分类号_____ 学号 M202070103

学校代码 10487 密级_____

华中科技大学

硕士学位论文

（学术型□ 专业型☑）

基于 Transformer 的集成学习在时序预测中的应用

学位申请人：李天峰

学 科 专 业：应用统计

指 导 教 师：刘小茂 副教授

答 辩 日 期：2022 年 5 月 12 日

A Dissertation Submitted in Partial Fulfillment of the Requirements for the Professional Master Degree

Application of Transformer-based Ensemble Learning in Time Series Prediction

Candidate : LI TianFeng
Major : Applied Statistics
Supervisor : Assoc. Prof. LIU XiaoMao

Huazhong University of Science and Technology
Wuhan 430074, P. R. China
May, 2022

独创性声明

本人声明所呈交的学位论文是我个人在导师指导下进行的研究工作及取得的研究成果。尽我所知，除文中已经标明引用的内容外，本论文不包含任何其他个人或集体已经发表或撰写过的研究成果。对本文的研究做出贡献的个人和集体，均已在文中以明确方式标明。本人完全意识到本声明的法律结果由本人承担。

学位论文作者签名：李跃峰

日期：2022年5月23日

学位论文版权使用授权书

本学位论文作者完全了解学校有关保留、使用学位论文的规定，即：学校有权保留并向国家有关部门或机构送交论文的复印件和电子版，允许论文被查阅和借阅。本人授权华中科技大学可以将本学位论文的全部或部分内容编入有关数据库进行检索，可以采用影印、缩印或扫描等复制手段保存和汇编本学位论文。

保　密□，在_____年解密后适用本授权书。

本论文　不保密☑。

（请在以上方框内打“√”）

学位论文作者签名：李跃峰　　指导教师签名：刘小茂

日期：2022年5月23日　　日期：2022年5月23日

摘 要

精准地时序预测对社会发展与进步十分重要。从气象学到金融学、从交通分析到市场分析，时序预测在社会各行各业都有实际应用场景。

自上世纪初，ARIMA、GARCH 等传统统计学模型被陆续提出，为时序预测提供了坚实的理论基础。但传统统计学模型结构单一、表达能力有限。随着人工智能的迅速发展，有学者陆续将 SVR、XGBoost 等机器学习模型应用到时序预测中，但机器学习模型的效果严重依赖特征工程，这就有很强的主观成分，无形中提高了时序预测的难度。而 RNN、LSTM 等深度学习模型，在时序预测中存在长序列依赖问题。

Transformer 模型自提出以来，凭借强大的自注意力机制在自然语言处理领域大放异彩，考虑到时序预测同自然语言处理的相似性，在于挖掘不同位置元素间的依赖性，因此可以将 Transformer 模型引入到时序预测中。为充分挖掘序列中的信息，本文考虑应用信号分解算法将时序数据中不同振幅和频率的分量拆分开，再对各子序列单独建模分析。另外，考虑到时序预测模型往往不能很好地处理高振幅高频率数据，且预测存在不同程度的延迟，故打算利用偏差修正来解决这一问题。据此，本文将 Transformer 模型与信号分解算法 CEEMDAN 和 SVR 偏差修正相融合，得到基于 Transformer 的集成学习模型：首先使用 CEEMDAN 方法将原始时序数据拆分成若干个振幅和频率互不相同的子序列；之后对各子序列利用 Transformer 模型分别学习和预测，得中间预测结果；最后，采用 SVR 模型对中间预测结果进行偏差修正。

为充分说明模型有效性，本文选用包括医疗数据、工业数据、金融数据、气象数据在内的经典时序数据集，利用 MSE 和 MAE 评价指标，将本文提出的基于 Transformer 的集成学习模型与 ARIMA、SVR、LSTM 模型进行对比分析。实验结果验证了 Transformer 模型在时序预测上的优越性能，以及序列分解和偏差修正可有效提高模型的预测精度。

关键词：时序预测；Transformer；信号分解；偏差修正；集成学习

Abstract

Accurate time series forecasting is very important for social development and progress. From meteorology to finance, from traffic analysis to market analysis, time series forecasting has practical application scenarios in all walks of life.

Since the beginning of the last century, traditional statistical models such as ARIMA and GARCH have been proposed one after another, providing a solid theoretical basis for time series forecasting. However, traditional statistical models have a single structure and limited expression capabilities. With the rapid development of artificial intelligence, some scholars have gradually applying machine learning models such as SVR and XGBoost to time series forecasting, but the effect of machine learning models relies heavily on feature engineering, which has a strong subjective component, which virtually increases the difficulty of time series forecasting. And deep learning such as RNN and LSTM model, there is a long series dependency problem in time series forecasting.

Since the Transformer model was proposed, it has shined in the field of natural language processing with its powerful self-attention mechanism. Considering the similarity between time series forecasting and natural language processing, it is to explore the dependencies between elements at different positions. Therefore, the Transformer model can be introduced into time series forecasting. In order to fully mine the information in the sequence, this paper considers applying the signal decomposition algorithm to separate the components of different amplitudes and frequencies in the time series data, and then models and analyzes each subsequence separately. In addition, considering that time series forecasting models often do not handle high-amplitude, high-frequency data well, and the prediction has different degrees of delay, so we intend to use bias correction to solve this problem. Accordingly, this paper integrates the Transformer model with the signal decomposition algorithm CEEMDAN and SVR bias correction to obtain the Transformer-based ensemble learning model: Firstly, the CEEMDAN method is used to split the original time series data into several sub-sequences with different amplitudes and frequencies; Then the Transformer model is used to learn and predict each sub-sequence separately, and the intermediate prediction results are obtained; Finally, the SVR model is used to correct the deviation of the intermediate prediction results.

In order to fully illustrate the effectiveness of the model, this paper selects classic time series data sets including medical data, industrial data, financial data, and meteorological data, and uses the MSE and MAE evaluation indicators to compare and analyze the Transformer-based ensemble learning model proposed in this paper with ARIMA, SVR and LSTM models. The experimental results verify the superior performance of the Transformer model in time series prediction, and the sequence decomposition and bias correction can effectively improve the prediction accuracy of the model.

Key words: Time Series Forecasting, Transformer, Signal Decomposition, Bias Correction, Integrated Learning

目 录

1 绪论

1.1 研究背景与意义

时序预测是指在现有时序数据的基础上，挖掘时序数据中不同位置元素间的内在关系，进行理论建模后对时序未来数据进行预测。时序预测可以利用该时序数据的历史数据，也可以利用其他多个相关变量的历史数据对该变量未来数据做预测[1]。时序预测可以通过时序分析，进行类推或者延展，常作为一种有效预测手段而被广泛应用于军工研发、网络通信、生物医学、经济金融、气象预测、交通分析等领域。

时序预测在早期的自然科学领域就发挥了十分重要的作用，已经体现在人民生活的方方面面，如通过商品销量的准确预测来指导企业制定运营策略；精准的气象灾害预测是防灾减灾的关键；风速预测是风力发电厂的选址和策略制定中不可避免的一环等等。时序预测为人民生产生活提供了巨大的便利，积极推动了科学的发展。

近些年时序数据逐渐呈现出规模大、复杂度高的特点，传统的时序预测通过时序的先验知识建立统计学模型，进而求解模型参数的方法越来越难以处理这些时序数据。上世纪末以来，人工智能的发展日新月异，成为众多科研学者的研究对象，众多机器学习、深度学习模型被先后提出，其中具有代表性的为卷积神经网络(Convolutional Neural Network, CNN)和递归神经网络(Recursive Neural Network, RNN)。基于图像识别提出的 CNN 和基于自然语言处理(Natural Language Processing, NLP)提出的 RNN 被调整后应用到时序预测中，并取得了不俗的表现。2017 年，Google 团队提出的 Transformer 模型在 NLP 领域的重大成功展示了注意力机制的强大建模能力。此后，例如 Informer、Autoformer 等基于 Transformer 的模型被先后提出。

尽管深度学习模型已经在时序预测的问题上取得了不错的效果，但这些模型仍然存在几个问题。一是仅利用历史时序数据对未来单一时序数据进行预测，忽略时序数据不同分量中单独蕴含的信息；二是时序模型的改进主要体现在模型内部结构，没有单独分析模型预测的残差序列；三是模型无法并行训练，且训练中会产生不同程度的长序列信息无法传递的问题。

复杂信号在时域中难以看出变化规律，但在频域中却能够明显看出其特征。将信号分解算法引入时序分解，得到的多个时序子序列都蕴含着各子独有的序列信息。本文采用信号分解算法处理时序数据，分解得到多个子序列后对各子序列单独分析建模，进一步深入地挖掘子序列的信息。之后将各子序列的预测结果加和即最终预测。

不同模型预测结果的残差序列各不相同，残差序列中也蕴含着该模型的信息。因此，对于残差序列单独建模，使用得到的结果对模型结果进行偏差修正对于时序预测十分必要。

1.2 国内外研究现状

二十世纪末期，时序预测方向的数据挖掘算法得到国内外学者的广泛关注与研究，时序预测方法从早期基于数理统计学发展到如今基于深度学习，其发展过程可大概分为三个阶段：基于传统统计学、基于机器学习和基于深度学习。

1. 传统统计学预测方法

英国学者 Yule[2]于上世纪初期提出自回归模型(Autoregressive, AR)，自此，时序预测这一新兴学科得到科研学者的青睐。后来，Kendall 等人[3]将 Slutzky[4]提出的移动平均模型(Moving Average, MA)与自回归模型相融合，提出了自回归移动平均模型(Autoregressive Moving Average, ARMA)。至此，线性平稳时序数据的分析有了坚实的理论基础。1970 年，Box 等[5]提出差分自回归移动平均模型(Autoregressive Integrated Moving Average, ARIMA)，对于非平稳时序数据，该模型首先应用差分运算将其平稳化，随后应用 ARMA 模型对平稳时序数据建模。上述方法均要求时序数据必须是单变量且同方差。随着时序相关理论的发展，学者们研究的关注点逐渐转向异方差、多变量及非线性的时序数据。

针对异方差时序预测，先后有学者提出自回归条件异方差模型(Autoregressive Conditional Heteroskedasticity, ARCH)[6]和广义自回归条件异方差模型(Generalized Autoregressive Conditional Heteroskedasticity, GARCH)[7]。针对多变量时序建模，Box 等[8]提出了协整理论，解决了多维非平稳时序数据的预测这一难题。针对非线性、非平稳的时序数据预测建模，Tong 等[9]提出了门限自回归模型(Threshold Autoregressive,

TAR)。

基于传统统计学的时序预测模型使用假设条件严格、局限性较大，且模型参数单一、表达能力有限，不能很好地处理复杂时序数据。此外，这类模型仅对历史时序数据进行分析挖掘，没有考虑其他相关因素的影响，模型预测精度不高。

2. 机器学习预测方法

近年来机器学习开始兴起，使得时序预测可以通过使用时序历史数据重复训练以逼近真实模型。机器学习模型对非线性数据有更强的学习能力，这类模型的预测精度相较于经典统计学模型有显著提升。

Kim 等[10]应用支持向量机(Support Vector Machines, SVM)预测股票价格，并将 SVM 的预测精度与其他方法进行比较，实验证实，SVM 在股票价格预测中有更高的预测性能。Das 等[11]在贝叶斯网络(Bayesian Network, BN)的基础上提出了新的网络结构并将其应用于气象数据预测，该网络结构首次考虑气象变量间的时空关系。随后把时空信息融合到语义贝叶斯网络(Semantic Bayesian Networks, SBN)，提出了基于 SBN 的多元时序预测网络。实验验证 SBN 在气象时序预测中的可行性。黄卿等[12]将 XGBoost 应用到期货价格预测中，并其预测结果与 SVM 和 BP 神经网络进行对比，结果显示 XGBoost 模型更优。

基于机器学习的时序预测主要是建立函数方程，应用历史数据构造特征来预测未来值。尽管机器学习模型有良好的非线性建模和泛化能力，但其仅仅从特征的角度出发进行建模，模型效果很大程度上依赖于特征工程，这就要求对数据有较深的理解，存在较强的主观性，间接提高了时序预测的难度。

3. 深度学习预测方法

与机器学习通过繁重的特征工程来提高模型的预测精度不同，深度学习方法通过其复杂的结构自动挖掘特征间复杂的非线性关系。

Lapedes 等[13]首次将前馈神经网络模型(Feed Forward Network, FFN)应用到时序预测中，解决了混沌时序预测问题。Park 等[14]研究发现，除了历史时序数据，将温度数据等多变量传入神经网络可以更好的预测电力时序数据，精度远远高于回归模型。郝晓辰等[15]提出时序卷积神经网络模型，该模型应用多个不同的卷积核以挖掘

时序数据中的复杂依赖关系以及多个时序数据之间的关联关系。实验证实，相比于单一的卷积核，该模型在提高预测精度的同时，保证了模型的泛化能力。为充分利用时序数据中前后时刻元素的依赖性，Heimes 等[16]使用 RNN 模型预测发动机的剩余寿命，取得了较好的结果。但在反向传播距离过长时，RNN 模型在参数训练中会产生梯度消失或梯度爆炸。为此，Hochreiter 等[17]提出长短期记忆神经网络(Long-term and Short-term Memory Network, LSTM)，LSTM 通过在模型中添加时间记忆单元，使得 RNN 模型反向传播中的异常梯度得到了有效地解决，因此可以很好的处理和识别时序数据中的间隔和延迟，从而 LSTM 被广泛应用到时序预测任务中。Nelson[18]将 LSTM 运用于股票预测，并得到了比机器学习模型更高的精度。李高盛等[19]将 LSTM 应用到公交站台客流量预测，LSTM 预测效果比 ARIMA 和 SVR 有一定的提升。Cho 等[20]简化了 LSTM 节点结构，提出门控循环单元网络(Gated Recurrent Unit Networks, GRU)。Chung 等[21]将 LSTM 和 GRU 应用到语音信号建模中，实验结果显示：在训练数据规模一定时，两模型效果相差不大。

除上述仅使用单一模型进行时序预测，融合多种模型的集成学习方法也被广泛应用到时序预测中。为解决时序数据的位置依赖性，Lai 等[22]利用 CNN 和 RNN 建模，提出了长短期时间序列网络，但该方法需预先选定周期参数。对于多元时序预测，Cirstea 等[23]将 CNN 与 RNN 结合，首先对每个时序数据使用 CNN 提取卷积特征，之后对卷积特征应用 RNN 进行预测。此外，也有学者将信号分解引入时序预测中。

信号分解算法发展大概经历了傅里叶变换、小波变换和经验模态分解三个阶段。傅里叶变换无法保持精准的频率分辨率[24]，相比之下，小波变换同时保留了时间和频率的信息，在非平稳信号的处理中性能更好。但小波变换需要提前制定基函数，且信号分解效果强依赖于基函数的选取。1998 年，Huang 等人[25]提出经验模态分解(Empirical Mode Decomposition, EMD)，其主要思路是将信号分解成多个频率和振幅互不相同的固有模态函数(Initial Mass Function, IMF)，由于每个 IMF 均是从数据本身分解出的，因此 EMD 保证了数据的完全非平稳性。但是在处理含间断的信号时，EMD 方法存在模态混叠问题，为了解决这个问题，Wu 和 Huang[26]根据白噪声零均值，且在频域中是分布均匀的特点，提出集成经验模态分解(Ensemble Empirical Mode

Decomposition, EEMD)。EEMD 多次向信号中添加白噪声，对得到的新序列使用 EMD 进行分解，之后计算分解结果的平均值，由于噪声分量的均值为零，这就使得最后的平均值消除了噪声的影响，即结果为原信号的分解结果。然而，由于 EEMD 加入白噪声的缘故，导致集成后的 IMF 分量产生偏差，集成更加耗时。Yeh 等[27]提出互补集合经验模态分解(Complementray Ensemble Empirical Mode Decomposition, CEEMD)，该方法多次向信号中添加正负噪声对，之后的平均加权时消除了信号中的残余分量。CEEMD 克服了 EEMD 分量存在偏差、分解误差大的问题[28]。但原信号添加一个正负噪声对后的两个信号分解得到的分量个数可能存在差异，这导致 CEEMD 方法求 IMF 平均值时无法对齐，导致误差产生[29-30]。为解决这个问题，Torres 等[31]提出自适应噪声的完整集合经验模态分解(Complete Ensemble Empirical Mode Decomposition with Adaptive Noise, CEEMDAN)，CEEMDAN 分解向信号中添加自适应噪声而改变原信号的极值点，解决了不同的信号加噪声实现的不同模式数的问题。

2010 年，王文波[32]将 EMD 和 BP 神经网络融合，并应用到股票预测中，实验表明，相较于传统统计学模型，该模型的预测精度显著提高。2019 年，贺毅岳等[33]应用支持向量回归(Support Vector Regression, SVR)做时序预测，提出 EMD-SVR 模型，通过与 ARMA-GARCH 等模型对比分析，EMD-SVR 模型的精度更高。Li 等[34]提出 CEEMD-LSTM 模型预测城市空气 PM2.5 浓度，结果表明集成学习模型有较好的时序预测能力。2020 年，Zhang 等人[35]建立 CEEMD-LSTM 模型，并将该模型应用到金融时序预测中，实验结果表明该模型在保持较高精度的同时有着良好的鲁棒性。

1.3 本文研究内容及创新点

基于传统统计学的时序预测方法局限性大、模型表达能力有限，不能很好地预测非线性复杂时序数据；基于机器学习的时序预测方法主要是建立函数方程，具有良好的非线性建模能力和泛化效果，但机器学习模型的效果很大程度上依赖特征工程，且不能全面深入地挖掘序列中不同位置数据在时间维度上的相关性；基于深度学习的时序预测方法通过复杂的网络结构挖掘序列中数据依赖性，但训练或预测中仍存在一些问题，如训练中的梯度消失或梯度爆炸、对高振幅高频率序列预测存在延迟等。

基于上述时序预测的研究现状、发展方向和存在的问题，本文将 CEEMDAN、Transformer 模型和 SVR 模型相融合，提出基于 Transformer 的集成学习模型。由于各模型对高频率高振幅数据的预测均存在不同程度的问题，因此集成学习模型的第一步是利用 CEEMDAN 对原始时序数据进行分解，将不同振幅、频率分量拆分开，得多个子序列供后续单独处理与分析；其次，鉴于 Transformer 模型强大的自注意力机制可深度挖掘序列中不同位置元素在时间维度上的依赖性，因此集成学习模型的第二步是对分解得到的若干子序列使用 Transformerm 模型单独预测，得模型的中间预测结果；最后，为了解决模型对高频率高幅度分量子序列预测偏差较高的问题，集成学习模型的第三步是偏差修正，计算中间预测结果的残差序列，运用 SVR 对其进行回归建模，将残差序列的预测结果与中间预测结果融合得模型最终预测结果。

为了充分验证模型有效性，在实证分析部分，本文将提出的集成学习模型应用到 6 个时序预测经典数据集，包括 ETT、electricity、exchange_rate、traffic、illness 和 temperature，利用均方误差(Mean Square Error, MSE)和平均绝对误差(Mean Absolute Error, MAE)两项评估指标，选取时序预测中的经典统计学模型 ARIMA、机器学习模型 SVR 和深度学习模型 LSTM，对比分析模型的预测结果。结果表明：本文构建的集成学习模型有效提高了时序预测的精度。主要结论如下：

1. 深度学习模型的精度受样本个数影响较大，样本有限时，相较于深度学习模型，传统统计学模型和机器学习模型的预测精度及模型鲁棒性更强；
2. 得益于 Attention 机制，Transformer 模型可以更好地学习到序列中蕴含的信息，其效果优于 LSTM 模型；
3. 时序分解将序列中不同频率和振幅的分量分解开进行单独的模型学习和训练，可提高了模型预测精度；
4. 偏差二次修正通过单独处理模型在高频率高幅度子序列上的预测偏差，更有针对性地处理预测偏差，进而提升模型预测精度；
5. 时序数据振幅越大，集成学习模型提升越显著。

本文创新点主要有以下几点：

1. 利用注意力机制挖掘时序数据不同时刻数据间的复杂关系，引入信号分解和

偏差修正，将 Transformer 模型中与信号分解 CEEMDAN 和 SVR 偏差修正进行融合，得到基于 Transformer 的集成学习模型；

2. 为充分说明模型有效性，选用包括医疗数据、工业数据、金融数据、气象数据在内的经典时序数据集，利用 MSE 和 MAE 评价指标，将本文提出的基于 Transformer 的集成学习模型与 ARIMA、SVR、LSTM 模型进行对比分析。

1.4 本文组织框架

本文共由五章组成。

第 1 章为绪论。首先介绍了时序预测的研究意义，并简单介绍了时序预测领域的理论发展，提出目前时序预测存在的一些问题。其次，总结了国内外时序预测模型相关研究的发展历史。再次介绍了本文的研究内容，对研究思路、关键技术和创新点进行概述。最后阐述了本文的组织框架。

第 2 章为相关理论介绍。先介绍了信号分解相关研究的发展历程，其中重点介绍了经验模型分解中常用算法的流程；然后介绍了时序预测常用方法，包括差分自回归移动平均模型、支持向量回归机、长短期记忆神经网络和 Transformer 模型；最后对本章的内容进行了总结。

第 3 章为基于 Transformer 的集成学习模型构建。对本文提出的集成学习模型具体思路、方法、步骤进行了展示，首先应用 CEEMDAN 对时序数据进行分解，之后对各子序列使用 Transformer 模型进行单独训练和预测，最后使用支持向量回归 SVR 对 Transformer 模型的预测偏差建模预测，将 Transformer 模型对各子序列的预测结果与 SVR 偏差预测结果相融合，得集成学习模型最终预测结果。

第 4 章为时序预测实证分析。首先明确了实验环境、编程软件版本和模型配置参数含义等实验基本信息。其次介绍了各数据集的来源、数据的统计性指标和数据的预处理。再后使用 MSE 和 MAE 评估总结各单模型(ARIMA、SVR、LSTM)和基于 Transformer 的集成学习模型在各数据集上的预测性能。之后对各模型预测效果进行对比，分析各模型预测结果优劣性及产生原因。最后对本章内容进行了小结。

第 5 章为总结与展望。对本文的工作进行了总结，对未来研究方向进行了展望。

2 时序预测相关理论介绍

2.1 信号分解方法

一些信号在时域中难以看出其变化规律，尤其是信号中的频率特征，但在频域却可以明显观察其频率、周期等特征。将信号分解算法引入时序分解，得到的多个时序子序列都蕴含着各自独有的序列信息。信号分解算法发展先后经历了傅里叶变换、小波分解和经验模态分解三个阶段。

2.1.1 傅里叶变换

傅立叶变换(Fourier Transform, FT)利用时频转换理论，可以将信号分解成若干个分量，之后对分解所得分量的频率和相位进行观测，利用信号的频谱特征研究信号的变化规律[36]。FT 也可以用于滤波，对分解得到的若干个分量给予适当的权重来过滤不同频率的输入信号，从而实现滤波功能[37]。

2.1.2 小波变换

Morlet 和 Grossmam 于 1984 年提出小波变换(Wavelet Transform, WT)，从此小波变换开始受到关注。经过数十年研究，小波变换在信号分解领域取得了较理想的应用成果[38]。凭借其自适应特性，小波变换可以根据信号频率动态调节分解窗口的大小，克服了短时傅立叶变换(Short-time Fourier Transform, STFT)的窗口不能随信号频率变化而变化的问题[39]。

但小波变换仍然受到海森堡测不准原理的束缚，在时变信号分析中，其时域和频域分辨率虽然较 STFT 有所改进，但是仍然不能同时取得最优，且其时频分析效果很大程度上取决于基函数的选取，而对未知信号的时频分析，基函数选取又十分困难。

2.1.3 经验模态分解

1998 年 Huang[25]提出经验模态分解(Empirical Mode Decomposition, EMD)，可将信号转变为平稳状态后分解为不同波动和趋势的一系列信号。EMD 不需要指定分解的层数和基函数，可突破海森堡测不准原理的束缚。

EMD 的核心思想是将非平稳信号分解成若干个频率信号和残波，分解得到的分量称为固有模态函数(Initial Mass Function, IMF)。随着分解的进行，IMF 分量的周期逐渐扩大，频率逐渐降低。分解出得到的 IMF 需满足以下条件：

1. 零点与极值点最多相差 1 个；
2. 在信号任意点，上下包络均值为零。

信号 EMD 结果如(2-1)式所示。

$$x(t)=\sum_{i=1}^{k} IMF_i(t)+r(t), \tag{2-1}$$

其中，$x(t)$ 为原始信号，$IMF_i(t), i=1,2,...,k$ 为分解得到的 k 个 IMF，$r(t)$ 为原始信号去掉所有 IMF 后得到的残差。

算法 2.1 经验模态分解 EMD

输入：原始时序数据 $\{x(t), t=1,2,\cdots,N\}$.

输出：固有模态分量以及残差序列 $IMF_1, IMF_2, ..., IMF_k, r(t)$.

1. 计算 $x(t)$ 的全部局部极值点，求解极值点的下包络线 $e_{min}(t)$ 和上包络线 $e_{max}(t)$.
2. 计算上下包络线每一时刻的平均值得平均包络 $m_1(t)$.
3. 将原始时序数据 $x(t)$ 减去平均包络 $m_1(t)$，得到时序数据 $h_1(t)$，公式如下。

$$h_1(t)=x(t)-m_1(t), t=1,2,\cdots,N.$$

4. 判断 $h_1(t)$ 是否符合 IMF 标准，若不符合，则将 $h_1(t)$ 作为分解数据，重复 k 次分解直到 IMF 序列符合标准，最后将满足 IMF 标准的 $h_{1k}(t)$ 作为一阶 IMF 分量，用 IMF_1 表示。
5. 将 IMF_1 从原始时序数据 $x(t)$ 中分离得到分解后的剩余分量 $r_1(t)$.

$$r_1(t)=x(t)-IMF_1.$$

6. 将 $r_1(t)$ 作为新的分解数据重复 1~5 步，直至最后无法分离出新的 IMF 序列，则分解结束。原始时序数据 $x(t)$ 可以表示为所有 IMF 分量之和加上残差序列 $r(t)$.

$$x(t)=\sum_{i=1}^{k} IMF_i(t)+r(t).$$

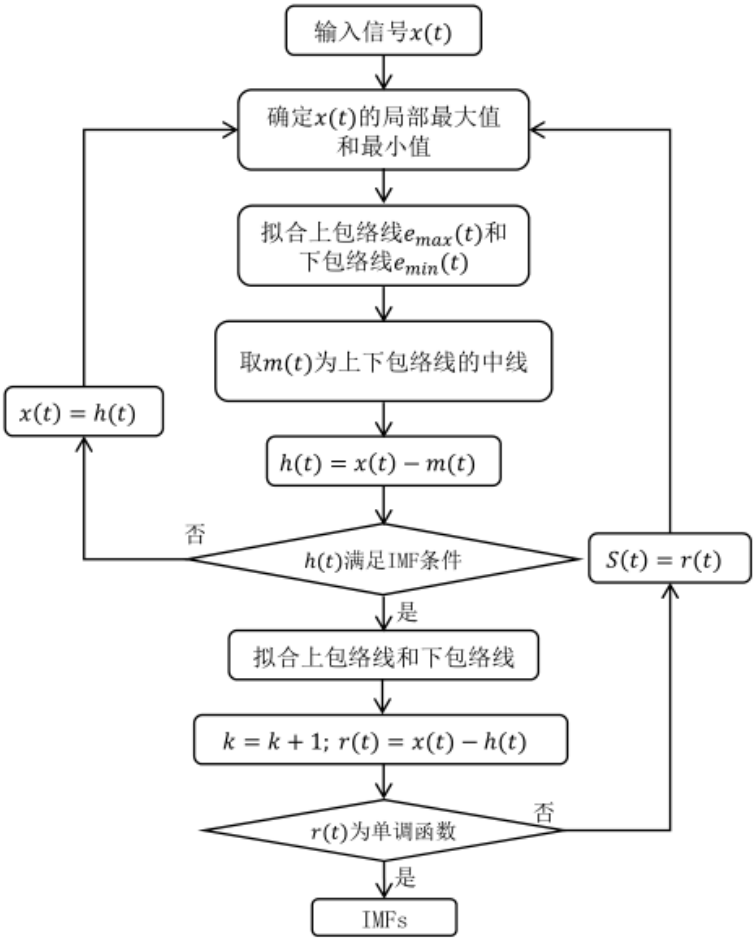

图 2-1　EMD 流程示意图

实际运用中发现 EMD 存在断点效应、模式混叠和难以判断分解停止条件等问题。为此，Handrin 等在 EMD 方法的基础上，利用频域中白噪声是分布均匀的特性，提出了集合经验模态分解(Ensemble Mode Decomposition, EEMD)。

EEMD 多次向信号中添加分布相同的白噪声，使得信号可以自动调整到适当的尺度上[40]。由于白噪声的零均值特点，多个信号平均之后白噪声将相互抵消，从而能够把平均集成得到的结果视为信号分解的最终结果。

算法 2.2 集合经验模态分解 EEMD

输入：原始时序数据$\{x(t), t=1,2,\cdots,N\}$，添加噪声次数M.

输出：输出固有模态分量以及残差序列$IMF_1(t), IMF_2(t), ..., IMF_J(t), r(t)$.

1. 向 $x(t)$ 中添加高斯白噪声 $n_i(t)$，得到新序列 $x_i(t)$.

$$x_i(t) = x(t) + n_i(t), t = 1, 2, \cdots, N, i = 1, 2, ..., M.$$

2. 对 $x_i(t)$ 使用 EMD 分解。

$$x_i(t) = \sum_{j=1}^{J} IMF_{i,j}(t) + r_i(t),$$

上式中，$IMF_{i,j}(t)$ 为使用 EMD 分解得到的第 j 个 IMF，$r_i(t)$ 为残差序列。

3. 重复 1~2 步 M 次，得到 IMF 集合。

$$IMF_{1,j}(t), IMF_{2,j}(t), ..., IMF_{M,j}(t), j = 1, 2, ..., J,$$
$$r_i(t), i = 1, 2, ..., M.$$

4. 对上述集合中阶数相同的 IMF 分量求均值，并计算残差序列的均值。

$$IMF_j(t) = \frac{1}{M}\sum_{i=1}^{M} IMF_{i,j}(t), r(t) = \frac{1}{M}\sum_{i=1}^{M} r_i(t),$$

其中，$IMF_j(t)$ 为 EEMD 分解得到的第 j 个 IMF，$r(t)$ 为分解后的残差序列。

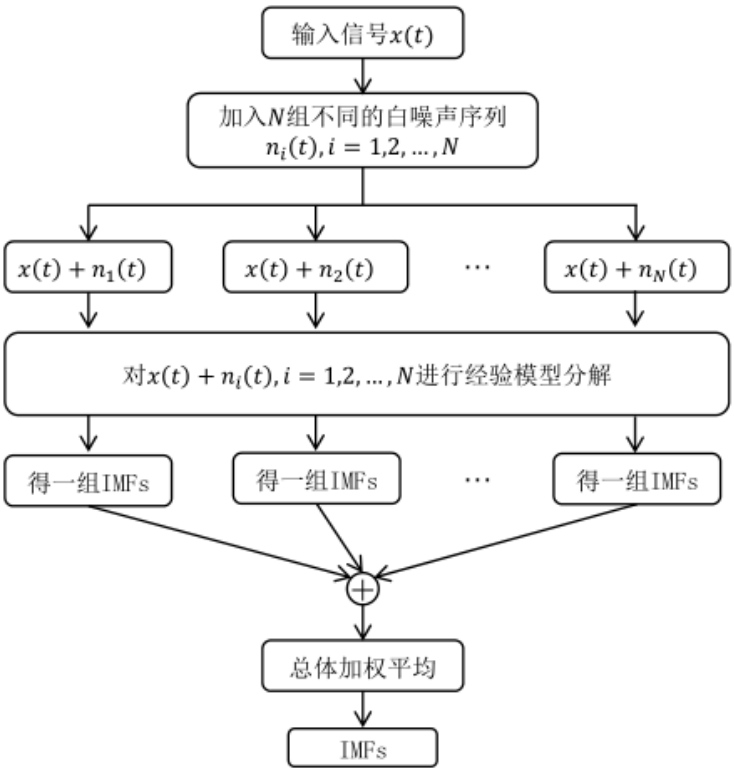

图 2-2　EEMD 流程示意图

EEMD 通过加入白噪声解决了 EMD 中的模态混叠等问题，但白噪声的加入导致集成后的 IMF 发生偏差，集成计算更加耗时，且理论上白噪声在集成后会相互抵消，但实际中存在噪声残留，尤其是在高频序列中。2011 年，Yeh 等[27]通过加入正负噪声消除信号中的噪声残留，提出了互补集合经验模态分解(Complementray Ensemble Empirical Mode Decomposition, CEEMD)。CEEMD 在 IMF 分量集合平均加权时消除了噪声分量，解决了集成偏差且耗时、误差大的问题，提高了信号分解的精度和效率。

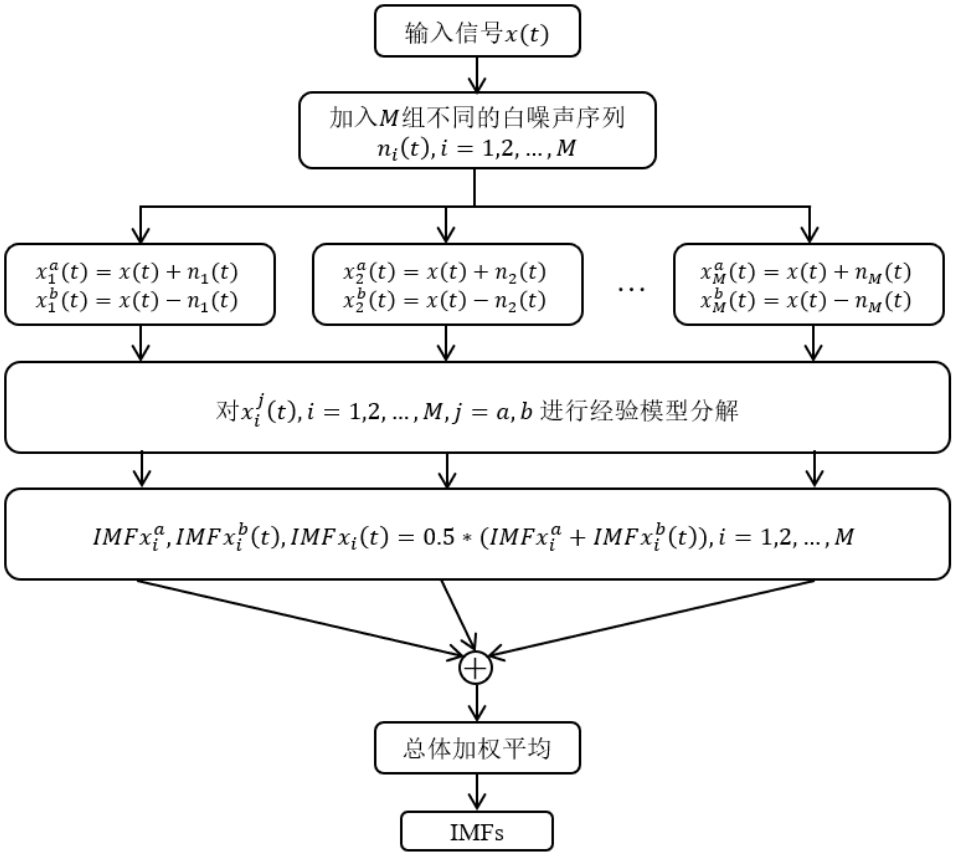

图 2-3　CEEMD 流程示意图

对加入正负噪声对的 $x_i^a(t)$ 和 $x_i^b(t)$ 进行 EMD 分解时，产生的 IMF 个数可能不同，使得最终集成平均时无法对齐。针对 CEEMD 的这个缺点，Torres 等[31]提出了完全自适应噪声集合经验模态分解(Complete Ensemble Empirical Mode Decomposition with Adaptive Noise, CEEMDAN)。该方法使用自适应高斯白噪声而非白噪声，消除直接加入白噪声产生的分量无法对齐问题，同时也克服了 EMD 中的模式混叠问题，从而降低误差，提高分解质量。

算法 2.3 完全自适应噪声集合经验模态分解 CEEMDAN

输入：原始时间序列数据$\{x(t),t=1,2,\cdots,N\}$，白噪声标准差ε，添加噪声次数M.

输出：固有模态分量以及残差序列$IMF_1(t),IMF_2(t),\cdots,IMF_K(t),r_K(t)$.

1. 调整标准差后，将噪声对$(\pm\varepsilon n_j(t))$加入序列$x(t)$，得新序列$x(t)+(-1)^q\varepsilon n_j(t)$，其中$q=1,2$，对其进行 EMD 分解，得到各序列的一阶固有模态函数$IMF_1^j(t)$.

$$E(x(t)+(-1)^q\varepsilon n_j(t))=IMF_1^j(t)+r^j(t),\ j=1,2,\cdots,M.$$

2. 计算M个一阶固有模态函数的均值，得 CEEMDAN 分解的一阶固有模态分量。

$$IMF_1(t)=\frac{1}{M}\sum_{j=1}^{M}IMF_1^j(t).$$

3. 计算原序列去除一阶固有模态函数后的残差。

$$r_1(t)=x(t)-IMF_1(t).$$

4. 判断残差序列是否为单调函数，若单调，则停止分解，否则重复上述步骤。

5. 若停止分解时固有模态函数为K个，得序列$x(t)$的 CEEMDAN 分解结果。

$$x(t)=\sum_{k=1}^{K}IMF_k(t)+r_K(t).$$

CEEMDAN 分解通过添加一组自适应高斯白噪声改变原始信号的极值分布情况，能够克服模式混叠、噪声对序列分解子序列无法对齐等问题，取得更好的分解效果，因此其理论和应用价值更为显著。

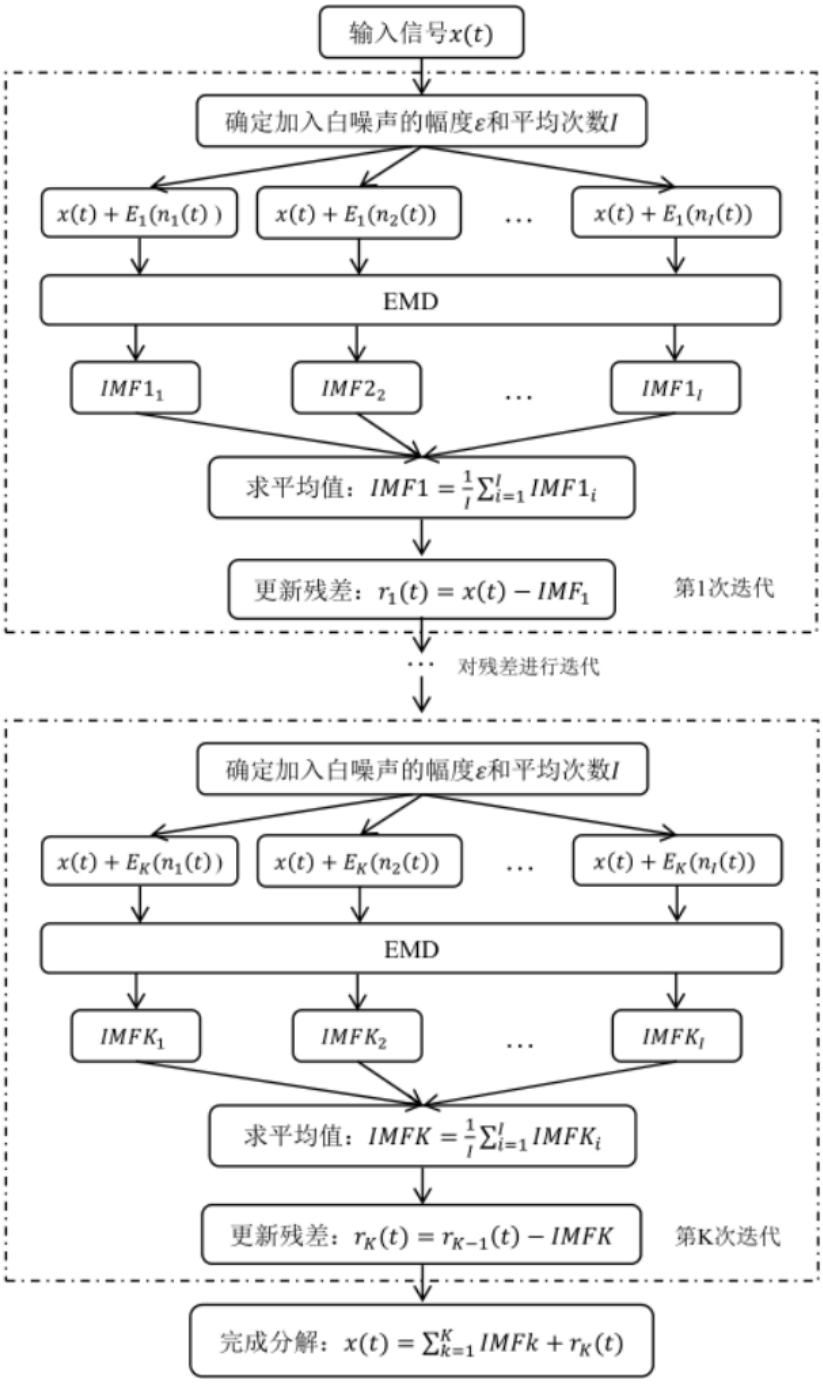

图 2-4　CEEMDAN 流程示意图

2.2 传统统计学方法

时序分析模型的发展先后历程了参数到非参数、线性到非线性。差分自回归移动平均模型(Auto Regressive Integrated Moving Average, ARIMA)作为一种经典线性参数模型，其应用的 Box-Jenkins 建模方法，被广泛运用于时序预测中。广义自回归条件异方差模型(Generalized Autoregressive Conditional Heteroskedastic, GARCH)是金融领域最常用的非线性参数模型之一，其对模型的预测误差的方差进行了分析，可用于非平稳金融时序数据的波动性进行建模分析[41]。但 GARCH 模型容易受到异常值的影响，且其对波动性较大的长序列预测效果稳定性较差。滑动转移自回归模型(Smooth Transition Autoregression, STAR)是数量经济学中的一种经典非线性参数时序预测模型，模型中的参数具有较强的经济学意义[42]。本文在时序预测实证分析中使用经典的 ARIMA 模型做模型对比分析，因此本节对 ARIMA 模型进行简单介绍。

统计学家 Jenkin 和 Box 于 1970 年联合发表经典时序预测著作[8]，提出经典的 ARIMA 模型，文章在 ARMA 模型的基础上，对 ARIMA 模型的原理、参数的识别、估计和假设检验进行了系统阐述。

ARIMA(p,d,q)模型的演变历程经历了 AR(p)，MA(q)和 ARMA(p,q)．下面对这几种模型进行简单阐述。

AR(p)为自回归模型，该模型假设序列$\{x_t, t=1,2,\cdots,N\}$在时刻t的取值x_t仅与时刻t前 p 个响应有关，而与这些时刻的扰动无关。 p 阶自回归可用(2-2)式描述。

$$x_t = \mu + \xi_t + \sum_{i=1}^{p} \gamma_i x_{t-i}, \tag{2-2}$$

上式中，x_t为序列在时刻t的取值， γ_i为自相关系数，p 为自回归阶数，μ 为常数项，ξ_t为残差。

MA(q)为移动平均模型，该模型可以刻画数据与其扰动值的关系，该模型假设序列在时刻t的取值x_t与时刻t前的响应无关，而与时刻t之前的q个时刻的扰动存在关联。 q 阶移动平均可用(2-3)式描述。

$$x_t = \mu + \xi_t + \sum_{i=1}^{q} \theta_i \xi_{t-i}, \tag{2-3}$$

上式中，θ_i 为偏自相关系数，q 为移动平均阶数，其余参数的含义与(2-2)式相同。

ARMA(p,q) 为自回归移动平均模型(Autoregressive Moving Average Models, ARMA)，该模型认为时序数据在时刻t的取值与该时刻前的若干个响应和扰动值均存在关系。ARMA(p,q) 模型可用(2-4)式表达。

$$x_t = \mu + \xi_t + \sum_{i=1}^{p} \gamma_i x_{t-i} + \sum_{i=1}^{q} \theta_i \xi_{t-i}, \tag{2-4}$$

上式中，参数的含义与(2-2)式和(2-3)式相同。

根据 Wold 分解定律，差分可以一个时序数据由非平稳变得平稳。ARIMA 模型的实质是对非平稳时序数据，使用差分运算将其平稳化，之后使用 ARMA 建模进行建模。ARIMA(p,d,q) 中 I 表示差分，d 为差分阶数，p 和 q 分别为自回归阶数和移动平均阶数，ARIMA(p,d,q) 模型可用(2-5)式表达。

$$(1-B)^d x_t = \mu + \xi_t + \sum_{i=1}^{p} \gamma_i x_{t-i} + \sum_{i=1}^{q} \theta_i \xi_{t-i}, \tag{2-5}$$

其中，B 为延迟算子，d 为差分阶数，其余参数含义与(2-2)式和(2-3)式相同。

ARIMA(p,d,q) 模型的自回归阶数 p 和移动平均阶数 q 需依据自相关函数(Autocorrelation function, ACF)和偏自相关函数(Partial Auto correlation Function, PACF)的实际值进行选取。

ACF 可衡量序列当前值 x_t 与其之前 k 个值之间的相关程度，其中包括直接和间接的相关性信息，其计算公式如(2-6)式。

$$\hat{\rho}_k = \frac{\sum_{t=1}^{n-k} (x_t - \bar{X})(x_{t+k} - \bar{X})}{\sum_{t=1}^{n} (x_t - \bar{X})^2}, \tag{2-6}$$

其中，$\hat{\rho}_k$ 为 ACF 的第 k 个值，$\bar{X}$ 为序列均值。

PACF 只衡量序列当前观测值 x_t 与滞后 k 项的观测值 x_{t-k} 之间的相关性，而降低了 x_t 和 x_{t-k} 之间的滞后项 $(x_{t-1}, x_{t-2}, \cdots, x_{t-k-1})$ 的影响。PACF 计算递推公式如(2-7)式。

$$\begin{cases} \hat{\phi}_{1,1} = \hat{\rho}_1, \\ \hat{\phi}_{k+1,k+1} = \dfrac{\hat{\rho}_{k+1} - \sum_{j=1}^{k} \hat{\phi}_{k,j} \hat{\rho}_{k+1-j}}{1 - \sum_{j=1}^{k} \hat{\phi}_{k,j} \hat{\rho}_j}, \\ \hat{\phi}_{k+1,j} = \hat{\phi}_{k,j} - \hat{\phi}_{k+1,k+1} \hat{\phi}_{k,k+1-j}, \; j = 1, 2, ..., k, \end{cases} \tag{2-7}$$

其中，$\hat{\phi}_{k,j}(j = 1, 2, ..., k-1)$为观测值$x_k$和其滞后项$x_j$之间偏自相关系数。

表 2-1　ARIMA 模型定阶基本原则

ACF	PACF	模型定阶
拖尾	p 阶截尾	$AR(p)$ 模型
q 阶截尾	拖尾	$MA(q)$ 模型
拖尾	拖尾	$ARMA(p,q)$ 模型

然而，表 2-1 的 ARIMA 模型定阶基本原则在实际建模中存在一定的困难。由于数据的随机性，样本的 ACF 和 PACF 并非会呈现出截尾的理想情况，而是会呈现出小幅度地震荡。另一方面，同一序列可由多个有效的 ARIMA 模型来拟合。由上述原因，在 ARIMA 模型定阶时往往采用 AIC 准则(Akaike Information Criterion, AIC)或 BIC 准则(Bayesian Inference Criterion, BIC)。

AIC、BIC 准则的思想可简述为：考虑到模型的预测精度和模型复杂程度，可以利用模型中待求参数的个数及其似然函数的值来度量模型效果，模型中的待求参数越少、样本的似然函数值越大，模型效果越好。AIC 函数和 BIC 函数如(2-8)式。

$$\begin{aligned} \mathrm{AIC} &= 2k - 2\ln(L), \\ \mathrm{BIC} &= k\ln(n) - 2\ln(L), \end{aligned} \tag{2-8}$$

其中，k 为模型中参数的个数，L 为样本似然函数值，n 为样本数量。AIC 或 BIC 的值越小的模型越优。

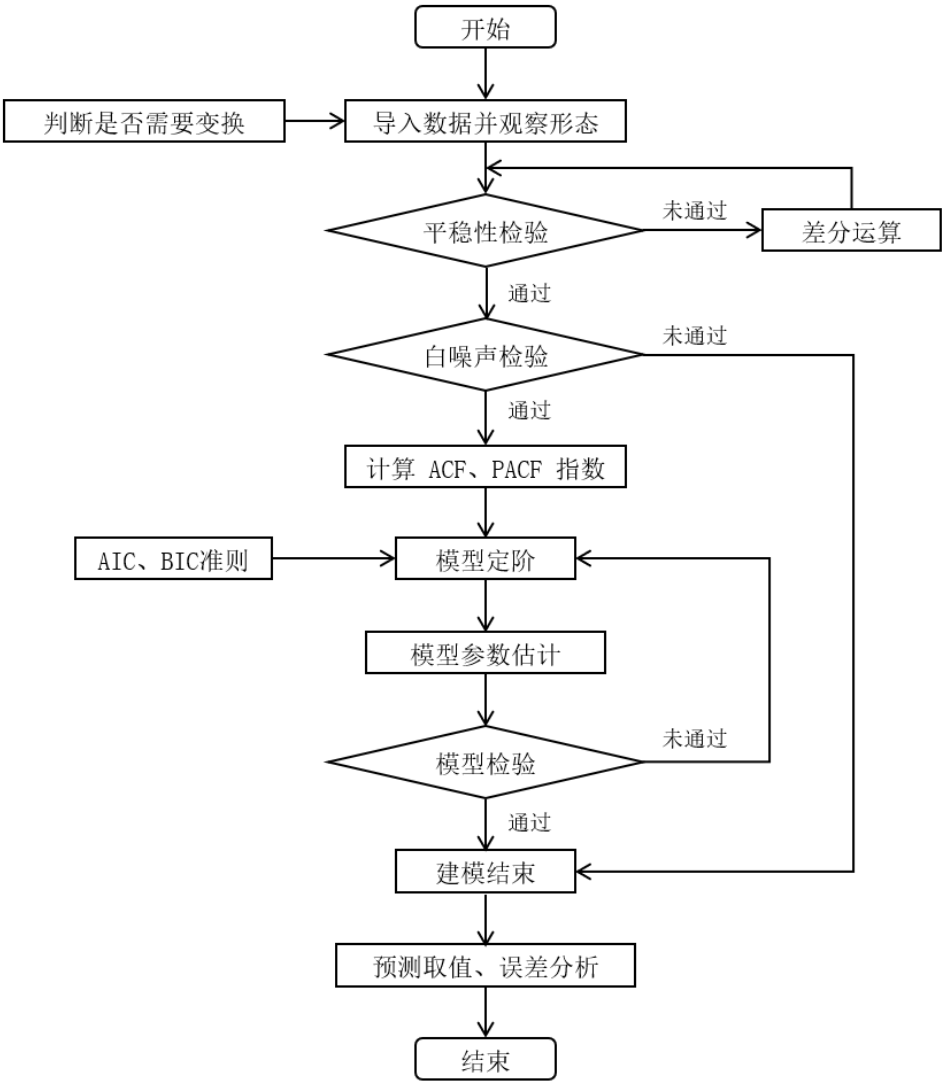

图 2-5　ARIMA 模型建模流程

2.3　机器学习方法

自机器学习迅速发展，有学者将时序预测归纳为回归问题，并应用机器学习方法进行时序预测，典型的模型有逻辑斯谛回归、支持向量回归机、XGBoost、LightGBM 等。本文在时序预测实证分析中选用支持向量机进行模型对比分析，因此接下来对支持向量回归机进行介绍。

1995 年，Cortes 和 Vapnik 等应用结构风险最小、统计学习理论的 VC 维和特征空间核函数等核心理论，提出支持向量机(Support Vector Machine, SVM)[43]。SVM 主

要应用于函数拟合以及非线性高维数据的分类与识别中，可分为支持向量分类机(Support Vector Classification, SVC)和支持向量回归机(Support Vector Regression, SVR)，本节主要介绍 SVR 模型。

设训练集样本集为$\{(x_1, y_1),(x_2, y_2),...,(x_N, y_N)\}$，$x_i=(x_{i,1},x_{i,2},...,x_{i,l})\in\mathbb{R}^l$ 为输入变量，$y_i\in\mathbb{R}$ 为输出变量，SVR 可以通过训练样本$\{(x_i,y_i),i=1,2,\cdots,N\}$构造输入变量 x 和输出变量 y 之间的函数。由样本数据信息的不同，SVR 又可分为线性和非线性[44]。

1. 线性回归

当样本特征为线性关系时，函数 $f(x)$ 的表达式如(2-9)式。

$$f(x)=w\cdot x+b, \tag{2-9}$$

其中，$x\in\mathbb{R}^l$ 为输入向量，$w\in\mathbb{R}^l$ 为权重，$b\in\mathbb{R}$ 为偏置，"·"表示内积运算。其最优化问题表示为(2-10)式。

$$\begin{aligned}&\min_{w,b}\ \frac{1}{2}\|w\|^2\\ &s.t.\ \ y_i-w\cdot x_i-b\le\varepsilon, i=1,2,...,N,\\ &\qquad w\cdot x_i+b-y_i\le\varepsilon, i=1,2,...,N,\end{aligned} \tag{2-10}$$

其中，ε 为不敏感损失值，即实际值和预测值之差小于等于 ε 时，认为结果准确，称为“硬间隔”，如图 2-6(a)所示。

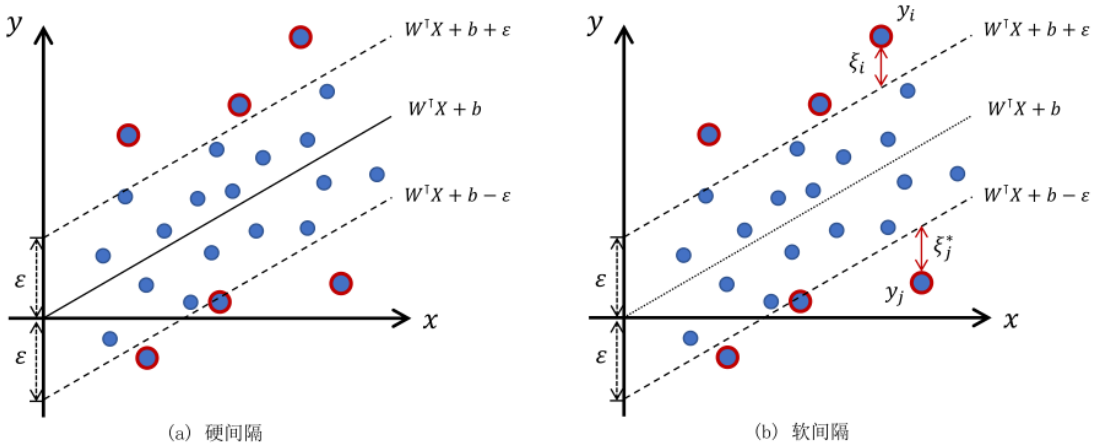

图 2-6　SVR 线性回归示意图

但在具体应用中，有些问题难以避免，为此引入松弛变量ξ_i,ξ_i^*，称为“软间隔”，

如图 2-6(b)所示，则(2-10)式最优化问题转化为(2-11)式。

$$\begin{aligned}\min_{w,b,\xi_i}\ & \frac{1}{2}\|w\|^2+C\cdot\frac{1}{N}\sum_{i=1}^{N}(\xi_i+\xi_i^*)\\ s.t.\ \ & y_i-((w\cdot x_i)+b)\le\varepsilon_i+\xi_i,\\ & ((w\cdot x_i)+b)-y_i\le\varepsilon_i+\xi_i^*,\\ & \xi_i\ge 0,\xi_i^*\ge 0,i=1,2,...,N,\end{aligned}\tag{2-11}$$

其中$C>0$为惩罚系数，C越大表示模型对超出ε界限的数据的惩罚越大。对于最优化问题(2-11)式，运用拉格朗日乘子法，得(2-12)式。

$$\begin{aligned}\max_{\alpha_i,\alpha_i^*,\beta_i,\beta_i^*}\min_{w,b}\ L=&\frac{1}{2}\|w\|^2+\frac{C}{N}\sum_{i=1}^{N}(\xi_i+\xi_i^*)-\sum_{i=1}^{N}(\beta_i\xi_i+\beta_i^*\xi_i^*)\\ &-\sum_{i=1}^{N}\alpha_i(\varepsilon_i+\xi_i-y_i+(w\cdot x_i)+b)\\ &-\sum_{i=1}^{N}\alpha_i^*(\varepsilon_i+\xi_i^*+y_i-(w\cdot x_i)-b),\end{aligned}\tag{2-12}$$

其中，$\alpha_i,\alpha_i^*,\beta_i,\beta_i^*\ge 0,i=1,2,...,N$为拉格朗日乘子。

求L对w,b,ξ_i,ξ_i^*的偏导数或梯度并令其等于零，得(2-13)式。

$$\begin{aligned}&\frac{\partial L}{\partial w}=w-\sum_{i=1}^{N}(\alpha_i-\alpha_i^*)x_i=0\to w=\sum_{i=1}^{N}(\alpha_i-\alpha_i^*)x_i,\\ &\frac{\partial L}{\partial b}=\sum_{i=1}^{N}(\alpha_i-\alpha_i^*)=0\to\sum_{i=1}^{N}(\alpha_i-\alpha_i^*)=0,\\ &\frac{\partial L}{\partial\xi_i}=\frac{C}{N}-\alpha_i-\eta_i=0\to\frac{C}{N}=\alpha_i+\eta_i,\\ &\frac{\partial L}{\partial\xi_i^*}=\frac{C}{N}-\alpha_i^*-\eta_i^*=0\to\frac{C}{N}=\alpha_i^*+\eta_i^*.\end{aligned}\tag{2-13}$$

整理后得最优化问题(2-14)式。

$$\begin{aligned}\min_{\alpha_i,\alpha_i^*}\ & L_D=\frac{1}{2}\sum_{i,j=1}^{N}(\alpha_i-\alpha_i^*)(\alpha_j-\alpha_j^*)(x_i\cdot x_j)+\varepsilon\sum_{i=1}^{N}(\alpha_i+\alpha_i^*)-\sum_{i=1}^{N}y_i(\alpha_i^*-\alpha_i)\\ s.t.\ \ & \sum_{i=1}^{N}(\alpha_i-\alpha_i^*)=0,\\ & 0\le\alpha_i,\alpha_i^*\le\frac{C}{N},i=1,2,...,N.\end{aligned}\tag{2-14}$$

利用 Karush-KarKuhn-Tucker(KKT)条件求解上述最优化问题，进而求得参数b的取值，确定如(2-15)式所示的回归方程。

$$f(x)=\sum_{i=1}^{N}(\hat{\alpha}_i^*-\hat{\alpha}_i)(x_i\cdot x)+\hat{b}, \tag{2-15}$$

其中，$\hat{\alpha}_i^*,\hat{\alpha}_i,\hat{b}$分别为$\alpha_i,\alpha_i^*,b$的最优解，$(x_i\cdot x)$为$x_i$与$x$的内积。

算法 2.4 线性$\varepsilon-$支持向量回归机

输入：训练集样本集为$\{(x_1,y_1),(x_2,y_2),...,(x_N,y_N)\},x_i\in\mathbb{R}^l,y_i\in\mathbb{R}$，其中$N$为训练集的样本总量。

输出：线性回归决策函数$f(x)=\sum_{i=1}^{N}(\hat{\alpha}_i^*-\hat{\alpha}_i)(x_i\cdot x)+\hat{b}.$

1. 选择适当的不敏感损失值$\varepsilon>0$和惩罚系数$C>0$.
2. 构造最优化问题。

$$\begin{aligned}\min_{\alpha_i,\alpha_i^*}\ & \frac{1}{2}\sum_{i,j=1}^{N}(\alpha_i-\alpha_i^*)(\alpha_j-\alpha_j^*)(x_i\cdot x_j)+\varepsilon\sum_{i=1}^{N}(\alpha_i+\alpha_i^*)-\sum_{i=1}^{N}y_i(\alpha_i^*-\alpha_i)\\ s.t.\ & \sum_{i=1}^{N}(\alpha_i-\alpha_i^*)=0,\\ & 0\le\alpha_i,\alpha_i^*\le\frac{C}{N},i=1,2,...,N.\end{aligned}$$

3. 求解上述最优化问题得参数估计值$\hat{\alpha}=(\hat{\alpha}_1,\hat{\alpha}_2,\cdots,\hat{\alpha}_N)^{\mathrm{T}},\hat{\alpha}^*=(\hat{\alpha}_1^*,\hat{\alpha}_2^*,\cdots,\hat{\alpha}_N^*)^{\mathrm{T}}$.
4. 按下列方式计算$\hat{b}$，从$(0,\frac{C}{N})$中选择$\hat{\alpha}_j$或$\hat{\alpha}_k^*$的一个分量。选到$\hat{\alpha}_j$或$\hat{\alpha}_k^*$，则$\hat{b}$分别按照下式计算。

$$\hat{b}=y_j-\sum_{i=1}^{N}(\hat{\alpha}_i^*-\hat{\alpha}_i)(x_i\cdot x_j)+\varepsilon,$$

$$\hat{b}=y_k-\sum_{i=1}^{N}(\hat{\alpha}_i^*-\hat{\alpha}_i)(x_i\cdot x_k)-\varepsilon.$$

5. 构造所求线性回归的决策函数。

$$f(x)=\sum_{i=1}^{N}(\hat{\alpha}_i^*-\hat{\alpha}_i)(x_i\cdot x)+\hat{b}.$$

2. 非线性回归

当训练集为非线性时，可应用非线性函数将原始数据升维，使得升维后的数据可线性回归，之后在升维后的特征空间进行线性回归。上述过程的得到的非线性回归公式如(2-16)式。

$$f(x)=w\cdot\phi(x)+b, \tag{2-16}$$

其中，$w\in\mathbb{R}^M, b\in\mathbb{R}$，$M$ 为特征升维后的维数，$\phi(x)$ 为输入特征升维的变换函数。此时，最优化问题表现形式如下(2-17)式。

$$\begin{aligned}\min_{w,b,\xi_i}\ & \frac{1}{2}\|w\|^2+\frac{C}{N}\sum_{i=1}^{N}(\xi_i+\xi_i^*)\\ s.t.\ & y_i-w\cdot\phi(x_i)-b\le\varepsilon_i+\xi_i, i=1,2,...,N,\\ & w\cdot\phi(x_i)+b-y_i\le\varepsilon_i+\xi_i^*, i=1,2,...,N,\\ & \xi_i\ge 0,\xi_i^*\ge 0, i=1,2,...,N.\end{aligned} \tag{2-17}$$

与线性情况的求解过程类似，非线性最优化问题(2-17)式的对偶问题为(2-18)式。

$$\begin{aligned}\min_{\alpha_i,\alpha_i^*}\ & L_D=\frac{1}{2}\sum_{i,j=1}^{N}(\alpha_i-\alpha_i^*)(\alpha_j-\alpha_j^*)K(x_i,x_j)+\varepsilon\sum_{i=1}^{N}(\alpha_i+\alpha_i^*)-\sum_{i=1}^{N}y_i(\alpha_i^*-\alpha_i)\\ s.t.\ & \sum_{i=1}^{N}(\alpha_i-\alpha_i^*)=0,\\ & 0\le\alpha_i,\alpha_i^*\le\frac{C}{N}, i=1,2,...,N,\end{aligned} \tag{2-18}$$

其中，$K(x_i,x_j)=\phi(x_i)\cdot\phi(x_j)$ 称为核函数。

利用 Karush-KarKuhn-Tucker(KKT)条件求解上述最优化问题，进而求得参数 b 的取值，求解得非线性函数回归方程(2-19)式。

$$f(x)=\sum_{i=1}^{N}(\hat{\alpha}_i^*-\hat{\alpha}_i)K(x_i,x)+\hat{b}, \tag{2-19}$$

其中，$\hat{\alpha}_i^*,\hat{\alpha}_i,\hat{b}$ 分别为 α_i,α_i^*,b 的最优解。

算法 2.5 非线性 ε – 支持向量回归机

输入：训练集样本集为 $\{(x_1, y_1),(x_2, y_2),...,(x_N, y_N)\}, x_i \in \mathbb{R}^l, y_i \in \mathbb{R}$，其中 N 为训练集的样本总量.

输出：决策函数 $f(x)=\sum_{i=1}^{N}(\hat{\alpha}_i^* - \hat{\alpha}_i)K(x_i, x)+\hat{b}.$

1. 选择核函数 $K(x, x')$，$\varepsilon > 0$ 和 $C > 0$.

2. 构造最优化问题。

$$\min_{\alpha_i, \alpha_i^*} \ \frac{1}{2}\sum_{i,j=1}^{N}(\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*)K(x_i, x_j)+\varepsilon\sum_{i=1}^{N}(\alpha_i + \alpha_i^*)-\sum_{i=1}^{N} y_i(\alpha_i^* - \alpha_i)$$

$$s.t. \quad \sum_{i=1}^{N}(\alpha_i - \alpha_i^*) = 0,$$

$$0 \le \alpha_i, \alpha_i^* \le \frac{C}{N}, i = 1, 2, ..., N.$$

3. 求解上述最优化问题得参数估计值 $\hat{\alpha} = (\hat{\alpha}_1, \hat{\alpha}_2, \cdots, \hat{\alpha}_N)^{\mathrm{T}}, \hat{\alpha}^* = (\hat{\alpha}_1^*, \hat{\alpha}_2^*, \cdots, \hat{\alpha}_N^*)^{\mathrm{T}}$.

4. 按下列方式计算 $\hat{b}$，从 $(0, \frac{C}{N})$ 中选择 $\hat{\alpha}_j$ 或 $\hat{\alpha}_k^*$ 的一个分量。选到 $\hat{\alpha}_j$ 或 $\hat{\alpha}_k^*$，则 $\hat{b}$ 分别按照下式计算。

$$\hat{b} = y_j - \sum_{i=1}^{N}(\hat{\alpha}_i^* - \hat{\alpha}_i)K(x_i, x_j)+\varepsilon,$$

$$\hat{b} = y_k - \sum_{i=1}^{N}(\hat{\alpha}_i^* - \hat{\alpha}_i)K(x_i, x_k)-\varepsilon.$$

5. 构造所求非线性回归决策函数。

$$f(x)=\sum_{i=1}^{N}(\hat{\alpha}_i^* - \hat{\alpha}_i)K(x_i, x)+\hat{b}.$$

选择合适的核函数对模型的建模效果十分关键。常用核函数包括以下几种：

1. 线性核函数：

$$K(x, y) = x \cdot y. \tag{2-20}$$

线性核函数参数少、训练快，适用于特征空间维度与样本个数接近时使用；

2. 多项式核函数：

$$K(x, y) = (xy+1)^p, p = 1, 2, ..., N. \tag{2-21}$$

多项式核函数适用于有限阶幂函数作为基向量对应的高阶特征空间中的线性问题中使用，但是阶数 p 难以选择。

3. 径向基核函数：

$$K(x,y)=\exp\left\{-\frac{(x-y)^2}{2\sigma^2}\right\}, \tag{2-22}$$

其中，σ 为核函数的宽度。适用于大部分数据，无论特征维度或高或低、样本或多或少，径向基核函数均可表现出较好的性能，其复杂度不随参数的变化而变化。

4. sigmoid 核函数：

$$\begin{aligned} K(x,y)&=\tanh(kx\cdot y+\theta), k>0, \theta<0, \\ \tanh(x)&=\frac{\sinh(x)}{\cosh(x)}=\frac{1-e^{-2x}}{1+e^{-2x}}. \end{aligned} \tag{2-23}$$

虽然 sigmoid 核函数是非正定核，但是在一些刻画概率的问题中很管用。

应用 SVR 建模预测时，不同的核函数会得到不同的预测结果，即使是同一个核函数，参数取值不同，结果也不一样。因此核函数的确定确定需要根据数据情况进行确定，且超参数需要训练择优。

2.4 深度学习方法

深度学习方法中，最具代表性的为卷积神经网络和递归神经网络，二者网络结构上存在差异。本节介绍在递归神经网络基础上改进的长短时记忆神经网络和基于注意力机制提出的 Transformer 模型。

2.4.1 长短时记忆神经网络

传统循环神经网络(Recursive Neural Network, RNN)的参数训练采用反向传播算法，当训练序列数据较长时，回传值陡增或陡降，使得训练无法收敛。长短时记忆神经网络(Long-term and Short-term Memory Network, LSTM)提出将信息存储在一个个节点中[17]，避免了传统 RNN 模型训练中长序列信息无法传递。

RNN 采用循环结构来保证信息的持久传递，其中循环单元结构如图 2-7 所示。

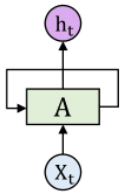

图 2-7　RNN 循环单元

RNN 循环单元结构中，神经网络节点 A 根据时刻t的值 X_t，计算学习后输出下一时刻的预测值 h_t. 可以将 RNN 结构看作是多个循环单元的联接，即每个神经网络模块都将学习到的信息向下传递。RNN 结构展开图如图 2-8 所示。

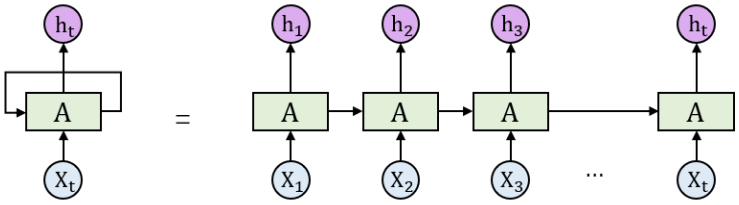

图 2-8　RNN 结构展开图

从图 2-8 可以看出，RNN 的链式结构可以实现历史信息的向后传递，很好的学习序列中的信息，因此可以很自然地应用到时序预测中。LSTM 将 RNN 中的节点进行巧妙设计，改造后的结构有更好的表现，LSTM 的网络结构如图 2-9 所示。

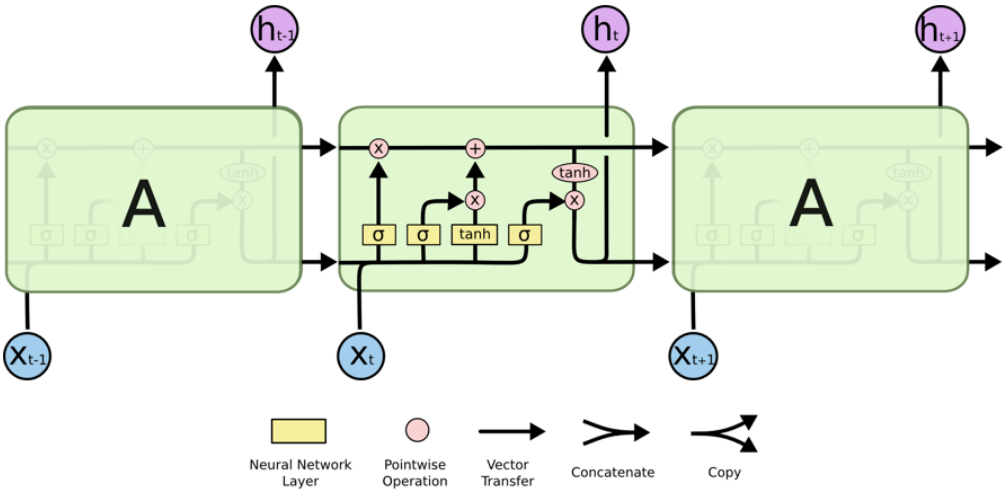

图 2-9　LSTM 网络结构示意图

LSTM 通过节点状态更新进行学习和信息挖掘，而节点内部以及相邻节点之间通过如图 2-10 所示的门结构进行信息流的传递。

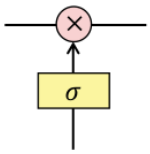

图 2-10　LSTM 门结构

门结构由一个 sigmoid 函数和点乘运算构成，sigmoid 函数用计算有多少信息可以被传递，其定义如(2-24)式，当函数值为 0 时，信息不传递，为 1 时全部传递。

$$f(x)=\frac{1}{1+e^{-x}}. \tag{2-24}$$

每个 LSTM 节点通过三个控制门实现信息流的传递与更新，详细结构如图 2-11 所示，下面详细介绍单个节点中信息流转路径以及三个控制门的作用和计算公式。

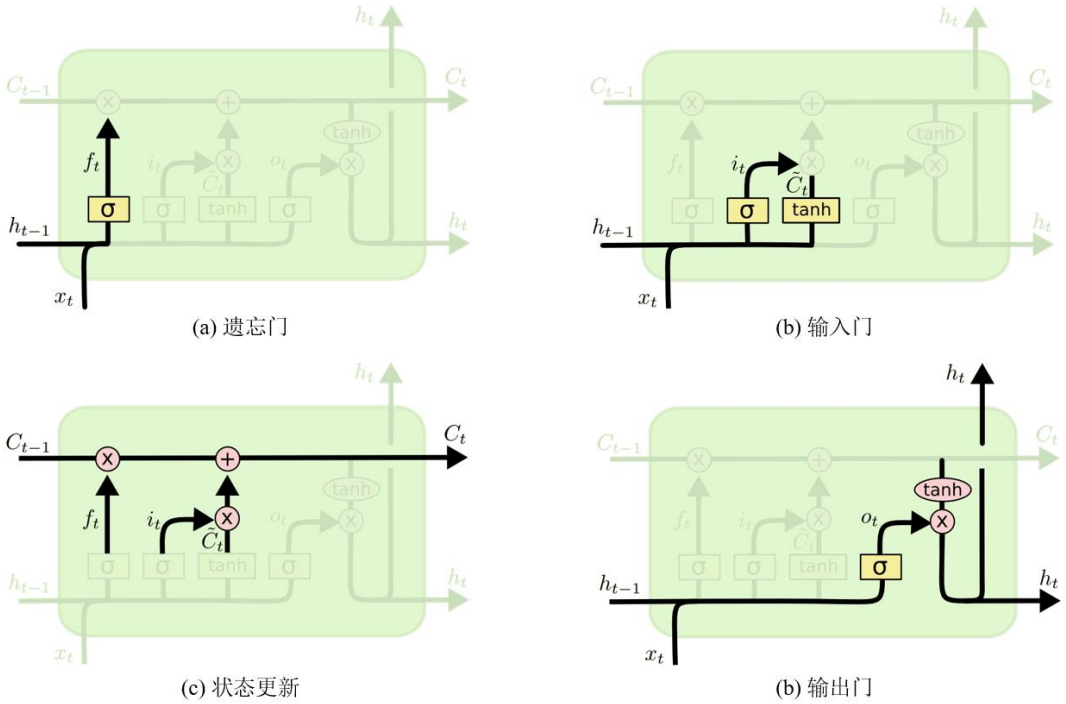

图 2-11　LSTM 节点结构示意图

遗忘门：该控制门决定上一时刻节点中的信息有多少进入当前时刻信息的计算。其结构如图 2-11(a)所示，输入为上一时刻的预测值以及当前时刻的真实值，按照(2-25)式计算输出。

$$f_t=\sigma(W_f\cdot[h_{t-1},x_t]+b_f), \tag{2-25}$$

其中，h_{t-1}为上一时刻的预测值，x_t为当前时刻的真实值，W_f和b_f为待学习的参数，

$\sigma(\cdot)$为 sigmoid 函数，公式如(2-24)式，f_t为遗忘门的输出，其取值范围是 0 到 1，节点状态更新时，它决定了上一时刻节点状态包含的信息有多少进入到当前时刻计算。

输入门：该控制门决定当前时刻输入信息有多少需要保留，即决定当前时间状态输入值以及上一时间状态的输出值需要保留多少。其结构如图 2-11(b)所示，输入为上一时刻的预测值以及当前时间的真实值，按照(2-26)式和(2-27)式计算输出。

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i), \tag{2-26}$$

其中，x_t，h_{t-1}和$\sigma(\cdot)$的含义与(2-25)式相同，W_i和b_i为待学习的参数，i_t为输入门的输出。

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C), \tag{2-27}$$

其中，x_t和h_{t-1}的含义与(2-25)式相同，W_C和b_C为待学习的参数，$\tanh(\cdot)$为双曲正切函数，公式如(2-23)式，$\tilde{C}_t$为待加入节点状态中的信息。

状态更新：状态更新是将输入门和遗忘门的输出进行合并，并传递给下一时刻的节点。如图 2-11(c)所示。更新后的节点中存储了部分历史信息和当前输入新增的信息，其计算公式如(2-28)式。

$$C_t = f_t \times C_{t-1} + i_t \times \tilde{C}_t, \tag{2-28}$$

其中，f_t, i_t和$\tilde{C}_t$的含义与(2-25)式、(2-26)式和(2-27)式相同，C_{t-1}为上一时刻节点中的信息，C_t为当前时刻更新后的节点信息。状态更新这一结构使得当前时刻的节点可以从上一时刻节点中选择部分信息进行学习，即信息转递。

输出门：该控制门决定当前节点中的多少信息被输出以及传递给下一时刻的节点。其结构如图 2-11(d)所示，输入为上一时刻的预测值以及当前时间的真实值，按照(2-29)式和(2-30)式计算输出门的输出。

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o), \tag{2-29}$$

其中，x_t, h_{t-1}和$\sigma(\cdot)$的含义与(2-25)式相同，W_o和b_o为待学习的参数，o_t为输出门的输出。

$$h_t = o_t \times \tanh(C_t), \tag{2-30}$$

其中，h_t 为模型输出值，即模型在当前时刻的预测值，同时会传递给下一时刻的节点，作为下一时刻节点的输入。

以上为 LSTM 模型的网络结构及计算公式，其参数的训练过程与经典 RNN 相同，采用反向传播算法。

2.4.2 Transformer

Google 团队于 2017 年提出 Transformer 模型[45]，凭借强大的注意力机制，Transformer 模型在语言翻译、语义理解等人工智能实际应用中取得了较好的效果。

Transformer 模型整体由若干个编码器(Encoder)和若干个解码器(Decoder)组成，其结构如图 2-12 所示。具体而言，模型包含多头注意力模块(Multi-Head Attention)，前馈神经网络层(Feed Forward)和残差连接与归一化层(Add & Norm)模块，相邻模块之间都有一个残差连接与归一化模块，残差连接可避免梯度消失和梯度爆炸，而归一化可在模型训练中加速收敛。另外，编码器和解码器之间还有一层编码器-解码器注意力层。下面详细讲解 Transformer 模型中的组件，包括序列编码(Embedding 和 Positional Embedding)、多头自注意力、前馈神经网络以及标准化与归一化。

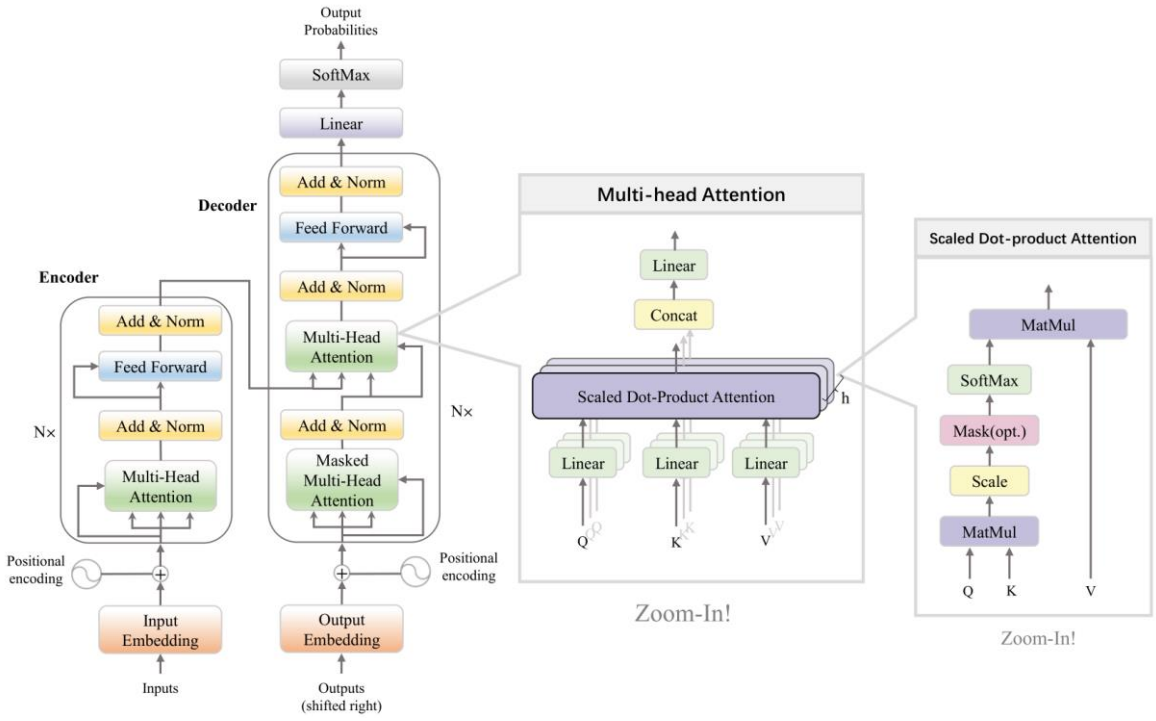

图 2-12　Transformer 模型结构示意图

2.4.2.1 序列编码

图 2-7 可以看出，首个解码器的输入是输入序列数据的 Embedding 编码和位置编码，可以将两者进行相加或拼接，之后每个编码器的输入为上一个编码器的输出。

不同于经典 RNN 训练过程中的迭代操作，Transformer 模型中没有天然存在的数据间前后序列关系。为此，Transformer 模型尝试采用多个不同频率的正余弦函数作为位置信息加入到输入序列中，位置编码计算公式如(2-31)式。

$$PE(t,2i)=\sin(\frac{t}{10000^{2i/d_{\text{model}}}}),PE(t,2i+1)=\cos(\frac{t}{10000^{2i/d_{\text{model}}}}), \tag{2-31}$$

上式中，t为时间步，i为输入特征的维度编号，d_{model}为输入特征的维度总数。编码长度和编码器个数为超参数，可以根据业务自行设定或训练确定。

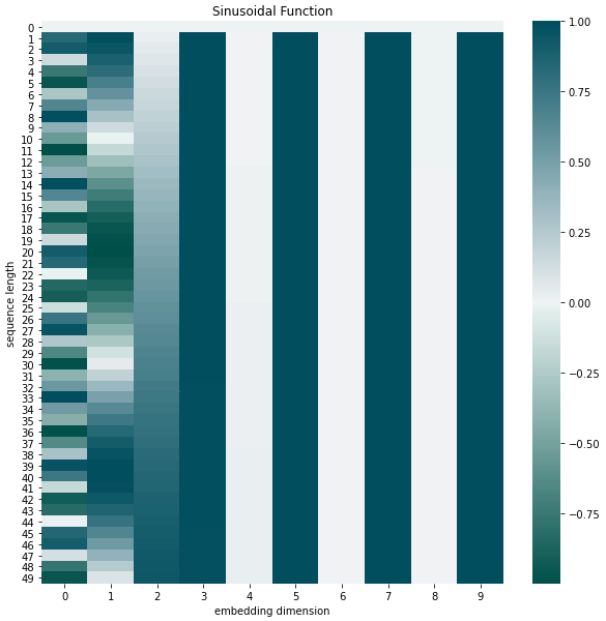

图 2-13　位置编码正余弦函数取值情况

2.4.2.2 多头自注意力机制

注意力机制(Attention Mechanism)可以理解为寻找序列 X 不同位置之间的关联性，其计算过程是将输入乘以查询向量参数矩阵 Q、键向量参数矩阵 K 和值向量参数矩阵 V 转化为三个向量，然后计算某个查询向量的子注意力向量。而多头注意力机制就是同时进行多个注意力机制的训练，且使用的参数矩阵 Q,K,V 互不相同，以实现从多个不同维度进行学习和挖掘序列信息。

Attention 计算过程可总结成(2-32)式。

$$\begin{aligned} Q &= \text{Linear}(X) = XW^Q, \\ K &= \text{Linear}(X) = XW^K, \\ V &= \text{Linear}(X) = XW^V, \\ \text{Attention}(Q,K,V) &= \text{softmax}(\frac{QK^{\mathrm{T}}}{\sqrt{d_k}})V, \end{aligned} \tag{2-32}$$

其中，X 为输入矩阵，W^Q, W^K, W^V 为线性变换权重矩阵，$Q \in \mathbb{R}^{n\times d_k}$ 为查询向量参数矩阵，$K \in \mathbb{R}^{m\times d_k}$ 为键向量参数矩阵，$V \in \mathbb{R}^{m\times d_v}$ 为值向量参数矩阵。Softmax 函数用于计算注意力权重。记 $Q=[q_1,q_2,\cdots,q_n]^T$，$K=[k_1,k_2,\cdots,k_m]^T$，$V=[v_1,v_2,\cdots,v_m]^T$，可以看到 k 和 v 是一一对应的。单看 Q 中的每一个元素，有(2-33)式。

$$\text{Attention}(q_t,K,V) = \sum_{s=1}^{m}\frac{1}{Z}\exp(\frac{q_t k_s^T}{\sqrt{d_k}})v_s, t=0,1,\cdots,n, \tag{2-33}$$

其中，Z 为归一化因子。从上式可以看出，每个 q_t 都被编码成了 $v_1,v_2,\cdots,v_m$ 的加权和，权重由 q_t 和 k_s 的共同决定。其中，缩放因子 $\sqrt{d_k}$ 起到调节作用。上述计算过程称为缩放点乘注意力(Scaled Dot-Product Attention)，如图 2-12 所示。

每组 Q,K,V 可提取序列中某一方面的信息，因此可以使用多组 Q,K,V 进行自注意力运算，用于提取多个方面的信息。之后将得到的多个输出结果进行拼接，以提高模型性能。基于上述分析，Google 团队提出了多头注意力机制(Multi-Head Attention Mechanism)，其定义如下。

$$\begin{aligned} \text{head}_i &= \text{Attention}(QW_i^Q, KW_i^K, VW_i^V), \\ \text{MultiHead}(Q,K,V) &= \text{Concat}(\text{head}_1,\text{head}_2,\cdots,\text{head}_h)W^O, \end{aligned} \tag{2-34}$$

其中，$W_i^Q, W_i^K \in \mathbb{R}^{d_k\times\tilde{d}_k}, W_i^V \in \mathbb{R}^{d_V\times\tilde{d}_V}$ 为第 i 头的线性变换权重矩阵，W^O 为降维矩阵。简单来说，Multi-Head Attention 就是使用 h 个不同的 Q,K 和 V 组合计算 Attention，之后把得到的 h 个 Attention 计算结果拼接起来，最后输出一个 $n\times(h\tilde{d}_v)$ 的序列。

在 Transformer 中，Attention 都是自注意力(Self Attention)，就是在一个序列内部做 Attention，即 $\text{Attention}(X,X,X)$，更准确地说，是 Multi-Head Self Attention，即 $\text{MultiHead}(X,X,X)$，此时，上述多头注意力机制计算过程如图 2-14 所示。

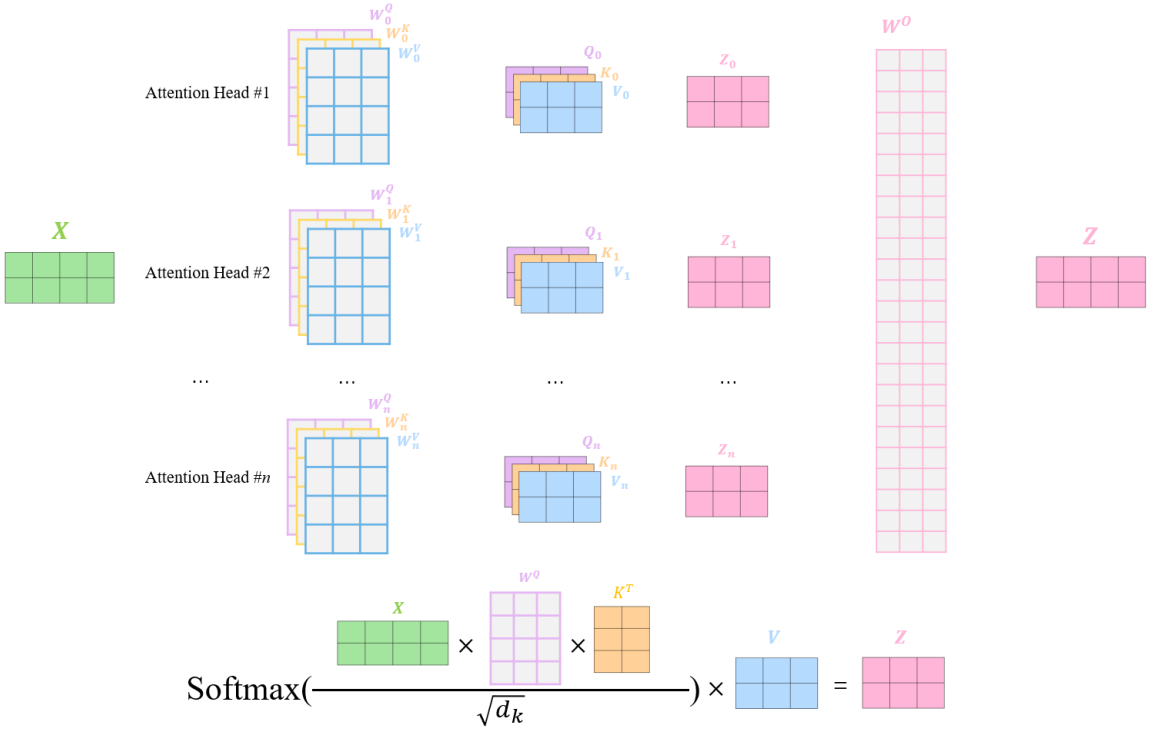

图 2-14　多头注意力矩阵计算过程示意图

2.4.2.3　前馈神经网络

前馈神经网络(Feed Forward Network, FFN)可实现非线性变换，Transformer 模型中的每个 Multi-Head Attention 模块的输出均流向 FNN 模块，各个 FFN 的训练相互独立。FFN 公式如(2-35)式。

$$\mathrm{FFN}(x)=\max(0,xW_1+b_1)W_2+b_2, \tag{2-35}$$

其中，W_1,W_2,b_1,b_2 为模型待学习参数，其维度根据输入数据的维度和自定义输出维度决定。

2.4.2.4　残差连接与归一化层

Transformer 模型中的每个子层输出之后都有残差连接与归一化模块。残差连接就是向输出中加入输入，可防止权重矩阵发生退化，进而避免梯度爆炸和梯度消失。残差连接的计算公式如(2-36)式。

$$X=X_{\text{input}}+\mathrm{SelfAttention}(Q,K,V). \tag{2-36}$$

归一化就是将输出矩阵中的元素按照所在列数据的均值和方差进行归一化，使得各隐藏变量服从标准正态分布。归一化操作可以加速模型训练收敛，对于输出矩阵

X 中的元素 x_{ij}，归一化的计算公式如(2-37)式。

$$\begin{aligned}\mu_j &= \frac{1}{m}\sum_{i=1}^{m} x_{ij}, \sigma_j^2 = \frac{1}{m}\sum_{i=1}^{m}(x_{ij}-\mu_j)^2, \\ \text{LayerNorm}(x_{ij}) &= \frac{x_{ij}-\mu_j}{\sqrt{\sigma_j^2+\varepsilon}},\end{aligned} \tag{2-37}$$

其中，μ_j 和 σ_j^2 分别为 j 个样本的均值和方差，$\varepsilon > 0$ 是为了避免方差为零而添加的调节因子。

2.5 本章小结

本章主要介绍时序预测中相关方法，分为信号分解、传统统计学时序预测、机器学习时序预测和深度学习时序预测。其中，信号分解部分简单介绍了相关研究的研究历程，总结各信号分解算法流程及优缺点，重点阐述了经验模态分解的四种算法；传统统计学方法主要介绍 ARIMA 模型相关理论和建模流程；机器学习方法主要介绍 SVR 模型的理论和细节；深度学习方法则主要介绍 LSTM 和 Transformer 模型的理论和细节。重点介绍将在下一章中用于时序分解的经验模态分解 CEEMDAN、偏差修正的支持向量回归 SVR 和时序预测的 Transformer 模型。

3 基于 Transformer 的集成学习模型构建

传统的时序预测方法局限性大、模型表达能力有限，不能很好地预测非线性复杂时序数据；基于机器学习的时序预测方法主要是建立函数方程，具有良好的非线性建模能力和泛化效果，但机器学习模型的效果很大程度上依赖特征工程，且不能全面深入地挖掘序列中不同位置数据在时间维度上的相关性；基于深度学习的时序预测方法通过复杂的网络结构挖掘序列中数据依赖性，但仍存在一些问题，如长序列信息无法传递、对高振幅高频率序列预测存在延迟等。

本文基于时序预测研究现状、相关理论及存在的问题，提出融合信号分解和偏差修正思想的基于 Transformer 的集成学习模型。首先利用 CEEMDAN 将原始时序数据分解成若干个振幅和频率互不相同的子序列，之后使用 Transformer 模型对各子序列进行训练和预测，得到中间预测结果。最后运用 SVR 对中间预测结果的偏差进行建模分析，将残差预测结果与中间预测结果融合，得最终预测结果。集成学习模型整体框架如图 3-1 所示。

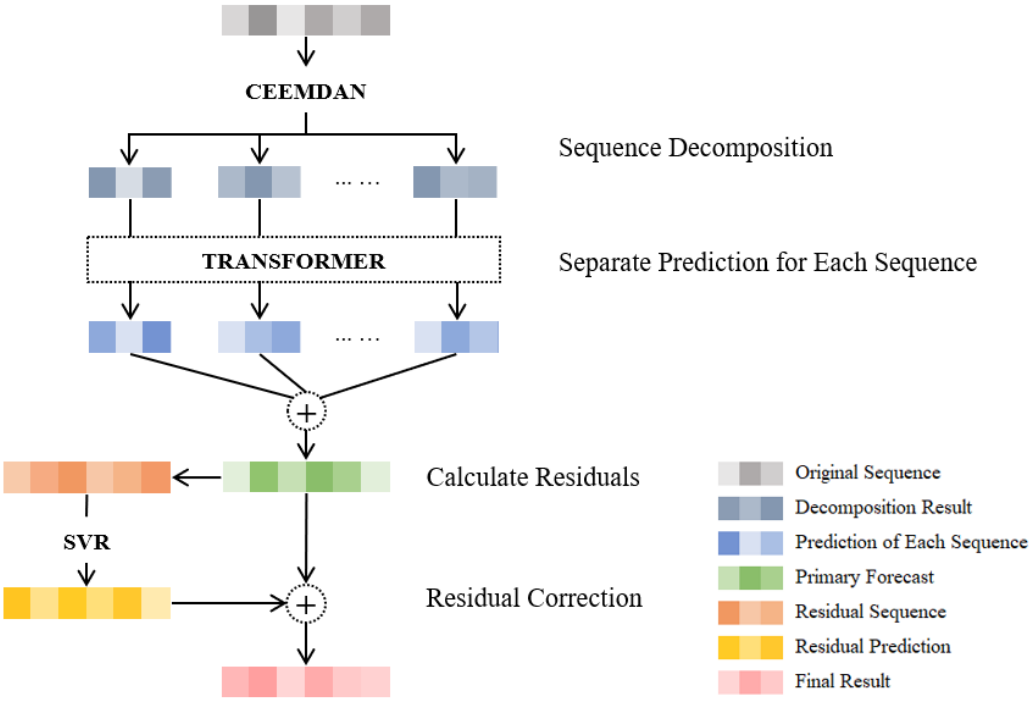

图 3-1 基于 Transformer 的集成学习模型框架

3.1 时序数据分解

现有时序预测模型，不论是传统统计学模型，还是机器学习模型，又或是深度学习模型，在时序预测时均存在不同方面、不同程度的问题，如预测延迟、高频度高幅度点预测偏差较大等。考虑到时序数据均由多个分量构成，单独预测各分量可更有针对性地提取序列中的信息，更全面深入的开展模型预测和训练。为此，本文考虑应用信号分解算法，在数据传入模型训练前对其进行分解，得到若干个不同幅度、不同频率的信号后，将各子序列单独建模分析。

CEEMDAN 分解通过添加一组自适应高斯白噪声改变原始信号的极值分布情况，能够克服模式混叠、噪声对序列分解子序列无法对齐等问题，可以取得更好的分解效果，因此其理论和应用价值更高。鉴于 CEEMDAN 的优秀性能，本文提出的集成学习模型采用 CEEMDAN 对原始数据进行分解。

3.2 Transformer 模型预测

传统的时序预测方法，如 ARIMA、GARCH 等，使用局限性大、模型表达能力有限，不能很好地处理非线性复杂时序数据；基于机器学习的时序预测方法，如 SVR、XGBoost 等，主要是建立函数方程，具有良好的非线性建模能力和泛化效果，但机器学习模型的效果很大程度上依赖特征工程，且不能全面深入地挖掘序列中不同位置数据在时间维度上的相关性；基于深度学习的时序预测方法，如 RNN、LSTM 等，通过复杂的网络结构挖掘序列中数据依赖性，但训练或预测中仍存在一些问题，如参数训练中的梯度消失或梯度爆炸、对高振幅高频率序列预测存在延迟等。基于上述分析，本文在集成学习中使用 Transformer 模型进行时序预测。

Transformer 模型自提出以来，被应用到众多领域，均取得了良好的效果。在时序预测中，Transformer 模型多头自注意机制可以有效提取时序数据中各元素间隐含的关系。不同于 RNN 模型的串行训练，由于 Transformer 模型训练通过矩阵计算进行，因此 Transformer 模型计算效率更高。考虑到时序数据不像 NLP 任务中的数据那么复杂，因此本文将 Transformer 模型中的 decoder 部分使用全连接层替换，避免模型过拟合而影响模型性能。改造后应用于时序预测的 Transformer 模型预测结构示意

图如图 3-2 所示，主要由嵌入层、encoder 层、decoder 层和全连接层构成。

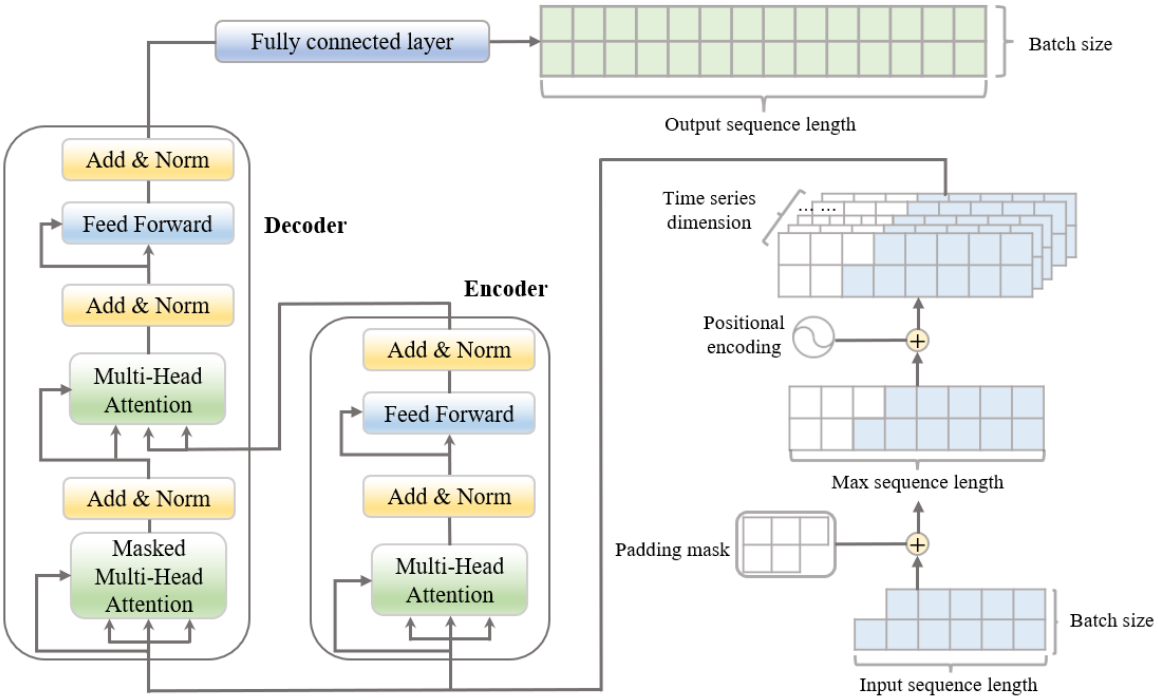

图 3-2　Transformer 模型时序预测结构

3.2.1　嵌入层

数据在传入 encoder 之前，需要先经过 embedding 编码，即将输入数据用固定维度的向量表示出来。这样做的不仅可以解决原数据维度过大引起的计算量陡增的问题，还可以简化模型，提高模型训练速度。考虑到时序数据间存在时间顺序关系，为了在数据中融入位置信息，需要在 embedding 后进行位置编码。

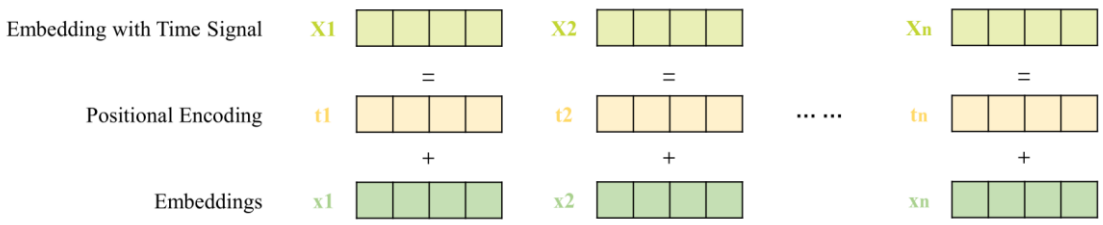

图 3-3　位置编码示意图

3.2.2 Encoder 层

单头自注意力机制不能全面的分析序列中的信息，因此本文 encoder 部分使用多头自注意力机制。多头注意力机制即将单头注意力机制重复多次，且每个子空间的参数矩阵互补相同，为的是可以多方面学习时序序列中蕴含的多维度信息。之后将多个自注意力输出的序列拼接，并利用适当维度的权重矩阵，使得拼接得到的矩阵的维度降低到与拼接前一致。

每个注意力机制计算中的首个 encoder 结构如图 3-4 所示(以两个输入为例)。

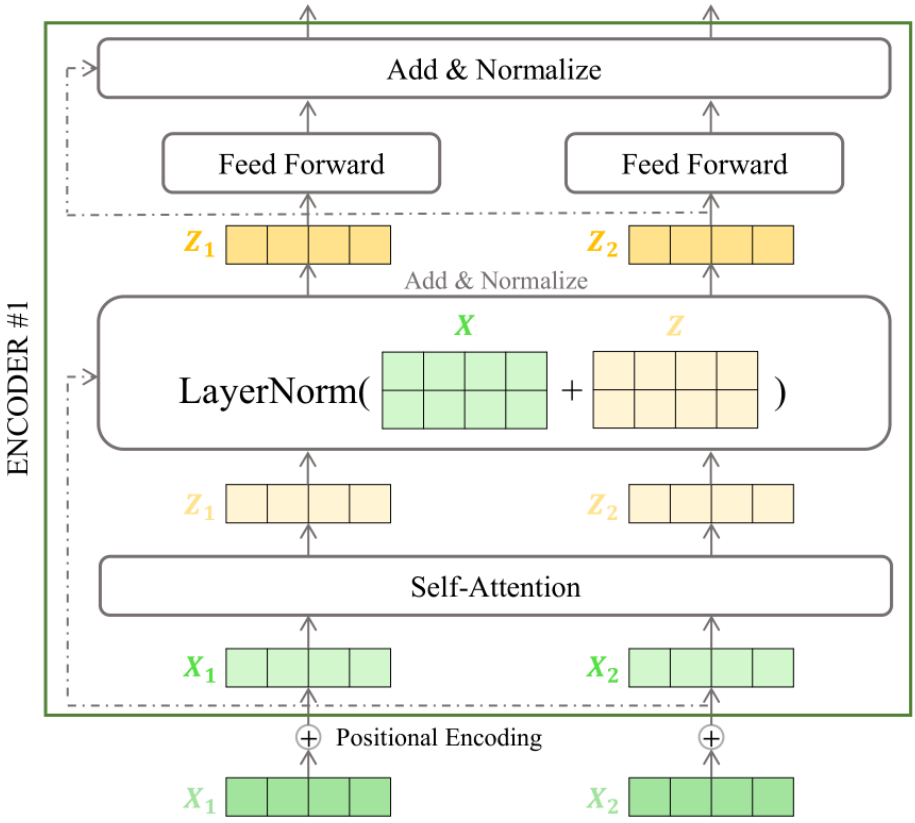

图 3-4　首个 encoder 结构示意图

每个 encoder 层有两个子层，第一个子层是多头自注意力机制，第二个子层是全连接前馈神经网络层。此外，考虑到模型训练中的问题，在子层之间添加残差连接与归一化模块。残差连接即在子层的输出中添加输入，以避免权重矩阵退化。而归一化就是将输出矩阵按列进行归一化，使隐藏变量服从标准正态分布，以加快模型训练。基于此，每个子层的输出即为 $\mathrm{LayerNorm}(x+\mathrm{Sublayer}(x))$. 其中，$\mathrm{Sublayer}(x)$ 是每个

子层的映射函数，LayerNorm(·)为归一化函数。最后，为了方便残差连接，模型中的所有子层和嵌入层的输出维度保持相同。

3.2.3 Decoder 层

decoder 层结构和 encoder 层相同，需要注意的是，对当前时间步做预测时，只能利用当前时间步之前的输入[46]。因此，在每个 decoder 中，首个 Attention 需要增加 mask，用于将矩阵 QK^T 的左上角元素置为-∞，此时，Attention 计算公式如(3-1)式。

$$\begin{aligned} Q_i = XW_i^Q, K_i = XW_i^K, V_i = XW_i^V, \\ \text{head}_i = \text{softmax}(\frac{Q_i K_i^T}{\sqrt{\tilde{d}_k}} \cdot \text{mask}) V_i, \\ \text{MultiHead}(Q,K,V) = \text{Concat}(\text{head}_1, \text{head}_2, ..., \text{head}_h) W^O, \end{aligned} \tag{3-1}$$

其中，$W_i^Q, W_i^K \in \mathbb{R}^{d_k \times \tilde{d}_k}, W_i^V \in \mathbb{R}^{d_v \times \tilde{d}_v}$ 为第 i 头的线性变换权重矩阵，W^O 为拼接后用于降维的线性变换权重矩阵。

3.2.4 全连接层

Transformer 模型中的序列编码和 encoder 层是将原始输入数据转换成隐藏特征，decoder 层是为了得到各预测值的得分，全连接层则是将得分转换成概率最后输出。全连接层结构如图 3-5 所示，其中，从输入层(Input layer)到隐藏层(Hidden layer)为线性计算，从隐藏层到输出层(Output layer)为非线性计算。

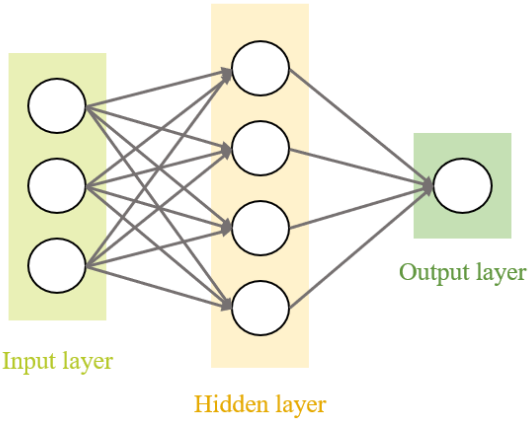

图 3-5　全连接层示意图

对于输入变量 $X^{(i)}=(x_1^{(i)},x_2^{(i)},\cdots,x_n^{(i)})$，全连接层计算公式如(3-2)式。

$$y^{(i)}=\text{sigmoid}(W^{\mathrm{T}}X^{(i)}+b), \tag{3-2}$$

其中，$W\in\mathbb{R}^n, b\in\mathbb{R}$ 为模型学习的参数。

3.3 偏差修正

模型预测结果的偏差除了可以用于反应模型的预测精度，还蕴含着一定的信息量，比如某些模型不能很好地处理序列中振幅过大或过小，又或是某些模型预测存在一定的滞后性等。因此对偏差序列建模分析后，将偏差序列的预测结果与 Transformer 模型的预测结果相融合可有针对性地解决 Transformer 模型的预测偏差。

为避免偏差修正导致集成学习模型过拟合，对于偏差序列的建模分析，不宜选择精度过高的模型。本文利用非线性 SVR 对偏差序列进行建模分析，将 SVR 预测的偏差序列与 Transformer 模型预测结果融合，得到集成学习模型最终的预测结果。偏差修正过程公式如(3-3)式。

$$\begin{aligned}\text{error}&=\text{truth}-\text{Transformer\_prediction},\\ \text{final\_prediction}&=\text{Transformer\_prediction}+\text{SVR(error)},\end{aligned} \tag{3-3}$$

其中，truth 为序列真实值，Transformer_prediction 为 Transformer 模型预测值，error 为 Transformer 模型预测偏差，SVR(error) 为 SVR 模型对偏差偏差序列的预测值，final_prediction 为模型最终预测结果。

3.4 本章小结

本章在时序预测研究现状、相关理论以及目前研究中存在的问题的基础上，提出基于 Transformer 的集成学习模型，集成学习模型主要包括 CEEMDAN 时序数据分解、Transformer 模型预测(序列编码、encoder 层、全连接层)和 SVR 偏差二次修正，其中重点介绍 Transformer 的实施细节，为接下来第四章的实证分析做理论准备。

4 时序预测实证分析

本章将前一章提出的基于 Transformer 的集成学习模型应用到时序预测中的数个经典数据集，包括 ETT、Electricity、Exchange_rate、Traffic、Weather、ILI 和 Temperature。为验证模型的有效性，本章利用 MSE 和 MAE 作为评估指标，选取时序预测中的经典统计学模型 ARIMA、机器学习模型 SVR 和深度学习模型 LSTM 与本文提出的基于 Transformer 的集成学习模型进行对比分析。

4.1 实验环境与配置

本文实验所用环境如下：操作系统为 Windows 10 家庭中文版(64 位)，处理器为 AMD Ryzen 5 4600H with Radeon Graphics 3.00 GHz，机带 RAM 为 16.0 GB，程序运行环境为 Python 3.9 以及 Python 3.8 (TensorFlow)，程序编辑器为 PyCharm Community Edition 2020.2.3 和 Jupyter Notebook (Anaconda3)。实证分析中的数据预处理、模型结果分析和可视化使用 Jupyter，模型训练使用 PyCharm。具体模型层面，ARIMA、SVR 和 CEEMDAN 使用 Python 3.9，LSTM 和 Transformer 分别使用 TensorFlow 和 torch 框架实现。深度学习模型训练超参数的含义如表 4-1 所示。

表 4-1 深度学习模型超参数含义

参数	含义
epoch	所有训练样本完成一次训练的过程
batch size	每次训练样本的个数
dropout ratio	节点不工作(输出设置为零)的概率
learning rate	每一次更新参数利用多少误差
learning rate decay step	学习率衰减步数，取值为 n 时表示每训练 n 个 epoch，将学习率降低一个数量级

4.2 评估指标

为比较不同模型的预测效果，本文采用均方误差(Mean Square Error, MSE)和平均绝对误差(Mean Absolute Error, MAE)作为模型效果的评价指标。

MSE 计算预测偏差平方和的均值，计算公式如(4-1)式。

$$\mathrm{MSE}=\frac{1}{N}\sum_{t=1}^{N}\left(y_t-\hat{y}_t\right)^2, \tag{4-1}$$

其中，y_t 和 $\hat{y}_t$ 分别为训练集的真实值和模型的预测值，N 为测试集的样本数。

MAE 计算预测值的平均绝对偏差，计算公式如(4-2)式。

$$\mathrm{MAE}=\frac{1}{N}\sum_{t=1}^{N}\left|y_t-\hat{y}_t\right|, \tag{4-2}$$

其中，参数含义与(4-1)式相同。

4.3 数据来源与预处理

本文实证分析部分选用时序相关研究中常用的六个数据集，覆盖气象、交通、金融、工业、医疗领域，各数据集信息如下：

1. ETT 数据集[1]：包含从电力变压器收集的数据，包括 2016 年 7 月至 2018 年 7 月期间每 15 分钟(ETTm1、ETTm2)/每小时(ETTh1、ETTh2)的负载和油温；
2. electricity 数据集[2]：包含 2012 年至 2014 年 321 名客户的每小时用电量；
3. exchange_rate 数据集[3]：1990 年至 2016 年八个国家货币的每日汇率；
4. traffic 数据集[4]：2016 年 7 月至 2018 年 7 月美国旧金山高速公路占用率；
5. weather 数据集[5]：马普生物地球化学研究所收集，记录 2020 年全年 21 个气象指标的数据，本文使用其中的温度数据，即 temperature 数据集；

[1] https://github.com/zhouhaoyi/ETDataset

[2] https://archive.ics.uci.edu/ml/datasets/ElectricityLoadDiagrams20112014

[3] https://github.com/laiguokun/multivariate-time-series-data

[4] http://pems.dot.ca.gov/

[5] https://www.bgc-jena.mpg.de/wetter/

6. illness 数据集[6]：包括美国疾病控制和预防中心于 2002 年至 2021 年间每周记录的流感类疾病(ILI)患者数据。

表 4-2　各数据集主要统计指标

数据集	长度	最大值	最小值	极差	均值	标准差	变异系数	偏度	峰度
temperature	3,650	26.30	0.00	26.30	11.18	4.07	2.75	0.17	-0.06
exchange_rate	7,588	0.88	0.39	0.49	0.65	0.12	5.68	-0.21	-0.69
electricity	26,304	6,035.00	0.00	6,035.00	3,335.88	552.74	6.04	0.77	0.77
illness	966	1,640,587	64,699	1,575,888	651,497.46	348,838.19	1.87	0.62	0.16
traffic	17,544	0.22	0.00	0.22	0.03	0.02	1.62	0.19	-0.24
ETTh1	17,420	46.01	-4.08	50.09	13.32	8.57	1.56	0.97	0.78
ETTh2	17,420	58.88	-2.65	61.52	26.61	11.89	2.24	0.10	-0.81
ETTm1	69,680	46.01	-4.22	50.23	13.32	8.56	1.56	0.97	0.77
ETTm2	69,680	58.88	-2.65	61.52	26.61	11.89	2.24	0.10	-0.81

从上表可以看出，某些数据集存在异常值。为保证模型效果，在数据传入模型训练之前，需要对数据进行一系列预处理，具体流程如图 4-1 所示。

图 4-1　数据预处理流程

异常值剔除：各数据集存在不同程度的异常值，本文采用基于滚动统计的方法剔除异常值。这种方法中，上限和下限是根据特定的统计量度创建的，本文采用 3σ 原则设定窗口的上限和下限，即 $[x_i-3\sigma_i, x_i+3\sigma_i]$，其中 σ_i 为窗口内数据的标准差。

缺失值填充：各数据集原始的缺失值和剔除后的异常值需要填充，考虑到时序数据的特性，本文采用滑窗的方法，取缺失值前窗口内数据的均值来填充缺失值。

去噪：时序数据中的噪声元素可能会导致预测模型偏差，所以一般情况下在构建任何模型之前都会有去除噪声的操作，时序数据最小化噪声的过程称为去噪。本文采用滚动平均值的方式去噪，即根据数据集的实际含义，固定窗口宽度后计算每个窗口的均值，新得到的数据作为去噪后的序列数据。

标准化：为消除数据集的量纲，这里采用如(4-3)式对数据进行标准化。

[6] https://gis.cdc.gov/grasp/fluview/fluportaldashboard.html

$$x_i = \frac{x_i - E(X)}{\sqrt{Var(X)}}, E(X) = \frac{1}{N}\sum_{i=1}^{N} x_i, Var(X) = \frac{1}{N-1}\sum_{i=1}^{N}(x_i - E(X))^2, \tag{4-3}$$

其中，$E(X)$和$Var(X)$分别为数据序列的均值和方差。

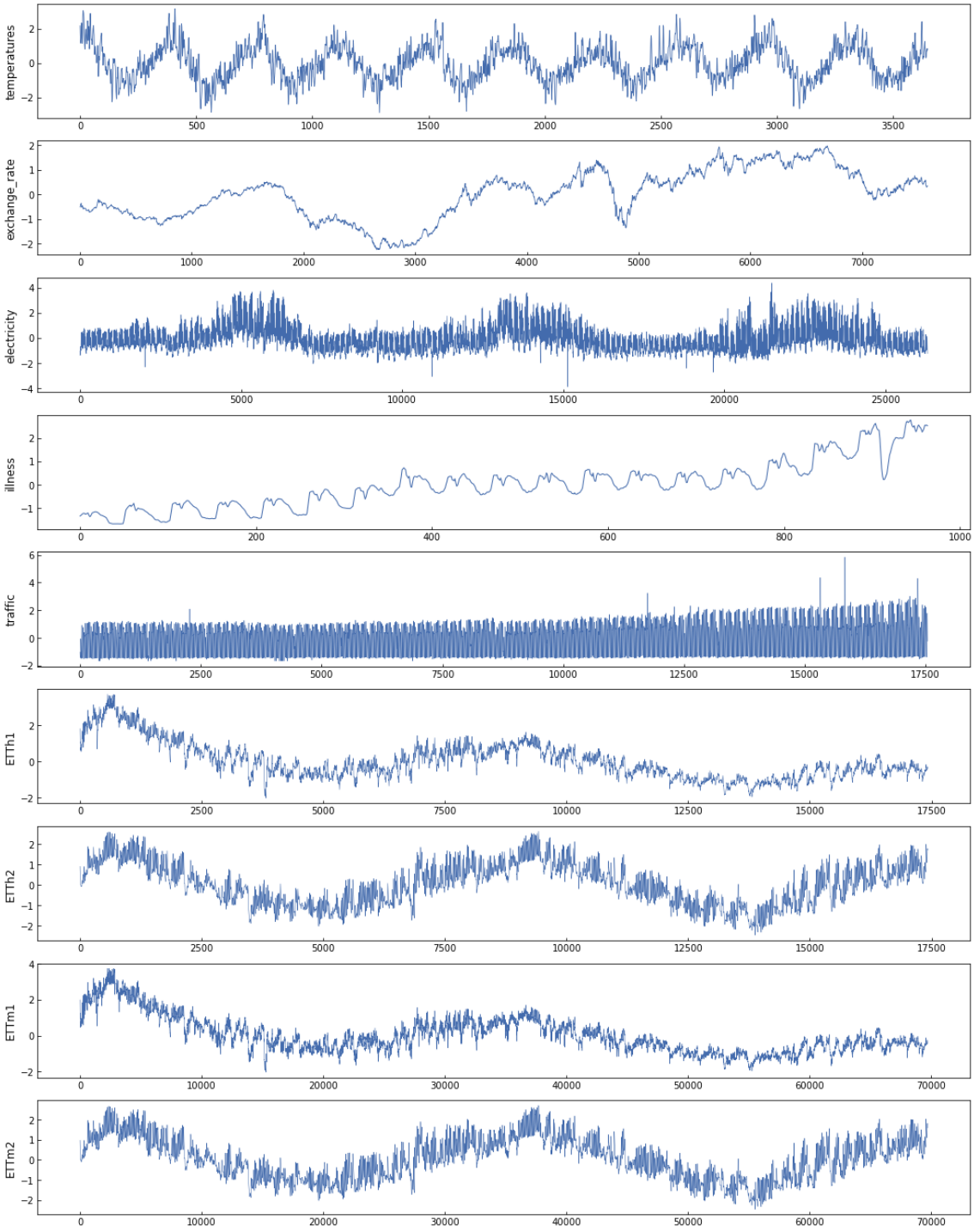

图 4-2　预处理后各数据集可视化

经过预处理后，数据集的统计指标如表 4-3 所示，预处理后各数据集可视化如图 4-2 所示，结合表 4-3 和图 4-2 可以看出，数据集分布情况不尽相同，具体表现为序列长度从 964 到 69,678、波动周期差异较大、波动幅度不同。实证分析中这样丰富的数据集更能验证模型效果。

表 4-3　预处理后各数据集的统计指标

数据集	长度	最大值	最小值	极差	均值	标准差	变异系数	偏度	峰度
temperature	3,648	3.15	-2.88	6.03	0.00	1.00	0.00	0.15	-0.29
exchange_rate	7,586	1.97	-2.26	4.23	0.00	1.00	0.00	-0.22	-0.69
electricity	26,302	4.37	-3.88	8.25	0.00	1.00	0.00	0.80	0.56
illness	964	2.77	-1.68	4.46	0.00	1.00	0.00	0.61	0.15
traffic	17,542	5.85	-1.68	7.52	0.00	1.00	0.00	0.13	-0.77
ETTh1	17,418	3.71	-2.03	5.74	0.00	1.00	0.00	0.97	0.78
ETTh2	17,418	2.64	-2.46	5.09	0.00	1.00	0.00	0.10	-0.82
ETTm1	69,678	3.75	-2.04	5.79	0.00	1.00	0.00	0.97	0.77
ETTm2	69,678	2.70	-2.46	5.17	0.00	1.00	0.00	0.10	-0.81

从图 4-2 各数据集可视化可以看出，ETT 数据集中的四个数据集大概形态几乎一致，其中 ETTh1 和 ETTm1 更为相近、ETTh2 和 ETTm2 更为相近。由于数据量过大而个人电脑计算力有限，因此对于 ETT 数据集，本文选用 ETTh1 进行建模分析。

4.4 单模型预测效果

本节选用 ARIMA、SVR 和 LSTM 进行训练预测，由于篇幅问题，这里只展示 temperature 数据集的建模分析过程，其余数据集的建模结果放在最后做总结分析。

4.4.1 ARIMA

本节以 temperature 数据集为例，按照 ARIMA 模型建模步骤，介绍建模过程中的几个重要步骤。

1. 数据平稳性检验

由于原始时序数据非平稳，对原始数据进行差分后进行平稳性检验。一阶差分后序列平稳性检验的 $\text{p-value} = 3.044 * 10^{-26} < 0.05$，$t$ 检验统计量为-14.062，小于检验统计值 5%的临界值-2.87，因此可判定序列是平稳的。

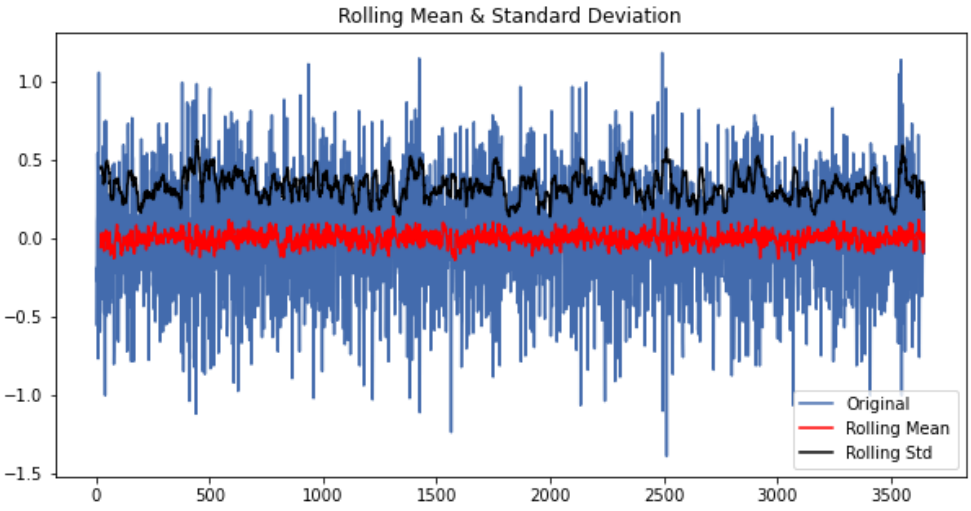

图 4-3　序列一阶差分结果

2. 自相关图与偏自相关图

一阶差分序列的自相关系数和偏自相关系数如图 4-4 所示，从图中可以看出，差分序列的自相关系数和偏自相关系数已经呈现拖尾性，因此可以开始对数据进行模型定阶。

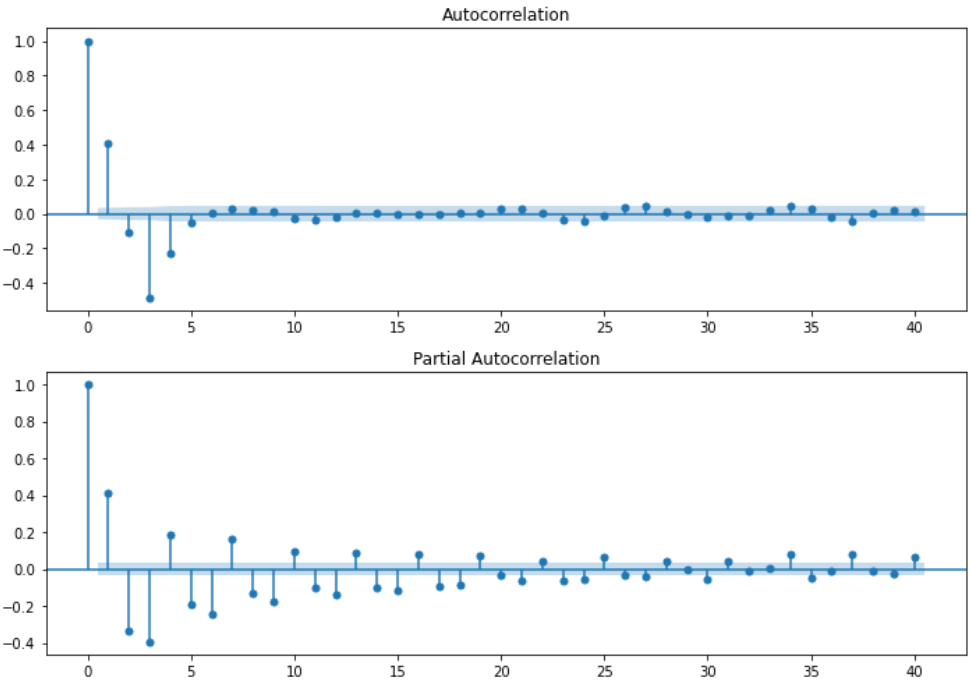

图 4-4　一阶差分序列自相关系数与偏自相关系数

3. 模型定阶

根据自相关和偏自相关系数图，初步确定使用ARIMA$(p,1,q)$模型进行预测。由于难以直接确定p和q的值，本文采用循环遍历的方法，限制在0~10的范围内且不同时为0，对所有可能阶数的ARIMA模型进行拟合，并计算各模型的AIC、BIC指数，结果如表4-4。由于篇幅的关系，下图仅展示AIC和BIC较小的前30个组合。

表4-4 p和q不同组合下AIRMA模型评价指标取值情况

p	q	AIC	BIC	p	q	AIC	BIC
2	7	**-747.95**	**-679.74**	1	9	-737.51	-663.09
2	5	**-742.59**	**-686.77**	2	3	-726.28	**-682.87**
1	6	-739.08	**-683.26**	7	4	-739.48	-658.86
1	7	**-740.11**	**-678.10**	0	5	-725.42	**-682.01**
4	8	**-748.80**	-661.97	8	4	**-740.31**	-653.49
5	5	**-743.34**	-668.92	2	10	-739.30	-652.48
5	4	**-740.52**	**-672.30**	5	6	-736.70	-656.08
2	8	**-742.69**	-668.27	1	4	-723.29	**-679.88**
4	6	**-742.45**	-668.03	1	10	-735.67	-655.05
2	9	**-744.41**	-663.79	2	4	-724.30	**-674.69**
1	8	-738.42	**-670.20**	3	3	-724.30	**-674.68**
4	5	-737.97	-669.75	0	6	-723.59	**-673.98**
6	4	**-740.05**	-665.63	1	5	-723.56	**-673.95**
4	7	**-741.33**	-660.71	4	9	-738.95	-645.92
5	7	**-743.24**	-656.41	1	1	-736.70	-649.87

如表4-4所示，$p=2,q=7$、$p=2,q=5$等几个模型的三个评价指标相对大小差距不大，由于$p=2,q=7$的AIC和BIC较小，因此定阶选用ARIMA(2,1,7)。

4. 残差分析

衡量ARIMA模型建模的效果主要通过分析残差序列，若残差序列为无自相关和偏自相关的白噪声序列，则表示模型效果较优。本文采用三个方式分析残差序列来评判ARIMA(2,1,7)模型。

一是残差序列的QQ图。从图4-5可以看出，ARIMA(2,1,7)模型残差序列的QQ图近似呈现出一条直线，因此可以粗略地认为残差序列服从正态分布。

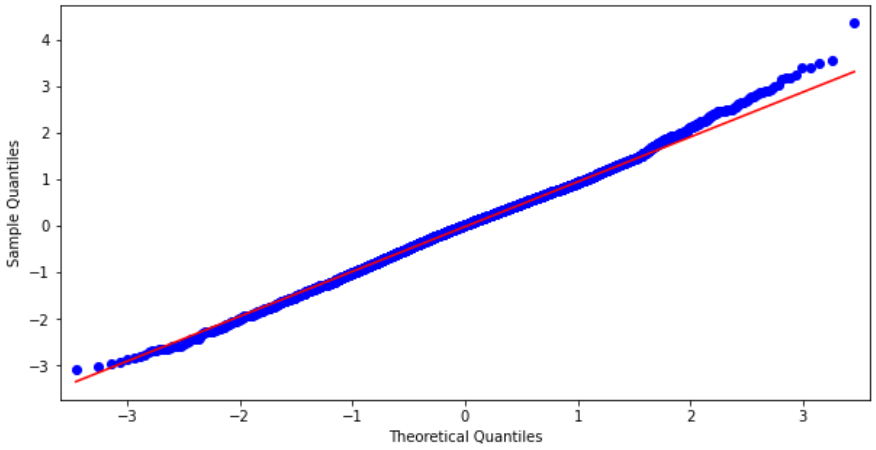

图 4-5　ARIMA 模型残差序列 QQ 图

二是 Durbin-Watson 检验。Durbin-Watson 检验值(记为 d 值)显著的接近于 0 或 4 时，则存在序列自相关性，而 d 值接近于 2 时，则不存在一阶自相关性。ARIMA(2,1,7) 模型残差序列计算得到的 d 值为 1.997，因此可以认为 ARIMA(2,1,7) 模型残差序列不存在一阶自相关。

三是 Ljung-Box 检验，即白噪声检验。Ljung-Box 检验的原理是检验序列各阶差分间的相关性(通常为 12 阶以内)，若检验概率小于给定的显著性水平(如 0.05、0.10)，则拒绝原假设，即不认为相关系数为零。表 4-5 为 ARIMA(2,1,7) 模型预测的残差序列的 Ljung-Box 检验结果。

表 4-5　Ljung-Box 检验结果

lag	AC	Q	Prob(>Q)
0	0.001	0.001	0.972
1	-0.001	0.007	0.996
2	0.003	0.031	0.999
3	-0.002	0.047	1.000
4	-0.001	0.050	1.000
5	-0.006	0.202	1.000
6	0.010	0.546	0.999
7	-0.007	0.732	0.999
8	0.008	0.944	1.000
9	-0.018	2.105	0.995
10	-0.029	5.125	0.925
11	-0.011	5.593	0.935
12	0.005	5.696	0.957

从表 4-5 可以看出，残差序列各阶 lag 白噪声检验的 P 值均大于 0.05，即不存在明显的自相关和偏自相关。

ARIMA(2,1,7)模型预测结果如图 4-6 所示，MSE、MAE 分别为 0.045、0.162。图 4-6 可以看出，ARIMA 模型预测结果较为理想，但预测存在一定程度的延迟性，尤其在高频时间段。

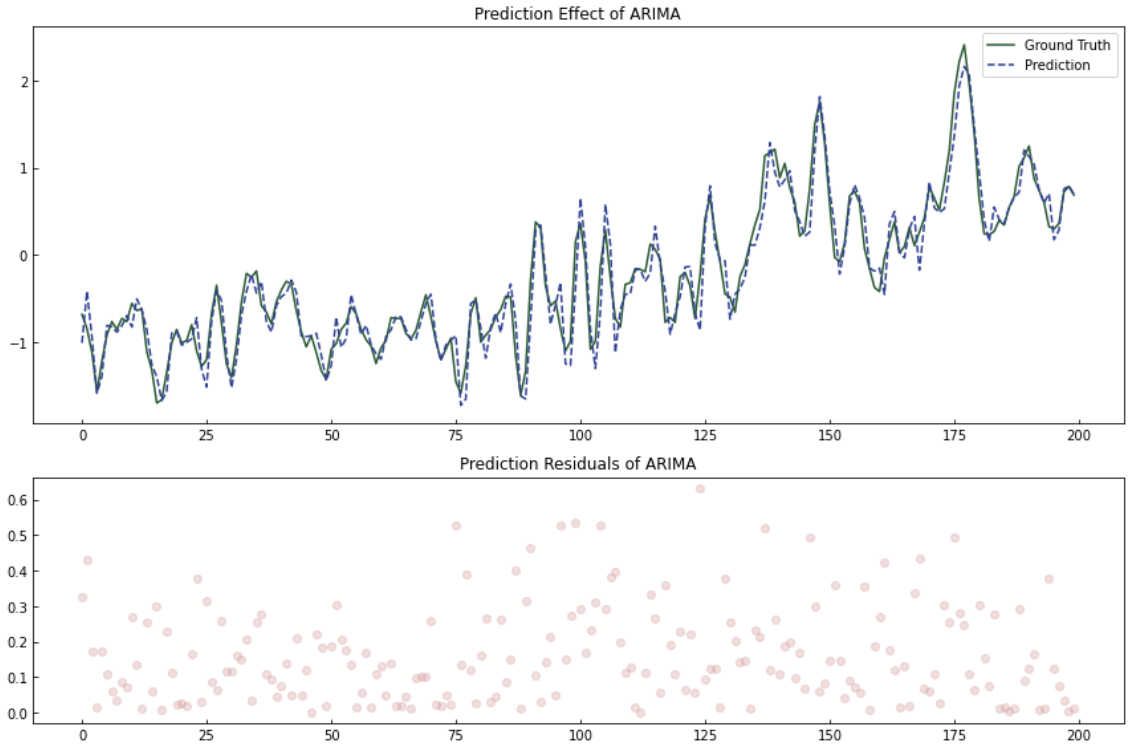

图 4-6　ARIMA 模型预测效果

4.4.2　SVR

本节根据第 2 章中介绍的 SVR 相关理论，使用 SVR 模型对 tenperature 数据集进行训练。训练集、验证集和测试集的样本量比例为 6:2:2，且考虑到 temperature 数据集样本数量不多(3,650 个)，本文采用十折交叉验证进行模型训练。设置输入变量个数为 20，即使用前 20 个时刻的数据来预测第 21 时刻的数据。超参数的选取采用网格寻优，其中，核函数 kernel 选用 RBF 径向基函数、参数 gamma 的取值范围通过调用 numpy.logspace(-5, 0, num=6, base=2.0)生成等比数据组[0.03125, 0.0625, 0.125, 0.25, 0.5, 1]，采用同样的方法生成参数 C 的取值范围为[0.03125, 0.0625, 0.125, 0.25,

0.5, 1, 2, 4, 8, 16, 32]，通过十折交叉验证的方法计算得最优超参数组合为{'C': 8.0, 'gamma': 0.03125, 'kernel': 'rbf'}。SVR 模型预测效果如下图 4-7 所示，SVR 对测试集预测结果的 MSE、MAE 分别为 0.026、0.125。

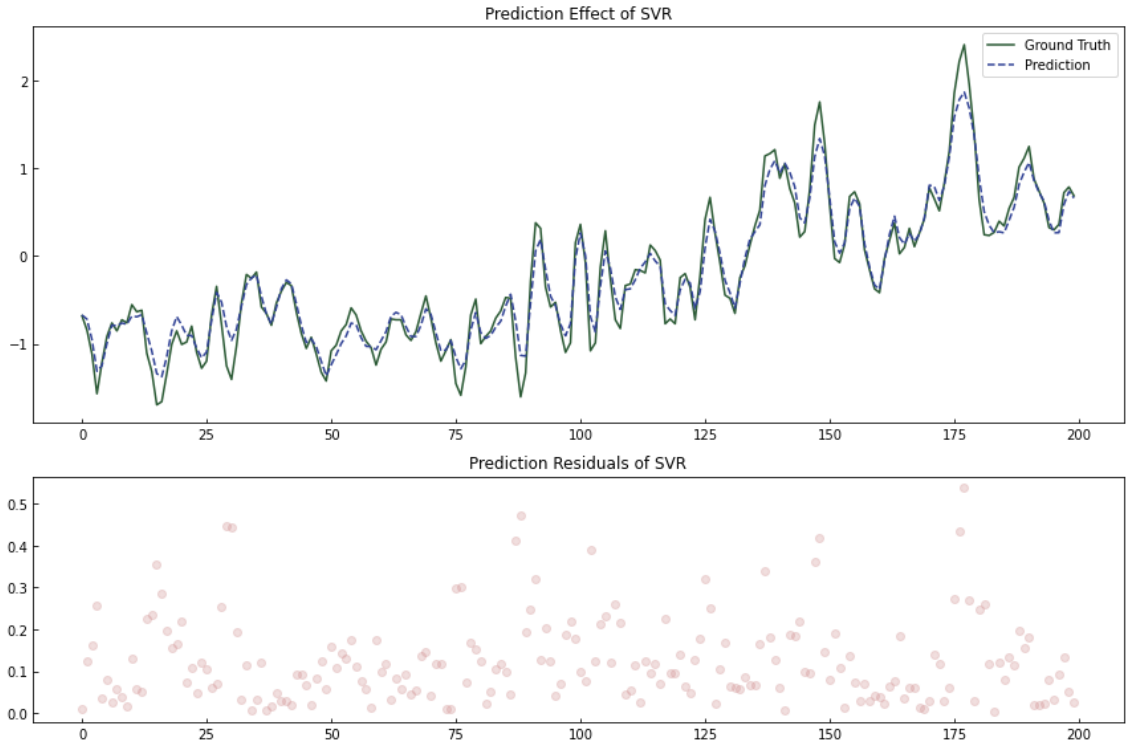

图 4-7　SVR 模型测试集预测效果

对比 ARIMA 模型，SVR 模型的表现更优，MSE 降低了 73.1%，MAE 降低了 29.6%。但需要注意的是，SVR 属于机器学习模型，其参数寻优的方式方法确定了模型的最佳超参数，且由于样本有限使用的交叉验证确定了模型理想的参数，这都使得 SVR 模型在测试集上的表现很优秀。

4.4.3　LSTM

本节根据第 2 章中介绍的 LSTM 相关模型和算法，使用 LSTM 模型对 temperature 数据集进行训练。为保证模型对比有效，LSTM 模型训练中的样本切分比例和变量个数与 SVR 模型保持一致。batch size 和 epoch 分别设置为 64 和 20，激活函数选用 Relu，学习率设置为 0.001。LSTM 模型训练中，验证集和测试集的损失函数值如图 4-8 所示。

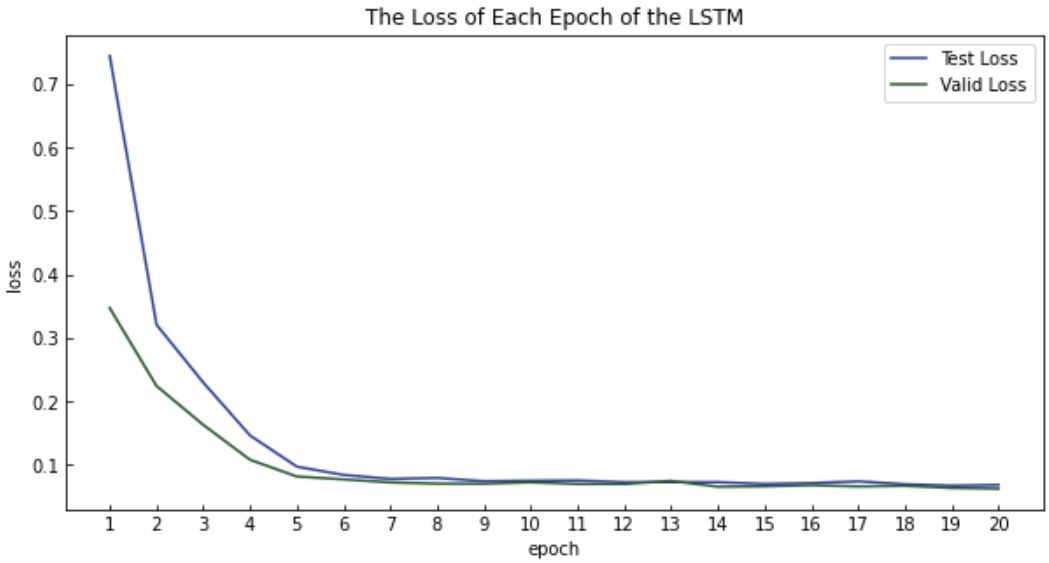

图 4-8　LSTM 模型训练各 epoch 损失函数值

图 4-8 可以看出，在 LSTM 模型训练的过程的每个 epoch 中，测试集和训练集的损失函数均迅速下降，约第 7 个 epoch 之后，损失趋于稳定值。LSTM 模型训练效果如下图 4-8 所示，模型对测试集预测结果的 MSE、MAE 分别为 0.063、0.197。

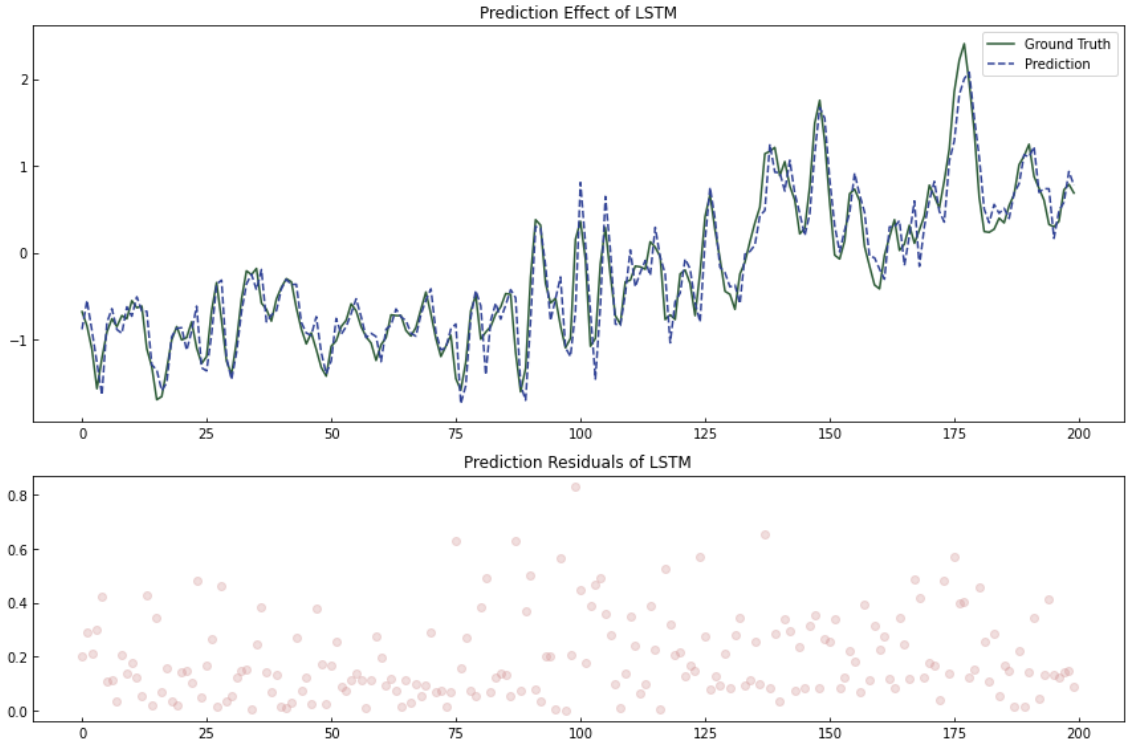

图 4-9　LSTM 模型测试集预测效果

对于相同的测试集和样本划分比例，LSMT 模型没有 SVR 模型效果好。二者除了模型结构不相同，其模型训练机制也不相同。另外，由于深度学习模型参数较多，需要大量的训练数据才能使得模型表现出理想效果，而本实例使用的 temperature 数据集仅有 3,650 个数据，这也是 LSTM 模型的效果没有 SVR 模型好的原因。

4.4.4 Transformer

本节根据第 2 章中介绍的 Transformer 相关模型和算法，使用 Transformer 模型对 temperature 数据集进行训练。

为保证模型对比有效，Transformer 模型训练的样本切分比例和参数配置与 LSTM 模型相同。模型训练分为 20 个 epoch，其中前 12 个 epoch 训练后测试集的预测情况如下图 4-10 所示。

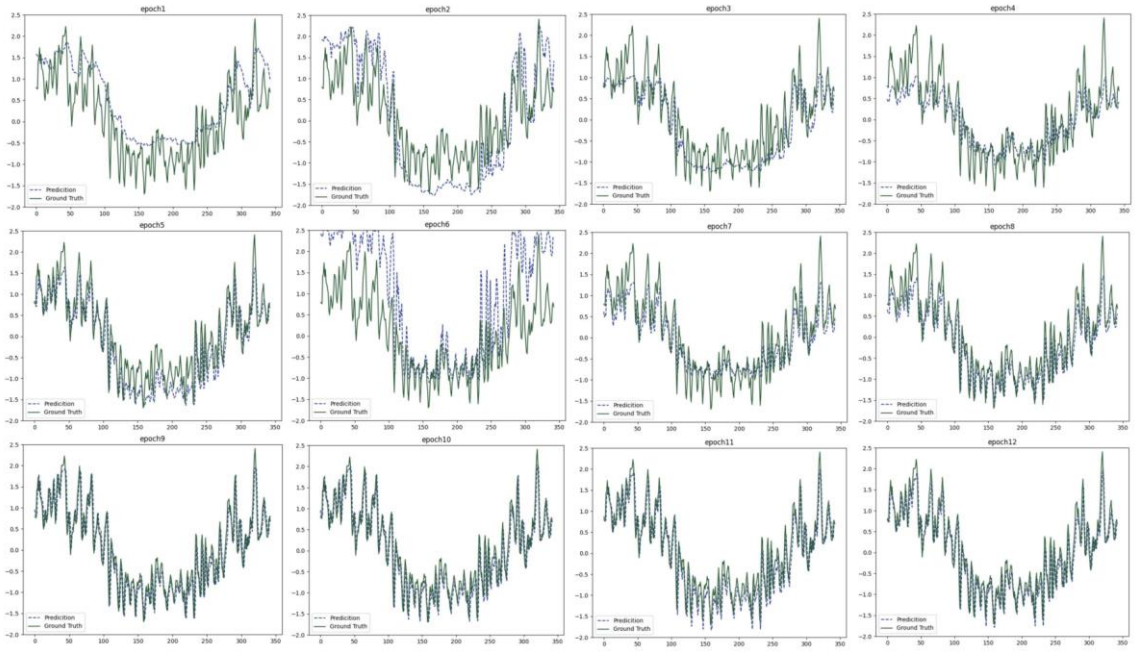

图 4-10　Transformer 模型前 12 个 epoch 训练后测试集预测效果

上图可以看出，模型前几个 epoch 训练后测试集的预测效果并不佳，但随着 epoch 的增加，模型学习到了更多的信息，训练后的模型在测试集上的预测效果逐步提高，epoch9 开始，模型已经可以较为稳定地、准确地预测测试集。经过 20 个 epoch 的训练，最终模型在测试集上的预测效果如图 4-11 所示。Transformer 模型对测试集预测结果的 MSE、MAE 分别为 0.051、0.170。

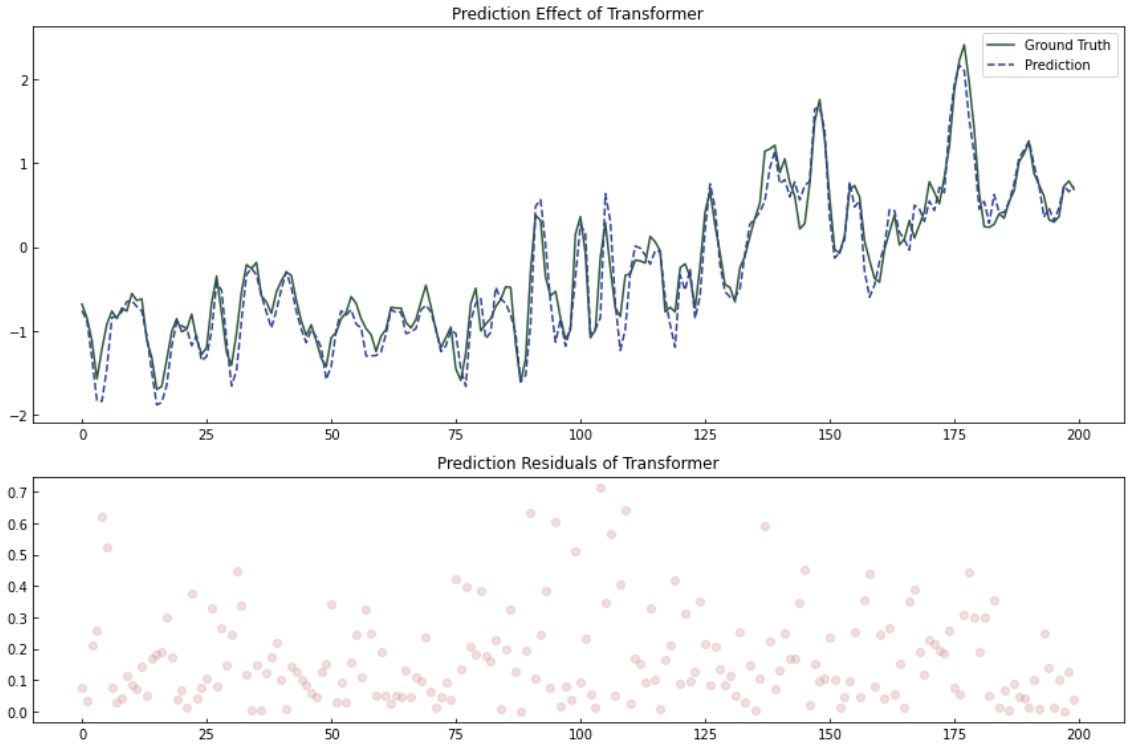

图 4-11　Transformer 模型测试集预测效果

对比 LSTM 模型，Transformer 模型在测试集上的 MSE 降低了 23.5%，MAE 降低了 15.9%，说明 Transformer 模型效果优于 LSTM 模型，但其效果仍然没有 SVR 模型好，原因与 LSTM 模型一致。temperature 数据集样本过少，导致训练得到的 Transformer 模型精度不高。

4.5　基于 Transformer 的集成学习预测效果

根据第 3 章中阐述的基于 Transformer 的集成学习模型构建，本文提出的模型的建模主要分为三步，分别为时序数据分解、Transformer 模型预测、SVR 偏差修正。本节按照上述步骤开展时序预测，并将每个步骤的建模结果可视化。

图 4-12　基于 Transformer 的集成学习模型流程

4.5.1 时序数据分解

利用 CEEMDAN 算法，对 temperature 数据集进行分解，得到 9 个 IMF 分量和一个残差分量共计 10 个子序列，分解结果如下图 4-13 所示。

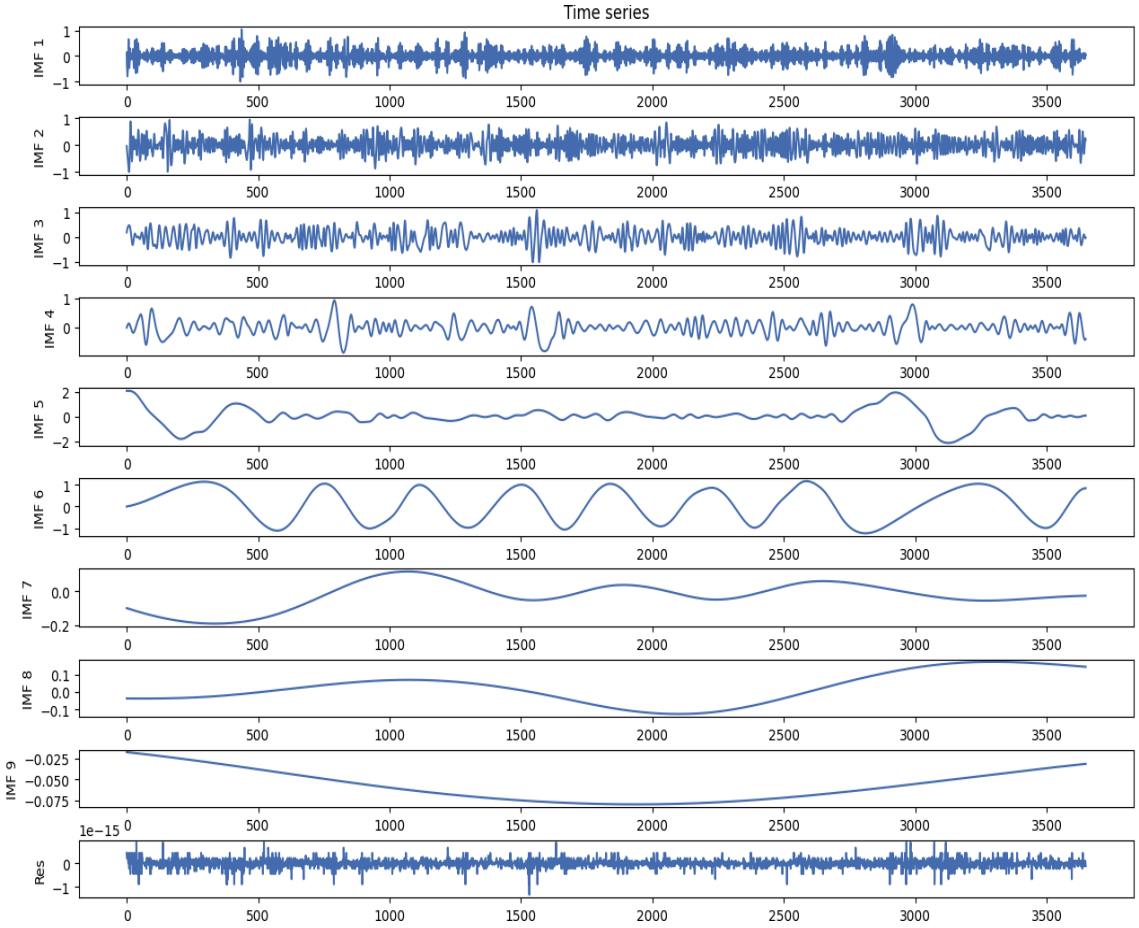

图 4-13 CEEMDAN 分解结果

图 4-13 可以看出，分解出的子序列可大致分为高频(IMF1~IMF2)、中频(IMF3~IMF4)、低频(IMF5~IMF9)和频率较大的残差 Res。接下来依次对各子序列单独建模。

4.5.2 Transformer 模型预测

使用 Transformer 模型对 CEEMDAN 分解得到的子序列单独建模。为保证模型效果可比性，此处模型参数与配置和 Transformer 单模型保持一致。Transformer 模型对各子序列预测结果如图 4-14 所示，图中实线为真实值，虚线为预测值。

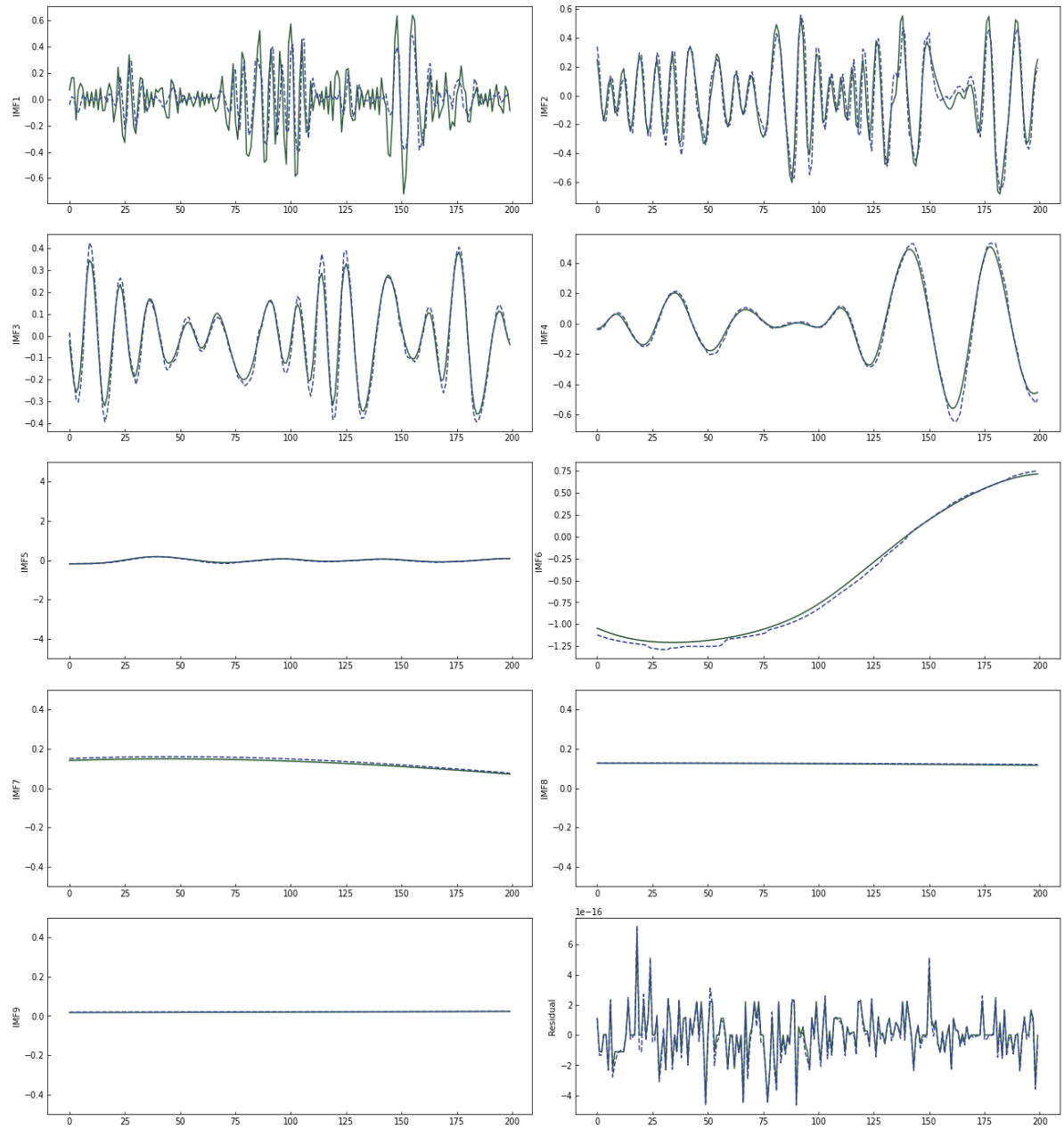

图 4-14　CEEMDAN 分解各子序列 Transformer 预测结果

可以看出 Transformer 模型各子序列的建模效果都很好，且其中对低频子序列的预测效果优于高频子序列。

将 10 个分量的预测结果相加得到原序列的中间预测结果，测试集预测结果 MSE、MAE 分别为 0.037、0.142。预测效果如图 4-15。

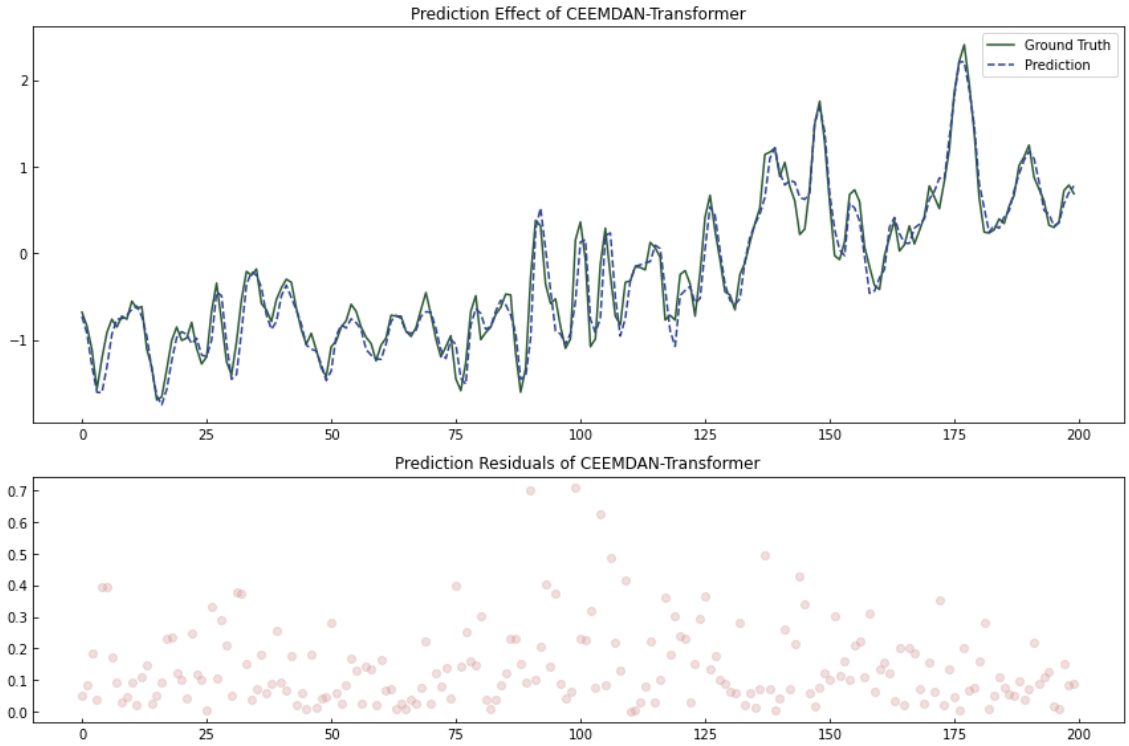

图 4-15　CEEMDAN-Transformer 预测结果

经过 CEEMDAN 分解和 Transformer 模型预测的效果好于单独使用 Transformer 模型预测，MSE 和 MAE 分别降低了 37.8%和 19.7%，证了 CEEMDAN 分解后 Transformer 预测的合理性。

图 4-15 中预测结果和偏差散点图可以看出，模型在部分区间内预测存在一定的滞后性，此时使用模型单独预测偏差序列，进行偏差修正，可以提高模型预测精度。

4.5.3　SVR 偏差修正

得到原序列的预测结果后，计算测试集预测结果的偏差序列。之后运用 SVR 进行建模。网格搜索确定最优参数组合为{'C': 32.0, 'gamma': 0.00390625, 'kernel': 'rbf'}。

残差序列的 SVR 预测结果如图 4-16 所示。偏差序列预测结果图可以看出，SVR 模型可在一定程度上预测出数据，但并相对并不准确。分析可以得知，偏差序列不同于原序列，所含信息有限，因此预测结果属于正常。另一方面，只要偏差预测值介于偏差真实值和 0 之间，即可在融合时降低偏差，图 4-15 可以看出，除了标记的少数区间外，偏差的预测值均可以实现降低集成学习模型预测偏差的目标。

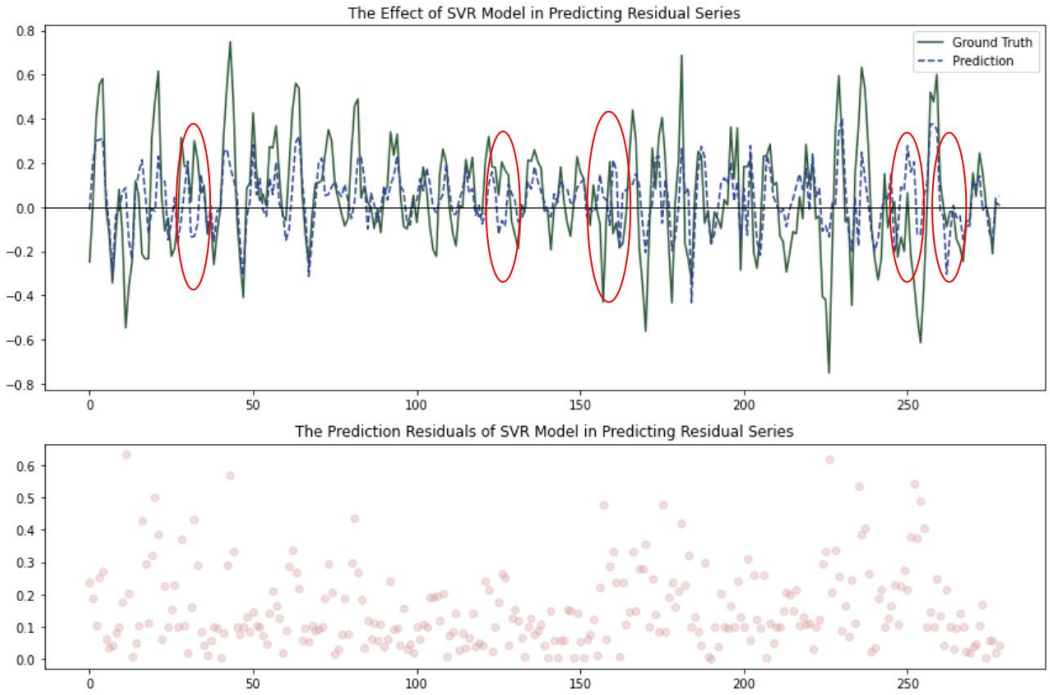

图 4-16 偏差序列 SVR 预测结果

将偏差预测结果与原序列预测结果融合到一起，得到集成学习模型最终预测结果，如下图 4-17 所示。测试集预测结果 MSE、MAE 分别为 0.017、0.112。

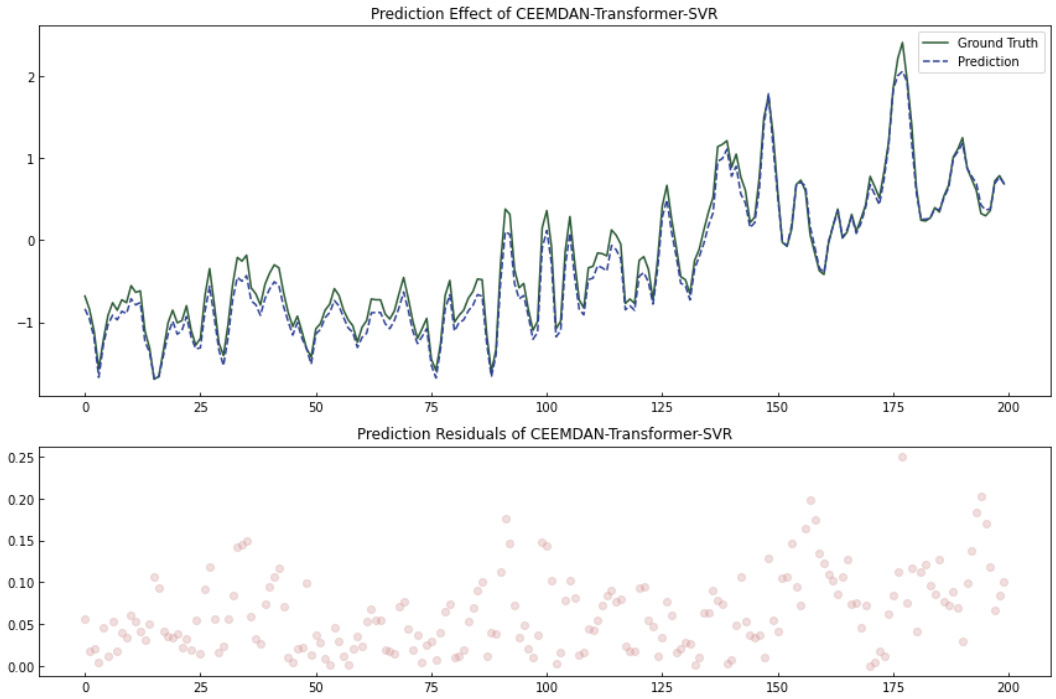

图 4-17 集成学习模型预测效果

偏差二次修正后模型在测试集上的 MAE 和 MSE 分别降低 117.6%和 12.8%，验证了偏差二次修正的效果。此外，从图 4-17 下半部分图的纵轴可以观察到，残差绝对值在 0.25 以内，相对于图 4-11 和图 4-15 中偏差绝对值最大约为 0.7，残差大幅度降低。

4.6 实验结果对比分析

为保证可视化效果，这里预测残差散点图绘制 200 个样本，但测试集预测值只绘制最后 50 个数据，如图 4-18 所示。

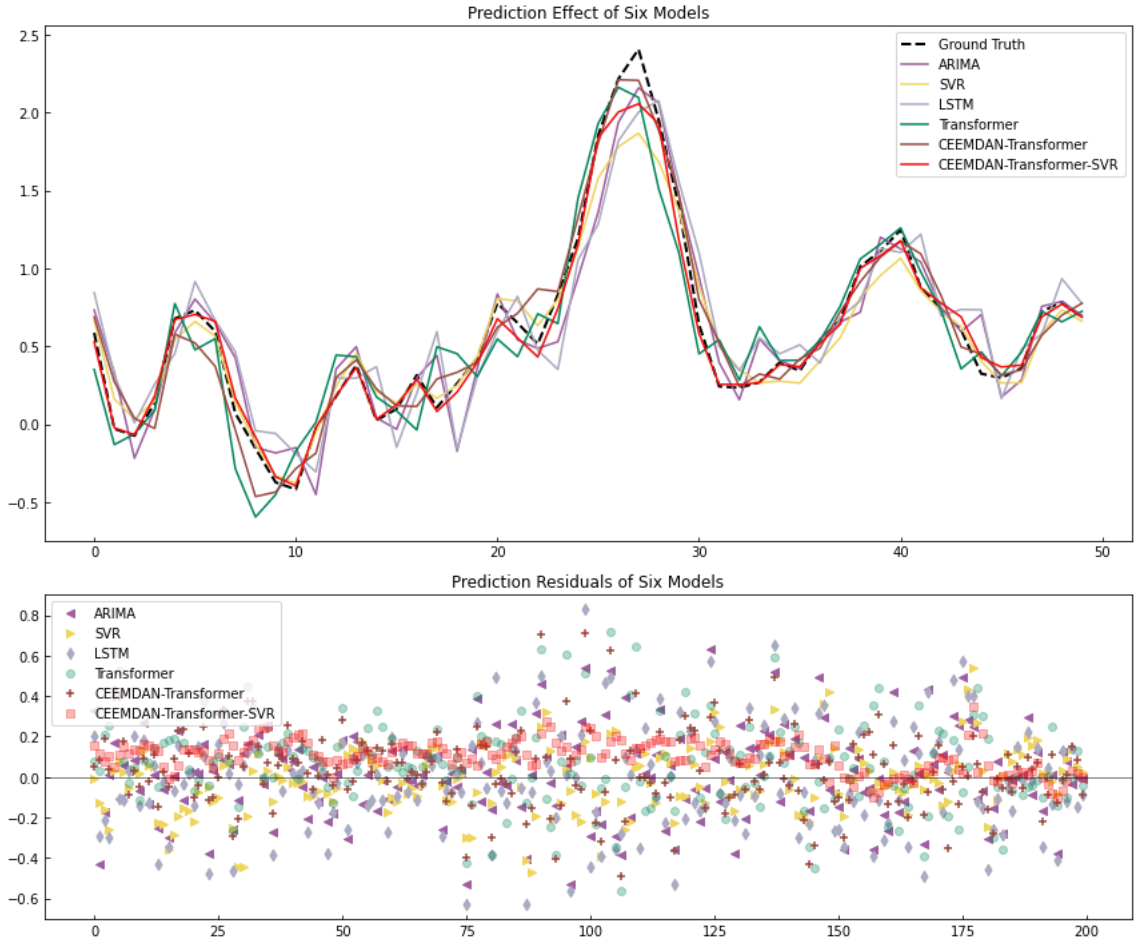

图 4-18　各模型预测效果

图 4-18 上半部分，即模型在最后 50 个时刻样本的预测效果图可十分清楚观察到，基于 Transformer 的集成学习模型效果明显优于另外几个模型。除集成学习模型外的几个模型均存在预测延迟的问题，经由上一节的分析可以得知偏差修正在一定

程度上有效地解决了预测延迟的问题。

图 4-18 下半部分，即模型在最后 200 个时刻样本预测残差散点图可以看出，基于 Transformer 的集成学习的预测残差明显小于其余模型。此外，从 Transformer 模型到 CEEMDAN-Transformer 模型，再到集成学习模型，模型的预测偏差在显著降低。

为了可以充分证实上述分析结论，本文将上述分析过程应用到多个数据集上，各模型在各数据集上的评价指标结果如表 4-6。

表 4-6　各组数据集模型评价指标

模 型		ARIMA		SVR		LSTM		Transformer		CEEMDAN-Transformer		CEEMDAN-Transformer-SVR	
评价指标		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
数据集	temperature	0.045	0.162	0.026	0.125	0.063	0.197	0.051	0.170	0.037	0.142	**0.017**	**0.112**
	exchange_rate	0.605	1.114	0.522	1.096	0.672	1.236	0.603	1.209	0.531	1.127	**0.505**	**1.088**
	electricity	0.592	0.873	0.449	0.573	0.325	0.461	0.313	0.442	0.279	0.312	**0.241**	**0.283**
	illness	3.835	1.541	**3.721**	**1.485**	6.313	2.349	5.353	2.033	4.624	1.808	3.859	1.431
	traffic	0.912	0.562	0.885	0.481	**0.820**	0.456	0.830	0.464	0.828	0.461	0.821	**0.450**
	ETTh1	2.613	1.273	2.552	1.221	2.026	1.062	1.992	1.002	1.732	0.982	**1.522**	**0.963**
计 数		0		2		1		0		0		9	

注：加粗字体为每个数据集 MSE 和 MAE 最小的模型。

上表可以看出，本文提出的基于 Transformer 的集成学习模型在除 illness 外的 5 各数据集均有良好的表现。为进一步分析不同模型对每一个数据集的预测效果，根据表 4-6 分别绘制了 MSE 和 MAE 的热力图。

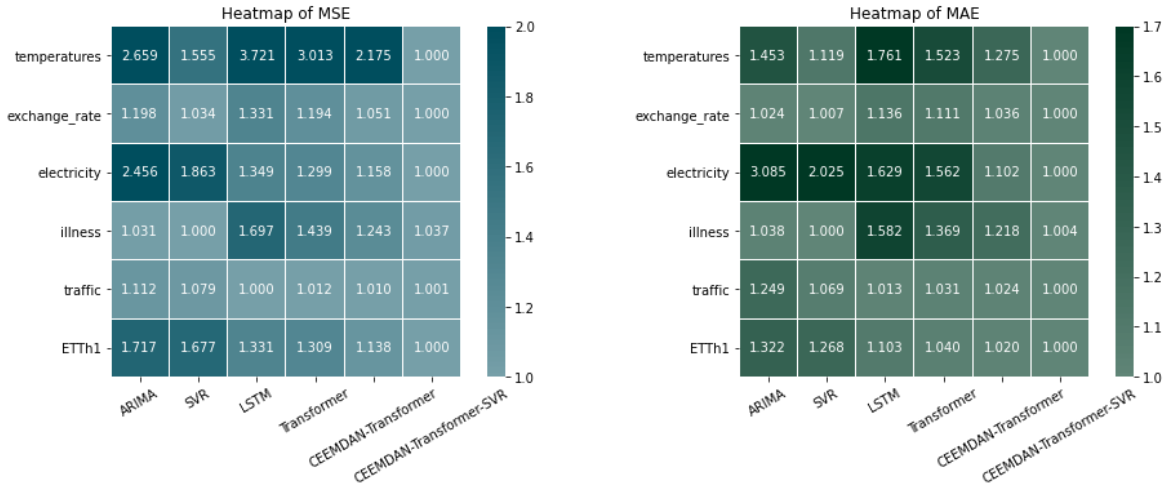

图 4-19　各组数据集模型评价指标热力图

在图 4-19 中，设每个数据集下表现最佳模型的评价指标取值为基数，计算该数据集下其他模型该评价指标的相对大小。

各数据集下各模型的表现各不相同，MSE 和 MAE 的热力图形态差异不大，因此接下来只分析每个模型每个数据集的 MSE。除了已经分析过的 temperature 数据集，接下来分析剩余的 5 个数据集的预测结果。

不同模型对 traffic 数据集(样本个数 17,542)预测结果差异不大，模型表现十分接近，图 4-2 可以看出 traffic 数据集变换形式简单，振幅和频率较为稳定，因此所含信息有限。exchange_rate 数据集(样本个数 7,586)和 illness(样本个数 964)下六个模型的表现差异不大，且传统的统计学模型 ARIMA 模型和机器学习模型 SVR 优于剩余的深度学习模型。分析可知 exchange_rate 数据集样本个数不多，此时深度学习模型(LSTM、Transformer)表现不理想，这种情况下 ARIMA 和 SVR 表现较好，尤其是在样本很少的 illness 数据集，通过十折交叉验证训练得到的 SVR 模型在六个模型中表现最优。

而随着样本个数的增加，深度学习模型在 electricity 数据集(样本个数 26,302)和 EETh1 数据集(样本个数 17,418)上的预测优势得以表现。electricity 数据集上，LSTM 模型的 MSE 较 ARIMA 和 SVR 提高了 82.1%和 38.1%，而 ETTh1 数据集则提高了 29.0%和 26.0%。

在 6 个数据集上，单独查看 Transformer 模型、CEEMDAN-Transformer 模型和 CEEMDAN-Transformer-SVR 模型，可以得知时序分解后预测和偏差二次修正的效果。

查看图 4-2，即“预处理后各数据集可视化”可以得知，各数据集振幅从大到小依次为 traffic、electricity、temperature、illness、ETTh1 和 exchange_rate。这也解释了集成学习模型对于不同数据集预测效果的提升不尽相同这一现象。

上述分析可总结出以下几点：

1. 深度学习模型的性能受限于样本量，样本有限时，经典统计学模型和机器学习模型表现更好；
2. 得益于 Attention 机制，Transformer 模型可以更好地学习到序列中蕴含的信息，其效果优于 LSTM 模型；

3. 时序分解将序列中不同频率和振幅的分量分解开进行单独的模型学习和训练，可提高模型预测精度；
4. 偏差二次修正通过单独处理 Transformer 模型在高频率高幅度子序列上的预测偏差，更有针对性地处理预测偏差，进而提升模型预测精度；
5. 时序数据振幅越大，集成学习模型提升越显著。

4.7 本章小结

本章主要将单模型 ARIMA、SVR、LSTM 和基于 Transformer 的集成学习模型在多个训练数据集上进行实证分析。

在第一小节中说明了实验环境和模型中参数的含义，第二小节阐述了本章应用的数据集的来源和预处理过程，第三小节给出了评价指标。考虑到叙述篇幅和冗余的问题，第四小节以其中一个数据集分建模分析过程为例，展示各单模型建模过程，其中由于 ARIMA 模型建模相比之下较复杂，因此对 ARIMA 模型的建模过程阐述较为详细。每个单模型建模后对其结果进行了初步的比较和分析。第五小节将本文提出的基于 Transformer 的集成学习模型应用到数据集中，逐步将建模结果可视化出来。在第六小节中，将各单模型和集成学习模型结果进行对比分析。

5 总结与展望

5.1 总结

本文以时序预测为研究对象，主要研究学习时序预测领域的算法和模型，将近几年研究热点——Transformer 模型，与时序分解算法 CEEMDAN 和 SVR 偏差修正融合到一起，提出基于 Transformer 的集成学习模型，并在六个数据集上验证了本文提出的集成学习模型的优越性能。在数据量较为理想的情况下，深度学习模型精度高于传统统计学模型和机器学习模型，且模型效果与数据集的频率与幅度紧密相关。

本文主要工作如下：

1. 对时序预测的研究背景与意义、研究现状和相关理论进行梳理和总结。重点阐述信号分解算法 CEEMDAN 和深度学习模型 Transformer 的原理、方法和步骤；
2. 将 CEEMDAN、Transformer 与 SVR 结合起来，提出基于 Transformer 的集成学习模型；
3. 将提出的集成学习模型应用到时序经典数据集(temperature、exchange_rate、electricity、illness、traffic、ETT)，为说明模型效果，选用 ARIMA、SVR、LSTM、Transformer 模型进行对比，利用 MSE 和 MAE 评价模型评价模型效果，并选用合适的可视化方法展示各模型建模过程结果。

本文研究结果表明：Transformer 模型中强大的 Attention 机制在时序预测中有较好的表现；其次，采用信号分解算法对时序数据进行分解后单独处理每个子序列可更好地挖掘时序数据中的信息；最后，偏差修正可针对性地处理模型预测偏差，提高模型预测精度。

5.2 展望

近几年深度学习得到众多学者的青睐，本文是在 Transformer 模型的基础上，融合信号分解和偏差修正，提出了集成学习模型。在模型实证分析的过程中注意到本文仍有一定的提升空间，主要是以下几点：

1. 时序数据分解后单独预测时，模型对高频数据的学习效果远不如低频数据，有学者提出可以将高频数据二次分解后再预测，但由于时间的关系，本文没有尝试这一办法。而对于低频数据，数据简单，使用复杂的模型浪费了大量的计算资源，可以考虑将低频数据融合到一起分析。相信平衡后模型的预测性能可以进一步得到提升；
2. 在数据预处理中，采用简单的 3σ 原则剔除异常值，但观察预处理后的数据，仍然有一些突变点。如何合理地检验和处理时序数据的异常值可以进一步提高时序的预测效果有待研究；
3. 有学者将参数寻优，如遗传算法、模拟退火算法，引入深度学习模型，使得深度学习模型在增加模型复杂度和计算量的情况下可以得到更高的精度，但由于时间的关系，本文在实证分析中并没有进行参数寻优，降低了计算时间的同时也牺牲了精度。

致 谢

行文至此，二十载漫漫求学路将近，无数思绪涌上心头。

宝剑锋从磨砺出，梅花香自苦寒来。从保家小学、榆树市第五中学、榆树市实验高级中学、到中国矿业大学、再到华中科技大学，这一路上经受了许多嘲笑和鄙夷，幸好有党和国家的关照、家人们的支持、老师们的肯定、同学们的陪伴，让我有继续走下去的动力。

饮水思其源，学成念吾师。感谢刘小茂老师给我机会做她的学生，刘老师对待学术和生活的态度使我受益匪浅，刘老师的耐心指导让我更加清楚努力的方向。遗憾的是，有些方面的工作我没有做的很好，离刘老师的期望还有一定的距离，期待未来有机会可以弥补。祝愿刘老师身体健康，阖家欢乐。

寒门难出贵子。我经历过有人来家里追债，却只能紧闭门窗拉上窗帘，装作家里没人的窘境。我目睹过父亲面对为几百元登门数次的债主时手足无措的样子。我记得父母因为贫穷而被亲戚看不起时的无奈与感慨。父母白手起家，辛苦了大半辈子，打工种地供三个孩子长大。一家人磕磕绊绊、相互扶持才走到今天。尽管父母给我的人生起点不高，但已经尽力给予我鼓励和自由。我知道自己算不上贵子，但我坚信增加见闻、拓宽视野后可以达到理想的高度。

岁月缱绻，葳蕤生香。很庆幸自己抗住了数不尽的压力走到现在。我深知未来有许多困难等着我，但我有信心和勇气可以面对。

愿走完这一生，回头看，是我自己的人生。

二零二二年五月 于喻家山下

参考文献

[1] 张美英,何杰. 时间序列预测模型研究综述. 数学的实践与认识, 2011, 41(18): 189-195

[2] Yule G U. On a method of investigating periodicities in distributed series, with special reference to Wolfer's sunspot numbers. Philosophical Transactions of the Royal Society of London Series A, 1927, 226: 267-298

[3] Kendall M, Wold H. A study in the analysis of stationary time series. Journal of the Royal Statistical Society Series A (General), 1954, 117(4): 484-490

[4] Slutzky E. The summation of random causes as the source of cyclic processes. Econometrica: Journal of the Econometric Society, 1937: 105-146

[5] Box G E P, Pierce D A. Distribution of residual in autoregressive-integrated moving average time series. Journal of the American Statal Association, 1970, 65(332): 1509-1526

[6] Engle R F. Autoregressive conditional heteroscedasticity with estimates of the variance of United Kingdom inflation. Econometrica: Journal of the Econometric Society, 1982: 987-1007

[7] Bollerslev T. Generalized autoregressive conditional heteroskedasticity. Journal of econometrics, 1986, 31(3): 307-327

[8] Box G E P , Jenkins G M. Time series analysis forecasting and control. Journal of Time Series Analysis, 1970, 3(3228)

[9] Tong H. Threshold models in non-linear time series analysis. Springer Science & Business Media, 2012

[10] Kim K. Financial time series forecasting using support vector machines. Neurocomputing, 2003, 55(1-2): 307-319

[11] Das M, Ghosh S K. A probabilistic approach for weather forecast using spatiotemporal interrelationships among climate variables, in: 2014 9th International Conference on Industrial and Information Systems (ICIIS). IEEE, 2014: 1-6

[12] 黄卿,谢合亮. 机器学习方法在股指期货预测中的应用研究——基于 BP 神经网络、SVM 和 XGBoost 的比较分析. 数学的实践与认识, 2018, 48(08): 297-307

[13] Lapedes A, Farber R. Nonlinear signal processing using neural networks: Prediction and system modelling. 1987

[14] Park J, Sandberg I W. Universal approximation using radial-basisfunction networks. Neural computation, 1991, 3(2): 246-257

[15] 郝晓辰, 杨跃, 杨黎明, 等. 基于时间序列卷积神经网络的水泥烧成过程能耗预测模型, 见: 2018 中国自动化大会(CAC2018). 2018: 557-563

[16] Heimes F O. Recurrent neural networks for remaining useful life estimation, in: 2008 international conference on prognostics and health management. IEEE, 2008: 1-6

[17] Hochreiter S, Schmidhuber J. Long short-term memory. Neural computation, 1997 9(8): 1735-1780

[18] Nelson D M Q, Pereira A C M, de Oliveira R A. Stock market's price movement prediction with LSTM neural networks, in: 2017 International Joint Conference on Neural Networks (IJCNN). IEEE, 2017: 1419-1426

[19] 李高盛, 彭玲, 李祥, 吴同. 基于 LSTM 的城市公交车站短时客流量预测研究. 公路交通科技, 2019, 36(02): 128-135

[20] Cho K, Merrienboer B V, Gulcehre C, et al. Learning phrase representations using RNN Encoder-Decoder for statistical machine translation. Computer Science, 2014

[21] Chung J, Gulcehre C, Cho K H, et al. Empirical evaluation of gated recurrent reural networks on sequence modeling. Eprint Arxiv, 2014

[22] Lai G, Chang W C, Yang Y, et al. Modeling long-and short-term temporal patterns with deep neural networks, in: The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval. 2018: 95-104

[23] Cirstea R G, Micu D V, Muresan G M, et al. Correlated time series forecasting using deep neural networks: A summary of results. arXiv preprint arXiv:1808.09794, 2018

[24] Birkhoff G. A limitation of fourier analysis. Journal of Mathematics and Mechanics，1967, 17(5): 443-447

[25] Huang N E, Shen S. The Hilbert-Huang transform and its applications. World Scientific, 2005

[26] Wu Z, Huang N E. Ensemble empirical mode decomposition: a noise-assisted data analysis method. Advances in Adaptive Data Analysis, 2009, 1(1): 1-41

[27] Yeh J, Shieh J, Huang N E. Complementary ensemble empirical mode decomposetion: a novel noise data analysis method. Advances in Adaptive Data Analysis, 2011, 02(02): 135-156

[28] 王妓, 李振春, 王德营. 基于 CEEMD 的地震数据小波阂值去噪方法研究. 石油物探, 2014, 53(02): 164-172

[29] 刘庆敏, 杨午阳, 田连玉, 等. 基于经验模态分解的地震相分析技术. 石油地球物理勘探, 2010(A01): 145-149

[30] Colominas M A, Schlotthauer G, Torres M E. Improved complete ensemble EMD: A suitable tool for biomedical signal processing. Biomedical Signal Processing & Control, 2014, 14: 19-29

[31] Torres M E, Colominas M A, Schlotthauer G, et al. A complete ensemble empirical mode decomposition with adaptive noise, in: 2011 IEEE international conference on acoustics, speech and signal processing (ICASSP). IEEE, 2011: 4144-4147

[32] 王文波, 费浦生, 奔旭明. 基于 EMD 与神经网络的中国股票市场预测. 系统工程理论与实践, 2010, 30(06): 1027-1033

[33] 贺毅岳, 高妮, 王峰虎等. EMD 分解下基于 SVR 的股票价格集成预测. 西北大学学报(自然科学版), 2019, 49(03): 329-336

[34] Li J, Shen J. Prediction of PM 2.5 concentration based on CEEMD-LSTM model, in: 2019 Chinese Control Conference (CCC). IEEE, 2019: 8439-8444

[35] Zhang Y A, Yan B, Aasma M. A novel deep learning framework: Panalysis of financial time series using CEEMD and LSTM. Expert Applications, 2020, 159: 113609

[36] Jeon H, Y Jung, Lee S, et al. Area-efficient short-time fourier transform processor for time–frequency analysis of non-stationary signals. Applied Sciences, 2020, 10(20): 7208

[37] 杨丽. 一种基于傅立叶变换的时延测量方法及应用. 通信技术, 2019, 52(09): 2167-2171

[38] 宋紫雯, 李晶. 信号去噪仿真实验研究. 微型电脑应用, 2020, 36(05): 72-75

[39] 刘合兵, 韩晶晶, 席磊. 小波变换—BP 神经网络的农产品价格预测研究. 中国农业信息, 2019, 31(6): 85-92

[40] 焦彦军,胡春. 基于改进 EEMD 方法的数字滤波器. 电力自动化设备, 2011, 31(11): 64-68

[41] Engle R F. Dynamic conditional correlation: A simple class of multivariate generalized autoregressive conditional heteroskedasticity models. Journal of Business & Economic Statistics, 2003, 20(3): 339-350

[42] Luukkonen R, Saikkonen P, Teräsvirta T. Testing linearity against smooth transition autoregressive models. Biometrika, 1988, 75(3): 491-499

[43] 李航. 统计学习方法. 北京: 清华大学出版社, 2012

[44] 邓乃扬, 田英杰. 数据挖掘中的新方法:支持向量机. 北京: 科学出版社, 2004

[45] Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need. Advances in neural information processing systems, 2017, 30

[46] Li S, Jin X, Xuan Y, et al. Enhancing the locality and breaking the memory bottleneck of transformer on time series forecasting. Advances in Neural Information Processing Systems, 2019

附录 1　攻读硕士学位期间取得的研究成果

发表与接收论文

[1] Jinzhou Yan, Xuejun Pei, Yi Yu, Kun Zhang, **Tianfeng Li**. EMI Receiver Modeling Based on Two-phase Scanning Mode, in: 2021 1st International Power Electronics and Application Symposium(PEAS). IEEE, 2021