

Article



Integrating Improved Coati Optimization Algorithm and Bidirectional Long Short-Term Memory Network for Advanced Fault Warning in Industrial Systems

Kaishi Ji^{1,*}, Azadeh Dogani^{2,*}, Nan Jin¹ and Xuesong Zhang¹

- ¹ Independent Researcher, No. 76 Chongmingdao East Road, Huangdao District, Qingdao 266000, China
- ² Department of Agricultural Economics, Faculty of Agriculture, Ferdowsi University of Mashhad, Mashhad 9177948974, Iran
- * Correspondence: 15254206875@163.com (K.J.); azadeh.dogani@alumni.um.ac.ir (A.D.)

Abstract: In today's industrial landscape, the imperative of fault warning for equipment and systems underscores its critical significance in research. The deployment of fault warning systems not only facilitates the early detection and identification of potential equipment failures, minimizing downtime and maintenance costs, but also bolsters equipment reliability and safety. However, the intricacies and non-linearity inherent in industrial data often pose challenges to traditional fault warning methods, resulting in diminished performance, especially with complex datasets. To address this challenge, we introduce a pioneering fault warning approach that integrates an enhanced Coati Optimization Algorithm (ICOA) with a Bidirectional Long Short-Term Memory (Bi-LSTM) network. Our strategy involves a triple approach incorporating chaos mapping, Gaussian walk, and random walk to mitigate the randomness of the initial solution in the conventional Coati Optimization Algorithm (COA). We augment its search capabilities through a dual population strategy, adaptive factors, and a stochastic differential variation strategy. The ICOA is employed for the optimal selection of Bi-LSTM parameters, effectively accomplishing the fault prediction task. Our method harnesses the global search capabilities of the COA and the sophisticated data analysis capabilities of the Bi-LSTM to enhance the accuracy and efficiency of fault warnings. In a practical application to a real-world case of induced draft fan fault warning, our results indicate that our method anticipates faults approximately two hours in advance. Furthermore, in comparison with other advanced methods, namely, the Improved Social Engineering Optimizer Optimized Backpropagation Network (ISEO-BP), the Sparrow Particle Swarm Hybrid Algorithm Optimized Light Gradient Boosting Machine (SSAPSO-LightGBM), and the Improved Butterfly Optimization Algorithm Optimized Bi-LSTM (MSBOA-Bi-LSTM), our proposed approach exhibits distinct advantages and robust prediction effects.

Keywords: fault warning; improved coati optimization algorithm; bidirectional long short-term memory; industrial data analysis; predictive maintenance

1. Introduction

Currently, the rapid advancement of industrial automation and digitization has led to the widespread implementation of numerous industrial equipment and systems across various sectors [1–3]. The smooth operation of these equipment and systems is crucial for ensuring production efficiency, reducing maintenance costs, and safeguarding worker safety. However, these equipment and systems may fail due to various factors, resulting in decreased production efficiency and potential safety hazards [3]. Consequently, the development of effective methods for predicting and preventing potential equipment failures has emerged as a critical issue in the industrial sector.

To address this issue, many researchers have proposed various solutions. For example, a novel model for warning faults in wind turbine gearboxes was developed by Luo et al. [3]. Their approach, based on the Back Propagation Neural Network (BPNN), uses a conditional



Citation: Ji, K.; Dogani, A.; Jin, N.; Zhang, X. Integrating Improved Coati Optimization Algorithm and Bidirectional Long Short-Term Memory Network for Advanced Fault Warning in Industrial Systems. *Processes* 2024, *12*, 479. https:// doi.org/10.3390/pr12030479

Academic Editor: Wei Sun

Received: 17 January 2024 Revised: 16 February 2024 Accepted: 20 February 2024 Published: 27 February 2024



Copyright: © 2024 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/). mutual information program to select relevant variables for network training. Hao et al. [4] applied a metaheuristic algorithm that combines particle swarm optimization and sparrow optimization to optimize LightGBM parameters. This was specifically used for fault warning in thermal power plants. A unique solution for fault warning in rotor and ball bearing systems was introduced by Liu et al. [5]. They integrated Support Vector Machines and General Regression Neural Network (GRNN) for this purpose. A battery fault warning and positioning strategy based on sound signals was proposed by Lyu et al. [6]. They also introduced a false alarm prevention mechanism based on wavelet transform. The safety of oil and gas pipelines was the focus of Yang et al.'s research [7]. They developed a model using machine learning on distributed fiber optic sensor signals that can accurately detect and identify damage events in real-time. Wu et al. [8] proposed a comprehensive strategy that integrates a deep local adaptive network, dual-phase qualitative trend analysis, and a five-state Bayesian network. This strategy transforms abnormal variable continuous data into trend state data for fault detection, identification, and diagnosis, offering a new technical means for fault warning. A novel solution for fault warning in complex hydraulic machinery was proposed by Zhou et al. [9], who used an entropy-based sparse strategy, LSTM, and envelope analysis data to predict bearing defects and identify problems. Meanwhile, Chen et al. [10] provided a new solution for centrifugal pump fault detection by enhancing the accuracy of the BPNN using parallel factor decomposition and a genetic algorithm. Gao et al. [11] proposed a dynamic modeling strategy for early fault detection and identification by amalgamating wavelet packet decomposition and graph theory, the efficacy of which has been affirmed by experimental results. Lin et al. [12] introduced a new technical method for fault detection in active phase change control devices by employing an advanced Sparrow Search algorithm to upgrade the BPNN. Jing et al. [13] introduced a micro-service fault identification strategy based on LightGBM and analyzed historical operation information to ensure high availability. Zhao et al. [14] studied sensor data using a deep learning strategy, specifically a deep autoencoder network, to provide early warnings about potential faults in wind turbine components. Tan et al. [15] enhanced the SEO method in the field of wind turbines to optimize the BPNN, the performance of which surpasses other methods according to experimental results.

After reviewing the above literature, we find that although machine learning techniques have been widely applied in fault warning tasks, efficient strategies like Bi-LSTM are relatively less used in practical fault warning applications. Despite the efficiency advantages of Bi-LSTM, it faces challenges in the complexity of parameter selection, which may lead to extended diagnostic time, increased operational costs, and reduced warning accuracy. These issues need to be addressed through comprehensive research and improvement. The integration of metaheuristic algorithms and machine learning techniques has been proven to be an effective method to improve fault warning performance. This integration takes advantage of the global search superiority of metaheuristic algorithms and the powerful data modeling capabilities of machine learning, thus achieving remarkable performance in fault warning tasks. However, according to the "no free lunch" theorem in the optimization field [16], there is no universal algorithm that can solve all optimization problems, which encourages researchers to continuously develop and improve novel algorithms to adapt to the development of problems.

Based on the above analysis, to achieve more efficient fault warning results, we apply the novel metaheuristic algorithm COA to optimize the model parameters of Bi-LSTM for the first time. The powerful search capability and intuitive parameter adjustment of COA make it highly flexible in practical applications. In addition, we innovatively improve COA to enhance its search performance when dealing with complex problems, addressing the issue that COA may fall into local optima. We integrate a new initialization strategy, the bi-population optimization strategy, and new search operators into COA, designing a brand-new improved COA (ICOA). We use ICOA to select the parameters of Bi-LSTM, construct a fault prediction model, and implement it in a real induced draft fan case. By comparing with other advanced algorithms, we verify the effectiveness of our proposed method.

In summary, compared with previous research, the contributions of this paper are as follows:

- We improve the COA to enhance its search performance when dealing with complex optimization problems. And for the first time, we apply it to the parameter optimization of the Bi-LSTM, expanding its application field.
- We propose a new fault warning method, which combines the ICOA and Bi-LSTM, to improve the accuracy and efficiency of fault warning, and verify it through a real-world case.
- Through comparison with other advanced algorithms, we prove the effectiveness of our proposed method.

The rest of this paper is organized as follows. Section 2 provides a detailed introduction to our method, including the design and implementation of the ICOA and Bi-LSTM. Section 3 reports our experimental results and analysis. Section 4 validates the superiority of the proposed algorithm. Finally, Section 5 summarizes the main contributions of this paper and discusses future research directions.

2. Proposed Method

In this section, we first introduce the framework of the Bi-LSTM network (Section 2.1). Then, we discuss the components of our custom algorithm (Section 2.2). Next, we present the steps of the proposed ICOA-Bi-LSTM and provide its flowchart (Section 2.3).

2.1. Bi-LSTM Network

Long Short-Term Memory is indeed a unique form of Recurrent Neural Network (RNN) that addresses certain issues in traditional RNNs, such as short memory time, gradient vanishing, and gradient explosion, by adopting a chained forward propagation structure. The essence of LSTM lies in its three gate structures: the forget gate, the input gate, and the output gate [17,18]. These gates collectively enable the LSTM to effectively learn and remember information over long sequences, thereby mitigating the limitations of traditional RNNs.

Next, we will introduce these components in detail [17].

The function of the forget gate is to determine whether to discard the cell state information from the previous moment and what information to retain in the current cell. This process can be represented by Equation (1).

$$f_t = \sigma \Big(W_f[h_{t-1}, x_t] + b_f \Big) \tag{1}$$

where h_{t-1} is the output from the previous moment, x_t is the input at the current moment, W_f is the weight matrix of the forget gate, b_f is the bias matrix of the forget gate, σ is the Sigmoid function, and f_t is the output of the forget gate.

The input gate is responsible for generating the information to be updated. First, it determines the information to be updated through the Sigmoid layer, then generates the candidate memory unit value $\overline{C_t}$ at the current moment through the tan *h* layer, and finally calculates the updated cell state C_t by multiplying the cell state C_{t-1} from the previous moment with the forget vector point by point. The calculation process is shown in Equations (2)–(4).

$$i_t = \sigma(W_i[h_{t-1}, x_t] + b_i) \tag{2}$$

$$\overline{C_t} = \tanh\left(W_c[h_{t-1}, x_t] + b_C\right) \tag{3}$$

$$C_t = f_t C_{t-1} + i_t \overline{C_t} \tag{4}$$

where W_i is the weight matrix of the input gate, b_i is the bias matrix of the input gate, W_c is the weight matrix of the LSTM unit, and b_c is the bias matrix of the LSTM unit.

The output gate obtains the initial output result o_t through the Sigmoid layer, and then obtains the final output result h_t by multiplying the outputs obtained from the tanh layer and the Sigmoid layer pair by pair. The calculation process is shown in Equations (5) and (6).

$$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o) \tag{5}$$

$$h_t = o_t \tanh\left(C_t\right) \tag{6}$$

where W_o is the weight matrix of the output gate, and b_o is the bias matrix of the output gate.

Bidirectional Long Short-Term Memory adds a reverse LSTM layer to the LSTM network to handle reverse information. The forward LSTM is responsible for obtaining the past information of the input sequence, and the reverse LSTM is responsible for obtaining the future information of the input sequence. Therefore, Bi-LSTM can better mine data features, improve long-term dependency problems, and improve prediction accuracy.

In the Bi-LSTM network, the hidden layer state h_t at each level is composed of the output of the forward and reverse hidden units and the input amount x_t at the current moment. The combination process of the hidden layer state at each level is as follows:

$$\vec{h}_t = \text{LSTM}(x_t, \vec{h}_{t-1})$$
 (7)

$$\overleftarrow{h}_{t} = \text{LSTM}(x_{t}, \overleftarrow{h}_{t+1})$$
(8)

$$y_t = w_{\overrightarrow{hy}} \stackrel{\rightarrow}{h}_t + w_t \stackrel{\leftarrow}{h}_t + b_y \tag{9}$$

where \vec{h}_t and \vec{h}_t represent the hidden state output of LSTM at moments t - 1 and t + 1, \vec{h}_t and \vec{h}_t represent the hidden state output of LSTM at moment t, y_t is the output of BiLSTM, $w_{\vec{h}y}$ represents the connection weight from the forward LSTM layer to the output layer, $w_{\vec{h}y}$ represents the connection weight from the reverse LSTM layer to the output layer, and b_y represents the bias of the output layer.

A classical Bi-LSTM network is shown in Figure 1.



Figure 1. Bi-LSTM network.

2.2. Proposed ICOA

2.2.1. Population Initialization

In optimization algorithms, the choice of the initial population has a significant impact on the final optimization result [19]. Traditional COA usually adopts a random initialization method to generate the initial population. However, this method may lead to insufficient diversity in the population, thereby affecting the global search capability of the algorithm [19–21]. To overcome this drawback, we propose a new initialization method. Specifically, for each individual in the population, we first generate a random number x(i) between 0 and 1. Then, we use Sine mapping to convert this random number into a chaotic sequence. This process can be represented by Equation (10).

$$F_i = \sin(\pi x(i)) \tag{10}$$

where F_i is the Sine mapping value of the *x*th individual.

Afterwards, we use the generated chaotic sequence to form the initial individuals that meet the specified conditions. For each dimension of each individual, we use Equation (11) to generate a value between the value range b_i^{upper} and b_i^{lower} .

$$x_{i,j} = b_j^{lower} + F_i \left(b_j^{upper} - b_j^{lower} \right)$$
(11)

where $x_{i,j}$ represents the digit after mapping of the *j*th dimension of the *i*th individual, b_j^{upper} denotes the upper bound of the *j*th dimension, and b_j^{lower} represents the lower bound of the *j*th dimension.

Finally, we apply the Gaussian walk strategy and random walk strategy to further optimize the initial improved individuals. For each dimension of each individual, we generate a random number p. If $p \ge 0.5$, we choose to use the Gaussian walk strategy for optimization, as shown in Equations (12) and (13).

$$\Delta = N(0,1) \times min\left(\left(b_j^{upper} - x_{i,j}\right), \left(x_{i,j} - b_j^{lower}\right)\right)$$
(12)

$$x_{i,j} = x_{i,j} + \Delta \tag{13}$$

where N(0,1) represents a random number following the ecological distribution between 0 and 1.

If p < 0.5, we choose the random walk strategy for optimization, as shown in Equations (14) and (15).

$$\Delta = U(-1,1) \times min\left(\left(b_j^{\text{upper}} - x_{i,j}\right), \left(x_{i,j} - b_j^{\text{lower}}\right)\right)$$
(14)

$$x_{i,j} = x_{i,j} + \Delta \tag{15}$$

where U(-1,1) represents a uniform distribution between -1 and 1.

We merge the individuals improved by the chaotic mapping, Gaussian walk, and random walk strategies, and select the first *N* individuals as the initial population. Through this method, we can generate an initial population with good diversity, thereby improving the global search capability of the algorithm.

2.2.2. Subpopulation Division

Upon generating the initial population, we partition it into two subpopulations. The objective of this division strategy is to amplify the diversity of the population, thereby further enhancing the global search capability of ICOA. Each subpopulation will independently execute the subsequent optimization process. We initially sort according to the fitness of each individual (that is, their performance). Subsequently, we pair the best-performing individual (highest fitness) and the worst-performing individual (lowest fitness), and allocate this pair of individuals into the first subpopulation. Next, we pair the second-best performing individual and the second-worst performing individual, and allocate this pair of individuals into the second subpopulation. We repeat this pairing and assignment process until all individuals are assigned to the two subpopulations.

2.2.3. Global Search Phase

Once the initial population is divided into two subpopulations, we commence the update process of the coati population in the search space. The first phase of this process is to emulate the strategy when coatis attack iguanas, that is, the global search phase. In this

phase, a group of coatis ascend the tree to reach an iguana and frighten it, while a few other coatis wait under the tree until the iguana descends to the ground. After the iguana falls to the ground, the coatis attack and eliminate it. This strategy prompts the coatis to relocate to different positions in the search space, thereby indicating that the COA optimization algorithm possesses exploratory capabilities in the global search of the problem space. In the COA algorithm, it is postulated that the position of the best population member corresponds to the position of the iguana. There is a perspective that half of the coatis will climb the tree, while the other half will wait for the iguana to fall to the ground. Therefore, we can simulate the position through the following mathematical model [19]:

$$X_{i,j}^{\text{P1}} = x_{i,j} + r(G_j - Ix_{i,j}), \ i = 1, 2, \cdots, \left[\frac{N}{2}\right]$$
 (16)

where $X_{i,j}^{P1}$ is the new position of the *i*th coati in the *j*th dimension, *r* is a random number between 0 and 1, G_j is the position of the iguana in the *j*th dimension (which actually refers to the position of the best member), *I* is a number randomly selected from the set {1,2}, *N* is the number of coatis, and $\left[\frac{N}{2}\right]$ is the largest integer not exceeding $\frac{N}{2}$.

When the iguana falls, it is placed at a random position in the search space. Based on this random position, the coatis on the ground will move in the search space. This step can be simulated by Equations (17) and (18) [19].

$$G_j^{\rm g} = b_j^{lower} + r \left(b_j^{upper} - b_j^{lower} \right) \tag{17}$$

where G_i^{g} is the position of the iguana in the *j*th dimension on the ground.

$$x_{i,j}^{\text{Pl}} = \begin{cases} x_{i,j} + r \left(G_j^{\text{g}} - I x_{i,j} \right), & \text{if } F_G{}^g \leqslant F_i \\ x_{i,j} + r \left(x_{i,j} - G_j^{\text{g}} \right), & \text{otherwise} \\ i = [N/2] + 1, [N/2] + 2, \cdots, N \end{cases}$$
(18)

where $F_G{}^g$ is the objective function value of the iguana after falling to the ground, and F_i is the objective function value of the *i*th coati.

To guide the search process more effectively, avoid falling into local optima, improve search efficiency, and improve the quality of solutions, we improved the traditional COA formula and introduced a weight factor w based on the adaptive factor. The calculation of w is shown in Equation (19) [19].

$$w_{i,j} = w_{\min} + f_{i,j} \cdot (w_{\max} - w_{\min}) \tag{19}$$

where $w_{i,j}$ is the weight factor of the *i*th coati in the *j*th dimension, and $f_{i,j}$ is the adaptive factor of the *i*th coati in the *j*th dimension, which can be calculated by Equation (20).

$$f_{i,j} = \alpha \cdot \frac{X_{\text{best},j} - X_{\text{current},i,j}}{X_{\text{best},j} - X_{\text{worst},j}} + \beta \cdot \frac{X_{\text{best},j} - X_{\text{current},i,j}}{X_{\text{best},j}}$$
(20)

where $f_{i,j}$ is the adaptive factor of the *i*th coati in the *j*th dimension. $X_{\text{best},j}$ represents the position of the best solution in the *j*th dimension. $X_{\text{current},i,j}$ is the current position of the *i*th coati in the jth dimension. $X_{\text{worst},j}$ is the position of the worst solution value in the *j*th dimension. α and β are constants used to adjust the influence of the adaptive factor.

Afterwards, the search of ICOA at this stage is shown as Equations (21) and (22) [19].

$$X_{i,j}^{P1} = x_{i,j} + w_{i,j} \cdot r(G_j - Ix_{i,j}), \quad i = 1, 2, \cdots, \left[\frac{N}{2}\right]; \quad j = 1, 2, \cdots, m$$
(21)

$$x_{i,j}^{\text{Pl}} = \begin{cases} x_{i,j} + w_{i,j}r(G_j^{\text{g}} - Ix_{i,j}), & \text{if } F_G^{\text{g}} \leqslant F_i \\ x_{i,j} + w_{i,j}r(x_{i,j} - G_j^{\text{g}}), & \text{otherwise} \end{cases},$$

$$i = [N/2] + 1, [N/2] + 2, \cdots, N$$
(22)

In this way, we can effectively simulate the strategy of coatis attacking iguanas, thus achieving the effective exploration of the search space in the global search stage. This method not only increases the diversity of the population but also improves the global search capability of the ICOA algorithm.

Afterwards, we compare the updated individuals with the original individuals. If the updated individual is better, we update the current body; otherwise, we keep the status quo, as shown in Equation (23).

$$X_{i} = \begin{cases} X_{i}^{\text{P1}}, & \text{if } F_{i}^{\text{P1}} \leqslant F_{i} \\ X_{i}, & \text{otherwise} \end{cases}$$
(23)

where F_i^{P1} is the objective function value of the *i*th coati at the new position, and F_i is the objective function value of the *i*th coati at the previous position.

2.2.4. Local Search

The second phase of the process of updating the position of coatis in the search space is the process of evading predators, also referred to as the exploitation stage. This stage is rooted in the natural behavior of coatis when they encounter predators and escape from them. When a predator attacks a coati, the coati will flee from its current position. This strategy guides the coati to a safe location in close proximity to its current position, demonstrating the COA algorithm's capability in local search [19].

To simulate this behavior, we generate a random position near the position of each coati. Specifically, the local lower bound $b_{j,L}^{loc}$ and local upper bound $b_{j,U}^{loc}$ of each decision variable can be calculated using Equation (24) [19–21].

$$b_{j,\ lower}^{loc} = \frac{b_j^{lower}}{t}, \qquad b_{j,upper}^{loc} = \frac{b_j^{upper}}{t}, \qquad t = 1, 2, \cdots, T$$
(24)

where $b_{j, lower}^{loc}$ is the local lower bound of the *j*th decision variable, $b_{j, upper}^{loc}$ is the local upper bound of the *j*th decision variable, *t* is the iteration number, and *T* is the maximum number of iterations.

Afterwards, each individual can be updated according to Equation (25).

$$X_{i,j}^{P2} = x_{i,j} + (1 - 2r) \left(b_{j,lower}^{loc} + r \left(b_{j,upper}^{loc} - b_{j,lower}^{loc} \right) \right)$$

$$i = 1, 2, \cdots, N$$
(25)

where $X_{i,j}^{P2}$ represents the new position of the *i*th coati in the *j*th dimension.

If the updated individual is better, we update the current individual; otherwise, we keep the status quo, as shown in Equation (26).

$$X_{i} = \begin{cases} X_{i}^{P2}, & if \ F_{i}^{P2} \leqslant F_{i} \\ X_{i}, & otherwise \end{cases}$$
(26)

where F_i^{P2} is the objective function value of the *i*th coati at the new position, F_i is the objective function value of the *i*th coati at the previous position, and X_i is the original position of the *i*th coati.

2.2.5. Population Information Exchange

At certain intervals, we set two subpopulations to exchange information. The specific steps of this process are as follows:

Step 1: Sort all coatis in each subpopulation according to fitness, with the best coatis at the front.

Step 2: Select the best coati (elite position) in each subpopulation and save their position information.

Step 3: Pass the saved position information to another subpopulation. The elite position information of subpopulation 1 will be passed to subpopulation 2, and the elite position information of subpopulation 2 will be passed to subpopulation 1.

Step 4: After receiving the new elite position information, each subpopulation will take the position of its worst (*E*) coatis as candidate positions. Then, for each candidate position, we calculate its fitness. If this fitness is worse than the elite position's fitness, then we introduce a stochastic differential variance strategy to update it.

This process can be represented by Equation (27).

$$X_{e,j}^{P3} = r_1 \times \left(x_{elite,j} - x_{e,j} \right) + r_2 \times \left(x_{rand,j} - x_{e,j} \right)$$

$$e = 1, 2, \cdots, E$$
(27)

where $X_{e,j}^{P3}$ is the new position of the *j*th dimension of the *e*th individual after stochastic differential variation, r_1 and r_2 are random numbers between 0 and 1, $x_{elite,j}$ is the position of the *j*th dimension of the elite individual obtained from another population, and $x_{rand,j}$ is the position of the *j*th dimension of a random individual.

2.3. ICOA-Bi-LSTM Framework

Incorporating the above steps, the specific framework for ICOA to optimize the hyperparameters of Bi-LSTM is as follows:

Step 1: Conduct data preprocessing. Perform standardization of the data.

Step 2: Establish the basic parameters of ICOA and the scope of Bi-LSTM hyperparameters. Due to the numerous parameters in Bi-LSTM, we utilize ICOA to optimize two critical hyperparameters: the initial learning rate and the number of hidden layer nodes, aiming for a more accurate fault warning model.

Step 3: Initialize ICOA individuals. Employ the Root Mean Square Error (RMSE) as the fitness function.

Step 4: Sort the fitness values and divide the population into two subpopulations.

Step 5: Execute the individual update process of ICOA's global search phase.

Step 6: Perform the individual update process of ICOA's local search phase.

Step **7**: Carry out the information exchange process between the two subpopulations of ICOA.

Step 8: Check if the termination conditions are met. If not, return to Step 4; otherwise, output the optimal hyperparameter combination.

Note that in calculating all fitness values (i.e., RMSE), we implement k-fold cross-validation for the training of the Bi-LSTM model with k = 10. K-fold cross-validation is a technique used in machine learning to assess model performance. The fundamental idea is to divide the dataset into ten equal parts (folds), followed by ten rounds of model training and evaluation. The process involves the following:

Step 1: Divide the dataset into ten parts, typically through sequential division.

Step 2: Conduct ten rounds of iteration. In each round, utilize one part as the test set and the other nine parts as the training set. Train the model using the training set while evaluating its performance on the test set.

Step 3: Record the model's performance indicators on the test set for each round.

Step 4: After ten rounds, compute the average of all performance indicators to obtain a more robust performance evaluation.

Finally, the flowchart of the proposed method is shown in Figure 2.



Figure 2. ICOA-Bi-LSTM flowchart.

3. Case Study

In this section, we first introduce the case (Section 3.1), then train the ICOA-Bi-LSTM model (Section 3.2), and finally, apply the trained model to fault warning to demonstrate its effectiveness (Section 3.3).

3.1. Case Statement

In this research, we utilized the issue of induced draft fan overload in a specific power plant as a case study to validate our early warning model. Figure 3 shows some components of an induced draft fan.

We collected 1500 fan samples that operate under normal conditions for model training, and we divided the samples into a training set and a test set at a ratio of 70% and 30%, respectively. The initial input data measurement points encompassed nine measurements: front bearing vibration, rear bearing vibration, inlet pressure, inlet and outlet differential pressure, inlet temperature, guide vane opening, speed, motor phase coil temperature, and bearing temperature. The main motor current of the induced draft fan served as the output data.



Figure 3. Image of some components of an induced draft fan (**a**). Air intake box large shaft; (**b**). Dynamic lobe actuator; (**c**). Shaft coupling.

3.2. Prediction Result

Given the high dimensionality of the input data, we employed the Pearson correlation coefficient to select measurement points that exhibit a high correlation with the main motor current of the induced draft fan [22,23]. These points are used as the inputs for the early warning model. This strategy is designed to prevent model overfitting and achieve data dimensionality reduction. The calculation formula for the Pearson correlation coefficient is shown in Equation (28).

$$r = \frac{\sum_{i=1}^{n} (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n} (x_i - \bar{x})^2 \sum_{i=1}^{n} (y_i - \bar{y})^2}}$$
(28)

where x_i represents the value of a certain parameter x in the *i*th sample among the input variables of operating parameters. y_i represents the value of the main motor current y of the induced draft fan in the *i*th sample. \overline{x} is the average value of the x samples among the n parameters. \overline{y} is the average value of the main motor current variable in the n samples. r is the correlation coefficient between the parameter x and the main motor current of the induced draft fan. The larger the absolute value of r, the higher the correlation between the variables. We report the mapping situation of its r value and correlation in Table 1.

Subsequently, we obtained the top four most relevant measurement points. We use these as the inputs for the model, as shown in Table 2.

Based on preliminary experiments and literature analysis [19–21], we set T = 200, N = 50, $\alpha = 0.3$, $\beta = 0.7$, E = 6, $w_{\text{max}} = 0.8$, and $w_{\text{min}} = 0.2$. Regarding the optimization space of Bi-LSTM, we set the learning rate to be within [0.001, 1], and the number of neurons in the hidden layer to be within [1, 100]. After that, we train the model and output the prediction results of the test set, as shown in Figure 4.

r Value	Correlation
$ r \ge 0.95$	Significant correlation
$0.95 > r \ge 0.8$	Strongly correlated
$0.8 > r \ge 0.5$	Moderately correlated
$0.5 > r \ge 0.3$	Weakly correlated
$ r \le 0.3$	Uncorrelated

Table 2. Top four relevant measurement points.

Front bearing vibration	0.958	
Rear bearing vibration	0.946	
Blower current	0.938	
Lower motor phase coil temperature	0.932	



Figure 4. Comparison of true and predicted values.

To more reasonably determine the warning threshold, on the test set, we calculated the residuals and used the sliding window method to set the fault warning threshold. The sliding window method can detect minor changes in system behavior in advance, thereby issuing a warning before a fault occurs to avoid or mitigate the loss caused by the fault [24]. Specifically, supposing the residual sequence within a certain period is $[X_1, X_2, ..., X_m]$, we selected a sliding window with a width of n ($n \le m$) and calculated the average residual in this window, as shown in Equation (29).

$$\overline{X} = \frac{1}{n} \sum_{i=1}^{n} X_i \tag{29}$$

where \overline{X} is the average residual under this window, and X_i is the residual sample in this window.

We set the number of sliding windows as 30, and the step size as 1. Finally, as depicted in Figure 5, the residuals after processing by the sliding window method are shown. It can be observed that all values lie between -1 and 0.6. Consequently, we established the upper and lower limits of the warning to 0.6 and -1, respectively. This implies that when we conduct real-time detection through the online sliding window method, a warning is triggered if the residual surpasses 0.6 or falls below -1. This approach ensures timely alerts for any potential anomalies detected by the system.



Figure 5. Residuals and warning thresholds after sliding window processing.

3.3. Warning Instance

We selected 500 samples before and after a specific overload fault of induced draft fan B of Unit 5 in a power plant in Shanghai as the test set. The SIS system of the power plant issued a fault alarm for this equipment around 12:00 on 20 June 2022. We also used the sliding window method to determine if a fault had occurred. The step size of the sliding window method was set to 1, and the width of the sliding window was 20. We input these 500 samples before and after the fault into the well-trained fault warning model after data preprocessing. The final results are shown in Figure 6.



Figure 6. Early warning model residuals and alarm triggers analysis.

As depicted in Figure 6, we observed that at the 441st sample, the residuals of the early warning model exceeded the threshold of 0.6, triggering an alarm. This alarm corresponds to the SIS sampling time of 10:06 on 20 June 2022, approximately 2 h ahead of the actual alarm point. This result demonstrates that our early warning model can effectively predict early failures of the power plant's induced draft fan, which is of significant importance in preventing or mitigating the losses caused by failures. This further validates the effectiveness and practicality of our method.

4. Algorithm Performance Analysis

Here, to validate the effectiveness of the algorithm, we compare the proposed algorithm with ISEO-BP [15], SSAPSO-LightGBM [4], and MSBOA-Bi-LSTM [25]. These algorithms have demonstrated excellent performance in the field of fault warning and have been verified by real-world industrial cases. By comparing them, we can more comprehensively understand the performance and advantages of our algorithm. We compare the three algorithms using three well-known model evaluation indicators, namely RMSE, Mean Absolute Error (MAE), and CPU time. Due to the randomness of the algorithm, we run each algorithm ten times and take its average value, as shown in Table 3. In addition, we use a ninety-five percent confidence interval to show its statistical results, as shown in Figure 7. According to the results in Table 3, the proposed algorithm has advantages in RMSE and MAE. However, it should be noted that due to the division of the population and the three-population initialization strategy, its CPU time increases. But the improvement in solution quality it brings is enough to offset this disadvantage. In addition, according to the 95% confidence interval, the proposed algorithm has the most stable effect on the three indicators.



Table 3. Comparative performance of algorithms.

Figure 7. Algorithm performance comparison with 95% confidence interval (**a**). RMSE comparison results; (**b**). MAE comparison results; (**c**). CPU comparison results.

5. Conclusions and Future Work

As the imperative for secure equipment operation and heightened production efficiency in industrial settings continues to escalate, the significance of fault warning has become increasingly paramount. The advent of machine learning technology has revolutionized fault warning, with the Bi-LSTM network emerging as a crucial algorithm to enhance its efficacy. However, the challenge lies in effectively determining the parameters that optimize its warning performance.

To address this challenge, we introduced a novel fault warning algorithm, namely ICOA-Bi-LSTM. Initially, we refined the COA and tailored the ICOA by implementing a triple strategy involving chaotic mapping, Gaussian walk, and random walk to overcome the inherent randomness of the initial solution in the traditional COA. Subsequently, we augmented its search capabilities through the integration of multiple improved search operators. Finally, the ICOA was employed for the optimal selection of Bi-LSTM parameters, successfully completing the fault warning task.

In a practical application involving induced draft fan fault warning, our method outperformed other advanced approaches in terms of fault prediction accuracy and generalization ability. This underscores the effectiveness of our solution in timely problem detection, risk mitigation for production interruptions, and reliable support for industrial production.

However, recognizing the substantial achievements in fault warning within this study, we acknowledge the vast exploration space that remains. Future endeavors could involve integrating our ICOA-Bi-LSTM strategy with other machine learning algorithms, such as support vector machines or decision trees, to bolster the model's robustness and predictive power [26,27]. Additionally, exploring ensemble learning methods like random forests or gradient boosting may further elevate the model's performance [28,29]. Lastly, the development of new optimization strategies for more effective parameter adjustments, including advanced hybrid meta-heuristic algorithms or the incorporation of additional optimization operators, presents exciting avenues for future research [30,31].

Author Contributions: Methodology, N.J.; Validation, K.J.; Formal analysis, N.J. and X.Z.; Investigation, K.J.; Writing—original draft, K.J.; Writing—review & editing, A.D., N.J. and X.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Data are contained within the article.

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

References

- Lu, G.; Wen, X.; He, G.; Yi, X.; Yan, P. Early fault warning and identification in condition monitoring of bearing via wavelet packet decomposition coupled with graph. *IEEE/ASME Trans. Mechatron.* 2021, 27, 3155–3164. [CrossRef]
- Li, J.; Liu, J.; Chen, Y. A fault warning for inter-turn short circuit of excitation winding of synchronous generator based on GRU-CNN. *Glob. Energy Interconnect.* 2022, 5, 236–248. [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]
- 4. Hao, Z.; Da, M.; Yu, L. A Fault Warning Method for Power Plant Induced Draft Fans Based on SSAPSO-Light GBM. *Therm. Power Eng.* **2023**, *2*, 153–160. [CrossRef]
- 5. Chu, W.L.; Lin, C.J.; Kao, K.C. Fault diagnosis of a rotor and ball-bearing system using DWT integrated with SVM, GRNN, and visual dot patterns. *Sensors* **2019**, *19*, 4806. [CrossRef] [PubMed]
- 6. Lyu, N.; Jin, Y.; Miao, S.; Xiong, R.; Xu, H.; Gao, J.; Liu, H.; Li, Y.; Han, X. Fault warning and location in battery energy storage systems via venting acoustic signal. *IEEE J. Emerg. Sel. Top. Power Electron.* **2021**, *11*, 100–108. [CrossRef]
- Yang, Y.; Li, Y.; Zhang, H. Pipeline safety early warning method for distributed signal using bilinear CNN and Light GBM. In Proceedings of the ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Toronto, ON, Canada, 6–11 June 2021; IEEE: Piscataway, NJ, USA, 2021; pp. 4110–4114.
- 8. Wu, H.; Fu, W.; Ren, X.; Wang, H.; Wang, E. A Three-Step Framework for Multimodal Industrial Process Monitoring Based on DLAN, TSQTA, and FSBN. *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]
- 10. 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]
- 11. 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]
- 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; IEEE: Piscataway, NJ, USA, 2022; pp. 348–353.
- Jing, N.; Li, H.; Zhao, Z. A microservice fault identification method based on LightGBM. In Proceedings of the 2022 IEEE 8th International Conference on Cloud Computing and Intelligent Systems (CCIS), Chengdu, China, 26–28 November 2022; IEEE: Piscataway, NJ, USA, 2022; pp. 709–713.
- 14. 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]
- 15. Tan, Y.; Zhan, C.; Pi, Y.; Zhang, C.; Song, J.; Chen, Y.; Golmohammadi, A.M. A Hybrid Algorithm Based on Social Engineering and Artificial Neural Network for Fault Warning Detection in Hydraulic Turbines. *Mathematics* **2023**, *11*, 2274. [CrossRef]
- 16. Wolpert, D.H.; Macready, W.G. No free lunch theorems for optimization. IEEE Trans. Evol. Comput. 1997, 1, 67-82. [CrossRef]
- 17. Smagulova, K.; James, A.P. A survey on LSTM memristive neural network architectures and applications. *Eur. Phys. J. Spec. Top.* **2019**, *228*, 2313–2324. [CrossRef]
- 18. Suebsombut, P.; Sekhari, A.; Sureephong, P.; Belhi, A.; Bouras, A. Field data forecasting using LSTM and Bi-LSTM approaches. *Appl. Sci.* **2021**, *11*, 11820. [CrossRef]
- 19. Dehghani, M.; Montazeri, Z.; Trojovská, E.; Trojovský, P. Coati Optimization Algorithm: A new bio-inspired metaheuristic algorithm for solving optimization problems. *Knowl.-Based Syst.* **2023**, 259, 110011. [CrossRef]
- Ren, Y.; Zhang, C.; Zhao, F.; Xiao, H.; Tian, G. An asynchronous parallel disassembly planning based on genetic algorithm. *Eur. J.* Oper. Res. 2018, 269, 647–660. [CrossRef]
- Hashim, F.A.; Houssein, E.H.; Mostafa, R.R.; Hussien, A.G.; Helmy, F. An efficient adaptive-mutated Coati optimization algorithm for feature selection and global optimization. *Alex. Eng. J.* 2023, *85*, 29–48. [CrossRef]
- 22. Asuero, A.G.; Sayago, A.; González, A.G. The correlation coefficient: An overview. *Crit. Rev. Anal. Chem.* 2006, 36, 41–59. [CrossRef]
- Egghe, L.; Leydesdorff, L. The relation between Pearson's correlation coefficient r and Salton's cosine measure. J. Am. Soc. Inf. Sci. Technol. 2009, 60, 1027–1036. [CrossRef]
- Zhang, L.; Lin, J.; Karim, R. Sliding window-based fault detection from high-dimensional data streams. *IEEE Trans. Syst. Man. Cybern. Syst.* 2016, 47, 289–303. [CrossRef]
- Jie, N.; Qing, H. Improved Butterfly Optimization Algorithm with Hybrid Strategy. Res. Comput. Appl. 2021, 6, 1718–1723+1738. [CrossRef]
- Mousapour Mamoudan, M.; Ostadi, A.; Pourkhodabakhsh, N.; Fathollahi-Fard, A.M.; Soleimani, F. Hybrid neural network-based metaheuristics for prediction of financial markets: A case study on global gold market. J. Comput. Des. Eng. 2023, 10, 1110–1125. [CrossRef]
- Gholizadeh, H.; Fathollahi-Fard, A.M.; Fazlollahtabar, H.; Charles, V. Fuzzy data-driven scenario-based robust data envelopment analysis for prediction and optimisation of an electrical discharge machine's parameters. *Expert. Syst. Appl.* 2022, 193, 116419. [CrossRef]
- Ghazikhani, A.; Babaeian, I.; Gheibi, M.; Hajiaghaei-Keshteli, M.; Fathollahi-Fard, A.M. A smart post-processing system for forecasting the climate precipitation based on machine learning computations. *Sustainability* 2022, 14, 6624. [CrossRef]
- 29. Fathollahi-Fard, A.M.; Wong, K.Y.; Aljuaid, M. An efficient adaptive large neighborhood search algorithm based on heuristics and reformulations for the generalized quadratic assignment problem. *Eng. Appl. Artif. Intell.* **2023**, *126*, 106802. [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]
- 31. Zhang, X.; Zhou, H.; Fu, C.; Mi, M.; Zhan, C.; Pham, D.T.; Fathollahi-Fard, A.M. Application and planning of an energy-oriented stochastic disassembly line balancing problem. *Environ. Sci. Pollut. Res.* **2023**. [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.