

基于优化后的内核极限学习机的短期风力发电功率预测

宣 畅

(浙江工商大学萨塞克斯人工智能学院, 浙江 杭州 310018)

✉xuan_chang@qq.com



摘 要: 为提高短期风力发电功率预测的精度, 经过对比选择了内核极限学习机(Kernel Extreme Learning Machine, KELM)预测模型的原始模型, 对该模型的内部参数进行研究, 选择了多元宇宙优化算法(Multi-Verse Optimizer, MVO)对其参数进行优化。还提出用鲸鱼优化算法(Whale Optimization Algorithm, WOA)来给MOV初始化种群, 以使MVO算法更不容易陷入局部最优, 从而有着更好的求解能力。通过此预测模型进行发电功率预测, 获得一个均方根误差(RMSE)值为0.0032、平均预测误差为0.00033的预测结果, 最后进行对比实验验证其具有较好的预测效果。

关键词: 发电量预测; 多元宇宙优化算法; 鲸鱼优化算法; 内核极限学习机

中图分类号: TP301 **文献标识码:** A

Short-term Wind Power Prediction based on Optimized Kernel Extreme Learning Machine

XUAN Chang

(Sussex School of Artificial Intelligence, Zhejiang Gongshang University, Hangzhou 310018, China)

✉xuan_chang@qq.com

Abstract: In order to improve the accuracy of short-term wind power prediction, this paper proposes to use the original model of Kernel Extreme Learning Machine (KELM) prediction model and study its internal parameters after comparison. Multi-Verse Optimizer (MVO) is selected to optimize its parameters. This paper also proposes to use Whale Optimization Algorithm (WOA) to initialize MOV population, so that MVO is much less likely to fall into local optimum, and thus it has better solving ability. The proposed prediction model is used to predict a power generation, and obtain a prediction result of RMSE value of 0.0032 and average prediction error of 0.00033. Finally, a comparative experiment verifies its good prediction effect.

Keywords: power generation prediction; multi-verse optimizer; whale optimization algorithm; kernel extreme learning machine

1 引言(Introduction)

在现代化发展中, 越来越需要一个稳定、节能、经济的电力来源, 因此利用可再生能源进行发电的技术越来越被广泛关注。在各种绿色可再生能源中, 风能是一种非常常见的绿色能源。近些年, 我国各个地区也在大修风电场来进行风力发电。但是, 风能是一种十分不稳定且随机性很强的资源, 且这个特性会造成风电场发电不稳定。因此, 需要通过一定的技术手段对风力发电功率进行预测, 这也有利于

风电场的建设与维护, 减轻风能不稳定的特性对电网造成的冲击。

针对目前阶段如何保证电力系统在得到最大化受益的同时, 减少风力发电带来的冲击, 相关领域的研究学者提出了许多方法, 如基于小波变换和BP神经网络预测法^[1]、支持向量机(Support Vector Machine, SVM)^[2]、人工蜂群算法优化神经网络预测法^[3]、深浅层神经网络^[4]等预测方法, 但效果并非最佳, 因此本文建立了一个新型预测模型。

2 算法介绍(Introduction to the algorithm)

2.1 内核极限学习机(KELM)

极限学习机(Extreme Learning Machine, ELM)虽然学习效率, 但基于经验风险最小化原理进行学习, 容易出现过拟合现象, 为此HUANG^[5]提出了KELM算法, 以提高其泛化能力。其数学模型推算如下:

$$F(X) = h(X)\beta \quad (1)$$

根据KKT(Karush-Kuhn-Tucker)^[6]解式(1)可以转换为如下等式:

$$L_{\text{KELM}} = \frac{1}{2}\|\beta\|^2 + \frac{c}{2}\sum_{i=1}^N\|\xi_i\|^2 - \sum_{i=1}^N\sum_{j=1}^m\alpha_{i,j}(h(X_i)\beta_j - t_{i,j} + \xi_{i,j}) \quad (2)$$

在式(2)中, c 表示惩罚参数, $\xi_i = [\xi_{i,1}, \xi_{i,2}, \dots, \xi_{i,m}]^T$ 表示训练误差, $\alpha_{i,j} = [\alpha_{i,1}, \alpha_{i,2}, \dots, \alpha_{i,m}]^T$ 表示拉格朗日乘子。求解可得矩阵 β 如下:

$$\beta = H^T \left(\frac{I}{c} + HH^T \right)^{-1} T \quad (3)$$

式(3)中, $T = \begin{bmatrix} T_1^T \\ T_2^T \\ \vdots \\ T_N^T \end{bmatrix} = \begin{bmatrix} t_{11} & t_{12} & \dots & t_{1m} \\ t_{21} & t_{22} & \dots & t_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ t_{N1} & t_{N2} & \dots & t_{Nm} \end{bmatrix}$ 为目标值矩阵,

$$H = \begin{bmatrix} h(X_1) \\ h(X_2) \\ \vdots \\ h(X_N) \end{bmatrix} = \begin{bmatrix} h_1(X_1) & h_2(X_1) & \dots & h_L(X_1) \\ h_1(X_2) & h_2(X_2) & \dots & h_L(X_2) \\ \vdots & \vdots & \ddots & \vdots \\ h_1(X_N) & h_2(X_N) & \dots & h_L(X_N) \end{bmatrix}$$
为隐含层输出矩阵。

然后通过应用Mercer条件定义KELM的核矩阵:

$$\Omega_{\text{ELM}} = HH^T = h(X_i) \cdot h(X_j) = K(X_i, X_j) \quad (4)$$

将式(2)、式(3)和式(4)代入式(1), 则KELM模型的实际输出可以写成紧凑形式:

$$F(X) = h(X)H^T \left(\frac{I}{c} + HH^T \right)^{-1} T = \begin{bmatrix} K(X, X_1) \\ K(X, X_2) \\ \vdots \\ K(X, X_N) \end{bmatrix} \left(\frac{I}{c} + \Omega_{\text{ELM}} \right)^{-1} T \quad (5)$$

从上述一系列推导过程来看, 与ELM人工调节参数的要求以及容易陷入局部最优的缺点不同, KELM解决了ELM中的随机初始化问题。同时, KELM具有较少的可调参数、更快的收敛速度和更好的泛化性能。由式(5)可以看出, KELM算法的性能主要和惩罚参数 c 及核参数 g 这两个值有直接关系。这两个值的变化也会影响模型预测的精度, 所以需要用其他算法来优化。

2.2 多元宇宙优化算法(MVO)

多元宇宙优化算法(MVO)是由MIRJALILI等^[7]根据白洞、黑洞、虫洞这三个主要概念提出来的。白洞拥有很强的斥力; 黑洞则有极高的引力; 虫洞是连接不同宇宙和传输物体的轨道。

MVO算法的优化执行过程分为两个阶段, 即探测和开

采。白洞和黑洞作用于探测阶段, 而虫洞则作用于开采阶段, 其数学模型如下:

首先是初始化一个随机宇宙种群 U :

$$U = \begin{bmatrix} x_1^1 & x_1^2 & \dots & x_1^d \\ x_2^1 & x_2^2 & \dots & x_2^d \\ \vdots & \vdots & \ddots & \vdots \\ x_n^1 & x_n^2 & \dots & x_n^d \end{bmatrix} \quad (6)$$

矩阵中 d 为维度即变量的个数, n 为候选解的个数, x_i^j 表示第 i 个宇宙的第 j 个变量。

接下来进行的是选择过程。由于每个宇宙个体的膨胀速度不同, 宇宙个体中的物体会通过连接白洞和黑洞的隧道传递。膨胀率越高, 物体通过隧道的概率越低。这个过程遵循轮盘赌机制, 如式(7)所示:

$$x_i^j = \begin{cases} x_k^j & r_1 < NI(U_i) \\ x_i^j & r_1 \geq NI(U_i) \end{cases} \quad (7)$$

式(7)中, U_i 表示第 i 个宇宙, $NI(U_i)$ 表示第 i 个宇宙的归一化膨胀率, x_i^j 表示第 i 个宇宙的第 j 个参数, r_1 是 $[0, 1]$ 内的随机数, x_k^j 则表示经过轮盘赌机制选择出的第 k 个宇宙中的第 j 个参数。

有了上述机制, 两个宇宙在没有摄动的情况下继续交换物体。为了维持多元宇宙给每个宇宙提供局部的变化的多样性和进行开发, 默认每个宇宙是存在虫洞的, 且虫洞可以随机地在空间中传输物体, 从而建立了一个机制, 这个机制的构想如下:

$$x_i^j = \begin{cases} \begin{cases} X_j + TDR \times ((ub_j - lb_j) \times r_4 + lb_j) & r_3 < 0.5 \\ X_j - TDR \times ((ub_j - lb_j) \times r_4 + lb_j) & r_3 \geq 0.5 \end{cases} & r_2 < WEP \\ x_i^j & r_2 \geq WEP \end{cases} \quad (8)$$

式(8)中, X_j 表示到目前为止搜索到的最优宇宙的第 j 个参数, ub_j 和 lb_j 分别为第 j 个变量的上界和下界, r_2 、 r_3 、 r_4 为 $[0, 1]$ 范围内的随机数。 TDR 和 WEP 分别为移动距离速率和虫洞存在概率。 WEP 用来定义虫洞在宇宙中存在的概率, 需要在迭代过程中线性增长, 以强调作为优化过程进展的开发。 TDR 定义一个物体在迄今为止获得的最佳宇宙中被虫洞传送的距离变化率的一个因素。与 WEP 相反, TDR 会随着迭代次数的增加而增加, 以便在获得的最佳宇宙周围进行更精确的开发或搜索。两个系数的自适应公式如下:

$$WEP = \min + l \times \left(\frac{\max - \min}{L} \right) \quad (9)$$

式(9)中, $\min$ 和 $\max$ 分别为最小值和最大值, 均为手动设置; l 为当前迭代次数; L 为最大迭代次数。

$$TDR = 1 - \frac{l^{1/p}}{L^{1/p}} \quad (10)$$

式(10)中, p 为迭代过程中的开发精度。

2.3 鲸鱼优化算法(WOA)

鲸鱼优化算法(WOA)是一种基于鲸鱼捕猎行为而提出的优化算法^[8], 主要过程分为:

(1) 包围猎物

鲸鱼能够搜寻猎物并将其包围。鲸鱼算法的初始位置是随机赋予的, 也就是说一开始鲸鱼的位置为默认的最佳位置, 在经过一轮搜寻后, 算法会更新得到一个新的鲸鱼位置。将适应度作为对比条件将两个位置进行对比, 如果新位置的适应度较好则更新最佳位置, 否则保留原位置。其公式如下:

$$\bar{D} = \left| \bar{C} \cdot \bar{X}^*(t) - \bar{X}(t) \right| \quad (11)$$

$$\bar{X}(t+1) = \bar{X}^*(t) - \bar{A} \cdot \bar{D} \quad (12)$$

式(11)和式(12)中, $\bar{A}$ 和 $\bar{C}$ 是系数向量, t 是当前迭代次数, $\bar{X}$ 是当前位置的向量, $\bar{X}^*$ 是迄今为止得到的最佳鲸鱼的位置向量。向量 $\bar{A}$ 和 $\bar{C}$ 的表达式如下:

$$\bar{A} = 2\bar{a} \cdot \bar{r} - \bar{a} \quad (13)$$

$$\bar{C} = 2 \cdot \bar{r} \quad (14)$$

式(13)中, $\bar{a}$ 在迭代过程中(包括搜索和开发阶段)从2到0线性递减, $\bar{r}$ 是[0,1]中的随机向量。

(2) 泡泡网螺旋更新

算法通过缩小环绕机制和螺旋更新位置机制对座头鲸的泡泡网行为进行数学建模, 其数学模型如下:

$$\bar{X}(t+1) = \bar{D}^l \cdot e^{bl} \cdot \cos(2\pi l) + \bar{X}^*(t) \quad (15)$$

式(15)中, $\bar{D}^l$ 表示迄今为止获得的最优解, b 是对数螺旋形状的常数, l 是[-1,1]中的随机数。

为了模拟上述两个同时发生的行为, 建立了一个存在如下逻辑关系的数学模型:

$$\bar{X}(t+1) = \begin{cases} \bar{X}^*(t) - \bar{A} \cdot \bar{D} & \text{if } p < 0.5 \\ \bar{D}^l \cdot e^{bl} \cdot \cos(2\pi l) + \bar{X}^*(t) & \text{if } p \geq 0.5 \end{cases} \quad (16)$$

式(16)中, p 是[0,1]中的随机数。

(3) 搜索阶段

加强对WOA的探索, 同样地, 基于向量 $\bar{A}$ 变化的方法也可以用于搜索过程。鲸鱼将使用从当前搜索代理中随机选择的位置来随机搜索猎物, 以相应地更新它们的位置。其数学模型如下:

$$\bar{D} = \left| \bar{C} \cdot \bar{X}_{\text{rand}} - \bar{X} \right| \quad (17)$$

$$\bar{X}(t+1) = \bar{X}_{\text{rand}} - \bar{A} \cdot \bar{D} \quad (18)$$

式(18)中, $\bar{X}_{\text{rand}}$ 是从当前种群中选择的随机位置向量(随机鲸鱼)。

3 基于WOA-MVO 优化的 KELM 风电场发电量预测(Power generation prediction of KELM wind farm based on WOA-MVO optimization)

3.1 数据收集与处理

本文所使用的数据集是通过Renewables这个开源的天气网站获取的。该网站通过NASA MERRA^[9]全球再分析模型和CM-SAF's SARAH^[10]卫星观测获取天气数据来工作。本文选择2019 年全年杭州的历史天气数据作为数据集的原始数据, 其中包含风电场的发电量和和其他九个特征。

在最初的数据集中有许多天气元素, 由于风电场的周围环境、空气的热对流或涡流均能引起风速变化, 导致风能具有强烈的随机性、波动性和不确定性。但也并不是所有因素都会影响风电场的发电量, 所以为了使得预测模型达到更好的预测效果, 本文还对此数据集进行了优化。

本文采用皮尔逊相关系数法对数据的特征值进行了相关性分析, 结果如图1和图2所示。

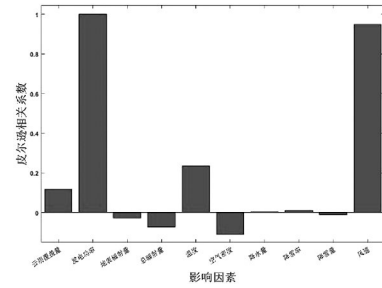


图1 皮尔逊相关系数法

Fig.1 Pearson correlation coefficient method

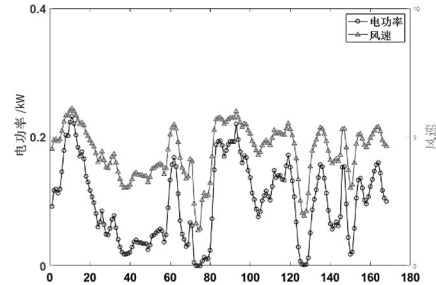


图2 风速与发电量的变化规律

Fig.2 Variation law of wind speed and power generation

图1表示的是各个天气因素与通过风电场发电量计算得到的皮尔逊系数的值, 皮尔逊系数大于0.1且其值越大, 相关性就越高。图2表示的是单个因素与发电量的变化规律。可以看出, 风电场发电量与风速是密切相关的, 其次就是温度、云层覆盖量和空气密度。

3.2 数据集的构建

经过上述分析, 本文选取风速、温度、空气密度和云层覆盖量四个特征点, 将它们与每天每小时持续365 天所测得的

真实发电量组合构建一个 8760×5 的数据集,部分数据集展示如表1所示。

表1 部分数据集

Tab.1 Partial data sets

发电功率/ $\times 10^3 \text{ kW}$	风速/ m/s	温度/ $^{\circ}\text{C}$	空气密度/ kg/m^3	云层覆盖率/%
0.092	4.531	1.068	1.308	0.393
0.117	4.873	2.851	1.301	0.304
0.119	4.903	4.186	1.295	0.319
0.113	4.824	5.015	1.29	0.393
0.119	4.903	5.521	1.286	0.529
0.146	5.231	5.642	1.284	0.546
0.179	5.588	5.398	1.284	0.504
0.203	5.831	4.786	1.285	0.439
0.202	5.824	3.38	1.288	0.315
0.223	6.015	1.15	1.291	0.22
0.232	6.094	0.437	1.294	0.191
0.221	5.997	-0.111	1.297	0.175
0.203	5.831	-0.538	1.299	0.205
0.184	5.644	-0.798	1.301	0.305
0.17	5.494	-0.729	1.302	0.535
0.177	5.556	-0.449	1.304	0.589
0.166	5.446	-0.618	1.305	0.535

3.3 建立WOA-MVO-KELM预测模型

虽然KELM具有较少的可调参数、更快的收敛速度和更好的泛化性能,但仍有不足之处,即惩罚参数 c 和核参数 g 是随机产生的,且这两个值的不同对结果也有着很大的影响。目前还没有公认统一的最佳方法来对KELM进行优化。通常是用网格搜索方法来优化参数,但是该方法在设置参数范围方面会遇到困难,并且容易陷入局部最优。

本文先利用WOA算法进行一个初始化种群的工作,通过多次运算WOA算法,每次得到一个最优位置向量并保存在一个矩阵里,最后将这个矩阵作为MVO算法的初始宇宙种群得到一个优化后的WOA-MVO混合优化算法。然后利用WOA-MVO混合优化算法对KELM模型里的两个参数进行进一步的寻优,利用黑洞、白洞和虫洞的开发与搜索获得一个最优的惩罚参数和核参数的组合,并将此组合赋予KELM模型。最后,本文提出了基于鲸鱼算法改进的多元宇宙算法优化的KELM,即WOA-MVO-KELM,该方法的总体框架如图3所示。

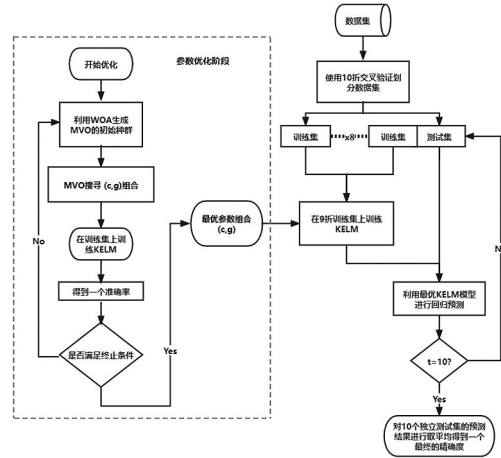


图3 WOA-MVO-KELM预测模型流程图

Fig.3 Flow chart of WOA-MVO-KELM prediction model

在图3中,预测模型主要由两个程序组成,包括参数优化和预测属性评估。左半部分为改进的MVO优化算法搜寻最优参数组合的过程,当内部参数优化过程终止时,将最优参数对输入KELM预测模型进行10折交叉验证,然后进行预测工作。最后通过计算得出均方差(MSE)、均方根误差(RMSE)和真实值与预测值的平均误差进行预测效果评估。MSE和RMSE的数学表达式如下:

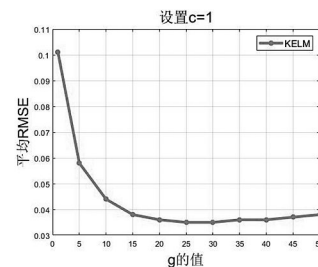
$$MSE = \frac{1}{m} \sum_{i=1}^m (y_i - \hat{y}_i)^2 \quad (19)$$

$$RMSE = \sqrt{\frac{1}{m} \sum_{i=1}^m (y_i - \hat{y}_i)^2} \quad (20)$$

式(19)和式(20)中, m 代表样本数量, $y_i - \hat{y}_i$ 代表预测值与真实值的误差。

3.4 内核极限学习机(KELM)参数研究

KELM的可设参数其实只有两个,分别是惩罚参数 c 和核参数 g ,相较于ELM,它无须设置隐藏节点的数目,因为它将这些烦琐的步骤都用一个内核函数所替代。而本文将通过对这两个参数的动态分析确定一个相对较好的参数组合,以便于后续对其他算法的研究对比。选取总数据集的四分之一即1,464个样本作为训练和测试的样本数据,分别设定惩罚参数 c 为1,核参数 g 取值为:0.1、5、10、15、20、25、30、35、40、45、50;再设定核参数 g 为1,惩罚参数 c 取值为:0.1、5、10、15、20、25、30、35、40、45、50。绘制出的平均均方根误差变化图如图4和图5所示。

图4 $c=1$ 平均均方根误差变化图Fig.4 When $c=1$, variation diagram of average RMSE

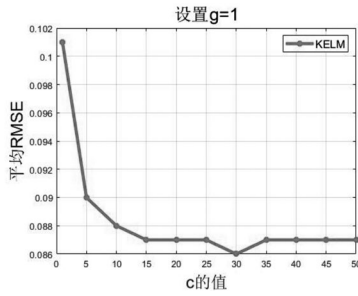
图5 $g=1$ 平均均方根误差变化图

Fig.5 When $g=1$, variation diagram of average RMSE

上述两幅图所示的为 c 和 g 在 $(0, 50]$ 区间内的变化范围, 为了实验的严谨性, 现在分别补充设定 $c=100$ (g 取100、200、300、400、500、600)和 $g=100$ (c 取100、200、300、400、500、600), 绘制平均均方根误差变化图, 如图6和图7所示。

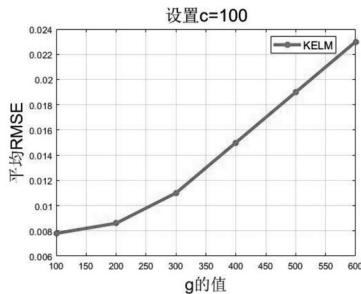
图6 $c=100$ 平均均方根误差变化图

Fig.6 When $c=100$, variation diagram of average RMSE

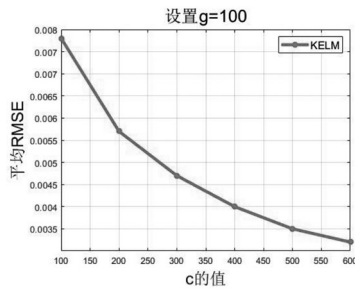
图7 $g=100$ 平均均方根误差变化图

Fig.7 When $g=100$, variation diagram of average RMSE

最后, 再将四次实验的惩罚参数 c 和核参数 g 的值混合起来, 绘制三维直观图, 可以清晰地分析出惩罚参数 c 相较于核参数 g 对预测的精准度影响更大, 如图8和图9所示。

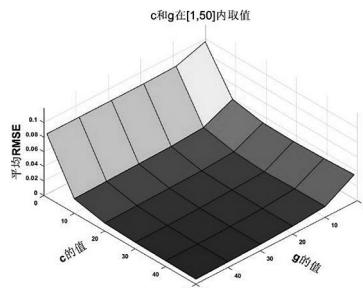
图8 c, g 各取 $[1, 50]$

Fig.8 c, g in $[1, 50]$

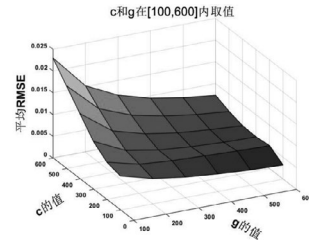
图9 c, g 各取 $[100, 600]$

Fig.9 c, g in $[100, 600]$

3.5 多元宇宙算法(MVO)参数研究

为了研究WOA-MVO-KELM中MVO的标准参数的影响, 在KELM所选核函数为径向基核函数的条件下, 选取总数据集的四分之一即1,464个样本作为训练和测试的样本数据。通过对MVO的研究可以发现, 在算法中存在三个重要参数, 一个为开发精度 p , 如式(21)所示:

$$TDR = 1 - \frac{l^{1/p}}{L^{1/p}} \quad (21)$$

另外两个参数分别为宇宙种群数量和最大迭代次数。因此本文进行了三组不同开发精度的对比实验, 分别取 $p=2$ 、 $p=4$ 和 $p=6$, 每组实验的变量分别为不同组合的种群数量和最大迭代次数。为确保实验的准确性, 三组实验的横纵坐标取值范围相同, 种群数取10、20、30、40、50这五个不同的值, 最大迭代次数取10、20、30、40、50、60、70、80这八个不同的值。然后对样本进行预测, 获得平均均方根误差, 结果如图10至图12所示。

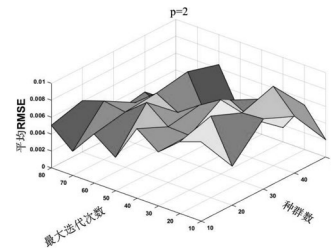
图10 $p=2$ 时, 最大迭代次数与种群数目组合下平均均方根误差变化图

Fig.10 When $p=2$, variation diagram of average RMSE under the combination of maximum iteration times and population number

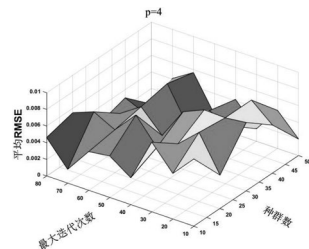
图11 $p=4$ 时, 最大迭代次数与种群数目组合下平均均方根误差变化图

Fig.11 When $p=4$, variation diagram of average RMSE under the combination of maximum iteration times and population number

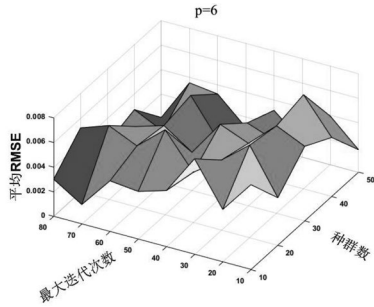


图12 $p=6$ 时，最大迭代次数与种群数目组合下平均均方根误差变化图

Fig.12 When $p=6$, variation diagram of average RMSE under the combination of maximum iteration times and population number

通过对比分析图10、图11、图12，可以得到开发精度 $p=6$ 时，算法的预测准确度最高且较为稳定，区间内算法的平均准确率相对较为稳定。如图13至图15所示为不同种群数与迭代次数组合下的气泡图，每一个气泡的半径为在此组合下的平均均方根误差值。通过分析和对比可以得出，迭代次数越高，模型预测的精度越高且种群数为50、迭代次数为80时效果最好。但迭代次数过高的话会导致运行时间过长从而降低算法的运算效率，所以本文最终选取开发精度 p 为6、种群数为50、最大迭代次数为30作为实验所用参数。

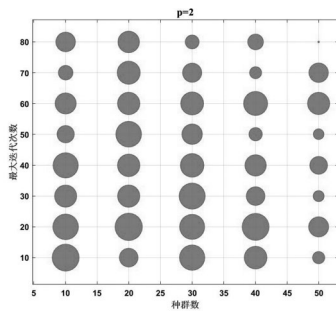


图13 $p=2$ 时，最大迭代次数与种群数目组合下的气泡图
Fig.13 When $p=2$, bubble graph under the combination of maximum iteration times and population number

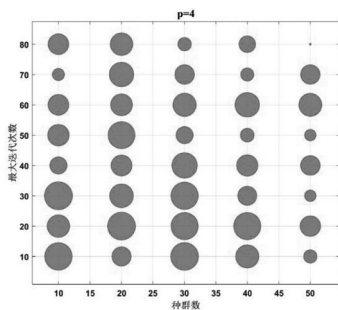


图14 $p=4$ 时，最大迭代次数与种群数目组合下的气泡图
Fig.14 When $p=4$, bubble graph under the combination of maximum iteration times and population number

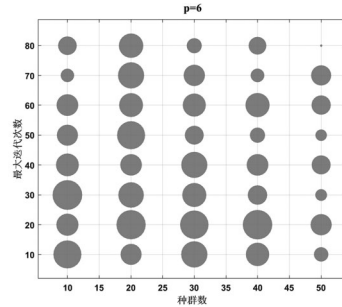


图15 $p=6$ 时，最大迭代次数与种群数目组合下的气泡图
Fig.15 When $p=6$, bubble graph under the combination of maximum iteration times and population number

3.6 发电功率预测实验结果与对比分析

(1)预测实验结果

设定好主预测模型WOA-MVO-KELM的参数后，本文同时选取了10组其他对比试验，以便更好地体现主模型的预测效果。10组对比实验分别有：KELM模型、MVO-KELM模型、WOA-KELM模型、MFO-KELM模型、PSO-KELM模型^[11]、GOA-KELM模型、ELM模型^[12]、BP神经网络模型、SVR模型、OSELM模型^[13]。本文所有实验是在MATLAB-R2021编译环境下实现的。由于所研究的模型为WOA-MVO-KELM，其可变的参数仅有MVO中的种群数目和最大迭代次数，因此本文选取这两个参数进行预测实验。将以上10种模型分别运行10次，记录下每次的数据。实验结果将通过以下四个指标进行算法性能评估，分别为：均方根误差(RMSE)、平均预测误差(Avg_error)、最好预测误差值(Best_error)和最坏预测误差值(Worst_error)。对上述四个值的10次数据取平均值并记录在表2和表3中。其中WOA-MVO-KELM模型为性能最好的模型。

表2 WOA-MVO-KELM与KELM相关的算法进行比较

Tab.2 Comparison between WOA-MVO-KELM and

评估指标	KELM related algorithms						
	WOA-MVO-KELM*	MVO-KELM	KELM	WOA-KELM	PSO-KELM	MFO-KELM	GOA-KELM
RMSE	0.0032	0.0097	0.0474	0.0094	0.0128	0.0098	0.0089
Avg_error	0.00033	0.00067	0.0112	0.00076	0.00089	0.00079	0.00068
Best_error	0	0	0.0004	0	0	0	0
Worst_error	0.0187	0.0510	0.3472	0.0728	0.1275	0.0971	0.0653

注：*表示性能最好的模型。

表3 WOA-MVO-KELM与其他算法进行比较

Tab.3 Comparison between WOA-MVO-KELM and other algorithms

评估指标	WOA-MVO-KELM*	BP	SVR	OSELM	ELM
RMSE	0.0032	0.1158	0.0374	0.0566	0.0825
Avg_error	0.00033	0.01583	0.00125	0.00531	0.01950
Best_error	0	0.0012	0.0001	0.0005	0.0006
Worst_error	0.0187	0.9821	0.2741	0.3354	0.4021

注：*表示性能最好的模型。

图16至图23为与KELM相关的预测模型的部分样本真实值与预测值的结果展示图和真实值及预测值的误差与0值的对比图。从对比实验结果图可以看出，本文所搭建的预测模型预测精确度最高，且误差几乎为与0线拟合，并且通过优化得到的最佳KELM参数组合为 $c=476.008$ ， $g=29.2314$ 。

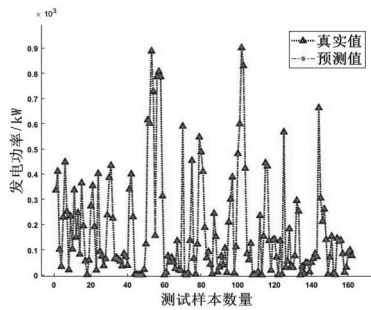


图16 WOA-MVO-KELM预测结果图

Fig.16 Prediction results diagram of WOA-MVO-KELM

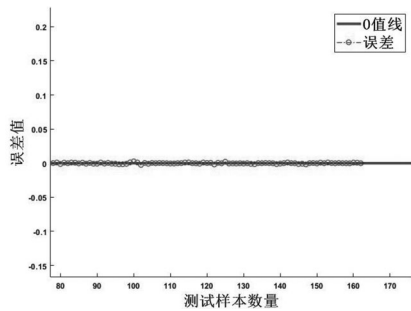


图17 WOA-MVO-KELM误差图

Fig.17 WOA-MVO-KELM error diagram

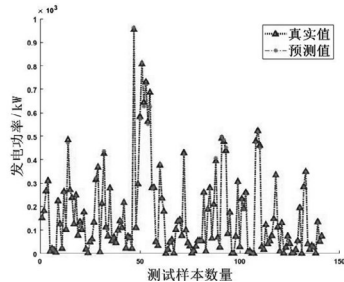


图18 MVO-KELM预测结果图

Fig.18 Prediction results diagram of MVO-KELM

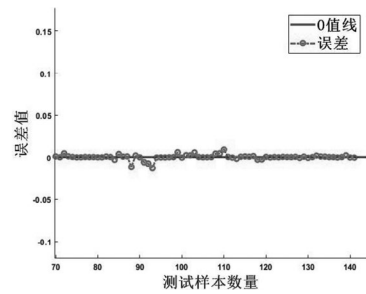


图19 MVO-KELM误差图

Fig.19 MVO-KELM error diagram

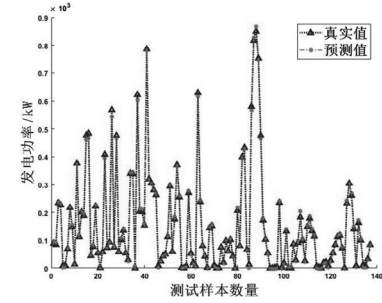


图20 PSO-KELM预测结果图

Fig.20 Prediction results diagram of PSO-KELM

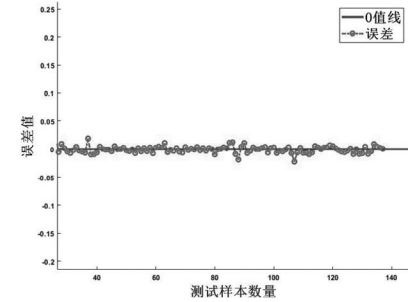


图21 PSO-KELM误差图

Fig.21 PSO-KELM error diagram

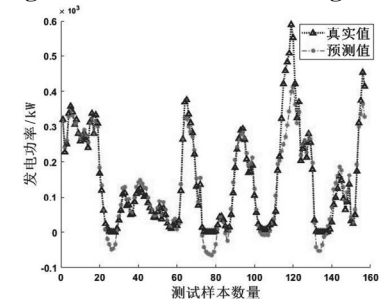


图22 KELM预测结果图

Fig.22 Prediction results diagram of KELM

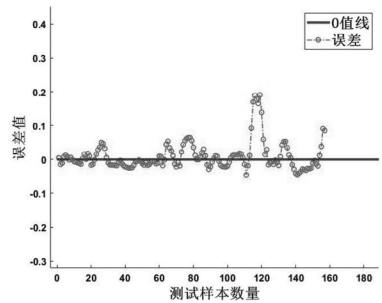


图23 KELM误差图

Fig.23 KELM error diagram

(2)稳定性分析

本文通过使用箱型图来分析算法的稳定性,箱型图最大的优点就是不受异常值的影响,能够准确稳定地描绘出数据的离散分布情况。每种算法经过10次运行,将每一次运算得到的均方根误差记录下来,最后将每个模型的10次结果绘制成两张箱型图,一张是WOA-MVO-KELM模型与其他KELM相关预测模型的对比,如图24所示;另一张是WOA-MVO-KELM模型与其他预测模型的对比,如图25所示。

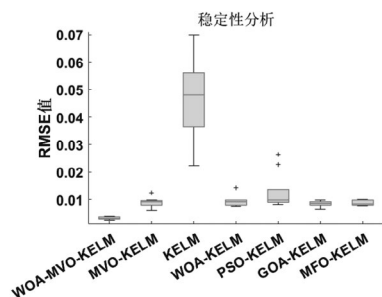


图24 均方根误差分布图(与其他KELM相关预测模型对比)

Fig.24 RMSE distribution diagram (Compared with other KELM related prediction models)

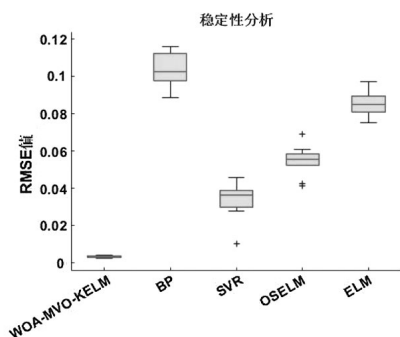


图25 均方根误差分布图(与其他预测模型对比)

Fig.25 RMSE distribution diagram (Compared with other prediction models)

图24和图25中的箱型代表实验结果的分布,在箱子内的值以中间那条中位线为界限分成上四分位数和下四分位数,箱子外上下两根线表示的是上限和下限,“+”代表的是该数据中的异常值。由这两张图可以看出,本文所组建的预测模型稳定性最强,因为其上下四分位数值间距极小且非常对称紧凑,也没有异常值,而其他算法的稳定性则参差不齐。本文还分析了WOA-MVO-KELM与其他算法的收敛曲线,如图26所示。

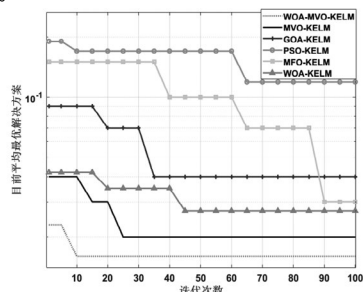


图26 WOA-MVO-KELM与其他算法的收敛曲线

Fig.26 Convergence curve of WOA-MVO-KELM and other algorithms

到目前为止的平均值表示在10次运行的每次迭代中获得的最佳解决方案的平均值。从图26可以看出, MVO优化算法的模型经过五次左右迭代后才收敛到最优,这可能是由于MOV在避免局部最优时,在迭代的初始步骤中没有找到一个好的解决方案进行利用,所以该算法一直在搜索空间中搜索,寻找好的解决方案收敛于它们,但是通过WOA算法对其初始种群的优化,发现WOA-MVO-KELM模型收敛次数在10次以下并且收敛到的结果最优,其他算法收敛速度慢或收敛结果差。通过对这些数据的分析,可以看到WOA-MVO-KELM算法可以快速收敛且收敛的结果最好。

4 结论(Conclusion)

本文针对目前短期风力发电预测的精确度不理想、效率低等一系列问题,设计了一个基于WOA-MVO混合优化算法的KELM预测模型。首先为解决KELM模型的随机参数问题,本文选用了MVO算法进行最优参数组合搜寻,在搜索之前先使用WOA算法对MVO算法的初始群数进行优化,得到一个较为优秀的种群代入WOA算法中。然后将最优参数组合放回KELM模型里进行训练,最终成功构建了一个新型预测模型。通过对真实风电场历史数据实验结果的分析,本文采用的WOA-MVO-KELM预测模型相较于ELM、BP神经网络、SVR等其他现存的预测模型具有更高的精度和良好的泛化能力。

参考文献(References)

- [1] 马晓博.基于小波变换和BP神经网络的短期风电功率预测[J].电力科学与技术学报,2015,30(02):92-97.
- [2] 王涛,高靖,王优胤,等.基于改进经验模态分解和支持向量机的风电功率预测研究[J].电测与仪表,2021,58(06):49-54.
- [3] 何廷一,田鑫森,李胜男,等.基于蜂群算法改进的BP神经网络风电功率预测[J].电力科学与技术学报,2018,33(04):22-28.
- [4] 曹天行,刘三明,王致杰,等.基于集合经验模态和深度学习组合的风电场功率短期预测研究[J].电测与仪表,2018,55(13):84-88.
- [5] HUANG G B. An insight into extreme learning machines: Random neurons, random features and kernels[J]. Cognitive Computation, 2014, 6(3):376-390.
- [6] ANKUR S, THARO S, KALYANMOY D. Using Karush-Kuhn-Tucker proximity measure for solving bilevel optimization problems[J]. Swarm and Evolutionary Computation, 2019, 44:496-510.
- [7] MIRJALILI S, MIRJALILI S M, HATAMLOU A. Multi-verse optimizer: A nature-inspired algorithm for global

- optimization[J]. *Neural Computing and Applications*, 2015, 27(2):495–513.
- [8] MIRJALILI S, LEWIS A. The whale optimization algorithm[J]. *Advances in Engineering Software*, 2016, 95:51–67.
- [9] RIENECKER M M, SUAREZ M J, GELARO R, et al. MERRA: NASA's modern-era retrospective analysis for research and applications[J]. *Journal of Climate*, 2011, 24(14):3624–3648.
- [10] MÜLLER R, UWE P, CHRISTINE T C, et al. Digging the METEOSAT treasure—3 decades of solar surface radiation[J]. *Remote Sensing*, 2015, 7(6):8067–8101.
- [11] PAL M, MAXWELL A E, WARNER T A. Kernel-based extreme learning machine for remote sensing image classification[J]. *Remote Sensing Letters*, 2013, 4(9):835–862.
- [12] LI L L, CHANG Y B, TSENG M L, et al. Wind power prediction using a novel model on wavelet decomposition—support vector machines—improved atomic search algorithm[J]. *Journal of Cleaner Production*, 2020, 270(6):121817.
- [13] LIANG N Y, HUANG G B, SARATCHANDRAN P, et al. A fast and accurate online sequential learning algorithm for feedforward networks[J]. *IEEE Transactions on Neural Networks*, 2006, 17(6):1441–1423.

作者简介:

宣 畅(1998–), 男, 硕士生. 研究领域: 神经网络, 机器学习, 机器人设计.

(上接第57页)

- [2] SHEN W, MAI Y, SU X, et al. A new electric hydraulic actuator adopted the variable displacement pump[J]. *Asian Journal of Control*, 2016, 18(1):178–191.
- [3] 麦云飞, 童骏民. 基于分数阶控制的液压加载系统设计与仿真[J]. *上海理工大学学报*, 2015, 37(6):589–593.
- [4] JIANG J, LU H, ZHOU R, et al. Development of hydraulic transformer in constant pressure rail system[J]. *Journal of Southeast University (Natural Science Edition)*, 2006, 36(5):869–874.
- [5] DING R, ZHANG J, XU B, et al. Programmable hydraulic control technique in construction machinery: Status, challenges and countermeasures[J]. *Automation in Construction*, 2018, 95:172–192.
- [6] 闫丽娟, 孙辉, 刘伟, 等. 行走工程机械液压混合动力技术[J]. *吉林大学学报(工学版)*, 2014, 44(2):364–368.
- [7] SHEN W, WANG J, HUANG H, et al. Fuzzy sliding mode control with state estimation for velocity control system of hydraulic cylinder using a new hydraulic transformer[J]. *European Journal of Control*, 2019, 48:104–114.
- [8] CHANDRA K, ALWI H, EDEARDS C. Fault detection in uncertain LPV systems with imperfect scheduling parameter using sliding mode observers[J]. *European Journal of Control*, 2017, 34:1–15.
- [9] AHN K, NAM D, JIN M. Adaptive backstepping control of an electrohydraulic actuator[J]. *IEEE/ASME Transactions on Mechatronics*, 2014, 19(3):987–995.
- [10] AZIMI H, BONAKDARI H, EBTEHAJ I, et al. A combined adaptive neuro-fuzzy inference system—firefly algorithm model for predicting the roller length of a hydraulic jump on a rough channel bed[J]. *Neural Computing and Applications*, 2018, 29(6):249–258.
- [11] LU H, JIANG J, WANG F. Study on hydraulic transformer driving linear load system based on fuzzy tuning PID control strategy[J]. *Machine Tool & Hydraulics*, 2009, 37(1):58–61.
- [12] SHEN W, WANG J. Adaptive fuzzy sliding mode control based on Pi-sigma fuzzy neural network for hydraulic hybrid control system using new hydraulic transformer[J]. *International Journal of Control Automation and Systems*, 2019, 17(7):1708–1716.
- [13] 沈伟, 刘帅, 武毅. 基于网络的液压马达伺服位置系统自适应鲁棒积分控制[J]. *上海理工大学学报*, 2021, 43(4):325–331.

作者简介:

柳金利(1997–), 男, 硕士生. 研究领域: 液压控制工程.