Preprints.org

Article

High-Precision Temperature Compensation Method for Downhole Pressure Sensors in Oil and Gas Wells Based on the C-I-WOA-BP Neural Network

Fei Wang [*] , Xing Zhang , Guowang Gao , Xintong Li , Xuzhi He

Article

High-Precision Temperature Compensation Method for Downhole Pressure Sensors in Oil and Gas Wells Based on the C-I-WOA-BP Neural Network

Fei Wang [1,*], Xing Zhang [2], Guowang Gao [3], Xintong Li [4] and Xuzhi He [5]

[1] School of Electronic Engineering. Xi'an Shiyou University; 200102@xsyu.edu.com
[2] School of Electronic Engineering. Xi'an Shiyou University; 1642737056@qq.com
[3] School of Electronic Engineering. Xi'an Shiyou University; wwgao1205@163.com
[4] School of Mechanical and Electrical Engineering. Xi'an University of Architecture and Technology; 146847@qq.com
[5] School of Biomedical Engineering. University of California, Davis; xzhe@ucdavis.edu
[*] Correspondence: 200102@xsyu.edu.cn; Tel.:+86-13259816111

Abstract: The Internet of Things (IoT) aims to connect all objects and facilitate data interchange, with data accuracy being paramount. While IoT has seen extensive adoption in conventional sectors, its application in the petroleum industry, especially with downhole equipment, poses challenges due to the extreme high-temperature and high-pressure conditions, making accurate measurements elusive. This Paper we aim to enhance the accuracy of downhole pressure measurements in slim-hole drilling parameter monitoring, mitigate the temperature influences on strain gauges, and surmount the intrinsic limitations of the traditional BP neural network, such as sluggish convergence speeds, high sensitivity to initial weights and biases, unstable learning rates, and a tendency to fall into local minima. This study introduces a temperature compensation model using a chaotic-initiated adaptive whale optimization algorithm (C-I-WOA) to optimize a BP neural network, which is henceforth referred to as the C-I-WOA-BP model. Initially, we harnessed chaotic mapping techniques to refine the initialization process of the whale optimization algorithm (WOA) population, thereby bolstering the heterogeneity of the algorithm. Subsequently, we integrated an adaptive weight recalibration mechanism, refined the encompassing methodology of the WOA, and bolstered its capabilities for expansive global scouting and meticulous local refinement in the solution arena. Subsequently, the WOA was used to optimize the weight coefficients of the BP neural network. Finally, we conducted a detailed comparative analysis by juxtaposing the performance metrics of the BP, C-WOA-BP, I-WOA-BP, and C-I-WOA-BP networks. The experimental results indicate that the C-I-WOA-BP model demonstrates faster convergence speed, superior global search capabilities, and enhanced local exploitation potential compared to the other models. Additionally, it provides a more stable and accurate reduction in temperature-induced errors in drilling pressure parameter measurements.

Keywords: BP neural network; temperature compensation model; chaotic mapping; whale optimization algorithm; adaptive weight

1. Introduction

Recently, small-borehole drilling technology has gained popularity in the oil industry due to the rising demand for oil and gas and the increasing cost of conventional borehole extraction. This technology offers advantages such as fewer consumables, lower exploration costs, and reduced environmental pollution [1]. Ensuring the accuracy of drilling-pressure measurements is a pivotal challenge in slim-hole drilling technology. In practical applications, extreme high-temperature downhole conditions lead to nonlinear errors in drilling pressure measurements, necessitating temperature compensation.

Compared to high costs, operational complexities, and limited flexibility of hardware compensation, software compensation is advantageous in terms of low cost, high flexibility, and

robust customization capabilities. However, the different software compensation methods have limitations. For example, linear interpolation can generate significant errors outside the sampled points. Polynomial fitting requires high-order polynomials for better data fitting of nonlinear sensors, leading to overfitting. Lookup table compensation relies on an extensive sample dataset and computational resources, and its ability to compensate for data in unknown temperature ranges is limited [2]. Meanwhile, the BP neural network technology can adaptively learn the nonlinear characteristics of sensors, offering exceptional flexibility and adaptability. However, this is constrained by issues such as slow convergence and the tendency to become trapped in the local minima [3,4]. To address these challenges, this study introduces a C-I-WOA-BP neural network model. It integrates a chaotic whale optimization algorithm (CWOA) with a multistrategy integrated whale optimization algorithm (IWOA) to further enhance the BP neural network. This combined approach leverages the strengths of the three algorithms, resulting in faster convergence rates, heightened global exploration capabilities, enhanced exploitation abilities, and superior stability.

2. Related Works

2.1. Numerical Analysis Methods

In numerical analysis, various techniques are employed, including cubic spline interpolation, the least squares method, and the gradient descent method. Specifically, high-degree polynomial interpolation may not always deliver the desired outcomes when addressing the compensation for temperature variations due to drilling pressure. Consequently, practitioners resort to piecewise interpolation. This approach segments the interpolation range into numerous subintervals by leveraging a polynomial of a lower degree for interpolation within each segment. A conspicuous limitation of conventional piecewise interpolation is the potential emergence of discontinuities or nonsmooth junctions at the extremities of these subintervals. However, this hiccup was elegantly circumvented using cubic spline interpolation. Characterized by piecewise cubic curves, this method guarantees second-order continuous derivatives at the interconnecting points, thereby ensuring seamless transitions at the junctures [5]. The term "least squares" in the least squares method stands for squaring; hence, it is also referred to as the method of minimum squares. This method serves as an essential tool in mathematical optimization modeling [6], with the primary intent of minimizing the cumulative squared discrepancies between observed and actual values by refining curve fitting. The gradient descent method is a pivotal optimization strategy for both numerical analysis and machine learning [7]. Its primary objective is to determine the minimal point of the loss function by iterative training and update the parameters of the machine-learning model to minimize this loss function [2,8]. In gradient descent, the "gradient" is a vector. For a singular function, the gradient encapsulates the derivative at a given point. For multivariate functions, the direction of the gradient indicates the steepest ascent. Consequently, the gradient descent method consistently pursues the counterdirection of the gradient in every iteration, ensuring swift convergence to the function's local nadir [8].

2.2. Neural Network Methods

Various strategies have been implemented in the field of neural-network enhancement. These include fusing particle swarm optimization (PSO) with backpropagation neural networks (PSO-BP), melding genetic algorithm (GA) with backpropagation neural networks (dubbed GA-BP), and merging ant colony optimization (ACO) with backpropagation neural networks (ACO-BP). The conceptual foundation of the PSO algorithm is based on the predation dynamics of avian flocks. Within this algorithmic framework, each particle is analogous to a bird in a flock, and its spatial position signifies a probable solution within the optimization landscape. In the BP neural network model for temperature compensation based on PSO, PSO was employed to globally search for the initial weights and thresholds of the BP neural network. Upon optimization, these weights and thresholds were seamlessly integrated into the BP network. This hybrid modality optimally harnesses the intrinsic search efficacy

of the BP neural network while simultaneously circumventing the pitfalls of local optima [3]. On the other hand, GAs emulate the intricate processes of natural selection and evolutionary biology, embodying a stochastic parallel search strategy. Using a predefined fitness function, the GA filters individuals based on the principles of selection, crossover, and mutation. Consequently, individuals with high fitness were retained, whereas those with low adaptability were excluded. The emergent population not only imbibes attributes from its antecedents but also showcases heightened fitness metrics. The objective of integrating GAs with BP neural networks is to leverage the GA to identify suitable initial weights and thresholds for the network. Finally, ACO is a heuristic anchored in the sustenance-seeking patterns of ant colonies. Ants chart trajectories governed by the pheromone gradient established by their predecessors. This autonomous foraging mechanism ensures diversity in the algorithm's search process, allowing it to escape the local optima and seek the global optima. Pairing this optimization technique with BP neural networks can further enhance the training accuracy of the network [9].

3. Principle of Drilling Pressure Measurement

The drilling pressure measurement was based on the strain effect of resistive strain gauges using a Wheatstone bridge for the measurement process. As the resistive strain gauge was subjected to the stress imposed by the drilling pressure, its resistance changed. Therefore, it is essential to use a Wheatstone bridge to convert this change in resistance into a readable electrical signal, thereby achieving an accurate measurement of the drilling pressure. Figure 1 illustrates the principles of the Wheatstone bridge. Owing to the harsh high-temperature conditions downhole, the resistive strain gauge exhibited nonlinear errors influenced by temperature. Temperature compensation is an indispensable step for ensuring the accuracy of drilling pressure measurements [10,11].

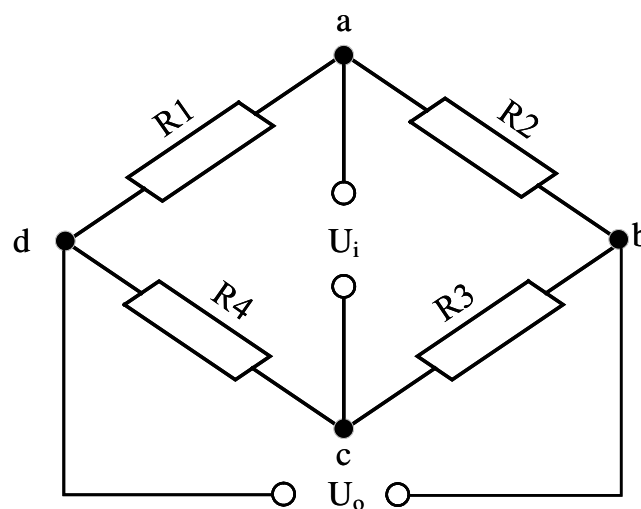**Figure 1.** Schematic diagram of Wheatstone bridge.

4. Principle of C-I-WOA-BP Neural Network

4.1. BP Neural Network

Owing to the nonlinear influence of temperature on resistive strain gauges, utilizing a BP neural network allows for a more precise modeling of this nonlinear relationship, thereby achieving effective compensation [12]. Figure 2 shows the BP neural network architecture designed for

temperature compensation, which comprises of input, hidden, and output layers. The input layer of the neural network incorporates the temperature Ut and output voltage Up from the drilling pressure measurement bridge. The desired output represents the actual force FN exerted by the drilling pressure, whereas the network output is the predicted drilling pressure, F_p. The hidden layer contains h neurons, the number of which is determined using the specific empirical formula h $=\sqrt{m+n}$+a, where m is the number of nodes in the input layer, n is the number of nodes in the output layer, and a is an integer ranging from 1 to 10 [13].
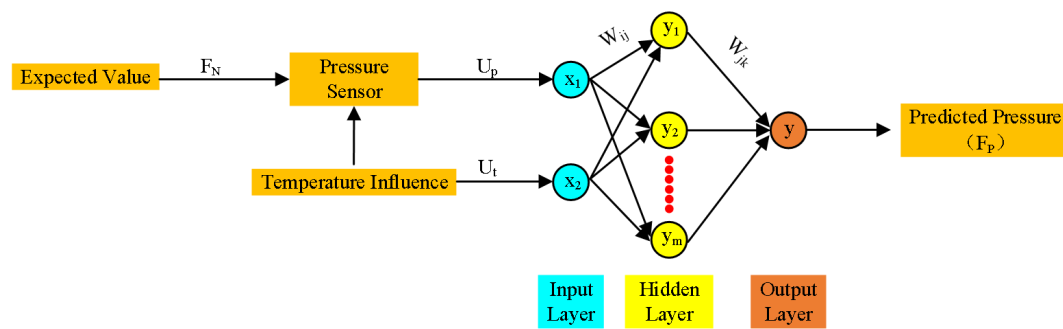

Figure 2. Structure of BP neural network for temperature compensation.

As shown in Figure 3 , the network achieved its minimum mean squared error when the number of neurons in the hidden layer was 5. Therefore, the 2-5-1 structure of the BP neural network is selected as the optimal configuration. W_{ij} represents the weight from the i-th neuron in the input layer to the j-th neuron in the hidden layer, and W_{jk} denotes the weight from the j-th neuron in the hidden layer to the k-th neuron in the output layer.
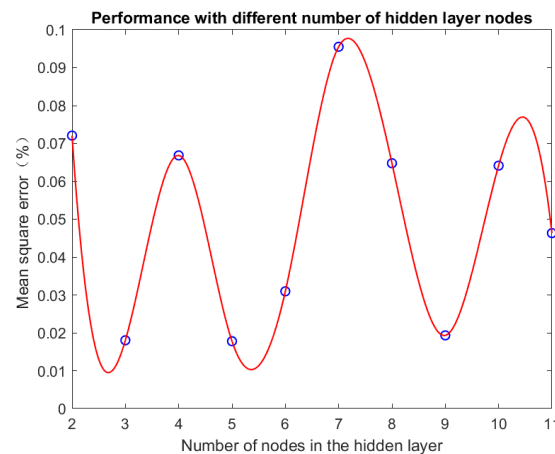
Figure 3. Performance with different number of hidden layer nodes.

4.2. BP Neural Network Based on Whale Optimization Algorithm

The WOA is a novel meta-heuristic optimization strategy inspired by the predatory behavior of humpback whales. Owing to their intuitive nature, ease of implementation, and the ability of meta-heuristic algorithms to operate without gradient information and avoid becoming trapped in local optima, they are gradually gaining widespread attention in engineering applications. Meta-heuristic algorithms can be broadly categorized into three main types [14], as illustrated in

Figure 4. Meta-heuristic algorithms can be broadly categorized into three main types, as shown in Figure 4. The first category includes intelligent optimization algorithms based on evolutionary principles, such as evolutionary strategies (ESs) and genetic algorithms (GAs). Next are intelligent optimization strategies grounded in the principles of physics, such as simulated annealing (SA) and black hole algorithm (BH). Lastly, there are intelligent optimization methods centered on swarm behaviors, including particle swarm optimization (PSO) and ant colony optimization (ACO), among others [15].
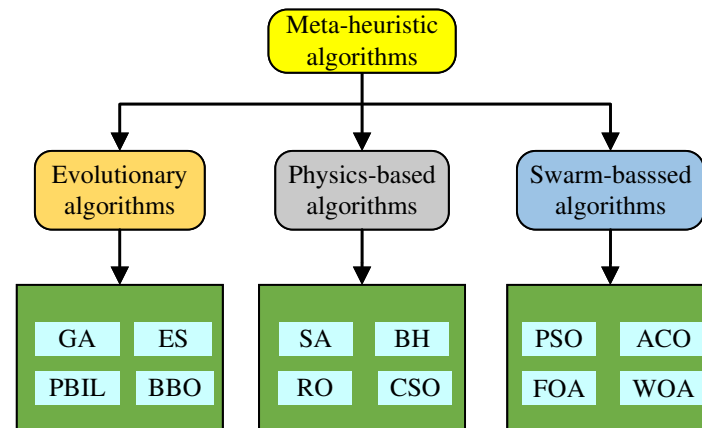

Figure 4. Classification of meta-heuristic algorithms.

The WOA is an intelligent optimization strategy based on swarm behavior. Its primary inspiration was drawn from the unique hunting techniques of humpback whales, such as encircling prey, bubble-net attacks (referred to as the exploitation phase), and searching for prey (also known as the exploration phase) [16,17]. By simulating these behaviors, we constructed a mathematical model of the WOA. The specific details of this model are as follows.

4.2.1. Encircling Prey

Encircling prey is based on a mathematical model constructed from the ability of humpback whales to identify and encircle gathered prey. This behavior can be described using the following mathematical expressions:

$$\vec{D} = \left| \vec{C} \cdot \vec{X}^*(t) - \vec{X}(t) \right|, \tag{1}$$

$$\vec{X}(t+1) = \vec{X}^*(t) - \vec{A} \cdot \vec{D}. \tag{2}$$

where t is the number of iterations, $\vec{D}$ is the distance vector between an individual whale and the global optimum solution, $\vec{X}^*(t)$ is the position vector of the whale with the best fit in the current iteration, and $\vec{X}(t)$ is the position vector of the whale during the current iteration. The coefficient vectors $\vec{A} and \vec{C}$ are used to adjust the search direction and step size.

The coefficient vectors $\vec{A}$ and $\vec{C}$ are expressed as:

$$\vec{A} = 2\vec{a} \cdot \vec{r}_1 - \vec{a}, \tag{3}$$

$$\vec{C} = 2 \cdot \vec{r}_2. \tag{4}$$

In the equation, $\vec{r}_1$ is a random vector in the [0,1] range, $\vec{r}_2$ is another random vector within the [0,1] range, and $\vec{a}$ is an adjustment coefficient that linearly decreases from 2 to 0 during the iteration process. The magnitude of $\vec{a}$ was determined using:

$$a = 2 - \frac{2t}{T_{\max}}. \tag{5}$$

where t is the current iteration number, and T_{max} is the maximum number of iterations.

Figure 5 illustrates the underlying mechanism of prey encircling in a two-dimensional space. The position (X, Y) of the whale in the current iteration is adjusted based on the optimal position (X^*, Y^*) of the iteration. This adjustment was achieved by modulating the values of $\vec{A}$ and $\vec{C}$ using the alteration vectors $\vec{r}_1$ and $\vec{r}_2$, respectively . Consequently, in the next iteration cycle, the whale can lie between the pivotal coordinates, as shown in Figure 5. At every iteration, the whale updates its position to approach the current optimal position, mimicking the behavior of encircling the prey. The same principle applies to the higher-dimensional scenarios. For instance, in an n-dimensional space, the whale moves around the current best position within the n-dimensional space. In practical applications, n represents the number of variables in the solution to a problem.
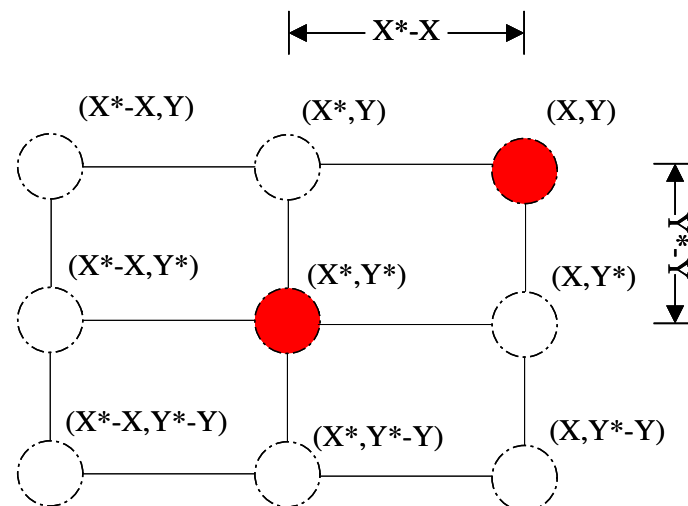

Figure 5. Schematic diagram of encircling prey (X^*, Y^* are the current optimal positions).

4.2.2. Bubble-net Attacking Method

When a humpback whale preys on its target, it employs encircling contractions and spiral updating to approach the prey. Mathematical models were designed for both the shrinking encircling mechanism and the spiral-updating position strategy to simulate the predatory behavior of the humpback whale.

One shrinking encircling mechanism: This behavior is achieved by gradually reducing the value of parameter $\vec{a}$ in Eq. 3 during the iteration process, where the value of $\vec{A}$ is set within the interval $[-a, a]$ and linearly decreases according to Eq. 5 throughout the iteration process. When the value of parameter $\vec{a}$ is within the range $[-1, 1]$, the position of the whale in the next iteration can be chosen at any point between the current whale position and the optimal position. As shown in Figure 6, in a two-dimensional space, when the value of parameter $\vec{A}$ is in the interval $[0, 1]$, the possible position of the whale will be within the area formed by points (X, Y) and (X^*, Y^*).

Two spiral updating position: As shown in Figure 7, this strategy first determines the distance between the whale at coordinate point (X, Y) and the current optimal position (X^*, Y^*).

Subsequently, a spiral equation was established between (X, Y) and (X^*, Y^*) to simulate humpback whale spiral-hunting behavior. The spiral equation is given by

$$\vec{X}(t+1) = \vec{D_i}' \cdot e^{bl} \cdot cos(2\pi l) + \vec{X}^*(t). \tag{6}$$

where $\vec{D_i}' = \vec{X}^*(t) - \vec{X}(t)$ represents the distance from the coordinates of the i-th whale to the optimal position, b is a constant used to define the logarithmic spiral contour, and l is a parameter randomly selected from the interval $[-1, 1]$.
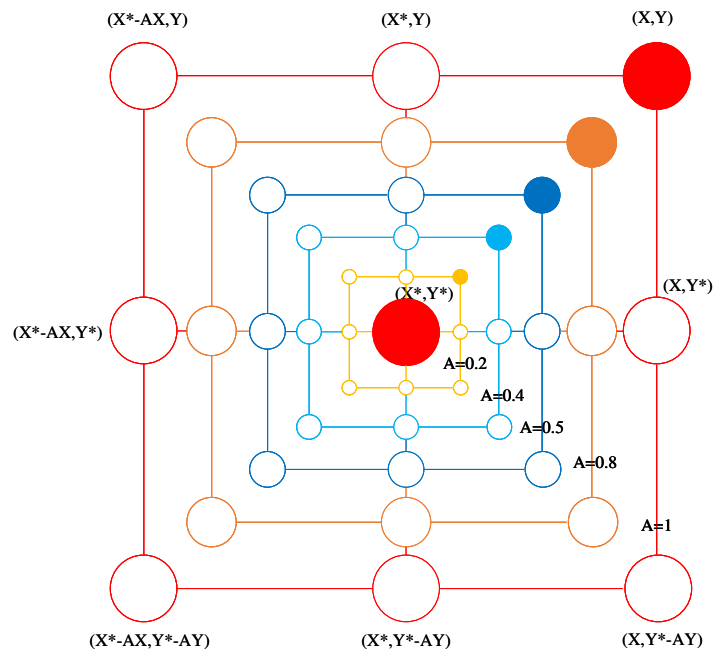

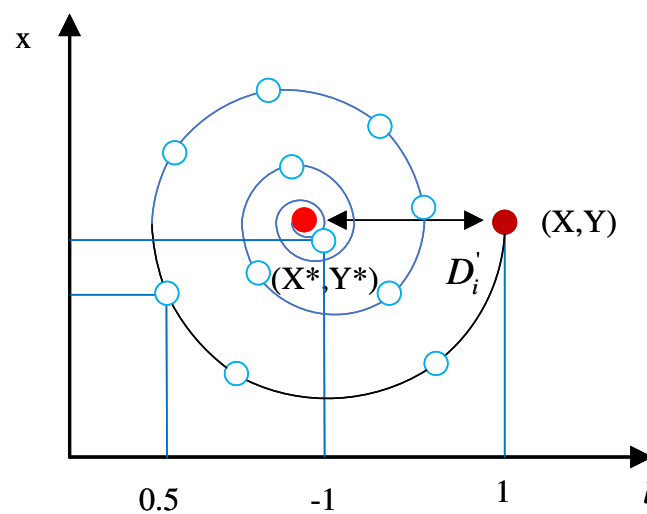**Figure 6.** Shrinking encircling mechanism.

Figure 7. Spiral updating position.

We assumed in the mathematical model that the probability of these two behaviors occurring was 50% to accurately simulate the humpback whale hunting process along a spiral path and the gradually narrowing circular swimming pattern. This was used to optimize the position of the whale during the iteration process. The mathematical model is as follows:

$$\vec{X}(t+1) = \begin{cases} \vec{X^*}(t) - \vec{A} \cdot \vec{D} & p < 0.5 \\ \vec{D_{i'}} \cdot e^{bl} \cdot cos(2\pi l) + \vec{X^*}(t) & p \geq 0.5 \end{cases}, \tag{7}$$

where p is a random number in the interval $[0, 1]$.

4.2.3. Search for Prey

In addition to the bubble-net attack method, whales engage in random searches for prey. This behavior is achieved based on the variation in the vector $\vec{A}$ when $|\vec{A}| \geq 1$ is used. Compared to the bubble-net attack strategy, during the random search phase, the whale's position $\vec{X}(t)$ is updated based on the position of the randomly selected whale $\vec{X_{rand}}$. At this juncture, the position update does not gravitate toward the randomly selected whale but rather moves away from it. The underlying rationale for this strategy is to expand the search scope and achieve broader exploration, thereby preventing premature convergence to the local optima. Consequently, the position of the whale deviates from that of the reference whale during $|\vec{A}| \geq 1$. The mathematical model is expressed as:

$$\vec{D} = \left| \vec{C} \cdot \vec{X_{\text{rand}}} - \vec{X} \right| \tag{8}$$

$$\vec{X}(t+1) = \vec{X_{\text{rand}}} - \vec{A} \cdot \vec{D}. \tag{9}$$

where $\vec{D}$ is the vector distance between the current whale position and the position of the randomly selected whale, and $\vec{X_{rand}}$ is the position vector of a whale randomly chosen from the current population.

The WOA is a swarm-based meta-heuristic search strategy. It starts with a set of randomly generated solutions, and as iterations progress, each whale individual updates its position based on the optimal individual position in the current iteration or based on the position of a randomly chosen whale. In addition, the value of the parameter $\vec{a}$ is gradually decreases linearly from2 to 0 to provide bubble-net attacking method and searching for prey, respectively. The whale positions are randomly selected to update the positions of the remaining whales in each iteration when condition $|\vec{A}| \geq 1$ is met and the optimal position is chosen to update the positions of the remaining whales in each iteration when condition $|\vec{A}| < 1$ is met. Additionally, based on the random probability value p, the WOA can opt for either the shrinking encircle strategy or the spiral update strategy during position updates. Ultimately, the WOA algorithm concludes once specific termination criteria are met. The pseudo code of the WOA algorithm is presented in Figure 8.
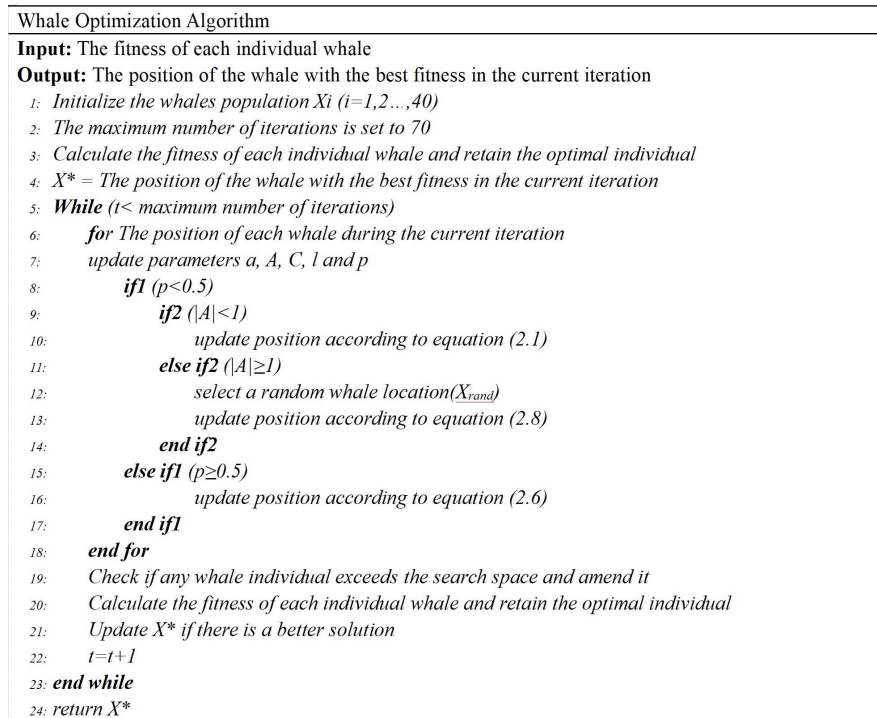
Whale Optimization Algorithm
Input: The fitness of each individual whale
Output: The position of the whale with the best fitness in the current iteration
1: *Initialize the whales population Xi (i=1,2.. ,40)*
2: *The maximum number of iterations is set to 70*
3: *Calculate the fitness of each individual whale and retain the optimal individual*
4: *X* = The position of the whale with the best fitness in the current iteration*
5: **While** *(t< maximum number of iterations)*
6:     **for** *The position of each whale during the current iteration*
7:     *update parameters a, A, C, l and p*
8:       **if1** *(p<0.5)*
9:         **if2** *(|A|<1)*
10:           *update position according to equation (2.1)*
11:         **else if2** *(|A|≥1)*
12:           *select a random whale location(X_{rand})*
13:           *update position according to equation (2.8)*
14:         **end if2**
15:       **else if1** *(p≥0.5)*
16:         *update position according to equation (2.6)*
17:       **end if1**
18:     **end for**
19:     *Check if any whale individual exceeds the search space and amend it*
20:     *Calculate the fitness of each individual whale and retain the optimal individual*
21:     *Update X* if there is a better solution*
22:     *t=t+1*
23: **end while**
24: *return X**

Figure 8. The process flowchart of the WOA algorithm.

4.3. Whale Optimization Algorithm Improvement

The traditional WOA is advantageous in terms of simplicity, few control parameters, the ability to avoid local optima, and has been widely applied in numerous domains. However, the WOA has certain limitations in terms of the convergence speed and precision. To address this issue, an adaptive WOA based on chaos mapping, referred to as CIWOABP, has been proposed.

4.3.1. Chaotic Mapping Whale Optimization Algorithm

Chaos mapping generates chaotic sequences through the repeated application of iterative functions. It exhibits nonlinearity, high sensitivity to initial values, excellent ergodicity, and unpredictability [18]. This mapping is frequently used for population initialization to enhance the algorithmic diversity and randomness [19,20]. Based on a comparison of 16 chaotic optimization algorithms in [21], the maximum Lyapunov exponent of cubic mapping outperforms those of sine mapping, circle mapping, Liebovitch mapping, intermittency mapping, singer mapping, and Kent mapping. It is comparable to the exponents of logistic, sinusoidal, tent, Gaussian, and Bernoulli mapping but lower than the exponents of Chebyshev, ICMIC, and piecewise mapping [22]. However, a holistic comparison revealed that the chaotic sequences generated by both Chebyshev and ICMIC mapping fell within the (-1,1) range. The formula complexity for piecewise mapping is high, and tent, Gaussian, and Bernoulli mapping use piecewise functions. Both sinusoidal and tent mappings tended to produce repetitive chaotic sequences. Therefore, considering all these factors, this study opted for cubic mapping to optimize the traditional whale algorithm. The formula is as follows:

$$x_{n+1} = \rho x_n(1 - x_n^2), \tag{10}$$

where $x_n \in (0,1)$, and ρ are control parameters.

4.3.2. Whale Optimization Algorithm for Combining Multiple Strategies

In the WOA, ensuring a balance between searching for prey and the bubble-net attack method is a key factor influencing the algorithm's performance and accuracy (i.e., the dynamic switching between the search processes of eq. 7 and eq. 9). By appropriately adjusting the switching parameters and search methods, the algorithm's flexibility and search efficiency can be enhanced [23]. The inertial weight, which acts as a balancing parameter, has been widely applied in various metaheuristic optimization algorithms [24,25]. To achieve a better balance between the preying and searching phases and enhance the algorithm's flexibility and search efficiency, we adjusted the value of the adaptive weight as follows:

$$\omega = (\omega_{\max} - \omega_{\min}) * \text{mm} * e^{(-\tau/t_{\max})} + \omega_{\min}, \tag{11}$$

where *mm* is the adjustment coefficient, τ is the current iteration number, and $t_{\max}$ is the maximum number of iterations. The adaptive weight gradually decreases as the number of iterations increases. In the first half of the search for prey and the bubble-net attack phase, a larger inertia weight can enhance the global search performance, prompting a deeper exploration of potential global optima by the algorithm, thus helping the algorithm avoid getting stuck in local optima. In the second half of the search, as the inertia weight decreases, the local search capability is strengthened, allowing the algorithm to find the optimum solution faster within a smaller search space, thereby accelerating the algorithm's convergence process. After incorporating the dynamic adaptive weight value adjustment, the improved strategies for encircling prey, bubble-net attacks, and searching for prey can be represented by the following equations:

$$\begin{cases} \vec{X}(t+1) = \omega\vec{X}^*(t) - \vec{A} \cdot \vec{D} & \text{if } p < 0.5, |A| < 1 \\ \vec{X}(t+1) = \omega\vec{D}' \cdot e^{bl} \cdot \cos(2\pi l) + \vec{X}^*(t) & \text{if } p \geq 0.5 \\ \vec{X}(t+1) = \omega\vec{X}_{\text{rand}} - \vec{A} \cdot \vec{D} & \text{if } p < 0.5, |A| \geq 1 \end{cases} \tag{12}$$

5. Temperature Error Compensation

5.1. Calibration Data for Drilling Pressure Parameters

A Honeywell ASDX100D44R pressure sensor and PT100-RTD temperature transmitter were used in the laboratory. In this experimental configuration, the temperature measurement range was set between 0°C and 78°C, with measurement points set at 6°C intervals, yielding a total of 14 measurement points. For pressure, the measurement range was from 40 kPa to 80 kPa, with measurement points arranged at 5 kPa intervals, resulting in a total of nine measurement points. Table 1 lists the calibration data for the pressure sensor, where p is the applied pressure in the experiment, Up is the output voltage signal, and Ut is the actual measured temperature.

Table 1. Pressure sensor calibration data.

p/kPa	0°C		6°C		12°C		18°C		24°C		30°C		36°C	
	Up/V	Ut/°C	Up/V	Ut/°C	Up/V	Ut/°C	Up/V	Ut/°C	Up/V	Ut/°C	Up/V	Ut/°C	Up/V	Ut/°C
40	83.24	0.34	83.41	6.59	86.95	12.45	87.123	18.57	88.56	24.51	90.46	30.21	91.36	36.32
45	95.71	0.37	96.37	6.61	98.36	12.51	101.54	18.62	103.18	24.56	104.84	30.41	106.12	36.41
50	104.81	0.42	106.03	6.65	108.03	12.56	111.32	18.65	112.24	24.62	113.87	30.43	115.54	36.52
55	113.35	0.46	115.62	6.73	118.37	12.58	120.51	18.67	122.56	24.75	123.67	30.47	125.17	36.56
60	124.41	0.57	129.08	6.76	130.09	12.62	131.86	18.71	133.23	24.78	134.51	30.62	135.62	36.61
65	136.46	0.63	138.53	6.79	141.39	12.65	143.23	18.73	144.04	24.83	146.13	30.64	147.86	36.65
70	147.88	0.74	149.38	6.83	151.31	12.68	152.64	18.78	153.62	24.87	155.05	30.71	156.63	36.68
75	155.78	0.78	157.25	6.87	158.55	12.71	160.46	18.81	162.87	24.76	164.05	30.75	165.74	36.71
80	164.84	0.81	167.72	6.93	168.31	12.73	172.04	18.83	173.16	24.51	174.83	30.83	176.04	36.76

p/kPa	42°C		48°C		54°C		60°C		66°C		72°C		78°C	
	Up/V	Ut/°C	Up/V	Ut/°C	Up/V	Ut/°C	Up/V	Ut/°C	Up/V	Ut/°C	Up/V	Ut/°C	Up/V	Ut/°C
40	92.65	42.65	94.14	48.36	95.65	54.36	96.04	60.24	96.74	66.45	97.45	72.24	98.24	78.46
45	107.84	42.68	109.23	48.39	110.36	54.39	110.87	60.26	111.63	66.47	112.31	72.26	113.79	78.48
50	117.01	42.71	118.54	48.42	119.65	54.42	121.24	60.34	121.87	66.51	122.47	72.29	123.14	78.51
55	126.74	42.73	128.24	48.46	129.41	54.48	130.63	60.39	131.42	66.56	131.89	72.34	132.41	78.53
60	137.13	42.76	138.76	48.51	139.87	54.51	140.74	60.42	141.36	66.59	141.76	72.37	142.32	78.57
65	149.21	42.82	150.63	48.56	151.64	54.53	152.68	60.47	153.13	66.62	153.89	72.41	154.65	78.62
70	158.14	42.86	159.64	48.63	161.13	54.56	161.97	60.51	162.75	66.63	163.45	72.46	164.27	78.65
75	167.16	42.87	168.74	48.68	169.65	54.57	170.12	60.58	170.84	66.69	171.53	72.48	171.82	78.68
80	176.48	42.93	177.96	48.74	179.21	54.63	180.04	60.64	180.76	66.84	181.46	72.53	181.76	78.74

5.2. Temperature compensation

In this study, we utilized Up and Ut as input parameters for the neural network, and the pressure p was set as the output value of the neural network. To ensure the efficiency and accuracy of network training, we selected pressure sensor calibration data at 0°C, 6°C, 18°C, 24°C, 36°C, 42°C, 54°C, 60°C, 72°C, and 78°C as the training dataset, while the data at 12°C, 30°C, 48°C, and 66°C were chosen as the test dataset. Furthermore, all data were normalized to ensure that the values fell between 0 and 1 and achieve optimal network performance. The constructed BP neural network structure was 2-5-1, with two input layers, five hidden layers, and one output layer. In the CIWOA configuration, we set an initial population size of 40 with a maximum iteration count of 70. The control parameter in the cubic mapping function was set to 1, and the adjustment coefficient mm in the adaptive weight adjustment function was set to 1. Because the results of the neural network training may vary with each iteration, we opted for multiple training sessions and ultimately utilized the compensated data from the test samples, as shown in Table 2.

Table 2. Test sample data after compensation.

P/kPa	12°C	30°C	48°C	66°C
40	40.4069	40.0503	39.9921	40.0371
45	44.7140	45.2678	45.2121	44.7404
50	49.7144	49.9058	49.9874	49.9548
55	55.1138	54.9168	55.0609	55.2089
60	60.2504	59.6878	59.8697	59.9747
65	65.0257	64.8659	65.1702	64.9708
70	70.3660	69.7658	69.9288	70.0387
75	74.8589	74.9841	75.1638	74.9393
80	79.6813	80.0090	79.9028	80.0323

5.3. Analysis of Results

To validate the performance of the proposed algorithm thoroughly, we employed five distinct neural network architectures: BP, WOABP, CWOABP, IWOABP, and CIWOABP. These networks were trained using test samples. A comparison between the values predicted by the BP neural network before and after optimization with each algorithm and the actual values is illustrated in Figure 9. An

error comparison of the BP neural network before and after optimization with each algorithm is shown in Figure 10.
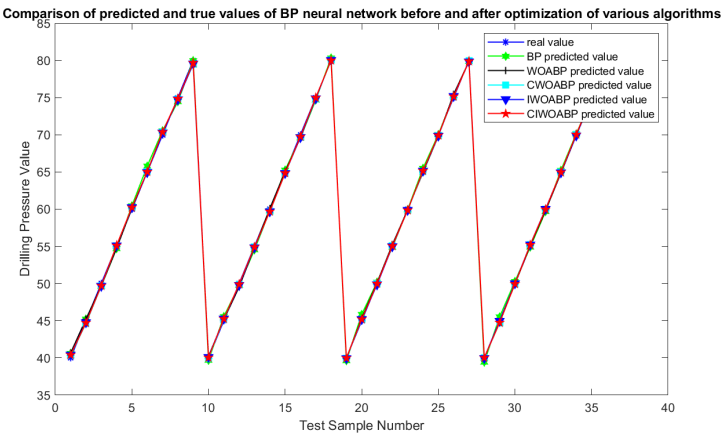

Figure 9. Comparison of predicted and true values of BP neural network before and after optimization of each algorithm.
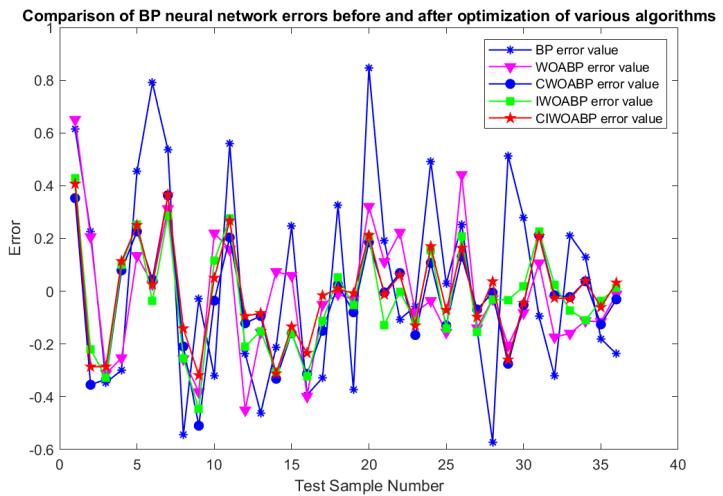

Figure 10. Comparison of BP neural network errors before and after optimization of each algorithm.

In Figure 9 and Figure 10, we can observe that, compared to the other optimization methods, the error of the temperature compensation using the CIWOABP neural network is noticeably smaller, with the error range consistently within 0.4. We listed the error comparison results for the different temperature compensation models in Table 3. to further evaluate the performance of each algorithm.

Table 3. Test sample data after compensation.

Digital	Sensitivity Temperature Coefficient	Mean Absolute Error	Mean Square Error	Mean Absolute Percentage Error
BP	0.0019/°C	0.42519	0.2226	0.76558%
WOABP	0.00093823/°C	0.18471	0.055066	0.33335%
CWOABP	0.00075132/°C	0.16081	0.039309	0.28167%
IWOABP	0.00090298/°C	0.18337	0.062036	0.3247%
CIWOABP	0.00065372/°C	0.14052	0.032773	0.25286%

Based on Table 3, the temperature compensation model using CIWOABP outperformed the other models across various evaluation metrics, including the sensitivity temperature coefficients, mean absolute error, mean square error, and mean absolute percentage error.

6. Conclusions

In this study, a sophisticated temperature compensation technique that leverages a CIWOABP neural network was introduced. This approach synthesizes the unmatched precision of the BP neural network with the streamlined control parameters of the WOA due to its advantages in circumventing local optima. It also capitalizes on the swift convergence attributes of the CWOA and draws on the expansive global search and meticulous local refinement capacities of the IWOA algorithm. It boasts advantages such as exceptional accuracy, innate resilience against being ensnared in local optima, rapid convergence velocities, and formidable capabilities in global searching and local refinement. This effectively addresses the conventional limitations of the BP neural network, such as the environment in local optimal solutions and lethargic convergence. The refined pressure sensor measurement system achieved a sensitivity temperature coefficient of $6.537210 - 4/$, and the average absolute percentage deviation stood at a mere 0.25286%, satisfying the pragmatic operational benchmarks.

Author Contributions: Fei Wang was responsible for the overall structure of the article and the design of the algorithm. Xing Zhang and GuoWang Gao took charge of the experimental section in the paper and the writing of the article and Xintong Li assisted with the experiments and provided linguistic modifications.

References

1. Bai, J.; Zhang, B.; Zhang, C.P. Status quo and development proposal of ultra-deep and ultra-small borehole directional drilling technology. *Drilling Process* **2018**, *41*, 5–8+143.
2. Zhu, Z.F.; Zhang, H.N. Research on nonlinear calibration and temperature compensation method of pressure transmitter. *Electronic Measurement Technology* **2021**, *44*, 71.
3. Wu, C.H.; Jiang, R.W. Temperature compensation method for pressure sensor based on PSO-LM-BP neural network. *Instrumentation Technology and Sensors* **2018**, pp. 129–133.
4. Liu, H.; Li, H.J. Research on temperature compensation method of pressure sensor based on BP neural network. *Journal of Sensing Technology* **2020**, *33*, 688–692+732.
5. Wang B B, L.H.J. Temperature compensation of silicon piezoresistive pressure sensor based on cubic spline interpolation. *Journal of Sensing Technology* **2015**, *28*, 1003–1007.
6. He, R.; Xiao, X.; Kang, Y.; Zhao, H.; Shao, W. Heterogeneous Pointer Network for Travelling Officer Problem. In Proceedings of the 2022 International Joint Conference on Neural Networks (IJCNN). IEEE, 2022, pp. 1–8.
7. He, X.; Trigila, C.; Ariño-Estrada, G.; Roncali, E. Potential of depth-of-interaction-based detection time correction in Cherenkov emitter crystals for TOF-PET. *IEEE Transactions on Radiation and Plasma Medical Sciences* **2022**, *7*, 233–240.
8. Ding H, Li J, S.H.L.e.a. Dual-path temperature compensation technique for fiber optic transformer based on the least squares method. *Electronic Measurement Technology* **2016**, *39*, 190–195.

9. Lv M C, L Q Q, S.X.e.a. Design of infrared temperature sensor based on GA-BP neural network temperature compensation. *Instrumentation Technology and Sensors* **2019**, *03*, 19–22.

10. Ma, H.G.; Zeng, G.H.; Huang, B. Research on temperature compensation of pressure transmitter based on WOA-BP. *Instrumentation Technology and Sensors* **2020**, pp. 33–36.

11. Zhou, Z.W.; Deng, T.Y.; Shi, L.; et al. Temperature compensation and field calibration method for piezoresistive pressure sensors. *Sensors and Microsystems* **2022**, *41*, 145–148.

12. Liu, H. Optimal selection of control parameters for automatic machining based on BP neural network. *Energy Reports* **2022**, *8*, 7016–7024.

13. Zhang, R.; Duan, Y.; Zhao, Y.; et al. Temperature Compensation of Elasto-Magneto-Electric (EME) Sensors in Cable Force Monitoring Using BP Neural Network. *Sensors* **2018**, *18*, 2176.

14. Shao, W.; Jin, Z.; Wang, S.; Kang, Y.; Xiao, X.; Menouar, H.; Zhang, Z.; Zhang, J.; Salim, F. Long-term spatio-temporal forecasting via dynamic multiple-graph attention. *arXiv preprint arXiv:2204.11008* **2022**.

15. Mirjalili, S.; Lewis, A. The whale optimization algorithm. *Advances in Engineering Software* **2016**, *95*, 51–67.

16. Xu, D.G.; Wang, Z.Q.; Guo, Y.X.; et al. A review of research on whale optimization algorithms. *Computer Application Research* **2023**, *40*, 328–336.

17. Sun, Y.; Wang, X.; Chen, Y.; et al. A modified whale optimization algorithm for large-scale global optimization problems. *Expert Systems with Applications* **2018**, *114*, 563–577.

18. Kang, Y.; Rahaman, M.S.; Ren, Y.; Sanderson, M.; White, R.W.; Salim, F.D. App usage on-the-move: Context-and commute-aware next app prediction. *Pervasive and Mobile Computing* **2022**, *87*, 101704.

19. Li, M.; Xu, G.; Lai, Q.; et al. A chaotic strategy-based quadratic Opposition-Based Learning adaptive variable-speed whale optimization algorithm. *Mathematics and Computers in Simulation (MATCOM)* **2022**, *193*.

20. Wu, G.; Li, Y. Non-maximum suppression for object detection based on the chaotic whale optimization algorithm. *Journal of Visual Communication and Image Representation* **2023**, *74*.

21. Feng, J.H.; Zhang, J.; Zhu, X.S.; et al. A novel chaos optimization algorithm. *Multimedia Tools & Applications* **2016**.

22. Zhang, M.j.; Zhang, H.; Chen, X.; et al. Gray wolf optimization algorithm based on Cubic mapping and applications. *Computer Engineering and Science* **2021**, *43*, 2035–2042.

23. Deng, H.; Liu, L.; Fang, J.; et al. A novel improved whale optimization algorithm for optimization problems with multi-strategy and hybrid algorithm. *Mathematics and Computers in Simulation (MATCOM)* **2023**, *205*.

24. Rauf, H.T.; Malik, S.; Shoaib, U.; et al. Adaptive inertia weight Bat algorithm with Sugeno-Function fuzzy search. *Applied Soft Computing* **2020**, *90*, 106–159.

25. Li, X.N.; Wu, H.; Yang, Q.; et al. A multistrategy hybrid adaptive whale optimization algorithm. *Journal of Computational Design and Engineering* **2022**, *9*, 1952–1973.