力系统负荷预测研究综述与发展方向的探讨 [J]. 电力系统自动化, 2004,28(17): 1-11.

[17] 王栋. 电力系统负荷预测综述 [J]. 电气开关,2020, 58(1): 6-8,20.

[18] 魏明奎，叶葳，沈靖，等. 基于自组织特征神经网络和最小二乘支持向量机的短期电力负荷预测方法 [J]. 现代电力,2021,38(1)：17-23.