Journal of Computer Science and Artificial Intelligence
ISSN: 3078-8242 (Print), ISSN: 3078-2309 (Online) | Vol. 2, No. 2, 2025

Compressor Oil Temperature Prediction Based on Optimization Algorithms and Deep Learning

Chenyang Chai[1, *]

[1]School of Computer Science and Technology, Henan Polytechnic University, Jiaozuo, 454000, China

[*]Corresponding Author: Chenyang Chai

Abstract: The prediction of lubrication oil temperature plays a crucial role in the performance optimization and fault diagnosis of twin-screw refrigeration compressors. However, due to the complexity of operating conditions and the nonlinear characteristics of the data, traditional prediction methods still face challenges in terms of accuracy and generalization ability. This paper proposes a lubrication oil temperature prediction method based on optimization algorithms and deep learning, integrating an improved Beluga Whale Optimization (BWO) algorithm with a Temporal Convolutional Network (TCN) and an Attention Mechanism. The improved BWO algorithm enhances global search capability and convergence speed by introducing a survival strategy and an optimal position update strategy while optimizing hyperparameter selection to improve the predictive performance of the deep learning model. In experiments, we conducted a comparative analysis of various optimization algorithms and the improved BWO. The experimental results demonstrate that the BWO-optimized TCN-Attention model (BWO-ATCNS) outperforms traditional methods in prediction accuracy, with a lower Mean Squared Error (MSE), reduced Root Mean Squared Error (RMSE), and a higher coefficient of determination (R^2). This study provides an efficient and reliable solution for lubrication oil temperature prediction in twin-screw refrigeration compressors and offers new insights for industrial predictive modeling.

Keywords: Lubrication Oil Temperature Prediction; Temporal Convolutional Network; Attention Mechanism; Beluga Whale Optimization Algorithm.

1. Introduction

Lubrication oil temperature is a critical parameter affecting the performance of twin-screw refrigeration compressors. Its stability is directly related to system efficiency, equipment lifespan, and failure rates. Fluctuations in lubrication oil temperature can lead to a decline in lubrication performance, increased component wear, and even system failures. Therefore, accurately predicting lubrication oil temperature is essential for optimizing equipment operation and reducing maintenance costs. However, due to the complex operating conditions of refrigeration compressors, which involve multiple nonlinear and time-varying factors, traditional physical modeling and statistical regression methods struggle to effectively capture the system's dynamic characteristics, limiting prediction accuracy. In recent years, with the rapid advancement of artificial intelligence, deep learning has demonstrated strong modeling capabilities in time-series forecasting tasks. Among deep learning models, the Temporal Convolutional Network (TCN) has been widely applied in industrial prediction tasks due to its advantages, such as strong parallel computing capabilities and superior ability to capture long-term dependencies. Additionally, the Attention Mechanism enhances model robustness by enabling a more focused extraction of key features. However, the predictive performance of deep learning models heavily depends on proper hyperparameter selection and optimization. Without effective optimization, models may suffer from slow convergence or get trapped in local optima. To address these challenges, this paper proposes a lubrication oil temperature prediction method that integrates optimization algorithms and deep learning. We introduce an improved Beluga Whale Optimization (BWO) algorithm, which enhances global search capability and convergence efficiency through survival strategies and optimal position update mechanisms, thereby optimizing the hyperparameters of the TCN-Attention model. Experimental results demonstrate that this approach outperforms traditional methods in terms of prediction accuracy, generalization ability, and computational efficiency, offering a highly accurate solution for lubrication oil temperature prediction and supporting the intelligent operation of twin-screw refrigeration compressors. The main contributions of this study are as follows:

1) A lubrication oil temperature prediction method based on optimization algorithms and deep learning is proposed, where an improved BWO algorithm is used to optimize the hyperparameters of the TCN-Attention model, enhancing prediction accuracy.

2) An improved Beluga Whale Optimization (BWO) algorithm is introduced, incorporating survival strategies and optimal position update mechanisms to improve search capability, convergence speed, and robustness.

3) Extensive experimental validation is conducted, comparing the proposed method with various optimization algorithms to demonstrate the superiority of the BWO-TCN-AM model.

The structure of this paper is organized as follows: Section 2 introduces the related research background and optimization algorithms. Section 3 details the integration of the improved BWO algorithm with the TCN-Attention model. Section 4 presents the experimental setup and result analysis. Finally, Section 5 concludes the study and discusses future research directions.

2. Theoretical Background

2.1. Temporal Convolutional Network (TCN)

Temporal Convolutional Networks (TCNs) have emerged as an effective alternative to recurrent neural networks (RNNs) for sequence modeling tasks. Unlike traditional RNN-based models such as Long Short-Term Memory (LSTM) and Gated Recurrent Units (GRU), TCNs leverage convolutional operations to process time-series data, enabling efficient parallel computation and improved long-term dependency learning[1].TCNs employ causal convolutions, ensuring that the prediction at a given time step depends only on past inputs, thereby maintaining the temporal order of the data. Additionally, they utilize dilated convolutions, where the convolutional filter skips a fixed number of input values, exponentially increasing the receptive field without requiring a large number of layers[2]. This allows TCNs to capture long-range dependencies efficiently. Another key feature of TCNs is residual connections, which facilitate gradient propagation and mitigate the vanishing gradient problem, making the model more stable during training[3]. Mathematically, the output of a TCN layer can be represented as:

$$y(t)=\sum_{i=0}^{k-1} w_i \cdot x(t-d\cdot i). \qquad (1)$$

where $y(t)$ is the output at time step t, w_i represents the convolutional filter weights, $x(t)$ is the input sequence, k is the filter size, and d is the dilation factor. By increasing d exponentially across layers, the receptive field of the network expands significantly, allowing it to capture long-term dependencies in time-series data[4]. Due to these advantages, TCNs have been widely adopted in industrial applications, including anomaly detection, predictive maintenance, and time-series forecasting[5].

2.2. Attention Mechanism

The Attention Mechanism is a key innovation in deep learning that allows models to selectively focus on important parts of the input sequence when making predictions. Originally introduced in natural language processing (NLP), attention has been successfully applied to time-series forecasting and other sequential tasks[6].

The core idea behind attention is to assign different importance weights to different time steps in an input sequence. This enables the model to dynamically adjust its focus based on contextual information, improving the ability to capture relevant patterns while reducing noise[7]. The general formulation of attention can be expressed as follows:

$$Attention(Q,K,V)=soft\max(\frac{QK^T}{\sqrt{d_k}})V. \qquad (2)$$

Where Q , K , V and are learned feature representations, d_k is the dimension of the key vectors,The softmax function ensures that attention weights sum to 1, normalizing the impact of different input tokens.

Among various attention mechanisms, self-attention (as used in Transformer models) has proven highly effective for sequence modeling, allowing the model to attend to all positions in the input simultaneously【8】. In the context of TCN-based time-series forecasting, incorporating an attention mechanism helps enhance the model's adaptability by weighting critical time steps more heavily.

2.3. Beluga Whale Optimization Algorithm (BWO)

The Beluga Whale Optimization (BWO) algorithm was proposed by Zhong et al. in 2022 and is a nature-inspired metaheuristic optimization algorithm[8]. The design of this algorithm is inspired by the natural behavioral patterns of beluga whales in their habitat, especially their characteristics in predation, group cooperation, and ecological adaptation. As highly social marine mammals, beluga whale groups typically migrate and hunt through coordinated behaviors such as synchronized swimming and mirror swimming. These behaviors not only demonstrate their group cooperation ability but also optimize the efficiency of exploring food resources in vast environments. During the hunting process, beluga whales employ highly flexible strategic hunting methods, enabling the group to cooperatively encircle prey and increase the success rate of capture. This characteristic provides insights for the global search phase in BWO, allowing the algorithm to conduct extensive exploration in complex search spaces and avoid getting trapped in local optima. The phenomenon of whale falls, where a whale carcass sinks to the ocean floor after death, providing nutrients to deep-sea ecosystems, further inspired the design of BWO. Whale falls not only reveal the regeneration and recycling mechanisms of ecological resources but also demonstrate the system's ability to achieve self-repair through random adjustments when facing challenges, as evidenced by information dissemination and position updates after individual deaths.In BWO, this mechanism is simulated as a random perturbation strategy, allowing the algorithm to introduce moderate randomness during the search process to enhance global convergence and prevent falling into local optima.

1) Population Initialization and Stage Transition Strategy

At the beginning of the algorithm, randomly initialize the initial position matrix of beluga whales. This matrix can be represented as:

$$X=lb+rand(n,nd)\times(ub-lb). \qquad (3)$$

Where X represents the initial position matrix of beluga whales. lb represents the minimum value of each dimension in the search space, the lower bound of the variables. ub represents the maximum value of each dimension in the search space, i.e., the upper bound of the variables. n represents the population size of beluga whales. nd represents the dimension of variables. $rand(n,nd)$ generates a matrix of size $n\times nd$ with elements uniformly distributed between 0 and 1.

During the optimization process, it is necessary to divide the algorithm into exploration and exploitation stages based on a balance factor, which is expressed as follows:

$$B_f=b0_i\times(1-t/2\times t_{\max}),b0_i\in(0,1). \qquad (4)$$

Where $b0_i$ is a random number in the range $[0,1]$, t is the current iteration number, and $t_{\max}$ is the maximum number of iterations. When the balance factor B_f is greater than 0.5, the algorithm enters the exploration stage; when the value of B_f is less than or equal to 0.5, the algorithm enters the exploitation stage.

2) Exploration Stage

When the value of B_f is greater than 0.5, the algorithm enters the exploration stage. The formula for updating the position of beluga whales during the exploration stage is as follows:

$$\begin{aligned} X_{i,j}^{t+1} &= X_{i,p_j}^t + (X_{r,p_1}^t - X_{i,p_j}^t)(1+r_1)\sin(2\pi r_2),\ j = even. \\ X_{i,j}^{t+1} &= X_{i,p_j}^t + (X_{r,pi}^t - X_{i,p_j}^t)(1+r_1)\cos(2\pi r_2),\ j = odd. \end{aligned} \tag{5}$$

Where $X_{i,j}^{t+1}$ is the updated position of the i beluga whale in dimension j . $p_j(j=1,2,...d)$ represents a random integer in dimension d is the position of the beluga whale in dimension j . X_{r,p_1}^t and X_{i,p_j}^t represent the current positions of the i and r beluga whales, respectively. r_1 and r_2 are random numbers in the range [0,1]. $\sin(2\pi r_2)$ and $\cos(2\pi r_2)$ indicate whether the beluga whale fins are facing the water surface, and based on the parity of the dimension, determine which exploration formula to use.

3) Exploitation Stage

During the exploitation stage, the beluga whale population shares their position information to facilitate foraging behavior. The Levy flight strategy is employed in the algorithm to capture prey, and the specific formula is as follows:

$$X_i^{t+1} = r_3 X_{best}^t - r_4 X_i^t + C_1 \times L_F \times (X_r^t - X_i^t). \tag{6}$$

$$C_1 = 2r_4(1 - t / t_{\max}). \tag{7}$$

Where X_i^{t+1} represents the updated position of the i beluga whale, and X_{best}^t is the best position of the beluga whale population. X_r^t and X_i^t represent the current positions of the r and i beluga whales, respectively. r_3 and r_4 are random numbers in the range (0,1), C_1 is a parameter that measures the random jump intensity in the Levy flight strategy. L_F represents the Levy flight function.

4) Whale Fall Phase

In the whale fall phase, the position of the beluga whale population is updated based on their locations and step sizes. The equation for the whale fall is as follows:

$$X_i^{t+1} = r_5 X_i^t - r_6 X_r^t + r_7 X_{step}. \tag{8}$$

where r_5 and r_6 are random numbers in the range $(0,1)$, and X_{step} represents the step size of the whale fall, which is defined as follows.

3. Proposed method

The original BWO algorithm primarily relies on the position update mechanism of individual beluga whales to explore and exploit the search space. However, this algorithm lacks an effective mechanism to maintain population diversity, which can lead to premature convergence to local optima in the early stages of the search, thereby affecting the algorithm's global search capability and overall optimization performance.To address this issue, this study introduces the concept of individual age and survival probability to simulate the survival conditions of beluga whales at different life stages. The age of an individual is linked to the optimal position within the population. Older individuals tend to rely more on the global best position for updates, accelerating convergence, whereas younger individuals retain a higher exploratory capability to maintain population diversity.This section presents an enhanced algorithm based on the Beluga Whale Optimization Algorithm, referred to as YOBWO. The key differences between the original BWO and YOBWO are as follows.

1) Survival Strategy

Based on the natural environment and life cycle of beluga whales, this study categorizes their growth into three stages: juvenile stage (0–7 years), prime stage (7–30 years), and old age stage (30–40 years). In both the juvenile and old age stages, beluga whales are more vulnerable due to their small size or physical weakness, making them more susceptible to predation and leading to their demise during foraging.To more accurately model the life cycle of beluga whales, this study adopts a Weibull[9] distribution-based survival probability model, defined by the following equation:

$$P(a) = e^{-(\frac{a}{\lambda})^k}. \tag{9}$$

where $P(a)$ represents the survival probability of a beluga whale, k is the shape parameter (with a value greater than 0) that determines the trend of mortality rate changes, and λ is the scale parameter (with a value greater than 0). A larger λ indicates that survival probability changes more gradually with age, while a smaller λ suggests more rapid variations in survival probability. Based on the beluga whale life cycle, it can be further subdivided into different stages as follows.

$$P(a) = \begin{cases} e^{-(\frac{a}{\lambda_1})^{k_1}}, & 0 \le a \le a_1 \\ e^{-(\frac{a-a_1}{\lambda_2})^{k_2}} \times P(a_1), & a_1 < a \le a_2 \\ e^{-(\frac{a-a_2}{\lambda_3})^{k_3}} \times P(a_2), & a_2 < a \le a_3 \\ 0, & a > a_3 \end{cases}. \tag{10}$$

Where $a_1 = 7$, $a_2 = 30$, $a_3 = 40$ correspond to the juvenile stage, prime stage, and old age stage, respectively. The parameters λ_i and k_i represent the scale and shape parameters for each stage.

During each iteration, the age a_i of an individual increases by one unit:

$$a_i^{(t+1)} = a_i^{(t)} + \Delta t. \quad (11)$$

where Δt is the time step, set to 1.Next, based on the survival probability function $P(a_i^{(t+1)})$, the survival of individual iii is determined probabilistically. For each individual, a random number $r_i \in [0,1]$ is generated within the range [0,1]. If $r_i > P(a_i^{(t+1)})$, individual i is considered preyed upon, in which case its position X_i is reinitialized, and its age is reset to $a_i = 0$.

2) Optimal Position Update Strategy

To further enhance the global search capability and convergence speed of the algorithm, this study establishes a relationship between an individual's age and the optimal position within the population. This association strategy integrates the individual's age with the exploration strategy of the original BWO algorithm, allowing individuals at different life cycle stages to exhibit diverse behaviors in the search process, thereby improving the overall performance of the algorithm. The exploration strategy of the original BWO algorithm primarily relies on random interactions and position updates among individuals. Specifically, individuals adjust their positions by interacting with randomly selected peers, utilizing sine and cosine functions for fine-tuning, which facilitates the exploration of the search space. However, when population diversity is low, this strategy tends to fall into local optima, limiting the algorithm's effectiveness. To address this issue, the improved age-based optimal position update strategy dynamically adjusts an individual's exploration and exploitation behaviors based on age. Older individuals (in the old-age stage) rely more on the global best position for exploitation, accelerating convergence. Younger individuals (in the juvenile stage) maintain a higher level of exploration ability, promoting population diversity. This association strategy is mathematically formulated as follows:

When $a_i^t \le a_{threshold}$, the individual is classified as a young individual. At this stage, the algorithm tends to enhance the exploration phase, expanding the search range through large-scale random perturbations and differential strategies to prevent premature convergence to local optima. The specific position update equations are as follows:

$$X_{i,j}^{t+1} = X_{i,pj}^t + \eta\delta_{i,pj}(X_{r,pi}^t - X_{k,p_j}^t)\sin(2\pi\beta_{i,j}) + \gamma\beta_{i,j}^{'}(X_{best,pj}^t - X_{i,pj}^t),\ j = even. \quad (12)$$

$$X_{i,j}^{t+1} = X_{i,pj}^t + \eta\delta_{i,pj}(X_{r,pi}^t - X_{k,p_j}^t)\cos(2\pi\beta_{i,j}) + \gamma\beta_{i,j}^{'}(X_{best,pj}^t - X_{i,pj}^t),\ j = odd. \quad (12)$$

Where η is the primary scaling factor for exploration step size. $X_{r,pj}^t$ and X_{k,p_j}^t are values randomly selected from the population in dimension j for two different individuals ($r \ne i, k \ne i$). $\delta_{i,pj}$, $\beta_{i,j}$, $\beta_{i,j}^{'}$ are random numbers in the range $[0,1]$, independently sampled. γ is a small optimal attraction coefficient, preventing excessive divergence of young individuals.

When $a_i^t > a_{threshold}$, the individual is classified as an older individual, and the algorithm enters the exploitation phase. At this stage, the search range is quickly narrowed by relying more on the global best solution or the second-best solution, thereby accelerating convergence toward the optimal solution. The position update equation is as follows:

$$X_{i,j}^{t+1} = X_{i,pj}^t + \alpha\delta_{i,pj}(X_{best,pj}^t - X_{i,pj}^t). \quad (13)$$

Where α is the optimal coefficient that controls the attraction towards the best solution. This coefficient adapts dynamically during iterations. $\delta_{i,pj}$ is a random number in the range [0,1], ensuring that disturbances vary across different dimensions and individuals. $X_{best,pj}^t$ represents the current global best solution in dimension j.

4. Experimental Settings

4.1. Dataset

The Screw Compressor Oil Temperature (SCOT) dataset is specifically designed for research in long sequence time-series forecasting (LSTF) related to industrial equipment monitoring. It comprises two years of data, collected from multiple operational screw compressors in real-world industrial environments. The dataset includes the target variable "oil temperature" (OT) along with several operational parameters such as discharge pressure, suction pressure, rotational speed, and cooling system status, providing valuable insights into the thermal dynamics and overall operational status of the screw compressor system. To accommodate various research needs, the SCOT dataset offers subsets with different sampling frequencies, allowing for both short-term fluctuation analysis and long-term trend predictions. These subsets are widely utilized in training and evaluating time-series forecasting models, particularly in areas such as compressor performance optimization, fault detection, and predictive maintenance. The dataset plays a crucial role in advancing intelligent industrial monitoring, enabling more efficient and reliable operation of screw compressors in refrigeration, HVAC, and other industrial applications.

4.2. Mean Value Correction

In data preprocessing, outliers can significantly impact statistical analysis and model results. To minimize the interference of outliers on overall data analysis, the mean value correction method is commonly used. This method is suitable for scenarios where the number of outliers is small, and the variation in adjacent data points is relatively stable. Before applying the mean value correction method, it is essential to confirm that the detected outliers result from measurement errors or noise rather than being intrinsic characteristics of the data, to avoid mistakenly removing valuable information. In practical implementation, statistical methods or visualization techniques are first used to identify outliers in the dataset. For each detected outlier, the mean of the adjacent data points is computed and used to replace the outlier, thereby smoothing fluctuations in the data. This approach helps improve the overall quality of the dataset, ensuring the reliability of subsequent analysis and model training.

4.3. Evaluation Metrics

To assess the performance of the proposed model, three commonly used evaluation metrics are employed: Coefficient of Determination (R^2), Mean Squared Error (MSE), and Mean Absolute Error (MAE). These metrics provide insights into the model's accuracy, reliability, and overall predictive performance.

$$R^2 = 1 - \frac{\sum_{i=1}^{n}\left(y_i - \hat{y}_i\right)^2}{\sum_{i=1}^{n}\left(y_i - \overline{y}_i\right)^2}. \tag{14}$$

$$MSE = \frac{1}{n}\sum_{i=1}^{n}\left(y_i - \hat{y}_i\right)^2. \tag{15}$$

$$MAE = \frac{1}{n}\sum_{i=1}^{n}\left|y_i - \hat{y}_i\right|. \tag{16}$$

Where y_i is the actual value, $\hat{y}_i$ is the predicted value, $\overline{y}_i$ is the mean of the actual values, n is the number of observations.

5. Experimental Results and Analyses

5.1. Algorithm Performance Evaluation

To validate the effectiveness of the proposed improvement strategy, this study conducts a comprehensive comparison of 23 classical benchmark functions against other mainstream optimization algorithms. The test functions are categorized into unimodal, multimodal, and fixed-dimension test functions. Unimodal test functions have a single global optimum and are mainly used to evaluate the algorithm's convergence speed and precision, while multimodal test functions contain multiple local optima, testing the algorithm's ability to escape local optima and search for the global optimum. Fixed-dimension test functions are multimodal benchmark functions constructed through transformations such as rotation, shifting, and biasing, designed to assess the exploratory capability of an optimization algorithm in specific dimensional spaces and its ability to avoid local optima. To ensure a fair comparison, the proposed Beluga Whale Optimization Algorithm (BWO) and its improved version (YOBWO) are evaluated alongside five other optimization algorithms: Sparrow Search Algorithm (SSA)[10], Northern Goshawk Optimization (NGO)[11], Whale Optimization Algorithm (WOA)[12], Zebra Optimization Algorithm (ZOA)[13], and Grey Wolf Optimizer (GWO)[14]. To minimize the impact of randomness in a single experiment, all algorithms are set with a population size of 30 and run for 1000 iterations, with each optimization algorithm performing 50 independent experiments. The best value, mean value, and standard deviation of the results are recorded as evaluation criteria. The test results are presented in Tables 1–3, with the best-performing results highlighted in bold. The evaluation standard is based on the proximity of the obtained results to the theoretical values of the benchmark functions, where a smaller deviation indicates better algorithm performance.

Table 1. Comparison Results of Unimodal Test Functions

Function	Evaluation Metric	SSA	BWO	NGO	WOA	ZOA	GWO	YOBWO
	Optimal Value	0.00E+00	0.00E+00	4.51E+03	1.52E-300	0.00E+00	3.67E-68	**0.00E+00**
F1	Mean Value	2.21E-21	0.00E+00	8.10E+03	6.39E-284	0.00E+00	4.41E-65	**0.00E+00**
	Standard Deviation	9.79E-21	0.00E+00	2.68E+03	0.00E+00	0.00E+00	9.04E-65	**0.00E+00**
	Optimal Value	0.00E+00	3.34E-270	2.91E+01	4.15E-169	7.04E-295	6.85E-40	**0.00E+00**
F2	Mean Value	5.95E-11	2.02E-261	4.72E+01	1.13E-160	1.72E-287	2.01E-38	**0.00E+00**
	Standard Deviation	2.17E-10	0.00E+00	1.18E+01	6.66E-160	0.00E+00	3.01E-38	**0.00E+00**
	Optimal Value	3.59E-149	0.00E+00	9.26E+03	1.34E+01	0.00E+00	9.70E-15	**0.00E+00**
F3	Mean Value	1.82E-16	0.00E+00	1.79E+04	1.00E+04	0.00E+00	1.69E-08	**0.00E+00**
	Standard Deviation	6.78E-16	0.00E+00	6.92E+03	5.58E+03	0.00E+00	5.63E-08	**0.00E+00**
	Optimal Value	8.04E-252	1.39E-263	2.34E+01	3.08E+00	2.02E-268	7.93E-13	**0.00E+00**
F4	Mean Value	4.97E-12	4.34E-254	3.39E+01	7.49E+00	3.44E-262	3.70E-11	**0.00E+00**
	Standard Deviation	1.94E-11	0.00E+00	4.89E+00	1.91E+01	0.00E+00	6.76E-11	**0.00E+00**
	Optimal Value	1.45E-09	2.50E+01	1.55E+06	2.30E+01	2.70E+01	1.45E-09	**1.02E-09**
F5	Mean Value	1.34E-06	2.60E+01	5.17E+06	2.50E+01	2.84E+01	1.34E-06	**0.92E-06**
	Standard Deviation	1.35E-06	4.99E-01	3.17E+06	6.57E-01	4.78E-01	1.35E-06	**0.91E-06**
	Optimal Value	6.70E-06	1.50E-06	4.36E+03	1.05E-05	2.49E+00	3.08E+00	**3.15E-13**
F6	Mean Value	2.08E-01	4.40E-05	8.15E+03	7.82E-05	3.43E+00	4.94E+00	**8.69E-10**
	Standard Deviation	2.32E-01	5.46E-05	2.11E+03	8.80E-05	4.55E-01	7.66E-01	**7.77E-10**
	Optimal Value	1.05E-05	4.49E-06	7.44E-01	3.75E-05	7.26E-06	5.62E-04	**1.25E-06**
F7	Mean Value	2.144E-04	6.11E-05	2.53E+00	8.95E-04	5.25E-05	1.65E-03	**3.73E-05**
	Standard Deviation	1.11E-04	4.43E-05	1.12E+00	8.85E-04	2.87E-05	6.65E-03	**3.76E-05**

Table 2. Comparison Results of Multimodal Test Functions

Function	Evaluation Metric	SSA	BWO	NGO	WOA	ZOA	GWO	YOBWO
	Optimal Value	-8.29E+03	**-1.25E+04**	-5.25E+03	-1.25E+04	-7.46E+03	-1.23E+04	-9.81E+03
F8	Mean Value	-6.65E+03	**-1.25E+04**	-3.86E+03	-1.14E+04	-5.39E+03	-8.34E+03	-6.62E+03
	Standard Deviation	1.08E+03	**1.03E+00**	6.33E+02	1.59E+03	7.10E+02	2.61E+03	1.18E+03
	Optimal Value	0.00E+00	0.00E+00	1.54E+02	0.00E+00	0.00E+00	0.00E+00	**0.00E+00**
F9	Mean Value	0.00E+00	0.00E+00	2.20E+02	8.81E+00	0.00E+00	1.29E+01	**0.00E+00**
	Standard Deviation	0.00E+00	0.00E+00	2.36E+01	2.99E+01	0.00E+00	8.52E+00	**0.00E+00**
	Optimal Value	4.44E-16	4.44E-16	1.17E+01	4.44E-16	4.44E-16	3.99E-15	**2.41E-16**
F10	Mean Value	8.54E-12	4.44E-16	1.45E+01	1.72E-15	1.86E-15	7.40E-15	**2.41E-16**
	Standard Deviation	3.22E-11	0.00E+00	1.17E+00	1.70E-15	1.74E-15	6.96E-16	**0.00E+00**
	Optimal Value	0.00E+00	0.00E+00	3.43E+01	0.00E+00	0.00E+00	0.00E+00	**0.00E+00**
F11	Mean Value	0.00E+00	0.00E+00	7.46E+01	0.00E+00	0.00E+00	4.34E-03	**0.00E+00**
	Standard Deviation	0.00E+00	0.00E+00	2.44E+01	0.00E+00	0.00E+00	9.55E-03	**0.00E+00**
	Optimal Value	**1.08E-13**	2.49E-06	1.37E+03	6.04E-06	1.21E-01	8.47E-07	1.16E-13
F12	Mean Value	4.95E-10	9.50E-05	1.04E+06	6.79E-02	3.03E-01	1.69E-02	**3.62E-10**
	Standard Deviation	5.06E-10	1.21E-04	1.30E+06	3.74E-01	8.93E-01	2.19E-02	**1.48E-10**
	Optimal Value	**1.61E-11**	7.54E-08	5.46E+05	2.97E-03	1.16E+00	2.40E-05	1.93E-11
F13	Mean Value	7.10E-09	1.68E-05	8.12E+06	8.45E-01	2.01E+00	2.19E-01	**2.93E-09**
	Standard Deviation	9.75E-09	2.95E-05	6.73E+06	3.21E+00	3.71E-01	1.41E-01	**2.17E-10**

Table 3. Comparison Results of Fixed-Dimension Multimodal Test Functions

Function	Evaluation Metric	SSA	BWO	NGO	WOA	ZOA	GWO	YOBWO
	Optimal Value	9.98E-01	9.98E-01	9.98E-01	9.98E-01	9.98E-01	9.98E-01	**9.98E-01**
F14	Mean Value	9.35E+00	**9.98E-01**	9.34E+00	2.93E+00	3.28E+00	1.17E+00	1.09E+00
	Standard Deviation	4.58E+00	**1.11E-16**	6.10E+00	3.78E+00	2.36E+00	5.13E-01	3.57E-01
	Optimal Value	3.07E-04	3.07E-04	4.39E-04	3.07E-04	3.07E-04	3.07E-04	**3.07E-04**
F15	Mean Value	3.07E-04	3.55E-04	1.60E-02	4.35E-04	1.65E-03	4.41E-03	**3.10E-04**
	Standard Deviation	5.43E-07	6.41-05	1.73E-02	3.99E-04	4.85E-03	7.97E-03	**2.97E-04**
	Optimal Value	-1.03E+00	-1.03E+00	-1.03E+00	-1.03E+00	-1.03E+00	-1.03E+00	**-1.03E+00**
F16	Mean Value	-1.03E+00	-1.03E+00	-1.03E+00	-9.98E-01	-1.03E+00	-1.03E+00	**-1.03E+00**
	Standard Deviation	1.16E-08	**6.28E-17**	7.01E-06	1.59E-01	4.38E-08	2.87E-09	1.25E-16
	Optimal Value	3.97E-01	3.97E-01	3.97E-01	3.97E-01	3.97E-01	3.97E-01	**3.97E-01**
F17	Mean Value	3.97E-01	3.97E-01	3.97E-01	3.97E-01	3.97E-01	3.97E-01	**3.97E-01**
	Standard Deviation	6.11E-07	2.10E-14	8.56E-05	5.96E-15	5.13E-05	1.78E-05	**5.55E-17**
	Optimal Value	3.00E+00	3.00E+00	3.00E+00	3.00E+00	3.00E+00	3.00E+00	**3.00E+00**
F18	Mean Value	3.00E+00	3.00E+00	3.54E+00	6.78E+00	3.54E+00	4.62E+00	**3.00E+00**
	Standard Deviation	3.71E-07	1.72E-15	3.77E+00	9.36E+00	3.77E+00	1.13E+01	**8.79E-16**
	Optimal Value	-3.86E+00	-3.86E+00	-3.86E+00	-3.86E+00	-3.86E+00	-3.86E+00	**-3.86E+00**
F19	Mean Value	-3.84E+00	-3.86E+00	-3.78E+00	-3.81E+00	-3.86E+00	-3.86E+00	**-3.86E+00**
	Standard Deviation	1.08E-01	3.07E-16	1.66E-01	1.83E-01	1.13E-03	1.76E-03	**6.58E-15**
	Optimal Value	-3.32E+00	-3.32E+00	-3.28+00	-3.32E+00	-3.32E+00	-3.32E+00	**-3.32E+00**
F20	Mean Value	-3.25E+00	-3.39E+00	-2.94E+00	-3.27E+00	-3.32E+00	-3.26E+00	**-3.47E+00**
	Standard Deviation	6.32E-02	5.06E-02	2.78E-01	5.86E-02	7.86E-02	6.56E-02	**5.24E-02**
	Optimal Value	-1.01E+01	-1.01E+01	-1.01E+01	-1.01E+01	-1.01E+01	-1.10E+01	**-1.01E+01**
F21	Mean Value	-9.84E+00	-1.01E+01	-6.42E+00	-7.61E+00	-8.63E+00	-9.34E+00	**-5.79E+00**
	Standard Deviation	1.21E+00	8.73E-10	2.34E+00	2.82E+00	2.07E+00	1.85E+00	**1.15E+00**
	Optimal Value	-1.04E+01	-1.04E+01	1.04E+01	-1.04E+01	-1.04E+01	-1.04E+01	**-1.04E+01**
F22	Mean Value	-1.02E+01	-1.04E+01	-5.19E+00	-6.40E+00	-7.00E+00	**-1.00E+01**	-8.37E+00
	Standard Deviation	7.44E-01	**3.25E-09**	3.00E+00	3.12E+00	2.68E+00	1.25E+00	2.38E+00
	Optimal Value	-1.05E+01	-1.05E+01	-1.05E+01	-1.05E+01	-1.05E+01	-1.05E+01	**-1.05E+01**
F23	Mean Value	-1.05E+01	1.05E+01	-5.86E+00	-6.29E+00	-7.88E+00	-1.05E+01	**-1.31E+01**
	Standard Deviation	4.77E-05	**3.49E-09**	3.28E+00	3.33E+00	2.76E+00	9.09E-05	2.35E+00

From the above results, it can be clearly observed that the proposed method outperforms the other compared algorithms in multiple aspects. Specifically, the improved algorithm achieves a lower mean error, smaller standard deviation, and results that are consistently closer to the theoretical optimal values of the benchmark functions. This demonstrates the algorithm's superior ability to balance exploration and exploitation, enabling it to converge more efficiently while avoiding local optima. Additionally, compared to traditional optimization algorithms, the proposed approach shows significant improvements in both unimodal and multimodal function tests, highlighting its robustness and adaptability to different types of optimization problems. These results confirm the effectiveness of the enhancements introduced in this study, further validating the advantages of the proposed method over existing state-of-the-art optimization algorithms.

5.2. Experimental Results of the YOBWO-TCN-AM Model

The experimental results of the YOBWO-optimized TCN-AM model demonstrate its superior performance in predicting the lubrication oil temperature of screw compressors. Compared to traditional optimization methods, the proposed approach significantly improves prediction accuracy by effectively fine-tuning the hyperparameters of the TCN-AM model through the enhanced YOBWO algorithm. The results indicate that the optimized model achieves lower Mean Squared Error, reduced Mean Absolute Error, and a higher

Coefficient of Determination, confirming its enhanced generalization ability and robustness. As shown in Table 4, the YOBWO-optimized model consistently outperforms other comparative methods across multiple evaluation metrics. By leveraging the improved exploration-exploitation balance in YOBWO, the model is able to better capture complex time-series patterns, ensuring more stable and reliable predictions under varying operating conditions. These findings highlight the effectiveness of the proposed optimization strategy in improving the efficiency and accuracy of deep learning-based temperature prediction models for industrial applications.

Table 4. Experimental Results

Models	R^2	MSE	MAE
TCN	0.9393	0.0219	0.1356
TCN-AM	0.9278	0.0261	0.1536
YOBWO-TCN-AM	0.9548	0.0204	0.1251

6. Conclusion

In this study, a YOBWO-optimized TCN-AM model was proposed for the prediction of lubrication oil temperature in screw compressors. By integrating the improved Beluga Whale Optimization Algorithm (YOBWO) with the Temporal Convolutional Network (TCN) and Attention Mechanism (AM), the model effectively enhances prediction accuracy and robustness. The improvements introduced in YOBWO, including the survival strategy and optimal position update mechanism, significantly enhance its global search capability and convergence speed, enabling more effective hyperparameter optimization for deep learning models. Comprehensive experiments were conducted using 23 classical benchmark functions and real-world lubrication oil temperature datasets. The results demonstrate that the proposed YOBWO-optimized TCN-AM model consistently outperforms other state-of-the-art optimization and deep learning models in terms of Mean Squared Error, Mean Absolute Error, and Coefficient of Determination. The model exhibits strong generalization ability and effectively captures complex time-series patterns in industrial applications. The findings of this study highlight the potential of integrating advanced optimization algorithms with deep learning models for predictive maintenance and intelligent monitoring in industrial systems. Future work will focus on extending this framework by incorporating multi-source sensor fusion and further optimizing the loss function and model architecture to enhance predictive performance under diverse operating conditions. Additionally, real-time deployment in industrial settings will be explored to validate the model's practical applicability and efficiency in real-world operations.

References

[1] Yang S, Ouyang H, Wu Y, et al. Experimental study of lubricating oil impact on pressure pulsation for twin-screw refrigeration compressor[J]. International Journal of Refrigeration, 2020, 112: 324-332.

[2] Nellis G, Klein S. Heat transfer[M]. Cambridge university press, 2008.

[3] Turns S R, Pauley L L. Thermodynamics: concepts and applications[M]. Cambridge University Press, 2020.

[4] Bengio Y, Courville A, Vincent P. Representation learning: A review and new perspectives[J]. IEEE transactions on pattern analysis and machine intelligence, 2013, 35(8): 1798-1828.

[5] Bai S, Kolter J Z, Koltun V. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling[J]. arXiv preprint arXiv:1803.01271, 2018.

[6] Vaswani A. Attention is all you need[J]. Advances in Neural Information Processing Systems, 2017.

[7] Bergstra J, Bengio Y. Random search for hyper-parameter optimization[J]. Journal of machine learning research, 2012, 13(2).

[8] Zhong C, Li G, Meng Z. Beluga whale optimization: A novel nature-inspired metaheuristic algorithm[J]. Knowledge-Based Systems, 2022, 251: 109215.

[9] Hallinan Jr A J. A review of the Weibull distribution[J]. Journal of quality technology, 1993, 25(2): 85-93.

[10] Xue J, Shen B. A novel swarm intelligence optimization approach: sparrow search algorithm[J]. Systems science & control engineering, 2020, 8(1): 22-34.

[11] Dehghani M, Hubálovský Š, Trojovský P. Northern goshawk optimization: a new swarm-based algorithm for solving optimization problems[J]. Ieee Access, 2021, 9: 162059-162080.

[12] Mirjalili S, Lewis A. The whale optimization algorithm[J]. Advances in engineering software, 2016, 95: 51-67.

[13] Trojovská E, Dehghani M, Trojovský P. Zebra optimization algorithm: A new bio-inspired optimization algorithm for solving optimization algorithm[J]. Ieee Access, 2022, 10: 49445-49473.

[14] Mirjalili S, Mirjalili S M, Lewis A. Grey wolf optimizer[J]. Advances in engineering software, 2014, 69: 46-61.