

基于含启发式算子的时间窗优化 LSTM 算法

摘要

为了减弱时间序列簇状异常噪声对预测准确度的负面影响，提出了一种基于启发式算子优化的 LSTM 时序预测算法。基于异常噪声的簇状分布特性和时间序列的后效性，设计一个滑动时间窗用于同时遍历原序列和拟合序列，以二者在同一时间窗下的偏差程度分析簇状异常噪声情况并构建修复原序列的决策模型和执行模型。通过含弱化缓冲算子的灰色预测模型作为约束模型提高决策准确度，以模拟退火算法(SA)的 Metropolis 准则作为目标函数在降温的过程中调整参数并重复实验，最后基于 LSTM 网络进行预测和测试。结果表明启发式算子优化能够提升特征值序列的平滑性，减弱时间序列的簇状异常噪声的产生。本算法相较于其它预测算法的预测准确度具有显著提高，消融实验验证了本算法的每个步骤都对最后的实验结果具有正向的优化效果。

1.Introduction

气象预测的研究意义巨大。除了常规情况下对天气的预测，还能够实现气候系统的变化研究，并基于此影响人们的交通、金融等活动。气象科学需要研究和理解大气中的各项因素，例如温度、降水、气压、风力等。而其中的温度是一项非常重要的参数，对于气象科学的研究颇有意义。温度的高低及变化将会直接影响到人们的日常生活，甚至会引发火灾等容易对人们的安全产生威胁的可能。考虑到气象预测的本质是对部分和气象科学相关的物理量在时间上的预测，因此气象预测问题和时间序列预测问题息息相关。

时间序列预测是一种通过一组或多组随时间变化的变量序列的内在规律反映未来该变量具体数值的一种手段，广泛应用于气象科学、交通科学、化工科学、金融分析等多个领域。传统数学层面通常使用如灰色预测（Gray Model, GM）通过对样本数据进行一系列缓冲运算，致力于发现样本数据见动态行为的规律性，从而更好地服务于预测结果[1]。自 Liu 等建立了基本缓冲算子理论方法后，现有文献尝试构造新的合适的缓冲算子，并应用于他们研究的各行各业[2]。

随着近些年算力的提升，以机器学习、深度学习为主的智能算法向传统的时间序列预测算法发出挑战。循环神经网络（Recurrent Neural Network, RNN）[3]通过在网络设计中引入时序概念，使得其数据适应性更强；[4]Hochreiter 等人在 RNN 的结构基础上进行改进并提出了长短期记忆神经网络（Long Short-Term Memory, LSTM）模型，通过一种门结构解决了传统 RNN 模型存在的梯度爆炸和梯度消失、长期性质记忆力不足等问题，能够更加适应于长期时序信息。目前，LSTM 模型已广泛应用于自然语言处理、时间序列预测等领域。[5]基于传统 FC-LSTM 网络架构提出建立 ConvLSTM 网络并成功提取出时序图片中的空间信息，基于手写数字集 MINST 对结果进行预测以测得模型精度，并进而应用于解决雷达外推问题；在[6]中，Hu 等人通过 SSA-Bi-LSTM 神经网络预测了电力的负荷方法；[7]通过将遗传算法(GA)混合进入 LSTM 从而优化了龙山的河

道流量预测；在[8]中，基于粒子群算法(PSO)优化了LSTM模型，并预测了股票的未来情况。具体来说，他们通过粒子群算法对 LSTM 的时间窗口大小、批处理大小、隐藏层单元数目作为 PSO 优化的对象，根据粒子的适应性将例子划分为三种粒子，最后以 PSO 确定的最优值构建 PSO-LSTM 模型。

常见的时间序列噪声可能因为噪声程度过大而产生异常，这种异常噪声通常在时间上呈簇状分布。例如，一年的天气中，位于寒潮前后时间的一段区间的数据噪声将远大于正常时间段的数据噪声。[xxx]提出了一种基于深度残差收缩网络和深度长短期记忆网络的端到端性能退化评估模型，根据捕捉簇状异常噪声进行铁路车轴箱轴承的故障检测。因此，通过弱化原数据的部分簇状分布的异常噪声可能有利于模型训练后的整体预测效果。另外，理想的无噪声序列是光滑的，这种光滑序列一般总可以找到一个确切的表达式用来近似表示它。如果这个光滑序列在未来会完全按照其之前的表现形式发展，那么可以直接通过先前求得的函数表达式来预测光滑序列。然而，实际处理的时间序列一般是具备噪声的，采用传统的最小二乘回归求出含噪声的时间序列的拟合后的近似表达式并作出预测的精准度不高，现代通常采用机器学习或深度学习的方法更好地提取时间序列的数据特征并进行预测。根据前文所述，合理地簇状异常值进行去噪成为提高预测精确度的关键方向。由于对原时间序列进行拟合所求得的近似表达式是不存在噪声的，本文通过将拟合后的时间序列视作一种噪声被完全消除的情况，合理地并行考虑某一时刻下的原时间序列数据和拟合时间序列数据并处理，从而达到去噪的目的。在这个基础上引入滑动去噪时间窗模型，就能够实现对簇状异常噪声的捕捉，以便更好地去噪。

文章旨在平衡噪声对算法提取数据特征的影响，从而提高时间序列的预测精度：本文设计了一种滑动去噪时间窗，用于从待分析的时间序列中尽可能提取出有效的数据特征，并减小异常噪声信号对时间序列预测的干扰。其基本思想为三个层面：①在一定长度的时间窗内对时间窗内的离散数据（本文称为“时间步”）进行分析，根据结果判定时间步数值是否有必要修复。②通过时间窗在整个时间序列的滑动，不断地识别和分析时间步并作出可能的修复，这是为了适应簇状异常噪声在序列内簇状分布的需要。③通过时间序列每一个时点的后效性情况进行启发式寻优，合理地调整时间窗相关参数，使得时间窗尽可能适应该序列的数据特征，提高预测准确度。考虑到本文所采用的数据集是 DAT（Daily Average Temperature）时间序列，因此在后文仅讨论傅里叶函数拟合方式。有必要说明，本文的具体方法在修改回归方程后可适用于其它形状的数据上，因此本文提出的算法具有更广泛的适用性。

本文的创新之处总结如下：

1. 本文充分考虑到异常噪声的簇状分布特征，通过建立滑动时间窗口分析时间窗内的局部序列，从而决策时间窗内原时间序列是否需要修复以及如何修复，避免对整个序列进行修改，这对修复簇状异常噪声更有效。修复过程将落实到时间窗内每一个时间步，进一步精确序列的优化。
2. 本文充分考虑到时间序列的后效性对预测的启发作用，每一个时间窗将通过灰色预测模型对未来的时间窗决策模型进行约束，提高算法对不同时间序列的适应性。采用弱化缓冲算子提高灰色预测模型的平稳性。
3. 本文使用模拟退火算法(SA)对算法参数进行启发式寻优，同时采取组合模型预测的方式综合传统预测模型和 LSTM 模型，减小算法的整体时间复杂度。

本文的其余部分结构如下：第二节提供了该算法研究的相关前人研究背景。第三节介绍了时间序列预测领域的一些通用方法，第四节提供了本文的时间窗模型设计。第五节给出了整套算法的工作流程。第六节给出了实验的仿真和分析，包括数据集和实验配置、xxx。第七节概括本文的结论，并且概述了未来的工作方向。

2.Related Work

过去的研究在某些情况或者方面存在一定的缺陷。在[9]中，通过 VMD 对时间序列进行模态分解，采用麻雀算法(SSA)对 VMD 进行寻优，根据最小冗余最大相关性(mRMR)筛选和优化特征集合，最后采用 LSTM 模型进行预测，这种计算方式虽然利用对相关性的量化提高了 LSTM 的预测精度，但是对原时间序列的变分模态分解(VMD)进行超参数寻优的可解释性较差，且 VMD 的变换在数理计算上较为繁琐；在[10]中，作者通过 ConvLSTM 解决了广西降水预报问题，其结果仅仅能够证明 ConvLSTM 在视频流处理的过程中具有更高的精度，但是几乎没有在数学或者算法上进行优化，因此精度有待进一步提高；[11]通过 LSTM 模型解决了电力负荷识别的相关问题。其较大的创新之处是在时间序列的预处理过程中引入高斯窗对事件是否为负荷事件进行判定，从而对负荷事件以及非负荷事件采用分治的方法以提高算法的精度。但是高斯窗的计算方式较为简单，不能很好地区分电压、电流于某一区间变化不剧烈的事件从属情况。另外，文章并没有对高斯窗的系数（例如高斯窗的长度、方差、阈值）进行寻优或分析，在实验的处理上具有一定的主观性。在处理时间序列的过程中，有研究曾对某一个维度的时间序列进行分析，也有对多个维度的时间序列同时进行分析，致力于求解出他们的相关性，如文献[12]通过将 ResNet 与 LSTM 模型进行结合，在引入注意力机制的情况下采用译码-解码的思想，预测亚利桑那州立大学的 Campus Metabolism 系统所提供的 Tempe 校区电、冷、热负荷的多维度历史数据，但是在该时间序列的数据约束下，算法难以发挥其应有的精确度，另外该研究忽略了求解过程中对算法的时间复杂度与准确度的平衡；在文献[13]中，通过 EMD 对时间序列进行经验模态分解，将时间序列数据分解后输入混合的 CNN-LSTM 模型并进行短期电力负荷预测；这种做法需要将电价与温度串联为词向量，但是该研究没有具体展开解释电价和温度以何种方式串联，因此不具备较高的可解释性。在文献[14]中，通过 Bi-LSTM 模型对非侵入式负荷(NILM)进行监测，但是在时间序列的基础上对时间序列进行投入和切除没有进行具体的实验论证，有待进一步优化。除此以外，学术界还尝试使用组合模型办法对时间序列进行预测。例如文献[15]通过 PCA-ARMA-EGARCH 通过多个维度对气温进行预测并进行证明分析，但是这种组成模型过于传统，当数据量扩大时将远落后于大多数深度学习算法；对周期性时间序列的预测也是目前的一大研究方向，如在文献[16]中，通过对周期性进行分解从而对未来电力价格进行周期性时间序列预测，但是其创新主要体现在对超参数的手动优化上，然而这种方式可能会陷入局部最优解，从而无法搜寻到全局最优解，另外手动调整参数的办法具有主观性。模拟退火用作参数寻优并不罕见，如在[17]中，利用模拟退火算法（Simulated Annealing, SA）优化 LSTM 的超参数，形成的预测模型具有较高的预测精度和泛化能力。在[18]中，针对存在重大波动和非平稳性时间序列的数据特征，提出机器学习相关方法处理这些效应，本文根据现

有研究基础，针对输入模型的时间序列波动性较大和不确定性问题，提出一种基于函数拟合的、以模拟退火算法和含弱化缓冲算子的灰色预测模型优化的长短期记忆神经网络（LSTM），模型从而实现一种周期性时间序列预测方法。[19]研究了优化傅里叶级数对时间序列的影响，但是在方法上不够[20]中使用了堆叠自动编码器和长短记忆的金融时间序列深度学习框架达到了较好的效果，但是该研究并未对算法作出创新，而仅仅是对比了四种常见的深度学习算法。另外，过去的研究大多集中在对神经网络超参数的寻优或者对网络拓扑结构的改变上，但对于对原有数据在算法上进行降噪处理的研究则甚少，而是否进行降噪处理或者降噪处理带来的影响将成为决定预测精度的关键。总体来说，在过去的时间序列算法研究中不难发现，智能算法的选取、算法的时间规模、超参数的设定或寻优以及算法处理中各个步骤的相关性和可解释性将极大地影响算法所表现的效果。这些内容又可进一步归纳为两个方面，一是对数据特征的采集方式有待提高，二是算法的时间复杂度与算法准确度有待于更好地权衡。

针对前期研究人员在研究过程当中存在的某些缺陷和不足，本文综合考虑了时间序列各个时间步的后效性影响，引入一套支持超参数寻优的时间窗用于提取序列相关特征，从而一方面提高算法的可解释性，另一方面提高应用在不同时间序列的泛化性。在超参数寻优过程中，本算法采用含弱化缓冲算子的灰色预测算法进行寻优，保证了计算过程的精简性，缩短了计算耗时，提升了特征值作为中间变量的平滑性，从而能够更好地进行超参数寻优。本算法通过模拟退火与组合预测模型相结合，尽最大努力将本算法的时间复杂度下降为 LSTM 本身，同时令结果尽可能逼近全局最优。具体来说，本算法以合理提取时间序列数据特征为研究目标，尽可能消除数据的局部剧烈抖动（本文定义为“异常值”）对神经网络训练的影响，同时通过与光滑的拟合序列进行算法对照以保证时间序列的粗糙程度，也意味着时间序列的数据特征将有效的保留，这对于深度学习模型的训练是必需的。本文称每一个时间序列上的离散点为“时间步”，以时间步作为基本对象进行研究。

3. Proposed methodology

3.1 长短期记忆神经网络

LSTM 是一种特殊的循环神经网络，通过多个同构单元格组成，善于通过内部状态长时间储存信息。LSTM 模型具有 3 个控制门，即由输入门、输出门、遗忘门三个复合控制门单元组成，能够对所有神经元进行读写和重置。LSTM 相比于以往的 RNN 增设了 softmax 层，通过选取对数似然函数作为误差反传的损失函数。LSTM 能够对信息进行选择性通过.[22,23]。

遗忘门 f 用于记录 x_t 的遗忘程度，其保存部分进入输入门；输入门 i 中的 x_i 经过激活函数 \tanh 和 sigmoid 的共同作用保存向量，最后经过输出门 h 输出当前数值。其表达式有如下所示：

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \quad (7)$$

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \quad (8)$$

$$\widetilde{C}_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \quad (9)$$

$$C_t = f_t C_{t-1} + i_t \widetilde{C}_t \quad (10)$$

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \quad (11)$$

$$h_t = o_t \tanh(C_t) \quad (12)$$

其中 x 表示 LSTM 单元的输入向量， f 、 i 、 o 分别表示遗忘门、输入门和输出门， h 表示单元输出向量， C 表示单元状态；下标 t 表示时刻， σ 、 \tanh 分别表示 sigmoid 和 \tanh 激活函数， W 为随机权重矩阵， b 为偏置向量。

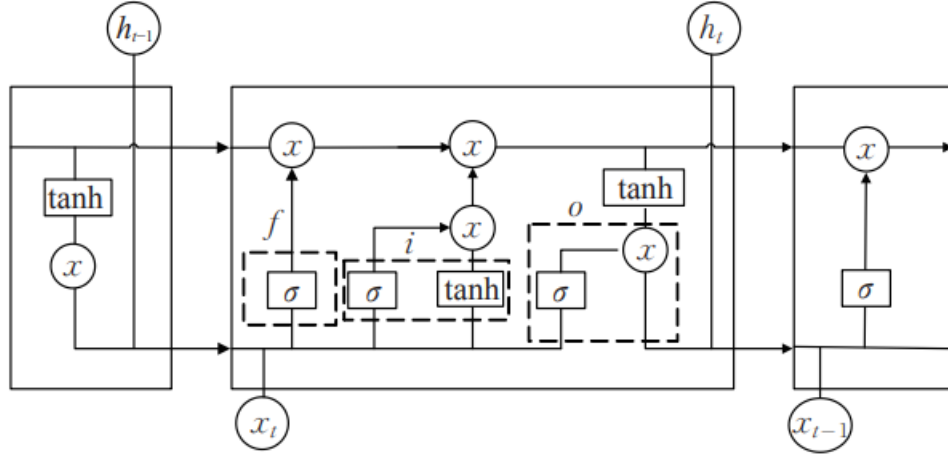


图 1 LSTM 拓扑结构

在 LSTM 的设计中，关键在于 C 在 t 时刻保持单元状态的记忆，并且通过遗忘门和输入门进行调节。输入门的作用是针对更新单元状态做出许可或者阻止的判断；输出门的作用是控制单元状态输出和传输到下一个单元；遗忘门的作用是让细胞记住或者忘记它之前的状态；反向传播算法是最常见的训练方法[24]。

3.2 弱化缓冲算子与灰色预测模型

在文献的理论基础上，引入弱化缓冲算子作用，降低扰动项对原始序列的冲击扰动作用，提高数据预测的准确性[26]。

设 $X = (x(1), x(2), \dots, x(n))$ 为原数据序列，令

$$XD = (x(1)d, x(2)d, \dots, x(n)d) \quad (15)$$

$$x(k)d = \frac{\sum_{j=k}^n \frac{x(j)}{j(j+1)}}{\sum_{i=k}^n \frac{1}{i(i+1)}} \quad (16)$$

则 XD 即为所求。其中 i 表示从 k 至 n 的自然数， $x(k)$ 表示序列中的第 k 个元素， $x(k)d$ 表示算子作用后的第 k 个元素。只要 X 符合单调递增、单调递减或者震荡序列的时候， D 均为弱化缓冲算子。

在灰色预测模型中，设原始数据为 $\{x^{(0)}(k)\}, (k = 1, 2, \dots, n)$, n 为灰色预测原始数据长度。对这些原始数据应用缓冲算子，得到异界缓冲生成序列 $\{x^{(0)}(k)d\}, (k = 1, 2, \dots, n)$ ，对应一阶累加和为

$$x^{(1)}(k) = \sum_{j=1}^k [x^{(0)}(j)d] \quad (17)$$

其均值生成序列为

$$z^{(1)}(k) = \frac{x^{(1)}(k) + x^{(1)}(k-1)}{2}, (k = 2, 3, \dots, n) \quad (18)$$

构造对应数据矩阵

$$B = \begin{bmatrix} -z^{(1)}(2) & 1 \\ \dots & \dots \\ -z^{(1)}(n) & 1 \end{bmatrix}, Y = \begin{bmatrix} x^{(0)}(2) \\ \dots \\ x^{(0)}(n) \end{bmatrix} \quad (19)$$

使用最小二乘法计算矩阵 B 得到

$$U = (B^T B)^{-1} B^T Y \quad (20)$$

根据灰色预测理论求解对应参数 a 、 b 后计算其对应的时间响应式

$$\hat{x}^{(0)}(k) = (1 - e^a) \left(x^{(0)} - \frac{b}{a} \right) e^{-a(k-1)} \quad (21)$$

其中 $\hat{x}^{(0)}(k)$ 表示第 k 个模拟预测值， a 、 b 为灰阶作用量和灰色发展系数。根据上述理论模型，能够计算基于弱化缓冲算子的 GM(1,1)模型。

4. Denoising time window modeling

本文采用最小二乘回归生成原时间序列对应的拟合时间序列，并采用一个滑动去噪时间窗用于完成簇状异常值的去噪工作。它有利于后期的深度学习模型提取相关特征并进行预测。本时间窗的介绍分为三个部分：首先介绍用于判定时间窗的窗内数据是否包含簇状异常噪声的决策模型，其次介绍时间窗判定窗内存在簇状异常噪声后对原时间序列的修复模型，最后介绍用于影响时间窗去噪规则的约束模型。

模型间的关系如下图所示：在未确定最终向 LSTM 输入的时间序列之前，每一轮去噪时间窗总是包括时间窗滑动和退火判定两个过程。在时间窗的滑动过程中，总是先通过决策模型根据因子库内的因子判定是否需要进入修复模型，如果进入修复模型，则进行修复并进入灰色预测模型，否则直接进入灰色预测模型。灰色预测模型通过弱化缓冲算子计算特征值向量 res ，并将结果反馈给因子库，直到滑动阶段结束。滑动结束后，退火判定会通过组合预测模型对当前启发式效果进行判定，如果预测准确度持续下降则选取过去所有退火组别中最有效的一组输入 LSTM 模型并预测，否则通过退火来调整因子库内参数，并重新建立时间窗口以重复实验。

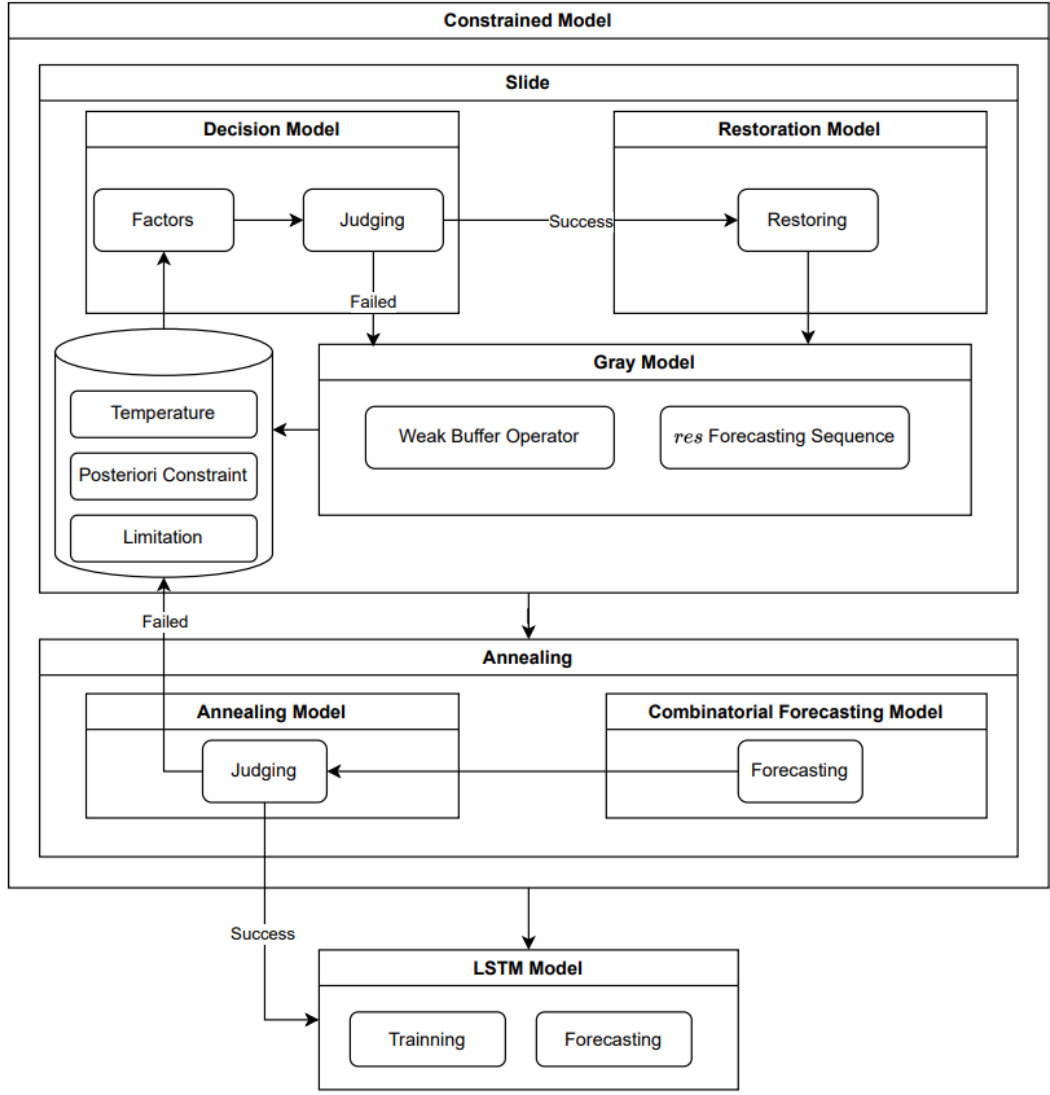


图 x 模型关系图

4.1 决策模型设计

本节阐述的决策模型将用于算法判定时间窗内的原序列是否有必要进行修复。在训练集设计中，滑动时间窗被视作是训练集角标 $index$ 迭代过程中的子循环，表现为在 $index$ 自加的过程中，时间窗内部会同步进行相关变量的计算。每次建立一个滑动时间窗口 i ，就会在当前 $index$ 的位置建立一个指针 j 。随着 $index$ 自加，指针 j 分别计算当前真实值与对应傅里叶函数值的差值的绝对值，并将该绝对值保存在下标为 j 的向量 Δy 中。时间窗存在一个自身的窗口长度 L ，当 j 迭代到该窗口长度大小的时候，时间窗基于当下子循环向量 Δy 求出时间窗 i 对应方差。设训练集原数据一维时序向量为 y ，通过拟合运算求得原数据对应的傅里叶函数为 F ，则其数学公式为

$$\Delta y_j = |y_{index} - F(index)| \quad (22)$$

$$u(i) = s^2_i = \sum_{j=1}^L (\Delta y_j)^2 \quad (23)$$

每计算一次时间窗，记录其时间窗特征值 $u(i)$ 为当前时间窗内各个离散时间点真实值与预测值的方差的求和，并给定一个新的下一轮理论特征值 $res(i+1)$ 用于下一次时间窗的算法流程控制判断。

·每次迭代一份时间窗 i ，通过流程控制决定是否执行时间窗后续算法。这里介绍两种情况：

1. 时间窗 i 不满足后续算法要求，则该时间窗内的原数据不进行后续算法处理。对于每一个当前时间窗内的原数据 y_{index} ，都有 $Y_{index} = y_{index}$ 。其中 Y 是经过前期优化算法处理得到的新的一维时序向量，这个向量将作为后期改进LSTM的训练集。在这种情况下，给定下一个时间窗的理论特征值变量为本次时间窗的特征值

$$res(i+1) = u(i) \quad (24)$$

考虑到时间窗有可能接近簇状异常噪声，下一个时间窗将仅在上一个时间窗的基础上移动一个时间步长，从而避免遗漏异常噪声。

2. 时间窗 i 满足后续算法要求，则在时间窗内建立一个新的下标 pos 表示时间窗的第一个元素位置，并重置指针 j 指向 pos 所指元素，进行时间步算法修正。其中下标 pos 指的是该元素在总序列的位置，指针 j 表示该元素在当前时间窗的位置。由于本步骤已经捕获窗口内的簇状异常噪声，下一个时间窗将在上一个时间窗的基础上移动一个时间窗口长度 L 。

4.2 修复模型设计

本节阐述当决策模型判定修复后时间窗内数据具体的修改方法。当时间窗内原始数据需要进行算法调整的时候，视连续傅里叶级数函数上与原始数据时刻对应的每一个变量为时间步，并对时间步进行对照修正。

下图给出了一个时间窗内的时间步调整方法。

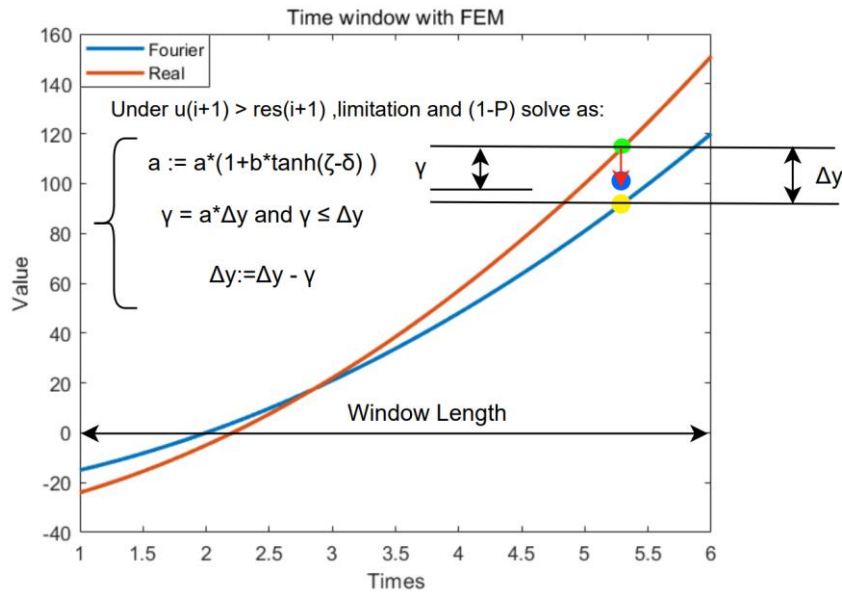


图 2 时间窗调整下的时间步处理

注意：为作图方便，上图的两个函数分别选自 $3x^2 + 6x - 24$ 与 $4x^2 + 7x - 35$ ，
非真实数据

对每一个时间步，求出其傅里叶函数对应值与实际值的相对误差

$$\zeta_j = \left| \frac{\Delta y_j}{y_{pos}} \right| \quad (25)$$

对相对误差进行分析：分别设置上限和下限 δ_1 和 δ_2 ，并根据该误差的所属区间确定是否有必要对该时间步进行修正。如果 $\zeta_j > \delta_1$ 或者 $\zeta_j < \delta_2$ 都有必要进行修正，这是为了防止数据与理论正常函数轨迹偏差太大，或者与理论正常函数相差太小导致无法提取特征的情况。在判断该时间步有必要进行调整时，记调整因子 a 为

$$a := a \left(1 + b \tanh(\zeta_j - \delta_q) \right) \quad (k = 1, 2) \quad (26)$$

调整因子是关于时间步的一个修饰变化程度的变量。该调整因子越大，调整的偏差值就越大。在建立第一个时间窗的时候，都会给出一个初始的 a_{begin} 用作后期调整因子的修改需要。

在公式（24）中， $\zeta_j - \delta_q$ 表示实际误差百分比与上下限百分比的具体差值，即该偏差值越大，调整因子本身改变的幅度就越大。当实际误差大于给定的误差上限的时候， $q = 1$ ， $\tanh(\zeta_j - \delta_q) > 0$ ；当实际误差小于给定的误差下限的时候， $q = 2$ ， $\tanh(\zeta_j - \delta_q) < 0$ 。考虑到双曲正切函数的值域介于 $(-1, 1)$ 区间，因此实际上通过双曲正切函数调整的调整因子可能并不能达到变化所要求，故设置常数 b 用于修饰双曲正切部分的偏差额度。

这个调整因子会进入下列的调整部分。其中记调整值为 γ_j ，有

$$\gamma_j = \begin{cases} a \Delta y_j & \text{if } \gamma_j < \Delta y_j \\ \Delta y_j & \text{else} \end{cases} \quad (27)$$

$$Y(pos) = \begin{cases} y_{pos} + \gamma_j & \text{if } y_{pos} < F(pos) \\ y_{pos} - \gamma_j & \text{else} \end{cases} \quad (28)$$

上述公式可表示为，计算一个偏差量额度 γ_j 为调整因子 a 与当前位置原始数据与傅里叶函数对应数据的差值的乘积。如果 $a > 1$ ，那么该偏差量额度重新调整为当前这个差值本身。这个偏差量额度会将原数据做一步调整，使得新向量的当前标量 $Y(pos)$ 更加接近傅里叶函数值。

根据上述时间步分析，概括为以下的工作流程：首先，对于所有时间窗，只有满足算法流程控制的时间窗才能够被允许进入调整阶段。在调整阶段，每一个时间窗内的时间步都会基于真实值和傅里叶理论值误差进行分析，符合条件的时间步将依次迭代计算调整因子 a 以及对应的偏差量额度 γ_j ，并将该偏差量额度和原数据经过运算形成新的数据 $Y(pos)$ ，这个数据构成的向量将参与后续

LSTM 预测的训练部分。这种计算方式利用时间序列的后效性，每一个时刻计算的迭代因子 α 将影响后期的滑动时间窗处理，提高了不同问题背景下参数与数据的适配程度。

4.3 约束模型设计

本节对时间序列的预测通过启发式方式构建约束模型，从而提高预测准确度。本文引入了含有弱化缓冲算子的灰色预测模型，利用时间序列的后效性性质对特征值序列进行处理，不断优化时间窗的参数，从而提高时间窗对簇状异常值的提取效果。本文同时引入模拟退火算法和组合预测算法，每次时间窗遍历结束后，使用距离深度学习模型准确度最接近的其它模型对结果进行预测并获取预测的准确度。为了进一步提高滑动窗口参数与数据的适应程度，采用模拟退火算法参与参数的调整。当预测准确度持续降低时停止退火操作，否则算法将通过退火调整滑动时间窗参数，并重新在时间序列的起点建立一套滑动时间窗并再次遍历。

4.3.1 后效性约束

每一次时间窗都能根据本次由方差求得的特征值迭代出下一个时间窗的理论特征值。在上文提到，对于不参与后续算法流程的时间窗，下一个时间窗的理论特征值即为本次时间窗的特征值（见公式（22））。对于参与后续算法流程的时间窗，则采取基于含缓冲算子的灰色预测手段进行预测。

根据本文 2.4 有关含弱化缓冲算子的灰色预测 $GM(1,1)$ 的方法，定义一个函数 G 满足

$$res = c * G(u) \quad (29)$$

在公式（27）中， c 为调整含缓冲算子的特征值函数的系数常数，防止由于灰色预测迭代失衡导致理论特征值 res 向量与每次时间窗对应的真实特征值 $u(i)$ 存在较大的差异。向量 u 为当前实验循环下所有以 i 为序号的时间窗构成的一维时序向量集合。函数 G 为经过含弱化缓冲算子的灰色预测办法，输出的 $G(u)$ 即向量 u 通过函数 G 预测的长度为 $i + 1$ 的向量，该向量更新当前的长度为 i 的特征值向量 res 。在这个办法下， $res(i + 1)$ 即为所需的下一个时间窗的理论特征值，这个特征值将参与算法流程控制。

4.3.2 序列修复约束

在本文设计的模型中，每一次时间窗滑动结束后（从序列开始滑向序列结尾）都会进入一个组合模型预测对本轮时间窗滑动效果进行判定，这个判定决定了是否有必要对时间窗的每个离散时间步进行分析和调整。本步骤采用一个概率

$$P = \exp\left(-\frac{|u_i - u_{i-1}|}{KT}\right) \quad (30)$$

来表示当前有多大的概率跳出该局部最优的情况（其中 T 为当前温度）。在每一个时间窗 i 运行决策模型时，如果 $u(i) > res(i)$ ，则以一个概率 $1 - P$ 在决策模型中令时间窗跳过本轮对时间序列的修复，否则按照原决策进行。

组合模型预测将获取该轮次时间窗处理的情况下的近似预测准确度，如果准确度持续下降，则选取当前所有退火实验组中准确度效果最好的一组时间序列用于输入 LSTM 并进行预测，否则重新构建滑动时间窗，并继续修复原序列。

4.3.3 组合模型预测

在实际问题分析过程中，不同的传统对比模型如 SVM, BP, ARIMA 等，可能预测精确度情况不一。在不同的情境下，最优模型的选取也是不同的。它们的共性在于传统对比模型的时间复杂度远小于神经网络训练、推理等形成的时间复杂度。因此，本算法分为两个阶段：

1. 退火阶段：退火阶段将持续构建滑动时间窗对原序列进行修复，每次修复后都会通过退火调整参数并重复实验，直到退火符合要求。
2. 停火阶段：采用退火阶段结果最好的一组实验通过 LSTM 进行预测。

为了在退火阶段获取当前轮次修复后的时间序列的预测效果，算法采用当前预测方法中距离 LSTM 模型预测准确度最为接近的传统数学预测方法或者普通机器学习方法对该序列进行预测，得到本轮退火的预测准确度，并参与是否退出退火阶段并进入停火阶段的判定，从而减小算法的时间复杂度。设滑动时间窗处理原时间序列的时间复杂度为 $O(n)$ ，本轮预测的时间复杂度为 $O(p)$ ，则每一轮次的算法时间复杂度为 $O(n + p)$ 。如果每一轮次都通过 LSTM 进行预测，则设 LSTM 预测的时间复杂度为 $O(q)$ ，原始方法的时间复杂度为 $O(n + q)$ 。进行 m 轮退火后的时间复杂度分别为 $O(m(n + p))$ 和 $O(m(n + q))$ 。由于 q 远大于 p ，当轮次 m 较高时，后者的时间复杂度将远大于前者。

LSTM 的网络设计如下：在本文的实验中，采用了一层 LSTM 的拓扑设计方法，在 LSTM 的最后加入了一层全连接层。当构造网络的初始参数之后，网络将通过输入的训练集数据序列进行训练，并预测一定时间步长的结果。这个结果会与先前的数据进行对比，计算预测的准确度。如果预测的准确度没有达到实验的预期效果，LSTM 预测实验将调整超参数并循环进行，直到结果比较符合为止。

5. Algorithm Design

在上述滑动时间窗处理原时间序列的基础上，本文提出了一种启发式优化的 LSTM 算法（简称 SA-GM-LSTM 算法），有效解决了簇状异常噪声对时间序列预测效果的负面影响。算法的执行操作如下图所示。

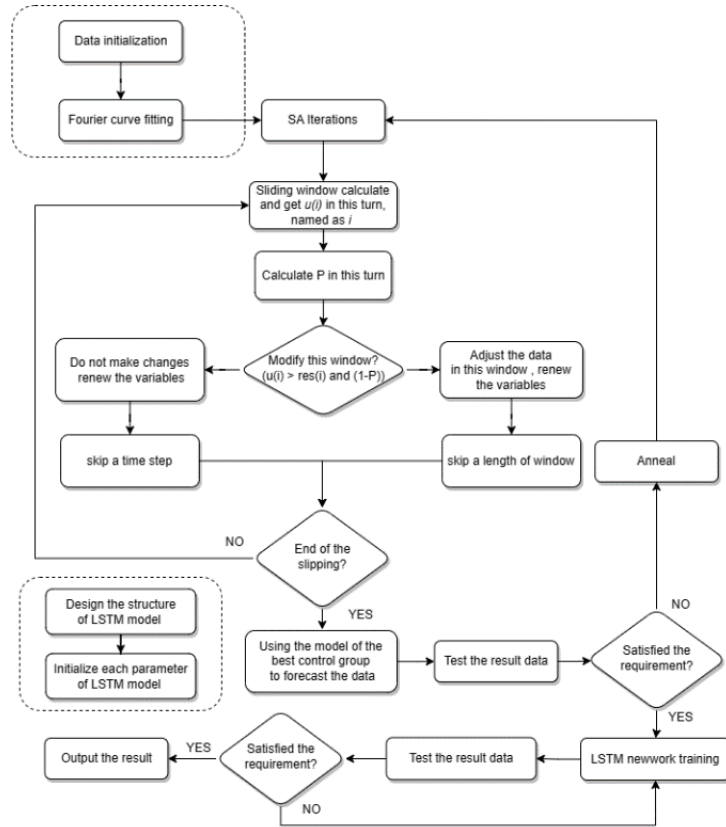


图 3 算法流程图概述

算法首先对数据进行预处理并将原时间序列通过最小二乘回归生成相应的拟合序列，然后建立一个启发式滑动时间窗并遍历这对时间序列，从而判定局部噪声是否存在异常，并在存在噪声异常的情况下对原始噪声进行修复。算法将以时间窗内原序列与拟合序列的离散程度大小作为判定该时间窗内的原数据噪声是否异常的依据，并在判定异常后依赖这对序列在时间窗内每一个相同的时间步下的差值对原数据噪声进行修复。

算法通过时间序列的后效性对滑动时间窗中用于判定簇状异常噪声事件的参数进行启发式寻优，使得滑动时间窗的参数在时间的推演下不断地适应本数据集。具体表现在每一次时间窗结束异常判定或判定异常后结束修复，都会基于含有弱化缓冲算子的灰色预测模型根据过往的各个离散程度形成的时间序列计算一份特征值时间序列，作为和下一个时间窗的离散程度进行异常判定的依据之一，一方面提高特征值时间序列的平滑性，进而使得该序列更能反映过去数据整体的后效性特征，另一方面根据过去数据的特征值序列能够反映数据集整体的离散程度情况，进而有效地启发本次时间窗下对异常的判定。

算法引入组合模型预测和模拟退火算法，设定一个初始温度作为滑动时间窗的参数之一，随后进行退火操作。退火操作要求在每一轮滑动时间窗遍历训练集后通过组合预测算法生成预测的时间序列，并将其与测试集进行对比，输出预测的准确度。当准确度在数轮退火中持续出现下降趋势时，取准确度最高的退火轮次对应的修复后的时间序列进行停火操作：停火操作要求算法对该序列进行 LSTM 预测，并输出最后的预测序列和预测的准确度，这是本算法经求解后形成的最终结果；如果数轮退火后未出现持续下降的趋势，算法将进一步做退火操作并重新建立一套滑动时间窗用于遍历序列。退火导致的降温将启发式地调整滑动时间窗的参数，进而影响时间窗的决策模型，有利于令优化结果

跳出局部最优，达到全局最优。为了减小算法的时间复杂度，要求组合算法在退火过程的每一个轮次都采用该数据集下最接近深度学习算法准确度的传统时间序列预测算法对训练集进行预测，并将预测准确度纳入停火的判定标准。

下面给出了本算法的有关伪代码表述。

Algorithm: SA-GM-LSTM Method

Input: Original time sequence Seq_1 , initial temperature T_0 , window length L , annealing speed k , two types of limitations δ_1 and δ_2 , the length of the sequence len , factor parameters a and b , a limitation of the increase lim , Test time sequence Seq_3

Output: Future data Seq_5

```

1: Pre-processing to remove outliers and defective values in  $Seq_1$ 
2: Fitting using least squares to obtain a fitted sequence shaped like a Fourier series,
   named as  $Seq_2$ 
3: While SA do not satisfied the requirement do
4:    $idx \leftarrow idx + 1$ , where  $idx$  first begins at 1.
5:    $T \leftarrow T_0 - k\Delta T$  in order to annealing
6:   Initialize  $pos, w_i \leftarrow 0$ , where  $pos$  means the start position of time window
7:   While  $L + pos \leq len$  do
8:      $w \leftarrow w + 1$ 
9:     Calculate the variance between  $Seq_1$  and  $Seq_2$  in the window, named as  $u(w)$ 
10:    Calculate the probability  $P \leftarrow \exp(-(u_w - u_{w-1})/kT)$ ,
       generate a random number from 0 to 1, named as  $Q$ 
11:    if  $i$  is not yet enough to make GM method or  $Q < P$  or  $u(w) \leq res$  do
12:       $res \leftarrow u(w)$ 
13:       $pos \leftarrow pos + 1$ 
14:    else do
15:      for  $j \leftarrow pos$  to  $pos + L$  do
16:         $\sigma(j) \leftarrow \text{abs}(Seq_1(j) - Seq_2(j))$ 
17:        if  $\sigma(j) < \delta_1$  or  $\sigma(j) > \delta_2$  do
18:           $a \leftarrow \max(lim, a \cdot (1 + b \cdot \tanh(\text{abs}(\sigma(j) - \delta_{type}))))$ , where
               $\delta_{type}$  refers to the conditions
19:        end if
20:         $\gamma \leftarrow a\sigma(j)$ 
21:         $Seq_1(j)$  changes  $\gamma$  in order to make closer to  $Seq_2(j)$ 
22:      end for
23:      Using the weak buffer operator to process the sequence  $u$  into  $v$ 
24:      Using the GM model to predict the next step of  $v$ 
25:       $res \leftarrow v(w + 1)$ 
26:       $pos \leftarrow pos + L$ 
27:    end if
28:  end while
29:  Using a suitable algorithm to forecast the data, named as  $Seq_4$ 
30:  Comparing the  $Seq_3$  and  $Seq_4$  to calculate the RMSE result  $ans(idx)$ 
31: end while
32: Transferring the sequence  $Seq_1$  into optimized LSTM neural network for training
33: Using the training model to forecast the future data  $Seq_5$ 

```

6.Experiment

6.1 数据和配置

本文以北京在 2006 年至 2022 年全市区地表以上 2 米的大气温度作为初始数据集。根据数据集统计，在 2006 年 1 月 1 日至 2010 年 1 月 19 日期间，该数据集统计了北京每日进行的 4 次采样，即每日的 2 点、8 点、14 点和 20 点，每次间隔 6 小时固定采样；在 2010 年 6 月 20 日至 2022 年 12 月 31 日期间，该数据集统计了北京每日进行的 8 次采样，即每日的 2 点、5 点、8 点、11 点、14 点、17 点、20 点和 23 点，每次间隔 3 小时固定采样。

在数据的预处理阶段，首先分别对每日多次采样的数据取平均值。采用迭代的方式对每日各采样对应温度进行求和后求平均值，对 2010 年 1 月 19 日以前的数据求和 4 次再求平均值，对之后的数据求和 8 次以后再求平均值，即可得到每日平均气温 DAT。数据经过处理以后存在 5 份 NaN 值，对于这 5 个值对应的日期，分别获取日期前一天和后一天的温度值进行求和，除以对应的 2 份天数，补全缺陷值。

为了能够进一步体现出含有弱化缓冲算子的灰色预测模型的精度，对 2006 年自 2022 年的所有 DAT 数据，以对应年份是否为闰年为标准继续做一步求和后求平均的处理，即可得到每一年的平均气温。

以 2006 年至 2018 年为训练集、2019 年至 2022 年为测试集，对数据集进行训练集和测试集的划分。实验的结果均由测试集相关计算得到。

本文采用的用于计算准确度的指标为国际通用的部分评价标准，Mean Absolute Error(MAE),Sum squared error(SSE), Mean Squared Error(MSE), Root Mean Squared Error(RMSE),Mean Absolute Percentage Error(MAPE),计算公式如下：

$$MAE = \frac{1}{n} \sum_{i=1}^n |\hat{y}_i - y_i| \quad (32)$$

$$SSE = \sum_{i=1}^n (\hat{y}_i - y_i)^2 \quad (33)$$

$$MSE = \frac{1}{n} \sum_{i=1}^n (\hat{y}_i - y_i)^2 \quad (34)$$

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (\hat{y}_i - y_i)^2} \quad (35)$$

$$MAPE = \frac{100\%}{n} \sum_{i=1}^n \left| \frac{\hat{y}_i - y_i}{y_i} \right| \quad (36)$$

上述公式中，设数据长度为 n 的序列变量，对应的自然数 i 从1遍历到 n ， y_i 为实际值， \hat{y}_i 为预测值。上述误差值越小，模型的准确性越好。

本文的所有实验均在 Windows10 64 位操作系统下进行，设备采用的处理器

为 Interi7-10750H 2.60GHz，显卡为 NVIDIA GeForce GTX 1650 Ti。实验环境是 MATLAB R2021b。

6.2 模型结果分析

本文对北京市自 2006 年至 2022 年的地表以上 2 米的温度数据进行时间序列预测，为了进一步得出算法对时间序列处理的效果，本文针对簇状异常噪声现象进行处理和分析。首先，针对经过 SA-GM 处理前后的时间序列以及傅里叶级数对应的时间序列，通过在 MATLAB 上打印以呈现。其次，通过计算经过 SA-GM 处理前后的时间序列本身对应的数值与傅里叶级数序列对应的数值在时间区间上的差值并打印，观察经过 SA-GM 处理以后时间序列的簇状异常噪声情况。其中，蓝色曲线代表北京市自 2006 年至 2022 年地表以上 2 米的实际温度时间序列，红色曲线代表对蓝色曲线在该时间区间下的拟合的傅里叶级数，黄色曲线代表蓝色曲线所对应的数据经过 SA-GM 处理以后的改进温度时间序列。

在参数的设置上，傅里叶级数表达式为

$$F(t) = c + \sum_{i=1}^8 a_i \cos(iwx) + b_i \sin(iwx)$$

(38)

拟合参数见如下表格。

表 3 原数据集处理后的参数表格

参数	数值	参数	数值
w	0.9078	c	13.75
a_1	-0.07389	b_1	0.01613
a_2	-0.2069	b_2	0.0252
a_3	0.0845	b_3	-0.3116
a_4	0.2382	b_4	-0.1607
a_5	0.06081	b_5	-0.1522
a_6	0.5811	b_6	-0.294
a_7	-12.6	b_7	8.2
a_8	-.07564	b_8	0.5436

将上述内容构建下图图像：

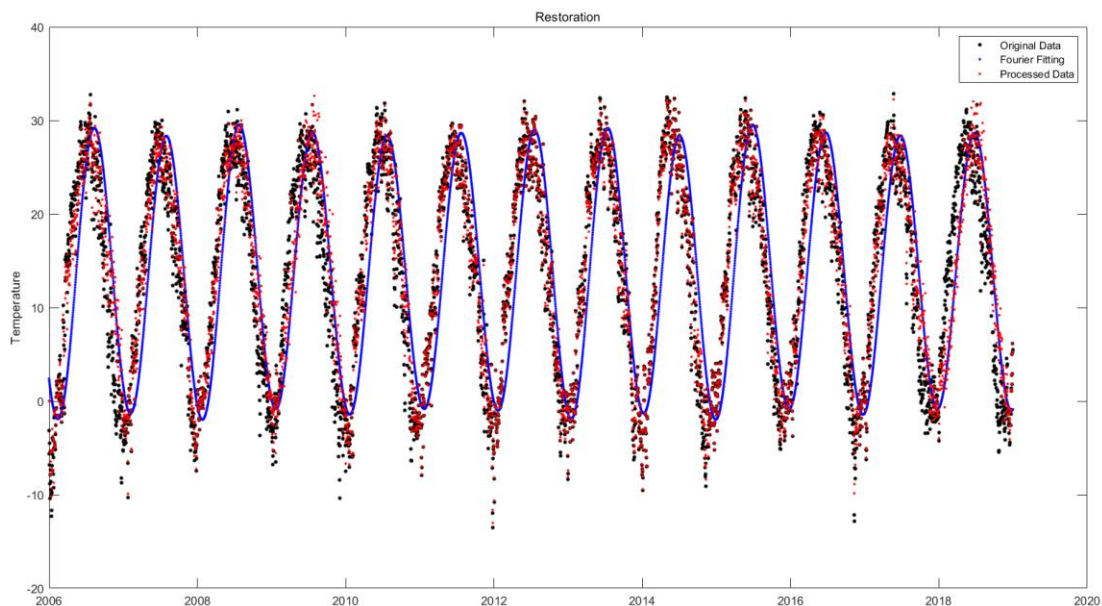


图 5 经过 SA-GM 前后处理的原时间序列及傅里叶级数对比

如图，可以看出在整体上经过处理后的时间序列与原时间序列相比，处理后的序列更加接近傅里叶级数。原来的时间序列总体上存在更多的相差幅度更大的特征，而处理后的时间序列在总体上则减弱了过强的序列特征，但同时又保留了一部分特征用于时间序列预测，因此，在预测中，经过处理后的时间序列不会过于受偏差较高的特征影响，在长远的预测中具有更好的时序性。

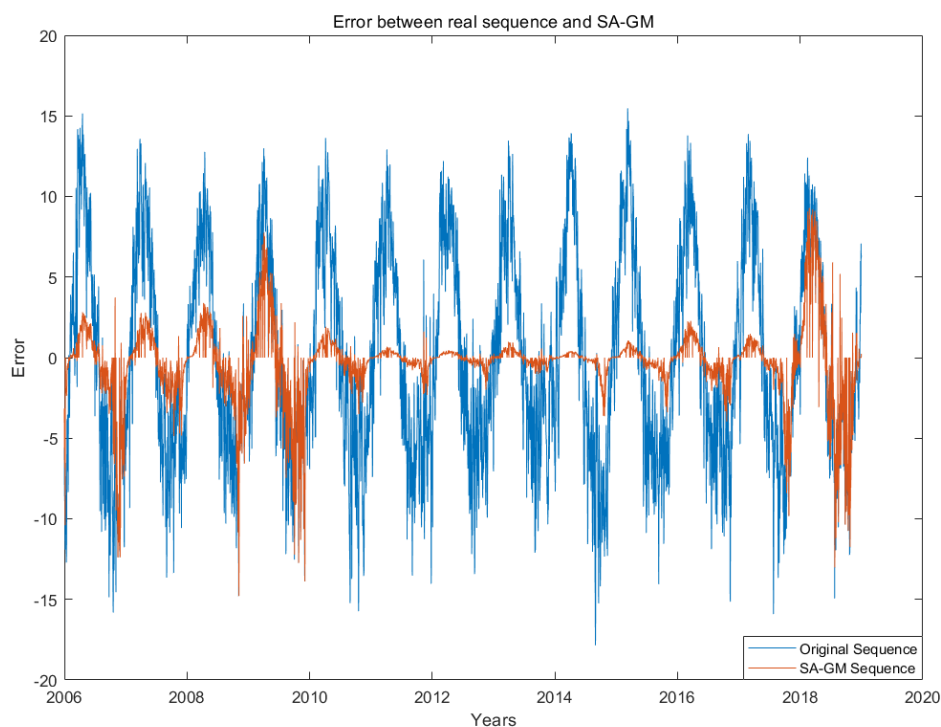


图 6 周期性时间序列与傅里叶级数间的差值

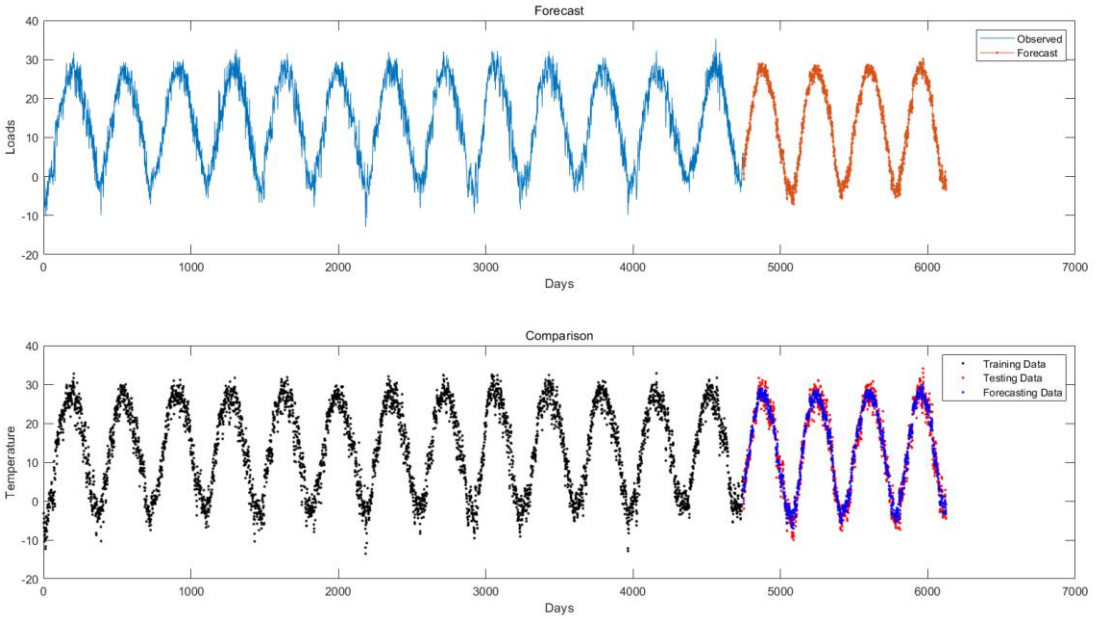
从图 6 可以看出，经过 SA-GM 修正后的序列与原序列相比，其绝对值在数值上表现更低。这意味着经过 SA-GM 修正后的时间序列更加逼近傅里叶级

数，也意味着一些过于剧烈的特征经过 SA-GM 的修正后被削弱。下面给出各自的时间序列在训练集上与傅里叶级数的相对误差。

表 4 训练集下原时间序列和经 SA-GM 处理的序列与拟合的傅里叶级数的误差

方法	MAE	SSE	MSE	RMSE	MAPE
初始	5.4035	193450.49	40.7695	6.3851	372.3611
SA-GM 修正	4.3761	135937.48	28.6482	5.3524	325.9521

从该误差可以看出，经过 SA-GM 修正以后的算法相比于原始序列更加接近于原始序列拟合的傅里叶级数，证明在数据的处理上，SA-GM 总体减少了较大误差的出现，使得原时间序列的数据集波动更为平缓，图像更为平滑。在后文介绍到，这种平缓的时序波动有利于在全局进行时间序列预测。



6.4 约束效果分析

6.5 模型对比分析

本文将傅里叶级数回归、Holt-winters、SARIMA、LSTM 分别基于日回归序列数据集进行预测，所求得的结果与本论文提出的模型的结果进行对比。

Holt-Winters模型中， $\alpha = 0.05, \beta = 0.05, \gamma = 0.05$, SARIMA 模型中，季节周期服从年周期,季节性差分数位 1,, $P = 3, Q = 3$, LSTM网络结构中，神经元数量为 288，迭代 300 次，初始 $\alpha = 0.005$ ，学习下降周期为 125，学习率下降因子为 0.2.损失函数以 MAE 为基准。通过调整参数，绘制下图不同时间序列预测算法的结果图对比。

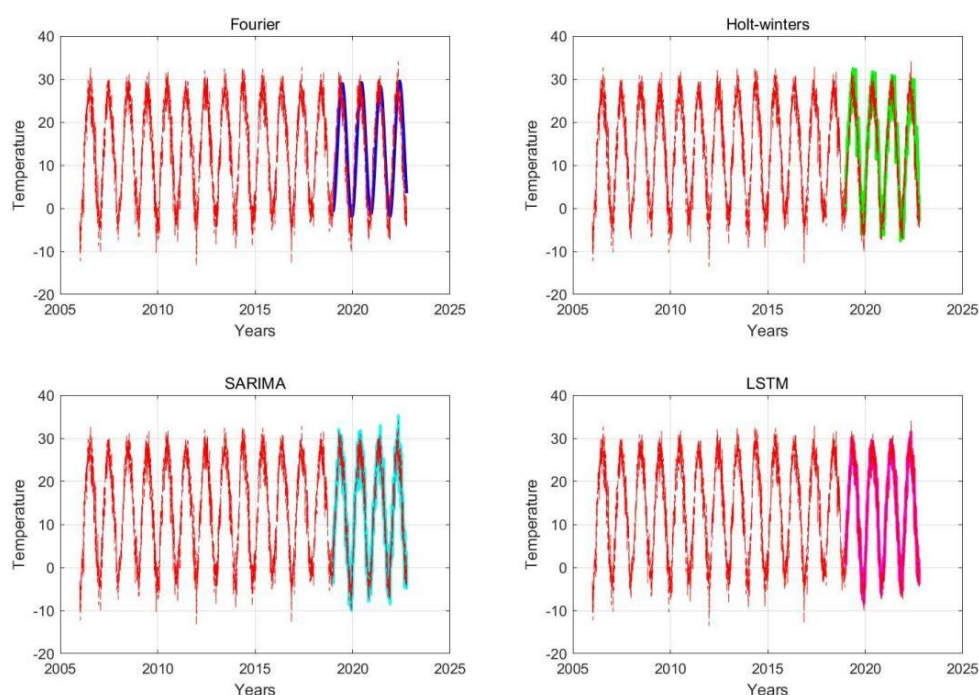


图 7 不同时间序列预测算法的预测结果比较

从上图可以看出，四种传统预测方式都能够较好地对未来的周期性时间序列进行预测。在第一幅图中，傅里叶级数对周期性时间序列的峰值和谷值的敏感性较差，当周期性时间序列处于相对极端的数值时，傅里叶级数预测无法体现，这可能是受限于傅里叶级数的函数表达式的原因。采用更高阶数的傅里叶级数可能能够解决此类现象的发生，但是也有可能陷入过拟合的情况；从第二幅图可以看出，Holt-Winters 模型对周期性时间序列的特征捕获能力略差，因此和测试数据相比较，Holt-Winters 的模型振幅较大；第三幅图和第四幅图在预测结果上更为良好，其中以 LSTM 为预测算法的第四幅图已经基本能够完全拟合周期性时间序列。

通过对原数据进行 SA-GM 算法的处理，得到了经过修改的 DAT 时间序列。实验组模型中， $a = 0.3, b = 0.1, c = 1, k = 0.3, T_0 = 1000, L = 5, \delta_1 = 0.5, \delta_2 = 0.03$,修正所采用的组合模型为 SARIMA，实验退火 13 次，经历时间窗修改数量为 100。图 8 为不同预测算法的数值对比以及效果图对比

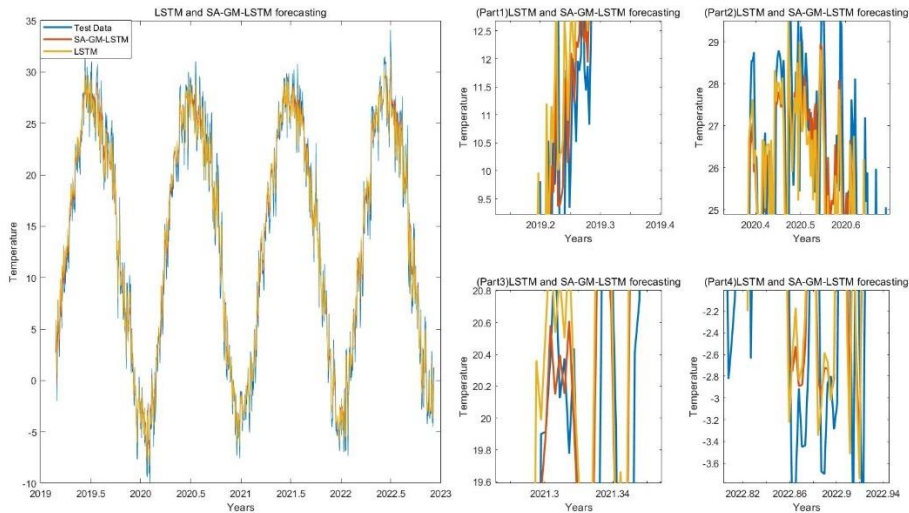


图 8 测试集中，SA-GM-LSTM 与原 LSTM 模型和测试集数据的对比

上图中，蓝色曲线为实际 DAT 测试集时间序列数据，红色曲线为 SA-GM-LSTM 通过处理 13 年的训练集数据（2006 年-2018 年）并经过 LSTM 模型预测的时间序列。黄色曲线为 DAT 训练集时间序列数据直接经过 LSTM 模型预测的时间序列。右图是根据测试集年份从 2019 年至 2022 年分别截取并放大的部分图像。从图像可以看出，总体上红色曲线相比于黄色曲线更靠近于蓝色曲线，这就是说，经过 SA-GM-LSTM 处理后的序列预测精度比传统的 LSTM 的精度更高。其中，右图第二个部分的子图在大约 2020 年 4 月的部分存在 LSTM 精度高于 SA-GM-LSTM 的精度现象，这是由于在整个正弦波序列的处理过程当中，滑动窗口对原时间序列的处理不均匀、不充分导致的，属于正常现象。

从图像中可以看出，蓝色曲线和黄色曲线存在较大的误差，在每一个周期的峰值和谷值更加体现出这种现象。而红色曲线在大多数情况下都能较好地拟合蓝色曲线。虽然在峰值和谷值处也出现了与真实的测试集有一定偏差的现象，但总体上红色曲线所对应的数值更加符合预测的要求。这证明了 SA-GM-LSTM 模型在预测的过程中优于传统的 LSTM 模型预测。

表 4 为各个预测算法的精度对比。

表 4 不同时间序列预测算法的各评价指标比较

方法	MAE	SSE	MSE	RMSE	MAPE
Holt-Winters	7.3206	102226	74.0770	8.6068	296.8287
SA-GM-Holt-Winters	6.0088	77101	55.8696	7.4746	325.2704
Fourier	5.8099	61827	44.8016	6.6934	221.9773
SARIMA	1.9168	8433.6	6.1113	2.4721	88.6367
LSTM	1.7623	7172.7	5.2052	2.2815	59.6972
SA-GM-LSTM	1.4226	4547.5	3.3000	1.8166	49.6148

从上述结果可以看出，傅里叶级数回归作为最初的回归模型，其精度有待提高，原因可能是因为傅里叶级数对应的函数对变化幅度较高的时间序列拟合效果不强。SARIMA 算法能够更好地提取季节性特征，因此其准确度高于 Holt-Winters 算法。LSTM 作为深度学习的预测方式，其精度比上述传统方法的精度更高。本文所提出的 SA-GM-LSTM 模型于实验中所提及的其他模型相比，具有最低的误差和最好的拟合程度，证明该模型在波动幅度较大的形似傅里叶级数

的周期性时间序列数据集预测中具有更好的效果。

6.6 消融实验分析

在原先的日回归实验的基础上，通过减少本算法的一些优化步骤进行消融实验，并对求得的结果进行分析，以证明算法各步骤的有效性。本文选取的消融实验对照组为三组，分别是除去弱化缓冲算子的消融实验、除去灰色预测的消融实验以及除去模拟退火的消融实验。

考虑到弱化缓冲算子是对灰色预测的优化，且总需要存在一个模型对未来进行预测。为保证实验的合理性，在除去弱化缓冲算子的对照组中，仍然保留灰色预测算法本身；在除去灰色预测算法的对照组中，将灰色预测算法替代为常规的线性回归预测算法；在除去模拟退火的算法的对照组，仍然设定一个温度的初始值，但是不进行退火的过程，也不进行迭代的操作；根据实验可以得到以下的实验结果。

表 5 消融实验中的各评价指标比较

方法	MAE	SSE	MSE	RMSE	MAPE
SA-GM-LSTM	1.4226	4547.5	3.3000	1.8166	49.6148
除去缓冲算子	1.5242	5195.2	3.7701	1.9417	45.0094
除去灰色预测	1.5529	5463.3	3.9645	1.9911	55.7611
除去模拟退火	1.6039	5866.4	4.2572	2.0633	54.6568

从实验结果可以看出，在消融实验中，去除每一个方法都会使其准确度下降。在除去缓冲算子的情况下，灰色预测因平滑性降低二导致预测震动现象严重，特征值的预测精确度下降；由于灰色预测更能鉴别系统因素之间发展趋势的相异程度，在将灰色预测算法替代为线性回归预测算法后精确度进一步下降。由于模拟退火能够在每次时间窗遍历结束以后通过预测误差判定是否需要重新进行滑动窗口遍历，故除去模拟退火算法后，滑动时间窗口只会在整个时间序列下遍历一次，效果不够显著。

7.Conclusions

为提高时间序列预测方法的预测精度，本文基于模拟退火算法和灰色预测算法提出了一种启发式算子优化 LSTM 时序预测算法。算子在设计的过程中能够避免陷入局部最优的缺陷，良好的捕捉到簇状异常噪声并进行修复，提高时间序列预测的准确度。本文对北京市 2006 年至 2022 年的地表气温数据进行了实验，表明经过优化 LSTM 处理的模型相较于传统的数学方法或者机器学习方法有着较高的预测精度，在时间序列预测中将具有广阔的应用前景。考虑到 LSTM 内部依然可以通过启发式算法（如 PSO 算法）进行输入参数的寻优，这将是下一步研究的方向。

参考文献

[1]Zeng M, Liu R, Gao M, et al. Demand forecasting for rural e-commerce logistics: a

- ~~gray prediction model based on weakening buffer operator[J]. Mobile information systems, 2022, 2022: 1-8.~~
- [2] Roda I, Macchi M. Factory-level performance evaluation of buffered multi-state production systems[J]. Journal of Manufacturing Systems, 2019, 50: 226-235.
- ~~[3] Yu Y, Si X, Hu C, et al. A review of recurrent neural networks: LSTM cells and network architectures[J]. Neural computation, 2019, 31(7): 1235-1270.~~
- ~~[4] Janiesch C, Zschech P, Heinrich K. Machine learning and deep learning[J]. Electronic Markets, 2021, 31(3): 685-695.~~
- [5] Hu Rule, Chen Yifir, Zhang Hai, et al. Bus load prediction method based on SSA-Bi-LSTM neural network[J]. Guangdong Electric Power, 2022.
- [6] Shi X, Chen Z, Wang H, et al. Convolutional LSTM network: A machine learning approach for precipitation nowcasting[J]. Advances in neural information processing systems, 2015, 28.
- [7] Hu Rule, Chen Yifir, Zhang Hai, et al. Bus load prediction method based on SSA-Bi-LSTM neural network[J]. Guangdong Electric Power, 2022.
- [8] Zhang Ajia, Wan Dingsheng. A hybrid GA-optimized LSTM-based flow prediction for small and medium-sized basins[J]. Computer Simulation, 2022.
- [9] Song G, Zhang YF, Bao FH, et al. A particle swarm optimization LSTM-based stock prediction model[J]. 2019.
- [10] Hu W, Zhang XY, Li ZE, et al. Short-term load forecasting based on optimized VMD-mRMR-LSTM model[J]. Power System Protection and Control, 2022, 50(1): 88-97.
- ~~[11] Liu Hengyong, Liu Yongli, Deng Shicong, et al. A LSTM model-based method for power load identification[J]. Electrical Measurement and Instrumentation, 2019, 56(23): 62-69.~~
- [12] Wang C, Wang Y, Zheng T, et al. Multiple load forecasting for integrated energy systems based on ResNet-LSTM network and attention mechanism[J]. Journal of Electrical Engineering Technology, 2022, 37(7): 1789-1799.
- [13] Xu Yan, Xiang Yifeng, Ma Tianxiang. Short-term power load forecasting based on EMD-CNN-LSTM hybrid model [J].
- [14] He HJ, Wang Hao, Xiao Y, et al. Non-intrusive load monitoring model based on Bi-LSTM algorithm[J]. Collection, 2019, 2.
- [15] Cui, H. R., Zhou, Y., Lu, X. F.. PCA-ARMA-EGARCH temperature prediction model and empirical analysis[J]. Mathematical Theory and Applications, 2019, 39(3): 90.
- [16] Xu Renchao, Yan Weiwu, Wang Guoliang, et al. Research on time series forecasting method and electricity price prediction based on periodic modeling[J]. Journal of Automation, 2020, 46(6): 1136-1144.
- [17] Teng J, Wu T, Yang L. Prediction Model of High-Speed Railway's Passenger Flow Seat Structure Based on Improved LSTM[M]//CICTP 2022. 2022: 1024-1033.
- ~~[18] Bao W, Yue J, Rao Y. A deep learning framework for financial time series using stacked autoencoders and long-short term memory[J]. PloS one, 2017, 12(7): e0180944.~~
- ~~[19] Ronning F. The role of Fourier series in mathematics and in signal theory[J].~~

~~International Journal of Research in Undergraduate Mathematics Education, 2021, 7(2): 189-210.~~

- [20]Mudassir M, Bennbaia S, Unal D, et al. Time-series forecasting of Bitcoin prices using high-dimensional features: a machine learning approach[J]. Neural computing and applications, 2020: 1-15.
- [21]Tong M, Dong J, Luo X, et al. Coal consumption forecasting using an optimized grey model: The case of the world's top three coal consumers[J]. Energy, 2022, 242: 122786.
- [22]Sherstinsky A. Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network[J]. Physica D: Nonlinear Phenomena, 2020, 404: 132306.
- [23]~~Staudemeyer R C, Morris E R. Understanding LSTM—a tutorial into long short-term memory recurrent neural networks[J]. arXiv preprint arXiv:1909.09586, 2019.~~
- [24]Ding G, Qin L. Study on the prediction of stock price based on the associated network model of LSTM[J]. International Journal of Machine Learning and Cybernetics, 2020, 11: 1307-1317.
- [25]~~Ghannadi P, Kourehli S S, Mirjalili S. A review of the application of the simulated annealing algorithm in structural health monitoring (1995-2021)[J]. Frattura ed Integrità Strutturale, 2023, 17(64): 51-76.~~
- [26]Chen J, Wu Z. A positive real order weakening buffer operator and its applications in grey prediction model[J]. Applied Soft Computing, 2021, 99: 106922.
- [27]von Sachs R. Nonparametric spectral analysis of multivariate time series[J]. Annual Review of Statistics and Its Application, 2020, 7: 361-386.
- [28]~~Han X, Dong Y, Yue L, et al. State transition simulated annealing algorithm for discrete continuous optimization problems[J]. IEEE Access, 2019, 7: 44391-44403.~~
- [29]~~Al Bataineh A, Kaur D. Immunocomputing-based approach for optimizing the topologies of LSTM networks[J]. IEEE Access, 2021, 9: 78993-79004.~~
- [30]Chen L, Xu G, Zhang S, et al. Health indicator construction of machinery based on end-to-end trainable convolution recurrent neural networks[J]. Journal of Manufacturing Systems, 2020, 54: 1-11.
- [31]Nath P, Saha P, Middya A I, et al. Long-term time-series pollution forecast using statistical and deep learning methods[J]. Neural Computing and Applications, 2021: 1-20.