processes

Article

Efficient Fault Warning Model Using Improved Red Deer Algorithm and Attention-Enhanced Bidirectional Long Short-Term Memory Network

Yutian Wang [1,2] and Mingli Wu [1,*]

[1] School of Electrical Engineering, Beijing Jiaotong University, Beijing 100044, China
[2] Information Analysis and Diagnosis Center, Guoneng Guohua (Beijing) Cogeneration Power Co., Ltd., Beijing 100018, China
* Correspondence: mlwu@bjtu.edu.cn

Abstract: The rapid advancement of industrial processes makes ensuring the stability of industrial equipment a critical factor in improving production efficiency and safeguarding operational safety. Fault warning systems, as a key technological means to enhance equipment stability, are increasingly gaining attention across industries. However, as equipment structures and functions become increasingly complex, traditional fault warning methods face challenges such as limited prediction accuracy and difficulties in meeting real-time requirements. To address these challenges, this paper proposes an innovative hybrid fault warning method. The proposed approach integrates a multi-strategy improved red deer optimization algorithm (MIRDA), attention mechanism, and bidirectional long short-term memory network (BiLSTM). Firstly, the red deer optimization algorithm (RDA) is enhanced through improvements in population initialization strategy, adaptive optimal guidance strategy, chaos regulation factor, and double-sided mirror reflection theory, thereby enhancing its optimization performance. Subsequently, the MIRDA is employed to optimize the hyperparameters of the BiLSTM model incorporating an attention mechanism. A predictive model is then constructed based on the optimized Attention-BiLSTM, which, combined with a sliding window approach, provides robust support for fault threshold identification. The proposed algorithm's efficacy is demonstrated through its application to real-world gas-fired power plant equipment fault cases. Comparative analyses with other advanced algorithms reveal its superior robustness and accuracy in efficiently issuing fault warnings. This research not only provides a more reliable safeguard for the stable operation of industrial equipment but also pioneers a new avenue for the application of metaheuristic algorithms.

Keywords: fault warning; multi-strategy improved red deer algorithm; attention mechanism; bidirectional long short-term memory network; industrial equipment stability

check for updates

Citation: Wang, Y.; Wu, M. Efficient Fault Warning Model Using Improved Red Deer Algorithm and Attention-Enhanced Bidirectional Long Short-Term Memory Network. *Processes* **2024**, *12*, 2253. https://doi.org/10.3390/pr12102253

Academic Editor: Weizhong Dai

Received: 15 September 2024
Revised: 4 October 2024
Accepted: 8 October 2024
Published: 15 October 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/).

1. Introduction

In the contemporary era, technological advancements have not only significantly propelled industrial development but have also led to an increasing complexity in industrial equipment [1,2]. This complexity poses unprecedented demands for precision in operation and production efficiency while simultaneously amplifying the dynamism and uncertainty of the production environment. Against this backdrop, ensuring the reliability and stability of industrial equipment has become particularly crucial [3]. Equipment failure can not only disrupt production processes but may also lead to a decline in product quality and even trigger severe safety incidents. Therefore, fault warning systems play an indispensable role in modern industry. They monitor equipment status in real-time, predict potential faults, and thus facilitate predictive maintenance [4].

Traditional fault warning methods mainly rely on statistical analysis. While these methods have played an important role in the past, they show significant limitations when

dealing with complex and high-dimensional industrial data [5,6]. Specifically, traditional methods often struggle to capture deep features and potential trends when faced with large-scale, multivariable, and high-noise industrial data, leading to insufficient warning effectiveness.

With the rapid development of computing technology and the advent of the big data era, machine learning models, especially deep learning algorithms, have gradually become prominent solutions in the field of fault warning [6]. These models have shown encouraging application results in various industrial scenarios, significantly improving the accuracy and real-time performance of fault warnings. These advanced algorithms can handle and analyze massive multidimensional data that traditional methods find difficult to manage, by learning hidden patterns in the data to achieve precise monitoring of equipment operating status. Additionally, they possess self-adaptive and evolving capabilities, with the accuracy of fault detection increasing as more data is accumulated and models are continuously iterated. This helps in the timely identification of anomalies, reducing false alarms and missed detections. Furthermore, they can integrate data from different sensors, providing cross-domain insights that enhance the comprehensiveness and depth of fault warnings. They can also continuously monitor system health without human intervention, thereby improving operational efficiency and reducing labor costs [7–11].

Among various deep learning algorithms, bidirectional long short-term memory (BiLSTM) networks have shown particularly excellent performance in fault warning systems. As a special type of recurrent neural network, BiLSTM networks' unique bidirectional nature allows for forward and backward processing of time series data, thereby capturing contextual information more comprehensively. This bidirectional structure enables BiLSTM networks to fully utilize past input information and effectively leverage future input information, leading to more accurate identification and prediction of early fault signals. This characteristic holds great potential in fault prediction scenarios, especially when considering data trend changes over long time spans [9].

However, most current research remains at the application level of basic LSTM architectures and has not fully explored the unique advantages of BiLSTM networks. As the data generated by industrial systems becomes increasingly complex, issues such as complex time dependencies and noise patterns become more prominent. Traditional LSTM models may struggle to handle these complex data features, resulting in decreased prediction accuracy and robustness. Therefore, further in-depth exploration of the BiLSTM method to develop a new approach that can more efficiently and accurately capture early fault signals has become imperative to ensure the robust operation of industrial systems and minimize potential system failure risks. Optimizing and improving BiLSTM networks is expected to significantly enhance the accuracy and timeliness of fault warnings, providing strong technical support for the safe and stable operation of industrial systems.

This paper proposes a novel hybrid fault warning method centered around an attention-enhanced bidirectional long short-term memory network (Attention-BiLSTM). The bidirectional processing allows the model to consider both past and future information, providing a comprehensive grasp of the temporal dynamics of the data [8,9]. The introduction of the attention mechanism endows the model with the ability to dynamically adjust its focus at different time steps, significantly enhancing the effectiveness of identifying key fault precursors [8,9]. This structural design enables the model to more accurately capture subtle changes and complex patterns in equipment operation data, providing a solid foundation for early fault detection.

The efficient operation of the Attention-BiLSTM model heavily depends on careful tuning of its hyperparameters. Traditional manual tuning methods are time-consuming, labor-intensive, and lack optimization accuracy [9]. To fully harness the Attention-BiLSTM model's potential, this study introduces MIRDA, an automated hyperparameter optimization tool. MIRDA, which simulates the evolutionary process of red deer populations, adopts an innovative population initialization strategy and integrates advanced search techniques and a boundary constraint mechanism. These enhancements significantly improve the

algorithm's exploration capabilities in complex parameter spaces, enabling more effective identification of optimal hyperparameter configurations.

Extensive experiments conducted on real industrial datasets have validated the effectiveness of this method. Results demonstrate that this hybrid warning approach has achieved significant improvements in key indicators such as predictive accuracy and warning timeliness, exhibiting superior performance. This not only enhances the reliability of fault warnings but also provides more robust support for preventive maintenance of industrial equipment.

In summary, the main contributions of this paper are as follows:

i. Innovatively proposing a hybrid fault warning method that integrates the MIRDA, attention mechanism, and BiLSTM for the first time.
ii. Making multifaceted improvements to the RDA, including an improved population initialization strategy, introducing an adaptive optimal guidance strategy, and incorporating a chaotic adjustment factor, among other improvements. These improvements significantly enhance its optimization efficiency and accuracy.
iii. Through extensive application and verification in real gas-fired power plant equipment fault cases, not only confirming the practicality and effectiveness of the proposed method but also highlighting its significant advantages in key indicators such as predictive accuracy and model robustness through systematic comparison with other advanced methods.

The structure of this paper is as follows: Section 2 reviews the latest developments in the related field and conducts an in-depth analysis. Section 3 introduces the components of the proposed method and their integration. Subsequently, Section 4 demonstrates the performance of the proposed method on real industrial datasets. Section 5 compares the proposed algorithm with other cutting-edge methods. Finally, we discuss the conclusion of this study and propose future research directions (Section 6).

2. Literature Review

The swift development of industry has created an urgent demand for advanced fault warning systems capable of real-time monitoring and predicting potential issues, thereby significantly reducing unexpected downtime and economic losses. In recent years, numerous researchers have explored innovative fault warning methods across various industrial sectors.

Chen et al. [1] (2023) combined a genetic algorithm with a back propagation neural network (GA-BP) for fault warning in the pitch system. This approach demonstrated significant improvements in fault detection accuracy. Similarly, Liang et al. [2] used a bidirectional recurrent neural network to establish a predictive model for wind turbine operation and achieve early fault warning. Correa et al. [3] employed a deep learning method based on the transformer model to predict faults in wind farm modules. Lu et al. [4] proposed a fault warning strategy based on an adaptive deep belief network for electric vehicle charging systems.

Li et al. [5] used the particle swarm optimization algorithm to adjust the parameters of a deep learning model to address minor turn-to-turn short circuits in the excitation winding. Pi et al. [6] applied an enhanced sand cat swarm optimization algorithm (ESCSO) to optimize the hyperparameters of a general regression neural network (GRNN) for various fault warnings. Li et al. [7] proposed an improved arithmetic optimization algorithm (SAOA) to optimize the hyperparameters of a light gradient boosting machine (LightGBM) for fault warning in power plant water pumps. Su et al. [8] combined a convolutional neural network with BiLSTM to develop an efficient diesel engine fault warning method. Tan et al. [9] used an advanced hunter-prey optimization algorithm to optimize the hyperparameters of the BiLSTM model and applied it to the fault warning of condensing water pumps. Cai et al. [10] used extreme gradient boosting technology to build a power distribution network outage prediction model, providing data support for fault warning and maintenance planning.

Additionally, Wang et al. [11] designed a multi-stage fusion LSTM model to predict the future state of valves by analyzing the spatiotemporal characteristics of operating data, in response to the complexity of reciprocating compressors. In the field of microservices architecture, Jing et al. [12] proposed a fault detection method based on a lightweight gradient boosting machine, which improved system reliability through in-depth analysis of historical data. Sun et al. [13] focused on equipment vibration prediction and developed an innovative model that integrates empirical mode decomposition, adaptive noise processing, and LSTM networks, optimizing its parameters with an improved particle swarm algorithm.

Based on an in-depth analysis of the aforementioned literature, various machine learning and deep learning techniques have shown significant potential in the field of fault warning. It is worth noting that the application of BiLSTM in this field is not yet widespread, and although there are some studies on BiLSTM in the field of fault warning, they are all based on the standard BiLSTM framework, and its intrinsic value needs further exploration. To address these limitations, the proposed fault warning model introduces the attention mechanism on the basis of the standard BiLSTM network structure. Moreover, combining metaheuristic algorithms with deep learning technology has been proven to be an effective way to enhance the performance of fault warning algorithms. Although existing methods have achieved certain results in practice, based on the "no free lunch" theorem [14], which states that no single algorithm can perform optimally across all problems, we recognize that each algorithm has its inherent limitations. To further improve the performance of the warning model, we propose an innovative MIRDA and combine it with the BiLSTM network structure based on the attention mechanism, successfully constructing a novel fault warning model. To our knowledge, this is the first time these three technologies have been integrated. This innovative approach not only optimizes the algorithm structure but also enhances the search efficiency, which is expected to improve the accuracy and efficiency of fault warning.

3. Proposed Method

In this section, we provide a detailed introduction to the components of the proposed fault warning method, including the BiLSTM network (Section 3.1), the attention mechanism (Section 3.2), the RDA (Section 3.3), and the MIRDA (Section 3.4). We also elucidate the process by which the components of MIRDA are integrated to fine-tune the hyperparameters of the Attention-BiLSTM model (Section 3.5).

3.1. Bidirectional Long Short-Term Memory Network

A bidirectional long short-term memory network is an advanced neural network architecture that is an extension of the LSTM model. The LSTM model is renowned for its ability to capture long-term dependencies in sequential data, and it achieves this through the use of three critical components: the forget gate, the input gate, and the output gate [9].

The forget gate in an LSTM cell determines which information to discard from the cell state, using the following formula:

$$F_t = \sigma\Big(W_f \cdot [H_{t-1},\ X_t] + b_f\Big) \tag{1}$$

where σ denotes the sigmoid function, F_t represents the output of the forget gate, W_f is the weight matrix associated with the forget gate, b_f is the bias term, H_{t-1} is the hidden state from the previous time step, and X_t is the input data at the current time step.

The input gate decides which new information to store in the cell state, consisting of two parts: the update decision and the candidate value generation. The update decision is given by:

$$I_t = \sigma(W_i \cdot [H_{t-1},\ X_t] + b_i) \tag{2}$$

where I_t is the output of the input gate, W_i is the weight matrix, and b_i is the bias term.

The candidate value generation process is defined as:

$$\widetilde{C}_t = tanh(W_c \cdot [H_{t-1},\ X_t] + b_c \Big) \tag{3}$$

where $\widetilde{C}_t$ represents the candidate value for the cell state, W_c is the weight matrix, and b_c is the bias term.

The cell state is updated according to the following rule:

$$C_t = F_t \cdot C_{t-1} + I_t \cdot \widetilde{C}_t \tag{4}$$

where C_t denotes the current cell state, and C_{t-1} is the cell state from the previous time step.

The output gate determines which information from the cell state will be passed to the hidden state, calculated as:

$$O_t = \sigma(W_o \cdot [H_{t-1},\ X_t] + b_0) \tag{5}$$

where O_t is the output of the output gate and b_0 is the bias term.

The above equation first generates values between 0 and 1 through the sigmoid activation function, enabling selective forgetting or remembering of information, which is crucial for handling long-term dependencies in sequences. Secondly, it combines the weighted previous hidden state and current input, allowing the LSTM to adapt to new information while considering historical context, making it suitable for sequence prediction tasks that require historical context. Finally, the bias term provides flexibility in adjusting the output values, helping the model better fit the data and capture complex patterns. This mechanism makes LSTM excel in handling temporal dynamic behaviors.

The final hidden state is computed as:

$$H_t = O_t \cdot \tanh(C_t) \tag{6}$$

Building upon the LSTM, the BiLSTM was introduced to process sequence information in both forward and backward directions.

The update process for the forward LSTM layer's hidden state is represented as:

$$h_{f,\,t} = \mathrm{LSTM}_{f(x_t,\ h_{f,\,(t-1)})} \tag{7}$$

where $LSTM_f$ denotes the forward LSTM computation, x_t is the input data at time step t, $h_{f,\,(t-1)}$ is the forward hidden state at the previous time step.

Similarly, the backward LSTM layer's hidden state update is given by:

$$h_{b,\,t} = \mathrm{LSTM}_{b(x_t,\ h_{b,\,(t+1)})} \tag{8}$$

where $LSTM_b$ represents the backward LSTM computation, $h_{b,\,(t+1)}$ is the backward hidden state at the subsequent time step.

The hidden states from both directions are then combined in the BiLSTM:

$$h_t = \left[h_{f,\,t}; h_{b,\,t}\right] \tag{9}$$

where the semicolon represents horizontal concatenation.

Finally, the output layer's computation is defined as:

$$y_t = W \cdot h_t + b \tag{10}$$

where y_t represents the output vector at time step t, W is the weight matrix mapping the hidden state to the output space, and b is the learnable offset of the output layer.

3.2. Attention Mechanism

The attention mechanism is a pivotal technique in sequence modeling that enhances the model's focus on key elements within an input sequence. It allows the model to adaptively concentrate on the most critical parts of the input sequence based on the demands of the task at hand. Then, we detail how the attention mechanism typically is implemented [15,16]:

Step 1: Computing the attention distribution

The attention distribution α_n represents the degree of focus on the nth input vector x_n. It is calculated through a normalization function, ensuring that the sum of all attention weights is 1. The formula is as follows:

$$\begin{aligned} \alpha_n = p(z = n \mid X, q) = \\ softmax(s(x_n, q)) = \end{aligned} \quad \frac{exp(s(x_n, q))}{\sum_{j=1}^{N} exp\left(s\left(x_j, q\right)\right)} \tag{11}$$

where X is the set of input vectors, q is the query vector related to the task, $s(\cdot)$ is the attention scoring function, N is the total number of input vectors, and z is the index position for selecting information.

Step 2: Computing the attention-scoring function

The attention-scoring function s evaluates the relevance between each element in the input sequence and the current output element. The formula is as follows:

$$s(x, q) = v^T tanh(Wx + Uq)\Big) \tag{12}$$

where W is the weight matrix for the input vector x, U is the weight matrix for the query vector q, and v is the weight vector used for scoring computation.

Step 3: Computing the weighted average

The weighted average of the input information is calculated based on the attention distribution α_n to generate the final output representation. The formula is as follows:

$$att(X, q) = \sum_{n=1}^{N} \alpha_n x_n \tag{13}$$

where α_n is the degree of focus on the nth input vector obtained through the softmax function, and x_n is the nth input vector.

By integrating the attention mechanism into the BiLSTM, the Attention-BiLSTM model can dynamically adjust its focus on different time points at each timestep. This flexibility significantly enhances the model's ability to process sequence data, making it more efficient and accurate in capturing key information and recognizing patterns. Finally, Figure 1 shows the integration process of the attention mechanism and BiLSTM network.

3.3. Red Deer Optimization Algorithm

The RDA is an efficient metaheuristic algorithm that mimics the natural behavior of red deer (RD). The algorithm has excellent convergence speed and robustness and exhibits superior performance on a variety of test problems. Compared with other state-of-the-art algorithms in the field, RDA has a clear competitive advantage [17]. The following are the steps of its optimization process [17]:

Step 1: Population initialization

Similar to other optimization algorithms, the RDA begins with the generation of an initial population of size N_{pop}. Each member of this population represents an RD, as depicted in Equation (14):

$$\text{RD} = [X_1, X_2, X_3, \ldots, X_{Nvar}] \tag{14}$$

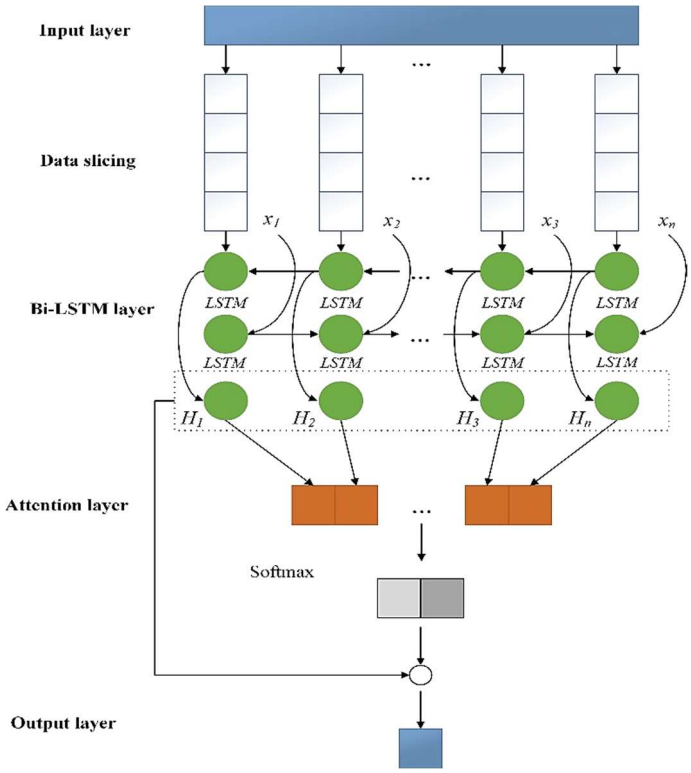


Figure 1. Attention-BiLSTM network structure.

The individuals are created by the following method:

$$\mathrm{RD} = LB + rand() \times (UB - LB) \tag{15}$$

where $Nvar$ denotes the dimension of decision variables, UB and LB define the upper and lower bounds of the search space, respectively. The function $rand()$ generates a random number between 0 and 1.

Subsequently, the RDA selects a subset of superior RDs to act as N_{male}, while the remainder serve as N_{hind} ($N_{\mathrm{hind}} = N_{\mathrm{pop}} - N_{\mathrm{male}}$).

Step 2: Roar male RDs

After role assignment, male RDs attempt to increase their attractiveness by roaring. In the RDA, each male RD generates a new position through movement. If the objective function (OF) value of this new position proves superior, the male RD adopts this position as the new solution. The formula for updating the position of the male RD is as follows:

$$\mathrm{male}_{\mathrm{new}} = \begin{cases} \mathrm{male}_{\mathrm{old}} + a_1 \times ((UB - LB) \times a_2) + LB), & if\ a_3 \geq 0.5 \\ \mathrm{male}_{\mathrm{old}} - a_1 \times ((UB - LB) \times a_2) + LB), & if\ a_3 < 0.5 \end{cases} \tag{16}$$

where $\mathrm{male}_{\mathrm{old}}$ is the current position of the male deer, $\mathrm{male}_{\mathrm{new}}$ is its updated position, a_1, a_2, and a_3 are generated randomly from a uniform distribution between zero and one.

Step 3: Select γ percent of the best male RD as male commanders

In nature, significant differences exist among male RDs. Some are stronger, more attractive, or more successful in territorial expansion. Accordingly, RDs are categorized into two types: commanders and stags.

The number of commanders, N_{Com}, is calculated using the following formula:

$$N_{\mathrm{Com}} = \mathrm{round}\{\gamma \cdot N_{\mathrm{male}}\} \tag{17}$$

where γ is an initial value of the algorithm model, ranging between 0 and 1.

The number of stags, N_{stag}, is then determined by:

$$N_{\text{stag}} = N_{\text{male}} - N_{\text{Com}} \tag{18}$$

where N_{stag} represents the number of stags relative to the male population.

Step 4: Fight between male commanders and stags

After identifying commanders and stags, each commander randomly selects a stag to engage in combat. Each fight results in two new solutions, with the best solution chosen as the new commander. The mathematical formulas used in the fight process are as follows:

$$\text{New1} = \frac{(\text{Com} + \text{Stag})}{2} + b_1 \times ((UB - LB) \times b_2) + LB \tag{19}$$

$$\text{New2} = \frac{(\text{Com} + \text{Stag})}{2} - b_1 \times ((UB - LB) \times b_2) + LB \tag{20}$$

where New1 and New2 are the two new solutions generated by the fight process, Com and Stag representing the symbols for the commander and stag, respectively, UB and LB confine the upper and lower bounds of the search space, b_1 and b_2 are generated randomly from a uniform distribution between 0 and 1.

Step 5: Form harems

After updating the commanders, different harems are formed. Each harem consists of one commander and a group of hinds. The number of hinds in a harem depends on the strength of the commander, defined as its OF value.

In RDA, harems are proportionally allocated to commanders as shown in the following equation:

$$V_n = v_n - \max_i\{v_i\} \tag{21}$$

where v_n represents the strength of the nth commander (i.e., its OF value), V_n is its normalized strength

To calculate the normalized strength of commanders, the following equation is used:

$$P_n = \left|\frac{V_n}{\sum_{i=1}^{N_{\text{Com}}} V_i}\right| \tag{22}$$

Subsequently, the number of hinds in a harem is calculated as follows:

$$N{\cdot}harem_n = \text{round}\{P_n \cdot N_{\text{hind}}\} \tag{23}$$

where $N{\cdot}harem_n$ is the number of hinds in the nth harem, and N_{hind} is the total number of hinds.

Step 6: Mate commander of a harem with a percent of hinds in his harem

In this phase, the commander in each harem mates with the hinds within that harem. The number of hinds participating in mating is determined by the following equation:

$$N{\cdot}harem_n^{\text{mate}} = \text{round}\{\alpha{\cdot}N{\cdot}harem_n\} \tag{24}$$

where $N{\cdot}harem_n^{\text{mate}}$ is the number of hinds mating with the commander in the nth harem, α is a random number between 0 and 1, set during the initial phase of the RDA.

Then, the mating process is formulated as:

$$\text{offs} = \frac{(\text{Com} + \text{Hind})}{2} + (UB - LB) \times c \tag{25}$$

where Com and Hind represent the symbols for the commander and hind, respectively. Offs is the new solution, and UB and LB are the upper and lower bounds, respectively. c is randomly generated from a uniform distribution between 0 and 1.

Step 7: Mate commander of a harem with β percent of hinds in another harem

In this phase, the commander of each harem attacks another harem to expand his territory. Assuming a randomly selected harem (named k) is chosen, the commander mates with β% of the hinds in this harem. The number of hinds from the harem mating with the commander is calculated as follows:

$$N \cdot harem_k^{\text{mate}} = \text{round}\{\beta.N.harem_k\} \tag{26}$$

where $N \cdot harem_k^{\text{mate}}$ is the number of hinds from the kth harem that mates with the commander and β is an initial parameter value of the algorithm model ranging between 0 and 1.

It should be noted that this mating process is also completed using Equation (25).

Step 8: Mate stag with the nearest hind

In this step, each stag mates with its nearest hind. During the breeding season, male RDs tend to pursue the most convenient hinds, potentially their preferred choice among all hinds, irrespective of harem territories. To identify the nearest hind, the distance between a stag and all hinds must be calculated in J-dimensional space:

$$d_i = \left(\sum_{j \in J} \left(stag_j - hind_j^i \right)^2 \right)^{1/2} \tag{27}$$

where d_i represents the spatial interval between the ith female deer and a stag within a multi-dimensional framework. The minimum value within this matrix identifies the female deer that is selected for mating.

Once a female deer is chosen, the subsequent action is the mating process, which is governed by the mathematical formula presented in Equation (25). It is important to note that this formula should account for stags rather than commanders in this context.

Step 9: Select the next generation

In selecting the next generation, the RDA adheres to two distinct strategies. First, the RDA retains all male RDs, including all commanders and stags (i.e., a percentage of the best solutions out of all solutions). The second strategy addresses the remainder of the population in the next generation. The RDA chooses hinds from all hinds and offspring generated by the mating process based on their fitness values, using either the fitness tournament or roulette wheel mechanism.

3.4. Multi-Strategy Improved Red Deer Optimization Algorithm

Based on the aforementioned steps of the RDA, there are certain deficiencies in its initial population construction, execution of search behaviors, and individual boundary constraint strategies. These issues may lead to a reduction in population diversity and a tendency for the algorithm to fall into local optima. In response to these shortcomings, we have designed a series of improvement measures aimed at significantly enhancing the optimization capabilities of RDA, making it more stable and effective in solving complex problems.

3.4.1. Restricted Inverse Learning Mechanism

Traditional RDA often relies on random methods for initializing the population. While straightforward, this approach may be limited by the quality of initial solutions and the diversity of the population, potentially leading the algorithm to become trapped in local optima and affecting the overall optimization performance. To address these limitations, the MIRDA introduces an innovative restricted inverse learning (RIL) mechanism. The

essence of the RIL mechanism is to perform purposeful inverse operations on individuals of the current initial population to generate new, high-quality restricted inverse solutions. Specifically, this mechanism first identifies a reasonable inverse range within the search space and then generates new solutions that are relative to the original ones within this range. This process not only enhances the quality of initial solutions but also significantly increases the diversity of the population. The calculation process is shown as follows:

$$\mathrm{RD}_{\mathrm{new}} = rand() \times (NUB + NLB - \mathrm{RD}_{\mathrm{old}}) \tag{28}$$

where $\mathrm{RD}_{\mathrm{new}}$ is the new solution after the RIL strategy, $\mathrm{RD}_{\mathrm{old}}$ is the randomly generated initial solution, NUB is the maximum value in each dimension of the randomly generated RD population, and NLB is the minimum value in each dimension of the randomly generated RD population.

After generating the new population, we merge it with the original population and sort them, selecting the best N_{pop} individuals as the starting RD population for MIRDA.

3.4.2. Chaos Adjustment Factor

In the second and fourth phases of the RDA, random numbers are used to enhance the randomness and search capability of the algorithm. However, the simple generation of random numbers may result in an uneven and incomplete search process, affecting the overall performance of the algorithm. Therefore, MIRDA introduces a chaos adjustment factor aimed at further improving the search efficiency and optimization capability of the algorithm. The characteristics of chaotic systems can generate more uniformly distributed sequences, which helps the algorithm explore the search space more comprehensively. Moreover, the irregularity of chaotic sequences can assist the algorithm in conducting more detailed searches in local areas, increasing the probability of finding the optimal solution.

There are various methods of chaotic mapping, with the most common being a Logistic map and Tent map [18,19]. However, the Logistic map has poor ergodicity, is sensitive to initial parameter settings, and has a higher density of mapping points at the edge positions and a relatively lower density in the middle, which does not align with the expected population equilibrium in this paper. In contrast, the Tent map has a flatter and more uniform distribution, and its uniformity is better than that of the Logistic map, which can improve the algorithm's optimization speed [20]. The Tent mapping expression is as follows:

$$\lambda_{t+1} = \begin{cases} 2\lambda_t, & 0 \leqslant \lambda_t \leqslant 0.5 \\ 2(1-\lambda_t), & 0.5 < \lambda_t \leqslant 1 \end{cases} \tag{29}$$

where λ is the chaos adjustment factor and t is the iteration round.

Based on the chaotic mapping factor, in MIRDA, the position update formulas in the second and fourth stages are adjusted as follows:

$$\mathrm{male}_{\mathrm{new}} = \begin{cases} \mathrm{male}_{\mathrm{old}} + \lambda \times a_1 \times ((UB - LB) \times a_2) + LB), & if\ a_3 \geq 0.5 \\ \mathrm{male}_{\mathrm{old}} - \lambda \times a_1 \times ((UB - LB) \times a_2) + LB), & if\ a_3 < 0.5 \end{cases} \tag{30}$$

$$\mathrm{New1} = \frac{(\mathrm{Com} + \mathrm{Stag})}{2} + \lambda \times b_1 \times ((UB - LB) \times b_2) + LB \tag{31}$$

$$\mathrm{New2} = \frac{(\mathrm{Com} + \mathrm{Stag})}{2} - \lambda \times b_1 \times ((UB - LB) \times b_2) + LB \tag{32}$$

To our knowledge, this is the first combination of the Tent map with RDA. By introducing the chaotic sequence of the Tent map into RDA, the randomness and unpredictability of the algorithm are effectively enhanced, thereby increasing the diversity of the population and effectively avoiding premature convergence.

3.4.3. Adaptive Optimal Guidance Strategy

In the mating phase of the RDA, the process of updating individual positions through Equation (25) is essentially a self-adjustment under the guidance of the optimal solution in each RD population. Although this method has shown some efficiency in utilizing the information of the optimal solution in the population, it does not fully consider a core issue that the algorithm must handle during the optimization process: how to find the right balance between extensive exploration in the early stage and in-depth development in the later stage. This balance is crucial for the algorithm to effectively search the solution space and ultimately converge to the optimal solution. Exploration ability allows the algorithm to cover a wider range of solution spaces, while development ability ensures that the algorithm can dig deeply into promising areas. However, if the algorithm is too inclined to explore, it may stagnate after finding a local optimal solution; on the contrary, if it is too focused on development, it may miss the global optimal solution. To overcome this challenge, the MIRDA introduces an adaptive optimal guidance strategy in the mating phase. Using the guidance factor ϕ dynamically adjusts the proportion of exploration and development according to the current search phase of the algorithm and the characteristics of the solution space, thereby enhancing the algorithm's ability to mine local optimal solutions while maintaining its global search capability, ensuring that the algorithm can approach the optimal solution more efficiently and accurately throughout the optimization process.

$$\text{offs} = \phi \times \text{Hind} + (1-\phi) \times \text{Com} \times \left(\frac{(\text{Com} + \text{Hind})}{2} + (UB - LB) \times c \right) \tag{33}$$

$$\phi(t) = R_1 + R_2 \times (1 - (T_{max} - t)/T_{max})^2 \tag{34}$$

where R_1 and R_2 are random numbers ranging between 0 and 1, T_{max} represents the maximum number of iterations allowed by the algorithm, and t denotes the current iteration.

3.4.4. Double Mirror Reflection Theory

In the iterative process of the standard RDA, it is common for some individuals in the population to exceed the predetermined search space range. The traditional method of handling this is to directly assign the values of the out-of-bounds individuals to the upper and lower bounds of the search space. However, this simple treatment has significant drawbacks: it causes out-of-bounds individuals to gather at the boundaries, making the population distribution uneven. This uneven distribution reduces the diversity of the population, thereby affecting the overall problem-solving performance of the algorithm. To solve this problem, MIRDA introduces the double mirror reflection theory for boundary constraints of individuals. Compared with traditional methods, the double mirror reflection theory can introduce some randomness when dealing with out-of-bounds individuals, making their regression positions more diverse. This method effectively maintains the diversity of the population, avoiding the aggregation of out-of-bounds individuals at the boundaries, thus solving the problem of uneven distribution [21]. The double mirror reflection process is shown in Equation (35). Note that Equation (35) is applied to each dimension in RD.

$$\text{RD} = \begin{cases} LB + \text{mod}(LB - \text{RD},\ UB - LB),\ \text{RD} < LB \\ UB - \text{mod}(\text{RD} - UB,\ UB - LB),\ \text{RD} > UB \end{cases} \tag{35}$$

where UB and LB are the upper and lower boundaries of the solution space, respectively, and mod is the modulo operator.

3.5. MIRDA Optimization of the Attention-BiLSTM Main Loop

During the execution of MIRDA, our main optimization goal is to determine the values of two key hyperparameters in the Attention-BiLSTM model: the number of neurons in the hidden layer and the learning rate. These two parameters have a significant

impact on the model's performance. To comprehensively assess the impact of different parameter combinations on the model's performance, we have adopted the method of five-fold cross-validation [22]. This method can effectively reduce the risk of overfitting and provide a more reliable performance estimate. In addition, we choose the root mean square error (RMSE) as the main performance evaluation metric. RMSE can intuitively reflect the deviation between predicted values and actual values and is a commonly used evaluation metric in regression problems. For detailed steps on calculating RMSE, readers can refer to reference [23]. Finally, the pseudocode for optimizing the hyperparameters of the Attention-BiLSTM model with MIRDA is as follows (Algorithm 1):

Algorithm 1: MIRDA main loop

01) Initialize the RD population according to Section 3.4.1
02) Form the hinds (N_{hind}) and male RDs (N_{male})
03) X* = the best solution (Attention-BiLSTM optimal hyperparameters)
04) $t = 0$
05) **While** ($t < T_{max}$)
06) Calculate λ and ϕ
07) **for** each male RD
08) Roar the male (Equation (30))
09) Update the position if better than the prior ones
10) **end for**
11) Sort the males and also form the stags and the commanders (Equations (17) and (18))
12) **for** each male commander
13) Fight between male commanders and stags (Equations (31) and (32))
14) Update the position of male commanders and stags
15) **end for**
16) Form harems (Equations (21), (22) and (23))
17) **for** each male commander
18) (Equation (24))
19) Mate a male commander with the selected hinds of his harem randomly (Equation (33))
20) Select a harem randomly and name it k
21) (Equation (26))
22) Mate a male commander with some of the selected hinds of the harem (Equation (33))
23) **end for**
24) **for** each stag
25) Calculate the distance between the stag and all hinds and select the nearest hind (Equation (27))
26) Mate stag with the selected hind (Equation (33))
27) **end for**
28) Select the next generation with the roulette wheel selection
29) Update X* if there is a better solution
30) $t = t + 1$;
31) **end while**
32) Return X

Furthermore, we provide the flowchart for MIRDA, as shown in Figure 2.

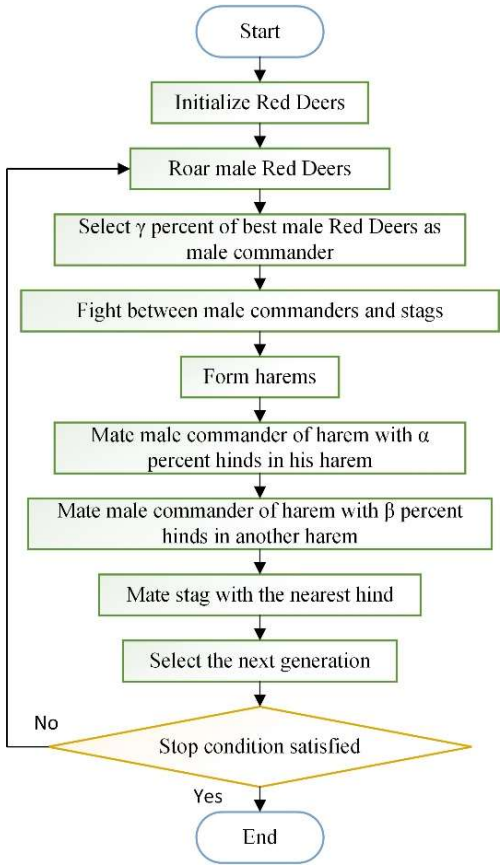


Figure 2. Flowchart of MIRDA.

4. Case Study

In this section, we perform a case study with real data from a feedwater pump at a power plant. We begin with data collection and preprocessing (Section 4.1), followed by the construction and training of the fault warning model (Section 4.2). Finally, we demonstrate the model's alerting effectiveness through an actual fault scenario (Section 4.3). The proposed method is implemented in MATLAB R2019a and runs on a PC with an Intel i5-3470 Core CPU at 3.2 GHz and 4 GB of RAM.

4.1. Data Processing

To verify the effectiveness of the proposed method, it is applied to the overload fault early warning of the high-pressure feedwater pump of Unit 1 at a power plant in Beijing. Data is collected from the plant's monitoring and data acquisition system, including various input measurement points, such as the driving end-bearing temperature of the feedwater pump, the non-driving end-bearing temperature of the feedwater pump, and the thrust-bearing temperature of the feedwater pump. The feedwater pump current serves as the output measurement point. Data were collected from 10:00 on 5 August 2023 to 00:00 on 5 September 2023, with a sampling rate of once every 5 min, totaling 8040 samples.

Given the complex operating environment of the feedwater pump, the historical data inevitably contain missing values and noise. Consequently, our initial step involves identifying and excluding null values, removing anomalous noise, and normalizing the data to ensure consistency and comparability. Then, to prevent overfitting due to the high dimensionality of the dataset, we employ the Pearson correlation coefficient [24,25] to identify measurement points significantly related to the feed pump current, thus ensuring the model's generalization capability. The correlation matrix, as indicated by the Pearson correlation coefficients, is presented in Table 1.

Table 1. Pearson correlation coefficients.

$\|r\|$ Value	Correlation
$\|r\| \geq 0.95$	Significant correlation
$0.95 > \|r\| \geq 0.8$	Strong correlation
$0.8 > \|r\| \geq 0.5$	Moderate correlation
$0.5 > \|r\| \geq 0.3$	Weak correlation
$\|r\| \leq 0.3$	No correlation

After data preprocessing and evaluation based on the Pearson correlation coefficients, five measurement points with the highest correlation to the feedwater pump current are selected as inputs, as shown in Table 2.

Table 2. Correlation coefficients of various feature vectors.

Feature Name	Pearson Correlation Coefficient
Feedwater pump drive end bearing temperature	0.965
Feedwater pump outlet pressure	0.953
Feedwater pump non-driving end bearing temperature	0.950
Feedwater pump export volume	0.948
Feedwater pump speed	0.942

4.2. Model Construction

After completing data processing, we utilize the first 1050 data groups as the training set and the subsequent 450 groups as the testing set. This partitioning ratio achieves a balance between model training and performance assessment, ensuring adequate training and rigorous evaluation. It should be noted that all data sets are collected under the healthy operating conditions of the feedwater pump. Each data set includes the aforementioned five input measurement points and a specific output measurement point. This ensures that the model can learn the data characteristics under normal operating conditions during training and testing, thereby improving the accuracy and reliability of the predictions. Figure 3 displays images of the feedwater pump.



Figure 3. Images of the feedwater pump.

The initial parameter configuration for the model is as follows: $T_{max} = 200, N_{pop} = 100$, $N_{\text{male}} = 15, N_{\text{hind}} = 75, \alpha = 0.9, \beta = 0.4, \gamma = 0.7$, Attention-BiLSTM learning rate is $[0.001,\ 1]$, hidden layer size is $[1,\ 150]$. These parameters are determined based on preliminary experiments and a review of the relevant literature [6–9,17], aiming to provide a solid foundation for the model's performance. In addition, it should be pointed out that we set the other parameters of the Attention-BiLSTM model based on the literature [26]. Due to space limitations, we have not listed them.

Finally, the predictive results of the testing set are displayed in Figure 4. The value on the vertical axis represents the actual and predicted current values of the feedwater pump. The red line represents the model's predicted current values of the feedwater pump, while the blue line represents the actual measured current values of the feedwater pump, serving as a visual representation of the model's effectiveness.

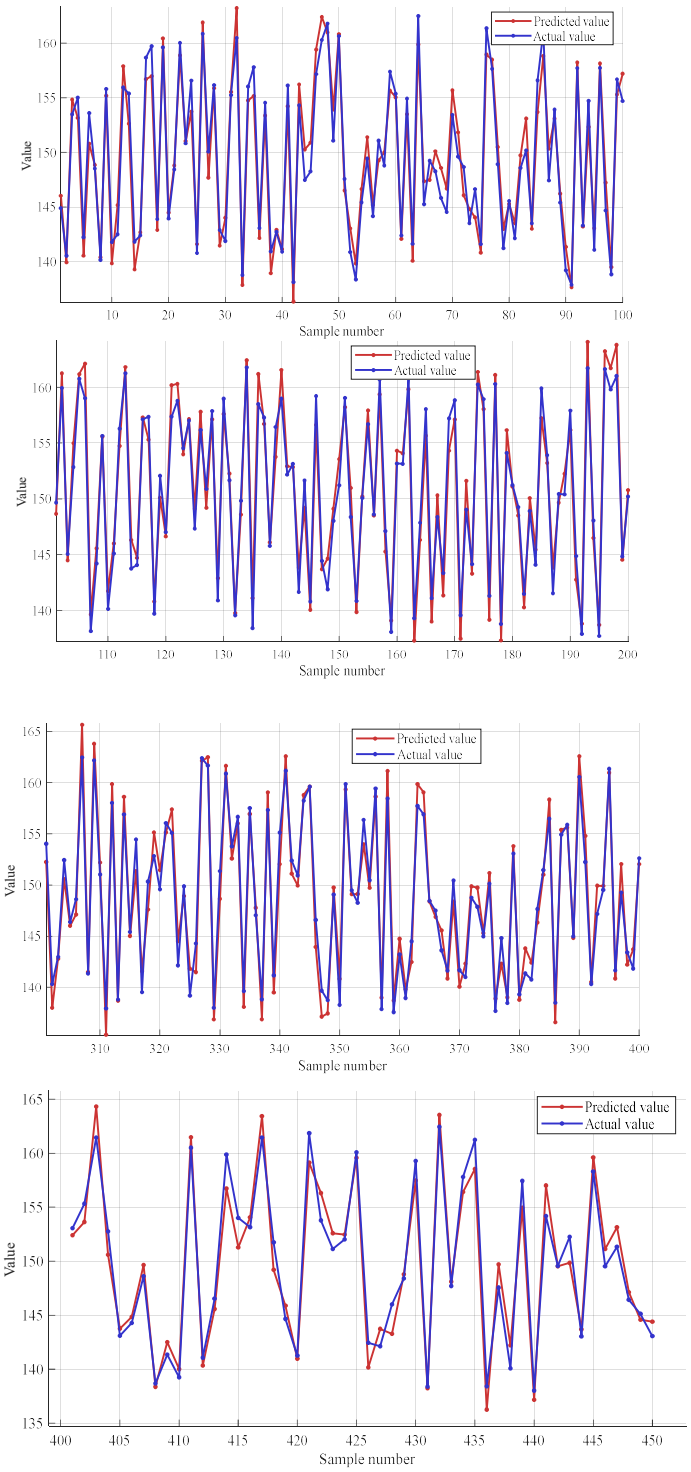


Figure 4. Predictive results on the testing set.

Then, to construct a flexible and effective early warning system, we select the residuals of the output measurement point as the criterion for choosing the warning threshold. We

employ the sliding window method to dynamically adjust the threshold in response to data variability, as shown in Equation (36):

$$\overline{X} = \frac{1}{n}\sum_{i=1}^{n} X_i \tag{36}$$

where $\overline{X}$ represents the average of the residual sequence, X_i denotes the set of residual values within a given window, and n defines the size of the window.

Subsequently, we set the window size n to 20 and the step size to 1. The window size of 20 is chosen to capture sufficient data points for meaningful analysis, while the step size of 1 ensures a detailed and continuous monitoring process. The average of residuals obtained after processing with the sliding window method is illustrated in Figure 5.

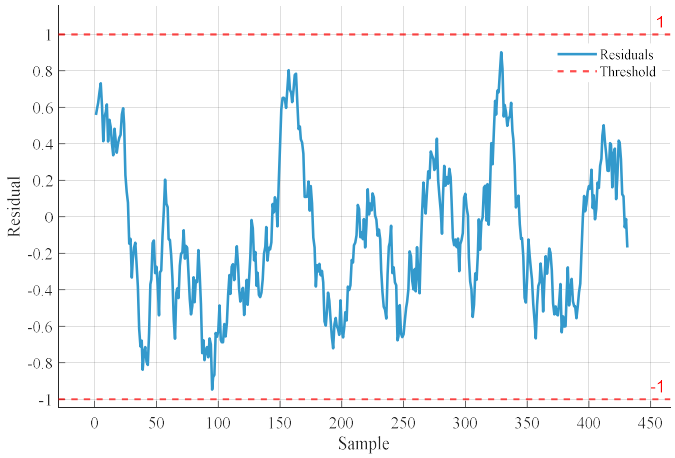


Figure 5. The average of residuals obtained after processing with the sliding window method.

By analyzing Figure 5, it is observed that the average of residuals mainly concentrates within the interval $(-1, 1)$. Hence, we decide to use this interval as the boundary standard for the warning threshold.

4.3. Fault Warning Example

In this section, we validate the effectiveness of the proposed fault warning model using an overload alarm event that occurred in the feedwater pump of Unit 1 at a power plant on 10 August 2023, at approximately 15:30. We collect 500 data samples from before and after the alarm. These data include the aforementioned measurement points, covering the data of the feedwater pump in both healthy and faulty operating states. After preprocessing the fault dataset, including data cleaning and normalization to standardize the data, we input it into the MIRDA-Attention-BiLSTM model to calculate the feedwater pump's current residuals.

Subsequently, we also set the window size to 20 and the step size to 1, analyzing the data using the sliding window method. Figure 6 presents the analysis results of the average residual of the feedwater pump current. The average residual curve shows that the preset threshold range is breached at the 407th sample point, which corresponds to 13:20 on 10 August 2023, approximately 2 h earlier than the system's actual recorded alarm time.

This result strongly demonstrates that the proposed method can effectively achieve early fault warning for power plant feedwater pumps. By detecting potential faults in advance, this method allows plant operators to implement preventive measures in a timely manner. This early detection has the potential to reduce equipment damage, lower maintenance costs, and improve overall operational efficiency.

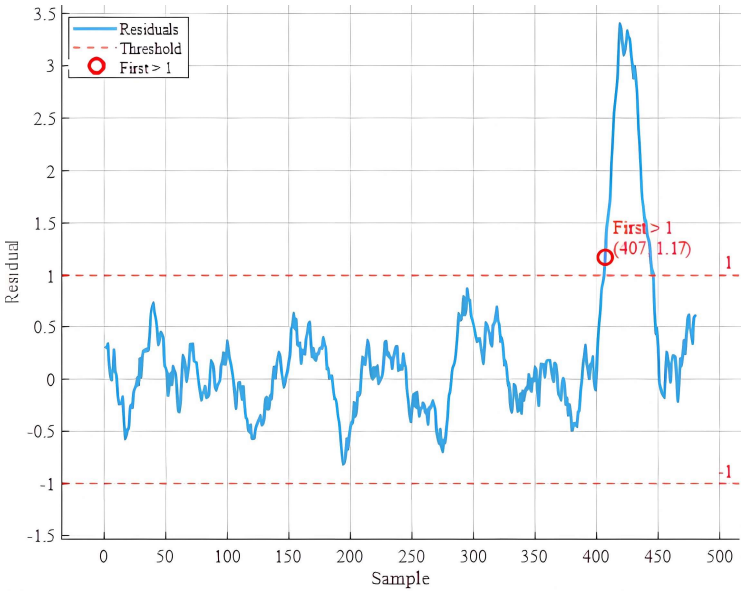


Figure 6. Analysis results of the average residuals for the fault warning example.

5. Algorithm Performance Analysis

In this section, we first analyze the effectiveness of the MIRDA algorithm (Section 5.1), then compare the MIRDA-Attention-BiLSTM with several other advanced algorithms to showcase its superior performance in fault warning tasks (Section 5.2), and finally, employ the Wilcoxon signed-rank test to evaluate the significance of performance differences among various algorithms, presenting the results intuitively through charts, thereby further validating the effectiveness of the proposed algorithm (Section 5.3).

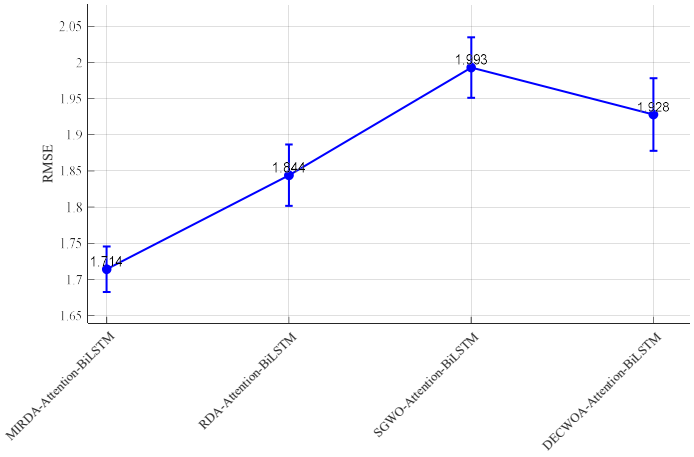
5.1. Effectiveness Analysis of MIRDA

To validate the effectiveness of the proposed MIRDA, we use the RMSE and the mean absolute percentage error (MAPE) as metrics, comparing them with several other advanced metaheuristic algorithms, including the RDA [17], improved grey wolf optimizer (SGWO) [27], and differential evolution chaotic whale optimization algorithm (DECWOA) [28]. These algorithms are applied to the parameter tuning of the Attention-BiLSTM model and are empirically studied in the aforementioned case. For the calculation method of MAPE, reference [23] can be similarly consulted.

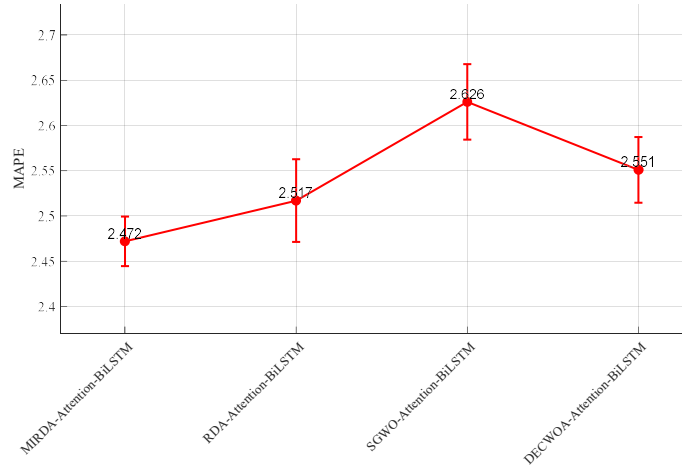
In the experiment, considering that comparing metaheuristic algorithms based on the maximum number of iterations is unfair [29,30], we instead use their running time as the stopping criterion, that is, each algorithm is given a 50 s time limit. Additionally, the population size is set to 100. This allows each metaheuristic algorithm to fully explore its solution space. The parameters and search ranges for MIRDA and Attention-BiLSTM are set the same as in the previous section. For all other parameters, we set each method according to the literature recommendations. Furthermore, considering the inherent randomness of metaheuristic algorithms, each algorithm is run independently ten times to ensure the reliability and stability of the results. To more comprehensively display the performance of the algorithms, we present the statistical results with a 95% confidence interval. Finally, the average values of these ten runs are shown in Table 3, and the 95% confidence intervals are illustrated in Figure 7. Additionally, Figure 7 displays the standard deviation of these ten runs.

Table 3. Average performance metrics of algorithms across ten independent runs (Section 5.1).

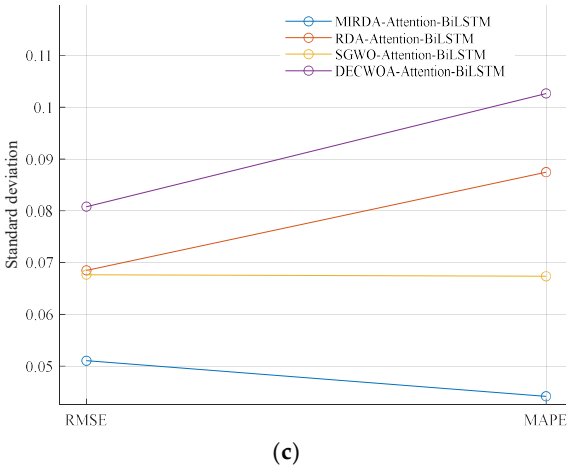
Algorithm Name	RMSE	MAPE
MIRDA-Attention-BiLSTM	1.714	2.472
RDA-Attention-BiLSTM	1.844	2.545
SGWO-Attention-BiLSTM	1.993	2.626
DECWOA-Attention-BiLSTM	1.928	2.591



(**a**)



(**b**)



(**c**)

Figure 7. The 95% confidence intervals and standard deviation of algorithm performance (Section 5.1): RMSE (**a**), MAPE (**b**), standard deviation (**c**).

Based on the results in Table 3 and Figure 7, compared to other advanced metaheuristic algorithms, MIRDA can significantly enhance the performance and stability of the model. Through multi-strategy improvements, MIRDA can more effectively explore the parameter space, find better parameter combinations, and thus improve the prediction accuracy of the model. Moreover, MIRDA maintains stable performance in multiple experiments, reducing performance fluctuations due to randomness and ensuring the reliability of the algorithm in practical applications.

5.2. Effectiveness Analysis of MIRDA-Attention-BiLSTM

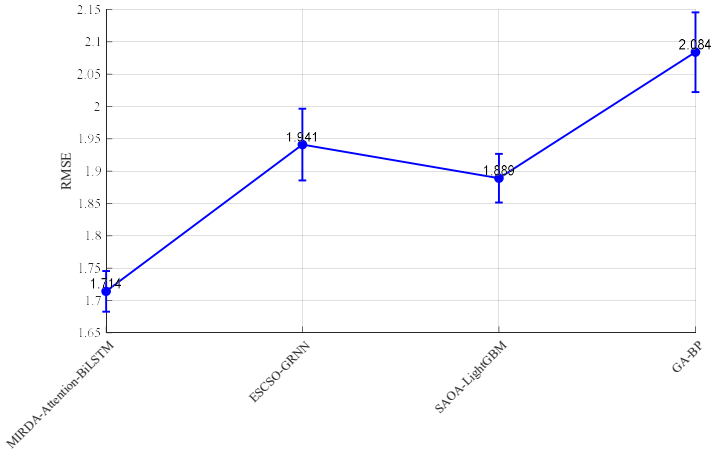
To verify the effectiveness of MIRDA-Attention-BiLSTM, we compare it with several state-of-the-art methods, including GA-BP [1], ESCSO-GRNN [4], and SAOA-LightGBM [5]. In this section, the total running time for each method is set as the stopping criterion, which we set to 120 s. The experimental process follows the same steps as described in the previous section. The average values of the experimental results over ten runs are presented in Table 4, and the 95% confidence intervals are graphically displayed in Figure 8. Figure 8 also displays the standard deviation of these ten runs.

Table 4. Average performance metrics of algorithms across ten independent runs (Section 5.2).

Algorithm Name	RMSE	MAPE
MIRDA-Attention-BiLSTM	1.714	2.472
ESCSO-GRNN	2.141	2.691
SAOA-LightGBM	1.950	2.562
GA-BP	2.284	2.735

Based on the data presented in Table 4 and Figure 8, the MIRDA-Attention-BiLSTM model achieves optimal performance in key metrics such as RMSE and MAPE, highlighting its outstanding predictive accuracy and adaptability in complex data environments. Further, 95% confidence interval analysis reveals the stability of the model in multiple experiments, which not only verifies the reliability of its predictive results but also provides strong evidence for its robustness in practical applications.

In summary, the MIRDA-Attention-BiLSTM model, by integrating the advantages of MIRDA, attention mechanism, and BiLSTM, significantly improves the predictive accuracy and robustness in complex fault warning tasks. This integrated strategy not only enhances the model's adaptability to complex fault patterns but also ensures its efficiency and reliability in practical applications, providing solid technical support for precise early warning.



(**a**)

Figure 8. *Cont.*

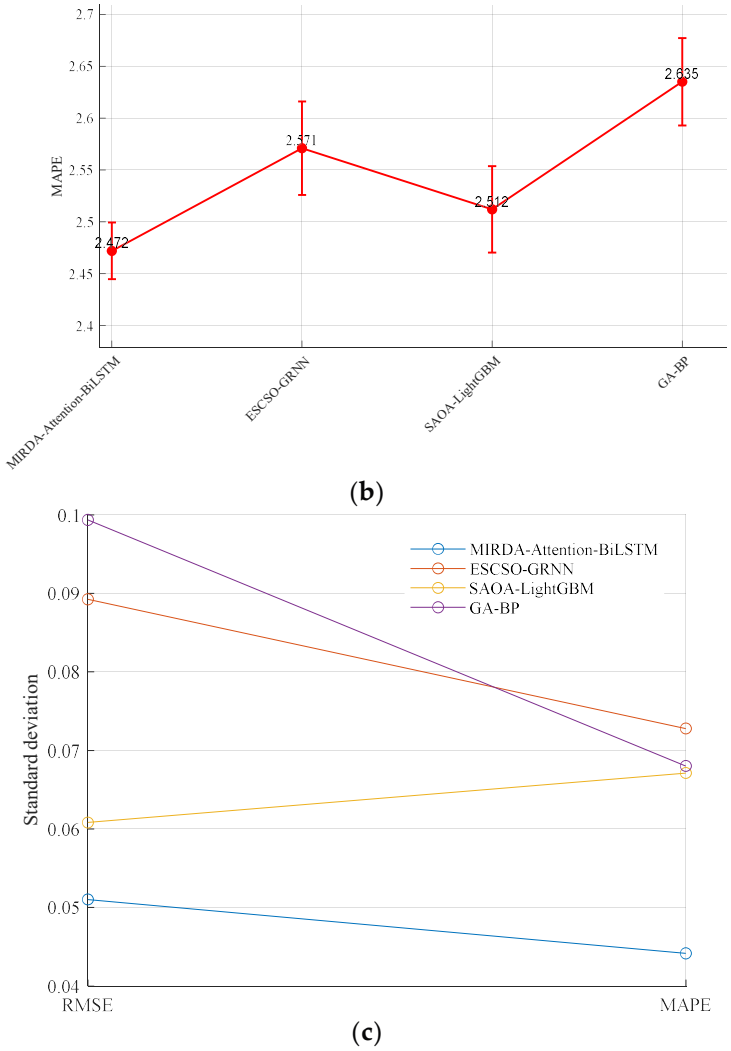


Figure 8. The 95% confidence intervals and standard deviation of algorithm performance (Section 5.2): RMSE (**a**), MAPE (**b**), standard deviation (**c**).

5.3. Statistical Analysis

In order to substantiate the findings from our comparative analysis, we delve into a rigorous statistical examination of the performance metrics of the various algorithms under consideration. The Wilcoxon signed-rank test [31], a non-parametric statistical hypothesis test particularly suited for paired samples and small datasets, is employed to assess the significance of the observed differences in performance.

Adhering to the guidelines provided in related literature, we establish a significance threshold of 0.05 for our *p*-values [32], which is a conventional level in statistical hypothesis testing that balances between avoiding false positives and retaining the power to detect true effects.

The outcomes of our statistical scrutiny, detailed in Tables 5–8, are annotated with symbols to succinctly convey the nature of the differences observed. Specifically, the presence of a "+" symbol denotes that a statistically significant disparity exists between the two algorithms in question, suggesting that the performance of one algorithm is markedly superior or inferior to the other. Conversely, the "~" symbol indicates that the algorithms exhibit no statistically significant difference, implying that their performance is effectively indistinguishable from a statistical perspective. Additionally, Figures 9 and 10 display the results from the table in an intuitive graphical representation.

Table 5. Statistical analysis of RMSE experimental results (Section 5.1).

	MIRDA-Attention-BiLSTM	RDA-Attention-BiLSTM	SGWO-Attention-BiLSTM	DECWOA-Attention-BiLSTM
MIRDA-Attention-BiLSTM	---	0.006(+)	0.002(+)	0.002(+)
RDA-Attention-BiLSTM	0.006(+)	---	0.002(+)	0.001(+)
SGWO-Attention-BiLSTM	0.002(+)	0.002(+)	---	0.004(+)
DECWOA-Attention-BiLSTM	0.002(+)	0.001(+)	0.004(+)	---

Table 6. Statistical analysis of MAPE experimental results (Section 5.1).

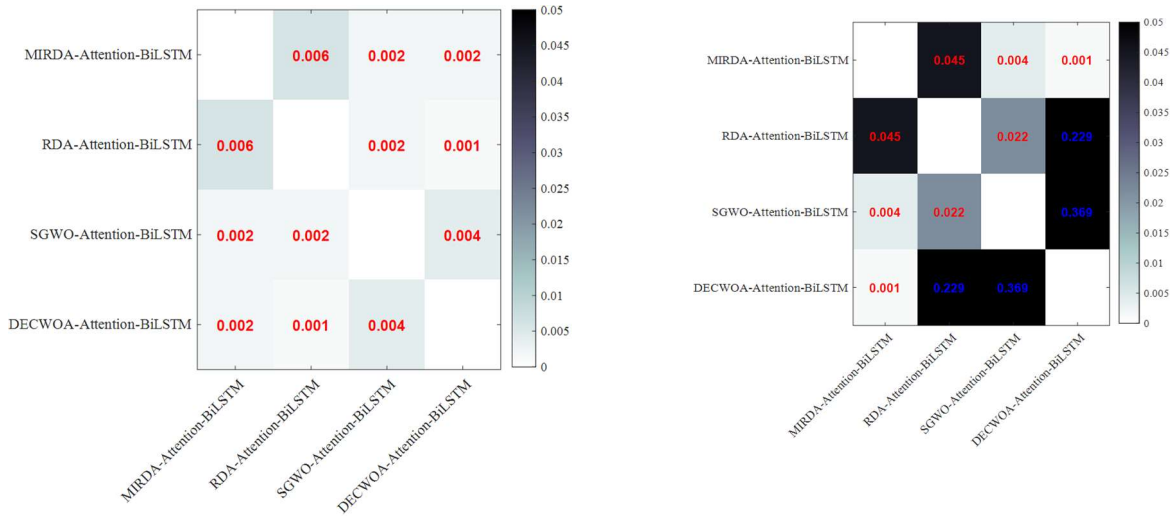
	MIRDA-Attention-BiLSTM	RDA-Attention-BiLSTM	SGWO-Attention-BiLSTM	DECWOA-Attention-BiLSTM
MIRDA-Attention-BiLSTM	---	0.045(+)	0.004(+)	0.001(+)
RDA-Attention-BiLSTM	0.045(+)	---	0.022(+)	0.229(+)
SGWO-Attention-BiLSTM	0.004(+)	0.022(+)	---	0.369(~)
DECWOA-Attention-BiLSTM	0.001(+)	0.229(+)	0.369(~)	---

Table 7. Statistical analysis of RMSE experimental results (Section 5.2).

	MIRDA-Attention-BiLSTM	ESCSO-GRNN	SAOA-LightGBM	GA-BP
MIRDA-Attention-BiLSTM	--	0.002(+)	0.001(+)	0.002(+)
ESCSO-GRNN	0.002(+)	--	0.023(+)	0.004(+)
SAOA-LightGBM	0.001(+)	0.023(+)	--	0.013(+)
GA-BP	0.002(+)	0.004(+)	0.013(+)	--

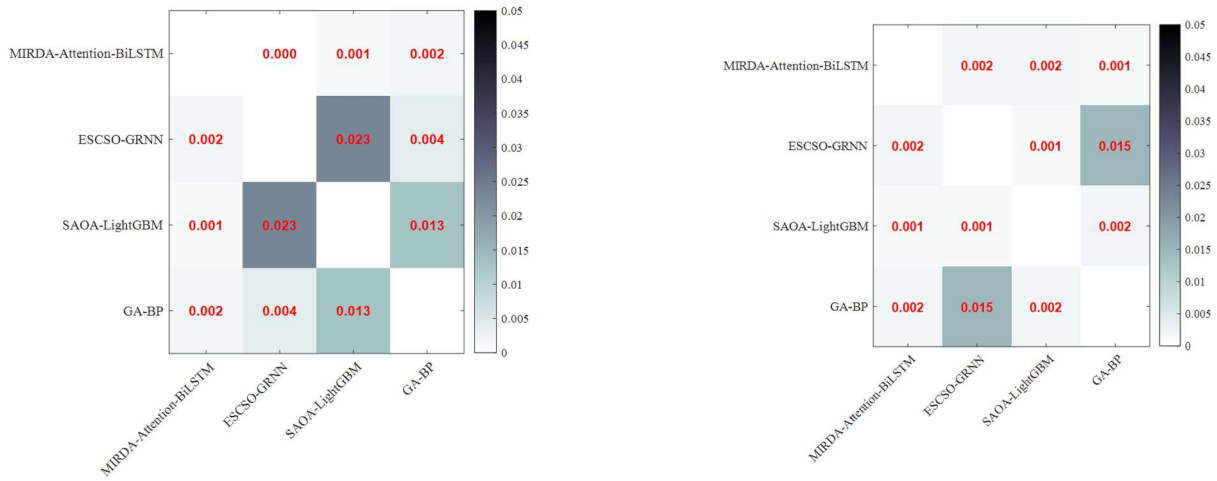
Table 8. Statistical analysis of MAPE experimental results (Section 5.2).

	MIRDA-Attention-BiLSTM	ESCSO-GRNN	SAOA-LightGBM	GA-BP
MIRDA-Attention-BiLSTM	--	0.002(+)	0.001(+)	0.002(+)
ESCSO-GRNN	0.002(+)	--	0.001(+)	0.015(+)
SAOA-LightGBM	0.001(+)	0.001(+)	--	0.002(+)
GA-BP	0.002(+)	0.015(+)	0.002(+)	--



(**a**) (**b**)

Figure 9. Graphical presentation of statistical results (Section 5.1): RMSE (**a**), MAPE (**b**).



(**a**) (**b**)

Figure 10. Graphical presentation of statistical results (Section 5.2): RMSE (**a**), MAPE (**b**).

Based on the comprehensive examination of the data presented in Tables 5–8 and the visual representations provided by Figures 9 and 10, it is evident that the statistical analysis outcomes for MIRDA and MIRDA-Attention-BiLSTM corroborate the initial observations with high statistical significance. This validation process not only bolsters the credibility of our research findings but also highlights the robustness and efficacy of the proposed methodologies.

6. Conclusions

This study introduces an innovative hybrid fault warning method integrating MIRDA, attention mechanism, and BiLSTM. Applied to a feedwater pump overload warning case, the method demonstrates exceptional performance, predicting the fault approximately two hours in advance, highlighting its potential for early fault detection and prevention in industrial environments. Experimental results show that the MIRDA-Attention-BiLSTM model significantly outperforms other advanced algorithms in fault warning tasks, achieving the lowest RMSE and MAPE values. The multi-strategy improvements in MIRDA, including the RIL mechanism, chaos adjustment factor, adaptive optimal guidance strategy, and double mirror reflection theory, substantially enhance the algorithm's optimization efficiency and accuracy. The integration of the attention mechanism with BiLSTM enables the model to capture long-term dependencies more effectively and focus on the most relevant features in the input sequence. Statistical analysis using the Wilcoxon signed-rank test further validates the significant performance improvements of this method over existing technologies, confirming its robustness and reliability in processing complex time series data. These findings have important implications for industrial fault warning systems, providing a reliable and effective tool for improving equipment stability, enhancing operational efficiency, and reducing maintenance costs in complex industrial environments. Future research could explore the applicability of this method to other types of industrial equipment and fault scenarios, as well as multiple faults within the same equipment. Additionally, integrating explainable AI techniques could enhance the interpretability of model predictions [33]. Furthermore, given the superior performance of MIRDA, applying it to hyperparameter tuning of other machine learning models is a promising direction for exploration. Additionally, considering the combination of MIRDA with other advanced metaheuristic algorithms [34–36] to further improve performance presents an intriguing avenue for future research.

Author Contributions: Formal analysis, Y.W.; Investigation, Y.W.; Methodology, Y.W. and M.W.; Resources, M.W.; Supervision, M.W.; Writing—original draft, Y.W.; Writing—review and editing, M.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The data presented in this study are available on request from the corresponding author.

Conflicts of Interest: Author Yutian Wang was employed by the company Guoneng Guohua (Beijing) Cogeneration Power Co., Ltd. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest. The company had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

References

1. 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, 2020 5th International Conference on Advances in Energy and Environment Research (ICAEER 2020), Shanghai, China, 18–20 September 2020*; EDP Sciences: Les Ulis, France, 2020; Volume 194, p. 03005.
2. Liang, T.; Qian, S.; Meng, Z.; Xie, G.F. Early fault warning of wind turbine based on BRNN and large sliding window. *J. Intell. Fuzzy Syst.* **2020**, *38*, 3389–3401.
3. Maldonado-Correa, J.; Torres-Cabrera, J.; Martín-Martínez, S.; Artigao, E.; Gómez-Lázaro, E. Wind turbine fault detection based on the transformer model using SCADA data. *Eng. Fail. Anal.* **2024**, *162*, 108354. [CrossRef]
4. 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]
5. 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]
6. Pi, Y.; Tan, Y.; Golmohammadi, A.M.; Guo, Y.; Xiao, Y.; Chen, Y. A Fault Warning Approach Using an Enhanced Sand Cat Swarm Optimization Algorithm and a Generalized Neural Network. *Processes* **2023**, *11*, 2543. [CrossRef]
7. Li, S.; Jin, N.; Dogani, A.; Yang, Y.; Zhang, M.; Gu, X. Enhancing LightGBM for Industrial Fault Warning: An Innovative Hybrid Algorithm. *Processes* **2024**, *12*, 221. [CrossRef]
8. Su, Y.; Gan, H.; Ji, Z. Research on Multi-Parameter Fault Early Warning for Marine Diesel Engine Based on PCA-CNN-BiLSTM. *J. Mar. Sci. Eng.* **2024**, *12*, 965. [CrossRef]
9. Tian, J.; Zhang, X.; Zheng, S.; Liu, Z.; Zhan, C. Synergising an Advanced Optimisation Technique with Deep Learning: A Novel Method in Fault Warning Systems. *Mathematics* **2024**, *12*, 1301. [CrossRef]
10. Cai, J.; Cai, Y.; Cai, H.; Shi, S.; Lin, Y.; Xie, M. Feeder fault warning of distribution network based on XGBoost. *J. Phys. Conf. Ser.* **2020**, *1639*, 012037. [CrossRef]
11. 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]
12. 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; pp. 709–713.
13. Sun, Y.; Wang, L.; Lei, Q.; Cao, Q. Research on the Vibration Prediction of Hydropower Unit Based on CEEMDAN-IPSO-LSTM. *Yellow River* **2023**, *45*, 156–162.
14. Tian, G.; Zhang, X.; Fathollahi-Fard, A.M.; Jiang, Z.; Zhang, C.; Yuan, G.; Pham, D.T. Hybrid evolutionary algorithm for stochastic multiobjective disassembly line balancing problem in remanufacturing. *Environ. Sci. Pollut. Res.* **2023**, 1–16. [CrossRef] [PubMed]
15. Niu, Z.; Zhong, G.; Yu, H. A review on the attention mechanism of deep learning. *Neurocomputing* **2021**, *452*, 48–62. [CrossRef]
16. Soydaner, D. Attention mechanism in neural networks: Where it comes and where it goes. *Neural Comput. Appl.* **2022**, *34*, 13371–13385. [CrossRef]
17. Fathollahi-Fard, A.M.; Hajiaghaei-Keshteli, M.; Tavakkoli-Moghaddam, R. Red deer algorithm (RDA): A new nature-inspired meta-heuristic. *Soft Comput.* **2020**, *24*, 14637–14665. [CrossRef]
18. Sayed, W.S.; Fahmy, H.A.; Rezk, A.A.; Radwan, A.G. Generalized smooth transition map between tent and logistic maps. *Int. J. Bifurc. Chaos* **2017**, *27*, 1730004. [CrossRef]
19. Layek, G.C. Some Maps. In *An Introduction to Dynamical Systems and Chaos*; Springer Nature: Singapore, 2024; pp. 447–496.
20. Hu, C.A.; Xiong, Y.R. An improved chaotic Harris hawk optimization algorithm with multiple strategies. *Comput. Eng. Sci.* **2023**, *9*, 1648–1660.
21. Wu, Z.X.; Liu, J.; Qin, T.; Chen, C.S.; Li, W.; Yang, J. An improved elite coyote optimization algorithm with multiple strategies. In *Computer Engineering and Science*; Springer: Berlin/Heidelberg, Germany, 2024; pp. 1–15.

22. Anguita, D.; Ghelardoni, L.; Ghio, A.; Oneto, L.; Ridella, S. The 'K' in K-fold Cross Validation. In Proceedings of the ESANN, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning, Bruges, Belgium, 25–27 April 2012; Volume 102, pp. 441–446.
23. Chicco, D.; Warrens, M.J.; Jurman, G. The coefficient of determination R-squared is more informative than SMAPE, MAE, MAPE, MSE and RMSE in regression analysis evaluation. *Peerj Comput. Sci.* **2021**, *7*, e623. [CrossRef]
24. Chen, S.Q.; Zhou, H.H.; Mao, D.J. Research on the fault warning method of coal grinding machine based on improved GWO-LightGBM. *Autom. Instrum.* **2024**, *45*, 106–110+115.
25. Cohen, I.; Huang, Y.; Chen, J.; Benesty, J. Pearson correlation coefficient. In *Noise Reduction in Speech Processing*; Springer: Berlin/Heidelberg, Germany, 2009; pp. 1–4.
26. Qiu, S.; Wang, Y.; Lv, Y.; Chen, F.; Zhao, J. Optimizing BiLSTM network attack prediction based on improved gray wolf algorithm. *Appl. Sci.* **2023**, *13*, 6871. [CrossRef]
27. Liu, W.; Sun, J.; Liu, G.; Fu, S.; Liu, M.; Zhu, Y.; Gao, Q. Improved GWO and its application in parameter optimization of Elman neural network. *PLoS ONE* **2023**, *18*, e0288071. [CrossRef] [PubMed]
28. Liu, L.; Zhang, R. Multistrategy improved whale optimization algorithm and its application. *Comput. Intell. Neurosci.* **2022**, *2022*, 3418269. [CrossRef] [PubMed]
29. Mernik, M.; Liu, S.H.; Karaboga, D.; Črepinšek, M. On clarifying misconceptions when comparing variants of the artificial bee colony algorithm by offering a new implementation. *Inf. Sci.* **2015**, *291*, 115–127. [CrossRef]
30. Draa, A. On the performances of the flower pollination algorithm—Qualitative and quantitative analyses. *Appl. Soft Comput.* **2015**, *34*, 349–371. [CrossRef]
31. Cong, X.; Hartings, J.A.; Rao, M.B.; Jandarov, R.A. A count weighted Wilcoxon rank-sum test and application to medical data. In *Communications in Statistics-Simulation and Computation*; Taylor & Francis Online: London, UK, 2023; pp. 1–11.
32. Perolat, J.; Couso, I.; Loquin, K.; Strauss, O. Generalizing the Wilcoxon rank-sum test for interval data. *Int. J. Approx. Reason.* **2015**, *56*, 108–121. [CrossRef]
33. Thunki, P.; Reddy SR, B.; Raparthi, M.; Maruthi, S.; Dodda, S.B.; Ravichandran, P. Explainable AI in Data Science—Enhancing Model Interpretability and Transparency. *Afr. J. Artif. Intell. Sustain. Dev.* **2021**, *1*, 1–8.
34. Fathollahi-Fard, A.M.; Wu, P.; Tian, G.; Yu, D.; Zhang, T.; Yang, J.; Wong, K.Y. An efficient multi-objective adaptive large neighborhood search algorithm for solving a disassembly line balancing model considering idle rate, smoothness, labor cost, and energy consumption. *Expert Syst. Appl.* **2024**, *250*, 123908. [CrossRef]
35. Zhang, X.; Fu, A.; Zhan, C.; Pham, D.T.; Zhao, Q.; Qiang, T.; Aljuaid, M.; Fu, C. Selective disassembly sequence planning under uncertainty using trapezoidal fuzzy numbers: A novel hybrid metaheuristic algorithm. *Eng. Appl. Artif. Intell.* **2024**, *128*, 107459. [CrossRef]
36. Şenel, F.A.; Gökçe, F.; Yüksel, A.S.; Yiğit, T. A novel hybrid PSO–GWO algorithm for optimization problems. *Eng. Comput.* **2019**, *35*, 1359–1373. [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.