Article

Power Transformer Fault Diagnosis Based on Random Forest and Improved Particle Swarm Optimization–Backpropagation–AdaBoost

Lei Zhou [1], Zhongjun Fu [1,*], Keyang Li [2], Yuhui Wang [1] and Hang Rao [1]

[1] School of Computer Engineering, Jiangsu University of Technology, Changzhou 213001, China; zhoulei183810@163.com (L.Z.); w17550777108@163.com (Y.W.); r18970306832@126.com (H.R.)
[2] Faculty of Engineering, The University of Sydney, Sydney, NSW 2006, Australia; anque1932@gmail.com
* Correspondence: june@jsut.edu.cn

Abstract: This paper proposes a novel fault diagnosis methodology for oil-immersed transformers to improve the diagnostic accuracy influenced by gas components in power transformer oil. Firstly, the Random Forest (RF) algorithm is utilized to evaluate and filter the raw data features, solving the problem of determining significant features in the dataset. Secondly, a multi-strategy Improved Particle Swarm Optimization (IPSO) is applied to optimize a double-hidden layer backpropagation neural network (BPNN), which overcomes the challenge of determining hyperparameters in the model. Four enhancement strategies, including SPM chaos mapping based on opposition-based learning, adaptive weight, spiral flight search, and crisscross strategies, are introduced based on traditional Particle Swarm Optimization (PSO) to enhance the model's optimization capabilities. Lastly, AdaBoost is integrated to fortify the resilience of the IPSO-BP network. Ablation experiments demonstrate an enhanced convergence rate and model accuracy of IPSO. Case analysis using Dissolved Gas Analysis (DGA) samples compares the proposed IPSO–BP–AdaBoost model with other swarm intelligence optimization algorithms integrated with BPNN. The experimental findings highlight the superior diagnostic accuracy and classification performance of the IPSO–BP–AdaBoost model.

Keywords: power transformer; fault diagnosis; random forest; backpropagation neural network; improved particle swarm optimization; AdaBoost

check for updates

Citation: Zhou, L.; Fu, Z.; Li, K.; Wang, Y.; Rao, H. Power Transformer Fault Diagnosis Based on Random Forest and Improved Particle Swarm Optimization–Backpropagation–AdaBoost. *Electronics* **2024**, *13*, 4149. https://doi.org/10.3390/electronics13214149

Academic Editor: Ahmed Abu-Siada

Received: 26 September 2024
Revised: 17 October 2024
Accepted: 21 October 2024
Published: 22 October 2024

1. Introduction

The oil-immersed transformer serves as the central component of the power grid, playing a crucial role in the transmission of electrical power. When the transformer malfunctions, it poses a dual risk to public safety and the economy, including incurring high costs for equipment repairs and significant financial losses resulting from power disruptions. Hence, it is necessary to precisely identify the internal problems of transformers beforehand to ensure the secure and reliable operation of the power grid system [1].

Many electrical power experts developed diagnostic methods for transformers in the 1960s. These traditional methods included vibration testing, infrared testing, voltage and current measurements, and dissolved gas analysis in oil [2]. Among these, the dissolved gas analysis primarily determines the type of transformer fault based on the relative concentrations of gases such as H_2, CH_4, C_2H_6, C_2H_4, and C_2H_2 produced during insulation failures or aging [3]. However, a limitation of this method is that the diagnostic results rely on expert experience and observations of the equipment parameters. With the emergence of artificial intelligence technology, various machine learning methods, including Support Vector Machines (SVMs), Multi-Layer Perceptrons (MLPs), and Extreme Learning Machines (ELMs), have been applied to transformer fault diagnosis nowadays.

Jin et al. [4] enhance the BP neural network by stacking multiple residual network modules and introducing an SVM to evaluate the extracted feature vectors from each layer. Ivanov and Palyukh [5] present a method for creating a training dataset specifically tailored to fuzzy neural networks, which enables a rapid probability estimation of abnormal critical events or accident causes within a transformer diagnostic system. Research by Kari et al. [6] employs a deep belief network to extract features from dissolved gases in oil and perform mean clustering for transformer fault classification. Song et al. propose [7] a novel transformer fault diagnosis model called Meta–OSSA–KELM, which combines Meta-Learning with a Kernel Extreme Learning Machine and opposition-based learning Sparrow Search Algorithm. By integrating chaos mapping and opposition-based learning concepts into the Sparrow Search Algorithm, the experiment results indicate that this model provides a stable and high-performance approach for transformer fault diagnosis.

While the above methods have achieved beneficial results, determining the optimal structure of the diagnostic model and its hyperparameters often faces challenges, limiting further enhancements in diagnostic effectiveness. Furthermore, the uncertainty regarding the types and quantities of features contained within fault samples restricts the widespread application of deep learning and other neural network methods [8]. Moreover, input features for fault diagnosis models typically rely on IEC or IEEE standard gas concentrations, gas ratios, or relative percentages. Nevertheless, a universally accepted feature set for diagnosing faults in oil-immersed transformers remains elusive. Consequently, previous studies have encountered issues such as a plethora of gas feature types, leading to inadequate analysis of features [9].

Therefore, the shortcomings leave significant room for further improving fault diagnosis performance. To address these two issues, this paper proposes a transformer fault diagnosis model based on feature selection and IPSO–BP–AdaBoost. The model first establishes an optimal feature set predicated on differences in detection accuracy caused by varying numbers of input features, employing the Random Forest (RF) method for feature selection. Then, it utilizes the multi-strategy Improved Particle Swarm Optimization (IPSO) method to optimize the key parameters in the backpropagation neural network (BPNN). Lastly, the AdaBoost algorithm is applied to enhance the robustness and generalization of the diagnosis model.

2. Feature Selection Based on Random Forest

In case of transformer failure, gases of different compositions and concentrations are released, and the specific gas volume fraction increases rapidly, mainly including methane (CH_4), ethane (C_2H_6), ethylene (C_2H_4), acetylene (C_2H_2), and hydrogen (H_2). The relationship between dissolved gases in oil and faults is as follows:

(1) When high-energy discharge occurs inside the transformer, characteristic gases such as acetylene C_2H_2 and H_2 are mainly produced in the oil, followed by C_2H_4, CH_4, and C_2H_6.
(2) When low-energy discharge occurs inside the transformer, the dissolved gases in the oil are mainly C_2H_2 and H_2, followed by CH_4 and C_2H_4.
(3) When partial discharge occurs inside the transformer, the characteristic gases produced in the oil vary with the discharge energy density. When the discharge density is below 10^{-9}C, the total hydrocarbon content is generally not high, and the main component is H_2, followed by CH_4.
(4) When the temperature at the fault point is low, the proportion of CH_4 is high. As the hot spot temperature increases (above 500 °C), the content of C_2H_4 and H_2 components sharply increases, with C_2H_4 content exceeding CH_4, but with H_2 content generally not exceeding 30% of the hydrogen hydrocarbon content [10].

Considering the large dispersion of contents among gases, the diagnosis accuracy of the traditional ratio method is only about 70% [11]. Therefore, this study created a comprehensive and diverse feature set. Through investigating relevant standards and publications on dissolved gas analysis at home and abroad, the generation rules of dissolved gas in transformer oil under different working conditions are summarized, and the non-code ratios of gases are introduced into fault diagnosis. Ref. [12] verified that non-code ratios as the input of the fault diagnosis model can effectively improve the accuracy of transformer fault diagnosis.

In this study, the fault sample dataset was collected from Refs. [13,14] and IEC TC10 databases, and a total of 596 DGA samples with known transformer operating states were selected. Six types of faulty samples are under consideration, including normal condition (NC), partial discharge (PD), low-energy discharge (D_1), high-energy discharge (D_2), medium- and low-temperature thermal defect (T_1), and high-temperature thermal defect (T_2), as shown in Table 1.

Table 1. Fault sample distribution.

Working Status	Category Label	Total Number of Samples
D_1	1	108
D_2	2	86
T_2	3	104
T_1	4	119
PD	5	113
NC	6	66

H_2, CH_4, C_2H_6, C_2H_4, and C_2H_2 are selected as the original gases, and 26 groups of gas features are obtained by combining the gas concentration, three-ratio method, and non-code ratio method, as shown in Table 2.

Table 2. Feature sets.

Number	Feature	Number	Feature
1	H_2	14	C_2H_2/TH
2	CH_4	15	$(CH_4 + C_2H_2/TH$
3	C_2H_6	16	$(CH_4 + C_2H_4)/TH$
4	C_2H_4	17	$(CH_4 + C_2H_6)/TH$
5	C_2H_2	18	$(C_2H_4 + C_2H_2)/TH$
6	TH	19	$(C_2H_6 + C_2H_2)/TH$
7	H_2/TH	20	$(C_2H_4 + C_2H_6)/TH$
8	CH_4/TH	21	TG
9	C_2H_2/C_2H_4	22	H_2/TG
10	CH_4/H_2	23	$(H_2 + CH_4)/TG$
11	C_2H_4/C_2H_6	24	$(H_2 + C_2H_4)/TG$
12	C_2H_6/TH	25	$(H_2 + C_2H_6)/TG$
13	C_2H_4/TH	26	$(H_2 + C_2H_2)/TG$

TH is the total hydrocarbon gas content, and TG is the total content of the five original characteristic gases.

Some of the 26 gas compositions may not be directly relevant to fault diagnosis, potentially leading to redundancy. In addition, the multi-dimensional dataset has an impact on the running time of the classification model. Therefore, it is crucial to determine an informative and concise subset of features. In this study, the out-of-bag data (OOB) error rate in Random Forest was used to calculate the importance of each feature. A higher ranking indicates a greater contribution of the feature to fault diagnosis. OOB error rate is the model error rate calculated on the out-of-bag data. Specifically, it involves using each decision tree to predict the corresponding out-of-bag error rate and then calculating the proportion of samples that predict errors [15]. The specific process is as follows.

Step 1: Calculate the initial OOB error rate:

For each decision tree in the Random Forest, its corresponding out-of-bag data error rate is calculated, denoted as err_1.

Step 2: Introduce noise interference:

Select a specific feature X and randomly add noise to all sample values of this feature X in the out-of-bag data.

Step 3: Recalculate OOB error rate:

After introducing noise interference to feature X, the error rate of data outside the bag is calculated again and denoted as err_2.

Step 4: Importance of calculation features:

The importance of feature X can be measured by comparing the difference between err_1 and err_2. Specifically, the importance of feature X can be expressed as the mean of $(err_2 - err_1)$; the greater the value, the greater the influence of feature X on the model's prediction and therefore the greater its importance.

In the experiment, the RF algorithm was run 30 times to eliminate randomness, and the average importance of 26 input features was calculated. Based on the average value, a ranking list was created accordingly. Subsequently, these features are sequentially used as inputs by incrementing or decrementing their ranking order until all features are considered. The importance of inputs and their corresponding diagnostic accuracy are depicted in Figure 1. For this research, the minimum leaf size is set at 8, and the number of decision trees is set at 400.

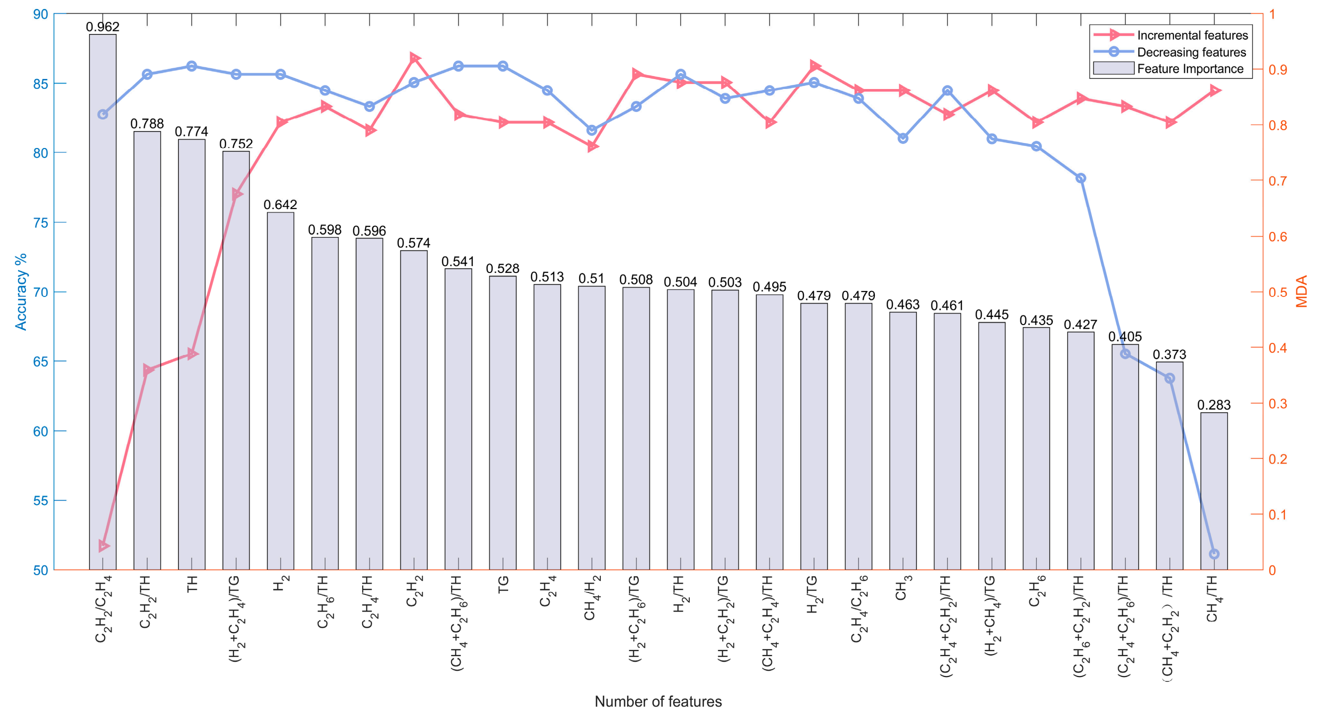

Figure 1. Feature importance evaluation and diagnostic accuracy curves.

The red curve illustrates how diagnostic accuracy evolves as more elements are incorporated, whereas the blue curve demonstrates the accuracy changes as elements are excluded. As can be seen from the red curve, the diagnostic accuracy increases from 51% to 87% as the first eight features are added sequentially, indicating that these gas combinations contribute greatly to the accuracy of the results. However, after the eighth feature, the accuracy remains relatively stable at around 83% $\pm$ 3, suggesting that the further addition of features does not lead to a significant increase in accuracy, suggesting that these additional features may be considered redundant. Through the analysis of the blue curve, the diagnostic accuracy gradually improves as the number of elements decreases from 26 to 14. As the number of features continues to decline, the diagnostic performance fluctuates slightly but remains at about 85%

These results indicate that not all gas combinations are crucial for accurate fault diagnosis models. The inclusion of features 9 to 26 did not significantly enhance diagnostic accuracy, and the reduction in some features could even improve it. In summary, this research selects a subset of eight of the most critical gas characteristics, which can reduce the workload of the model while providing optimal diagnostic performance, as shown in Table 3.

Table 3. Optimal feature subset.

Number	Feature Item	Number	Feature Item
1	C_2H_2/C_2H_4	5	H_2
2	C_2H_2/TH	6	C_2H_6/TH
3	TH	7	C_2H_4/TH
4	$(H_2 + C_2H_4)/TG$	8	C_2H_2

3. BPNN Network Model

The backpropagation neural network (BPNN) is a multi-layer feedforward neural network trained with the error backpropagation algorithm. It comprises an input, a hidden, and an output layer, exhibiting classification capabilities for arbitrarily complex problems and excellent multi-dimensional function mapping abilities. It solves the XOR problem, which simple perceptrons are unable to handle. Its workflow primarily consists of forward and backward propagation, with the forward propagation formula being:

$$u_j = f\left(\sum_{i=1}^{n} w_{ij}x_i + \overset{u}{\underset{j}{\theta}}\right) \tag{1}$$

$$y = f\left(\sum_{j=1}^{m} w_j u_j + \theta^y\right) \tag{2}$$

where x_i represents the input variable; y represents the output variable; u_j is the hidden layer output; f is the mapping relationship of the activation function; w_{ij} is the weight of the ith input variable and the jth hidden layer neuron; θ_j^u is the offset term of the jth neuron of hidden layer u; w_j is the weight of the jth neuron connected with y; θ^y is the offset of y. The loss function is the mean square error between the real value and the predicted value, and the objective function is:

$$E = \frac{1}{n}\sum_{i=1}^{n}(|y_i - \hat{y}_i|)^2 \tag{3}$$

where y_i is the predicted value and $\hat{y}_i$ is the actual value. In the process of backpropagation, the gradient descent method is used to adjust the weights and biases of each neuron according to the error. The updating formulas of the weights and biases of the output layer are as follows:

$$w_{ij} = w_{ij} - \alpha\left(\delta_j\left(n^{(u-1)}\right)^T + w_{ij}\right) \tag{4}$$

$$\theta_j^u = \theta_j^u - \alpha\delta_j \tag{5}$$

where α is the learning rate, which determines the speed of moving parameters to the optimal value, and δ_j is the error term of neuron j.

In addition, a BPNN typically contains one or more hidden layers that introduce nonlinear transformations to approximate complex nonlinear functions. The addition of a hidden layer can enhance the nonlinear fitting ability of the model and improve the expression ability of the model. For some complex problems, a single hidden layer may not be enough to capture the underlying structure of the data, while two hidden layers can provide a richer representation of features [16]. The backpropagation neural network structure used in this paper is shown in Figure 2.

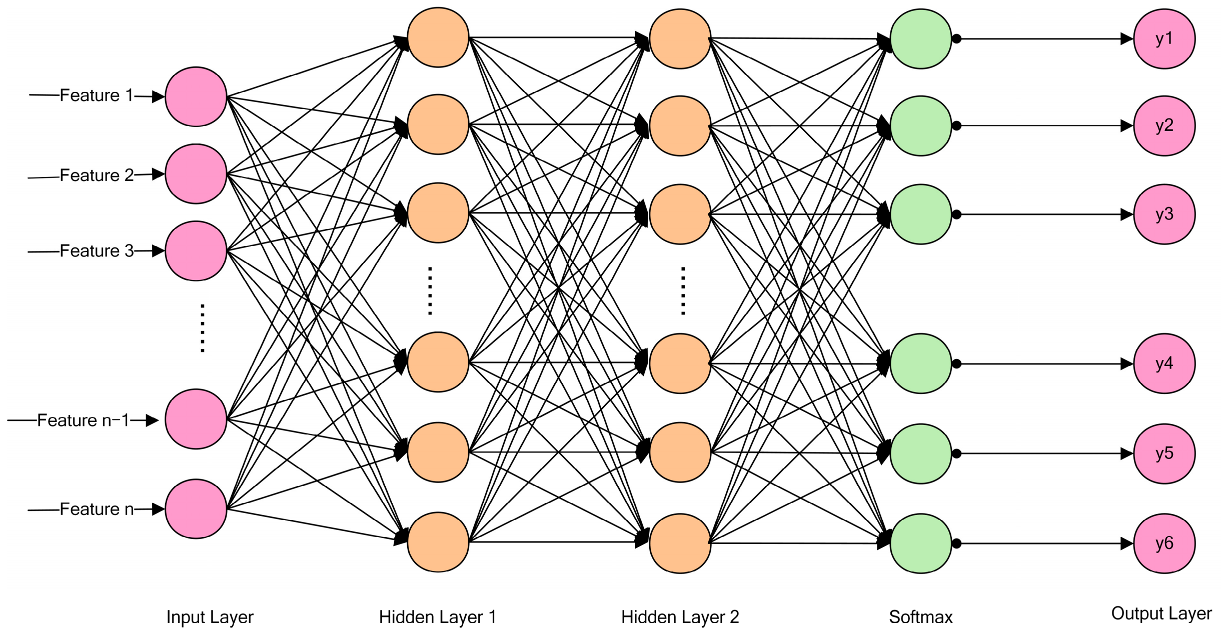

Figure 2. Structure of BPNN.

4. Theoretical Basis of AdaBoost Algorithm

AdaBoost is an ensemble learning method that assigns initial weights to training samples, trains the first weak classifier, and dynamically adjusts sample weights based on the classification accuracy of the weak classifier, giving more attention to misclassified samples [17,18]. After each round, the weight of the weak classifier is adjusted based on its error ϵ_t. The formula for calculating the weight of the weak classifier is as follows:

$$\alpha_t = \frac{1}{2}\ln\left(\frac{1-\epsilon_t}{\epsilon_t}\right) \tag{6}$$

Before generating the next base classifier, AdaBoost will amplify the weights of the misclassified samples while reducing the weights of the correctly classified samples. This approach makes the next iteration of the algorithm more focused on the misclassified samples [19,20]. The formula for updating the sample weights is as follows:

$$w_{t+1}(i) = \frac{w_t(i)log_e(-\alpha_t y_i H_t(x_i))}{Z_t} \tag{7}$$

Finally, a strong classifier is obtained based on the weights of all weak classifiers. When the number of weak classifiers is set to T, the strong classifier outputs according to the following formula:

$$f(x) = \sum_{t=1}^{T} \alpha_t H_t(x_i) \tag{8}$$

where α_t represents the weight of each weak classifier and $H_t(x_i)$ represents the output of each weak classifier. The model structure based on BP–AdaBoost is shown in Figure 3.

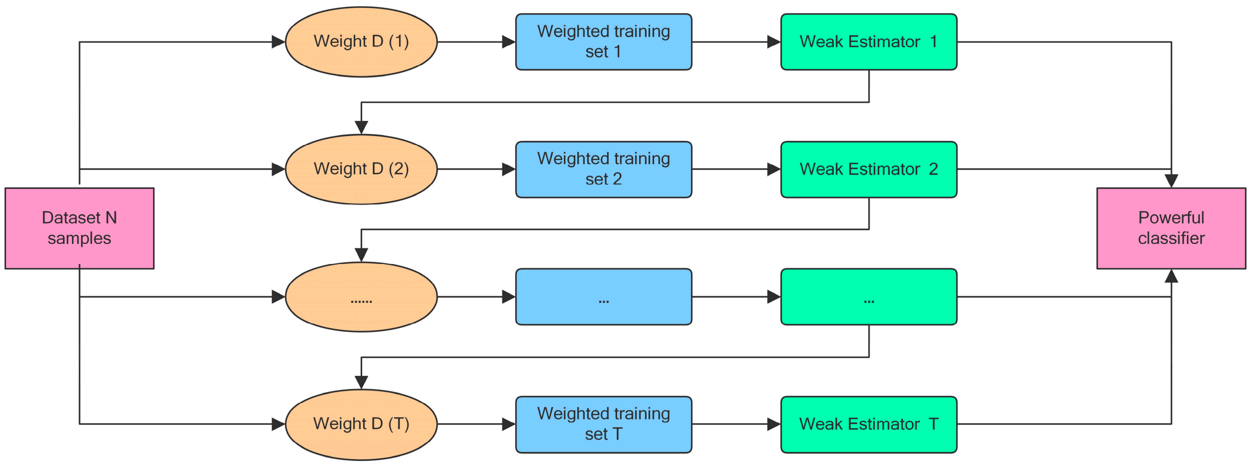

Figure 3. AdaBoost algorithm model.

5. The Improved Particle Swarm Optimization Algorithm Used in This Research

The performance of a backpropagation neural network is often highly sensitive to the initial weight setting. Inappropriate initial weights may lead to the network converging to suboptimal solutions or failing to converge altogether [21]. To address this issue, this paper utilizes an enhanced IPSO algorithm to optimize the weights and biases of neurons in the BP network. By using prediction accuracy as the objective function, the IPSO algorithm aims to minimize the fitness value of particles while achieving rapid convergence.

5.1. The Basic Principle of Particle Swarm Optimization

In a D-dimensional target search space, there are N particles forming a community, each of which is a D-dimensional vector, and its spatial position can be represented as:

$$x_i = (x_{i,1}, x_{i,2}, \ldots, x_{i,D}) \tag{9}$$

The flight speed of the *i*th particle is also a D-dimensional vector, denoted as:

$$v_i = (v_{i,1}, v_{i,2}, \ldots, v_{i,D}) \tag{10}$$

The optimal position found by the *i*th particle is called the individual optimal position, and the optimal position in the population is called the global optimal position, respectively, denoted as:

$$Pbest_i = (Pbest_{i,1}, Pbest_{i,2}, \ldots, Pbest_{i,D}) \tag{11}$$

$$Gbest_i = (Gbest_{i,1}, Gbest_{i,2}, \ldots, Gbest_{i,D}) \tag{12}$$

In traditional particle swarm optimization algorithms, the position of particles is determined by their position and velocity in the previous iteration. Each particle demonstrates individual and collective behavior, adjusting its speed and position based on the historical optimal values of both the individual and the population [22]. The velocity updating formula of particles in the particle swarm is:

$$v_{ij}(t+1) = wv_{ij}(t) + c_1 r_1 \left(Pbest_{ij}(t) - x_{ij}(t)\right) + c_2 r_2 \left(Gbest_{ij}(t) - x_{ij}(t)\right) \tag{13}$$

The position updating formula of particles is:

$$x_{ij}(t+1) = x_{ij}(t) + v_{ij}(t+1) \tag{14}$$

Based on the formula, it is evident that the initialization stage plays a crucial role in the optimization effect of the swarm intelligence algorithm itself, as the random distribution of the particle swarm and its uncertain quality can sometimes lead to local optima in early iterations. Furthermore, subsequent iterations face challenges in achieving fast convergence due to the lack of a mutation mechanism. To address these issues, this paper proposes four collaborative optimization strategies that aim to enhance global search and local exploration capabilities, accelerate convergence, and improve the algorithm's accuracy.

5.2. SPM Chaos Mapping Based on Opposition-Based Learning

In the traditional particle swarm optimization algorithm, the population initialization process exhibits strong randomness, often resulting in poor performance of each particle. Chaos mapping, known for its excellent space ergodicity, can distribute particles more evenly [23] and is commonly employed in various swarm algorithms. Among these methods, the SPM chaotic function has been shown to enhance the diversity of particles in the population [24]. Therefore, this paper adopts the SPM function as the chaotic function. The SPM function formula is:

$$x(i+1) = \begin{cases} mod\left(\frac{x(i)}{\eta} + \mu sin(\pi x(i)) + r\,,1\right), \\ 0 \le x(i) < \eta \\ mod\left(\frac{x(i)}{0.5-\eta} + \mu sin(\pi x(i)) + r\,,1\right), \\ 0 \le x(i) < 0.5 \\ mod\left(\frac{\frac{(1-x(i))}{\eta}}{0.5-\eta} + \mu sin(\pi(1-x(i))) + r\,,1\right), \\ 0.5 \le x(i) < 1-\eta \\ mod\left(\frac{(1-x(i))}{\eta} + \mu sin(\pi(1-x(i))) + r\,,1\right), \\ 1-\eta \le x(i) < 1 \end{cases} \tag{15}$$

where $x(i+1)$ represents the (i+1)th particle, $\eta \in (0,1)$, and when $\mu \in (0,1)$ the system is in a chaotic state. The generated initial solution is determined based on the upper and lower bounds U_b and L_b of the optimized weights and biases. The particle distribution and spectrogram of the SPM chaotic function and three other chaotic functions are shown in Figure 4.

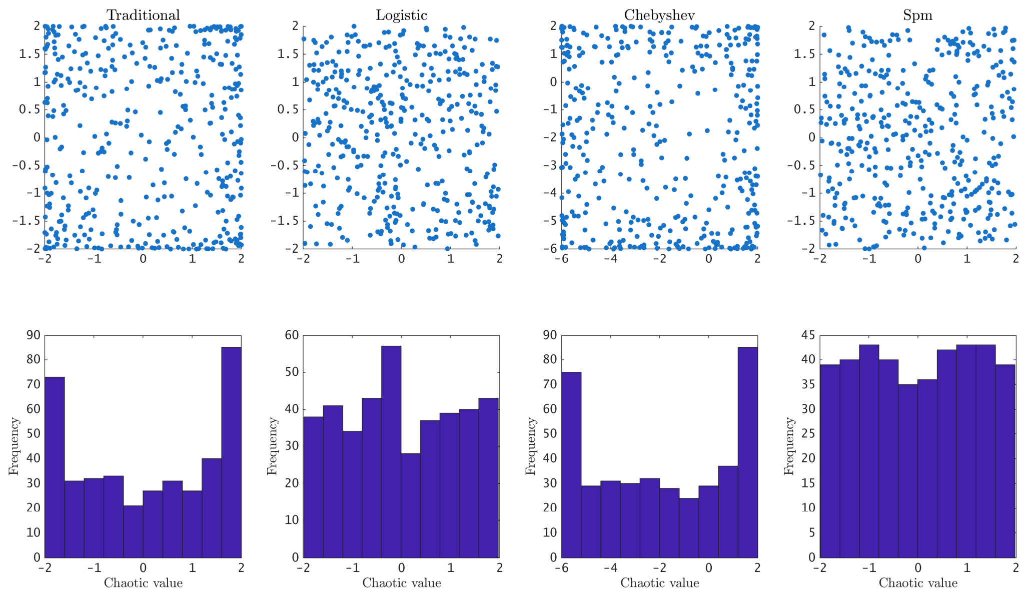

Figure 4. Particle distributions and spectrograms when using four chaotic functions.

When the population size is large, chaos mapping significantly enhances the convergence efficiency of the algorithm. However, obtaining points of good quality is often challenging when the population size is small. Therefore, this paper introduces an opposition-based learning (OBL) strategy [25], which aims to find as many high-quality points as possible to accelerate the convergence speed. Initially, N solutions are generated using SPM mapping during population initialization, followed by the generation of N opposite solutions using opposition-based learning. Subsequently, fitness evaluation and sorting are conducted for a total of 2N solutions, with the top N solutions selected based on better fitness. This approach enhances the diversity and exploration capabilities of the population during the search process. Such a strategy facilitates the algorithm in escaping local optima and accelerating convergence rates. The effectiveness of this strategy has been validated through the improvement of various swarm intelligence optimization algorithms.

The SPM chaotic function is then reapplied to evenly distribute the selected solutions in the solution space. The formula for opposition-based learning is as follows:

$$x_o(t) = (L_b + U_b) - J * x(t) \tag{16}$$

where L_b and U_b are the lower and upper bounds of solution variables, respectively. J is a random number in the range [0,1].

5.3. Nonlinear Adaptive Weight

According to the updating formula of particle velocity, the influence of $v_{ij}(t)$ on $v_{ij}(t+1)$ mainly depends on the size of the weight w. Therefore, this paper introduces a new nonlinear function to calculate w. The nonlinear function formula is:

$$w = \left(1 - \frac{t}{t_{max}}\right)^{\frac{1}{3}} \tag{17}$$

During the initial phases of iteration, the value of w should be relatively large with a small rate of change, enabling particles to conduct a wide range of explorations across the entire solution space. During the later stages of iteration, the value of w should decrease but have a high rate of change, which facilitates local exploration and accelerates convergence. Therefore, Equation (17) possesses functional attributes that effectively fulfill the needs for both global and local particle exploration. Figure 5 shows the comparison of the change curve of w during the iteration process before and after the improvement.

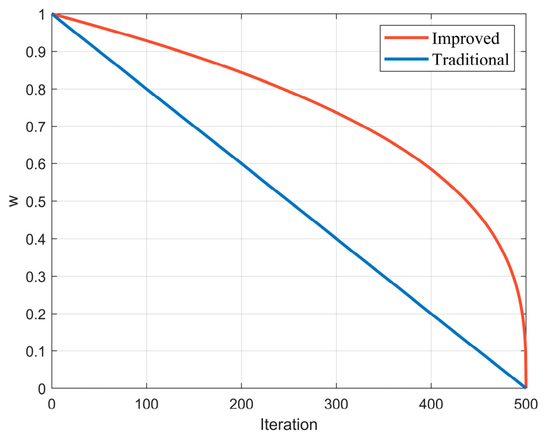

Figure 5. The curve of inertial weight.

5.4. Crossover Mutation Strategy

To enhance the diversity of the population and avoid becoming stuck in suboptimal solutions, this paper introduces a mutation strategy to modify the population and global optimal solutions [26]. In the early stages of particle iteration, as the individual positions within the particle swarm may still be relatively dispersed, employing horizontal crossover during this phase can further augment particle diversity, facilitate global searches, and prevent premature convergence to local optima. The mathematical formula for horizontal crossover is:

$$Mx_{i,d}^{hc} = r_1 * x_{i,d} + (1 - r_1) * x_{j,d} + s_1 * \left(x_{i,d} - x_{j,d}\right) \tag{18}$$

$$Mx_{j,d}^{hc} = r_2 * x_{j,d} + (1 - r_2) * x_{i,d} + s_2 * \left(x_{j,d} - x_{i,d}\right) \tag{19}$$

where $x_{i,d}$ and $x_{j,d}$ represent the parents x_i and x_j of dimension d, respectively; $Mx_{i,d}^{hc}$ and $Mx_{j,d}^{hc}$ are the offspring of x_i and x_j after crossing in the d dimension; r_1 and r_2 are random numbers in the range $[0, 1]$, which are used to control the weight distribution in the crossover operation. In horizontal crossover, these random numbers determine the degree of information mixing between parent particles. r_1 controls the linear combination ratio between $x_{i,d}$ and $x_{j,d}$, while $(1 - r_1)$ controls the linear combination ratio of another part. Because r_1 is random, each crossover operation will generate different offspring particles, which helps to increase the diversity of the population. s_1 and s_2 are random numbers in the range $[-1, 1]$, which are used to introduce disturbance or mutation in the crossover operation. This disturbance helps the algorithm jump out of the local optimal solution and explore a broader search space. In the formula, $s_2 * \left(x_{j,d} - x_{i,d}\right)$ is a disturbance term, which adjusts the value of offspring particles according to the difference between $x_{i,d}$ and $x_{j,d}$ and the random value of s_1.

Through the randomness of r_1 and r_2, each crossover operation will generate different offspring particles, which increases the diversity of the population. At the same time, the disturbance introduced by s_1 and s_2 makes the offspring particles different from the parent particles, which further increases the diversity of the population.

In the later stages of iteration, the particle swarm may gradually converge around local optimal solutions. Vertical crossover can assist in breaking the positional inertia of particles and guide them toward new search areas by performing crossover operations in different dimensions [27]. The mathematical equation for vertical crossover is:

$$Mx_{i,d}^{vc} = p * x_{i,d1} + (1 - p) * x_{i,d2} \tag{20}$$

where $Mx_{i,d}^{vc}$ is generated by the vertical crossover of x_i in the $d1$ and $d2$ dimensions; $p \in (0, 1)$. By employing both horizontal crossover and vertical crossover, particles can share information and update their positions across many dimensions and particles. This enables them to thoroughly explore the search space.

5.5. Spiral Search Strategy

Inspired by the prey encirclement behavior of the leading whales in the Whale Optimization Algorithm (WOA), in the iterative process of the whale algorithm, individual whales utilize a spiral search strategy to update their positions relative to prey. This not only ensures the convergence speed of the algorithm but also enhances individual diversity [28]. In the early stages of the search phase, this paper assigns a higher probability of selecting the traditional PSO particle update strategy to improve the convergence speed of the algorithm, and in the later stages, it selects the spiral search method with a higher probability to accelerate convergence [29,30]. The spiral formula is as follows:

$$x(t+1) = D' \cdot e^{z \cdot l} \cdot \cos(2\pi l) + x^*(t+1) \tag{21}$$

$$D' = |x^*(t+1) - x(t)| \tag{22}$$

where l represents a random number in the range $[0, 1]$. The spiral parameter Z cannot remain constant because it limits the search to a monotonic pattern, which may trap particles in local optima and impair the overall search capability of the algorithm. Consequently, Z is designed as an adaptive variable, dynamically adjusting the spiral shape of particle search trajectories. This design enhances the particles' exploration of unknown regions, ultimately improving the algorithm's search efficiency and global search performance. The formula is as follows:

$$Z = e^{k \cdot \cos(\pi \cdot (1 - \frac{t}{t_{max}}))} \tag{23}$$

where k represents the variation coefficient, and in this paper, $k = 5$. Z varies with the number of operations. Specifically, the value of Z will vary between e^{-k} and e^k. When Z approaches its maximum value (e^k) and l approaches 1, e^k will be very large, thus having a significant impact on the particle position. On the other hand, when Z approaches its minimum value (e^k) and l approaches 0, e^{zl} approaches 1, with a smaller impact on the particle position. Through the above stages, the whole process of the IPSO algorithm is complete, and its flowchart is shown in Figure 6.

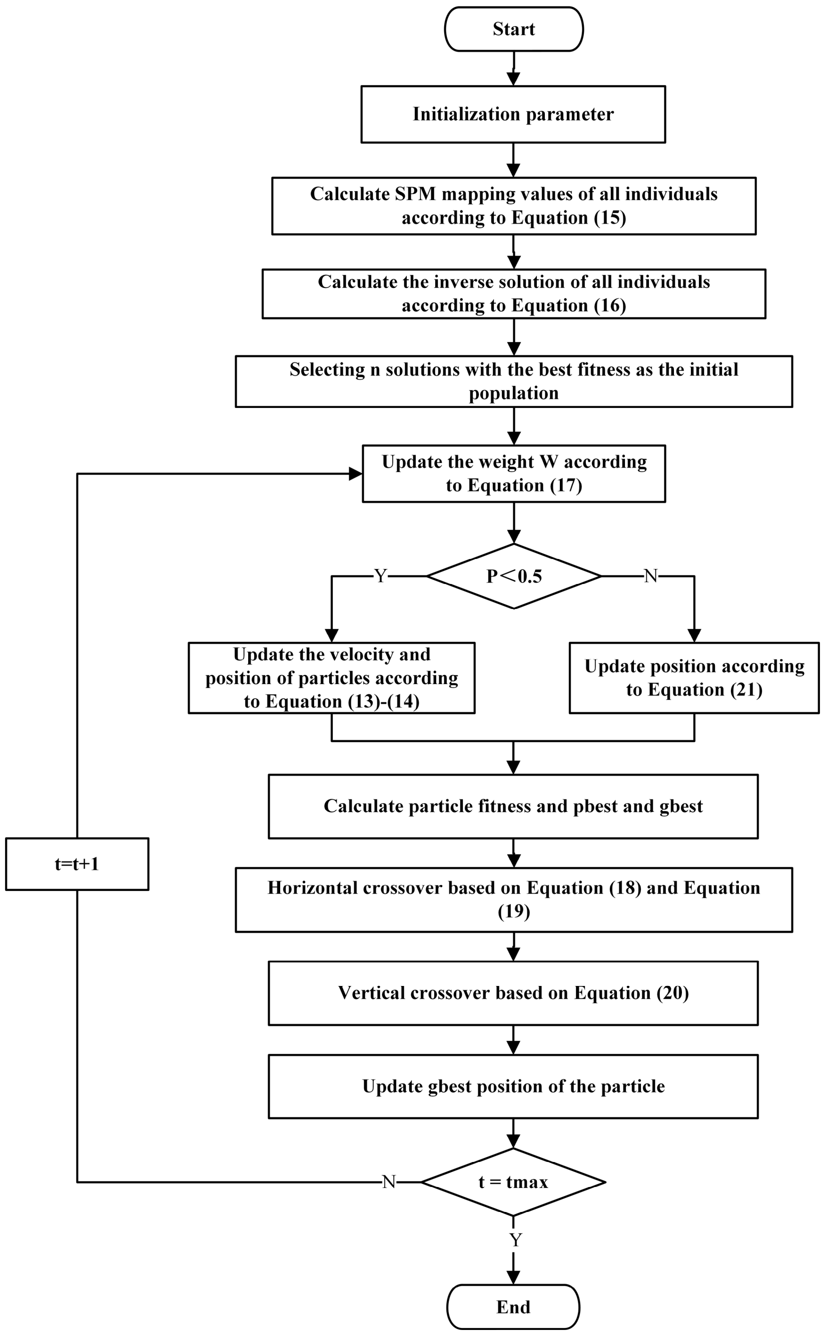
Start
Initialization parameter
Calculate SPM mapping values of all individuals according to Equation (15)
Calculate the inverse solution of all individuals according to Equation (16)
Selecting n solutions with the best fitness as the initial population
Update the weight W according to Equation (17)
P<0.5
Y
N
Update the velocity and position of particles according to Equation (13)-(14)
Update position according to Equation (21)
Calculate particle fitness and pbest and gbest
Horizontal crossover based on Equation (18) and Equation (19)
Vertical crossover based on Equation (20)
Update gbest position of the particle
t=t+1
t = tmax
N
Y
End

Figure 6. The workflow of IPSO.

6. Ablation Experiment

To assess the impact of four strategies on enhancing the performance of the particle swarm model, this paper compares the traditional PSO with an improved version, IPSO, which incorporates these strategies alongside the backpropagation neural network. Table 4 illustrates the enhanced PSO incorporating various strategies and relevant parameter settings. The experimental setup of this study is based on Intel Corporation's Core i7-12700H hardware processor and MATLAB 2020 software platform.

Table 4. Improved PSO with the introduction of different strategies.

Algorithm	Introduced Strategies	Parameter Setting
PSO	None	$C_1 = C_2 = 2$
IPSO1	Nonlinear weight Crossover strategy	$C_1 = C_2 = 2$
IPSO2	Nonlinear weight Crossover strategy Chaos mapping	$C_1 = C_2 = 2\eta = 0.4\mu = 0.5$
IPSO3	Nonlinear weight Crossover strategy Chaos mapping based on opposition-based learning	$C_1 = C_2 = 2\eta = 0.4\mu = 0.5$
IPSO4	Nonlinear weight Crossover strategy Chaos mapping based on opposition-based learning Spiral search strategy	$C_1 = C_2 = 2\eta = 0.4\mu = 0.5k = 5$

In the ablation experiment, the model iteration count was set to 30, the population size was 20, and both the individual learning factor C_1 and social learning factor C_2 were 2.

As can be seen from Figure 7, the introduction of the nonlinear weight and crossover strategy in IPSO1 enhances the algorithm's local optimization ability in the later stages. After introducing chaos mapping, the convergence speed of IPSO2 is improved in the early iteration stages, but its optimization ability has not improved much. By introducing chaos mapping based on opposition-based learning, IPSO3 can find a superior global optimum during the later stages of iteration. After introducing spiral search, IPSO4 can obtain a better set of optimal parameters faster than IPSO3, which proves the effectiveness of the proposed strategy.

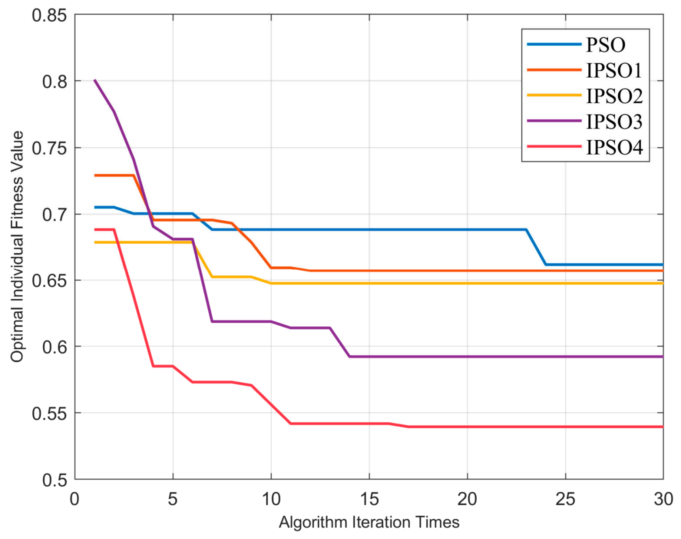

Figure 7. Comparison of iteration curves of PSO with different strategies added.

7. Test Function Experiment

To validate the performance of the improved algorithm, the DBO, SSA, PSO, and HO algorithms and the proposed IPSO in this paper were tested using single-peak and multi-peak test functions from the CEC2017 test suite. Parameter Settings for the DBO, SSA, PSO, and HO algorithms can be found in Appendix A.

The single-peak test functions F1, F3, and F4 were employed to assess the local exploration capabilities of the algorithms, while the multi-peak test functions F5–F10 served to evaluate their global search abilities. The mathematical formulas and boundaries for all functions are shown in Table 5.

Table 5. Standard test function table.

Function	Search Range
$F_1(x) = x_1^2 + 10^6 \sum_{i=2}^{D} x_i^2$	[−100,100]
$F_3(x) = \sum_{i=1}^{D} x_i^2 + \left(\sum_{i=1}^{D} 0.5x_i\right)^2 + \left(\sum_{i=1}^{D} 0.5x_i\right)^4$	[−100,100]
$F_4(x) = \sum_{i=1}^{D-1} \left(100\left(x_i^2 - x_{i+1}\right)^2 + (x_i - 1)^2\right)$	[−100,100]
$F_8(x) = \sum_{i=1}^{D} \left(z_i^2 - 10\cos\left(2\pi z_i\right) + 10\right) + F13^*$	[−100,100]
$F_9(x) = \sin^2(\pi w_1) + \sum_{i=1}^{D-1} (w_i - 1)^2 \left[1 + 10\sin^2(\pi w_i + 1)\right] + (w_D - 1)^2 \left[1 + \sin^2(\pi w_D)\right]$	[−100,100]
$F_{10}(x) = 418.9829 \times D - \sum_{i=1}^{D} g(z_i)$	[−100,100]

All algorithms underwent 500 iterations with a population size of 100. Specifically, the dimension of test function F1 was set to 30, and the dimensions of F3, F4, F8, F9, and F10 were set to 50. To ensure the reliability of the results, each algorithm was tested independently 30 times on each test function, yielding best, worst, and average values and standard deviations. The experimental results are summarized in Table 6.

The outcomes of the single-peak test function reveal that the IPSO algorithm exhibits superior optimization capabilities and convergence rates in comparison to three other benchmark algorithms. In the context of multimodal test functions, the IPSO algorithm stands out with exceptional optimization performance, particularly when confronted with complex, multi-dimensional functions. Furthermore, analysis of the standard deviation underscores the significantly higher stability of the IPSO function, attributable to its enhanced population diversity. The integration of spiral search and crossover strategies ensures comprehensive exploration of the global search space, thereby effectively mitigating the issue of severe homogeneity among individuals during late iterations. Overall, the IPSO algorithm introduced in this study has undergone substantial improvements in terms of precision and stability, with notable enhancements in convergence accuracy and stability, particularly for both unimodal and multimodal functions. To display the effect of optimization intuitively, this paper selects the optimization graphs of six functions, as shown in Figure 8.

Based on the iteration curves, it can be concluded that compared to the improved IPSO, the DBO, SSA, PSO and HO algorithms require more iterations to achieve the optimal value or to reach the same accuracy.

This is because, in the iterative process of IPSO, the introduction of multiple strategies not only increases the diversity of high-quality particles in the population but also effectively balances the global search and local search so that the particles can find better global optimal values and improve the convergence speed of the algorithm.

Table 6. Evaluation indicators of test function results.

Function	Result	IPSO	DBO	PSO	HO	SSA
F1	Best value	3.22×10^3	1.83×10^6	1.47×10^4	6.8×10^5	3.88×10^3
	Standard deviation	2.18×10^4	5.81×10^7	7.08×10^8	2.42×10^7	4.11×10^3
	Average value	2.08×10^4	4.83×10^7	3.17×10^8	1.44×10^7	9.87×10^3
	Worst value	5.79×10^4	1.41×10^8	1.58×10^9	5.73×10^7	1.5×10^4
F3	Best value	8.4×10^3	3.37×10^4	1.43×10^4	2.37×10^4	3.26×10^4
	Standard deviation	7.46×10^3	2.8×10^4	1.1×10^4	7.43×10^3	7.56×10^3
	Average value	2.16×10^4	9.49×10^4	3.16×10^4	4.14×10^4	5.25×10^4
	Worst value	4.28×10^4	1.65×10^5	6.46×10^4	5.47×10^4	6.8×10^4
F4	Best value	4.99×10^2	8.23×10^2	6.37×10^2	6.79×10^2	5.27×10^2
	Standard deviation	59.9	2.25×10^2	1.96×10^2	2.62×10^2	2.52×10^2
	Average value	5.9×10^2	1.21×10^3	9.1×10^2	1.06×10^3	7.75×10^2
	Worst value	7.06×10^2	1.77×10^3	1.45×10^3	1.75×10^3	1.59×10^3
F8	Best value	9.57×10^2	1.11×10^3	1.1×10^3	1.06×10^3	1.02×10^3
	Standard deviation	33.4	84.1	40.2	33	54.6
	Average value	1.03×10^3	1.24×10^3	1.20×10^3	1.14×10^3	1.17×10^3
	Worst value	1.08×10^3	1.41×10^3	1.29×10^3	1.2×10^3	1.28×10^3
F9	Best value	6.86×10^3	9.4×10^3	9.6×10^3	9.46×10^3	9.97×10^3
	Standard deviation	7.14×10^2	6.67×10^3	1.49×10^3	5.19×10^3	1.46×10^3
	Average value	7.99×10^3	1.92×10^4	1.36×10^4	1.78×10^4	1.36×10^4
	Worst value	8.62×10^3	3.84×10^4	1.80×10^4	2.87×10^4	1.76×10^4
F10	Best value	6.86×10^3	7.41×10^3	5.91×10^3	6.36×10^3	6.39×10^3
	Standard deviation	7.14×10^2	9.95×10^2	1.25×10^3	1.1×10^3	8.23×10^2
	Average value	7.99×10^3	9.22×10^3	8.15×10^3	8.36×10^3	8.31×10^3
	Worst value	8.62×10^3	1.16×10^4	1.12×10^4	1.1×10^4	1×10^4

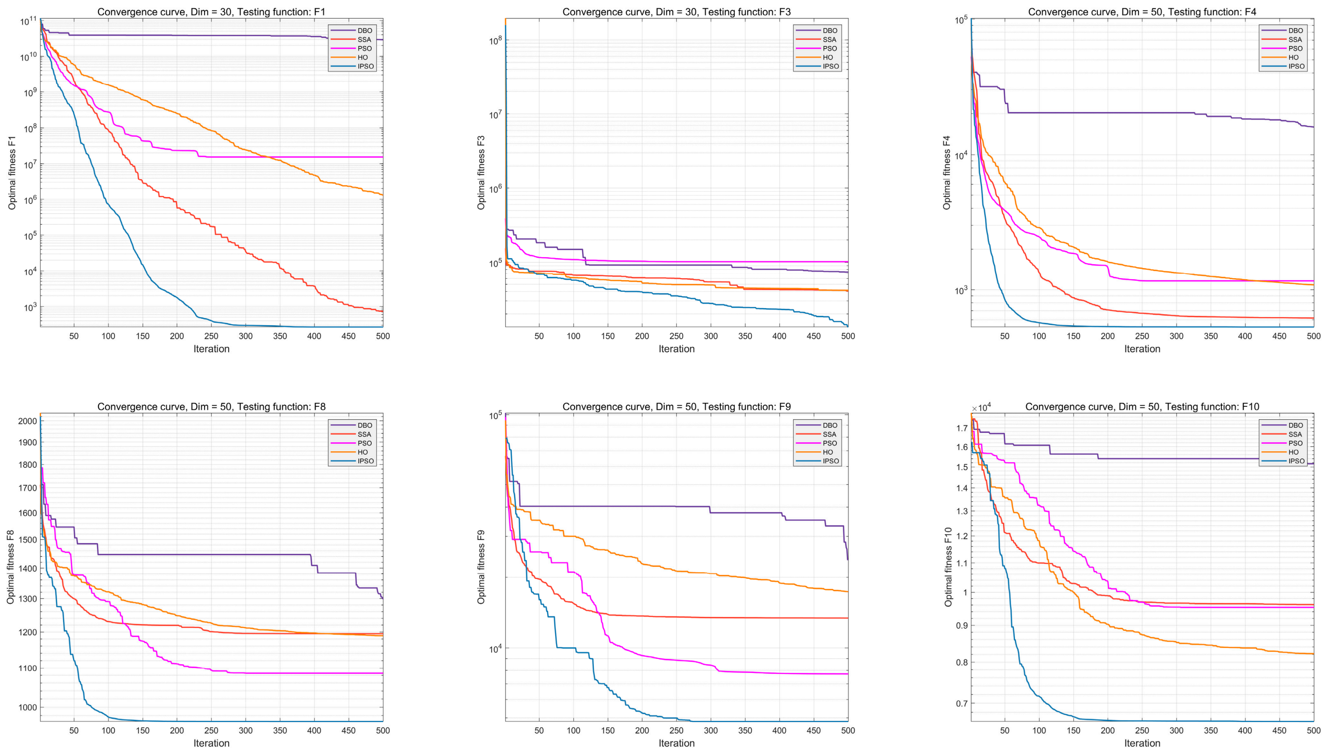

Figure 8. Iteration curves of five models on different test functions.

On the other hand, statistical analysis helps to confirm the significance of differences between the results obtained by different algorithms. This study uses a nonparametric statistical test called the Wilcoxon rank sum test. Statistical testing has an output parameter called a p-value, which determines the significance level of two algorithms. In this study, two algorithms are considered statistically different only if the p-value resulting from the Wilcoxon rank sum test is less than 0.05. Details are shown in Table 7.

Table 7. *p*-values obtained from test function.

Function	IPSO versus DBO	IPSO versus PSO	IPSO versus HO	IPSO versus SSA
F1	5.26×10^{-4}	1.95×10^{-3}	7.94×10^{-3}	9.52×10^{-2}
F3	2.61×10^{-10}	0.42	7.62×10^{-3}	1.68×10^{-3}
F4	7.77×10^{-9}	4.36×10^{-4}	3.34×10^{-11}	0.22
F8	3.67×10^{-3}	1.46×10^{-10}	1.17×10^{-3}	4.24×10^{-2}
F9	4.23×10^{-3}	3.64×10^{-2}	1.41×10^{-9}	3.01×10^{-4}
F10	3.55×10^{-1}	7.22×10^{-6}	4.55×10^{-1}	9.63×10^{-2}

It is observed that the *p*-value is less than 0.05 in most cases, which indicates that the optimization ability of IPSO is statistically superior compared with other algorithms.

8. Example of Transformer Fault Data Diagnosis

8.1. Transformer Fault Diagnosis Model Based on Improved IPSO–BP–AdaBoost

To verify the effectiveness of the model in practical transformer fault diagnosis, DGA samples collected from previously published literature and IEC TC10 databases were used to create a fault diagnosis model. Among them, 70% of the sample dataset (417 samples) was used as a training set, and the remaining 30% of the samples (179 samples) were used as a test set. The model operation steps are as follows:

Step 1: Initialize the relevant parameters of IPSO: population size, maximum iteration times, spatial dimensions, upper and lower bounds of the solution, and learning factors. Initialize AdaBoost-related parameters: weak classifier weight α and sample weight D.
Step 2: Use the SPM function to generate N solutions according to Equation (15).
Step 3: Use opposition-based learning to generate N opposite solutions according to Equation (16).
Step 4: Calculate the fitness of a total of 2N solutions and select the top N ranked solutions.
Step 5: Update the particle positions according to Equations (13) and (21).
Step 6: Determine whether the maximum iteration times have been reached. If so, output the global optimal individual position; otherwise, return to the loop.
Step 7: Calculate the weight of each sample according to Equation (7).
Step 8: Calculate the weight of the weak classifier according to Equation (6).
Step 9: Determine whether each weak classifier has been trained. If so, integrate them according to the weight of the weak classifier and output the result, which is the strong classifier result according to Equation (8). Otherwise, return to the loop.

The diagnostic model used in this paper is shown in Figure 9.

8.2. Diagnostic Accuracy of Different Models and Feature Combinations

Firstly, the experiment aims to determine the optimal number of neurons in the two hidden layers of the backpropagation neural network (BPNN). The initial configuration is set at 8 neurons, with a subsequent increase to 16. To mitigate randomness, the experiment was conducted over 30 iterations, and the average values were calculated. The results are presented in Figure 10.

It is evident that the BPNN achieves optimal diagnostic performance when the number of neurons is set at 12. However, it is noteworthy that without undergoing algorithmic optimization, the diagnostic accuracy remains relatively low. Secondly, the experiment compares the key parameters of the backpropagation neural network (BPNN) based on various optimization algorithms, including the Sparrow Search Algorithm (SSA), Particle Swarm Optimization (PSO), the Dung Beetle Optimization Algorithm (DBO), and the

Hippopotamus Optimization (HO) algorithm. The number of iterations for each optimization algorithm in the experiment is set to 30, the population size is 20, and the BPNN is configured with 8 input nodes, 2 hidden layers each containing 12 neurons, and 6 output nodes. The iteration curves of all models are shown in Figure 11.

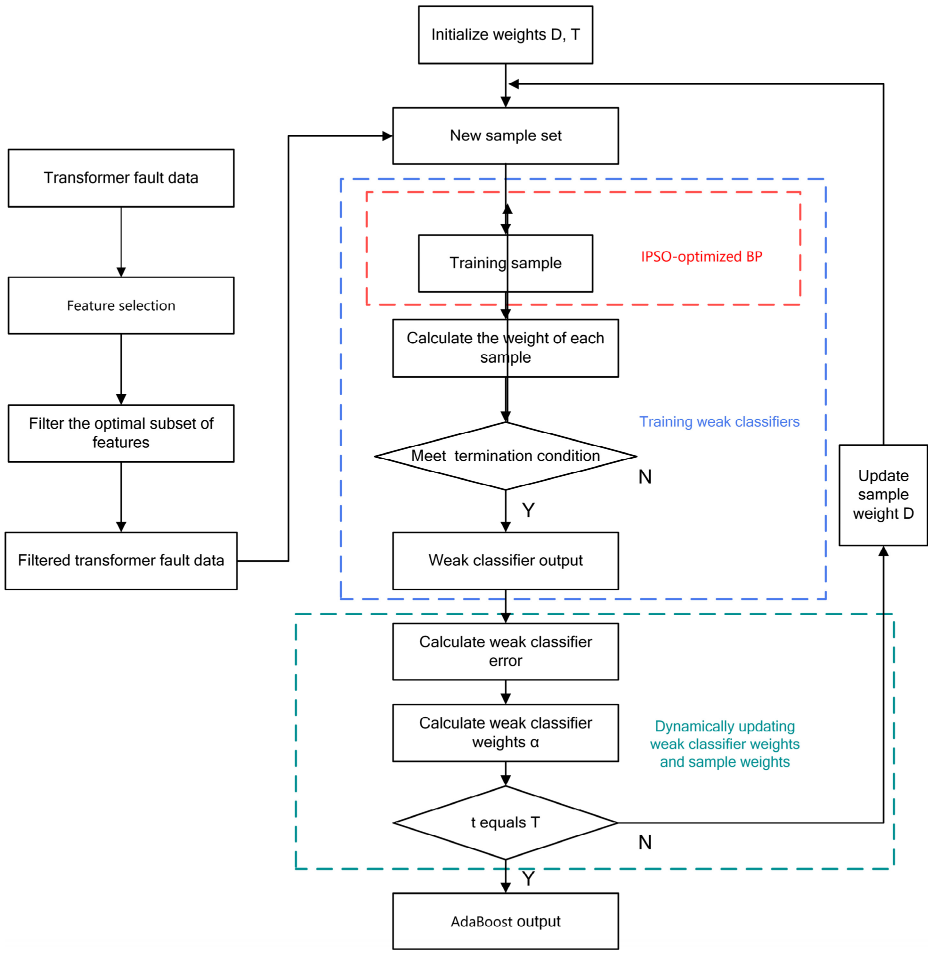
Initialize weights D, T
New sample set
Transformer fault data
Feature selection
Filter the optimal subset of features
Filtered transformer fault data
Training sample
IPSO-optimized BP
Calculate the weight of each sample
Training weak classifiers
Meet termination condition
N
Y
Update sample weight D
Weak classifier output
Calculate weak classifier error
Calculate weak classifier weights α
Dynamically updating weak classifier weights and sample weights
t equals T
N
Y
AdaBoost output

Figure 9. Transformer fault diagnosis model.

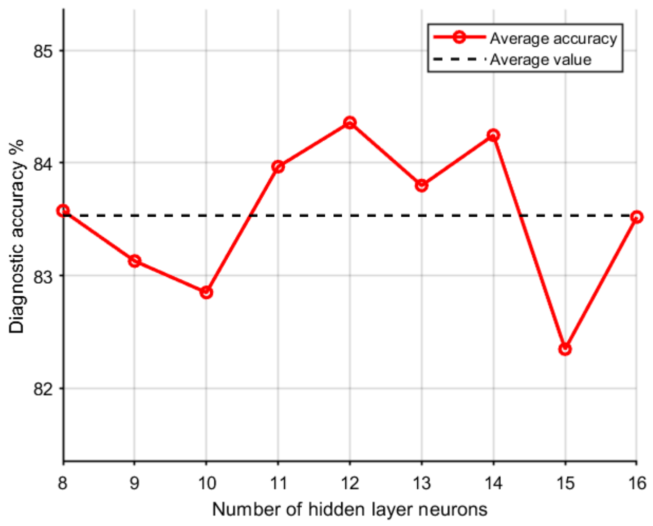

Figure 10. Determination of the number of hidden layer neurons.

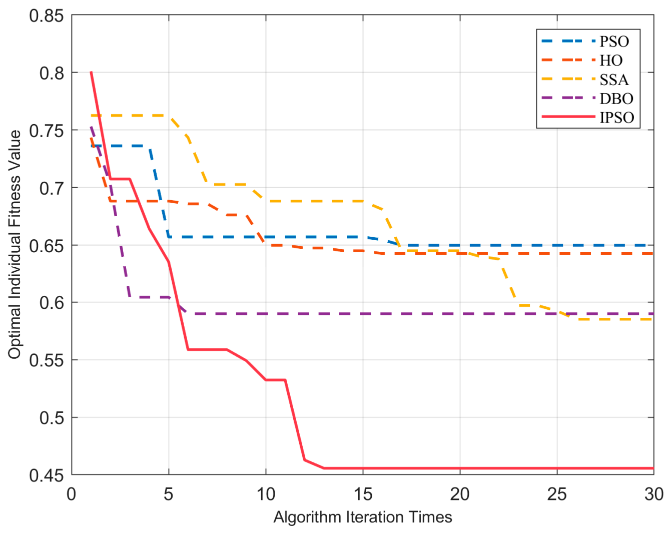

Figure 11. Fitness curves are based on different optimization algorithms.

The experimental results indicate that, compared to other optimization algorithms, the proposed IPSO algorithm in this paper possesses excellent global and local search capabilities, achieving the fastest convergence while obtaining the smallest fitness value. Some other algorithms lack local search capabilities in the later stages of iteration or require longer iteration times to complete optimization. It is noteworthy that the optimization ability of IPSO has significantly enhanced compared to PSO, further substantiating the effectiveness of incorporating the four proposed strategies.

To further validate the effectiveness of the feature selection and models used in this paper for practical transformer fault diagnosis, a comparative experiment was conducted between the Duval triangle method, the Rogers ratios method [31], the IEC standard code method, the clustering method, the ANN method, the conditional probability method, the modified Rogers ratios method, the modified IEC code method, and the IPSO–BP–AdaBoost method. The conditional probability method uses a Multivariate Normal Probability Density Function (MVNPDF). Each method was tested independently 30 times, and the average values were taken to minimize the randomness of single experimental results. The experimental results are summarized in Figure 12, where the modified Rogers ratio method and IEC code method are represented as *Rogers4 and *IEC60599, respectively. The closer the color of the bar chart is to yellow, the higher the average accuracy; conversely, the closer the color is to blue, the lower the average accuracy.

As seen in Figure 12, the accuracies of different methods for fault diagnosis are 64.34% for Duval triangles, 47.87% for Rogers' 4-ratios, 56.19% for IEC, 77.07% for clustering, 79.96% for the ANN technique, 74.02% for the conditional probability method, 68.93% for refining IEC, and 62.98% for refining Rogers' 4-ratios. The IPSO–BP–AdaBoost used in this paper has the highest diagnostic accuracy (89.6%), which proves the effectiveness of the method.

To verify the superiority of feature selection, PSO, DBP, SSA, HO, and IPSO algorithms are used to optimize BP–AdaBoost, and the accuracy of 26 input features and 8 input optimal features is compared. All results are averaged after 30 experiments, as shown in Figure 13.

It is obvious that the diagnostic accuracy of all models is improved after the use of the best feature set. Meanwhile, IPSO–BP–AdaBoost, proposed in this paper, has the highest accuracy regardless of the input of all feature sets or the input of the best feature set. Figure 14 shows the best performance of each model in 30 tests.

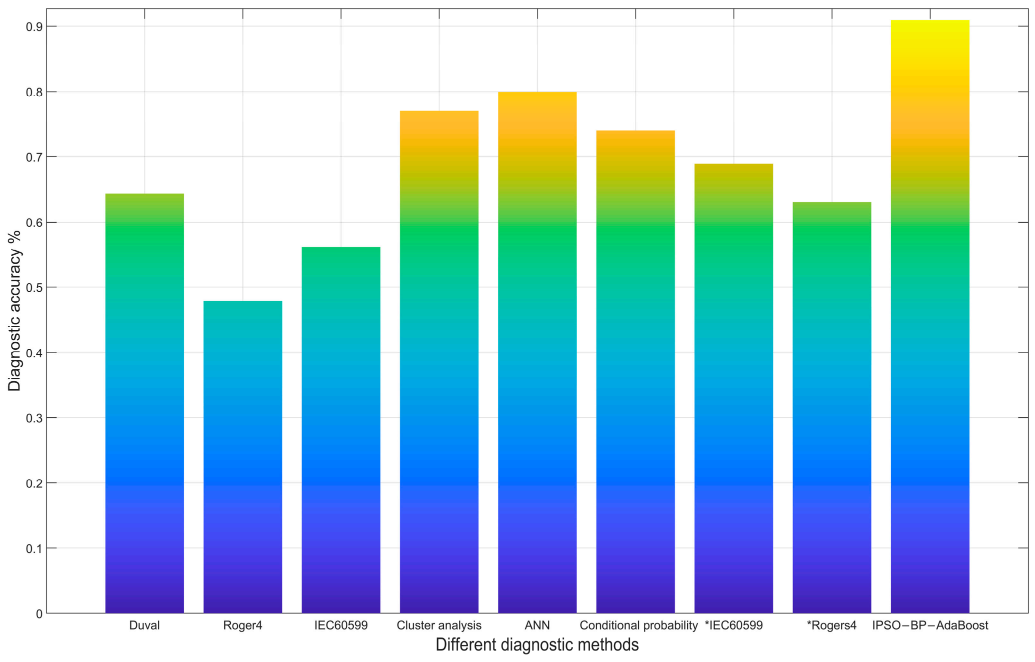

Figure 12. Diagnostic results of different methods.

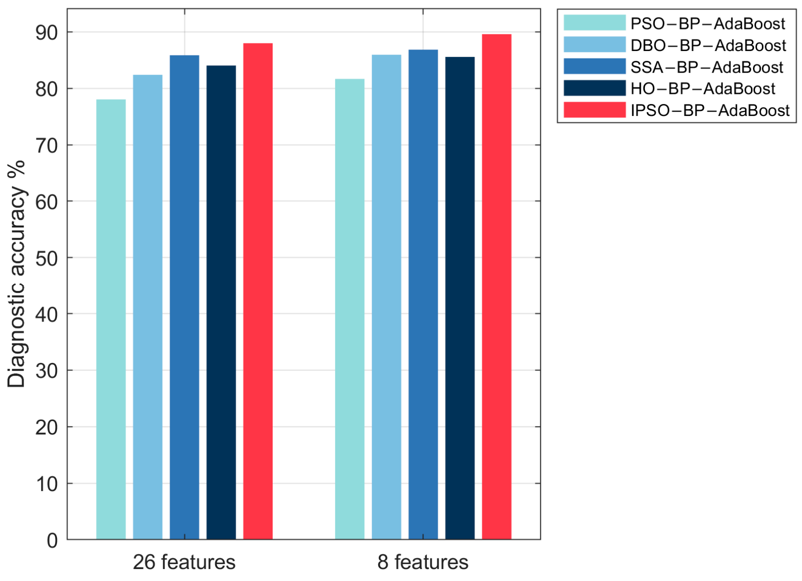

Figure 13. Average diagnostic accuracies of different feature sets.

As illustrated in Figure 14, the IPSO demonstrates the highest accuracy in classifying various fault categories. This not only validates the effectiveness of the improved strategy proposed in this paper when compared to traditional PSO but also highlights its advantages over other intelligent algorithms.

In addition to the average diagnostic accuracy, this paper summarizes the kappa coefficient, F1-scores, precisions, and recalls of various models. The kappa coefficient is a measure that assesses the agreement between the predictions of a classification model and the actual results. It effectively solves the problem of data imbalance and random prediction. Precision represents the proportion of predicted positive instances that are actually positive, while recall represents the proportion of actual positive instances that the model correctly identifies. The F1-score is a measure that combines precision and recall. A higher F1-score means better model performance, making it more reliable than accuracy in scenarios involving unbalanced sample classes, as shown in Table 8.

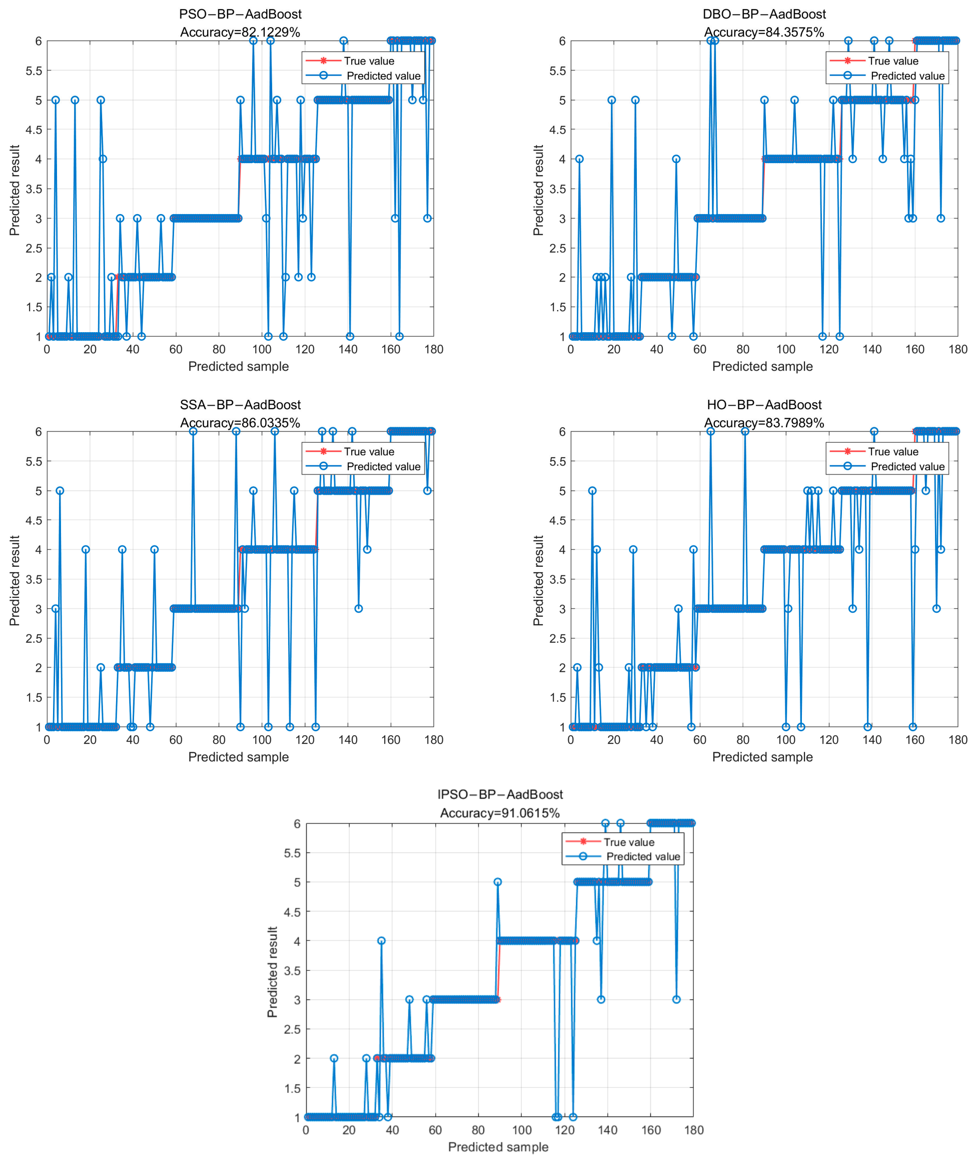

Figure 14. The best diagnostic accuracies of five models.

Table 8. Comparative evaluation of model diagnostic performance.

Models	Kappa Coefficient	F1-Score	Recall	Precision
PSO–BP–AdaBoost	0.7723	0.8090	0.8111	0.8068
DBO–BP–AdaBoost	0.8126	0.8398	0.8379	0.8418
SSA–BP–AdaBoost	0.8327	0.8617	0.8614	0.8620
HO–BP–AdaBoost	0.8326	0.8589	0.8599	0.8579
IPSO–BP–AdaBoost	0.8928	0.9145	0.9166	0.9124

It can be concluded from Table 8 that among the five models, IPSO–BP–AdaBoost shows the best performance, which is superior to other models in classification accuracy (Kappa coefficient), comprehensive performance (F1-score), ability to find all positive samples (recall), and prediction accuracy (precision). SSA–BP–AdaBoost and HO–BP–AdaBoost also showed better performance but were slightly inferior to IPSO–BP–AdaBoost in various indexes. DBO–BP–AdaBoost and PSO–BP–AdaBoost, on the other hand, display a degree of effectiveness, yet their performance is comparatively less robust when compared to the other three models, particularly IPSO–BP–AdaBoost.

This article further compares four deep learning models—DBNs (Deep Belief Networks), CNNs (Convolutional Neural Networks), ELMs (Extreme Learning Machine), and BiGRUs (Bidirectional Gated Recurrent Units)—with IPSO–BP–AdaBoost. All models utilized the dataset that had undergone feature selection, and each model was run 30 times to obtain the average accuracy, as presented in Table 9.

Table 9. Comparison results with four deep learning models.

Models	Highest Accuracy Rate (%)	Minimum Accuracy Rate (%)	Average Accuracy Rate (%)
DBN	80.00%	50.83%	64.27%
CNN	86.03%	79.88%	82.64%
ELM	89.38%	80.44%	85.27%
BiGRU	85.47%	81.00%	83.89%
IPSO–BP–AdaBoost	91.06%	87.68%	89.60%

From Table 9, DBN has the lowest average accuracy and the largest difference between the highest and lowest accuracy, indicating poor stability of the model. The average accuracy of CNN and ELM is higher than that of DBN, but the robustness of both models needs improvement. Although BiGRU exhibits good robustness, its average accuracy is relatively low. The IPSO–BP–AdaBoost proposed in this paper can maintain high accuracy while ensuring minimal fluctuation in each running result. This indicates that the model has good robustness and consistency and can provide reliable diagnostic results.

9. Conclusions

In this article, the IPSO-BP AdaBoost algorithm, integrated with advanced feature selection techniques, is employed to diagnose faults in power transformers. The feature selection method utilizes Random Forests to construct more compact, precise, and informative feature subsets. The optimally selected features include C_2H_2/C_2H_4, C_2H_2/TH, TH, $(H_2 + C_2H_4)/TG$, H_2, C_2H_6/TH, C_2H_4/TH, and C_2H_2. The IPSO algorithm, augmented with multiple strategic enhancements, is utilized to optimize the initial weights and biases of the BPNN. The innovative application of chaos mapping and opposition-based learning methods for generating initial solutions has proven to be advantageous for parameter optimization and mitigates the initialization sensitivity issue of the BPNN.

DGA samples sourced from domestic transformer data and the IEC-TC10 database were employed to validate the efficacy of the optimal feature subset. Comparative analysis of fault diagnosis performance between the optimal feature subset and other gas ratio methods revealed that the proposed feature subset delivers superior diagnostic performance with an accuracy of 91.06%. This substantiates the advantages and effectiveness of the proposed methodology. Additionally, comparisons with other models underscore the high accuracy and robustness of the IPSO–BP–AdaBoost diagnostic model. However, due to its high precision, it requires a longer training time. Consequently, future research will focus on optimizing the model further to reduce training duration while simultaneously enhancing accuracy. On the other hand, in future research, in addition to the gas characteristics in the oil, it is necessary to further study the relationship between other physical parameters and transformer faults, to comprehensively judge transformer faults from multiple angles.

Author Contributions: Conceptualization, Z.F.; methodology, L.Z.; validation, L.Z.; formal analysis, L.Z.; data curation, K.L., H.R., and Y.W.; writing—original draft preparation, L.Z.; writing—review and editing, Z.F., K.L., Y.W., and H.R.; visualization, L.Z.; supervision, Z.F.; funding acquisition, Z.F. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The original contributions presented in the study are included in the article, further inquiries can be directed to the corresponding author.

Acknowledgments: We thank Ayman HobAllah and Ibrahim B. M. Taha for professional guidance.

Conflicts of Interest: The authors declare no conflicts of interest.

Appendix A

Table A1. Parameter description of other algorithms in this paper.

Algorithm	Parameter Settings
PSO	Cognitive and social constants: 2 and 2 Inertia weight: linear reduction from 0.9 to 0.1
DBO	Probability of encounter: 0.1 Deflection coefficient: 0.1 Natural coefficient: 1 or −1 Constant: 0.3
SSA	Leader position update probability: 0.5 Proportion of the number of leaders: 0.2

References

1. Faiz, J.; Soleimani, M. Dissolved gas analysis evaluation in electric power transformers using conventional methods a review. *IEEE Trans. Dielectr. Electr. Insul.* **2017**, *24*, 1239–1248. [CrossRef]
2. Song, X. Research on D-S Evidence Theory and Its Application in Transformer Fault Diagnosis. Master's Thesis, Huaibei Normal University, Huaibei, China, 2023. [CrossRef]
3. Liu, Y.; Ni, Y.P. Transformer fault diagnosis method based on grey correlation analysis of three ratios. *High-Volt. Technol.* **2002**, *10*, 16–17,27. [CrossRef]
4. Jin, Y.; Wu, H.; Zheng, J.; Zhang, J.; Liu, Z. Power Transformer Fault Diagnosis Based on Improved BP Neural Network. *Electronics* **2023**, *12*, 3526. [CrossRef]
5. Ivanov, V.K.; Palyukh, B.V. Application of Evidence Theory for Training Fuzzy Neural Networks in Diagnostic Systems. *Pattern Recognit. Image Anal.* **2023**, *33*, 354–359. [CrossRef]
6. Kari, T.; He, Z.; Rouzi, A.; Zhang, Z.; Ma, X.; Du, L. Power transformer fault diagnosis using random forest and optimized kernel extreme learning machine. *Intell. Autom. Soft Comput.* **2023**, *37*, 691–705. [CrossRef]
7. Yu, S.; Tan, W.; Zhang, C.; Tang, C.; Cai, L.; Hu, D. Power transformer's fault diagnosis based on a meta-learning approach to kernel extreme learning machine with opposition-based learning sparrow search algorithm. *J. Intell. Fuzzy Syst.* **2023**, *44*, 455–466. [CrossRef]
8. Bai, X.; Zang, Y.; Ge, L.; Li, C.; Li, J.; Yuan, X. Selection Method of Feature Derived from Dissolved Gas in Oil for Fault Diagnosis of Transformers. *High Volt. Eng.* **2022**, *48*, 1–15.
9. Zhang, G.; Chen, K.; Fang, R.; Wang, K.; Zhang, X. Transformer fault diagnosis based on DGA and a whale algorithm optimizing a LogitBoost-decision tree. *Power Syst. Prot. Control* **2023**, *51*, 63–72. [CrossRef]
10. Cao, H.; Zhou, C.; Meng, Y.; Shen, J.; Xie, X. Advancement in transformer fault diagnosis technology. *Front. Energy Res.* **2024**, *12*, 1437614. [CrossRef]
11. *JWG D1/A2.47*; Advances in DGA Interpretation. CIGRE: Paris, France, 2019.
12. Dai, J.; Song, H.; Yang, Y.; Chen, Y.; Sheng, G.; Jiang, X. ReLU-DBN method for transformer fault diagnosis based on gas analysis in oil. *Power Grid Technol.* **2018**, *42*, 658–664. [CrossRef]
13. Sun, C. Transformer Fault Diagnosis Based on Machine Learning Algorithms. Master's Thesis, Shanghai Jiao Tong University, Shanghai, China, 2019. [CrossRef]
14. Hoballah, A.; Mansour, D.-E.A.; Taha, I.B.M. Hybrid Grey Wolf Optimizer for Transformer Fault Diagnosis Using Dissolved Gases Considering Uncertainty in Measurements. *IEEE Access* **2020**, *8*, 139176–139187. [CrossRef]
15. Wang, Z. Research on Feature Selection Methods based on Random Forest. *Teh. Vjesn.* **2023**, *30*, 623–633.
16. Yu, G.; Zhao, Y.; Fu, Z.; Chen, Z. Application of back propagation neural network in the analysis of isothermal elastohydrodynamic lubrication. *Tribol. Int.* **2024**, *198*, 109883. [CrossRef]
17. Huang, S.; Zhang, J.; He, Y.; Fu, X.; Fan, L.; Yao, G.; Wen, Y. Short-Term Load Forecasting Based on the CEEMDAN-Sample Entropy-BPNN-Transformer. *Energies* **2022**, *15*, 3659. [CrossRef]
18. Zuo, W.; He, Z.; Yang, Y. Transformer health prediction based on digital twins and SSA-BP. *Comput. Digit. Eng.* **2023**, *51*, 2457–2463.
19. Zhang, Z.; Kong, W.; Li, L.; Zhao, H.; Xin, C. Prediction of Transformer Oil Temperature Based on an Improved PSO Neural Network Algorithm. *Recent Adv. Electr. Electron. Eng.* **2024**, *17*, e270423216280. [CrossRef]

20. Li, J.; Li, G.; Hai, C.; Guo, M. Transformer Fault Diagnosis Based on Multi-Class AdaBoost Algorithm. *IEEE Access* **2022**, *10*, 1522–1532. [CrossRef]
21. Wang, F.; Yuan, G.; Guo, C.; Li, Z. Research on fault diagnosis method of aviation cable based on improved AdaBoost. *Adv. Mech. Eng.* **2022**, *14*, 16878132221125762. [CrossRef]
22. Zhang, Y.; Qu, J.; Fang, X.; Luo, G. Motor bearing fault diagnosis based on multi-feature fusion and PSO-BP. In Proceedings of the 2021 IEEE 4th Student Conference on Electric Machines and Systems (SCEMS), Huzhou, China, 1–3 December 2021; pp. 1–5. [CrossRef]
23. Lu, W.; Shi, C.; Fu, H.; Xu, Y. A Power Transformer Fault Diagnosis Method Based on Improved Sand Cat Swarm Optimization Algorithm and Bidirectional Gated Recurrent Unit. *Electronics* **2023**, *12*, 672. [CrossRef]
24. Lou, L.; Zhang, H. Grey Wolf Optimization algorithm based on Hybrid Multi-strategy. In Proceedings of the 2023 8th International Conference on Intelligent Computing and Signal Processing (ICSP), Xi'an, China, 21–23 April 2023; pp. 1342–1345. [CrossRef]
25. Hu, Y.; Xiong, R.; Li, J.; Zhou, C.; Wu, Q. An Improved Sand Cat Swarm Operation and Its Application in Engineering. *IEEE Access* **2023**, *11*, 68664–68681. [CrossRef]
26. Tizhoosh, H.R. Opposition-Based Learning: A New Scheme for Machine Intelligence. In Proceedings of the International Conference on Computational Intelligence for Modelling, Control and Automation and International Conference on Intelligent Agents, Web Technologies and Internet Commerce (CIMCA-IAWTIC'06), Vienna, Austria, 28–30 November 2005; pp. 695–701. [CrossRef]
27. Nazari, M.; Esnaashari, M.; Parvizimosaed, M.; Damia, A. A Noval Reduced Particle Swarm Optimization with Improved Learning Strategy and Crossover Operator. In Proceedings of the 2023 28th International Computer Conference, Computer Society of Iran (CSICC), Tehran, Iran, 25–26 January 2023; pp. 1–5. [CrossRef]
28. Cao, J.; Lu, M. Industrial water consumption prediction based on hybrid strategy improved SSA-SVM. *Hydroelectr. Energy Sci.* **2023**, *41*, 28–31. [CrossRef]
29. Zhang, D.; Zhao, Y.; Ding, J.; Wang, Z.; Xu, J. Multi-Strategy Fusion Improved Adaptive Hunger Games Search. *IEEE Access* **2023**, *11*, 67400–67410. [CrossRef]
30. Gao, J.; Xing, Q.; Li, L.; Fan, C. Improved Particle Swarm Optimization Algorithm Using Projection Spiral Search. *J. Xi'an Jiaotong Univ.* **2018**, *52*, 48–54.
31. Ibrahim, S.I.; Ghoneim, S.S.M.; Taha, I.B.M. DGALab: An extensible software implementation for DGA. IET Gener. *Transm. Distrib.* **2018**, *12*, 4117–4124. [CrossRef]