

DOI: 10. 13382/j. jemi. B2306407

# 融合两阶段分解与 iJaya-ELM 的短期风速预测模型<sup>\*</sup>

王逸文 王维莉 刘贤超 胡炜琴  
(上海海事大学物流研究中心 上海 201306)

**摘 要:**准确的预测风速对于风电场的安全运行和高效发电具有重要意义。针对已有文献在风速预测问题中采用的单一分解策略存在固有缺陷,优化预测模型效果不稳定等问题,提出了一种融合两阶段分解与 iJaya-ELM 的混合预测模型。首先,对原始风速序列进行 ICEEMDAN 分解,得到 12 个分量后基于排列熵值重构为高频项、中频项与低频项;随后对高频项进行奇异谱分解滤去序列噪声;提出一种改进的 Jaya 算法 iJaya,利用 iJaya 算法获取极限学习机 ELM 的最优连接权值与阈值,最后将各个分量的预测结果线性集成得到最终结果。以我国甘肃地区风电场风速数据进行模型验证,并利用新疆地区数据集测试其鲁棒性与通用性。实验结果表明,iJaya 算法具有较强的寻优精度与稳定性,两阶段分解能够深度挖掘风速序列的特征;该混合模型能够有效提升风速预测精度,平均绝对误差与均方误差分别为 0.067 9 和 0.134 5。

**关键词:** 风速预测;ICEEMDAN;奇异谱分析;Jaya 算法;极限学习机;两阶段分解

**中图分类号:** TM614      **文献标识码:** A      **国家标准学科分类代码:** 470.40

## Two-stage decomposition and iJaya-ELM short-term wind speed prediction model

Wang Yiwen Wang Weili Liu Xianchao Hu Weiqin  
(Logistics Research Center, Shanghai Maritime University, Shanghai 201306, China)

**Abstract:** Accurate prediction of wind speed is of great significance for safe operation and efficient power generation of wind farms. Aiming at the inherent defects of the single decomposition strategy used in existing literatures in wind speed prediction and the unstable effect of the optimized prediction model, a hybrid prediction model combining two-stage decomposition and iJaya-ELM is proposed. First, ICEEMDAN decomposition is performed on the original wind speed sequence, and 12 components are obtained, and reconstructed into high frequency terms, middle frequency terms and low frequency terms based on the permutation entropy. Then, the high frequency term is filtered by singular spectrum decomposition to remove the sequence noise. An improved Jaya algorithm, iJaya, is proposed to obtain the optimal connection weights and thresholds of ELM. Finally, the predictive results of each component are linearly integrated to obtain the final results. The model is validated by wind speed data of wind farm in Gansu province of China, and its robustness and universality are tested by wind speed data of Xinjiang region. The experimental results show that the iJaya algorithm is of strong optimization accuracy and stability, and the two-stage decomposition can deeply excavate the characteristics of wind speed series. The hybrid model can effectively improve the wind speed prediction accuracy, and the average absolute error and mean square error are 0.067 9 and 0.134 5, respectively.

**Keywords:** wind speed prediction; ICEEMDAN; singular spectrum analysis; Jaya algorithm; extreme learning machine; two-stage decomposition

## 0 引言

近年来,随着世界各国对环境保护的重视,可再生能源占总消耗能源的比重不断增加,风能作为一种绿色、清洁能源得到了广泛关注。据国际能源署(IEA)预测,到2030年全球风力发电将增长两倍以上,占到总装机容量的20%以上。在我国,风能发电已成为可再生能源的重要组成部分,截至2022年底,全国风力发电装机容量已经达到了365.4 GW,同比增长11.2%。在风能发电中,风速变化是影响发电效率和经济性的重要因素之一。因此,准确预测风速对于风电场的安全运行和高效发电具有重要意义。

在过去的几十年里,许多研究者已经尝试使用各种技术和方法来预测风速。传统的风速预测方法主要包括物理方法<sup>[1]</sup>、统计方法<sup>[2]</sup>、时序方法<sup>[3]</sup>等。然而,由于风速具有时变特征<sup>[4]</sup>,以及受到气象条件、地形等多种因素的影响,且短期预测相比超短期预测规律性更弱、非平稳性更强,传统的预测方法具有较多限制。随着人工智能技术的发展,机器学习与信号分解方法逐渐受到了研究者的关注,机器学习方法具有较好的非线性拟合能力,如文献[5]在考虑风速属性简约的基础上,建立双层长短时记忆网络(long short-term memory, LSTM)短期风速预测模型,取得了较高的预测精度;文献[6]通过提取风速序列统计特征,结合极限梯度提升算法(extreme gradient boosting, XGboost)进行预测建模;文献[7]提出了一种改进Stacking算法的短期风速组合预测方法,并利用改进萤火虫算法对模型关键参数进行优化,但该优化算法控制参数较多且对结果影响较大;文献[8]提出了一种基于Jaya算法优化支持向量机(support vector regression, SVR)的风速预测方法,但传统Jaya算法寻优精度相对较低。而信号分解方法可继续细分为两种:分解去噪策略和分解集成策略<sup>[9]</sup>,前者是指序列分解后将结果中的高频项或残差项当作噪声去除,从而提取序列有效信息,如文献[10]利用奇异谱分解(singular spectrum analysis, SSA)方法去噪;后者则是将数据分解为不同特征的分量,直接对各分量分别建立模型预测再集成,如文献[11]基于自适应噪声完备集合经验模态分解(complete EEMD with adaptive noise, CEEMDAN)对风速序列进行分解,最后叠加各子序列预测结果;文献[12]基于变分模态分解(variational mode decomposition, VMD)分解对各分量建立最小二乘支持向量机模型进行预测;文献[13]采用局部均值分解(local mean decomposition, LMD)方法将原始风速序列分解为若干个平稳的风速子序列后进行预测叠加。当前研究将上述信号分解与机器学习方法相结合,证明其能够有效提升预测精度,但单一

分解策略无法充分提取数据特征,且部分分解方法存在模态混叠、端点效应等固有缺陷,因此该类方法有待进一步研究。

在实际预测过程中,仍有两个问题值得关注。一是风速的变化不仅受气候和环境等自然因素的影响,还受到各种外界因素和噪声的干扰,如风机振动、建筑物遮挡、地形地貌等因素。这些因素会使得风速的预测变得复杂和困难,况且利用附带噪声的数据去预测未来数据易导致误差增大,进而影响预测效果。二是预测模型的稳定性对发电安全较为重要<sup>[14]</sup>,而已有研究在预测模型的超参数优化方面,存在优化算法控制参数不确定的情况。例如,粒子群算法的两个学习因子会极大影响其全局寻优能力与局部搜索能力,进而导致预测效果不稳定、模型适用性较差等问题。

因此,为提高风速预测精度,本文从挖掘风速数据有效特征和优化预测模型性能两方面出发,提出了一种融合两阶段分解与 iJaya-ELM 的预测模型。在分解方法选择方面,本文将分解集成策略与分解去噪策略相结合,采用改进的自适应噪声完备集合经验模态分解(improved complete eemd with adaptive noise, ICEEMDAN)与奇异谱分析方法,有效降低模态混叠、端点效应的同时能够滤去序列中的高频噪声和非周期性的异常现象,并利用排列熵进行复杂度分析,从而最大程度保留分量中的有效信号;在预测方法方面,本文提出一种改进的Jaya(improved-Jaya, iJaya)算法优化极限学习机(extreme learning machine, ELM)模型的超参数,提高预测精度的同时减少优化算法控制参数难以确定的现象。最后,以我国甘肃地区的风速数据为例,验证了本文所提模型的有效性,并利用我国新疆地区的风速数据,进一步验证本文模型的鲁棒性。

## 1 理论与方法

### 1.1 ICEEMDAN

传统模态分解类方法大多存在模态混叠、白噪声传递等缺陷<sup>[15]</sup>,而ICEEMDAN方法能够有效解决上述问题<sup>[16]</sup>,且相比于CEEMDAN算法残余噪声更小、物理意义更强,对于非平稳序列有着更强的特征提取能力。因此,本文采用ICEEMDAN对指数进行一次分解,其步骤如下:

1) 对原始指数序列  $x$  添加  $i$  组白噪声  $\omega^{(i)}$ , 得到序列  $x^{(i)}$ :

$$x^{(i)} = x + \beta_0 E_1(\omega^{(i)}) \quad (1)$$

2) 对  $x^{(i)}$  进行包络计算并求平均值得到第1组残差  $r_1$ , 并计算第1个模态分量  $IMF_1$ :

$$r_1 = \frac{1}{I} \sum_{i=1}^I N(x^{(i)}) \quad (2)$$

$$IMF_1 = x - r_1 \quad (3)$$

3) 继续在第 1 组残差中添加白噪声, 构造序列计算第 2 组残差  $r_2$ , 并计算第 2 个模态  $IMF_2$ :

$$r_2 = \frac{1}{I} \sum_{i=1}^I N(r_1 + \beta_1 E_2(\omega^{(i)})) \quad (4)$$

$$IMF_2 = r_1 - r_2 \quad (5)$$

4) 重复步骤 3), 计算第  $k$  组残差  $r_k$  和第  $k$  个模态分量  $IMF_k$ :

$$r_k = \frac{1}{I} \sum_{i=1}^I N(r_{k-1} + \beta_{k-1} E_k(\omega^{(i)})) \quad (6)$$

$$IMF_k = r_{k-1} - r_k \quad (7)$$

5) 重复上述步骤, 直到残差不可分解或达到最大迭代次数  $K$ 。

其中,  $\beta_k$  为每次控制白噪声能量的参数 ( $k = 1, 2, \dots, K$ ),  $K$  为设置的最大迭代次数;  $E_k(\cdot)$  为通过 EMD 算法获取的第  $k$  个模态;  $N(\cdot)$  为包络计算。

在实际预测过程中, 过多的分量显著增大了预测计算量, 且不利于高频分量的判定与后续两阶段分解, 因此本文引入样本熵计算各个分量序列复杂度。

## 1.2 奇异谱分析 (SSA)

SSA 通过构造在时间序列上的特定矩阵的奇异值分解 (singular value decomposition, SVD), 从而提取出时序数据中的不同成分序列, 如趋势、周期、噪声等, 它既不需要假设参数模型, 也不需要假设平稳性条件, 因而具有非常广泛的适用性<sup>[17]</sup>。鉴于其良好的去噪能力和鲁棒性, 本文在第 2 阶段分解使用奇异谱分析去噪, 其步骤为:

1) 嵌入。选择合适的窗口长度  $L$  (一般取  $L < N/2$ , 可根据经验确定) 将原始时间序列进行滞后排列得到轨迹矩阵, 并令  $K = N - L + 1$ , 构造出  $L \times K$  阶的轨迹矩阵, 如式 (8) 所示:

$$X = \begin{bmatrix} x_1 & x_2 & \cdots & x_K \\ x_2 & x_3 & \cdots & x_{K+1} \\ \vdots & \vdots & \ddots & \vdots \\ x_L & x_{L+1} & \cdots & x_N \end{bmatrix} \quad (8)$$

2) 分解。对轨迹矩阵进行 SVD 处理, 由于直接对轨迹矩阵分解比较困难, 因此按照式 (9) 计算轨迹矩阵的协方差矩阵:

$$S = XX^T \quad (9)$$

接着对  $S$  进行特征值分解得到特征值  $\lambda_1 > \lambda_2 > \dots > \lambda_L \geq 0$  和对应的特征向量  $U = [U_1, U_2, \dots, U_L]$ ,  $\lambda_1 > \lambda_2 > \dots > \lambda_L \geq 0$  为原序列的奇异谱, 并且有:

$$X = \sum_{m=1}^L \sqrt{\lambda_m} U_m V_m^T, V_m = \frac{X^T U_m}{\sqrt{\lambda_m}}, m = 1, 2, \dots, L \quad (10)$$

其中,  $\lambda_i$  对应的特征向量  $U_i$  反映了时间序列的演变型, 称为时间经验正交函数 (T-EOF)。

3) 分组。将下标集合  $\{1, 2, \dots, m\}$  划分成  $M$  个互不相交的子集  $I_1, I_2, \dots, I_M$ , 令  $I = \{i_1, i_2, \dots, i_p\}$ , 则对应于  $I$  的合成矩阵有  $X_I = X_{i1} + X_{i2} + \dots + X_{ip}$ , 同时有:

$$X = X_{I1} + X_{I2} + \dots + X_{IM} \quad (11)$$

4) 重构。将分组矩阵  $X_I = (y_{ij})_{L \times K}$  用对角平均法转换为所对应的长度为  $T$  的序列  $RC_{Ii} = (rc_1, rc_2, \dots, rc_T)$ ,  $RC_{Ii}$  中第  $k$  个元素为矩阵  $X_I$  中满足  $i + j = k + 1$  的所有元素的均值。具体操作如式 (12) 所示:

$$X_I = \begin{cases} \frac{1}{k} \sum_{m=1}^k y_{m, k-m+1}, 1 \leq k < L \\ \frac{1}{L} \sum_{m=1}^L y_{m, k-m+1}, L \leq k < K \\ \frac{1}{T-k+1} \sum_{m=k-K+1}^{T-k+1} y_{m, k-m+1}, K \leq k \leq T \end{cases} \quad (12)$$

## 1.3 Jaya 算法

Jaya 算法是由 Rao 等<sup>[18]</sup>于 2016 年提出的一种新型元启发式算法, 该算法通过持续改进的思想, 不断向最优个体靠近的同时远离最差个体, 从而不断提高解的质量。相比于其他智能优化算法, Jaya 算法具有控制参数少、全局搜索能力强等特点, 被广泛应用于车间调度、参数优化领域。该算法的迭代公式为:

$$X'_{i,j,k} = X_{i,j,k} + r_1(X_{i,best,k} - |X_{i,j,k}|) - r_2(|X_{i,worst,k} - |X_{i,j,k}||) \quad (13)$$

其中,  $X_{i,j,k}$  代表个体  $j$  的第  $i$  个变量在第  $k$  次迭代中的值,  $r_1$  与  $r_2$  表示两个取值范围为  $[0, 1]$  的随机变量;  $X'_{i,j,k}$  代表更新后个体  $j$  的第  $i$  个变量在第  $k$  次迭代中的值;  $X_{i,best,k}$  代表最优个体在第  $k$  次迭代中的第  $i$  个变量值;  $X_{i,worst,k}$  则代表最差个体在第  $k$  次迭代中的第  $i$  个变量值。

## 1.4 极限学习机 (ELM)

ELM 算法通过随机选取网络的输入权值和隐层偏置, 通过解析计算得到输出权值, 解决传统神经网络中学习速率慢、迭代时间长、学习参数需要人为提前设置等问题<sup>[19]</sup>。假设有一组训练样本  $(x_i, y_i)$ ,  $\mathbf{x}_i = (x_{i1}, x_{i2}, \dots, x_{in})^T \in \mathbb{R}^n$  为网络的样本输入向量,  $\mathbf{y}_i = (y_{i1}, y_{i2}, \dots, y_{im})^T \in \mathbb{R}^m$  为网络的输出向量, 标准的带  $L$  个隐层神经元的 ELM 的一般形式如式 (14) 所示。

$$f_L(x) = \sum_{i=1}^L \beta_i G_i(x) = \sum_{i=1}^L \beta_i G_i(a_i, b_i, X) \quad (14)$$

式中:  $\beta_i$  为输出权重,  $G_i(a_i, b_i, X)$  为激活函数,  $a_i$  为网络的输入权重,  $b_i$  为第  $i$  个隐层的单元阈值。然而, ELM 输入层与隐含层的连接权值与阈值会影响其预测精度, 因此本文提出一种改进 Jaya 算法对这两个参数进行优化。



## 2 ICEEMDAN-SSA-iJaya-ELM 混合模型构建

### 2.1 改进 Jaya 算法

尽管标准 Jaya 算法趋利避害的特点能够适应工程问题中全局优化的需求和思想,但在收敛速度、局部搜索与寻优精度等方面仍存在着一定的缺陷。针对上述问题,为了进一步提高算法寻优精度与收敛速度,本文对其作了两方面改进。

#### 1) 基于正弦混沌因子的迭代方式

标准 Jaya 算法在更新公式中采用两个取值范围为  $[0, 1]$  随机变量  $r_1, r_2$  进行迭代,尽管其机制简单,但遇到复杂问题时,随机变量的随机性与盲目性无法保证其平衡全局搜索与局部寻优的能力,易导致其前期全局搜索不充分、后期精细挖掘能力不强等问题。而混沌因子能较好地解决这一问题,混沌行为是一种非线性随机现象,当其作为扰动因子时,能够使算法的迭代过程更为平滑,在前期加快搜索进度,后期注重局部搜索,较好地平衡全局搜索与局部寻优,进而提高寻优精度。为此,本文引入一种正弦混沌因子,替代原始的随机变量,其公式更新为:

$$X'_{i,j,k} = X_{i,j,k} + sn_i(X_{i,best,k} - |X_{i,j,k}|) - sn_2(X_{i,worst,k} - |X_{i,j,k}|) \quad (15)$$

其中,  $sn_i$  表示第  $i$  次迭代过程产生的正弦混沌数。

#### 2) 以最优解为牵引的扰动机制

在标准 Jaya 算法中,种群的最优个体与最差个体发挥了重要作用,其依靠最优个体与最差个体的牵引不断更新。然而,此种更新方式存在一定的局限性,当种群的最优个体与最差个体陷入局部最优时,易出现搜索停滞的现象,从而无法获得更优解。为了避免无效搜索与“早熟”收敛,本文提出了一种以最优解为牵引的扰动机制。当算法在迭代过程中出现连续 5 次没有更新解的情况,则对其采取一种扰动机制,增强其跳出局部最优的概率,从而更好地寻找全局最优解,具体公式为:

$$X'_{new} = \frac{X_{new} + X_{best} * \omega}{2} \quad (16)$$

其中,  $X'_{new}$  表示经过扰动机制后新产生的个体;  $X_{new}$  表示迭代过程中连续 5 次没有进行更新的个体;  $X_{best}$  为当前迭代过程中的最优解;  $\omega$  为权重因子,表示受最优解牵引的程度。为了避免权重选取的盲目性,本文采用一种线性动态递减的权重因子,其在算法迭代前期能够获得较大值,提升全局搜索能力;在迭代后期获得较小值,提升局部寻优能力,其公式为:

$$\omega = \frac{0.32 \times (\text{Max}_{iter} - iter)}{\text{Max}_{iter}} \quad (17)$$

为了验证本文改进 Jaya 算法的性能,本文选取 6 种常用的标准测试函数进行测试,包括 Sphere 函数(F1,最

小值为 0)、Schwefel 2.22 函数(F2,最小值为 0)、Schwefel 2.21 函数(F3,最小值为 0)、Quartic 函数(F4,最小值为 0)、Rastrigin 函数(F5,最小值为 0)、Ackley 函数(F6,最小值为 0)。同时,选取标准 Jaya 算法、粒子群算法(particle swarm optimization, PSO)、灰狼优化算法(grey wolf optimizer, GWO)、鲸鱼优化算法(whale optimization algorithm, WOA)作为对比算法。考虑到不同优化算法求解具有一定的随机性,本文设置所有算法种群规模为 30,最大迭代次数为 500,并将每种算法独立运行 10 次,取每种算法的最优值、最差值、平均值与标准差,以此衡量各种优化算法的优劣。本文实验在 Window 10 x64 操作系统下进行,CPU 为 Intel(R) Core(TM) i7-8700 @ 3.20 GHz,内存为 8 GB,GPU 为 NVIDIA GeForce GTX 1060 6 GB,算法在 MATLAB R2020a 进行仿真实验。得到的结果如表 1 所示。

由表 1 可知,标准 Jaya 算法尽管其控制参数较少,但在寻优精度上总体显著劣于灰狼优化算法与鲸鱼算法,仅在部分函数上优于粒子群算法。而本文所提改进 Jaya 算法在上述 4 种测试函数中均取得最优结果,其中在函数 F5 上已取得理论最优值,说明该算法相比于标准 Jaya 算法性能显著提升,且有较高的寻优精度。在搜索稳定性方面,本文算法的标准差均优于其他 4 种优化算法,表明该算法的稳定性较好。综上,本文将所提 iJaya 算法用于优化 ELM 的连接权值与阈值,以进一步提升 ELM 算法的预测能力。

### 2.2 混合预测模型

基于前文所述各算法的基本原理,本文提出的 ICEEMDAN-SSA-iJaya-ELM 混合预测模型,其具体步骤为:

1) 第 1 阶段分解。首先基于 ICEEMDAN 对原始风速序列进行第 1 阶段分解,得到若干个不同复杂度的分量。

2) 分量重构。为降低计算量与复杂度,采用 SE 算法计算各个分量的熵值,随后根据熵值将各分量重构为高频项、中频项、低频项。

3) 第 2 阶段分解。对于高频项,采用 SSA 算法进行第 2 阶段分解得到若干分量,计算各分量的特征值与方差贡献率,根据特征值去除噪声,得到去噪后的高频项,从而最大程度保留高频项的有效信号。

4) 超参数优化。基于 iJaya 算法的全局搜索能力,对各个分量的 ELM 的连接权值与阈值进行寻优。

5) 预测。利用优化后的 ELM 对去噪后的高频项、中频项、低频项分别进行建模预测,最后将各项得到的预测值线性求和形成最终预测结果。

6) 结果评价。选择多种评价指标对模型的预测结果进行评价,以此衡量各模型的优劣性。

表 1 测试函数结果对比  
Table 1 Comparison of test function results

| 测试函数 | 算法    | 最优值                          | 最差值                           | 平均值                           | 标准差                           |
|------|-------|------------------------------|-------------------------------|-------------------------------|-------------------------------|
| F1   | iJaya | <b>0.00×10<sup>0</sup></b>   | <b>7.99×10<sup>-158</sup></b> | <b>2.57×10<sup>-157</sup></b> | <b>3.35×10<sup>-158</sup></b> |
|      | Jaya  | 2.01×10 <sup>0</sup>         | 2.19×10 <sup>0</sup>          | 8.96×10 <sup>0</sup>          | 4.36×10 <sup>0</sup>          |
|      | PSO   | 1.12×10 <sup>3</sup>         | 1.02×10 <sup>3</sup>          | 4.69×10 <sup>3</sup>          | 2.31×10 <sup>3</sup>          |
|      | GWO   | 3.92×10 <sup>-29</sup>       | 5.44×10 <sup>-28</sup>        | 1.64×10 <sup>-27</sup>        | 6.60×10 <sup>-28</sup>        |
|      | WOA   | 5.83×10 <sup>-81</sup>       | 3.16×10 <sup>-71</sup>        | 1.00×10 <sup>-70</sup>        | 1.00×10 <sup>-71</sup>        |
| F2   | iJaya | <b>0.00×10<sup>0</sup></b>   | <b>1.15×10<sup>-77</sup></b>  | <b>3.64×10<sup>-77</sup></b>  | <b>4.80×10<sup>-78</sup></b>  |
|      | Jaya  | 6.01×10 <sup>-1</sup>        | 4.70×10 <sup>-1</sup>         | 2.09×10 <sup>0</sup>          | 1.35×10 <sup>0</sup>          |
|      | PSO   | 1.06×10 <sup>1</sup>         | 4.34×10 <sup>0</sup>          | 2.59×10 <sup>1</sup>          | 1.81×10 <sup>1</sup>          |
|      | GWO   | 2.41×10 <sup>-17</sup>       | 6.61×10 <sup>-17</sup>        | 2.31×10 <sup>-16</sup>        | 1.06×10 <sup>-16</sup>        |
|      | WOA   | 2.23×10 <sup>-56</sup>       | 8.26×10 <sup>-51</sup>        | 2.54×10 <sup>-50</sup>        | 4.76×10 <sup>-51</sup>        |
| F3   | iJaya | <b>0.00×10<sup>0</sup></b>   | <b>1.41×10<sup>-84</sup></b>  | <b>4.42×10<sup>-84</sup></b>  | <b>8.48×10<sup>-85</sup></b>  |
|      | Jaya  | 1.30×10 <sup>1</sup>         | 2.98×10 <sup>0</sup>          | 2.13×10 <sup>1</sup>          | 1.78×10 <sup>1</sup>          |
|      | PSO   | 2.01×10 <sup>1</sup>         | 5.24×10 <sup>0</sup>          | 3.67×10 <sup>1</sup>          | 3.05×10 <sup>1</sup>          |
|      | GWO   | 1.23×10 <sup>-7</sup>        | 3.17×10 <sup>-6</sup>         | 1.05×10 <sup>-5</sup>         | 1.73×10 <sup>-6</sup>         |
|      | WOA   | 3.02×10 <sup>0</sup>         | 3.13×10 <sup>1</sup>          | 9.19×10 <sup>1</sup>          | 5.36×10 <sup>1</sup>          |
| F4   | iJaya | <b>3.42×10<sup>-6</sup></b>  | <b>8.26×10<sup>-5</sup></b>   | <b>2.72×10<sup>-4</sup></b>   | <b>8.05×10<sup>-5</sup></b>   |
|      | Jaya  | 4.93×10 <sup>-2</sup>        | 4.62×10 <sup>-2</sup>         | 1.97×10 <sup>-1</sup>         | 1.21×10 <sup>-1</sup>         |
|      | PSO   | 5.24×10 <sup>-1</sup>        | 3.53×10 <sup>-1</sup>         | 1.75×10 <sup>0</sup>          | 8.55×10 <sup>-1</sup>         |
|      | GWO   | 8.92×10 <sup>-4</sup>        | 9.19×10 <sup>-4</sup>         | 3.49×10 <sup>-3</sup>         | 1.95×10 <sup>-3</sup>         |
|      | WOA   | 4.28×10 <sup>-5</sup>        | 2.91×10 <sup>-3</sup>         | 8.92×10 <sup>-3</sup>         | 4.33×10 <sup>-3</sup>         |
| F5   | iJaya | <b>0.00×10<sup>0</sup></b>   | <b>0.00×10<sup>0</sup></b>    | <b>0.00×10<sup>0</sup></b>    | <b>0.00×10<sup>0</sup></b>    |
|      | Jaya  | 2.14×10 <sup>2</sup>         | 1.71×10 <sup>1</sup>          | 2.68×10 <sup>2</sup>          | 2.34×10 <sup>2</sup>          |
|      | PSO   | 7.18×10 <sup>1</sup>         | 2.14×10 <sup>1</sup>          | 1.48×10 <sup>2</sup>          | 1.03×10 <sup>2</sup>          |
|      | GWO   | 5.68×10 <sup>-14</sup>       | 2.09×10 <sup>0</sup>          | 5.05×10 <sup>0</sup>          | 1.28×10 <sup>0</sup>          |
|      | WOA   | 0.00×10 <sup>0</sup>         | 0.00×10 <sup>0</sup>          | 0.00×10 <sup>0</sup>          | 0.00×10 <sup>0</sup>          |
| F6   | iJaya | <b>8.88×10<sup>-16</sup></b> | <b>8.88×10<sup>-16</sup></b>  | <b>8.88×10<sup>-16</sup></b>  | <b>0.00×10<sup>0</sup></b>    |
|      | Jaya  | 1.65×10 <sup>0</sup>         | 4.66×10 <sup>-1</sup>         | 3.07×10 <sup>0</sup>          | 2.41×10 <sup>0</sup>          |
|      | PSO   | 1.06×10 <sup>1</sup>         | 8.99×10 <sup>-1</sup>         | 1.33×10 <sup>1</sup>          | 1.14×10 <sup>1</sup>          |
|      | GWO   | 7.55×10 <sup>-14</sup>       | 1.73×10 <sup>-14</sup>        | 1.22×10 <sup>-13</sup>        | 1.00×10 <sup>-13</sup>        |
|      | WOA   | 8.88×10 <sup>-16</sup>       | 3.11×10 <sup>-15</sup>        | 7.99×10 <sup>-15</sup>        | 4.80×10 <sup>-15</sup>        |

3 实证分析

3.1 数据描述

选用我国甘肃某地区风电场 2019 年 2 月 1 日~2019 年 2 月 28 日风速数据进行分析,该风电场每隔 5 min 进行一次数据采集,共计 8 064 个数据点。原始风速序列如图 1 所示。由图 1 可知,原始风速序列呈现明显的非线性、不规则波动特征,且存在较多的噪声。为验证模型有效性,本文选取前 7 258 条为训练集,后 806 条为测试集。

3.2 数据分解与重构

为进一步挖掘数据内在波动趋势,首先采用 ICEEMDAN 分解对原始风速序列进行分解。设置 ICEEMDAN 最大迭代次数为 1 000,添加白噪声次数为 500,白噪声标准差为 0.2,得到的结果如图 2 所示。由图 2 可知,数据经 ICEEMDAN 分解后得到了 12 个分量,依次由高频向低频变化。

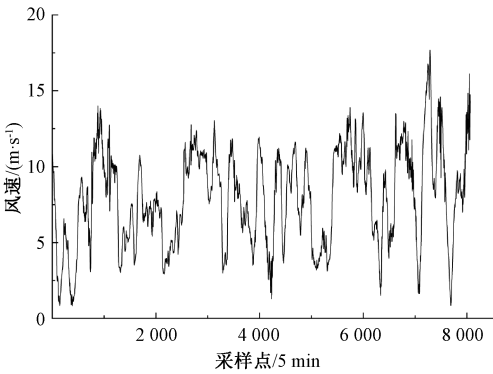


图 1 原始风速序列数据(甘肃)

Fig. 1 Original wind speed sequence data (Gansu)

随后基于排列熵原理计算各分量熵值。熵值大小反映了各分量的复杂程度,熵值越大,说明该分量越复杂。表 2 展示了 12 个分量的样本熵计算结果。由表 2 可知,前 4 个分量熵值远远大于其他分量,复杂度较高;分量 5~6 的熵值在 0.45~0.7,复杂度中等;其余分量熵值均

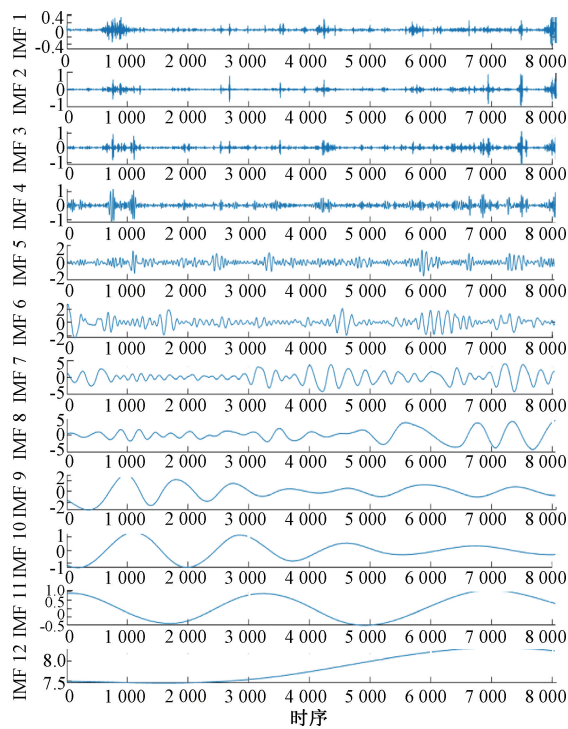


图2 ICEEMDAN 分解结果

Fig.2 ICEEMDAN decomposition results

小于 0.45,复杂度较低。结合以往文献对于熵值高低的判断<sup>[20]</sup>,本文进一步将前 4 个分量作为高频项,体现短期风速序列的不规则波动,对整体趋势影响较小;第 5~6 个分量作为中频项,体现风速序列的周期性特征,对整体趋势影响一般;第 7~12 个分量作为低频项,反映出原始风速序列的趋势特性。重构后的分量如图 3 所示。

表 2 各分量排列熵值

| Table 2 Permutation entropy of each component |           |       |           |
|-----------------------------------------------|-----------|-------|-----------|
| 分量                                            | 排列熵值      | 分量    | 排列熵值      |
| IMF1                                          | 0.978 787 | IMF7  | 0.418 154 |
| IMF2                                          | 0.872 782 | IMF8  | 0.403 958 |
| IMF3                                          | 0.719 525 | IMF9  | 0.393 660 |
| IMF4                                          | 0.602 887 | IMF10 | 0.391 992 |
| IMF5                                          | 0.510 836 | IMF11 | 0.387 652 |
| IMF6                                          | 0.454 096 | IMF12 | 0.350 730 |

由图 3 可知,高频分量噪声较多,若直接预测较为困难。为了进一步提升高频项预测效果,在第 2 阶段分解中,本文对重组后的高频项进行奇异谱分解去噪。一般来说,过小的窗口长度无法提取主要信息,而过大的窗口长度会增加模型的复杂度。在本研究中,高频项序列较为复杂,且原序列的主要成分通常在前几个分量之中,因此嵌入窗口长度设置不宜过多。经多次调试,将嵌入窗口长度设置为 10,得到 10 个序列分量,如图 4 所示。随后计算其特征值与方差贡献率,如表 3 所示。

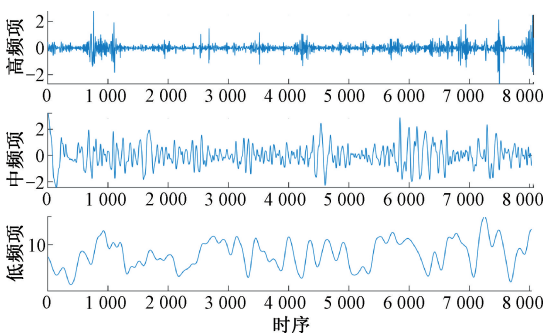


图3 重构分量结果

Fig.3 Reconstructed component results

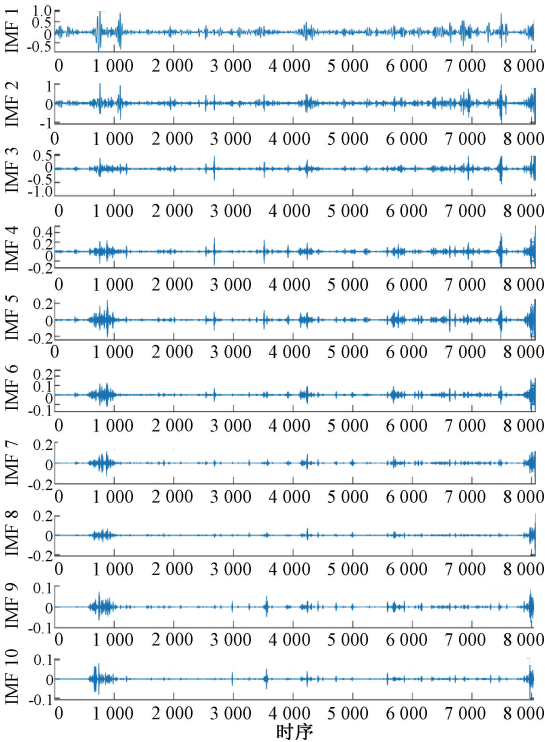


图4 奇异谱分解结果

Fig.4 Results of SSD

表 3 奇异谱分解对应的特征值及方差贡献率

Table 3 Eigenvalues and variance contribution rates corresponding to SSD

| 分量序号 | 特征值        | 特征值求和      | 方差贡献率/% | 累计方差贡献率/% |
|------|------------|------------|---------|-----------|
| 1    | 52.053 120 | 52.053 12  | 28.71   | 28.71     |
| 2    | 49.929 670 | 101.982 80 | 27.54   | 56.25     |
| 3    | 27.455 530 | 129.438 30 | 15.14   | 71.40     |
| 4    | 17.521 720 | 146.960 00 | 9.67    | 81.06     |
| 5    | 11.124 070 | 158.084 10 | 6.14    | 87.20     |
| 6    | 7.807 030  | 165.891 10 | 4.31    | 91.51     |
| 7    | 5.408 130  | 171.299 30 | 2.98    | 94.49     |
| 8    | 4.125 764  | 175.425 00 | 2.28    | 96.77     |
| 9    | 3.054 656  | 178.479 70 | 1.68    | 98.45     |
| 10   | 2.807 292  | 181.287 00 | 1.55    | 100.00    |

由图 4 和表 3 可知,经奇异谱分解后的分量均呈现出无规则的波动,其中前 6 个分量保留了约 92%的信息,能够体现原高频序列的变化特征。为了尽可能保留原序列的信息,根据高频项与原始序列相关性不大且波动频繁的特征,经过多次试验,本文选取前 6 个分量作为重构后的高频项,其余分量作为噪声去除。

3.3 评价指标

考虑模型的绝对误差与相对误差,本文选取平均绝对百分比误差(mean absolute percentage error,MAPE)、平均绝对误差(mean absolute error,MAE)、均方误差(mean square error)作为模型误差分析的指标。其中 MAPE、MAE、MSE 的公式分别为:

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

(18)

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

(19)

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

(20)

3.4 结果分析

预测实验在 Window 10 x64 操作系统下进行,CPU 为 Intel (R) Core (TM) i7-8700 @ 3.20 GHz,内存为 8 GB,GPU 为 NVIDIA GeForce GTX 1060 6 GB。预测之前,首先对样本数据进行归一化处理,如式(21)所示:

$$x' = \frac{x_i - x_{\min}}{x_{\max} - x_{\min}}$$

(21)

式中:  $x'$  为归一化后的数据,  $x_i$  为原始数据,  $x_{\max}$  为原始数据最大值,  $x_{\min}$  为原始数据最小值。

优化算法的计算效率通常能够评价其性能的优劣,因此本文首先对优化算法的计算效率进行分析。在对 ELM 模型的连接权值与阈值进行优化时,均设置优化算法种群数量为 30,最大迭代次数为 50,将本文提出的 iJaya 优化算法与传统 Jaya 算法的适应度值进行比较,得到的结果如图 5 所示。

由图 5 可知,传统 Jaya 算法在迭代约 34 次时收敛,适应度约 0.001 1;而本文提出的 iJaya 算法在迭代约 24 次时收敛,适应度约 0.000 7。说明改进后的 Jaya 算法收敛速度更快、寻优能力更强,可能是由于本文加入的混沌因子能够平衡算法全局搜索与局部寻优的能力,且以最优解为牵引的扰动机制能够增大算法跳出局部最优的概率,不断找寻更优解,进而提升计算效率。

其次,为了验证本文模型的性能,从分解策略、优化算法、单一模型 3 个角度,与 10 个模型进行了对比。其中,分解策略方面,选取单一模式分解的 ICEEMDAN+iJaya+ELM、CEEMDAN+iJaya+ELM、EEMD+iJaya+ELM 这 3 种模型进行对比,以验证两阶段分解以及分解去噪策

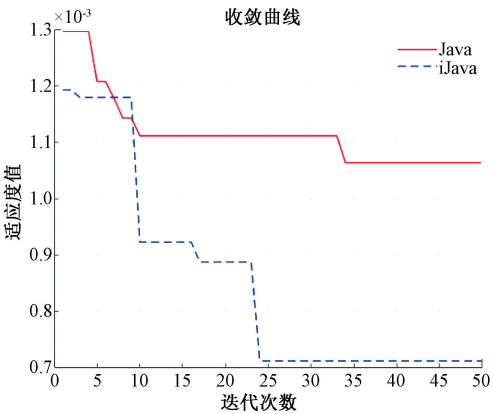


图 5 优化算法计算效率比较

Fig. 5 Calculation efficiency comparison of optimization algorithms

略的有效性;优化算法方面,选取未改进的 Jaya 算法优化 ELM 以及粒子群算法 PSO 优化 ELM 进行对比,同时,本文也比较了两阶段分解后的优化算法预测效果;单一模型方面,选取 ELM、SVR、BP (back propagation) 神经网络 3 种典型的单一模型进行对比,对比结果如表 4 所示。

表 4 各模型误差指标对比(甘肃)

Table 4 Comparison of error indexes of each model (Gansu)

| 序号 | 模型                     | 评价指标    |         |          |
|----|------------------------|---------|---------|----------|
|    |                        | MAE     | RMSE    | MAPE     |
| ①  | ICEEMDAN+SSA+iJaya+ELM | 0.067 9 | 0.134 5 | 0.668 6% |
| ②  | ICEEMDAN+SSA+Jaya+ELM  | 0.075 5 | 0.160 0 | 0.729 7% |
| ③  | ICEEMDAN+iJaya+ELM     | 0.104 0 | 0.207 4 | 1.020 4% |
| ④  | CEEMDAN+iJaya+ELM      | 0.110 2 | 0.223 9 | 1.273 3% |
| ⑤  | EEMD+iJaya+ELM         | 0.105 8 | 0.213 8 | 1.085 9% |
| ⑥  | iJaya-ELM              | 0.121 4 | 0.236 6 | 1.249 5% |
| ⑦  | Jaya-ELM               | 0.132 8 | 0.248 9 | 1.294 7% |
| ⑧  | PSO-ELM                | 0.150 6 | 0.269 4 | 1.519 2% |
| ⑨  | ELM                    | 0.170 4 | 0.315 8 | 1.994 7% |
| ⑩  | SVR                    | 0.178 6 | 0.322 3 | 2.065 3% |
| ⑪  | BP                     | 0.534 8 | 1.002 7 | 6.509 9% |

表 4 展示了不同对比模型在测试集上的性能,结果分析如下:

1) 与其他基准模型相比,本文提出的融合两阶段分解与 iJaya-ELM 混合预测模型取得了最优精度,其 MAE、RMSE、MAPE 值分别为 0.067 9、0.134 5、0.668 6%,显著低于其他模型。

2) 分解策略方面,经过奇异谱分解去噪后的模型误差显著低于未经奇异谱分解的模型,其中 MAE、RMSE、MAPE 分别降低了 34.71%、35.15%、34.48%,说明原始



风速序列较不平滑,存在较多噪声,如果对原始序列不加处理地直接进行建模和预测无疑会对模型的预测能力产生影响,而经过奇异谱分解并重构后的序列能够更加清晰地反应原始风速序列的特征,进而提升预测精度,也进一步证实了两阶段分解的必要性。同时,从分解方法上看,ICEEMDAN 方法显著优于 CEEMDAN、EEMD 两种分解方法,可能的原因是 ICEEMDAN 相比其他两种方法能够更加有效地解决模态混叠、端点效应等一系列固有缺陷。

3) 优化算法方面,经过优化后的模型普遍比未经优化的模型精度更好。进一步看,本文提出的 iJaya 优化算法预测精度均优于原始的 Jaya 算法,两阶段分解前后的 MAE、RMSE、MAPE 指标分别降低了 10.07%、15.94%、8.37%与 8.58%、4.94%、3.49%,也显著优于经过粒子群算法优化的 ELM 模型,进一步说明了优化 ELM 模型连接权值与阈值的必要性。同时,Jaya 类算法控制参数较少,也能够保证模型的稳定性。

4) 与 SVR、BP 等单一模型相比,ELM 模型也展现了良好的性能,其相比于 SVR、BP 模型 MAE、RMSE、MAPE 指标分别降低了 1.93%、1.78%、2.53%与 40.67%、39.14%、24.19%,且极限学习机在计算复杂度与泛化能力上具有一定的优势,因而本文选用 ELM 作为预测模型具有一定的科学性。

3.5 进一步分析

为进一步验证本文提出模型的鲁棒性与通用性,选用我国新疆地区某风电场测风塔 30 m 风速数据,时间范围为 2019 年 3 月 1 日~2019 年 5 月 31 日,共计 3 个月。该风电场每隔 15 min 进行一次采样,共得到 8 832 个采样点,如图 6 所示。选取前 7 949 个数据作为训练集,后 883 个数据作为测试集,分解步骤与实验环境与上文相同,得到的结果如表 5 所示。

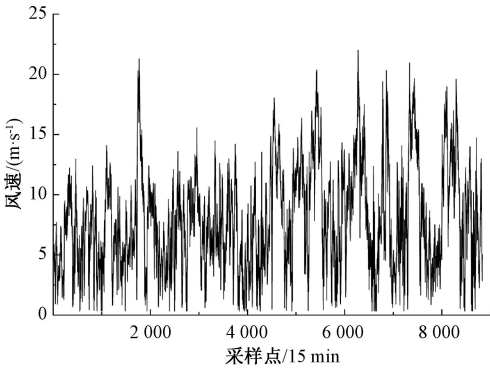


图 6 原始风速序列数据(新疆)

Fig. 6 Original wind speed sequence data (Xinjiang)

由表 5 可知,本文提出的模型 MAE、RMSE、MAPE 分别为 0.536 5、0.749 6、11.20%,仍然全部优于其他基准

模型。同时,经过两阶段分解后的模型显著优于单一分解策略的模型,能够适应不同步长的风速序列,较好地提取风速序列特征,再次证明了上文的结论。而本文提出的 iJaya 算法能明显提升预测模型的预测精度,具有良好的寻优精度与稳定性。

表 5 各模型误差指标对比(新疆)

Table 5 Comparison of error indexes of each model (Xinjiang)

| 序号 | 模型                     | 评价指标    |         |           |
|----|------------------------|---------|---------|-----------|
|    |                        | MAE     | RMSE    | MAPE      |
| ①  | ICEEMDAN+SSA+iJaya+ELM | 0.536 5 | 0.749 6 | 11.197 2% |
| ②  | ICEEMDAN+SSA+Jaya+ELM  | 0.541 1 | 0.752 7 | 11.313 6% |
| ③  | ICEEMDAN+iJaya+ELM     | 0.605 4 | 0.814 7 | 11.995 2% |
| ④  | CEEMDAN+iJaya+ELM      | 0.690 1 | 0.902 5 | 14.754 4% |
| ⑤  | EEMD+iJaya+ELM         | 0.687 3 | 0.902   | 14.796 3% |
| ⑥  | iJaya-ELM              | 0.858 9 | 1.214 2 | 18.810 5% |
| ⑦  | Jaya-ELM               | 0.862 2 | 1.216 6 | 18.871 2% |
| ⑧  | PSO-ELM                | 0.885 6 | 1.241 7 | 19.381 9% |
| ⑨  | ELM                    | 1.334 6 | 2.328 5 | 24.725 6% |
| ⑩  | SVR                    | 1.360 8 | 2.370 6 | 25.368 4% |
| ⑪  | BP                     | 2.149 6 | 3.725 9 | 31.613 1% |

4 结 论

针对风电场风速序列噪声较多、预测效果不稳定等情况,本文从挖掘风速数据有效特征和优化预测模型性能两方面出发,提出了一种融合两阶段分解与 iJaya-ELM 的风速预测模型,并利用我国甘肃与新疆地区风电场风速数据进行实证分析,主要结论如下:

1) 在分解集成策略的基础上进一步采用分解去噪策略能够深度挖掘风速序列特征,提高风速预测精度,弥补单一分解策略的不足。

2) 本文提出的 iJaya 算法弥补了传统 Jaya 算法寻优精度较低、局部搜索能力不足、收敛速度较慢等缺陷,有效提高了 ELM 模型的精度。

3) 本文提出的融合两阶段分解与 iJaya-ELM 模型在各对比模型中表现均最优,能够适应不同步长的风速序列,具有较好的稳定性与通用性。

未来,可进一步考虑各风电机组的时空相关性,在解决风速相关性的基础上提升风速预测精度。

参考文献

[ 1 ] ZHAO J, GUO Z H, SU Z Y, et al. An improved multi-step forecasting model based on WRF ensembles and creative fuzzy systems for wind speed [ J ]. Applied Energy, 2016, 162:808-826.



- [ 2 ] XIE K, LIAO Q, TAI H M, et al. Non-homogeneous Markov wind speed time series model considering daily and seasonal variation characteristics [ J ]. IEEE Transactions on Sustainable Energy, 2017, 8 ( 3 ) : 1281-1290.
- [ 3 ] 丁藤, 冯冬涵, 林晓凡, 等. 基于修正后 ARIMA-GARCH 模型的超短期风速预测 [ J ]. 电网技术, 2017, 41(6) : 1808-1814.  
DING T, FENG D H, LIN X F, et al. Ultra-short term wind speed prediction based on modified ARIMA-GARCH model [ J ]. Power System Technology, 2017, 41(6) : 1808-1814.
- [ 4 ] 孙荣富, 张涛, 和青, 等. 风电功率预测关键技术及应用综述 [ J ]. 高电压技术, 2021, 47(4) : 1129-1143.  
SUN R F, ZHANG T, HE Q, et al. Review on key technologies and applications of wind power prediction [ J ]. High Voltage Technology, 2021, 47(4) : 1129-1143.
- [ 5 ] 潘超, 李润宇, 蔡国伟, 等. 考虑风速属性约简聚类的组合预测模型 [ J ]. 电网技术, 2022, 46 ( 4 ) : 1355-1362.  
PAN CH, LI R Y, CAI G W, et al. Combined prediction model considering wind speed attribute reduction clustering [ J ]. Power System Technology, 2022, 46(4) : 1355-1362.
- [ 6 ] 钱宇, 何益丰, 谢斌鑫, 等. 基于特征工程的 XGboost 风速短期预测 [ J ]. 武汉大学学报 ( 工学版 ), 2022, 55(10) : 1057-1064.  
QIAN Y, HE Y F, XIE B X, et al. Wind speed short-term prediction based on XGboost Feature Engineering [ J ]. Journal of Wuhan University ( Engineering Science ), 2022, 55(10) : 1057-1064.
- [ 7 ] 李永刚, 王月, 刘丰瑞, 等. 基于 Stacking 融合的短期风速预测组合模型 [ J ]. 电网技术, 2020, 44 ( 8 ) : 2875-2882.  
LI Y G, WANG Y, LIU F R, et al. Short-term wind speed forecasting combination model based on Stacking fusion [ J ]. Power System Technology, 2020, 44 ( 8 ) : 2875-2882.
- [ 8 ] LIU M, CAO Z, ZHANG J, et al. Short-term wind speed forecasting based on the Jaya-SVM model [ J ]. International Journal of Electrical Power & Energy Systems, 2020, 121 : 106056.
- [ 9 ] 梁小珍, 邬志坤, 杨明歌, 等. 基于二次分解策略和模糊时间序列模型的航空客运需求预测研究 [ J ]. 中国管理科学, 2020, 28(12) : 108-117.  
LIANG X ZH, WU ZH K, YANG M G, et al. Research on air passenger demand forecast based on quadratic decomposition strategy and fuzzy time series model [ J ]. Chinese Journal of Management Science, 2020, 28(12) : 108-117.
- [ 10 ] 李嘉文, 盛德仁, 李蔚. 基于分解去噪和 LSSVM 的短期风速预测 [ J ]. 能源工程, 2021, 213(4) : 17-24.  
LI J W, SHENG D R, LI W. Short-term wind speed prediction based on decomposition denoising and LSSVM [ J ]. Energy Engineering, 2021, 213(4) : 17-24.
- [ 11 ] 王贺, 陈蕪峰, 熊敏, 等. 融合 CEEMDAN 和 ICS-LSTM 的短期风速预测建模 [ J ]. 电子测量与仪器学报, 2022, 36(4) : 17-23.  
WANG H, CHEN H F, XIONG M, et al. Short-term wind speed prediction modeling based on CEEMDAN and ICS-LSTM [ J ]. Journal of Electronic Measurement and Instrumentation, 2022, 36(4) : 17-23.
- [ 12 ] 张妍, 韩璞, 王东风, 等. 基于变分模态分解和 LSSVM 的风电场短期风速预测 [ J ]. 太阳能学报, 2018, 39(1) : 194-202.  
ZHANG Y, HAN P, WANG D F, et al. Short-term wind speed prediction of wind farm based on variational mode decomposition and LSSVM [ J ]. Acta Energiæ Solaris Sinica, 2018, 39(1) : 194-202.
- [ 13 ] 桑茂景, 谢丽蓉, 李进卫, 等. 基于 LMD-IMVO-LSSVM 的短期风速预测 [ J ]. 可再生能源, 2021, 39 ( 9 ) : 1197-1203.  
SANG M J, XIE L R, LI J W, et al. Short-term wind speed prediction based on LDM-IMVO-LSSVM [ J ]. Renewable Energy, 2021, 39(9) : 1197-1203.
- [ 14 ] REN Y, SUGANTHAN P N, SRIKANTH N. A comparative study of empirical mode decomposition-based short-term wind speed forecasting methods [ J ]. IEEE Transactions on Sustainable Energy, 2014, 6 ( 1 ) : 236-244.
- [ 15 ] 刘起超, 周云龙, 陈聪. 基于 CEEMDAN 和概率神经网络的起伏振动气液两相流型识别 [ J ]. 仪器仪表学报, 2021, 42(10) : 84-93.  
LIU Q CH, ZHOU Y L, CHEN C. Recognition of gas-liquid two-phase flow pattern with fluctuating vibration based on CEEMDAN and probabilistic neural network [ J ]. Chinese Journal of Scientific Instrument, 2021, 42(10) : 84-93.
- [ 16 ] TORRES M E, COLOMINAS M A, SCHLOTTHAUER G, et al. A complete ensemble empirical mode decomposition with adaptive noise [ C ]. 2011 IEEE International Conference on Acoustics, Speech and Signal Processing ( ICASSP ). IEEE, 2011 : 4144-4147.
- [ 17 ] 王逸文, 王维莉, 陈怡霏, 等. 基于奇异谱分析的长时交通流混合预测模型 [ J ]. 电子测量与仪器学报, 2022, 36(11) : 98-106.  
WANG Y W, WANG W L, CHEN Y F, et al. Mixed

prediction model of long-term traffic flow based on singular spectrum analysis [ J ]. Journal of Electronic Measurement and Instrumentation, 2022, 36 ( 11 ): 98-106.

[ 18 ] RAO R. Jaya: A simple and new optimization algorithm for solving constrained and unconstrained optimization problems [ J ]. International Journal of Industrial Engineering Computations, 2016, 7(1): 19-34.

[ 19 ] 王逸文,王维莉. 基于 LSTM-RELM 组合模型的电商 GMV 预测研究 [ J ]. 计算机工程与应用, 2023, 59(10): 321-327.

WANG Y W, WANG W L. Research on E-commerce GMV prediction based on LSTM-RELM combination model [ J ]. Computer Engineering and Applications, 2023, 59(10): 321-327.

[ 20 ] 武新章,梁祥宇,朱虹谕,等. 基于 CEEMDAN-GRA-PCC-ATCN 的短期风电功率预测 [ J ]. 山东大学学报 (工学版), 2022, 52(6): 146-156.

WU X ZH, LIANG X Y, ZHU H Y, et al. Short-term wind power prediction based on CEEMDAN-GRA-PCC-ATCN [ J ]. Journal of Shandong University ( Engineering Science ), 2022, 52(6): 146-156.

作者简介



王逸文, 2021 年于厦门大学嘉庚学院获得学士学位, 现为上海海事大学硕士研究生, 主要研究方向为智能算法与系统优化。  
E-mail: 929458033@qq.com

**Wang Yiwen** received his B. Sc. degree from Xiamen University TKK College in 2021.

Now he is a M. Sc. candidate at Shanghai Maritime University. His main research interests include intelligent algorithm and system optimization.



王维莉 (通信作者), 2009 年于武汉大学获得学士学位, 2011 于武汉大学获得硕士学位, 2015 年于香港城市大学获得博士学位, 现为上海海事大学副教授, 主要研究方向为复杂系统建模与仿真。

E-mail: wlwang@shmtu.edu.cn

**Wang Weili** (Corresponding author) received her B. Sc. degree from Wuhan University in 2009, M. Sc. degree from Wuhan University in 2011, and Ph. D. degree from City University of Hong Kong in 2015, respectively. Now she is an associate professor in Shanghai Maritime University. Her main research interest includes modeling and simulation of complex system.