Hindawi
International Transactions on Electrical Energy Systems
Volume 2023, Article ID 8812152, 12 pages
https://doi.org/10.1155/2023/8812152

Research Article

Application of WCA-RBF Neural Network in Fault Diagnosis of Analog Circuits

Zhiwei Jin

College of Computer and Artificial Intelligence, Henan Finance University, Zhengzhou 450046, China

Correspondence should be addressed to Zhiwei Jin; jinzhiwei@hafu.edu.cn

Received 27 June 2023; Revised 13 November 2023; Accepted 16 November 2023; Published 30 November 2023

Academic Editor: Santi A. Rizzo

With the fast growth of electronic technology today, the hybrid circuit of analog and digital circuits has become a trend in the growth of electronic technology. To address fault diagnosis in analog circuits, Haar wavelet is applied for fault feature extraction. The K-means clustering method and pseudoinverse algorithm were used to optimize the center value and weight value of the radial basis function neural network, respectively. The adaptive step size was improved in the wolf pack algorithm, the parameters of the radial basis function neural network were optimized based on the wolf pack algorithm, and a wolf pack algorithm optimized radial basis function neural network model was constructed. The test results show that this model converges after 40 times of training, with an error value of 10^{-3} and an average value of the mean squared error of 0.45. Comparing the fault diagnosis rates of the original model, genetic algorithm optimized radial basis function neural network model, and wolf colony algorithm optimized radial basis function neural network model, the last model has the best fault diagnosis rate, reaching 95.52%. The wolf colony algorithm is utilized to optimize the radial basis function neural network model to diagnose the faults in the standard filter circuit, and the fault diagnosis rate reaches 96.17%. The findings express that the radial basis function neural network model optimized by the wolf colony algorithm has a good diagnosis effect for different faults in the analog circuit.

1. Introduction

Analog circuits (ACs) are circuit systems composed of various electronic components that can generate simple or complex output signals for continuous input signals. ACs have a crucial impact on modern electronic technology and can be utilized in various fields [1]. In the design and manufacturing of ACs, due to open or short circuits in electronic components, the signal in the circuit is interrupted or the current is too high, resulting in open or short circuit faults. It is also possible that the capacitor may malfunction, resulting in abnormal transmission or processing of signals in the circuit [2]. Therefore, locating and diagnosing faults in ACs while ensuring their integrity is an indispensable part of AC design and manufacturing. To solve fault diagnosis (FD) in ACs, this research uses Haar wavelet to extract fault features, improves the adaptive step size in the wolf colony algorithm, and optimizes the radial basis function (RBF) based on the wolf colony algorithm (WCA). The neural network model of the wolf colony algorithm is constructed to optimize the RBF.

This study contains four parts. The first part is the research on the diagnosis methods of circuit faults and the application of RBF neural network by scholars all over the world. In the second part, fault features are extracted by using the accurate characteristics of signal analysis of the Haar wavelet. The K-clustering algorithm and pseudoinverse algorithm is introduced to optimize the center and weight values of radial basis function neural networks. Based on WCA, the RBF is further improved to build the AC diagnosis model. The third part is the performance test of wolf colony optimization RBF and its utilization in an example. The fourth part summarizes the article and proposes shortcomings.

2. Related Works

FD of ACs is the monitoring and locating of possible faults in ACs. FD plays a crucial role in AC design and manufacturing, helping engineers to quickly and accurately detect and repair faults. Some scholars have conducted relevant research on FD methods applied to circuit faults. Moezi and Karga [3] proposed a multiobjective optimization fault classification method based on Hellinger distance, which extracted fault features using intrinsic modulus functions and preserved useful features using nondominated sorting genetic algorithms. Moreover, support vector machines and neural networks were used for fault identification. The effectiveness of this method has been demonstrated through experiments. Kou et al. [4] analyzed the problem of open circuit FD and location in neutral point circuits and proposed an FD method. It improved the robustness of FD classifiers by using knowledge-driven transformations and data-driven techniques. It analyzed and extracted AC fault feature data of inverters, transformed the data to process fault samples, and validated that the slope of the current trajectory was not influenced by different loads. Through experiments, it has been proven that this method has strong predictive ability when used in circuit FD. Huang et al. [5] put forward a circuit FD method with model predictive control, which mainly included fault detection and location. This model could not only monitor the condition of the driving system but also distinguish the types of faults. Therefore, faults could be accurately monitored and located. The performance and robustness of FD were evidenced through examples. Based on the converter circuit switch FD method, Zhou et al. [6] selected the inductance current in the main controller system as the diagnostic variable and introduced an adaptive threshold to handle the uncertainty of converter circuit parameters and the impact of changes in converter operating points. This method avoided the use of additional sensors and had good robustness against false alarms. Through simulation experiments and analysis, this method had stability and effectiveness for FD in circuits.

RBF has better learning speed and generalization performance than traditional feedforward neural network, so it is often used in complex real-time applications. Some scholars have studied the application of RBF. Sun et al. [7] raised an amplitude saturation controller. Based on RBF, the amplitude saturation controller was improved, and a supervisory controller based on RBF was raised. The proposed control method's effectiveness and robustness in time delay were verified through simulation experiments. Lu et al. [8] proposed a teaching quality assessment method with genetic algorithm optimized RBF (GA-RBF) to solve the low quality of English interpretation teaching at present. Principal component analysis was applied to choose teaching quality assessment metrics, and GA was utilized to improve the initial weights of the model. It was proved that this method could well assess the quality of interpretation teaching, with perfect accuracy and real-time effectiveness. Based on multilayer perceptron and RBF, Hashem Fath et al. [9] raised an intelligent reliable model to estimate the solution gasoline ratio. The model was tested using crude oil samples from around the world, and the results showed that the predicted values of the samples were consistent with the true values, demonstrating strong consistency. Comparing the improved model with the traditional model, the model exhibited higher accuracy and efficiency. Yao et al. [10] proposed an adaptive control strategy. The RBF was used to adaptively adjust the virtual inertia value, improve the response of the model, and reduce the frequency overshoot. The damping coefficient was adaptively adjusted with changes in inertia. The research outcomes indicated that this method had good vibration reduction performance. Based on the research status of gesture recognition of EMG signals, Yu et al. [11] used the PSO (particle swarm optimization) algorithm to optimize RBF's center and width values and conducted noise reduction processing and segmentation detection on the basis of summation. Simulation experiments have shown that the model achieved a high recognition rate in gesture recognition. Sohrabi et al. [12] used two time series and the RBF method to estimate the coal price. The time series method used Monte Carlo simulation and MATLAB software to estimate coal prices. Collecting daily coal price data from 2018 to 2020 and using this model for testing, the experimental results achieved high accuracy in estimating coal prices.

In summary, in circuit fault identification, there are different diagnostic methods for different faults. This study utilizes the Haar wavelet to extract fault features. Also, introduce the K-clustering algorithm and pseudoinverse algorithm to optimize the center value and weight value of the radial basis function neural network. Based on the wolf pack algorithm, further improve the radial basis function neural network and construct an analog circuit diagnosis model. Therefore, the fault identification method based on RBF plays an important role in AC fault identification.

3. Fault Feature Extraction and FD Model Construction

ACs are broadly utilized in production and daily life, and the increasing complexity of circuit systems has led to improvements in circuit FD methods. This section is composed of three parts for research. The first part utilizes the accurate characteristics of the Haar wavelet in signal analysis to extract fault features. The second part optimizes the RBF from different angles. In the third part, based on the WCA, the RBF is further improved to build an AC diagnosis model.

3.1. Fault Feature Extraction of Haar Wavelet. Before extracting fault features, it is necessary to detect and recognize faults. Random resonance is a nonlinear signal processing method that can effectively identify faults. The principle of stochastic resonance is to use a nonlinear oscillation system to process the input signal and adjust the system parameters to match its frequency with the fault component frequency of the signal. When the system frequency is close to the frequency of the fault component, the amplitude will significantly increase, making the fault component prominent. Haar wavelet is the simplest wavelet

function with the characteristics of fast computation and high efficiency. It can provide accurate local information in time and frequency and is often used in the fields of signal compression and edge detection. Haar wavelet belongs to a type of wavelet transform and is widely used in FD research. Based on iterative operation and fast Fourier transform, the Haar wavelet can perform multiscale analysis on signals, capture signal characteristics at different frequencies and scales, and has strong multiscale analysis ability [13]. When a circuit malfunctions, compared to other wavelet functions, the Haar wavelet can extract high-frequency and low-frequency information in analog circuits faster, reduce the input dimension of the neural network, and simplify the structure of the neural network. In the Haar wavelet transform, signals can be decomposed into two parts: approximate functions and detail coefficients. By decomposing the signal multiple times, approximate and detailed information at different scales can be obtained until the set scale is reached or no further decomposition can be performed [14]. The Haar wavelet satisfies the two scale equation, as displayed in the following equation:

$$\omega(x) = \sum_{k \in Z} c_k \omega(2x - k), \tag{1}$$

where c_k means the wavelet coefficient; $\omega(x)$ represents the scaling function; k denotes the displacement parameter; x indicates the independent variable of the wavelet function. The Haar wavelet equation is shown in the following equation:

$$\phi(x) = \omega(2x) - \omega(2x - 1), \tag{2}$$

where $\phi(x)$ denotes wavelet, and when the wavelet coefficients c_k are equal to −1 and 1, the wavelet is expressed as Haar wavelet. The waveform and scaling equation of the Haar wavelet are shown in Figure 1.

By using the Haar wavelet transform, the original signal is decomposed into subsignals of different scales and frequencies, which can effectively process some transient signals [15]. When processing the signal, a step function is recommended to approximate the signal, and the original signal expression is shown in the following equation:

$$f_j(x) = \sum_{k \in Z} c_k^j \phi\left(2^j x - k\right) \in V_j, \tag{3}$$

where c_k^j means the wavelet coefficient; k denotes the displacement parameter; x expresses the independent variable of the Haar wavelet function; V_j indicates the step function. It decomposes f_j using the decomposition expression shown in the following equation:

$$f_j = f_{j-1} + w_{j-1}, \tag{4}$$

where f_j refers to the initial signal; w_{j-1} stands for the approximate signal; w_{j-1} represents different components.

The signal is discretized, and the approximate signal is decomposed. The expression for further decomposition is shown in the following equation [16]:

$$f_j = w_{j-1} + w_{j-2} + w_{j-3} + \cdots + w_0 + f_0, \tag{5}$$

where $w_{j-1} + w_{j-2} + w_{j-3} + \cdots + w_0$ denotes subsignals of different frequencies and f_0 expresses an accurate approximation.

When processing the signal, removing the $w_{j-1} + w_{j-2} + w_{j-3} + \cdots + w_0$ in equation (5) can achieve denoising of the signal. By removing the smaller values in $w_{j-1} + w_{j-2} + w_{j-3} + \cdots + w_0$ and retaining the larger absolute values, data compression processing can be achieved. When an AC malfunctions, the extracted signals may differ between high- and low-frequency signals. Haar wavelet can transform the coefficients of high- and low-frequency parts to obtain new energy value combination features. The Haar wavelet decomposition diagram is shown in Figure 2.

When a circuit malfunctions, the high- and low-frequency parts will be different. The fault features extracted by the Haar wavelet are composed of energy values converted by coefficients at different frequencies. First, it needs to sample and extract the fault signal and perform wavelet decomposition to obtain high- and low-frequency coefficient sequences. Then, three-layer wavelet decomposition is performed on the sampling sequence number, as shown in Figure 2. When an AC malfunctions, the extracted signals may differ between high- and low-frequency signals. Haar wavelet can transform the coefficients of high- and low-frequency parts to obtain new energy value combination features [7]. The energy expression is shown in the following equation:

$$0.1 + \frac{(E - \min(E))}{(\max(E) - \min(E)) * (0.9 - 0.1)}, \tag{6}$$

where $\max(E)$ refers to the maximum energy value and $\min(E)$ stands for the minimum energy value.

By decomposing the Haar wavelet, the algorithm can achieve simple operations and is suitable for real-time processing and embedded system scenarios, improving the calculation speed of the algorithm. Moreover, wavelet decomposition provides a special multiscale analysis framework, enabling efficient feature signal extraction and noise removal [17].

3.2. Construction of RBF Neural Network Optimization Model. RBF is a feedforward neural network. It is applied in function approximation, classification, and data mining by using RBF as an activation function [7]. The structure of RBF is composed of three layers. The input layer receives the feature vectors of the samples as the input values of the model. The hidden layer performs nonlinear processing on the features of the input layer, and the output layer can provide target values for output based on the output values of the hidden layer [18]. The structural diagram of the RBF is shown in Figure 3.

In Figure 3, the input and output values can be represented by an input vector and an output vector, respectively. The expression of the k-th output value in the model is shown in the following equation:

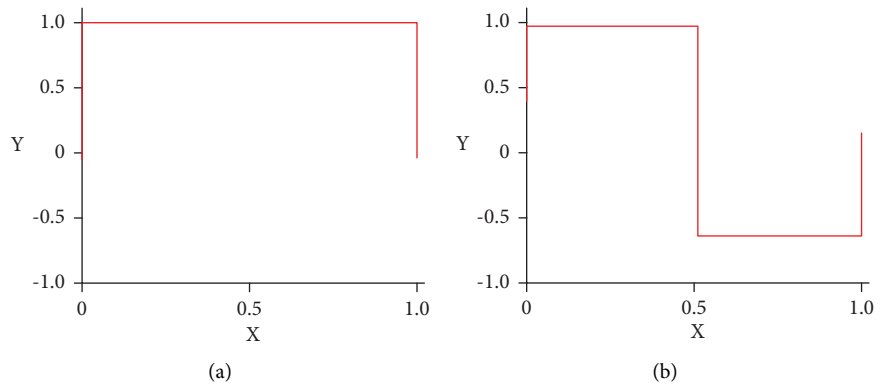

Figure 1: Waveform diagram and scaling function diagram of Haar wavelet function. (a) Haar scaling function. (b) Haar wavelet.

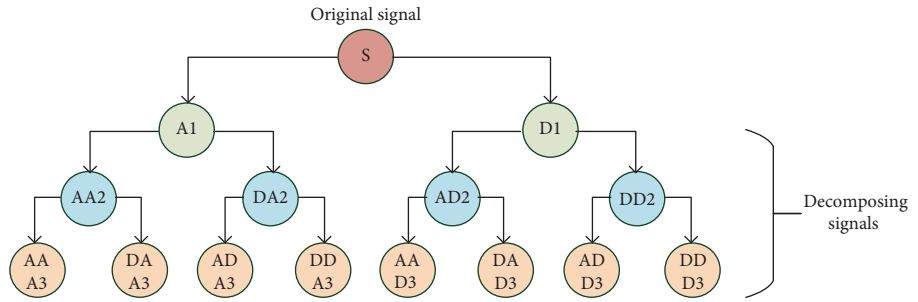

Figure 2: Schematic diagram of Haar wavelet decomposition.

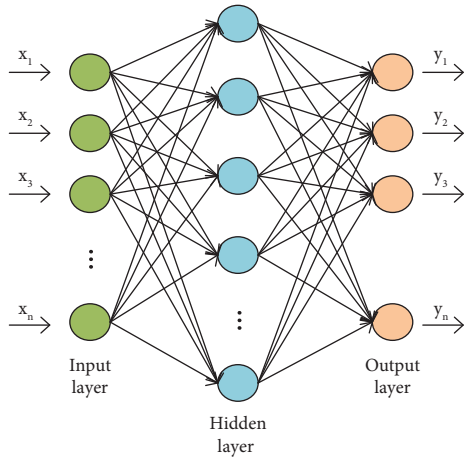

Figure 3: RBF neural network structure diagram.

$$y_k = \sum_{i=1}^{h} W_{\text{ik}} * \phi\left(\|x - c_i\|, \delta_i\right), \tag{7}$$

where $\|x - c_i\|$ stands for the norm of the vector value $x - c_i$; c_i denotes the center vector; δ_i refers to the basis function width; W_{ik} expresses the hidden and output layers' weights. It selects the Gaussian function as the basis function of the RBF, as expressed in the following equation:

$$\phi\left(\|x - c_i\|, \delta_i\right) = \exp\left\{\left| -\frac{\|x - c_i\|}{\delta_i^2}\right|\right\}. \tag{8}$$

In equation (8), although RBF has many advantages, they also have shortcomings. When using RBF networks, most of the time, parameter values such as center, width, and weight are randomly generated. However, the randomly generated values are difficult to achieve the desired optimal value. The training speed of RBF is relatively slow, so in the early stage of training, the K-means clustering method is introduced to find the center vector. First, it initializes the RBF, determines the amount of input nodes and hidden layer nodes, and selects a vector with the same number of training samples as the initial clustering center. Select K vectors with the same number of training samples as the initial clustering center, and the clustering center is $c_1, c_2, \cdots, c_K$. It calculates the Euclidean distance between the input initial value and the cluster center point and performs similarity matching on the samples based on the minimum Euclidean distance [19]. The sample clustering distance satisfies the following equation:

$$d_j = \min\left\|x_i - c_j\right\| \ (j = 1, 2, \cdots, k), \tag{9}$$

where d_j indicates the minimum Euclidean distance; $\min\|x_i - c_j\|$ denotes the minimum norm value of the vector value; c_j stands for the center vector.

It integrates all clustering sets and then takes the mean of the sample values. If the distribution of the sample keeps changing, it repeats the Euclidean distance and means calculations until the cluster distribution remains unchanged, and the final output cluster center is the center vector c_j obtained. The dispersion degree equation of each cluster point is shown in the following equation:

$$d_j\text{inner} = \sqrt{\sum \left\| x_i - c_j \right\|^2}, \tag{10}$$

where $\sqrt{\sum \|x_i - c_j\|^2}$ refers to the square of the sum of the minimum norm values of the vector values. The K-means clustering method flowchart is shown in Figure 4.

In Figure 4, the first is to initialize the data, input sample information, and classify it. According to the K-means method, the center of the clustering is selected and whether the algorithm converges is determined. If it converges, it needs to output; if not, it needs to repeat the clustering steps. In RBF, the width is a very important parameter, which can affect the fitting ability, generalization ability, and calculation efficiency of neural network [20]. According to equation (10), it calculates the width of the basis function, as shown in the following equation:

$$\delta_j = d_j - d_j\text{inner}, \tag{11}$$

where d_j refers to the clustering distance of the sample and δ_j means the degree of sample dispersion.

This study takes the width value of the function as 25 to prevent the neural network from being too sensitive to data and unable to generalize to other datasets due to the small width value of the function and the small scope of the radial basis function, which may lead to overfitting in the neural network. When the width of the function is too large, the network cannot capture smaller scale features of the data, resulting in underfitting [21]. The schematic diagram of the RBF model is expressed in Figure 5.

In RBF, the input values are processed through the Gaussian function, and the K-means clustering method is introduced to find the center vector. After undergoing the Haar wavelet transform, the output value of the RBF is obtained. Weights can map input data into high-dimensional feature spaces. In the hidden layer, the output value is jointly decided by the weight matrix and the RBF, and the characteristic value in the data can be extracted. Therefore, the pseudoinverse algorithm is applied to optimize the weight values, and the optimization expression is shown in the following equation:

$$W = \phi^+ d, \tag{12}$$

where W expresses the weight; d denotes the expected response vector; ϕ^+ means the pseudomatrix of the matrix.

The calculation method for pseudomatrices is shown in the following equation:

$$\phi^+ = \left(\phi^T \phi\right)^{-1} \phi^T, \tag{13}$$

where ϕ^T means the transposition of the matrix; $(\phi^T \phi)^{-1}$ refers to the inverse of the matrix.

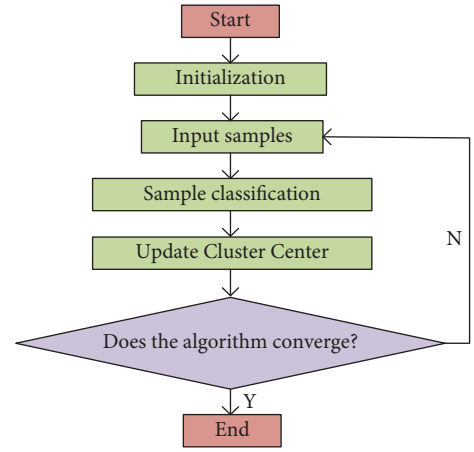

Figure 4: Flowchart of K-means algorithm improved RBF.

The recursive orthogonal least-squares method is an online learning algorithm used to solve linear regression problems. Every time a new data value arrives, the prediction model is gradually improved by merging the previous calculation result with the new data value and updating the weight value recursively. First, it needs to perform orthogonal decomposition on the output matrix of the second hidden layer. The expression is shown in the following equation:

$$H = Q[R, 0]^T, \tag{14}$$

where Q denotes the orthogonal matrix of the matrix and R means the upper triangular matrix.

By multiplying Q^T by Q to the left, the equation is obtained as shown in the following equation:

$$Q^T Q = \left[D', D''\right], \tag{15}$$

where Q^T denotes the transposition of the matrix; D' expresses the expected signal matrix; D'' stands for the error matrix.

The norm representation of a matrix is shown in equation (16), based on the orthogonal invariance of norm F, $\|QA\|_F = \|A\|_F$.

$$J(w) = \left\| Q\left(\left[D', D''\right]^T - [R, O]^T W \right) \right\|_F^2, \tag{16}$$

where $\|D''\|_F^2$ refers to the residual error value and W indicates the weight matrix.

When the amount of samples is large, a recursive algorithm is introduced to obtain the norm equation of the matrix as shown in the following equation:

$$J(t) = \left\| D'(t) - R(t) W(t) \right\|_F^2 + \left\| d''(t) \right\|_F^2 + \left\| D''(t-1) \right\|_F^2, \tag{17}$$

where $W(t)$ represents the weight matrix, and the expression is $W(t) = R^{-1}(t) D'(t)$. $e(t) = \|d'(t)\|_F^2 + \|D''(t-1)\|_F^2$ represents the residual error value.

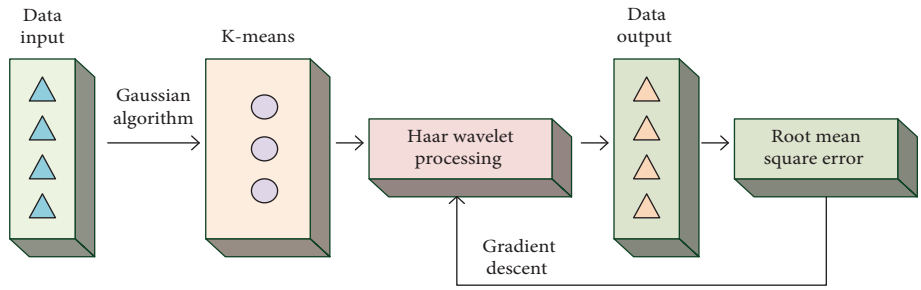

Figure 5: Schematic diagram of RBF neural network model.

According to the above steps, the weights of the RBF neural network can be optimized. First, initialize the upper corner matrix $R(t) = R(0) = aI$ and residual matrix $D''(t) = \|D''\|_F^2 = 0$. Among them, $a = 0.01$. Then, solve for $Q(t)$, $R(t)$, $\|d'(t)\|_F^2$, and $\|D''(t)\|_F^2$. If t is less than the maximum value, $t = t + 1$. If the requirements are not met, proceed to the next step. Calculate $Q(N)$, $R(N)$, W, and $\|D''(N)\|_F^2$ and finally stop the process. Therefore, by optimizing the weights in the RBF, the error between the network output and the actual output is minimized, improving the accuracy and robustness of the model.

3.3. Construction of RBF Model Optimized by WCA. The WCA optimizes the parameter problem in the model by simulating the hunting behavior of wolf packs [22]. Compared with other algorithms, the WCA has a strong global search ability, and each wolf in the wolf pack is an independent entity that can accelerate the search process through parallel computing. At the same time, it is difficult to set parameters, and the rate of convergence is slow in the WCA. To improve the searchability of the WCA, adaptive step size optimization is carried out on the WCA. The hunting model of a wolf pack is shown in Figure 6.

The algorithm mainly simulates the behavior of wolves in a wolf pack, so that each wolf represents a parameter problem and searches for the global optimal solution through cooperation and competition among wolves [23]. First, the population of wolves is randomly set to N, and the feasible domain search space is D. The position and state of any wolf in the space can be represented by the following equation:

$$X_i = (x_{i1}, x_{i2}, x_{i3}, \cdots x_{il}), 1 \le i \le N, 1 \le d \le D, \tag{18}$$

where i represents any wolf and d denotes the distance between wolves.

The concentration function of wolf perception of prey odor is shown in the following equation:

$$F = \sum_{i=1}^{T} |y_i - o_i|, \tag{19}$$

where T means the amount of times an individual in the population has walked between feasible regions; y_i refers to the actual output value of the wolf; o_i denotes the predicted output value of the wolf.

According to the perceived concentration of prey odor in wolf packs, wolves can be divided into head, probe, and fierce wolves. When the odor concentration perceived by the probe wolf is higher than that perceived by the head wolf, the probe wolf takes over the position of the head wolf. If the odor concentration perceived by the probe wolf is less than that perceived by the head wolf, the probe wolf swims in different directions once, returns to its original position, and records the odor concentration perceived in different directions. Finally, it chooses the direction with the highest concentration of prey for walking, and the position transformation equation is shown in the following equation:

$$x_{\mathrm{id}}^{k+1} = x_{\mathrm{id}}^{k} + \sin\left(2\pi \times \frac{p}{h}\right) \mathrm{Step}_a^d, \tag{20}$$

where x_{id}^k indicates the probe wolf position in the feasible domain search space after the k iteration; Step_a^d expresses the walking step size; h stands for the direction that can be explored, while p denotes the direction that the wolf pack ultimately chooses to walk.

As the wolf approaches its prey, it will summon fierce wolves from the pack to attack. If the odor perception concentration of the fierce wolf is higher than that of the head wolf, it replaces the head wolf to summon. If the odor perception density of the fierce wolf is lower than that of the head wolf, the fierce wolf will continue to approach the head wolf. The expression for the position change of a fierce wolf in the feasible domain space is shown in the following equation:t

$$x_{\mathrm{id}}^{k+1} = x_{\mathrm{id}}^{k} + \mathrm{Step}_b^d \times \frac{\left(g_d^k - x_{\mathrm{id}}^k\right)}{\left|g_d^k - x_{\mathrm{id}}^k\right|}, \tag{21}$$

where g_d^k expresses the head wolf position in the feasible domain space after the $k + 1$st iteration.

Step_b^d means the stride of the fierce wolf towards the head wolf. Probe wolves track their prey, while fierce wolves assist the head wolves in capturing their prey. After $k + 1$ iterations, the position of the wolf pack within the feasible range is shown in the following equation:

$$x_{\mathrm{id}}^{k+1} = x_{\mathrm{id}}^{k} + \lambda \cdot \mathrm{Step}_c^d \cdot \left|G_d^k - x_{\mathrm{id}}^k\right|, \tag{22}$$

where G_d^k denotes the head wolf position; Step_c^d expresses the step length of the siege prey; λ refers to a natural number between $[-1, 1]$.

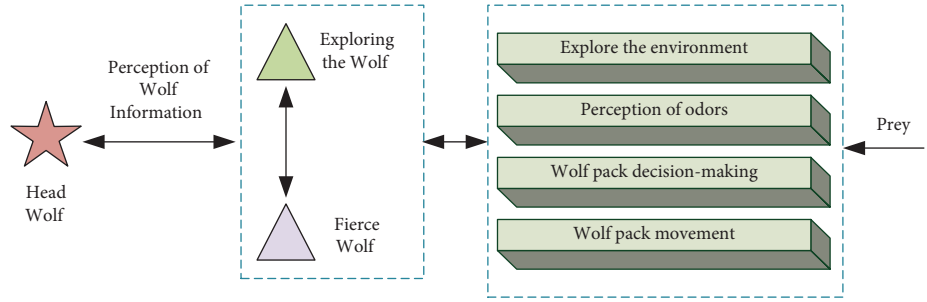

Figure 6: Wolf hunting model.

The relationship between the steps of the entire hunting is shown in the following equation:

$$\text{Step}_a^d = \frac{\text{Step}_b^d}{2} = 2\text{Step}_c^d = \frac{\left|\max_d - \min_d\right|}{S}, \tag{23}$$

where S represents the step size factor, which can be used to indicate the accuracy of the above formula solution.

To ensure the stability and accuracy of the algorithm, it needs to maintain the diversity of individuals in the wolf pack, eliminate the wolf with the lowest fitness value in the wolf pack, and generate the head wolf for the next iteration. Finally, it determines whether the algorithm meets the target value or reaches the maximum amount of iterations. In order to prevent traditional wolf pack algorithms from easily falling into local optima, unlike the standard wolf pack algorithm where all fierce wolves participate in summoning behavior, the improved wolf pack algorithm also searches for the optimal value within its spatial range during the summoning process. The adaptive step size of the traditional wolf pack algorithm is improved. After $k+1$ iterations in the feasible domain space, the position of the wolf pack is shown in the following equation:

$$\begin{aligned} x_{\text{id}}^{k+1} &= x_{\text{id}}^k + \text{Step} \cdot \frac{\left(g_d^k - x_{\text{id}}^k\right)}{\left|g_d^k - x_{\text{id}}^k\right|} \\ &= x_{\text{id}}^k + \text{rand} \cdot \text{norm}\left(x_{\text{id}}^k - g_d^k\right) \cdot \frac{\left(g_d^k - x_k^k\right)}{\left|g_d^k - x_{\text{id}}^k\right|}. \end{aligned} \tag{24}$$

In the feasible domain space, after an iterative optimization process, the fierce wolf assists the probing wolf in rounding up the prey. For the k-th generation wolf pack, the position of the prey is consistent with that of the head wolf. At this point, the position of the entire wolf pack in the feasible domain space is shown in the following equation:

$$\begin{aligned} x_{\text{id}}^{k+1} &= x_{\text{id}}^k + \lambda \cdot \text{Step} \cdot \left(G_d^k - x_{\text{id}}^k\right) \\ &= x_{\text{id}}^k + \lambda \cdot \text{rand} \cdot \text{norm}\left(x_{\text{id}}^k - G_d^k\right) \cdot \left|G_d^k - x_{\text{id}}^k\right|. \end{aligned} \tag{25}$$

The flowchart of the wolf colony algorithm improved the radial basis function neural network (WCA-RBF) model as shown in Figure 7.

4. Model Performance Testing and Simulation Experiment Analysis

To achieve FD in ACs, the RBF was optimized with the WCA, and a WCA-RBF model was constructed. This section contains two parts to test the model. The first part is the performance test of the wolf colony optimization RBF model, and the second part is the simulation experiment of the faults in the Sallen–Key bandpass filter.

4.1. Improved WCA and Optimized RBF Model Performance Testing. To verify the optimization effect of the improved WCA in data processing, a multimodal Schaffer benchmark function and a single-mode sphere benchmark function were selected. Under MATLAB simulation software, the standard WCA was compared and tested with the improved WCA. The iteration curves of the two algorithms are shown in Figure 8.

Figure 8(a) shows the test results of two algorithms under the multimodal Schaffer benchmark function. The standard WCA and improved WCA tended to stabilize after 70 and 60 iterations, with a fitness value of 10^{-1} and 10^{-3}, respectively. Figure 8(b) shows the test results of two algorithms under a single-mode sphere reference function. The standard WCA tended to stabilize after 60 iterations, with a fitness value of 10^{-1}. The improved WCA tended to stabilize after 55 iterations, with a fitness of 10^{-3}. From Figure 8, the rate of convergence of the two algorithms was basically the same, but at the beginning of training, the rate of convergence of the improved WCA was significantly faster than that of the standard WCA. In the later iterative optimization, the convergence accuracy of the improved WCA was significantly higher than that of the standard WCA. The improved WCA algorithm had better global optimization ability, faster rate of convergence, and higher convergence accuracy. To examine the performance of the WCA-RBF and the RBF, 100 sets of working data in the circuit were selected. Using MATLAB simulation software, the first 60 sets of data were applied as training data, and the last 40 sets of data were utilized as model testing data. The error curve obtained from the test is shown in Figure 9.

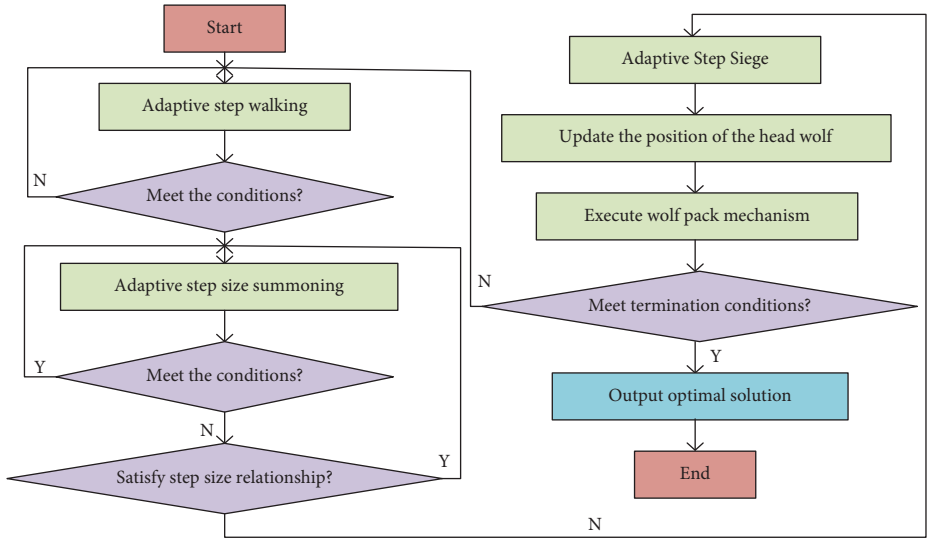

Figure 7: Flowchart of WCA-RBF.

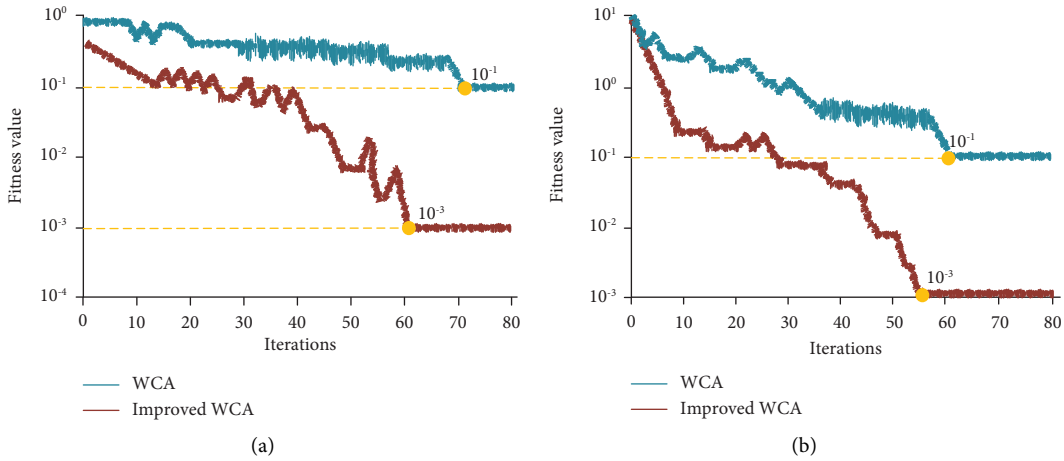

Figure 8: Iterative curve of WCA and improved WCA. (a) Test result graph under multimodal Schaffer benchmark function. (b) Test result graph under single-mode sphere benchmark function.

From Figure 9, the error values of the two neural network models decreased with the increase in training times. The test results of the two neural network models in the test and training sets were basically consistent. Figure 9(a) shows the error variation curve of the RBF model during the training. After 60 training sessions, the RBF model converged with an error value of 10^{-1}. Figure 9(b) shows the error variation curve of the WCA-RBF model during the training. The WCA-RBF model converged after 40 training sessions, with an error value of 10^{-3}. The RBF model optimized by the WCA had a faster rate of convergence, and the accuracy of FD has also been improved. To evidence the performance of the WCA-RBF in FD, GA-RBF was used for comparative testing with the WCA-RBF model. It inputs 50 sets of FD samples into two models and conducts 10 tests each. The test results are shown in Figure 10.

Figure 10(a) shows the mean squared error (MSE) curves of GA-RBF and WCA-RBF models. The average value of the MSE of the GA-RBF model was 0.78, and that of the WCA-RBF model was 0.45. Figure 10(b) shows the consumption time curves of the GA-RBF and the WCA-RBF models. The average consumption time of the GA-RBF model was 275 s, while that of the WCA-RBF model was 148 s. The error value distribution of the WCA-RBF model

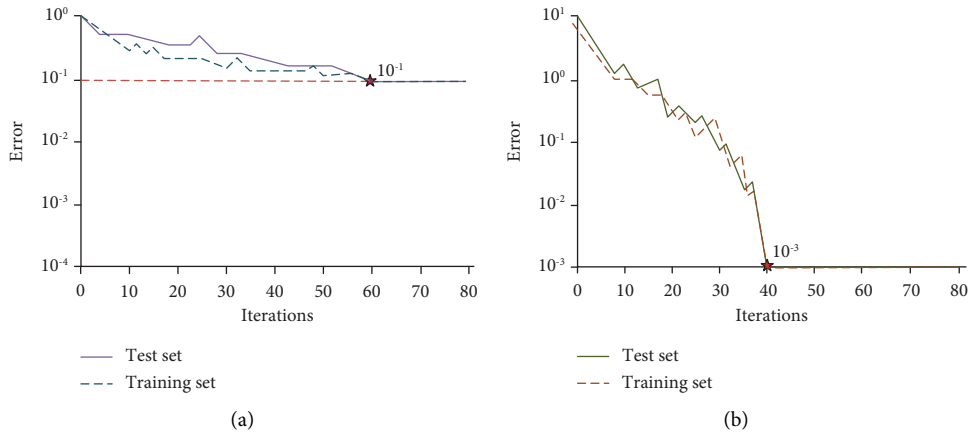

Figure 9: Training error curves of two models. (a) Error variation curve of RBF neural network model. (b) Error variation curve of WCA-RBF neural network model.

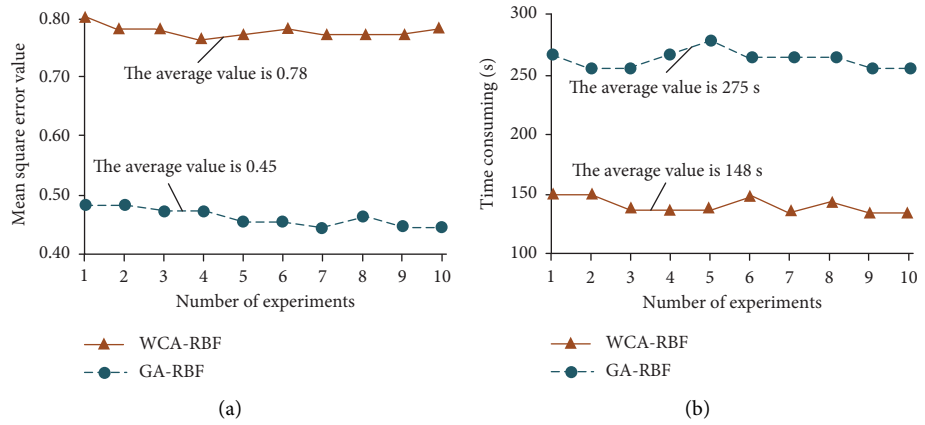

Figure 10: MSE and consumption time curve. (a) Mean square error curve. (b) Consumption time curve.

was more concentrated, the FD results were more accurate, and the testing efficiency was faster. It selected 10 different fault types from 100 training data and tested the original RBF, GA-RBF, and WCA-RBF models. The accuracy of FD is expressed in Table 1.

From Table 1, the original RBF model had the lowest FD rate, with an average diagnosis rate of 64.89%. The FD rate of the GA-RBF model was in the middle, with an average diagnosis rate of 84.58%. The WCA-RBF model had the best FD rate, reaching 95.52%.

4.2. Case Study of FD. To assess the effect of the WCA-RBF in ACs, FD was performed on soft faults in ACs. The soft faults in ACs were divided into 8 fault states and 1 normal state. The test experiment sets the input node of the model to 6 and the output node to 4. The wavelet transform test data for each fault type are expressed in Table 2.

Table 2 shows the test data of 8 fault states and 1 normal state after normalization. The up and down arrows represented soft fault situations with an up deviation of 50% and a down deviation of 50%, respectively. It selected the filtering circuit in the international standard circuit for testing, with nominal values of $R1 = R2 = R3 = 10\,k$, $R4 = R5 = 10\,k$, $R6 = 3\,k$, $R7 = 7\,k$, $C1 = C2 = 20\,nF$. The allowable tolerances for capacitance and resistance were set to 10% and 5%, respectively. Using OrCAD/PSpice 10.5 software for AC excitation of the standard filtering circuit, Monte Carlo analysis was performed on 8 fault states and normal states, and the output data were used at frequency intervals. It extracted 200 characteristic values at different frequencies of the out end of the third filter in the standard filtering circuit for testing. The MC simulation circuit diagram obtained through PSpice is expressed in Figure 11.

From Figure 11, the overall trend of the MC simulation diagram in both states was consistent. Figure 11(a) shows the MC simulation diagram in the case of C2 being too large. From the graph, the voltage value before the peak was basically the same as under normal conditions, and the voltage value at the peak was 1.2 V. Figure 11(b) shows the MC simulation diagram under normal conditions. The peak in the figure was relatively high, and the voltage value at the peak was 1.4 V. The voltage values of both graphs showed a decreasing trend after the peak, and at 10 KHz, the voltage

Table 1: Accuracy of FD by different diagnostic methods.

Fault number	RBF (%)	GA-RBF (%)	WCA-RGF (%)
No. 1	60.85	83.52	95.25
No. 2	65.58	84.56	94.88
No. 3	64.31	82.59	95.62
No. 4	67.25	86.06	95.35
No. 5	68.23	84.26	95.42
No. 6	64.55	84.09	94.99
No. 7	64.89	80.99	96.32
No. 8	65.22	87.52	96.25
No. 9	63.52	86.24	97.05
No. 10	64.56	85.98	94.12

Table 2: Wavelet transform data of different fault types.

Failure mode	EA5	E5	E4	E3	E2	E1
Normal	0.8000	0.1355	0.1152	0.1035	0.1008	0.1000
C1↑	0.8000	0.1362	0.1145	0.1028	0.1009	0.1000
C1↓	0.8000	0.1342	0.1126	0.1034	0.1007	0.1000
C2↑	0.8000	0.1356	0.1125	0.1037	0.1008	0.1000
C2↓	0.8000	0.1354	0.1136	0.1045	0.1008	0.1000
R2↑	0.8000	0.1325	0.1148	0.1333	0.1007	0.1000
R2↓	0.8000	0.1325	0.1205	0.1035	0.1008	0.1000
R3↑	0.8000	0.1406	0.1185	0.1032	0.1008	0.1000
R3↓	0.8000	0.1368	0.1145	0.1033	0.1006	0.1000

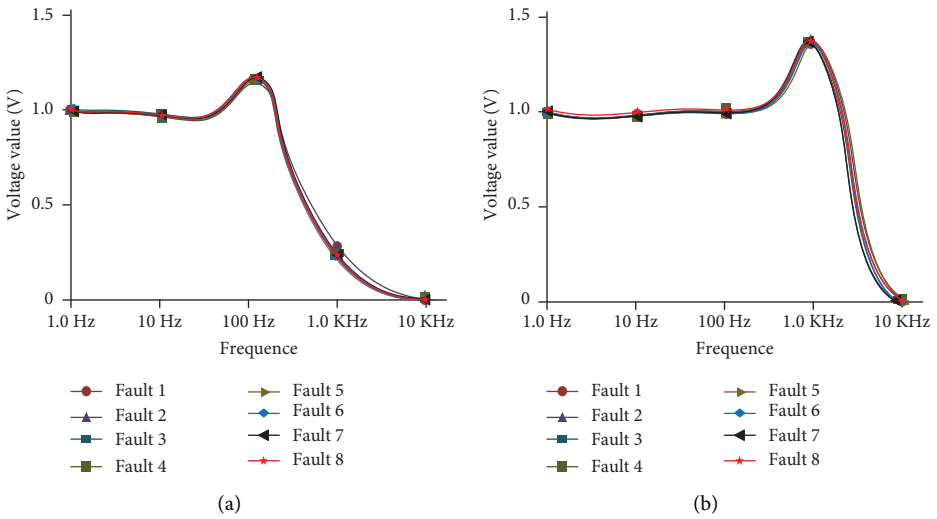

Figure 11: MC simulation diagram under different scenarios. (a) MC simulation diagram in the case of C2 being too large. (b) MC simulation diagram under normal conditions.

dropped to 0 V. The original RBF, GA-RBF, and WCA-RBF models were used for FD as shown in Figure 11. The diagnostic time is shown in Figure 12.

In Figure 12, the average testing time required for the RBF, GA-RBF, and WCA-RBF models was 56.5 seconds, 48.3 seconds, and 41.5 seconds, respectively. The WCA-RBF model consumed less time and tested faster. Compared the diagnostic rates of the RBF neural network model, GA-RBF neural network model, WCA-RBF neural network model, and the improved RBF neural network model based on the particle swarm optimization algorithm, the results are shown in Table 3.

From Table 3, the original RBF model had the lowest FD rate, with an average diagnosis rate of 70.36%. The fault diagnosis rate of the GA-RBF model was slightly better than that of the RBF model, with an average diagnosis rate of 86.24%. The average diagnostic rate of the PSO-RBF model was 94.46%. The WCA-RBF model had the best fault diagnosis rate, reaching 96.17%. From this, GA-RBF and

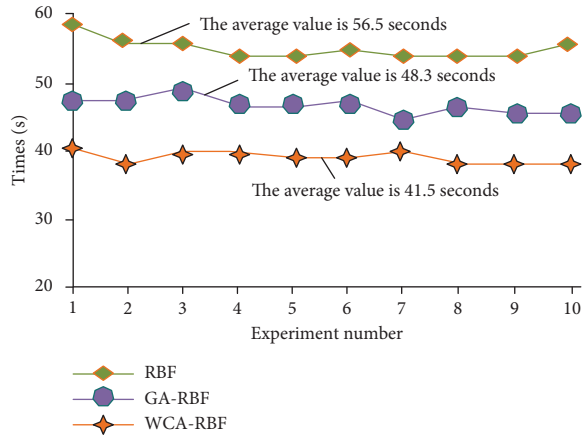

Figure 12: Test time curve.

Table 3: Accuracy of different fault types.

Fault number	RBF (%)	GA-RBF (%)	WCA-RBF (%)	PSO-RBF (%)
Normal	69.82	85.39	95.36	93.41
C1↑	68.59	86.99	95.48	92.59
C1↓	72.05	87.35	96.05	94.57
C2↑	71.33	86.59	96.35	95.29
C2↓	70.25	85.76	97.54	95.85
R2↑	70.55	86.32	95.88	94.13
R2↓	69.98	84.95	95.36	94.73
R3↑	70.53	87.29	96.77	94.16
R3↓	70.14	85.55	96.82	95.37

WCA-RBF models had an improved diagnostic rate compared to the unoptimized RBF model. The RBF model optimized by the WCA had the highest diagnostic rate; therefore, the RBF model optimized by the WCA had good diagnostic performance. It has been proven that the WCA optimized the RBF model and performed well in AC FD.

5. Conclusions

With the complexity and expansion of circuit systems, AC FD plays a crucial role in AC design and manufacturing. This research aimed at the problem of AC FD, optimized the RBF based on the WCA, and constructed the model of RBF optimized by the WCA. The test findings of the model showed that the WCA-RBF model converged after 40 times of training, and the error value was 10^{-3}. The traditional RBF model converged after 80 times of training, and the error value was 10^{-1}. The average value of MSE of the GA-RBF model was 0.78. The mean value of MSE of the WCA-RBF model was 0.45. The FD rate of the original RBF, GA-RBF, and WCA-RBF models was compared. The FD rate of the WCA-RBF model was the best, reaching 95.52%. The error value distribution of the WCA-RBF model was more concentrated, the FD rate was higher, and the FD result was more accurate. The WCA was used to optimize the RBF model to diagnose the faults in the CTSV filter circuit, and the FD rate reached 96.17%. Therefore, the WCA-RBF model had a good diagnosis effect for different faults in the AC.

Abbreviations

ACs: Analog circuits
FD: Fault diagnosis
RBF: Radial basis function
WCA: Wolf colony algorithm
GA-RBF: Genetic algorithm-radial basis function
EMG: Electromyogram
PSO: Particle swarm optimization
WCA-RBF: Wolf colony algorithm-radial basis function
PSO-RBF: Particle swarm optimization-radial basis function
MSE: Mean squared error.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The author declares that there are no conflicts of interest.

References

[1] W. He, Y. He, and B. Li, "Generative adversarial networks with comprehensive wavelet feature for fault diagnosis of analog circuits," *IEEE Transactions on Instrumentation and Measurement*, vol. 69, no. 9, pp. 6640–6650, 2020.

[2] L. Ji, C. Fu, and W. Sun, "Soft fault diagnosis of analog circuits based on a ResNet with circuit spectrum map," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 68, no. 7, pp. 2841–2849, 2021.

[3] A. Moezi and S. M. Kargar, "Fault isolation of analog circuit using an optimized ensemble empirical mode decomposition approach based on multi-objective optimization," *Proceedings of the Institution of Mechanical Engineers- Part I: Journal of Systems and Control Engineering*, vol. 235, no. 9, pp. 1555–1570, 2021.

[4] L. Kou, C. Liu, G. Cai, J. Zhou, Q. Yuan, and S. Pang, "Fault diagnosis for open-circuit faults in NPC inverter based on knowledge-driven and data-driven approaches," *IET Power Electronics*, vol. 13, no. 6, pp. 1236–1245, 2020.

[5] W. Huang, J. Du, W. Hua et al., "Current-based open-circuit fault diagnosis for PMSM drives with model predictive control," *IEEE Transactions on Power Electronics*, vol. 36, no. 9, pp. 10695–10704, 2021.

[6] S. Zhuo, A. Gaillard, L. Xu, C. Liu, D. Paire, and F. Gao, "An observer-based switch open-circuit fault diagnosis of DC–DC converter for fuel cell application," *IEEE Transactions on Industry Applications*, vol. 56, no. 3, pp. 3159–3167, 2020.

[7] Y. Sun, J. Xu, G. Lin, W. Ji, and L. Wang, "RBF neural network-based supervisor control for maglev vehicles on an elastic track with network time delay," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 1, pp. 509–519, 2022.

[8] C. Lu, B. He, and R. Zhang, "Evaluation of English interpretation teaching quality based on GA optimized RBF neural network," *Journal of Intelligent and Fuzzy Systems*, vol. 40, no. 2, pp. 3185–3192, 2021.

[9] A. Hashemi Fath, F. Madanifar, and M. Abbasi, "Implementation of multilayer perceptron (MLP) and radial basis function (RBF) neural networks to predict solution gas-oil ratio of crude oil systems," *Petroleum*, vol. 6, no. 1, pp. 80–91, 2020.

[10] F. Yao, J. Zhao, X. Li, L. Mao, and K. Qu, "RBF neural network based virtual synchronous generator control with improved frequency stability," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 6, pp. 4014–4024, 2021.

[11] M. Yu, G. Li, D. Jiang et al., "Application of PSO-RBF neural network in gesture recognition of continuous surface EMG signals," *Journal of Intelligent and Fuzzy Systems*, vol. 38, no. 3, pp. 2469–2480, 2020.

[12] P. Sohrabi, B. Jodeiri Shokri, and H. Dehghani, "Predicting coal price using time series methods and combination of radial basis function (RBF) neural network with time series," *Mineral Economics*, vol. 36, no. 2, pp. 207–216, 2023.

[13] S. Kumar, R. Kumar, R. P. Agarwal, and B. Samet, "A study of fractional Lotka-Volterra population model using Haar wavelet and Adams-Bashforth-Moulton methods," *Mathematical Methods in the Applied Sciences*, vol. 43, no. 8, pp. 5564–5578, 2020.

[14] K. Shah, Z. A. Khan, A. Ali, R. Amin, H. Khan, and A. Khan, "Haar wavelet collocation approach for the solution of fractional order COVID-19 model using Caputo derivative," *Alexandria Engineering Journal*, vol. 59, no. 5, pp. 3221–3231, 2020.

[15] V. Mehandiratta, M. Mehra, and G. Leugering, "An approach based on Haar wavelet for the approximation of fractional calculus with application to initial and boundary value problems," *Mathematical Methods in the Applied Sciences*, vol. 44, no. 4, pp. 3195–3213, 2021.

[16] Y. C. Du, H. X. Wang, H. R. Shi, and H. Y. Liu, "Wavelet based soft threshold denoising for vortex flowmeter," in *Proceedings of the 2009 IEEE Instrumentation and Measurement Technology Conference*, pp. 139–143, Singapore, May 2009.

[17] F. Bulut, "Low dynamic range histogram equalization (LDR-HE) via quantized Haar wavelet transform," *The Visual Computer*, vol. 38, no. 6, pp. 2239–2255, 2022.

[18] N. Shakeel and S. Shakeel, "Context-free word importance scores for attacking neural networks," *Journal of Computational and Cognitive Engineering*, vol. 1, no. 4, pp. 187–192, 2022.

[19] A. Sharafian, A. Sharifi, and W. Zhang, "Fractional sliding mode based on RBF neural network observer: application to HIV infection mathematical model," *Computers and Mathematics with Applications*, vol. 79, no. 11, pp. 3179–3188, 2020.

[20] X. Sun, Z. Zhou, and Y. Wang, "Water resource carrying capacity and obstacle factors in the Yellow River basin based on the RBF neural network model," *Environmental Science and Pollution Research*, vol. 30, no. 9, pp. 22743–22759, 2022.

[21] X. Zhang, P. Zhang, T. Wang, Y. Zheng, L. Qiu, and S. Sun, "Compressive strength and anti-chloride ion penetration assessment of geopolymer mortar merging PVA fiber and nano-SiO2 using RBF–BP composite neural network," *Nanotechnology Reviews*, vol. 11, no. 1, pp. 1181–1192, 2022.

[22] S. Dereli, "A new modified grey wolf optimization algorithm proposal for a fundamental engineering problem in robotics," *Neural Computing and Applications*, vol. 33, no. 21, pp. 14119–14131, 2021.

[23] M. Qaraad, S. Amjad, N. K. Hussein, and M. A. Elhosseini, "Large scale SALP-based grey wolf optimization for feature selection and global optimization," *Neural Computing and Applications*, vol. 34, no. 11, pp. 8989–9014, 2022.