Research Article
Multiteam Competitive Optimization Algorithm and Its Application in Bearing Fault Diagnosis

Bo Zheng,1 Huiying Gao,1 Xin Ma,2 and Xiaoqiang Zhang1

1Aviation Engineering Institute, Civil Aviation Flight University of China, Guanghan 618307, China
2Air Traffic Management College, Civil Aviation Flight University of China, Guanghan 618307, China

Correspondence should be addressed to Xiaoqiang Zhang; xqzhanguestc@163.com

Received 1 March 2021; Accepted 25 June 2021; Published 8 July 2021

A novel multiteam competitive optimization (MTCO) algorithm has been proposed to diagnose the fault patterns of bearings. This algorithm is inspired by competitive behaviors of multiple teams. It is a three-level organization structure; thus, more potential optimal areas can be searched. By imitating human thinking, such as the betrayal and replacement behavior along with the introduction of an acceptable vector, new strategies within the MTCO are designed to increase the diversity and guide jumping out of location suboptimal areas. In addition to this, a kernel function has been introduced to reduce the recognition errors caused by data which are nonlinearly distributed in original space. The obtained experimental results demonstrate that the proposed MTCO is globally stable and optimal decision performance. After that the MTCO is applied for the fault diagnosis of bearings, and it has also been compared with other commonly used methods. The comparison indicates that the proposed algorithm has higher recognition accuracy.

1. Introduction

Fault diagnosis, which has been formed as an independent interdisciplinary during the 1960s, plays a significant role in modern industrial processes. It is the key to prevent serious accidents, ensure safety and obtain maximum economic benefits [1, 2]. With the development of industrial equipment, new data-driven methods including intelligence algorithms, data fusion, multivariate statistical analysis, signal processing and machine learning are paid much attention and get rapid development [3, 4]. A large number of researches on data-driven methods have been proposed to promote the development of fault diagnosis technology [5].

In recent decade, the researches on fault diagnosis mainly focus on data-driven methods. Yuan and Chu [6] investigated the discrete particle swarm optimization (PSO) algorithm to select the best fault features and optimize the performance of the support vector machine (SVM) classifier. There the experimental results proved that the developed method shows better performance than the SVM based on principal component analysis (PCA), or SVM based on genetic algorithm (GA) in the application of fault diagnosis of the turbo pump rotor. Zheng and Ma [7] introduced a method using the dual-mutation PSO to determine the optimal parameters for SVM. The experiments for damage type recognition of civil aeroengine indicated that the method could achieve higher diagnostic accuracy than that of the single or multiple kernels SVM where parameters were set randomly. Korürek and Doğan [8] introduced the data-driven method into the field of life sciences, and the classifier radial basis function (RBF) neural network whose structure was evolved by PSO, the optimized RBF network could classify electrocardiogram beats with a smaller size of network without making any concessions on the classification performance. Based on the above studies and other works presented by Lu et al. and Shang et al. [9, 10], a conclusion can be made that an appropriate classifier is essential for detection and identification of unknown fault patterns. Inspired by the heuristic random search process of swarm, this paper proposes a swarm intelligence algorithm called multiteam competitive optimization (MTCO) algorithm. This proposed algorithm is a data-driven method and mainly
used to recognize unknown fault patterns. The essence of this algorithm is to search an optimal single-class center for each unknown fault pattern. Lots of researchers have focused on studying human decision methods by imitating human behaviors. Among them, some improved methods have achieved remarkable calculation effect, such as the work presented by Zheng [11]. Research on human learning behaviors had shown that the best planners adjusted their decisions considering the current state and their perception of the best experiences from others. Based on this idea, Tanweer et al. [12] designed two learning strategies for PSO incorporating the best human learning strategies for finding the optimum solution. Cheng and Jin [13] developed a social learning PSO (SL-PSO), which allowed the individuals to learn behaviors from other better particles in the current swarm. There the comparative results showed the SL-PSO performed well on low-dimensional problems and was promising for solving large-scale problems as well. Li et al. [14] proposed an information sharing mechanism (ISM) to make each particle share its best search information, through which all the particles could take advantage of the shared information. In this way, a competitive and cooperative operator inspired by human behaviors was designed to control the shared information being utilized in a proper and efficient way. The competitive and cooperative PSO with ISM (CCPSO-ISM) can prevent the premature convergence when solving global optimization problems. In addition, aging had been explored by biologists to be an important mechanism for maintaining diversity. In a human social, aging made the old leader of the colony become weak, which provides opportunities for the other individuals to challenge the leadership position. Inspired by this natural phenomenon, Zheng et al. [15] incorporated this mutation mechanism into PSO and PSO with a trying mutation strategy, to overcome the problem of premature convergence without significantly impairing the fast-converging feature of PSO.

In this paper, based on the traditional PSO, we constructed a new organization structure for the particles, in which the competition of multiple teams is introduced to recognize unknown fault patterns and then find the optimal decision. The particles are redefined as members and classified into three groups, that is, staff belonging to the majority, leaders belonging to the minority, and the only boss. The organization structure of the proposed algorithm is shown in Figure 1. These staff follow different leaders and the leaders are accountable towards the boss. Finally, the boss adjusts the decision again based on the optimal decision made by staff and leaders. So the method is termed as MTCO algorithm.

Aiming to search more potential optimal areas, the MTCO algorithm is conducive to get rid of the premature convergence effectively and overcome the influence of randomness on the optimal decision solution. Thus, the global optimal decision solution can be obtained with a higher probability. Therefore, the proposed algorithm can obviously improve the accuracy of unknown fault patterns via looking for the optimal class centers of different fault patterns.

The rest of the paper is organized as follows. In Section 2, MTCO algorithm has been proposed. Section 3 introduces the recognition process of the MTCO algorithm. After that the application problem that is the fault diagnosis of bearing has been discussed in Section 4. Finally, in the last section, conclusion has been drawn.

2. The MTCO Algorithm

2.1. The Optimization Principle. The MTCO algorithm is constructed on the basis of the traditional PSO. But the meanings of particles, updating strategy, and organization structure have been redesigned. The traditional PSO has to suffer from the premature convergence due to trap into local optimal areas, or the lack of diversity [16, 17]. Also, the traditional PSO is not very efficient when coping with complex multimodal functions because the whole particles only follow the single global extremum and cannot efficiently exchange search information. Inspired by the competition of multiple teams, this paper designs a new organization structure and corresponding updating strategy to solve the exiting problems mentioned above.

The fitness function $\text{fit}()$ existing in the traditional PSO is redefined as the decision function $\text{dec}()$ for the competition behaviors of multiple teams. And the fitness value is also redefined as the decision value. Assume that it is a maximization problem. As shown in Figure 1, the global extremum $m_{g,d}$ is redefined as the only boss, and the ordinary individuals are redesigned as leaders $m_i$ and staff $m_s$, respectively. The number of leaders is $n_l$ and the number of staff is $n_s$. Leaders and staff have their own individual extremum $m_{i,d}$ and $m_{s,d}$, respectively. Each team only has one leader. The staff in one team make decisions under the guidance of their team leaders. So the leaders belong to the minority with higher decision values, and the staff belong to the majority with relatively low decision values, usually, $n_l < 10\% \cdot n_s$.

Based on this structure, the procedure of the MTCO is described as follows.

Step 1. Initialize each individual’s initial velocity and position randomly using a certain method. Each position is a potential optimal decision solution.

Step 2. Calculate individual’s decision value and sort these values in an ascending order according to their decision values. The individual with the highest decision value is the boss $m_{g,d}$.

Step 3. Choose $n_l$ individuals as leaders from the top of ranking. The rest of $n_s$ individuals are the staff. Then the staff will be randomly assigned to different teams, and each team may have different number of staff according to their decision values. The higher the decision value is, the more staff the team has. The number $N(m_{i,j})$ of staff belonging to $m_{i,j}$ is calculated by

$$N(m_{i,j}) = \frac{n_s \cdot j}{\text{sum}(n_l)}.$$  \hspace{1cm} (1)

where $j = 1, \ldots, n_l$ denotes the leaders’ ordinal in an ascending order. $\text{sum}(n_l) = 1 + \cdots + n_l$. 

2 Mathematical Problems in Engineering
Step 4. Update these staff’s velocities and positions using equations (2) and (3), respectively.

\[
\begin{align*}
v_i^{t+1} &= \omega_i^t v_i^t + c_1 r_1 a_i \left( m_{i,j}^{t-1} d(i,j) - m_i^t \right) \\
&\quad + c_2 r_2 a_2 \left( m_{j}^t d(i,j) - m_i^t \right), \\
m_i^{t+1} &= m_i^t + v_i^{t+1},
\end{align*}
\]

and \(\omega_i^{t+1}\) can be calculated by

\[
\omega_i^{t+1} = \omega_i^t - \frac{(\omega_{\text{max}} - \omega_{\text{min}})}{t_{\text{max}}},
\]

where \(i = 1, \ldots, n,\) denotes the staff’s ordinal in an ascending order. \(t\) denotes the current iteration number, \(t_{\text{max}}\) is the maximum iteration number. \(c_1\) and \(c_2\) are constants, usually, \(c_1 = 1.5,\) and \(c_2 = 1.7.\) \(r_1\) and \(r_2\) are random control parameter sets in interval \([0,1]\). \(m_{i,j}^{t-1} d(i,j)\) represents the individual extremum of the \(j\)th leader which is the leader of the \(i\)th staff. \(a_i\) and \(a_2\) are newly designed as acceptable vector which is a binary. \(\omega_i^t\) is the inertia weight, \(\omega_{\text{max}}\) is the maximum value, and \(\omega_{\text{min}}\) is the minimum value. The initial value of \(\omega_i^t\) is \(\omega_{\text{max}},\) \(\omega_i^t\) is a linear decreasing inertia weight adopted by these staff. It is because that these staff should gradually converge to their leaders to search for better decision values. So the linear decreasing inertia weight adopted here is reasonable.

In this study, \(\omega_i^t v_i^t\) is denoted as the cumulative experience which represents the gradual process from indistinct to clarity for cognizing the decision problem. \(m_{i,j}^{t-1} d(i,j) - m_i^t\) is explained as the independent-amendment ability. It means that the staff amend their decisions depending on their own recognition. And in a team, the leader’s opinion usually affects the staff’s judgment. So \(m_{i,j}^{t-1} d(i,j) - m_i^t\) is described as the dependent-amendment ability which is affected by the leader. In practical engineering, some uncertainty or random events around the staff may affect the decision-making. It will lead to a better or worse decision. So after amending a decision, the staff may accept some amending suggestions or reject it. The acceptable vector just imitates this process. Moreover, another function of acceptable vector is used to increase the diversity of staff and avoid the aggregation of staff. The acceptable vector is a binary vector as the same dimensions as staff, which can be calculated by

\[
a_i = \text{bin} (\text{unifrnd} (0, 1, D) - r_N),
\]

where \(a_i = [a_{i1}, a_{i2}, \ldots, a_{iD}]\), \(a_{ij}\) is an element of \(a_i\). \(\text{bin} (\cdot)\) is a function for the purpose that if \(a_{ij} \leq 0\), \(a_{ij} = 0;\) otherwise \(a_{ij} = 1\). \(r_N\) is a random threshold in interval \([0, 1]\). \(\text{unifrnd} (\cdot)\) is a function used for generating a uniform distribution real vector with \(D\) dimensions in interval \([0, 1]\). So if \(a_{ij} = 1\), the amendment for the \(j\)th dimension will be accepted. Otherwise, the amendment will be rejected.

Actually, the updating process of staff is consistent with the process of human decision-making. Meanwhile, the values of \(r_1\) and \(r_2\) can control the personality of the staff who tend to depend on their own amendment or leader’s guidance, or a compromise.

In this step, the betrayal mechanism is introduced to guide the staff leaving the current team leader when they cannot change their individual extremum. We design a two-dimensional betrayal vector \(b = [b_L, b_U]\) called betrayal vector. \(b_L\) and \(b_U\) are lower and upper limit values, respectively, initialized randomly. When \(\text{dec} (m_{i,j}^{t-1}) = \text{dec} (m_{i,j}^{t})\) is lower than \(\text{dec} (m_{i,j}^{t})\), \(b_L = b_L + 1\). It means that if the staff cannot achieve a better decision solution, it will create discontent for leader, like human, the increasing of discontent is gradually cumulative. When \(b_L = b_U\), the staff may leave the current leader and enter another team with a certain probability, the probability of \(m_{i,j} \in m_{i,j}\) which denotes the \(i\)th staff belongs to the \(j\)th leader is as follows:

\[
P \left( m_{i,j} \in m_{i,j} \right) = \frac{j}{\sum(n_j)}
\]

Due to \(j = 1, \ldots, n,\) the staff which intends to betray its current leader may go to other teams, or may still follow its current leader. After reselecting the new leader, then initialize the betrayal vector \(b\) again.
If $\text{dec}(\mathbf{m}_{s(i)}^{t+1})$ is higher than $\text{dec}(\mathbf{m}_{l,j,d(i)}^{t+1})$, $\mathbf{m}_{s(i)}^{t+1} = \mathbf{m}_{l,j,d(i)}^{t+1}$; if $\text{dec}(\mathbf{m}_{s(i)}^{t+1})$ is higher than $\text{dec}(\mathbf{m}_{l,j,d(i-j),d(i)}^{t+1})$, $\mathbf{m}_{l,j,d(i-j),d(i)}^{t+1} = \mathbf{m}_{s(i)}^{t+1}$; and if $\text{dec}(\mathbf{m}_{g}^{t+1})$ is higher than $\text{dec}(\mathbf{m}_{g}^{t, d(i)})$, $\mathbf{m}_{g}^{t, d(i)} = \mathbf{m}_{g}^{t+1}$. Using this updating strategy, the staff can make better decisions. Meanwhile, the communication with leaders and boss can be completed, which will provide better references for leaders and boss.

Step 5. Update these leaders’ velocities and positions using equations (7) and (8), respectively.

$$
\mathbf{v}_{l(j)}^{t+1} = \mathbf{v}_{l(j)}^{t} + c_1 r_1 \mathbf{a}_1 (\mathbf{m}_{l,j,d(i)}^{t} - \mathbf{m}_{l(j)}^{t}) + c_2 r_2 (\mathbf{m}_{g}^{t, d(i)} - \mathbf{m}_{l(j)}^{t}),
$$

(7)

$$
\mathbf{m}_{l(j)}^{t+1} = \mathbf{m}_{l(j)}^{t} + \mathbf{v}_{l(j)}^{t+1}.
$$

(8)

Similarly, $\omega_{l(j)}^{t+1}$ can be calculated by

$$
\omega_{l(j)}^{t+1} = \omega_{l(j)}^{t} + \left(\frac{\omega_{\text{max}} - \omega_{\text{min}}}{t_{\text{max}}} \right).
$$

(9)

For these leaders, the dependent-amendment ability in equation (7) is influenced by the boss $\mathbf{m}_{g}^{t, d(i)}$ compared with staff’s velocities updating formula. The meanings of other terms in equation (7) are described as same as equation (2). The function of acceptable vector in equation (7) is to reduce the over dependency on the boss and avoid trapping into local suboptimal decision areas. But $w_{l(j)}^{t+1}$ is a linear increasing inertia weight. The reason is that these leaders can obtain higher decisions. They have a duty to explore more solution space for searching for the global optimal decision value. So $\omega_{l(j)}^{t+1}$ and $\omega_{l(j)}^{t}$ can balance the exploitation and exploration ability between staff and leaders, which increases the probability of obtaining the global optimal decision solution.

A leader cannot be a leader for ever. Its leader status depends on its decision-making ability. So a replacement mechanism is necessary for these leaders when they have relative lower decision values than some staff. At the iteration $t$, if some leaders drop outside the top $n_l$ in the ranking, they will be relegated to staff, and given a new leader and new betrayal vector randomly, the staff with higher decision solutions will replace the leaders. Meanwhile, these staff belonging to the relegated leaders will reselect a new leader with the probability calculated by equation (9–). The replacement mechanism can ensure the whole staff follow the excellent leaders all the time, which is conducive to make the global optimal decision solution.

If $\text{dec}(\mathbf{m}_{l(j)}^{t+1})$ is higher than $\text{dec}(\mathbf{m}_{l,j,d(i)}^{t+1})$, $\mathbf{m}_{l(j)}^{t+1} = \mathbf{m}_{l,j,d(i)}^{t+1}$; and if $\text{dec}(\mathbf{m}_{l(j)}^{t+1})$ is higher than $\text{dec}(\mathbf{m}_{g}^{t, d(i)})$, $\mathbf{m}_{g}^{t, d(i)} = \mathbf{m}_{l(j)}^{t+1}$.

Step 6. Update the boss’s position using

$$
\mathbf{m}_{g}^{t+1} = \mathbf{m}_{g}^{t} + \mathbf{a} \cdot \text{unifrnd} \left( -\omega_{r}^{t}, \omega_{r}^{t}, D \right),
$$

(10)

and $\omega_{r}^{t}$ can be calculated using

$$
\omega_{r}^{t} = \omega_{r} - \frac{t \cdot (\omega_{r} - \omega_{r_{\text{max}}})}{t_{\text{max}}}.
$$

(11)

$\omega_{r}^{t}$ is also a linear decreasing inertia weight, where $\omega_{r_{\text{max}}}$ is initial value of $\omega_{r}$, and $\omega_{r_{\text{min}}}$ is the final value of $\omega_{r}$, usually $\omega_{r_{\text{max}}} > \omega_{r_{\text{min}}}$. In the prophase of iteration, a larger $\omega_{r}$ is conducive to explore the larger solution space roughly. In the later stage of iteration, a smaller $\omega_{r}$ is conducive to exploit the more precise solution.

On the basis of global optimal decision solution made by all subordinates, the boss will amend the current global decision value with random free updating strategy. The decision-making for the future is still an uncertain process. Uncertainty factors also influence the amendment of boss, which leads to a higher or lower decision solution. So acceptable vector still works on equation (10). The elite strategy is used to update the boss’s position. The description for elite strategy is like that if $\text{dec}(\mathbf{m}_{g}^{t})$ is higher than $\text{dec}(\mathbf{m}_{g}^{t}, d(i))$, retain $\mathbf{m}_{g}^{t}$; otherwise, retain $\mathbf{m}_{g}^{t}, d(i)$.

Step 7. Stop the iteration if the terminal condition $t = t_{\text{max}}$ is satisfied; otherwise, go to Step 2.

Finally, the flow chart of the MTCO algorithm is illustrated in Figure 2.

The MTCO algorithm can increase the diversity of staff and make the leaders get rid of the over dependency on the global optimal decision solution effectively. The layered organization structure can make the staff follow more potential optimal decision solutions. Meanwhile, the boss of each iteration also takes part in making decision, which increases the exploitation ability for obtaining optimal decision solution. So in the following section, the MTCO algorithm would be verified for its global optimal decision ability.

2.2. Optimization Experiment and Comparison. Some well-known test functions are used for comparing the performances of different optimization algorithms. The commonly used test functions include the Schaffer function $f_{\text{Sch}}$, the Griewank function $f_{\text{Gri}}$, and the Rosenbrock function $f_{\text{Ros}}$, which are given as follows:

$$
f_{\text{Sch}} = 0.5 + \frac{\left( \sin \sqrt{x_1^2 + x_2^2} \right)^2 - 0.5}{1 + 0.001(\sqrt{x_1^2 + x_2^2})},
$$

(12)

$$
f_{\text{Gri}} = \sum_{i=1}^{n} \frac{x_i^2}{4000} - \prod_{i=1}^{n} \cos \frac{x_i}{\sqrt{i}} + 1,
$$

$$
f_{\text{Ros}} = \sum_{i=1}^{n-1} \left[ 100(x_{i+1}^2 - x_i^2)^2 + (x_i - 1)^2 \right].
$$

The graphics of these functions in the three-dimensional coordinate are shown in Figure 3. $f_{\text{Sch}}$ is a complex two-dimension function having countless local extremum. This can acquire the global extremum at the location of $(x_1, x_2) = (0, 0)$, so it is difficult to search the global extremum. $f_{\text{Gri}}$ has lots of local extremum point. The global
extremum is 0, which can be searched at the location of 
\((x_1, x_2, \ldots, x_n) = (0, 0, \ldots, 0)\). The Griewank function is a

typical nonlinear multimode function, which is always regarded as a more complex optimization problem. As for 
\(f_{Ros}\), its global extremum is located in a smooth and narrow

area similar to valley formed by parabolas, and thus the

information provided by \(f_{Ros}\) is limited that many opti-
mization algorithms have a problem on the identification of

searching direction.

The self-regulating PSO (SR-PSO) proposed by Tanweer

et al. [12], the seeker optimization algorithm (SOA) proposed

by Dai [18], and the ALC-PSO proposed by Singh et al. [19]

were used to obtain relative better optimization results by

imitating some human decision-making behaviors. Therefore,

the MTCO algorithm will compare with them, besides

Standard PSO (SPSO). The population size for all algorithms

is 60, and the total iteration number is set as 600. As for the

MTCO algorithm, the number of leaders is set as 3, \(\omega_{\text{max}}\) is set

as 0.9, \(\omega_{\text{min}}\) is set as 0.2, \(\omega_{\epsilon}\) is set as 0.5, \(\omega_{re}\) is set as \(5e^{-5}\), the

\(b_L\) is set randomly in interval [80, 90], and the \(b_U\) is set

randomly in interval [95, 100], \(b_L\) and \(b_U\) are both integers.

How to set the parameters for other algorithms mentioned

have been described in previous studies [12, 18, 19], and the

terminal conditions is \(t = t_{\text{max}}\). At the same time, random

settings are applied on initializing the parameters including

velocities and positions. To compare the global optimal de-

cision ability, the algorithms are calculated one hundred times

repeatedly, which can verify the influence of randomness.

Min, Mean, Max, and STD (Standard Deviation) of fitness

values can be used for performance verification. Table 1 shows

the comparison results for Schaffer, Griewank, and Rose-
brock functions, respectively.

These comparisons demonstrate that the proposed

MTCO algorithm has much stronger decision-making

ability than other algorithms. Its stable and reliable per-
formance ensures that the global optimal decision solution

can be obtained with higher probability. Accordingly, the

MTCO can be applied on recognizing unknown fault pat-
terns for the fault diagnosis of critical equipment.

3. Recognition Process of the MTCO Algorithm

3.1. The Recognition Principle. The recognition principle for

recognizing the unknown fault patterns is shown in Figure 4.

As shown in Figure 4, there are two categories of data

with two dimensions representing different patterns. Firstly,
Figure 3: Graphics of three commonly used functions. (a) The graphic of the Schaffer function. (b) The graphic of the Griewank function. (c) The graphic of the Rosenbrock function.

| Test function     | Algorithm | Min     | Mean    | Max     | STD    |
|-------------------|-----------|---------|---------|---------|--------|
| Schaffer          | Search ranged: [-50, 50], dimension: 50 | MTCO    | 0.0097  | 0.0185  | 0.0372 | 0.0130 |
|                   |           | SRPSO   | 0.2277  | 0.3194  | 0.3961 | 0.0331 |
|                   |           | ALCPSO  | 0.2727  | 0.3389  | 0.3961 | 0.0303 |
|                   |           | SOA     | 0.0782  | 0.1060  | 0.1728 | 0.0262 |
|                   |           | SPSO    | 0.2277  | 0.2990  | 0.3733 | 0.0333 |
| Griewank          | Search ranged: [-50, 50], dimension: 50 | MTCO    | 1.1171×10^{-6} | 0.0038 | 0.0296 | 0.0071 |
|                   |           | SRPSO   | 9.4550×10^{-6} | 0.0050 | 0.0392 | 0.0089 |
|                   |           | ALCPSO  | 0.0139  | 0.0536  | 0.0984 | 0.0191 |
|                   |           | SOA     | 0.0042  | 0.0177  | 0.0556 | 0.0123 |
|                   |           | SPSO    | 0.0196  | 0.0406  | 0.0741 | 0.0125 |
| Rosenbrock        | Search ranged: [-50, 50], dimension: 50 | MTCO    | 41.4554 | 57.3950 | 184.7532 | 27.5666 |
|                   |           | SRPSO   | 41.6435 | 82.1011 | 458.1885 | 56.6290 |
|                   |           | ALCPSO  | 131.7000 | 373.2552 | 1.1812人 + 03 | 188.4061 |
|                   |           | SOA     | 54.2086 | 223.3483 | 1.0162人 + 03 | 214.7484 |
|                   |           | SPSO    | 77.9133 | 379.2736 | 1.2122人 + 03 | 216.9144 |
a decision function is selected to determine an optimal class center for each category of data, and the class center need to meet three optimization conditions, that is, shorter intra-class distance, longer interclass distance, and higher recognition accuracy. Virtually, the intraclass distance is defined as the sum of distances from all known samples having the same class label to their class center. The interclass distance is defined as the sum of distances between all the class centers. And higher recognition accuracy means the proportion of the known samples classified into right categories. For an unknown sample, its recognition criterion still depends on the distance. That is to say, if the distance from an unknown sample to the center of Class 1 is the shortest, this unknown sample should be classified into this class. In this study, the Euclidean distance (ED) is used to express the distance between two samples, which can be calculated by

$$\text{ED}_{ij} = \|x_i - x_j\| = \sqrt{\|x_i - x_j\|^2} = \sqrt{\langle x_i - x_j, x_i - x_j \rangle}$$

and the criterion can be expressed as follows:

$$\langle y_j \rangle_{\text{category}} = \langle y | \text{ED} = \min \{\text{ED}_{j1}, \ldots, \text{ED}_{j\hat{n}_c} \} \rangle,$$

where $y_j$ is a unknown sample, $n_c$ is the total number of classes, $\| \cdot \|$ is the 2-norm, $x_{i,d}$ is the $d$th element of sample $x_i$ and $x_{j,d}$ is the $d$th element of sample $x_j$.

As previously mentioned, the decision function should meet three conditions, and thus the proposed decision function is defined as

$$\text{dec}(m) = \frac{N_x}{N} + \frac{\sum_{j=1}^{n_c} \sum_{i \in \pi(j)} \|c_i - c_j\|}{\sum_{i=1}^{n_c} \sum_{j \neq i} \|x_i - c_i\|}$$

where $m$ is a member, $N$ is the total number of train samples, $N_x$ is the number of train samples classified into the correct classes, $n_c(j)$ is the number of the train samples belonging to the $j$th class, and $c_j$ is the $j$th class center vector. This is a maximization decision problem, and a larger value of decision function means the MTCO algorithm has stronger recognition ability for unknown samples.

### 3.2. Kernel Function in the MTCO Algorithm

A kernel function is paid more attention with the development of SVM. The kernel function must meet the Mercer condition and can combine the nonlinear mapping and inner-product of two vectors in feature space so that the nonlinear mapping is carried out implicitly. So the kernel function makes the mapping from the original space with nonlinear distribution to the high-dimension feature space with linear distribution, the latter of which can be realized easily and simply. RBF kernel and polynomial kernel are the most commonly used kernel functions [20], which are given as follows:

**Polynomial kernel:**

$$k(x_i, x_j) = (x_i \cdot x_j + 1)^d,$$

**RBF Kernel:**

$$k(x_i, x_j) = \exp \left( -\frac{\|x_i - x_j\|^2}{2\sigma^2} \right).$$

According to the Mercer condition, $k(x_i, x_j) = \langle \Phi(x_i) \cdot \Phi(x_j) \rangle$, where $\Phi(\cdot)$ is the mapping function [21], the ED of two samples can also calculated in feature space via mapped by $\Phi(\cdot)$, which can remove the influence of nonlinear distribution of data on the recognizing unknown samples, the ED in feature space can be calculated by

$$\text{ED}_{ij} = \|\Phi(x_i) - \Phi(x_j)\| = \sqrt{\langle \Phi(x_i) - \Phi(x_j) \rangle}$$

$$= \sqrt{\langle \Phi(x_i) - \Phi(x_j), \Phi(x_i) - \Phi(x_j) \rangle}$$

$$= \sqrt{\langle \Phi(x_i), \Phi(x_i) \rangle - 2\langle \Phi(x_i), \Phi(x_j) \rangle + \langle \Phi(x_j), \Phi(x_j) \rangle}$$

$$= \sqrt{k(x_i, x_i) - 2k(x_i, x_j) + k(x_j, x_j)}.$$

Using equation (17), the decision function can be recalculated in feature space. After applying the kernel function, the data with linear distribution in feature space can be obviously conducive to recognition.

It is critical for mapping and decision-making, that how to choose an appropriate kernel function and determine corresponding kernel parameters. With the help of powerful optimization ability of the MTCO algorithm, it is easy to determine appropriate parameters for the kernel function. Kernel functions are generally categorized into two groups, including global kernel and local kernel. The former is more suitable for classification. The polynomial kernel is a typical
local kernel, and the RBF kernel is a typical global kernel. Due to \( \| x_i - x_j \| = 0 \), the RBF kernel is chosen for the MTCO algorithm. Equation (17) is then rewritten as follows:

\[
ED_{ij} = \sqrt{2 - 2k(x_i, x_j)},
\]

(18)

on the basis of equation (18), the decision function is rewritten as follows:

\[
dec(m) = \frac{N_c}{N} + \frac{\sum_{i=1}^{n_c} \sum_{j=1}^{n_c} \sqrt{2 - 2k(c_i, c_j)}}{\sum_{i=1}^{n_c} \sum_{j=1}^{n_c} \sqrt{2 - 2k(x_i, x_j)}}
\]

(19)

After calculating the optimal decision value in the high-dimension feature space, the Standard MTCO (S-MTCO) algorithm is transformed into Kernel MTCO (K-MTCO) algorithm.

3.3. Recognition Experiment and Comparison. To verify the recognition performance of the K-MTCO algorithm, several classic and typical data sets are used, whose relative information is listed in Table 2. All data sets are collected from University of California Irvine (UCI) machine-learning repository, whereas the train number and test number are selected randomly. In this study, some most widely used fault diagnosis classifiers such as SVM [22], back-propagation (BP) network [23], and learning vector quantization (LVQ) network [24] besides S-MTCO algorithm are used to compare with the K-MTCO algorithm. From this, one may have the necessity of kernel function.

The setting for K-MTCO and S-MTCO is similar as such the population size is 80; the total iteration number is 800; the number of leaders is 6; \( \omega_{\text{max}} \) is 0.9; \( \omega_{\text{min}} \) is 0.2; \( \omega_{\text{re}} \) is 0.021; the \( b_L \) is set randomly in the interval [85, 90]; and the \( b_r \) is set randomly in interval [95, 100], the number of parameters optimized by the S-MTCO is \( n_L \cdot D \), and optimized by K-MTCO is \( n_L \cdot D + 1 \), where ‘1’ denotes the kernel parameter \( \sigma \). The \( k \)-NN is sensitive to the parameter \( k \), so we will verify the influence of difference values of \( k \) on the recognition. Besides \( k = 1 \) will be listed in the tables, we will also list the value of \( k \) that makes the recognition accuracy highest and lowest. As for the settings of the parameters and network structures of SVM, BP, and LVQ network, please refer to the references mentioned above. Especially, the error goals for BP and LVQ network are set as 0.001, and the iteration numbers for them are 2,000 and 500, respectively. All 4 data sets should be scaled to be in interval [0, 1], which can reduce the searching range and improve the algorithm’s efficiency.

Tables 3 to 6 show the comparison results using different algorithms. Similarly, random settings also exist in the BP network, the LVQ network, and SVM, and the calculation is repeated 3 times for verifying the influence of the randomness on the recognition accuracy.

Tables 3 to 6 list the ratio between the samples recognized correctly and the total samples of each class. For example, in the second line of the row of the K-MTCO in Table 3, the notation ‘(31/33)’ means that the second class has 33 samples, but only 31 samples are recognized correctly.

These results indicate that the MTCO algorithm is effectual and its accuracy is usually higher than other algorithms. The K-MTCO algorithm has the highest recognition performance. In the following section, the MTCO algorithm will be applied for the fault diagnosis of the bearing.

4. Application in Bearing Fault Diagnosis

Rolling element bearings may lead to fatal breakdowns and even unacceptably long-time maintenance stops if the faults occur [25] as the key components located between the stationary and rotating part of the motors. Therefore, it is significant to recognize unknown fault patterns accurately occurred in the bearings. In this section, some features are firstly extracted from vibration signals of bearing and then the MTCO algorithm is applied for the fault diagnosis of bearing.

4.1. Collection of Bearing Vibration Signals. The vibration signals are obtained from Case Western Reserve University (CWRU) bearing data center. As shown in Figure 5, the test stand consists of a 2 hp (horse power) motor, a torque transducer, and a dynamometer. The test bearings manufactured by Svenska Kullager Fabriken (SKF) support the motor shaft. Single-point faults were introduced to the test bearings using electro-discharge machining (EDM) with fault diameters of 0.007 inches and 0.021 inches. Vibration data are collected by accelerometers for normal bearings, single-point drive-end bearings defects which include outer race fault with defect size of 0.007 inches and 0.021 inches, inner race fault with defect size of 0.007 inches and 0.021 inches, and ball fault with defect size of 0.007 inches and 0.021 inches. Data are collected at 12,000 samples per second for drive-end bearing experiments. In this study, there are 7 original vibration signals covering the 7 drive end bearing conditions including 1 normal condition and 6 types’ faults. Each original signal is divided into 100 subsignals with 12,000 data points.

4.2. Feature Extraction. Four time-domain features such as mean, root mean square (RMS), clearance factor and kurtosis factor are chosen here. In fact, mean describes the static character of the signal; RMS reflects the signal amplitude characteristic, and it is not sensitive to the early faults but with a good stability; clearance and kurtosis factor are dimensionless statistical parameters, which are sensitive to the early faults, but have worse stability. Based on the above consideration, these four features in time domain is chosen. Their expressions are shown in the following equations:
| Table 2: Information on data sets. |
|-----------------------------------|
| Name                | Dimensions | Categories | Train samples | Test samples | Area       | Year |
|----------------------|------------|------------|---------------|--------------|------------|------|
| Wine                 | 13         | 3          | 98 (34 + 38 + 26) | 80 (25 + 33 + 22) | Physical  | 1991 |
| Haberman’s survival  | 3          | 2          | 121 (80 + 41)   | 105 (65 + 40)  | Life      | 1999 |
| Seeds                | 7          | 3          | 120 (40 + 40 + 40) | 90 (30 + 30 + 30) | Life      | 2012 |
| Indian liver patients| 10         | 2          | 213 (106 + 107)  | 120 (60 + 60)   | Life      | 2012 |

| Table 3: The comparison of test samples recognition accuracy for wine. |
|---------------------------------------------------------------------|
| Algorithm | Recognition accuracy 1 | Recognition accuracy 2 | Recognition accuracy 3 |
|-----------|------------------------|------------------------|------------------------|
| K-MTCO    | 97.5% (78/80) (25/25) | 97.5% (78/80) (25/25) | 97.5% (78/80) (25/25) |
| S-MTCO    | 97.5% (78/80) (25/25) | 97.5% (78/80) (25/25) | 96.25% (77/80) (25/25) |
| SVM       | 96.25% (77/80) (24/25) | 97.5% (78/80) (25/25) | 96.25% (77/80) (24/25) |
| BP        | 95% (76/80) (25/25)    | 95% (76/80) (25/25)    | 92.50% (74/80) (25/25) |
| LVQ       | 92.50% (74/80) (25/25) | 92.50% (74/80) (25/25) | 93.75% (77/80) (25/25) |
| k-NN      | k = 1: 91.25% (73/80) (25/25) | k = 1: 96.25% (77/80) (25/25) | k = 1: 96.25% (77/80) (0/25) |

| Table 4: The comparison of test samples recognition accuracy for Haberman’s survival. |
|-------------------------------------------------------------------------------------------------------------------------------------|
| Algorithm | Recognition accuracy 1 | Recognition accuracy 2 | Recognition accuracy 3 |
|-----------|------------------------|------------------------|------------------------|
| K-MTCO    | 69.52% (73/105) (60/65) | 69.52% (73/105) (60/65) | 69.52% (73/105) (60/65) |
| S-MTCO    | 68.57% (72/105) (61/65) | 68.57% (72/105) (61/65) | 68.57% (72/105) (61/65) |
| SVM       | 44.76% (47/105) (19/65) | 44.76% (47/105) (19/65) | 44.76% (47/105) (19/65) |
| BP        | 67.62% (71/105) (62/65) | 67.62% (71/105) (62/65) | 67.62% (71/105) (62/65) |
| LVQ       | 53.33% (56/105) (45/65) | 53.33% (56/105) (45/65) | 53.33% (56/105) (45/65) |
| k-NN      | k = 1: 53.33% (56/105) (33/65) | k = 10: 67.62% (71/105) (58/65) | k = 3: 49.52% (52/105) (31/65) |

| Table 5: The comparison of test samples recognition accuracy for seeds. |
|-------------------------------------------------------------------------|
| Algorithm | Recognition accuracy 1 | Recognition accuracy 2 | Recognition accuracy 3 |
|-----------|------------------------|------------------------|------------------------|
| K-MTCO    | 97.78% (88/90) (29/30) | 97.78% (88/90) (29/30) | 97.78% (88/90) (28/30) |
| S-MTCO    | 95.56% (86/90) (27/30) | 95.56% (86/90) (27/30) | 95.56% (86/90) (27/30) |
| SVM       | 95.56% (86/90) (27/30) | 95.56% (86/90) (27/30) | 95.56% (86/90) (27/30) |
| BP        | 94.44% (85/90) (25/30) | 94.44% (85/90) (25/30) | 94.44% (85/90) (25/30) |
| LVQ       | 95.56% (86/90) (27/30) | 95.56% (86/90) (27/30) | 95.56% (86/90) (27/30) |
| k-NN      | k = 1: 95.56% (86/90) (26/30) | k = 4: 97.78% (88/90) (28/30) | k = 8: 92.22% (83/90) (24/30) |
\[ s_{t(\text{mean})} = \frac{1}{n_t} \sum_{i=1}^{n_t} s_i, \]
\[ s_{t(\text{rms})} = \sqrt{\frac{1}{n_t} \sum_{i=1}^{n_t} s_i^2}, \]
\[ \text{CL}_f(s_t) = \frac{\max(s_t)}{\left( \frac{1}{n_t} \sum_{i=1}^{n_t} |s_i| \right)^2}, \]
\[ K_f(s_t) = \frac{1}{n_f} \sum_{i=1}^{n_f} s_i^4 \left( s_{t(\text{rms})} \right)^4, \]

where \( n_f \) is the total number of data points, \( s_t \) is a time-domain signal, \( s_i \) is the \( i \)th data point of signal \( s_t \), and \( \max(s_t) \) is a function for calculating the maximum data point.

After applying the fast Fourier transform (FFT) to the signal \( s_t \), three features in frequency domain are calculated by the following equations:

\[ F_1 = \frac{1}{n_f} \sum_{i=1}^{n_f} s_i, \]
\[ F_2 = \frac{1}{n_f} \sum_{i=1}^{n_f} (s_i - F_1)^2, \]
\[ F_3 = \frac{1}{n_f} \cdot \sqrt{\frac{1}{n_f} \sum_{i=1}^{n_f} (s_i - F_1)^3}, \]

where \( n_f \) is the total number of spectrum lines, \( s_f \) is a frequency-domain signal, and \( s_i \) is the \( i \)th data point of signal \( s_f \). \( F_1 \) describes the size of vibration energy in frequency domain, \( F_2 \) is deviation that reflect the dispersion or concentration of the spectrum, and \( F_3 \) also reflect the dispersion or concentration of the spectrum.

Table 7 shows the distribution of 7 different classes data. Each class data are divided in a proportion of 60% for training and 40% for testing randomly. Using the last six attributes, two three-dimensional data distribution of bearing conditions is shown in Figure 6. This indicates that the condition data have relatively better separability. Furthermore, Figure 6 also indicates that the feature extraction is helpful for recognizing these fault patterns. Before the recognition, an optimal single-class center is determined for each working condition.

4.3. Fault Diagnosis. Now, the MTCO algorithm is applied to recognize the bearing patterns. Substantially, the proposed algorithm can recognize multiclass fault patterns easily, whereas SVM is designed for recognizing the double-class problems [26]. So the MTCO algorithm can work well when coping with the multiclass fault patterns of bearing. In the subsection, the SVM [27], BP network [23], LVQ network [24], S-MTCO algorithm, and the swarm intelligence based on supervised PSO (S-PSO) classification algorithm [22] are compared with the K-MTCO algorithm, and the calculation is also repeated 3 times for the same purpose. The parameters for SVM are set according to ICD (intercluster distance in the feature space) proposed by Zhang et al. [27], and the parameters for S-PSO classification algorithm are set according to the method proposed by Zheng and Gao [22].
The parameters for other algorithms are as same as that in Section 3.3.

The comparison results shown in Table 8 demonstrate that the MTFO algorithm, especially the K-MTCO algorithm, consistently has higher recognition accuracy than other algorithms. And the results calculated by the K-MTCO algorithm are quite a remarkable achievement from an engineering point of view. So this stable and reliable global optimal decision-making ability completely makes the proposed algorithm powerful for bearing fault diagnosis.

5. Conclusion

In this paper, a novel MTFO algorithm has been proposed. The optimization and recognition performance of the MTFO algorithm is evaluated using different typical characteristic test functions and data sets. The comparisons have been made with various commonly used intelligence algorithms based on human behaviors and pattern recognition. From the results, it can be seen that the MTFO algorithm performs consistently better than other human behavior-based algorithms, such as SRPSO, ALCPSO, SOA, SPSO. Moreover, it can also be concluded that the proposed algorithm has a stable and reliable global optimal decision-making ability. Finally, the MTFO algorithm is applied to identify multiple bearing faults. The recognition results indicate that the recognition accuracy calculated by the MTFO algorithm is better and higher than other commonly used algorithms including SVM, BP, and LVQ especially when the kernel function is introduced to the proposed algorithm.

### Table 7: Description for bearing data.

| The number of samples | For training | For testing | Defect size (inches) | Operation condition | Label |
|-----------------------|--------------|-------------|----------------------|---------------------|-------|
| 100                   | 60           | 40          | 0.007                | Normal              | 1     |
| 100                   | 60           | 40          | 0.021                | Outer race          | 2     |
| 100                   | 60           | 40          | 0.007                | Outer race          | 3     |
| 100                   | 60           | 40          | 0.021                | Inner race          | 4     |
| 100                   | 60           | 40          | 0.007                | Inner race          | 5     |
| 100                   | 60           | 40          | 0.021                | Ball                | 6     |
| 100                   | 60           | 40          | 0.021                | Ball                | 7     |

### Table 8: The comparison of test sample recognition accuracy for bearing data.

| Algorithm | Recognition accuracy 1 | Recognition accuracy 2 | Recognition accuracy 3 |
|-----------|------------------------|------------------------|------------------------|
| K-MTCO    | 99.64% (279/280)       | 99.64% (279/280)       | 99.64% (279/280)       |
| S-MTCO    | 99.29% (278/280)       | 99.29% (278/280)       | 99.29% (278/280)       |
| SVM       | 98.93% (277/280)       | 98.93% (277/280)       | 98.93% (277/280)       |
| BP        | 99.29% (278/280)       | 98.57% (276/280)       | 98.93% (277/280)       |
| LVQ       | 84.64% (237/280)       | 85.71% (240/280)       | 85.00% (238/280)       |
| S-PSO     | 98.93% (277/280)       | 98.21% (275/280)       | 98.21% (275/280)       |

Figure 6: Data distribution of bearing faults: (a) time-domain feature distribution and (b) frequency-domain feature distribution.
Data Availability

The data used to support the findings of this study are included within the article.

Disclosure

The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

Conflicts of Interest

The authors declare no conflict of interest.

Acknowledgments

This research was supported by Sichuan Science and Technology Program, grant numbers 2019YJ0395 and 2021YJ0519, China Civil Aviation Administration Development Foundation Educational Talents Program (14002600100018J034), General Foundation of Civil Aviation Flight University of China (2019-053), Youth Foundation of Civil Aviation Flight University of China (Q2018-139).

References

[1] K. Li, P. Chen, and S. Wang, “An intelligent diagnosis method for rotating machinery using least squares mapping and a fuzzy neural network,” Sensors, vol. 12, no. 5, pp. 5919–5939, 2012.

[2] Y. F. Li, H. Z. Huang, N. Liu, and H. Xiao, “A new fault tree analysis method: fuzzy dynamic fault tree analysis,” Eksplotacija i Niezawodnosc-Maintenance and Reliability, vol. 14, no. 3, pp. 208–214, 2012.

[3] H.-Z. Huang, Y. Liu, Y. Li, L. Xue, and Z. Wang, “New evaluation methods for conceptual design selection using computational intelligence techniques,” Journal of Mechanical Science and Technology, vol. 27, no. 3, pp. 733–746, 2013.

[4] H.-Z. Huang, H.-K. Wang, Y.-F. Li, L. Zhang, and Z. Liu, “Support vector machine based estimation of remaining useful life: current research status and future trends,” Journal of Mechanical Science and Technology, vol. 29, no. 1, pp. 151–163, 2015.

[5] B. Zheng, H. Gao, X. Ma, and X. Zhang, “Graph partition based on dimensionless similarity and its application to fault diagnosis,” IEEE Access, vol. 9, pp. 35573–35583, 2021.

[6] S.-F. Yuan and F.-L. Chu, “Fault diagnostics based on particle swarm optimisation and support vector machines,” Mechanical Systems and Signal Processing, vol. 21, no. 4, pp. 1787–1798, 2007.

[7] B. Zheng and X. Ma, “Application on damage types recognition in civil aeroengine based on SVM optimized by DMPSO,” Computer Science, vol. 47, no. 11A, pp. 132–138, 2020.

[8] M. Korürek and B. Doğan, “ECG beat classification using particle swarm optimization and radial basis function neural network,” Expert Systems with Applications, vol. 37, no. 12, pp. 7563–7569, 2010.

[9] L. Lu, J. Yan, and C. W. de Silva, “Dominant feature selection for the fault diagnosis of rotary machines using modified genetic algorithm and empirical mode decomposition,” Journal of Sound and Vibration, vol. 344, pp. 464–483, 2015.

[10] W. Shang, X. Zhou, and J. Yuan, “An intelligent fault diagnosis system for newly assembled transmission,” Expert Systems with Applications, vol. 41, no. 9, pp. 4060–4072, 2014.

[11] B. Zheng, Research on Aeroengine Fault Diagnosis and Performance Parameters Prediction Based on Particle Swarm Optimization, University of Electronic Science and Technology, Chengdu, China, 2018.

[12] M. R. Tanweer, S. Suresh, and N. Sundararajan, “Self regulating particle swarm optimization algorithm,” Information Sciences, vol. 294, pp. 182–202, 2015.

[13] R. Cheng and Y. Jin, “A social learning particle swarm optimization algorithm for scalable optimization,” Information Sciences, vol. 291, pp. 43–60, 2015.

[14] Y. Li, Z.-H. Zhan, S. Lin, J. Zhang, and X. Luo, “Competitive and cooperative particle swarm optimization with information sharing mechanism for global optimization problems,” Information Sciences, vol. 293, pp. 370–382, 2015.

[15] B. Zheng, X. Ma, H. Y. Gao, and X. Q. Zhang, “Direct position updating-based trying-mutation particle swarm optimization algorithm and its application on reliability optimization,” Mechanical Science and Technology for Aerospace Engineering, vol. 40, no. 1, pp. 155–164, 2021.

[16] F.-Q. Lu, M. Huang, W.-K. Ching, and T. K. Siu, “Credit portfolio management using two-level particle swarm optimization,” Information Sciences, vol. 237, pp. 162–175, 2013.

[17] H. Wang, H. Sun, C. Li, S. Rahnamayan, and J.-s. Pan, “Diversity enhanced particle swarm optimization with neighborhood search,” Information Sciences, vol. 223, pp. 119–135, 2013.

[18] C. H. Dai, Seeker Optimization Algorithm and Its Application, Southwest Jiaotong University, Chengdu, China, 2009.

[19] R. P. Singh, V. Mukherjee, and S. P. Ghoshal, “Optimal reactive power dispatch by particle swarm optimization with an aging leader and challengers,” Applied Soft Computing, vol. 29, pp. 398–309, 2015.

[20] G. Mehmet and A. Ethem, “Cost-conscious multiple kernel learning,” Pattern Recognition Letters, vol. 31, pp. 959–965, 2010.

[21] V. N. Vapnik, “An overview of statistical learning theory,” IEEE Transactions on Neural Networks, vol. 10, no. 5, pp. 988–999, 1999.

[22] B. Zheng and F. Gao, “Fault diagnosis method based on S-PSO classification algorithm,” Acta Aeronautica et Astronautica Sinica, vol. 36, no. 11, pp. 3640–3651, 2015.

[23] J. Ben Ali, N. Fnaiech, L. Saidi, B. Chebel-Morello, and F. Fnaiech, “Application of empirical mode decomposition and artificial neural network for automatic bearing fault diagnosis based on vibration signals,” Applied Acoustics, vol. 89, pp. 16–27, 2015.

[24] B. Biswal, M. Biswal, S. Hasan, and P. K. Dash, “Nonstationary power signal time series data classification using LVQ classifier,” Applied Soft Computing, vol. 18, pp. 158–166, 2014.

[25] C. Shen, D. Wang, F. Kong, and P. W. Ise, “Fault diagnosis of rotating machinery based on the statistical parameters of wavelet packet paving and a generic support vector regressor classifier,” Measurement, vol. 46, no. 4, pp. 1551–1564, 2013.

[26] K. Zhu, X. Song, and D. Xue, “A roller bearing fault diagnosis method based on hierarchical entropy and support vector machine with particle swarm optimization algorithm,” Measurement, vol. 47, pp. 669–675, 2014.

[27] X. Zhang, D. Qiu, and F. Chen, “Support vector machine with parameter optimization by a novel hybrid method and its application to fault diagnosis,” Neurocomputing, vol. 149, pp. 641–651, 2015.