

## Article

# Developing a Hybrid Algorithm Based on an Equilibrium Optimizer and an Improved Backpropagation Neural Network for Fault Warning

Jiang Liu <sup>1,\*</sup>, Changshu Zhan <sup>1,2</sup>, Haiyang Wang <sup>1</sup>, Xingqin Zhang <sup>2</sup>, Xichao Liang <sup>1</sup>, Shuangqing Zheng <sup>1</sup>, Zhou Meng <sup>1</sup> and Guishan Zhou <sup>1</sup>

<sup>1</sup> Independent Researcher, No. 78, Canghai Road, Bohai New District, Huanghua 061100, China

<sup>2</sup> Transportation College, Northeast Forestry University, Harbin 150040, China

\* Correspondence: 16151060@ceic.com

**Abstract:** In today's rapidly evolving manufacturing landscape with the advent of intelligent technologies, ensuring smooth equipment operation and fostering stable business growth rely heavily on accurate early fault detection and timely maintenance. Machine learning techniques have proven to be effective in detecting faults in modern production processes. Among various machine learning algorithms, the Backpropagation (BP) neural network is a commonly used model for fault detection. However, due to the intricacies of the BP neural network training process and the challenges posed by local minima, it has certain limitations in practical applications, which hinder its ability to meet efficiency and accuracy requirements in real-world scenarios. This paper aims to optimize BP networks and develop more effective fault warning methods. The primary contribution of this research is the proposal of a novel hybrid algorithm that combines a random wandering strategy within the main loop of an equilibrium optimizer (EO), a local search operator inspired by simulated annealing, and an adaptive learning strategy within the BP neural network. Through analysis and comparison of multiple sets of experimental data, the algorithm demonstrates exceptional accuracy and stability in fault warning tasks, effectively predicting the future operation of equipment and systems. This innovative approach not only overcomes the limitations of traditional BP neural networks, but also provides an efficient and reliable solution for fault detection and early warning in practical applications.

**Keywords:** fault warning; BP neural network; deep learning; enhanced equilibrium optimizer



**Citation:** Liu, J.; Zhan, C.; Wang, H.; Zhang, X.; Liang, X.; Zheng, S.; Meng, Z.; Zhou, G. Developing a Hybrid Algorithm Based on an Equilibrium Optimizer and an Improved Backpropagation Neural Network for Fault Warning. *Processes* **2023**, *11*, 1813. <https://doi.org/10.3390/pr11061813>

Academic Editor: Jie Zhang

Received: 6 May 2023

Revised: 6 June 2023

Accepted: 6 June 2023

Published: 14 June 2023



**Copyright:** © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

With the ongoing advancement of modern industry, an increasing variety of mechanical equipment, industrial production lines, and electrical control systems are being utilized [1,2]. However, during long-term operation, these devices and systems inevitably encounter various failures, resulting in significant challenges and losses for both production and maintenance [3]. Consequently, developing effective methods for early warning and diagnosis of equipment and system faults has emerged as a pressing issue to address [4].

In this context, adopting machine learning-based fault warning methods has become a vital approach to modern industrial fault prevention [5]. Fault warning is a management strategy that enables early detection of faults and implementation of appropriate measures to prevent or mitigate their impact on enterprises or consumers. Not only can fault warning enhance operational efficiency and customer satisfaction for businesses, but it also reduces maintenance expenses and minimizes production line downtime caused by failures [6].

BP neural networks are widely used in the field of fault early warning, primarily due to their strong learning capability, rapid information processing speed, and error adaptivity benefits. However, despite the remarkable results achieved by BP neural networks, they

possess a series of non-negligible drawbacks that negatively impact the accuracy and stability of fault early warning systems.

Firstly, BP neural networks are prone to falling into the local minima problem. When the weight update during training becomes stuck at a local optimal solution, the neural network may not be able to find the global optimum solution, ultimately affecting the warning accuracy. Secondly, selecting an appropriate BP neural network configuration is both complex and challenging. Failing to suitably determine the parameters can result in suboptimal network performance. Furthermore, BP neural networks suffer from slow convergence, and the training process often takes a substantial amount of time. This increases the response delay of the early warning system, which may allow a fault to occur before a warning is issued.

The primary objective of this paper is to effectively optimize BP neural networks for developing more efficient fault warning methods. According to the No Free Lunch theorem of optimization problems, current algorithms may not be proficient in addressing this issue. This paper's key contribution is the introduction of an enhanced metaheuristic algorithm, based on EO, which incorporates a random wandering strategy and the concept of simulated annealing. To the best of our knowledge, based on the existing literature, this metaheuristic algorithm has not been employed to solve any variant of BP neural networks. In this study, we implement it practically. Furthermore, we integrate an adaptive learning strategy into the BP neural network and compare its performance with other cutting-edge fault warning algorithms.

The remainder of this paper is organized as follows: Section 2 provides a comprehensive literature review of the latest advancements in fault warning algorithms. In Section 3, we present our proposed hybrid approach in detail, highlighting the integration of an EO with a Backpropagation (BP) neural network. Section 4 presents a case study involving widely used industrial equipment, showcasing the application of our approach for failure warning. To evaluate the effectiveness of our hybrid method, Section 5 compares it with other state-of-the-art techniques. Finally, Section 6 concludes the paper with a discussion of key findings, limitations, and potential directions for future research.

## 2. Literature Review

At present, the growing interest in early fault detection has led researchers to develop various techniques. Zhao et al. presented a deep learning approach—deep autoencoder network—that examines sensor data for issuing early warnings about component failures in wind turbine systems [7]. Wang et al. devised an enhanced deep learning, multistage, fusion LSTM model for predicting future reciprocating compressor valve parameters by studying operational data's spatiotemporal features, thus achieving fault early warning goals [8]. Gao et al. employed adaptive deep belief networks and charging data analysis to create an early warning method for electric vehicle charging processes, training the network with historical charging data and offering early warnings using real-time and predicted data [9]. Luo et al. introduced a conditional mutual information technique for selecting valuable variables from multiple options for network training, subsequently developing a BP neural network-based wind turbine gearbox fault diagnosis model utilizing real-time data computations [10]. Wang et al. established a power distribution transformer pre-warning model that accounts for extreme weather conditions and various nonlinear situations, integrating weather data into BP neural network training [11]. Chen et al. optimized the BP neural network with genetic algorithms to issue warnings regarding wind turbine pitch system faults, filtering pitch system parameters with strong power correlation based on SCADA system-monitored parameters for network training [12]. Jiang et al. also created a GA-BP model and investigated a state-based baking machine maintenance method using operational data. They determined weight selection input data through the entropy weighting method, effectively avoiding the influence of subjective factors by selecting reasonable data input samples [13]. Zhang et al. optimized the BP neural network with an improved grey wolf algorithm, examining an electric vehicle charging safety pre-warning model based on

charging statistics, and providing early warnings by comparing post-network fitting data with original data [14]. Chen et al. proposed a BP neural network optimization technique employing parallel factor decomposition and GA, efficiently extracting intricate information from equipment operation using the parallel factor decomposition method, achieving data mining, and considerably enhancing centrifugal pump fault detection efficiency [15]. Lin et al. optimized the BP neural network with an improved sparrow search algorithm, applying the model to active phase-change control device fault detection using specific equipment information [16]. Wu et al. suggested a hybrid method that combined a deep local adaptive network, two-stage qualitative trend analysis, and a five-state Bayesian network for extracting trend states from local moving window data, converting continuous data of abnormal variables into trend state information for fault detection, identification, and diagnosis [17]. Zhou et al. introduced an entropy-based sparsity technique, utilizing LSTM network and envelope analysis data to predict bearing defects and identify issues in complex hydraulic machinery (such as axial piston pumps) [18].

After conducting a comprehensive review of the above literature, we have drawn the following conclusions:

(a) In modern engineering fields, fault warning for equipment and systems is a crucial task. Numerous fault warning technologies are continuously emerging. However, compared to other methods, BP neural networks exhibit several advantages, making them the preferred solution for fault warnings.

(b) BP neural networks face some practical shortcomings, such as local minima and difficulty in manual parameter selection. Slow training speed and the propensity to fall into local minima can negatively impact fault warning systems, leading to prolonged diagnosis times, increased operational costs, and diminished warning accuracy. Addressing this issue is vital, as it can enhance the efficiency and precision of fault warnings, reduce operational expenses, and optimize maintenance strategies. By refining training methods, we can achieve faster and more reliable fault detection in practical applications, ultimately ensuring efficient equipment maintenance and promoting the stability of production processes. Optimizing them using metaheuristic algorithms serves as an essential solution.

(c) The No Free Lunch Theorem suggests that algorithms performing well on some problems may perform poorly on others [19]. Consequently, in accordance with the No Free Lunch Theorem, it is necessary to persistently explore the application of algorithms in new areas to identify the optimal algorithm suitable for specific tasks or scenarios.

EO, proposed in 2020, is a novel optimization algorithm inspired by the physical phenomenon of control volume mass balance. It is characterized by robust optimization capabilities and rapid convergence. Results from various case studies indicate that EO outperforms numerous classical and contemporary algorithms, such as the particle swarm algorithm, grey wolf optimizer, genetic algorithm, gravitational search algorithm, and sparrow search algorithm. According to the NFL theory, we used this excellent algorithm.

With the above background, compared to previous research, this paper offers the following contributions:

(a) We designed an improved equilibrium optimizer (IEO) by incorporating a simulated annealing algorithm into the main loop process of the conventional equilibrium optimizer and augmenting it with an enhanced local search operator that utilizes a random wandering strategy. Experimental analysis effectively demonstrates that our introduced strategy significantly enhances the search capability of the IEO.

(b) Fixed parameters may result in slow convergence and performance degradation for BP neural networks. In this study, we integrated an adaptive update strategy for the parameterization into the BP neural network, effectively improving its performance.

(c) This paper proposes a novel fault warning model and strategy using BP neural networks and IEO, determines its parameters through Taguchi's experimental method, and validates the effectiveness of the method via real-world analysis. Comparisons with other state-of-the-art methods further showcase the exceptional performance of our proposed method, providing a new approach to fault warning.

In summary, this study presents a significant contribution to the field by effectively improving the EO through the combination of SA and the incorporation of a random wandering strategy. For the first time, we have integrated this enhanced algorithm with a BP neural network, which substantially expands its application domain. Furthermore, our research introduces a parameter calibration analysis and fault warning strategy based on this model. To validate our proposed hybrid method, we collected data from real-world industrial equipment commonly used in practice, conducted a thorough investigation of its failure modes and fault anomalies, and tested our method using this case study. The results underscored the high accuracy of our technique in identifying faults, demonstrating that our research can effectively elevate the level of failure warning and contribute to sustainable industrial development.

### 3. Proposed Hybrid Method

In this section, we first describe the improved BP neural network (Section 3.1), followed by a description of IEO (Section 3.2), and finally the framework of IEO-BP is constructed (Section 3.3).

#### 3.1. Improved BP Model

The general flow of the BP neural network is as follows [20]:

Step 1: Initialization: Randomly initialize the connection weights and thresholds.

Step 2: Forward propagation to calculate the output value: the input samples are passed through the input layer to the output layer through the implicit layer, and the output value is calculated, as depicted in Equations (1) and (2).

Input layer to the implied layer:

$$z_j = f \left( \sum_{i=1}^d w_{ji} x_i + b_j \right) \quad (1)$$

Implicit layer to output layer:

$$y_k = f \left( \sum_{j=1}^q w_{kj} z_j + b_k \right) \quad (2)$$

$w_{ji}$  is the connection weight from the input layer to the hidden layer,  $w_{kj}$  is the connection weight from the hidden layer to the output layer,  $b_j$  is the threshold of the hidden layer,  $b_k$  is the threshold of the output layer,  $x_i$  is the  $i$ th feature of the input sample,  $z_j$  is the output of the neuron in the hidden layer.

Step 3: Root-mean-square error (RMSE) calculation: The error is calculated using the difference between the output value and the desired output value. It is calculated as in Equation (3).

$$RMSE = \sqrt{\frac{1}{m} \sum_{i=1}^m (y_k - t_k)^2} \quad (3)$$

where  $y_k$  is the output of the output layer neurons,  $t_k$  is the desired output value, and  $m$  is the number of output layer neurons.

Step 4: Back propagation to adjust the weights and thresholds: the connection weights and thresholds are adjusted by error back propagation from the output layer back to the input layer, as described in Equations (4) and (5).

$$\delta_k = (t_k - y_k) f'(net_k) \quad (4)$$

$$\delta_j = f'(net_j) \sum_{k=1}^c w_{kj} \delta_k \quad (5)$$

where  $f(net)$  is the activation function, and  $f'(net)$  is the derivative of the activation function.

Step 5: The weights and thresholds are updated, as shown in Equations (6)–(9).

$$w_{ij}(t+1) = w_{ij}(t) + \eta \delta_j x_i \quad (6)$$

$$b_j(t+1) = b_j(t) - \eta \delta_j \quad (7)$$

$$w_{jk}(t+1) = w_{jk}(t) + \eta \delta_k z_j \quad (8)$$

$$b_k(t+1) = b_k(t) - \eta \delta_k \quad (9)$$

where  $q$  is the number of neurons in the hidden layer, and  $\eta$  is the learning rate.

Step 6: Repeat 2–4 steps until the error reaches convergence or the training number reaches the limit.

This is followed by our improvement of the traditional BP neural network:

When training a neural network, selecting an appropriate learning rate is crucial to achieving a fast convergence rate and maintaining stability [21]. Utilizing a fixed learning rate may result in issues such as under-learning or overfitting during the training process. As a solution, we employ an adaptive learning rate (exponential decay) to train the BP neural network. In exponential decay, the learning rate decreases as the number of training rounds or iterations increases. This strategy helps reduce the magnitude of weight updates as the optimal solution is approached, enabling more accurate finetuning. Specifically, we adaptively adjust the learning rate according to Equation (10).

$$\eta_{\text{new}} = \eta * I^{\frac{E}{T}} \quad (10)$$

where  $I$  is a constant between 0 and 1, representing the decrease in the learning rate after each decay step, usually set to 0.5;  $E$  denotes the number of current iterations; and  $T$  represents the time interval for decaying the learning rate.

### 3.2. Enhanced EO Model

In addition to the traditional EO procedure (Sections 3.2.2–3.2.5), we improved the EO using a local search operator combined with a random wander strategy with the idea of simulated annealing (Section 3.2.6).

EO is a novel intelligent algorithm inspired by the mass balance equation in physics [20]. The mass balance equation describes the process of mass entry, exit, and generation within a controlled volume and can be expressed as a first-order differential equation, as shown in Equation (11).

$$V \frac{dC}{dt} = QC_{\text{eq}} - QC + G \quad (11)$$

where  $V$  is the control volume;  $C$  represents the concentration within the control volume;  $Q$  denotes the volumetric flow rate into or out of the control volume;  $C_{\text{eq}}$  represents the concentration in the control volume at equilibrium; and  $G$  signifies the mass production rate within the control volume.

The rate of mass change of the control volume,  $V \frac{dC}{dt}$ , when it is 0 means that the control volume enters a stable equilibrium state.

Let  $\lambda = \frac{Q}{V}$ , the transformation of Equation (10) gives  $\frac{dC}{\lambda C_{\text{eq}} - \lambda C + \frac{G}{V}} = dt$ .

Let  $t_0$  and  $C_0$  be the initial time and concentration values, respectively, and integrate both sides of Equation (11) simultaneously to obtain.

$$\int_{C_0}^c \frac{dC}{\lambda C_{\text{eq}} - \lambda C + \frac{G}{V}} = \int_{t_0} dt \quad (12)$$

Solve Equation (12) to obtain Equation (13).

$$C = C_{eq} + (C_0 - C_{eq})F + \frac{G}{\lambda V}(1 - F) \quad (13)$$

where  $F = \exp(-\lambda(t - t_0))$ .

### 3.2.1. Initialization

Similar to most heuristic algorithms, the initialization process of the balanced optimizer can be expressed as Equation (14).

$$C_i^{\text{init}} = C_{\min} + \text{randi}(C_{\min} - C_{\max}) \quad i = 1 \dots 2 \dots n \quad (14)$$

where  $C_i^{\text{init}}$  is the initial concentration vector of the  $i$ th individual:  $C_{\min}$  and  $C_{\max}$  are the lower and upper limit vectors of the individual,  $\text{randi}$  is a random vector between  $[0, 1]$ .

### 3.2.2. Establishing the Equilibrium Pool

The equilibrium state represents the ultimate state that the algorithm converges to. During the optimization process, the equilibrium pool serves as a source of candidate solutions for the entire optimization procedure. In our proposed method, we introduce a bootstrap optimization process. Specifically, the EO selects the top four individuals in terms of fitness from the equilibrium pool and calculates their average, creating a "fifth individual". Subsequently, one of these five individuals is randomly selected with equal probability to guide the rest of the optimization process, as demonstrated in Equation (15).

$$\vec{C}_{\text{eq. pool}} = \left\{ \vec{C}_{\text{eq} (1)}, \vec{C}_{\text{eq} (2)}, \vec{C}_{\text{eq} (3)}, \vec{C}_{\text{eq} (4)}, \vec{C}_{\text{eq} (\pi x)} \right\} \quad (15)$$

where  $\vec{C}_{\text{eq(ave)}} = \frac{C_{\text{eq}(1)} + C_{\text{eq}(2)} + C_{\text{eq}(3)} + C_{\text{eq}(4)}}{4} \mathbf{e}$ .

The probability of each of the five individuals in the equilibrium pool being selected as the solution for the bootstrap optimization process is identical, with all having a 0.2 chance.

The bootstrap optimization process plays a crucial role in enhancing the exploration and exploitation capabilities of the algorithm. Introducing randomness and diversity through the selection of individuals from the equilibrium pool helps to prevent the algorithm from getting stuck in local optima.

### 3.2.3. Exponential Terms

The exponential term plays a crucial role in the algorithm's update process and can be represented as Equation (16).

$$\vec{F} = e^{-\vec{\lambda}(t-t_0)} \quad (16)$$

where  $-\vec{\lambda}$  is a random vector between  $[0, 1]$ .

The variable  $t$  is defined as a function that diminishes with an increasing number of iterations, as illustrated in Equation (17).

$$t = \left( 1 - \frac{E}{\text{Maxit}} \right)^{a_2 \frac{E}{\text{Maxit}}} \quad (17)$$

where  $E$  and  $\text{Maxit}$  are the current iteration number and the maximum iteration number, respectively;  $a_2$  is a constant, generally taken as 1.

In order to guarantee the algorithm's convergence while simultaneously enhancing its search and exploitation capabilities, consider:

$$\vec{t}_0 = \frac{1}{\vec{\lambda}} \ln \left( -a_1 \text{sign}(\vec{r} - 0.5) \left[ 1 - e^{-\vec{\lambda}t} \right] \right) + t \quad (18)$$

where  $a_1$  is a constant, generally taken as 2;  $sign$  is a mathematical sign function;  $\vec{r}$  is a random vector between  $[0, 1]$ .

Bringing Equation (18) into Equation (16), we can obtain Equation (19).

$$\vec{F} = a_1 \text{sign}(\vec{r} - 0.5) \left[ e^{-\vec{\lambda}t} - 1 \right] \quad (19)$$

### 3.2.4. Generation Rate

The algorithm generation rate is characterized as a first-order exponential decay process, illustrated in Equation (20).

$$\vec{G} = \vec{G}_0 e^{-\vec{k}(t-t_0)} \quad (20)$$

In order to achieve a more controllable and systematic search pattern, the algorithm sets  $k = \lambda$  and incorporates the previously derived exponential term to describe the generation rate, as represented in Equation (21).

$$\vec{G} = \vec{G}_0 e^{-\vec{\lambda}(t-t_0)} = \vec{G}_0 \vec{F} \quad (21)$$

where

$$\vec{G}_0 = GCP \left( \vec{C}_{eq} - \vec{\lambda} \vec{C} \right) \quad (22)$$

$$GCP = \begin{cases} 0.5r_1 & r_2 \geq GP \\ 0 & r_2 < GP \end{cases} \quad (23)$$

where  $r_1$  and  $r_2$  are random numbers within the range  $[0, 1]$ ;  $GP$  is the generation probability, which is typically set to 0.5.

In summary, the final update formula for the balanced optimizer is defined in Equation (24).

$$\vec{C} = \vec{C}_{eq} + \left( \vec{C} - \vec{C}_{eq} \right) \cdot \vec{F} + \frac{\vec{G}}{\lambda V} \left( 1 - \vec{F} \right) \quad (24)$$

where the  $V$  value is generally taken as a constant 1.

In Equation (24), the first term represents the concentration at equilibrium, while the second and third terms characterize changes in concentration. Specifically, the second term enhances the algorithm's search capability by inducing significant changes in the individual close to the equilibrium state. Meanwhile, the third term improves the utilization capability by refining the obtained solution through minor adjustments in concentration.

### 3.2.5. Individual Memory Storage

Drawing inspiration from the concept of individual best in particle swarm optimization, the balance optimizer introduces an individual memory storage mechanism [21]. After  $E$  iterations (where  $E = 2$ ), the fitness value achieved by each individual is compared with the fitness value obtained after  $E-1$  selections. If the fitness value of the individual improves after the  $E$ -th iteration, both the individual's position and fitness value are updated accordingly. Otherwise, no update occurs, and the individual retains the position and fitness value obtained after the  $E-1$  selection for the next iteration. This mechanism primarily aims to enhance the algorithm's utilization capacity.

### 3.2.6. Enhanced Local Search Strategy Based on SA

In the literature, it is pointed out that EO has advantages, such as rapid convergence, but it also has the disadvantage of easily falling into local optima [22–25]. To address this issue, this section combines SA with EO and designs a local search.

SA is an optimization algorithm that seeks the global optimal solution in a complex search space. In SA, the temperature ( $T1$ ) is a crucial parameter that imitates the distribution of energy states of atoms within solid-state physics at various temperatures. This temperature parameter represents the likelihood of accepting suboptimal solutions in the search space. To explore the solution space more extensively, the algorithm begins with a higher initial temperature. As the number of iterations increases, the temperature is progressively reduced, consequently decreasing the probability of accepting an inferior solution. This procedure is known as “cooling” or “annealing”. By suitably adjusting the cooling rate and cooling function, the solution space can be thoroughly explored, ultimately converging to the global optimal solution. The following are the basic steps of the algorithm:

*Step 1:* Random initialization: determine an initial solution  $x$ , which is usually generated randomly, according to the characteristics of the problem.

Initial temperature setting: initialize the parameter  $T1$  (temperature) to a larger value in order to make the search process easier to jump out of the local minima.

*Step 2:* Iterative loop: for each temperature  $T1$ , a certain number of *subit* iterations are performed; each iteration starts from the current solution  $x$  to explore a new solution  $x_{new}$ .

*Step 3:* Random perturbation: according to the characteristics of the problem, some random perturbations are made to the existing solution  $x$  to obtain a new solution  $x_{new}$ .

*Step 4:* Evaluate the function: calculate the quality of  $x_{new}$ , i.e., estimate the value of the function to be solved  $f_{x_{new}}$ .

*Step 5:* Decision function: decide whether to accept the new solution  $x_{new}$  according to the metropolis criterion:

If  $f_{x_{new}} < f_x$ , then the new solution is accepted.

If  $f_{x_{new}} \geq f_x$ , then accept the new solution with probability  $p$ .

$$p = e^{-\delta e/t} \quad (25)$$

where  $\delta e = f(x_{new}) - f(x)$  is the energy difference,  $t$  is the temperature, and  $e$  is the natural constant.

*Step 6:* Temperature update: cooling according to certain cooling rules.

$$t_{k+1} = At_k \quad (26)$$

where  $A$  is the cooling rate, gradually reduce the temperature.

*Step 7:* Stopping condition: when the stopping condition is reached, i.e., the maximum number of iterations or the minimum temperature.

It should be noted that we set a random wandering strategy in this step to better improve the search performance, which is calculated as shown in Equation (27).

$$x'_i(l) = x_i(l) + \varepsilon(x_j(l) - x_k(l)) \quad (27)$$

where  $x'_i(l)$  is the updated new solution;  $x_j(l)$  and  $x_k(l)$  are two random solutions;  $\varepsilon$  is the scaling factor,  $\varepsilon \sim U(0, 1)$ ,  $U$  is uniformly distributed.

### 3.3. IEO-BP Model

To address the issues of weak self-adaptation and local minima in BP neural networks, we first employ IEO to globally pre-optimize the weights and thresholds of BP neural networks. Next, we assign these optimal weights and thresholds as initial values for BP neural networks and use the optimized parameters for training. This approach leads to the final fault BP neural network structure for early warning. The specific ISEO-BP process includes the following steps:

*Step 1:* Input neural network parameters, such as the number of hidden layer neurons, activation function, training times, training rate, and target error to be achieved during training.



*Step 2:* Input IEO algorithm parameters and use the *RMSE* of neural network prediction as the IEO fitness function. Execute the IEO algorithm process.

*Step 3:* Train the constructed BP neural network using the weights and thresholds obtained from IEO optimization, resulting in the optimized BP neural network structure.

*Step 4:* Input test data into the trained BP neural network to obtain output data. Perform data analysis on the output.

IEO-BP encompasses several crucial stages. Firstly, the neural network is set up by randomly allocating weights and biases to its neurons. Following this, input data traverses through the network, resulting in output via a combination of weighted and non-linear activation functions. The generated output is then compared to the actual labels to determine the error.

At this juncture, the IEO component is introduced, creating an initial solution for the optimization process. This solution experiences an iterative search procedure that assesses fitness values and selects novel candidate solutions. The derived solution is subsequently integrated into the BP neural network, serving as weights and thresholds to begin iterations.

The iteration cycle carries on until a pre-specified number of iterations are executed or the error is minimized to a satisfactory level. By employing this method, IEO-BP effectively merges the benefits of both BP neural networks and IEO algorithms, offering a powerful and efficient solution for equipment fault warning detection.

#### 4. Case Study

In this section, we first describe the case use (Sections 4.1 and 4.2), followed by an effective selection of the IEO parameters (Section 4.3), and finally, the training and testing of the IEO-BP network (Section 4.4).

##### 4.1. Case Description

The vapor feed pump is a widely used type of pump in various industries, primarily for supplying water to boilers and other equipment. It offers:

- **High Efficiency:** The pump quickly delivers water to the target equipment, ensuring a consistent and stable flow. This significantly improves water usage efficiency.
- **Reliability:** Allows for continuous operation, even under considerable loads and extended periods of use.
- **Continuity:** Equipped with a dual water supply system (electric or steam), the pump can continue functioning if one system encounters issues, thus guaranteeing continuity in the production process.

Due to the widespread application of vapor feed pumps, we have chosen this pump as a test example to evaluate the effectiveness of IEO-BP.

Following our investigation and analysis, we identified five primary fault types for steam feed pumps and their corresponding data anomalies.

Furthermore, we gathered 2800 sets of regular operation data from steam feed pumps and 30 sets of failure data for each fault type. The data collection relied on sensors, and the standard operation data encompassed a variety of pump performance metrics under different operating conditions, including pressure, flow rate, and temperature. We utilized this data to train an IEO-BP network model to recognize normal operating characteristics. The fault data encompass five primary fault types, among others, which are employed to assess the IEO-BP's ability to detect and provide early warning for these faults. By leveraging this comprehensive data for training and testing purposes, we could effectively evaluate the practicality and efficiency of the IEO-BP approach in real-world applications.

Our data collection methods are as follows:

1. On-site temperature and vibration sensors complete the data acquisition.
2. Collected data are sent to the control system.
3. The control system sends the data to the SIS system via the OPC data interface.
4. Access to the data from the SIS system for analysis and processing.

Regarding the sensors we utilized, temperature sensors typically transform temperature variations into electrical signals by altering thermocouple resistance. This process enables us to acquire temperature data. In this case, we have employed the widely used voltage divider circuit. Once the supply and voltage divider resistor R1 are established, we can determine the relationship between output voltage and temperature. We then choose an appropriate voltage divider resistor and calculate the corresponding voltage divider value for each temperature based on the Resistance-Temperature (R-T) table.

Vibration sensors primarily consist of three types: acceleration sensors, velocity sensors, and displacement sensors. These sensors measure vibration acceleration, vibration velocity, and vibration displacement, respectively.

As for pressure sensors, they convert pressure fluctuations into changes in resistance. However, since directly capturing resistance as a signal is challenging, we need to transform the resistance change into a voltage or current change. This conversion allows the acquisition card to collect data efficiently.

The Appendix A shows our collection of fault pictures.

#### 4.2. Description of Fault Types and Characteristics

We have gathered and identified five primary types of faults, as depicted in Table 1. The table presents the fault type in the left column, the anomaly measurement point during the fault in the middle column, and the abnormality type of the fault measurement point in the right column. Table 2 showcases a few examples of measurement points exhibiting anomalous data.

**Table 1.** Fault types and their measurement points.

Fault Types	Abnormal Measurement Point	Exception Type
High bearing vibration (H1): Causes parts to wear out or equipment to operate erratically	(1) Pump front-shaft vibration 1 abnormal (2) Pump front-shaft vibration 2 abnormal (3) Abnormal rear-shaft vibration 1 (4) Abnormal rear-shaft vibration 2	Abnormally large displacement amplitude
High temperature of thrust bearing (H2): Causes overheating of bearings and shortens service life	(1) Small steam engine thrust bearing temperature 1 abnormal (2) Small steam engine thrust bearing temperature 2 abnormal (3) Small steam engine thrust bearing temperature 3 abnormal (4) Small steam engine thrust bearing temperature 4 abnormal (5) Small steam engine thrust bearing temperature 5 abnormal (6) Small steam engine thrust bearing temperature 6 abnormal	Abnormally high temperature
Low lubricant oil pressure (H3): Affects the normal operation of the lubrication system and equipment performance	(1) Abnormal pressure in the lube oil supply bus of steam pump	Abnormally low pressure
High lubricant oil temperature (H4): Causes oil deterioration or insufficient cooling	(1) Abnormal temperature of lube oil supply bus bar of steam pump	Abnormally high temperature
Abnormal lubricant pressure (H5): Affects lubrication effect	(1) Abnormal air pump lubricant pressure	Abnormally high pressure

**Table 2.** Fault types and their abnormal data.

Fault Types	Normal Data	Abnormal Data
High bearing vibration (H1)	(1) 35.84 $\mu\text{m}$ (2) 31.72 $\mu\text{m}$ (3) 24.77 $\mu\text{m}$ (4) 21.31 $\mu\text{m}$	(1) 79.19 $\mu\text{m}$ (2) 76.01 $\mu\text{m}$ (3) 55.81 $\mu\text{m}$ (4) 55.1 $\mu\text{m}$
High temperature of thrust bearing (H2)	(1) 54.5 $^{\circ}\text{C}$ (2) 54.8 $^{\circ}\text{C}$ (3) 54.2 $^{\circ}\text{C}$ (4) 55.1 $^{\circ}\text{C}$ (5) 55.6 $^{\circ}\text{C}$ (6) 55.1 $^{\circ}\text{C}$	(1) 89.2 $^{\circ}\text{C}$ (2) 89.7 $^{\circ}\text{C}$ (3) 88.5 $^{\circ}\text{C}$ (4) 83.9 $^{\circ}\text{C}$ (5) 85.1 $^{\circ}\text{C}$ (6) 83.2 $^{\circ}\text{C}$
Low lubricant oil pressure (H3)	780 KPa	370 KPa
High lubricant oil temperature (H4)	42.5 $^{\circ}\text{C}$	56.2 $^{\circ}\text{C}$
Abnormal lubricant pressure (H5)	182 KPa	91 KPa

#### 4.3. IEO-BP Parameter Calibration

Appropriate parameters have a significant impact on algorithms [26,27]. In machine learning, many algorithms require certain parameters to be set in order to tune their behavior. These parameters can affect the performance of the model training process and the final performance of the model. If the parameters are not set properly, they may cause the algorithm's performance to degrade or even fail, resulting in the model not converging correctly. Therefore, proper parameter selection and tuning are required to ensure the best performance of the algorithm.

Before using IEO-BP for the training of the vapor feed pump network, we first adjusted its parameters, setting the number of BP network training to 1000, the training error to 0.02, and the learning rate to 0.001. We provided three reference values for each of the remaining parameters, as shown in Table 3, based on pre-experiments and literature analysis [12–14]. It is important to note that we chose *Softsign*, *Tanh*, and *ReLU* as activation functions. The *Softsign*, *Tanh*, and *ReLU* activation functions perform well in dealing with nonlinear problems. They are widely used activation functions in deep learning and neural networks and help to improve the performance of the models on various tasks. Here is a brief description of these three models:

**Table 3.** IEO-BP parameters and their reference values.

Input layer to implicit layer activation function ( <i>Fun1</i> )	<i>Softsign</i>	<i>Tanh</i>	<i>ReLU</i>
Implicit layer to output layer activation function ( <i>Fun2</i> )	<i>Softsign</i>	<i>Tanh</i>	<i>ReLU</i>
Number of neurons in the hidden layer ( <i>NL</i> )	10	11	12
<i>I</i>	0.95	0.92	0.90
<i>T</i>	10	15	20
<i>Maxit</i>	50	150	200
<i>Npop</i>	30	40	50
<i>Subit</i>	20	30	50
<i>T1</i> (Initial temperature)	1000	1500	1800
<i>T0</i> (Minimum temperature)	50	100	150
<i>A</i> (Cooling rate)	0.97	0.95	0.92

##### (1) Softsign:

*Softsign* functions are simpler and more efficient to compute than other S-shaped curves, such as Sigmoid and Tanh.

The output range  $(-1, 1)$  is useful to avoid excessively large or small output values in some scenarios.

(2) Tanh:

The *Tanh* function is symmetric with respect to the origin of the coordinates compared to the Sigmoid function, so it may have better performance in some applications.

Output range  $(-1, 1)$  alleviates the gradient vanishing problem.

(3) ReLU:

The *ReLU* function has good stability and low computational complexity when training deep neural networks.

*ReLU* has linear and nonlinear properties that help improve the expressiveness of the model and help it learn more complex mathematical functions.

Mitigates the gradient vanishing problem and helps converge faster.

It should be noted that all the codes were written in MATLAB 2018b software on an operating system using an (Intel) CI(TM) i7-10850H CPU @ 2.70 GHz, 2712 MHz, 6 Core(s) and 12 Logical Processor(s).

Based on the results in Table 3, conducting the full test would require a significant amount of resources. Therefore, we used the Taguchi test method to form an orthogonal array to conduct a reasonable number of tests. This method is based on the design principle of "orthogonal table", where multiple variables are combined and arranged so that each variable is tested at different levels. This maximizes the amount of useful data obtained and minimizes possible confounding factors.

We use the relative percentage deviation (*RPD*) to measure the performance of IEO-BP for each combination of parameters, which is calculated by Equation (28).

$$RPD = \frac{Alg_{Sol} - Min_{Sol}}{Min_{Sol}} \quad (28)$$

where  $Alg_{Sol}$  is the *RMSE* under the current parameter pair,  $Min_{Sol}$  is the minimum *RMSE* among all experimental times.

To effectively evaluate the performance of models with varying parameters, we employ K-fold Cross Validation, a widespread method for assessing machine learning model performance. This technique involves using a subset of the dataset for multiple training and validation iterations, ensuring the stability of evaluation results. The procedure includes the following steps:

1. Randomly divide the dataset into K disjoint subsets.
2. For each subset, execute the following steps:
  - Set the current subset as the validation set and merge the remaining K-1 subsets to form the training set.
  - Train the model with the training set.
  - Evaluate the model performance using the validation set and record the evaluation results.
3. Calculate the average of the K evaluation results to obtain the final performance evaluation metric of the model.

We select  $K = 5$  and use Root-Mean-Square Error (*RMSE*) as the evaluation metric. We record the average result after five runs with the current combination of parameters. As per the Taguchi method's recommendation, we utilize the L27 orthogonal array. The subsequent outcomes of the 27 experiments are presented in the Table 4.

We then used Equation (28) to calculate the *RPD* for each group of experiments; we select the mean *RPD* value of each parameter across all experiments to determine its optimal level. After calibration, the final parameter settings are displayed in Table 5.

**Table 4.** Cross-validation results.

Number of Experiments	RMSE	Number of Experiments	RMSE	Number of Experiments	RMSE
L1	73.22	L10	71.60	L19	70.33
L2	72.18	L11	74.15	L20	74.21
L3	70.97	L12	71.44	L21	73.68
L4	72.76	L13	72.50	L22	71.80
L5	71.88	L14	74.57	L23	71.36
L6	72.13	L15	70.04	L24	73.29
L7	73.10	L16	70.47	L25	70.89
L8	74.72	L17	71.19	L26	74.43
L9	73.07	L18	73.81	L27	72.90

**Table 5.** Calibration results of IEO-BP parameters.

<i>Fun1</i>	<i>softsign</i>
<i>Fun2</i>	<i>ReLU</i>
<i>NL</i>	10
<i>I</i>	0.92
<i>T</i>	20
<i>Maxit</i>	200
<i>Npop</i>	50
<i>Subit</i>	50
<i>T1</i>	1800
<i>T0</i>	100
<i>A</i>	0.97

#### 4.4. Network Training and Early Warning Testing

After crossover experiments and the Taguchi method to determine the optimum parameter levels, we first performed the network training of the IEO-BP model using the health sample data collected in Section 3.1. All data were divided into training and test sets according to an 8:2 ratio. The true values of the measured points and their predicted values under normal operation of these five main fault types are shown in Figure 1. Based on the true and predicted values, we can obtain our fault warning method, which is described in detail below. In addition, the RMSE convergence results for the training set and the RMSE convergence results for the test set are depicted in Figures 2 and 3, respectively, with the test sets observed every 20 iterations.

According to the results in Figure 1, our IEO-BP model demonstrates a good prediction effect; in addition, according to the results in Figures 2 and 3, the convergence of IEO-BP is also faster and, at the same time, more stable on the test sets, and we can propose a fault warning strategy based on the difference between the predicted and true values. The specific steps for implementing a fault warning strategy are as follows:

Step 1: Set the threshold value: Based on historical data analysis, establish a reasonable threshold value for the difference between predicted and actual observed values. This threshold should account for both normal equipment fluctuations and abnormal fluctuations that occur during faults.

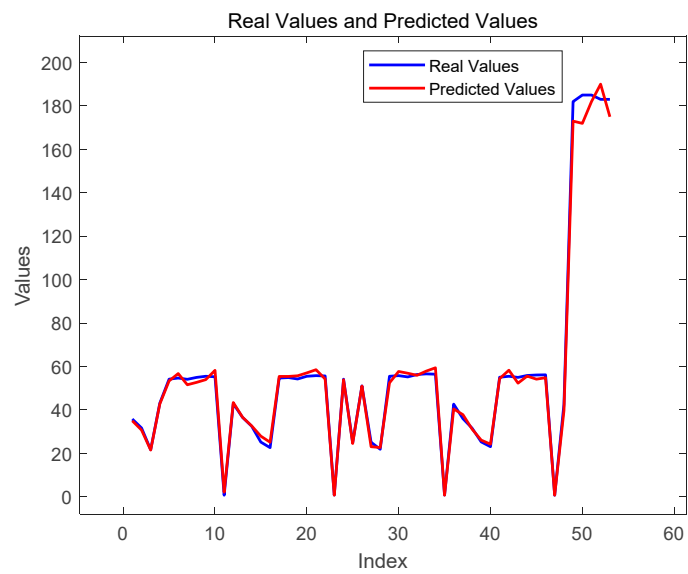


Figure 1. True and predicted values of measurement points.

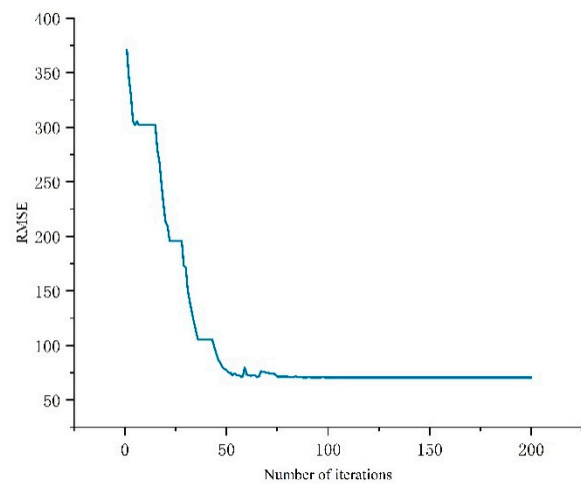


Figure 2. Training sets iterative network loss function graph.

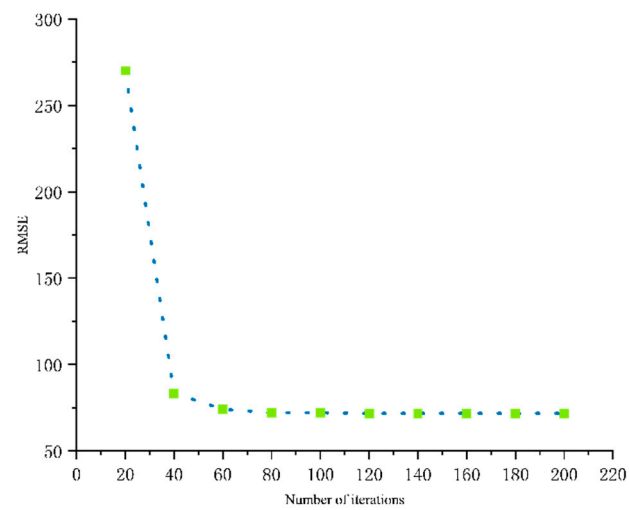


Figure 3. Test sets iterative network loss function graph.

Step 2: Real-time warning: Input real-time equipment data into the trained prediction model to obtain predicted values. Calculate the difference between the predicted and actual observed values. If the difference exceeds the predetermined threshold, issue a fault warning signal.

Step 3: Fault diagnosis and processing: Upon receiving the fault warning signal, conduct further inspection and diagnosis of the equipment. Depending on the diagnostic results, take appropriate measures to prevent or mitigate losses caused by equipment failure.

Step 4: Continuous optimization: Consistently collect equipment operation data and update the prediction model to maintain accuracy. Regularly evaluate the effectiveness of the warning strategy, adjust threshold settings, and make other necessary improvements.

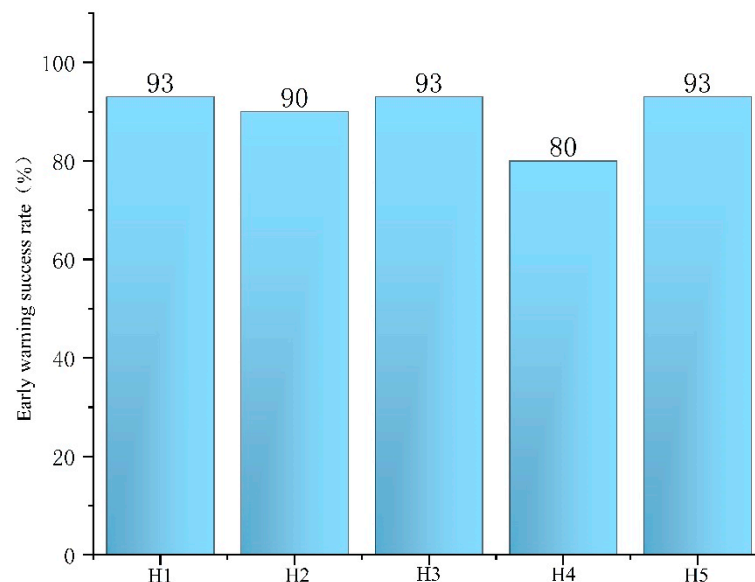
By employing this fault warning strategy based on the difference between predicted and actual values, abnormal equipment conditions can be detected and addressed promptly, thereby enhancing the operational efficiency, safety, and service life of the equipment.

This failure warning strategy can also be applied to other devices. The trained network can output point data, and if the output data deviates from the value set by the decision maker, a fault warning judgment can be triggered.

Subsequently, using the trained network and the proposed fault warning test, we conducted fault warning tests for five fault modes. The final test results are displayed in Table 6, demonstrating that IEO-BP can effectively achieve the purpose of fault warning. Its histogram is presented in Figure 4.

**Table 6.** IEO-BP fault test results.

H1	28/30
H2	27/30
H3	28/30
H4	24/30
H5	28/30



**Figure 4.** Histogram of fault warning test results.

It should be noted that the warning strategy we adopt involves issuing a warning immediately when the error between the predicted and true values exceeds the limit set by the decision maker during operation. Considering the randomness, the decision maker can either conduct troubleshooting immediately on this occasion or wait until the next warning.

Based on the above-mentioned experiments, IEO-BP can effectively achieve the purpose of fault warning, with its warning success rate for H1, H5, H3 reaching 93%, and for other problems being higher than 85%, which is within a reliable confidence range. However, we also note that its early warning accuracy for H4 is only 80%; thus, we need to strengthen the training of IEO-BP in this aspect. This issue arises from the settings of fault warning thresholds. Some fault data points may not be sensitive enough, causing deviations that do not reach the threshold. If the threshold is set too low, accuracy would increase, but it might also lead to false alarms. In actual practice, it is necessary to choose the threshold based on specific requirements and conditions.

## 5. Algorithm Performance Analysis

In this section, we first validate the effectiveness of the IEO improvement strategy (Section 5.1) and then demonstrate its performance in further comparison with other algorithms (Section 5.2).

### 5.1. IEO-BP Analysis

To verify the performance of the local search operator we set and the adaptive selection strategy of the BP neural network, we compared the convergence results of the test set with the RSME values; after removing the IEO-BP and SA strategies and removing the adaptive selection strategy, we set the number of iterations to 200, 500 and the initial population size to 30, 40, 50. Other parameters were the same as in the previous section, and fifteen trials were performed to take the average. The final results are shown in Table 7.

**Table 7.** IEO-BP performance analysis.

Algorithm Parameters	IEO-BP		IEO-BP Removal SA	
<i>Maxit, Npop</i>	<i>RMSE</i>	Number of First Convergence	<i>RMSE</i>	Number of First Convergence
30,200	73.21	120.35	75.96	123.72
30,500	72.98	303.23	75.09	320.53
40,200	73.56	108.76	76.01	126.89
40,500	72.66	248.52	74.98	267.36
50,200	71.03	108.76	73.26	135.86
50,500	70.36	282.99	72.76	297.55
Algorithm parameters	IEO-BP removal adaptive learning strategy			
30,200	75.72	130.88		
30,500	75.18	332.59		
40,200	74.92	135.69		
40,500	74.02	282.53		
50,200	73.03	140.52		
50,500	72.98	308.32		

According to the results in Table 7, we can see that the IEO-BP strategy outperforms the other two methods in different situations, achieving optimal values in terms of convergence speed and solution error, which fully demonstrates the effectiveness of these two strategies.

### 5.2. Comparison with Other Algorithms

To verify the algorithm's effectiveness, we chose three algorithms to solve the above cases simultaneously, namely, GA-BP [12], SVM-BP [28], and AFSA-BP [29]. Since the algorithms are randomized, we ran each algorithm fifteen times to ensure fairness. We also determined the optimal parameters of each algorithm according to the literature

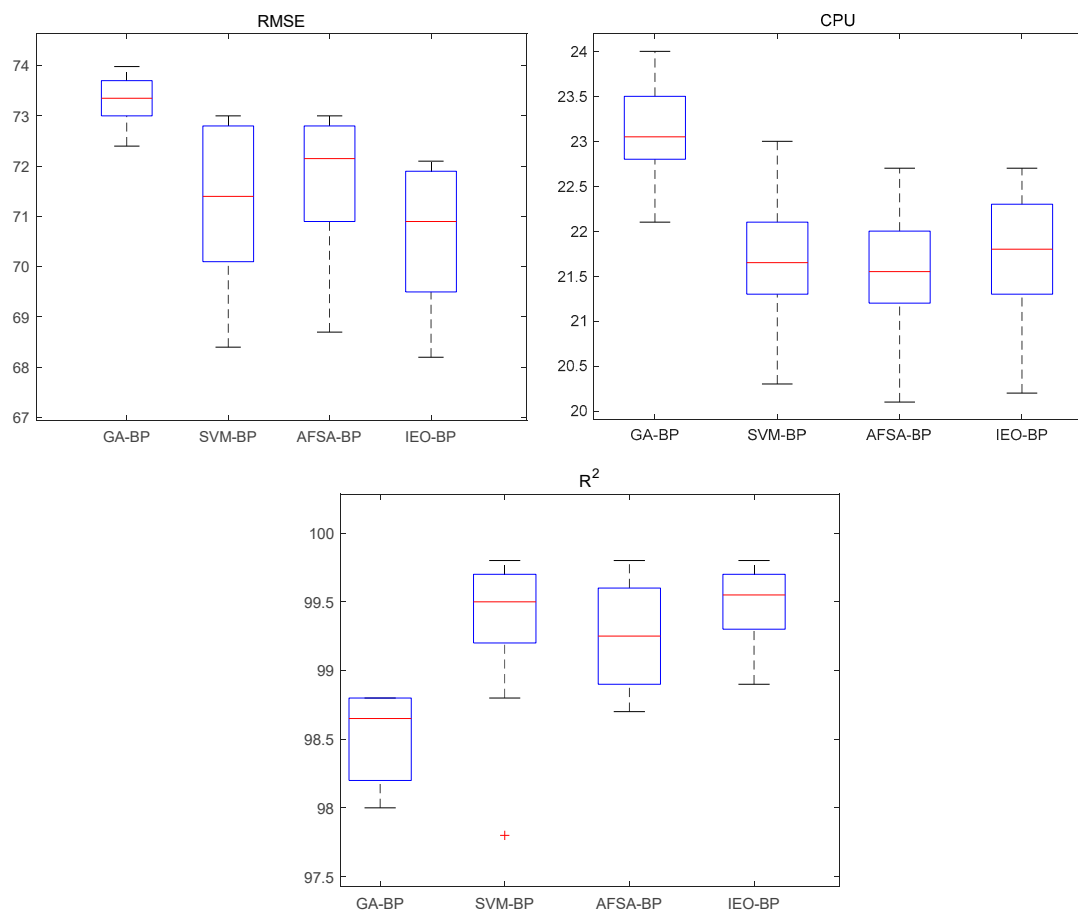


analysis [12,28,29] and integration of the cross-validated Taguchi experimental method. We compared the network training results using  $RMSE$ ,  $R^2$ , and the algorithm running time as evaluation indices and compared the performance of the trained network with the warning correct rate.  $RMSE$  is a measure of the difference between a model's predicted outcome and its actual value. It is obtained by calculating the average of the sum of the squares of the differences between the predicted and actual values, and then finding the square root.  $R^2$  is a statistical indicator used to measure the goodness of fit of a regression model. The larger the  $R^2$  value, the better the model fits the data; conversely, a smaller  $R^2$  value indicates that the model fits the data less well.  $CPU$  time measures the efficiency of the algorithm.

The average results of  $RMSE$ ,  $R^2$ , and the algorithm running time are shown in Table 8. Their fifteen boxplots of statistical results are displayed in Figure 5, and the results of the correct prediction rate are shown in Table 9.

**Table 8.** Algorithm comparison results.

Algorithms	$RMSE$	$R^2$	$CPU/s$
GA-BP	73.28	98.42	23.10
SVM-BP	71.21	99.30	21.66
AFSA-BP	71.70	99.15	21.55
IEO-BP	70.61	99.46	21.73



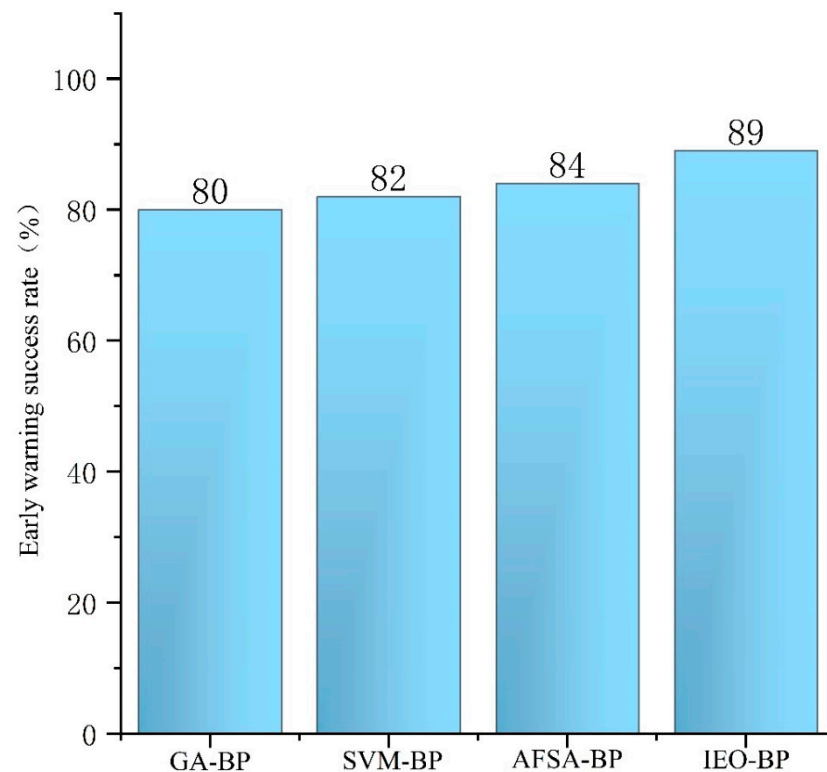
**Figure 5.** Statistical results of algorithm performance test (+ represents outliers).

**Table 9.** Fault warning test results.

GA-BP	120/150 (80%)
SVM-BP	123/150 (82%)
AFSA-BP	126/150 (84%)
IEO-BP	133/150 (89%)

Based on the results in Table 8, it can be concluded that our IEO-BP obtains optimal values for both metrics,  $RMSE$  and  $R^2$ , fully demonstrating its accuracy in terms of prediction. In terms of program running time, AFSA-BP has the lowest running time, while IEO-BP occupies a moderate position. From the results in Table 9, IEO-BP also achieves the highest level in terms of fault warning accuracy.

Additionally, the statistical results (Figure 6) show that GA-BP has the best stability in terms of  $RMSE$  among the three algorithms, with the shortest box plot length; however, its  $RMSE$  value is the highest. IEO-BP ranks second in terms of stability in this index. Although there is no significant difference among the three algorithms in terms of stability in terms of solution time and  $R^2$ , IEO-BP has a significantly better value.

**Figure 6.** Fault warning test performance comparison histogram.

In conclusion, compared to the other three algorithms, IEO-BP has certain advantages and can effectively achieve the fault warning goal.

## 6. Conclusions and Future Work

Fault warning is a reliable method for promoting the sustainable development of industrial equipment. Among various fault warning techniques, the BP neural network stands out as the most common and efficient approach. However, it has certain shortcomings. To enhance the efficiency of fault warning, this paper introduces a hybrid algorithm called IEO-BP. In IEO, we incorporate an SA-based random perturbation local search operator to effectively boost the exploration ability of the algorithm. For the BP neural

network, we add an adaptive learning rate to improve its prediction performance. Subsequently, we combine IEO with the improved BP neural network for fault warning analysis. Experimental results demonstrate that IEO-BP effectively achieves the fault warning objective, displaying notable advantages in comparison with other algorithms. In terms of performance comparison, our method achieved the best values for  $RMSE$  and  $R^2$ , with its solution efficiency in the middle of the range, thus striking a balance between efficiency and quality. In the fault warning test, its effectiveness improved by 11% compared to GA-BP, 8.5% compared to SVM-BP, and 6% compared to AFSA-BP, resulting in an average effectiveness improvement of 8.5%. Additionally, our proposed algorithm enhancement strategy demonstrates its effectiveness by exhibiting a faster convergence speed and higher solution accuracy compared to conventional EO.

Our research not only addresses the limitations of EO, but also expands its application area by integrating it with the improved BP neural network to propose a novel solution for fault warning issues, thereby fostering enhanced industrial development to meet contemporary demands.

Despite the successful investigation of a hybrid approach for fault warning, there remains ample room for future research. Firstly, fuzzy languages can be employed to represent uncertainties in the operation of realistic industrial equipment [2,3]. Secondly, IEO can be combined with other metaheuristics [30,31]. Lastly, our IEO-BP framework can be applied on various equipment according to practical requirements, or further improved to propose more sophisticated warning strategies; examples include the use of more adaptive learning rate expressions and the use of hybrid metaheuristics, in addition to extensions such as these to more neural network structures [32–36].

**Author Contributions:** Data curation, J.L., G.Z. and X.Z.; Methodology, C.Z.; Project administration, Z.M.; Writing—original draft, X.L.; Writing—review and editing, H.W. and S.Z. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Not applicable.

**Conflicts of Interest:** The authors declare no conflict of interest.

## Appendix A



High temperature of bearing (H2)



High bearing vibration (H1)



High bearing vibration (H1)

**Figure A1.** Cont.

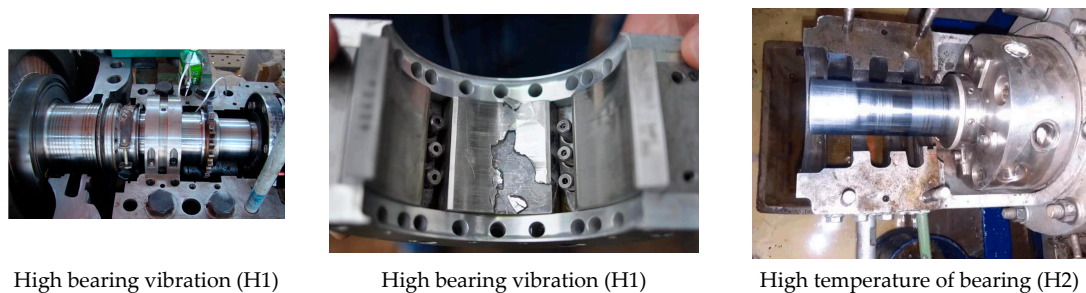


Figure A1. Failure mode graph.

## References

- Du, N.; Fathollahi-Fard, A.M.; Wong, K.Y. Wildlife resource conservation and utilization for achieving sustainable development in China: Main barriers and problem identification. *Environ. Sci. Pollut. Res.* **2023**, 1–20. [\[CrossRef\]](#)
- Tian, G.; Lu, W.; Zhang, X.; Zhan, M.; Dulebenets, M.A.; Aleksandrov, A.; Fathollahi-Fard, A.M.; Ivanov, M. A survey of multi-criteria decision-making techniques for green logistics and low-carbon transportation systems. *Environ. Sci. Pollut. Res.* **2023**, 1–23. [\[CrossRef\]](#) [\[PubMed\]](#)
- Tian, G.; Zhang, C.; Fathollahi-Fard, A.M.; Li, Z.; Zhang, C.; Jiang, Z. An enhanced social engineering optimiser for solving an energy-efficient disassembly line balancing problem based on bucket brigades and cloud theory. *IEEE Trans. Ind. Inform.* **2022**, *19*, 7148–7159. [\[CrossRef\]](#)
- Wang, Z. Fault early warning of wind turbine generator based on residual autoencoder network. In Proceedings of the 6th International Conference on High Performance Compilation, Computing and Communications, Kuala Lumpur, Malaysia, 22–24 March 2017; pp. 189–194.
- Jieyang, P.; Kimmig, A.; Dongkun, W.; Niu, Z.; Zhi, F.; Jiahai, W.; Liu, X.; Ovtcharova, J. A systematic review of data-driven approaches to fault diagnosis and early warning. *J. Intell. Manuf.* **2022**, 1–28. [\[CrossRef\]](#)
- Yuan, L.; Qiu, L.; Zhang, C. Research on Normal Behavior Models for Status Monitoring and Fault Early Warning of Pitch Motors. *Appl. Sci.* **2022**, *12*, 7747. [\[CrossRef\]](#)
- Zhao, H.; Liu, H.; Hu, W.; Yan, X. Anomaly detection and fault analysis of wind turbine components based on deep learning network. *Renew. Energy* **2018**, *127*, 825–834. [\[CrossRef\]](#)
- Wang, H.; Chen, J.; Zhu, X.; Song, L.; Dong, F. Early warning of reciprocating compressor valve fault based on deep learning network and multi-source information fusion. *Trans. Inst. Meas. Control* **2023**, *45*, 777–789. [\[CrossRef\]](#)
- Gao, D.; Wang, Y.; Zheng, X.; Yang, Q. A fault warning method for electric vehicle charging process based on adaptive deep belief network. *World Electr. Veh. J.* **2021**, *12*, 265. [\[CrossRef\]](#)
- Luo, Z.; Liu, C.; Liu, S. A novel fault prediction method of wind turbine gearbox based on pair-copula construction and BP neural network. *IEEE Access* **2020**, *8*, 91924–91939. [\[CrossRef\]](#)
- Wang, H.; Luan, L.; Rao, Y.; Yang, L.; Zhou, K.; Chen, J. Early warning of distribution transformer based on bp neural network considering the influence of extreme weather. In Proceedings of the 2021 IEEE 10th Data Driven Control and Learning Systems Conference (DDCLS), Suzhou, China, 14–16 May 2021; pp. 595–599.
- Chen, S.; Ma, Y.; Ma, L. Fault early warning of pitch system of wind turbine based on GA-BP neural network model. In *E3S Web of Conferences*; EDP Sciences: Ulis, France, 2020; Volume 194, p. 03005.
- Jiang, H.; Yu, Z.; Wang, Y.; Zhang, B.; Song, J.; Wei, J. The state prediction method of the silk dryer based on the GA-BP model. *Sci. Rep.* **2022**, *12*, 14615. [\[CrossRef\]](#)
- Zhang, L.; Gao, T.; Cai, G.; Hai, K.L. Research on electric vehicle charging safety warning model based on back propagation neural network optimized by improved gray wolf algorithm. *J. Energy Storage* **2022**, *49*, 104092. [\[CrossRef\]](#)
- Chen, H.; Li, S.; Li, M. Multi-Channel High-Dimensional Data Analysis with PARAFAC-GA-BP for Nonstationary Mechanical Fault Diagnosis. *Int. J. Turbomach. Propuls. Power* **2022**, *7*, 19. [\[CrossRef\]](#)
- Lin, J.; Zhao, Y.; Cui, B.; Li, Z. Fault diagnosis of active phase change control device based on SGSSA-BP neural network. In Proceedings of the 2022 IEEE International Conference on Artificial Intelligence and Computer Applications (ICAICA), Dalian, China, 24–26 June 2022; pp. 348–353.
- Wu, H.; Fu, W.; Ren, X.; Wang, H.; Wang, E. A Three-Step Framework for Multimodal Industrial Process Monitoring Based on DLAN, TSQTA, and FSN. *Processes* **2023**, *11*, 318. [\[CrossRef\]](#)
- Zhou, Y.; Kumar, A.; Parkash, C.; Vashishtha, G.; Tang, H.; Xiang, J. A novel entropy-based sparsity measure for prognosis of bearing defects and development of a sparsogram to select sensitive filtering band of an axial piston pump. *Measurement* **2022**, *203*, 111997. [\[CrossRef\]](#)
- Wolpert, D.H.; Macready, W.G. No free lunch theorems for optimization. *IEEE Trans. Evol. Comput.* **1997**, *1*, 67–82. [\[CrossRef\]](#)
- Zhang, W.; Han, G.; Wang, J.; Liu, Y. A BP neural network prediction model based on dynamic cuckoo search optimization algorithm for industrial equipment fault prediction. *IEEE Access* **2019**, *7*, 11736–11746. [\[CrossRef\]](#)

21. Jin, W.; Li, Z.J.; Wei, L.S.; Zhen, H. The improvements of BP neural network learning algorithm. In Proceedings of the WCC 2000—ICSP 2000. 2000 5th International Conference on Signal Processing Proceedings. 16th World Computer Congress 2000, Beijing, China, 21–25 August 2000; Volume 3, pp. 1647–1649.
22. Faramarzi, A.; Heidarinejad, M.; Stephens, B.; Mirjalili, S. Equilibrium optimizer: A novel optimization algorithm. *Knowl. Based Syst.* **2020**, *191*, 105190. [[CrossRef](#)]
23. Ouadfel, S.; Abd Elaziz, M. Efficient high-dimension feature selection based on enhanced equilibrium optimizer. *Expert Syst. Appl.* **2022**, *187*, 115882. [[CrossRef](#)]
24. Houssein, E.H.; Hassan, M.H.; Mahdy, M.A.; Kamel, S. Development and application of equilibrium optimizer for optimal power flow calculation of power system. *Appl. Intell.* **2023**, *53*, 7232–7253. [[CrossRef](#)]
25. Gupta, S.; Deep, K.; Mirjalili, S. An efficient equilibrium optimizer with mutation strategy for numerical optimization. *Appl. Soft Comput.* **2020**, *96*, 106542. [[CrossRef](#)]
26. Fathollahi-Fard, A.M.; Ahmadi, A.; Karimi, B. Sustainable and Robust Home Healthcare Logistics: A Response to the COVID-19 Pandemic. *Symmetry* **2022**, *14*, 193. [[CrossRef](#)]
27. Fathollahi-Fard, A.M.; Hajiaghahi-Keshteli, M.; Tavakkoli-Moghaddam, R.; Smith, N.R. Bi-level programming for home health care supply chain considering outsourcing. *J. Ind. Inf. Integr.* **2021**, *25*, 100246. [[CrossRef](#)]
28. Zheng, X.; Chen, F.; Guan, C.; Wang, R. Fault diagnosis of naval steam power system based on SVM-BP integrated learning. *Ship Sci. Technol.* **2023**, *45*, 97–101.
29. Chen, W.J.; Zhu, F.; Zhang, T.Y.; Zhang, J.; Zhang, F.M.; Xie, D.; Ru, W.; Song, M.; Fan, Q. AFSA-BP neural network based photovoltaic power prediction method. *Zhejiang Electr. Power* **2022**, *4*, 7–13. [[CrossRef](#)]
30. Fathollahi-Fard, A.M.; Hajiaghahi-Keshteli, M.; Tavakkoli-Moghaddam, R. The social engineering optimizer (SEO). *Eng. Appl. Artif. Intell.* **2018**, *72*, 267–293. [[CrossRef](#)]
31. Tian, G.; Zhang, C.; Zhang, X.; Feng, Y.; Yuan, G.; Peng, T.; Pham, D.T. Multi-Objective Evolutionary Algorithm with Machine Learning and Local Search for an Energy-Efficient Disassembly Line Balancing Problem in Remanufacturing. *J. Manuf. Sci. Eng.* **2023**, *145*, 051002. [[CrossRef](#)]
32. Ke, H.; Liu, H.; Tian, G. An uncertain random programming model for project scheduling problem. *Int. J. Intell. Syst.* **2015**, *30*, 66–79. [[CrossRef](#)]
33. Zhang, X.; Yang, F.; Guo, Y.; Yu, H.; Wang, Z.; Zhang, Q. Adaptive Differential Privacy Mechanism Based on Entropy Theory for Preserving Deep Neural Networks. *Mathematics* **2023**, *11*, 330. [[CrossRef](#)]
34. Tian, G.; Fathollahi-Fard, A.M.; Ren, Y.; Li, Z.; Jiang, X. Multi-objective scheduling of priority-based rescue vehicles to extinguish forest fires using a multi-objective discrete gravitational search algorithm. *Inf. Sci.* **2022**, *608*, 578–596. [[CrossRef](#)]
35. Minghui, H.; Ya, H.; Xinzhi, L.; Ziyuan, L.; Jiang, Z.; Bo, M.A. Digital twin model of gas turbine and its application in warning of performance fault. *Chin. J. Aeronaut.* **2023**, *36*, 449–470.
36. Liu, X.; Li, J.; Shao, L.; Liu, H.; Ren, L.; Zhu, L. Transformer Fault Early Warning Analysis Based on Hierarchical Clustering Combined with Decision Trees. *Energies* **2023**, *16*, 1168. [[CrossRef](#)]

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.