

基于 BO-WT-VMD 组合模型矿井提升机钢丝绳的缺陷分析

摘 要

在矿业生产过程中，提升机负责运送人员、设备、煤炭和各类物料，是连接地上与地下整个生产系统的重要纽带。然而，在矿井提升钢丝绳运行过程中，由于其长期处于超负荷状态，极易出现疲劳、老化、磨损和结构断丝，对整个矿井提升系统的安全运行造成严重的威胁。如何在不损害钢丝绳性能及不影响钢丝绳使用的前提下，通过信号建立一个有效的钢丝绳故障监测模型具有重要意义。

针对问题一，为了准确识别钢丝绳缺陷的故障数据，消除被测信号中的噪音，本文综合利用**基于贝叶斯优化-小波分析-变分模态分解(BO-WT-VMD)的组合降噪模型**进行降噪。具体地：首先利用小波变换（WT）处理信号，为了对低频信号漂移进行捕捉，利用变分模态分解算法(Variational Mode Decomposition, VMD)对小波变换处理后的信号进行模态分解，通过计算分解的各个模态与原信号之间的相关系数，完成信号的模态选择和重构。然而，VMD 算法的模态数目 K 的大小对序列分解和预测精度都有较大影响。因此，本文采用贝叶斯优化算法(BO)对 VMD 算法的模态数目 K 以及的 α 值进行寻参，找到最优超参数组合，有效提升了算法性能。并且在仿真实验和真实数据中，BO-WT-VMD 算法的 SNR 和 RMSE（分别为 17.3403，0.1010）都略强于 WT（15.6170，0.1232）和 VMD（15.6492，0.1227），说明该算法能够有效地去除信号噪音。

为了监测去噪信号的故障数目，位置与程度，本文采取**局部离群因子（LOF）算法**对去噪信号进行分析，通过判断每个点输出的局部离群因子找出故障信号点并统计，利用量化定义公式计算出损伤程度与位置。以 1 号钢丝为例：共检测出 1827 个异常点，其中微小断丝，变形，轻度断丝，中度断丝和内部断股的数目分别为 185，27，105，849，661，其故障位置为 19，554，765，878，....333989。

针对问题二，本文首先选择了 24 个特征指标，包括 11 个时域统计特征以及 13 个频域统计特征，通过傅里叶变换实现频域与时域的相互转化。然后，使用 SPSSPRO 上的**熵权-TOPSIS 模型**，对 6 根钢丝绳的 10 次往返的安全性能进行综合评价，得到安全性能评价值，根据其值得出排名为：2 号钢丝绳(0.5528)>1 号钢丝绳(0.4142)>3 号钢丝绳(0.2296)>6 号钢丝绳(0.1799)>4 号钢丝绳(0.1411)>5 号钢丝绳(0.1359)。

关键词：磁漏检测 小波分析 变分模态分解 局部异常因子检测 熵权-TOPSIS

目 录

一、 问题重述	1
1.1 问题背景.....	1
1.2 问题的提出.....	1
二、 问题分析	1
2.1 问题一的分析.....	1
2.2 问题二的分析.....	2
三、 模型假设	2
四、 符号说明	2
五、 模型的建立与求解	3
5.1 问题一模型的建立与求解.....	3
5.1.1 基于贝叶斯优化-小波分析-变分模态分解组合的降噪模型.....	3
5.1.2 仿真实验.....	8
5.1.3 真实数据检验.....	10
5.1.4 基于 LOF 的异常信号监测	10
5.2 问题二模型的建立与求解.....	13
5.2.1 问题概述.....	13
5.2.2 构建特征.....	13
5.2.3 基于熵权-TOPSIS 的钢丝绳安全量化评价模型	15
六、 模型评价	17
6.1 模型的优点.....	17
6.2 模型的缺点.....	17
参考文献	17
附录	19

一、问题重述

1.1 问题背景

矿井提升机是由电机带动机械设备，带动钢丝绳使容器在井筒中升降，以完成输送任务。为使得矿井提升过程更加安全，钢丝绳的全面安全监控成为了矿井物料搬运任务的关键。目前，基于“磁平衡”的检测技术已经广泛应用于钢丝绳的故障检测工作中。该技术能通过数据信号进行分析，在不损坏钢丝绳性能以及不影响钢丝绳的日常使用的前提下，监测出钢丝绳是否出现局部断丝、变形、磨损甚至钢丝绳大范围受损等故障，从而进行全面安全监控，并能明显提高钢丝绳维护效率，进而减少维护成本。因此，探索有效且高效检测钢丝绳损伤的检测方法和损伤判别的研究是十分必要和迫切的^[1-2]。

1.2 问题的提出

由于钢丝绳存在断丝、变形、磨损、锈蚀等缺陷故障，而钢丝绳在监测时有某种程度的晃动影响，这都使监测数据产生误差，出现不同程度的“噪音”，进而导致误报，降低了钢丝绳的维护效率。建立数学模型及设计相应算法以减少“磁平衡”检测数据中“噪音”并给出钢丝绳安全性能的量化评价方法是必要的。

根据题目要求，本文需要解决如下问题：

- (1) 针对题目所给的附件一的监测数据，建立数学模型与设计算法，去除钢丝绳在井筒升降过程中的噪音，并找出钢丝绳完成 10 次往返后的缺陷数量、缺陷程度以及缺陷位置。
- (2) 结合问题一的结果，建立数学模型，对钢丝绳的安全性能进行量化评价，并计算附件 1 中 6 根钢丝绳 10 次往返后的安全性能评价价值。

二、问题分析

2.1 问题一的分析

为了准确识别并提取钢丝绳缺陷信号特征，可利用小波分析对所给信号进行去噪分析，而小波变换方法对低频信号漂移，不能及时过滤低频漂移噪声信号。而变分模态分解(Variational Mode Decomposition, VMD)算法^[3]也可进行去噪分析，但是算法中的 K 和 α 的取值根据实际情况的不同而借助个人经验实现预设，欠缺可靠性，具有很强的主观性。因此，本文将使用贝叶斯优化来改进 VMD 算法，解决 VMD 算法的不足之处。由

于以上小波分析与改进后的 VMD 算法的存在缺陷, 本文综合利用基于贝叶斯优化-小波分析-变分模态分解(BO-WT-VMD)的组合降噪模型, 对原始信号进行去噪处理。为了找出去噪信号故障的程度、位置和数量, 本文采取局部离群因子检测技术(Local Outlier Factor, LOF), 对于一段去噪信号, 通过计算每点在相应区间的局部离群因子进行故障检测, 从而确认故障点的位置、数目和程度。

2.2 问题二的分析

为了对钢丝绳的安全性能进行量化评价, 本文使用熵权-TOPSIS 模型对 6 根钢丝绳的 10 次往返进行评价, 得到钢丝绳 10 次往返的评价值。结合生活实际, 钢丝绳下行之后必须上行, 因此将相邻两次的往返的安全性能评价取平均值, 得到 5 趟来回的安全性能评价。为得到最终的安全性能评价, 本文再使用专家赋权法, 对每一次往返进行赋权, 最终通过计算得到钢丝绳的安全性能评价。

三、模型假设

为了能够合理的简化问题, 我们在模型建立的过程中提出如下假设:

- 忽略传感器的测量误差。
- 假设测量数据是连续的。
- 根据钢丝绳长 960 米, 0.4 米一个脉冲, 计算得到应有 2400 个脉冲, 大于数据文件中的 1816 个脉冲, 故假设钢丝绳长有某一段未测量。
- 假设钢丝绳的缺陷会随运行次数的增加而累加。

四、符号说明

符号	说明	符号	说明
a	尺度因子	$\phi_k(t)$	相位函数
b	时移因子	ω_k	中心频率
N	数据长度	μ_k	模态形态
$f(i\Delta s)$	输入信号	$\mu_k(t)$	调幅调频函数
$Y(n)$	通过距离采样的输入信号	C	相关系数
$H(n)$	详细高频信息	$Cov(x_i, y_i)$	协方差
$L(n)$	一般低频信息	σ_i	标准差
α	二次惩罚因子	ρ	相似度阈值

符号	说明	符号	说明
$\delta(t)$	冲激函数	ε	判别阈值
K	分解尺度	l	第 l 个脉冲
m	第 l 个脉冲共有 m 帧	n	第 l 个脉冲的第 n 个 帧

五、模型的建立与求解

5.1 问题一模型的建立与求解

5.1.1 基于贝叶斯优化-小波分析-变分模态分解组合的降噪模型

小波变换（WT）是一种被广泛使用的无线检测信号预处理方法，可用于提取奇异损伤信号并通过对时间和空间频率的局部分析滤除干扰信号^[4]。

变分模态分解（VMD）算法通过非递归的处理方法来构造并求解约束变分问题，从而实现对原信号的分解。这种方法有效地避免了 EMD 分解中出现的模态混叠、边界效应等问题，对于分解复杂信号也有很好的效果，并且有一定的抗噪声干扰的能力。

本文将使用贝叶斯优化（BO）来改进 VMD 算法，综合利用基于贝叶斯优化-小波分析-变分模态分解(BO-WT-VMD)组合降噪模型，对原始信号进行去噪处理。其大致过程如下：

首先，利用小波变换确定小波层数，再使用小波变换对原始信号进行去噪。其次，利用贝叶斯优化确定 VMD 的参数 K 与 α ，（即，首先随机初始化超参数组合与先验数据集，通过高斯模型 GP 增加采样点，得到后验数据集，若后验数据集满足终止条件，则确定优化后的参数组合，得到小波变换的参数；若后验数据集不满足终止条件，则通过循环高斯模型及后面的步骤，直至后验数据集满足终止条件，确定优化后的参数组合，得到 VMD 的参数 K 与 α ）。最后，将所得参数 K 与 α 的值带入 VMD 算法中，得到分解后的 K 个模态，然后通过皮尔逊系数选择最优的模态，剔除剩余模态，最终得到去噪后的重构数据。具体框架图如下：

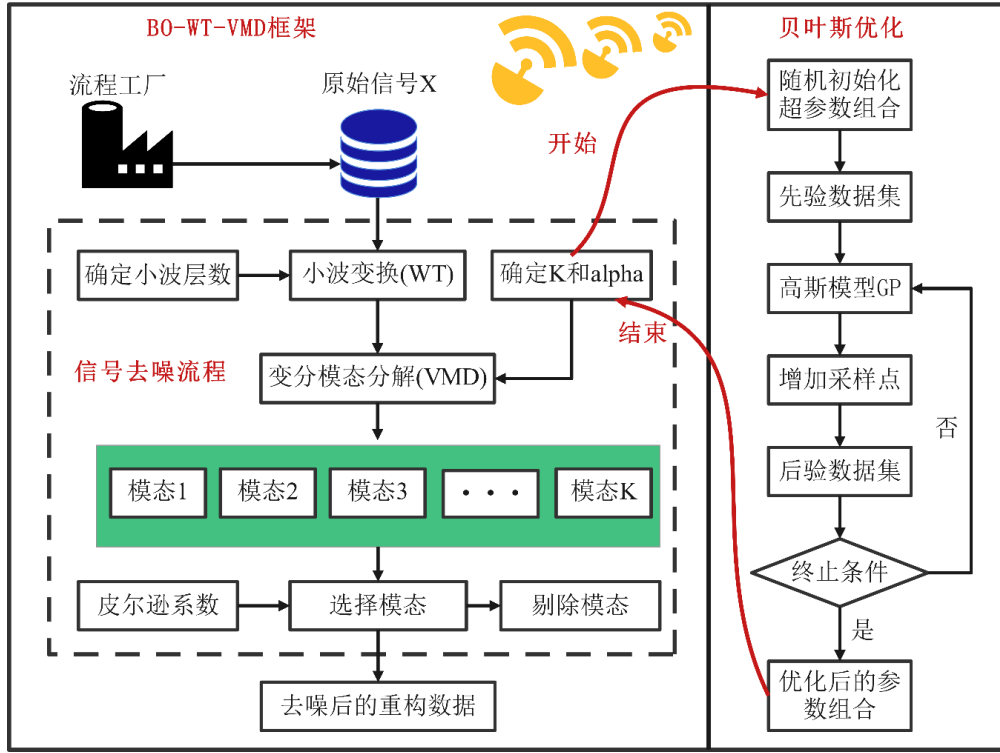


图 1 BO-WT-VMD 框架图

(1)基于小波变换的去噪方法

小波分析是将信号与小波函数进行卷积，把信号分解成不同频段和时段成分，是一种变分辨率的时频分析方法。

假设 $f(x)$ 是平方可积函数的一个函数，即 $f(x) \in L^2(R)$ 。然后将 $f(x)$ 的小波变换定义为：

$$\begin{aligned} W_f(a, b) &= \langle f, \psi_{a,b}(x) \rangle \psi_{a,b}(x) dx \\ &= \frac{1}{\sqrt{a}} \int_{-\infty}^{+\infty} f(x) \psi_{a,b}(x) dx = \frac{1}{\sqrt{a}} \int_{-\infty}^{+\infty} f(x) \psi\left(\frac{x-b}{a}\right) dx \end{aligned} \quad (1)$$

式中， a 和 b 分别是尺度因子和时移因子， $\psi_{a,b}(x)$ 是一个小波函数序列。

钢丝绳检测信号为一维离散信号。因此，在数值计算中，需要将小波变换的尺度因子和时移因子分开。

设比例因子 $a = a_0^m, b = k a_0^n b_0$ ，其中 $a_0 > 1, b \neq 0$ ， m 与 n 均为整数。小波函数为

$$h_{m,n}(x) = \frac{1}{\sqrt{a_0^m}} h\left(\frac{1}{a_0^m} x - nb\right) \quad (2)$$

当 h, a_0, b_0 取合适值时， $h_{m,n}(x)$ 构成标准正交基，正交基为 sym8。阈值公式为软硬阈值折中的方法，公式具体如下：

$$T = \frac{\text{median}(|cD1|)}{0.6745} \sqrt{2 \ln N} \quad (3)$$

式中， $cD1$ 为第一层分解的细节系数， N 为数据长度。

同时，小波变换采用如下的阈值函数进行变换：

$$\tilde{w}_{k,j} = \begin{cases} w_{j,k} - a\lambda, & w_{j,k} \geq \lambda \\ 0, & |w_{j,k}| < \lambda \\ w_{j,k} + a\lambda, & w_{j,k} \leq -\lambda \end{cases} \quad (4)$$

式中， a 为 0 时为硬阈值函数， a 为 1 时为软阈值函数。由于本文采取软硬阈值折中的方法，故 a 取值为 0.5。

为了有效地过滤钢丝绳的损伤干扰信号，实现煤矿提升钢丝绳的在线检测，本研究设计了钢丝绳损伤信号的预处理模型。处理模型如图 2 所示：

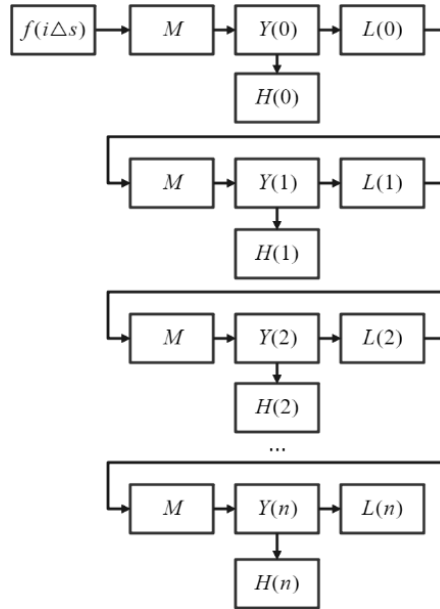


图 2 小波变换具体流程图

其中， $f(i\Delta s)$ 表示通过距离采样的输入信号， M 表示形态滤波器， $Y(0), Y(1), \dots, Y(n)$ 表示由形态滤波器处理的信号。 $H(i)$ 和 $L(i)$ 分别表示经过第 i 层小波变换后的故障信号的详细高频信息和一般低频信息。根据图 2 的结构，钢丝绳损伤信号的预处理模型在小波变换的每一层中都有一个附加的形态学滤波器。此外，利用形态滤波器，去除了操作过程中由于钢丝绳摇晃而产生的一维钢丝绳损伤信号的漂移信号。然后，利用小波变换方法进行进一步分析，具体步骤如下：

- 对噪音信号进行小波分解。选择对应的分解尺度与小波基，进行小波分解，得到一组小波系数。
- 对小波分解的各层高频系数进行阈值量化，从而得出小波系数的估计值。

- 对已阈值量化处理的小波系数，进行逆小波变换以重构信号，得到去噪信号。

(2)基于 VMD 算法的去噪方法

变分模态分解 (Variational Modal Decomposition, VMD)^[12]算法是对变分问题的一种求解过程，该算法所进行的自适应信号分解过程是通过求解寻找约束变化模型的最优解来实现的。在该算法模型中，将输入信号 $x(t)$ 根据预设的分解尺度 K 以及二次惩罚因子 α 被分解成 K 个稀疏模式的 IMF 分量，用 $u_k(t)$ 表示。每个 IMF 分量都有一个中心频率的有限带宽，中心频率和带宽在分解过程中将会不断更新。每个 IMF 分量又被定义为调幅调频信号，即

$$u_k(t) = A_k(t) \cos(\phi_k(t)) \quad (5)$$

其中 $A_k(t)$ 是 $u_k(t)$ 的瞬时赋值， $\phi_k(t)$ 为其相位函数，其瞬时相位 $\omega_k(t) = \phi_k(t)$ 。

利用交替方向乘子表，计算增广拉格朗日函数的鞍点，得到最优解 $\{u_k\}, \{w_k\}$ 和 λ_k 。将原始信号 f 分解为 K 个 IMF 分量，具体步骤如下：

① 设 $n=0$ ，初始化 $\{\mu_k^1\}$ ， $\{w_k^1\}$ 和 λ^1 ， n 为迭代次数；

② 当 $n=1$ ，循环开始，并更新 $\{\mu_k\}$ ， $\{w_k\}$ 和 λ_k 。

(a) 当 $w \geq 0$ 时，迭代更新 $\{u_k\}$ 的值，具体数学表达式为：

$$\hat{u}_k^{n+1}(w) = \frac{\hat{f}(w) - \sum_{i \neq k} \hat{u}_i(w) + \hat{\lambda}(w)/2}{1 + 2\alpha(w - w_k)^2} \quad (6)$$

(b) 迭代更新 $\{w_k\}$ 的值，具体迭代为：

$$w_k^{n+1} = \frac{\int_0^\infty w |\hat{u}_k(w)|^2 dw}{\int_0^\infty |\hat{u}_k(w)|^2 dw} \quad (7)$$

(c) 当 IMF 分量及其中心频率被更新时，拉格朗日乘数 λ 也被更新：

$$\hat{\lambda}^{n+1}(w) = \hat{\lambda}^n(w) + \tau \left[\hat{f}(w) - \sum_k \hat{u}_k^{n+1}(w) \right] \quad (8)$$

③ 设 $n = n + 1$ ，重复迭代更新 μ_k, w_k ，直至满足一下收敛条件：

$$\frac{\sum_k \|u_k^{n+1} - u_k^n\|_2^2}{\|u_k^n\|_2^2} < \varepsilon \quad (9)$$

其中 ε 为给定的判别阈值。满足约束条件时，停止迭代并输出 K 个模态分量。

(3)确定最佳模态分量

利用如图 2 所示提出的小波包与 VMD 相结合的降噪方法流程图，对采集到的声波信号利用小波包分析进行降噪预处理，确定小波包分解层数、小波基函数及阈值函数，得到初步处理后的去噪信号；其次，对初步处理后的信号进行 VMD 分解，得到 K 个 IMF 分量。根据式 (10) 相关系数准则计算每个分量的相关系数，选择阈值 $\rho > C_{\max}/(10C_{\max} - 3)$ 的 IMF 分量判定为有效分量，其中 C_{\max} 为各分量与原始信号的最大相关值；最终，提取有效 IMF 分量并进行信号重构，得到去噪后的重构信号即为最终处理后的去噪泄露信号。

相关系数是指去噪后信号与原始信号之间的相似度：

$$C = \frac{Cov(x_i, y_i)}{\sigma_x \sigma_y} \quad (10)$$

式中 $Cov(x_i, y_i)$ 和 σ_x, σ_y 分别为 x_i 和 y_i 的协方差和标准差。 C 越趋于 1，去噪效果就越好。

(4)贝叶斯优化算法

贝叶斯优化(Bayesian Optimization, BO)是在函数方程未知的情况下根据已有的采样点预估函数最大值的一种算法，其目标是找到全局最优解^[7]。它有效地解决了序贯决策理论中经典的机器智能问题，能通过较主动选择相对好的策略来确定下一个最有“潜力”的评估^[8]。在贝叶斯优化算法中，高斯过程能建立与目标函数近似的模型，并能获取下一个评估的位置。在某些场景下，合理选用高斯过程可以有效改进贝叶斯优化结果。

若设待优化模型的超参数集合为 $X = \{x_1, x_2, \dots, x_n\}$ ，贝叶斯优化算法的目标函数为 $f(x)$ ，则贝叶斯优化算法遵循如下的二项分布：

$$f(x) \sim GP(\mu, k(x, x')) \quad (11)$$

式中， GP 为高斯分布， μ 为均值， $k(x, x')$ 为协方差函数。

利用贝叶斯优化算法，经过 t 次迭代试验后可得到数据 $(x_t, f(x_t))$ ，此时有数据集 $D_t = \{(x_1, f(x_1)), (x_2, f(x_2)), \dots, (x_t, f(x_t))\}$ ，然后，我们便需要预测集合第 $t+1$ 处的数据 $(x_{t+1}, f(x_{t+1}))$ 。普遍认为第 $t+1$ 处的点分布在某个 $t+1$ 维度的高斯分布中，也就是：

$$\begin{bmatrix} f_{1:t} \\ f_{t+1} \end{bmatrix} \sim GP\left(\mu, \begin{bmatrix} K & k^T \\ k & k(x_{t+1}, x_{t+1}) \end{bmatrix}\right) \quad (12)$$

其中， K 为协方差函数矩阵， $k = [(x_{t+1}, x_1), (x_{t+1}, x_2), \dots, (x_{t+1}, x_t)]$ ，由此可以得出 f_{t+1} 的分布，即

$$P(f_{t+1}|D, x_{t+1}) = GP(u(x_{t+1}), \delta^2(x_{t+1})) \quad (13)$$

其中 $\mu(x_{t+1}) = kK^{-1}f_{1:t}$ ，它表示预测均值； $\delta^2(x_{t+1})$ 为预测协方差，其值为 $k(x_{t+1}, x_{t+1}) - kK^{-1}f^T$ 。由此可得集合 $t+1$ 处的数据 $(x_{t+1}, f(x_{t+1}))$ 。

以下为贝叶斯优化算法以伪代码形式的执行流程：

算法：贝叶斯优化算法流程

```

for  $t = 1, 2, \dots$ , do
    最大化采集函数，得到下一个评估点：  $x_t = \arg \max_{x \in \chi} \alpha(x|D_{1:t-1})$ ；
    评估目标函数值  $y_t = f(x_t) + \varepsilon_t$ ；
    整合数据：  $D_t = D_{t-1} \cup \{x_t, y_t\}$ ，并且更新概率代理模型；
end for

```

5.1.2 仿真实验

为了对本文提出的 BO-WT-VMD 降噪方法的可行性进行分析，不失一般性，先对仿真信号进行降噪分析，仿真信号表达式为：

$$\begin{cases} y_1(t) = \cos 4\pi t \\ y_2(t) = 0.5 \cos 48\pi t \\ y_3(t) = 0.25 \cos 576\pi t \\ y(t) = y_1(t) + y_2(t) + y_3(t) + n \end{cases} \quad (14)$$

其中，信号频率为 2、24、288。观测信号由不同分贝的高斯白噪声和有线信号叠加而成。评价指标属于比较客观的参数指标：信噪比（SNR）和均方误差（RMSE）。SNR 表示去噪能力，SNR 值越大越好；RMSE 表示去噪前后的信号幅值的差异，RMSE 越小越好。

为比较不同算法的去噪效果，本文分别使用 VMD、WT、BO-WT-VMD 算法，对仿真信号进行降噪，几种算法去噪性能结果如下：

表 1 多种算法降噪性能表

	VMD	WT	BO-WT-VMD
SNR	15.6492	15.6170	17.3403
RMSE	0.1227	0.1232	0.1010

通过对比上表数据可知，在三个降噪算法中，BO-WT-VMD 算法的 SNR 值最大，为 17.3403，RMSE 值最小，为 0.1010。故可以说明 BO-WT-VMD 模型对“噪声”数据的降噪性能更好。

仿真函数值的降噪效果如下图所示：

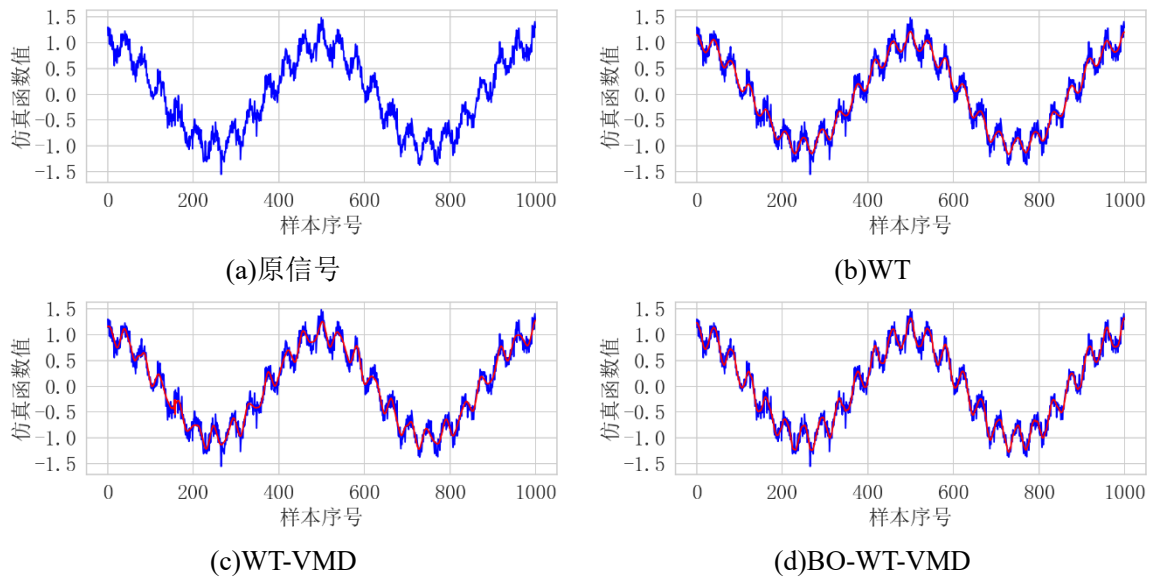


图 3 仿真函数值的降噪前后对比图

通过分析上图可知，WT 算法无法有效返回仿真函数的有效信息；WT-VMD 算法在某些极值点无法很好的拟合；相比之下，BO-WT-VMD 算法很好弥补了以上算法的不足，对仿真函数值的降噪效果优秀，因此 BO-WT-VMD 算法具有良好的降噪效果。模型对信号的模态分解结果如下：

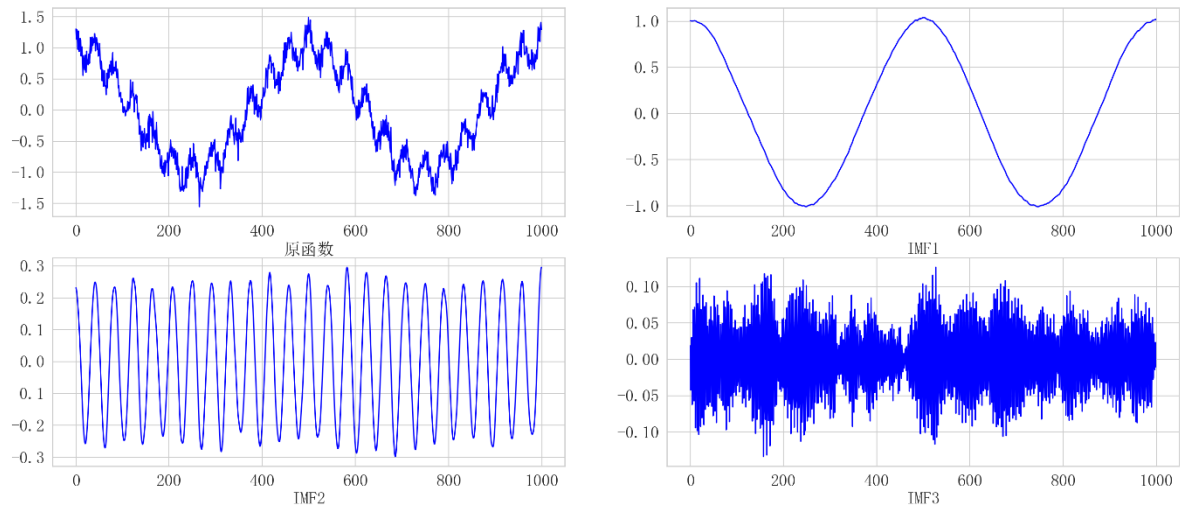


图 4 模态分解结果图

5.1.3 真实数据检验

为检验 BO-WT-VMD 算法在本题信号的降噪效果，本文抽取分别从 1 号电压和 2 号电压中抽取样本序号为 5000~6000 的电压，使用 BO-WT-VMD 算法对其进行降噪预处理，并分析其降噪效果，降噪前后对比图如下图所示：

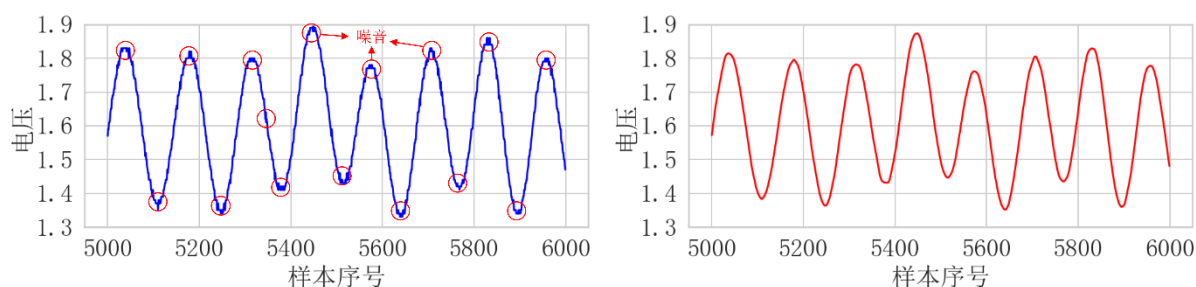


图 5 1 号电压降噪前后对比图

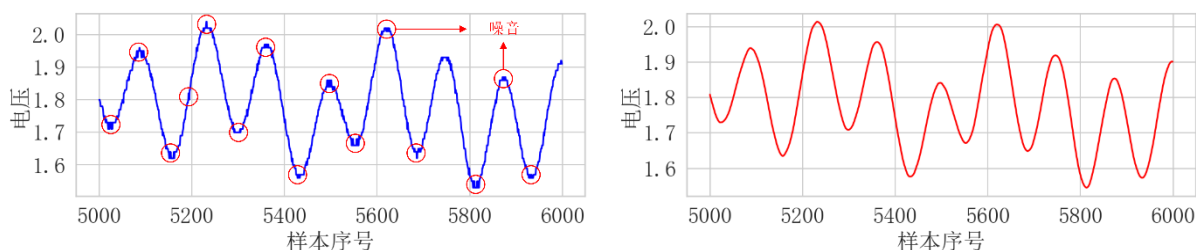


图 6 2 号电压降噪前后对比图

通过分析上图可知，BO-WT-VMD 算法对本题所给信号数据也具有优秀的降噪效果，因此，BO-WT-VMD 算法可用于本文数据的降噪。

5.1.4 基于 LOF 的异常信号监测

LOF 算法主要涉及的概念有数据对象的 k -距离、 k -距离邻域、数据对象的可达距离、可达密度和局部离群因子。相关概念定义如下：

1. 对象 p 的 k 距离 $d_i(p)$ ：设 k 为一正整数，数据对象 p 的 k 距离记作 $d_i(p)$ 。在数据集 D 中，将两个数据对象 p 与 o 的距离记作 $d(p, o)$ 。

若使得 $d_k(p) = d(p, o)$ ，需满足：

- 1) 数据集 D 中至少存在不包括 p 的 k 个点 $o' \in C/\{p\}$ ，满足 $d(p, o') \leq d(p, o)$ 。
- 2) 数据集 D 中至多存在不包括 p 在内的 $k-1$ 个点 $o' \in C/\{p\}$ ，满足 $d(p, o') < d(p, o)$ 。

2. 对象 p 的第 k 距离邻域：数据对象 p 的 k 距离邻域，就是所有与 p 的距离小于等于

k 的距离的数据对象 o 的集合，即：

$$N_k(p) = \{o \in D | d(p, o) \leq d_k(p)\} \quad (15)$$

3. 可达距离：数据对象 p 和 o 的可达距离记为：

$$reach-dist_k(p, o) = \max\{d_k(p), d(p, o)\} \quad (16)$$

其中 k 为一正整数。

4. 局部可达密度：数据对象 p 的局部可达密度表示点 p 的第 k 邻域内到点 p 的平均可达距离的倒数，即：

$$lrd_k(p) = 1 / \left(\frac{\sum_{o \in N_k(p)} reach-dist_k(p, o)}{|N_k(p)|} \right) \quad (17)$$

5. 局部离群因子：数据对象 p 的局部离群因子表示点 p 的邻域 $N_k(p)$ 的局部可达密度与点 p 的局部可达密度之比的平均数，即：

$$LOF_k(p) = \frac{\sum_{o \in N_k(p)} \frac{lrd_k(o)}{lrd_k(p)}}{|N_k(p)|} \quad (18)$$

若 LOF 值越接近于 1，表明 p 与其邻域对象密度越接近，越可能与邻域同属一簇；若 LOF 值越大于 1，则说明 p 的密度小于其邻域点密度越多，越可能是异常点。

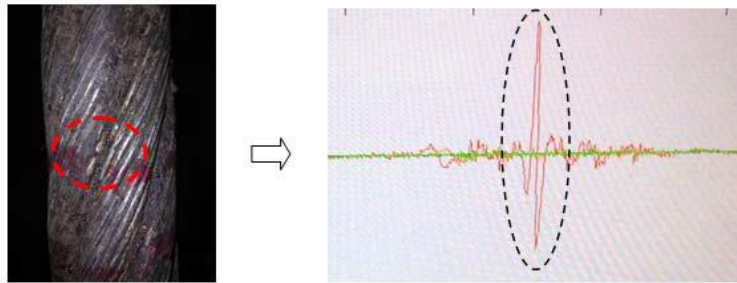


图 7 故障与信号对应图

通过图 7 可知，钢丝绳的故障信号会出现局部异常的信号^[10]。通过 LOF 算法，得到可能的异常点，为检验算法的合理性，本文通过分析 2 号电压的第 248000~250000 和第 240400~242400 个信号数据的电压变化趋势与局部异常因子的变化趋势（分别记为变化趋势图 a 和变化趋势图 b ），判断算法的可靠性。变化趋势见下图：

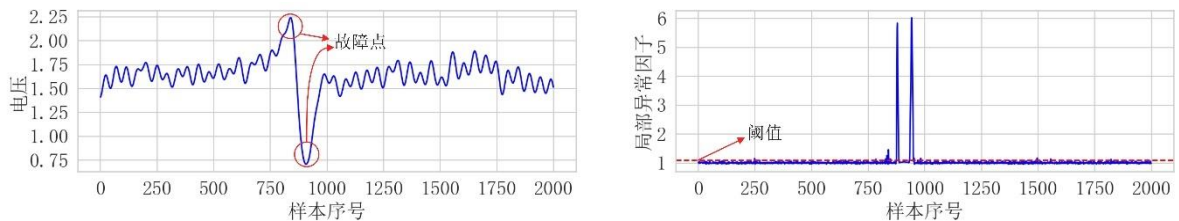


图 8 电压与局部异常因子变化趋势图 *a*

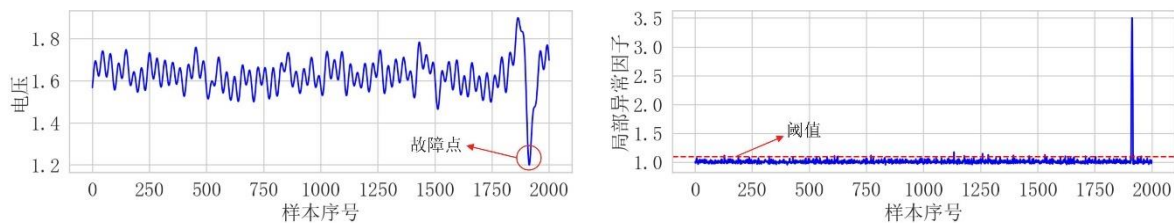


图 9 电压与局部异常因子变化趋势图 *b*

通过对上图的分析可知，电压发生突变的点，在对应的局部异常因子上也有明显的突变，故该算法对异常因子有极为准确的监测。通过 *LOF* 算法的监测，得到了 6 根钢丝绳在往返运行 10 次后的异常因子数，各钢丝绳的缺陷数量如图 11 所示：

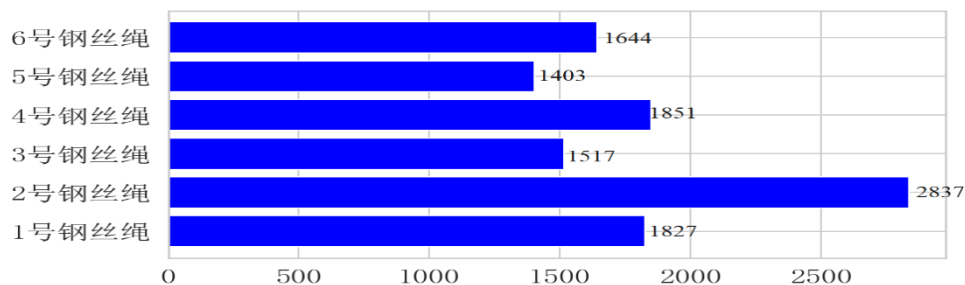


图 10 各钢丝绳的缺陷数量图

各钢丝绳不同程度的缺陷数量以及缺陷对应位置由表 2，表 3 所示：

表 2 各钢丝绳不同程度缺陷数量表

缺陷程度	1 号钢丝	2 号钢丝	3 号钢丝	4 号钢丝	5 号钢丝	6 号钢丝
微小断丝	185	216	147	170	121	135
变形	27	103	56	87	69	64
轻度断丝	105	301	187	210	182	200
中度断丝	849	1206	692	879	657	825
内部断股	661	1011	435	505	374	420

表 3 各钢丝绳缺陷所在位置

钢丝绳	缺陷位置(帧数)
1 号钢丝绳	19,554,765,878,973,1160,1524,1720,1761,1961,2121,2212,2559,..., 332781,332871,333193,333276,333334,333989
2 号钢丝绳	1,63,291,365,366,367,368,369,370,371,372,403,404,405,406,428,..., 333312,333325,334175,334178,334397,334693
3 号钢丝绳	214,615,682,746,794,799,863,867,1288,1375,1407,1464,1564,1578,...,

钢丝绳	缺陷位置(帧数)
4 号钢丝绳	333710,333975,334006,334480,334596,334669,334684,334687 38,148,336,525,659,770,837,872,1068,1212,1217,1222,1227,1232,..., 333256,333278,333482,333682,333866,333998,334514
5 号钢丝绳	37,346,482,765,879,1565,1591,1650,2107,2266,2268,2756,2762,..., 331960,332147,332189,332667,332793,333543,334301
6 号钢丝绳	609,624,936,992,1076,1104,1229,1543,1913,2006,2084,2146,..., 333319,333364,333782,334090,334697,334752

5.2 问题二模型的建立与求解

5.2.1 问题概述

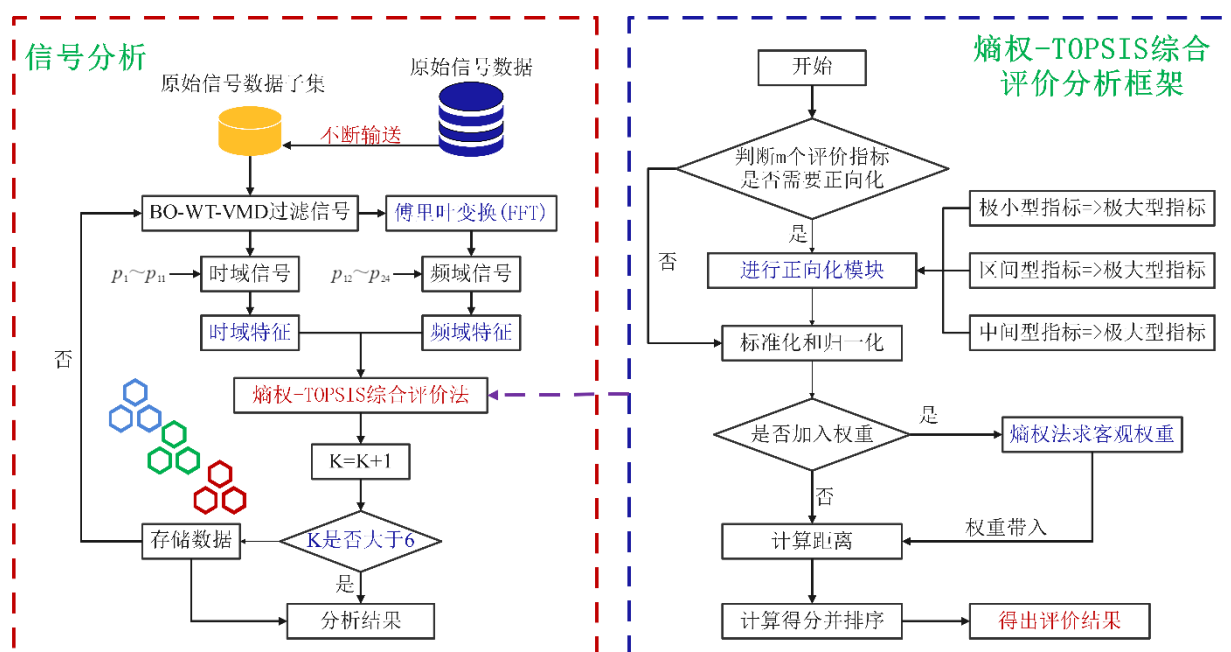


图 11 问题二流程框架图

针对此问题，我们首先利用问题一种建立的 BO-WT-VMD 算法来过滤原始信号数据，并通过傅里叶变换实现时域信号与频域信号的相互转化，进而提取出时域特征以及频域特征。同时，我们通过 SPSSPRO 上的熵权-TOPSIS 综合评价法对 6 根钢丝绳的 10 次往返的安全性能进行综合评价，然后采取专家权重对前面 10 次往返评价的结果进行加权求和，得到结果即安全性能评价值，最后再根据结果进行排名，找出安全性能最好的钢丝绳,流程框架图由图 12 可知。

5.2.2 构建特征

在问题二中，我们选择了 24 个特征指标 ($p_1 - p_{24}$)。11 个指标 ($p_1 - p_{11}$) 是时域

统计特征，13 个指标（ p_{12} - p_{24} ）是频域统计特征。所谓时域（Time domain），是描述数学函数或物理信号对时间的关系。而频域（frequency domain），是描述信号在频率方面特性时用到的一种坐标系。两者若相互转化，需要进行傅里叶变换

$$\widehat{f(k)} = \langle f, E_k \rangle = \int_{T^n} f(x) e^{-2\pi i k \cdot x} dx \quad (19)$$

当钢丝绳被检测出缺陷时，时域信号可能会发生变化。它的振幅和分布可能与正常条件下的时域信号不同。此外，频谱及其分布也可能发生变化，这意味着可能会出现新的频率分量，并且频谱的收敛性可能会发生变化^[9]。参数 p_1 和 p_3 - p_5 可以反映时域上的振动振幅和能量。参数 p_2 和 p_6 - p_{11} 可以表示信号在时域上的时间序列分布。参数 p_{12} 可以表示频域内的振动能。参数 p_{13} - p_{15} 、 p_{17} 和 p_{21} - p_{24} 可以描述频谱功率的收敛性。参数 p_{16} 和 p_{18} - p_{20} 可以表示主频率的位置变化，特征参数见表 4。

表 4 特征参数

时域特征指标		频域特征指标	
$p_1 = \frac{\sum_{n=1}^N x(n)}{N}$	$p_7 = \frac{\sum_{n=1}^N (x(n) - p_1)^4}{(N-1)p_2^4}$	$p_{12} = \frac{\sum_{k=1}^K s(k)}{K}$	$p_{19} = \sqrt{\frac{\sum_{k=1}^K f_k^4 s(k)}{\sum_{k=1}^K f_k^2 s(k)}}$
$p_2 = \sqrt{\frac{\sum_{n=1}^N (x_n - p_1)^2}{N-1}}$	$p_8 = \frac{p_5}{p_4}$	$p_{13} = \frac{\sum_{k=1}^K (s(k) - p_{12})^2}{K-1}$	$p_{20} = \frac{\sum_{k=1}^K f_k^2 s(k)}{\sum_{k=1}^K s(k) \sum_{k=1}^K f_k^4 s(k)}$
$p_3 = \left(\frac{\sum_{n=1}^N \sqrt{ x(n) }}{N} \right)^2$	$p_9 = \frac{p_5}{p_3}$	$p_{14} = \frac{\sum_{k=1}^K (s(k) - p_{12})^3}{K(\sqrt{p_{13}})^3}$	$p_{21} = \frac{p_{17}}{p_{16}}$
$p_4 = \sqrt{\frac{\sum_{n=1}^N (x(n))^2}{N}}$	$p_{10} = \frac{p_4}{\frac{1}{N} \sum_{n=1}^N x(n) }$	$p_{15} = \frac{\sum_{k=1}^K (s(k) - p_{12})^4}{Kp_{13}^2}$	$p_{22} = \frac{\sum_{k=1}^K (f_k - p_{16})^3 s(k)}{Kp_{17}^3}$
$p_5 = \max x(n) $	$p_{11} = \frac{p_5}{\frac{1}{N} \sum_{n=1}^N x(n) }$	$p_{16} = \frac{\sum_{k=1}^K f_k s(k)}{\sum_{k=1}^K s(k)}$	$p_{23} = \frac{\sum_{k=1}^K (f_k - p_{16})^4 s(k)}{Kp_{17}^4}$
$p_6 = \frac{\sum_{n=1}^N (x(n) - p_1)^3}{(N-1)p_2^3}$		$p_{17} = \sqrt{\frac{\sum_{k=1}^K (f_k - p_{16})^2 s(k)}{K}}$	$p_{24} = \frac{\sum_{k=1}^K \sqrt{f_k - p_{16}} s(k)}{K\sqrt{p_{17}}}$

时域特征指标	频域特征指标
$p_{18} = \sqrt{\frac{\sum_{k=1}^K f_k^2 s(k)}{\sum_{k=1}^K s(k)}}$	

对于表 5 指标, $x(n)$ 是 $n=1, 2, \dots, N$ 的信号序列; N 是数据点的数量; $s(k)$ 是 $k=1, 2, \dots, K$ 的频谱, K 是频谱线的数量; f_k 是第 k 条频谱线的频率值。

5.2.3 基于熵权-TOPSIS 的钢丝绳安全量化评价模型

首先, 我们使用熵权法计算指标权重。我们产生归一化的矩阵 Z

$$Z = \begin{bmatrix} z_{11} & z_{12} & \cdots & z_{1m} \\ z_{21} & z_{22} & \cdots & z_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ z_{n1} & z_{n2} & \cdots & z_{nm} \end{bmatrix} \quad (20)$$

在得到归一化矩阵后, 我们必须对其进行标准化处理, 即对矩阵的每个元素进行如下运算

$$w_{ij} = \frac{z_{ij}}{\sqrt{\sum_{i=1}^n z_{ij}^2}} \quad (21)$$

计算第 j 个指标的信息熵的公式如下

$$e_j = -\frac{1}{\ln n} \sum_{i=1}^n \ln(p_{ij}) \quad (j=1, 2, \dots, m) \quad (22)$$

式中, p_{ij} 为概率矩阵元素。最后, 我们确定了每个特征的客观权重

$$\omega_j = \frac{1-e_j}{\sum_{j=1}^m d_j} \quad (j=1, 2, \dots, m) \quad (23)$$

然后在 TOPSIS 模型中使用生成的权重计算每个钢丝绳每次上行与下行的安全性能得分，并且为更好地量化钢丝绳的安全性能，本文在钢丝绳连续上下往返的过程中，将连续的上行与下行的评价值取平均，得到该次往返的综合评价值。具体结果如下图所示。

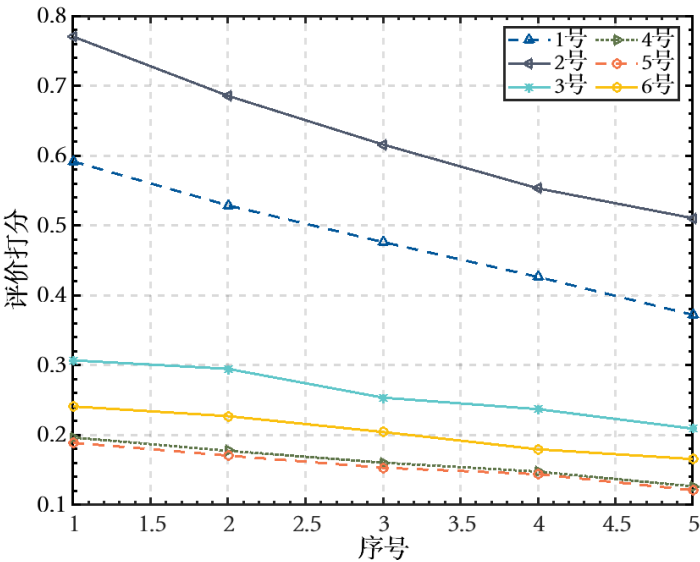


图 12 钢丝绳安全性能量化评价值

由上图可知，随着钢丝绳的运行次数的增加，钢丝绳的安全性能评价值会随之下降，并且，6 根钢丝绳的安全性能评价值始终保持 2 号>1 号>3 号>6 号>4 号>5 号。

为得到 6 根钢丝绳在这 10 次往返过程中的总体安全性能评价值，本文将采用专家赋权法，对钢丝绳每一次往返给出专家权重，并根据专家权重计算得到 6 根钢丝绳的安全性能评价值。专家权重如下表所示：

表 5 专家权重表

	第一次往返	第二次往返	第三次往返	第四次往返	第五次往返
权重	0.04	0.06	0.11	0.23	0.56

得到专家权重后，本文将根据此专家权重以及每一次往返的评分，计算每一根钢丝绳的综合安全性能评价值，结果如表 6 所示：

表 6 综合安全性能评价值

	1 号绳	2 号绳	3 号绳	4 号绳	5 号绳	6 号绳
安全性能评价值	0.4142	0.5528	0.2296	0.1411	0.1356	0.1799

通过表 6 所示，比较 6 个绳子的安全性能评价值，本文可以得出，绳子安全性能由

大到小可以排序为：2 号>1 号>3 号>6 号>4 号>5 号。2 号绳子的安全性能最好。

六、 模型评价

6.1 模型的优点

1. BO-WT-VMD 算法利用小波变换对监测数据进行降噪处理。不同于传统小波变换采取的硬阈值法或软阈值法，本模型采取软硬阈值折中的方法来对小波变换进行改进，以解决小波变换硬软阈值法信号处理后的失真较大等问题。
2. BO-WT-VMD 算法采用了变分模态分解算法来修正小波变换。算法可以捕捉低频信号得漂移及速度变化，可以精确地对于信号去噪，找出噪音信号的真实波段。
3. BO-WT-VMD 算法使用贝叶斯优化算法对 VMD 算法的主要参数 K 和 α 的取值进行了优化，使得模型具有相对较高的可靠性与客观性。
4. LOF 算法对数据的分布没有要求，不需要假定数据属于某个特定的概率分布。

6.2 模型的缺点

1. BO-WT-VMD 算法相较于小波变换，运算效率更慢。
2. 熵权法对数据分布做出了强有力的假设，若数据分布不均匀，会对后续的计算产生一定的影响。

参考文献

- [1] 赵志科. 矿井提升钢丝绳的动态检测与故障诊断方法研究[D].中国矿业大学,2018.
- [2] Peterka, Pavel, et al. "Failure analysis of hoisting steel wire rope." Engineering Failure Analysis 45 (2014): 96-105.
- [3] Lu, Jingyi, et al. "An improved variational mode decomposition method based on the optimization of salp swarm algorithm used for denoising of natural gas pipeline leakage signal." Measurement 185 (2021): 110107.
- [4] 韩梦方. 基于小波变换的矿井提升机钢丝绳故障识别研究 [D]. 江苏：中国矿业大学，2016.
- [5] 郑明杰，李颖. 多智能仪器间抗电磁互干扰方法仿真 [J]. 计算机仿真，2019,36

(3):422-425.

- [6] Tian J, Wang H, Zhou J, et al. Study of pre-processing model of coal-mine hoist wire-rope fatigue damage signal[J]. International Journal of Mining Science and Technology, 2015, 25(6): 1017-1021.
- [7] 朱汇龙, 刘晓燕, 刘瑶。基于贝叶斯新型深度学习超参数优化的研究 [J]. 数据通信, 2019 (2):35-38,46.
- [8] 常淼, 沈艳霞。基于贝叶斯优化 CNN 的风电轴承故障诊断策略 [J]. 噪声与振动控制, 2021,41 (6):77-83.
- [9] Lei Y, He Z, Zi Y, et al. Fault diagnosis of rotating machinery based on multiple ANFIS combination with GAs[J]. Mechanical systems and signal processing, 2007, 21(5): 2280-2294.
- [10] Sun Y, Wu J, Feng B, et al. An opening electric-MFL detector for the NDT of in-service mine hoist wire[J]. IEEE Sensors Journal, 2014, 14(6): 2042-2047.
- [11] Stankevych O, Skalskyi V, Klym B, et al. Identification of fracture mechanisms in cementitious composites using wavelet transform of acoustic emission signals[J]. Procedia Structural Integrity, 2022, 36: 114-121.
- [12] Lu J, Yue J, Zhu L, et al. An improved variational mode decomposition method based on the optimization of salp swarm algorithm used for denoising of natural gas pipeline leakage signal[J]. Measurement, 2021, 185: 110107.

附录

附录一：通过脉冲的定义计算每帧位置

```
from collections import Counter
dis=np.array([])
for i in range(len(c)):
    op=c[i]
    lens=op
    delda=0.4/lens
    n=np.linspace(0.4*(i)+delda,0.4*(i+1),lens)
    dis=np.hstack([dis,n])
```

附录二：BO-WT-VMD 算法核心代码

```
def sgn(num):
    if(num > 0.0):
        return 1.0
    elif(num == 0.0):
        return 0.0
    else:
        return -1.0

def wavelet_noising(new_df):
    data = new_df
    # data = data.values.T.tolist() # 将np.ndarray()转为列表
    w = pywt.Wavelet('sym8')#选择sym8 小波基

    # 5 层小波分解
    [ca5, cd5, cd4, cd3, cd2, cd1] = pywt.wavedec(data, w, level=5)
    length1 = len(cd1)
    length0 = len(data)

    Cd1 = np.array(cd1)
    abs_cd1 = np.abs(Cd1)
    median_cd1 = np.median(abs_cd1)

    sigma = (1.0 / 0.6745) * median_cd1
    lamda = sigma * math.sqrt(2.0 * math.log(float(length0 ),
math.e))#固定阈值计算
    usecoeffs = []
    usecoeffs.append(ca5) # 向列表末尾添加对象
```

#软硬阈值折中的方法

`a = 0.5`

```
for k in range(length1):
    if (abs(cd1[k]) >= lamda):
        cd1[k] = sgn(cd1[k]) * (abs(cd1[k]) - a * lamda)
    else:
        cd1[k] = 0.0
```

```
length2 = len(cd2)
for k in range(length2):
    if (abs(cd2[k]) >= lamda):
        cd2[k] = sgn(cd2[k]) * (abs(cd2[k]) - a * lamda)
    else:
        cd2[k] = 0.0
```

```
length3 = len(cd3)
for k in range(length3):
    if (abs(cd3[k]) >= lamda):
        cd3[k] = sgn(cd3[k]) * (abs(cd3[k]) - a * lamda)
    else:
        cd3[k] = 0.0
```

```
length4 = len(cd4)
for k in range(length4):
    if (abs(cd4[k]) >= lamda):
        cd4[k] = sgn(cd4[k]) * (abs(cd4[k]) - a * lamda)
    else:
        cd4[k] = 0.0
```

```
length5 = len(cd5)
for k in range(length5):
    if (abs(cd5[k]) >= lamda):
        cd5[k] = sgn(cd5[k]) * (abs(cd5[k]) - a * lamda)
    else:
        cd5[k] = 0.0
```

```
usecoeffs.append(cd5)
usecoeffs.append(cd4)
usecoeffs.append(cd3)
usecoeffs.append(cd2)
usecoeffs.append(cd1)
recoeffs = pywt.waverec(usecoeffs, w)#信号重构
```

```

    return recoeffs

def objective(trial):
    x1=trial.suggest_float('x1',1000,2000,log=True)
    x2=trial.suggest_int("x2",1,8)
    def VMD_im(x):
        alpha = x1          # moderate bandwidth constraint
        tau = 0.             # noise-tolerance (no strict fidelity
enforcement)
        K = x2               # 3 modes
        DC = 0               # no DC part imposed
        init = 1             # initialize omegas uniformly
        tol = 1e-7
        u, u_hat, omega = VMD(f, alpha, tau, K, DC, init, tol)
        li=[]
        for i in range(u.shape[0]):
            a=np.corrcoef(f,u[i])[0][1]
            li.append(a)
        C=np.array(li)
        t=np.max(C)
        limit=t/(10*t-3)
        t_panduan=C[C>=limit]
        sum0=0
        for i in range(len(t_panduan)):
            sum0+=u[i]
        return sum0

def sgn(num):
    if(num > 0.0):
        return 1.0
    elif(num == 0.0):
        return 0.0
    else:
        return -1.0

def wavelet_noising(new_df):
    data = new_df
    # data = data.values.T.tolist() # 将np.ndarray()转为列表
    w = pywt.Wavelet('sym8')#选择sym8 小波基
    [ca5, cd5, cd4, cd3, cd2, cd1] = pywt.wavedec(data, w,
level=5) # 5 层小波分解

    length1 = len(cd1)
    length0 = len(data)

```

```

Cd1 = np.array(cd1)
abs_cd1 = np.abs(Cd1)
median_cd1 = np.median(abs_cd1)

sigma = (1.0 / 0.6745) * median_cd1
lamda = sigma * math.sqrt(2.0 * math.log(float(length0 ),
math.e))#固定阈值计算
usecoeffs = []
usecoeffs.append(ca5) # 向列表末尾添加对象

#软硬阈值折中的方法
a = 0.5

for k in range(length1):
    if (abs(cd1[k]) >= lamda):
        cd1[k] = sgn(cd1[k]) * (abs(cd1[k]) - a * lamda)
    else:
        cd1[k] = 0.0

length2 = len(cd2)
for k in range(length2):
    if (abs(cd2[k]) >= lamda):
        cd2[k] = sgn(cd2[k]) * (abs(cd2[k]) - a * lamda)
    else:
        cd2[k] = 0.0

length3 = len(cd3)
for k in range(length3):
    if (abs(cd3[k]) >= lamda):
        cd3[k] = sgn(cd3[k]) * (abs(cd3[k]) - a * lamda)
    else:
        cd3[k] = 0.0

length4 = len(cd4)
for k in range(length4):
    if (abs(cd4[k]) >= lamda):
        cd4[k] = sgn(cd4[k]) * (abs(cd4[k]) - a * lamda)
    else:
        cd4[k] = 0.0

length5 = len(cd5)
for k in range(length5):
    if (abs(cd5[k]) >= lamda):

```

```

        cd5[k] = sgn(cd5[k]) * (abs(cd5[k]) - a * lamda)
    else:
        cd5[k] = 0.0

    usecoeffs.append(cd5)
    usecoeffs.append(cd4)
    usecoeffs.append(cd3)
    usecoeffs.append(cd2)
    usecoeffs.append(cd1)
    recoeffs = pywt.waverec(usecoeffs, w)#信号重构
    return recoeffs
xiaobo=wavelet_noising(f)
xiaobo_vmd=VMD_im(xiaobo)
score=SNR(f,xiaobo_vmd)
return score
study = optuna.create_study(direction="maximize")
study.optimize(objective, n_trials=100)
print(study.best_trial)

def VMD_im_bo(x):
    alpha=study.best_params["x1"]
    tau = 0.          # noise-tolerance (no strict fidelity
enforcement)
    K = study.best_params["x2"]          # 3 modes
    DC = 0          # no DC part imposed
    init = 1          # initialize omegas uniformly
    tol = 1e-7
    u, u_hat, omega = VMD(f, alpha, tau, K, DC, init, tol)
    li=[]
    for i in range(u.shape[0]):
        a=np.corrcoef(f,u[i])[0][1]
        li.append(a)
        C=np.array(li)
    t=np.max(C)
    limit=t/(10*t-3)
    t_panduan=C[C>=limit]
    sum0=0
    for i in range(len(t_panduan)):
        sum0+=u[i]
    return sum0

```

附录三：特征指标构建核心代码

```
import numpy as np
import scipy.stats

class Fea_Extra():
    def __init__(self, Signal, Fs = 25600):
        self.signal = Signal
        self.Fs = Fs

    def Time_fea(self, signal_):
        """
        提取时域特征 11 类
        """
        N = len(signal_)
        y = signal_
        t_mean_1 = np.mean(y) # 1_均值 (平均幅值)
        t_std_2 = np.std(y, ddof=1) # 2_标准差
        t_fgf_3 = ((np.mean(np.sqrt(np.abs(y)))))**2 # 3_方根幅值
        t_rms_4 = np.sqrt((np.mean(y**2))) # 4_RMS 均方根
        t_pp_5 = 0.5*(np.max(y)-np.min(y)) # 5_峰峰值
        #t_skew_6 = np.sum((t_mean_1)**3)/((N-1)*(t_std_3)**3)
        t_skew_6 = scipy.stats.skew(y) # 6_偏度 skewness

        #t_kur_7 = np.sum((y-t_mean_1)**4)/((N-1)*(t_std_3)**4)
        t_kur_7 = scipy.stats.kurtosis(y) # 7_峭度 Kurtosis
        t_cres_8 = np.max(np.abs(y))/t_rms_4 #8 峰值因子 Crest Factor
        t_clear_9 = np.max(np.abs(y))/t_fgf_3 #9_裕度因子 Clearance
        Factor
        t_shape_10 = (N * t_rms_4)/(np.sum(np.abs(y))) # 10_波形因子
        Shape fator
        t_imp_11 = ( np.max(np.abs(y)))/(np.mean(np.abs(y)))# 11_脉冲
        指数 Impulse Fator
        t_fea = np.array([t_mean_1, t_std_2, t_fgf_3, t_rms_4, t_pp_5,
                        t_skew_6, t_kur_7, t_cres_8, t_clear_9,
                        t_shape_10, t_imp_11 ])
        return t_fea

    def Fre_fea(self, signal_):
        """
        提取频域特征 13 类
        :param signal_:
        :return:
        """
```

```

    """
    L = len(signal_)
    PL = abs(np.fft.fft(signal_ / L))[: int(L / 2)]
    PL[0] = 0
    f = np.fft.fftfreq(L, 1 / self.Fs)[: int(L / 2)]
    x = f
    y = PL
    K = len(y)

    f_12 = np.mean(y)
    f_13 = np.var(y)
    f_14 = (np.sum((y - f_12)**3))/(K * ((np.sqrt(f_13))**3))
    f_15 = (np.sum((y - f_12)**4))/(K * ((f_13)**2))
    f_16 = (np.sum(x * y))/(np.sum(y))
    f_17 = np.sqrt((np.mean(((x - f_16)**2)*(y))))
    f_18 = np.sqrt((np.sum((x**2)*y))/(np.sum(y)))
    f_19 = np.sqrt((np.sum((x**4)*y))/(np.sum((x**2)*y)))
    f_20 =
(np.sum((x**2)*y))/(np.sqrt((np.sum(y))*(np.sum((x**4)*y))))
    f_21 = f_17/f_16
    f_22 = (np.sum(((x - f_16)**3)*y))/(K * (f_17**3))
    f_23 = (np.sum(((x - f_16)**4)*y))/(K * (f_17**4))
    f_fea = np.array([f_12, f_13, f_14, f_15, f_16, f_17, f_18,
f_19, f_20, f_21, f_22, f_23])
    return f_fea

def Both_Fea(self):
    """
    :return: 时域、频域特征 array
    """
    t_fea = self.Time_fea(self.signal)
    f_fea = self.Fre_fea(self.signal)
    fea = np.append(np.array(t_fea), np.array(f_fea))
    return fea

```